
Bridging

Hubs vs Bridging, Interconexão de LANs, VLANs,
Spanning Tree

PEDRO MARTINS

April 15, 2018

Contents

1	Hubs vs Switches	3
1.1	Repeaters/Hubs	3
1.2	Bridges/Switches	3
1.3	Bridges/switches vs repeaters/hubs	6
1.4	Bridges	8
1.4.1	Address Learning	8
	Tabela de Encaminhamento	9
	TTL - Time to Live of a MAC Address Table ENtries	9
2	Interconexão de LANs	9
2.1	Routers	12
2.2	Domínios de colisão e broadcast	13
2.2.1	Como é que os switches respondem a colisões?	13
3	VLANs - Virtual LANs	14
3.1	Ligação entre switches	15
3.1.1	IEEE802.1Q Standard	17
3.1.2	Redudância	17
4	Spanning tree	17
4.1	Bellman Equations	21
4.2	Forwarding baseado em Spanning Trees	23
4.3	Funcionamento da Spanning Tree	25
4.3.1	Exemplo de Mudança	30
4.4	Protocolo IEEE 802.1D	30
4.5	Manutenção da Spanning Tree	30
4.5.1	Order of configuration messages	32
4.6	Construção da Spanning Tree	33
4.6.1	Exemplo	35
4.7	Breakdown em bridges ou LANs	36
4.7.1	Ciclos temporários	36
4.8	Bridge Port States	36
4.8.1	TTL of entries of the forwarding tables	40
	Exemplo de uma spanning tree ineficiente	40
4.9	Protocolos	43
4.9.1	802.1p	43
4.9.2	802.1w Rapid Spanning Tree Protocol	43
4.9.3	802.1s - Multiple Spanning Tree Protocol	43

1 Hubs vs Switches

1.1 Repeaters/Hubs

- Um repetidor numa rede de telecomunicações opera como um decodificador + amplificador
 - Decodifica a informação que lhe chega
 - * Amostra
 - * Quantifica
 - * Converte de analógico para digital
 - Amplifica
 - * Amplifica a potência do sinal
 - * transmite
- Opera na **1ª camada, bit a bit**
- Unidade de operação/trabalho: bit
- Implica que o meio só possa ser **usado** por uma estação de cada vez
- Interligam segmentos do mesmo tipo de LANs, i.e., interligam diferentes segmentos da mesma LAN
- A largura de banda agregada é limitada pela taxa de transmissão da LAN
- Os bits vão sendo transmitidos à medida que vão chegando
- A bitrate é partilhada globalmente
- Opera no nível físico
- Regenera os sinais
- Um **hub** é um repetidor com múltiplas portas
- Pode fazer a conversão entre meios físicos de propagação do sinal
 - e.g., elétrico <-> ótico

1.2 Bridges/Switches

- Analiso o pacote quando chega
 - Tomo decisões com base no destinatário
 - * Se o destinatário está na mesma porta que o emissor, não envio o pacote para mais nenhuma porta
 - * Se o destinatário esta na porta x, que não coincide com a mesma porta que o emissor, envio para apenas para a porta x
 - método de **Store and Forwarding + Filtering** aos pacotes
 - * Selectividade da porta de destino
 - * Permite que as portas operem a diferentes taxas de transmissão
 - * Não envia os pacotes para uma porta se não precisar de o fazer
- Unidade de operação/trabalho: pacote
- Interligam diferentes **LANs**
- Mais complexos que os **repeaters/hubs**

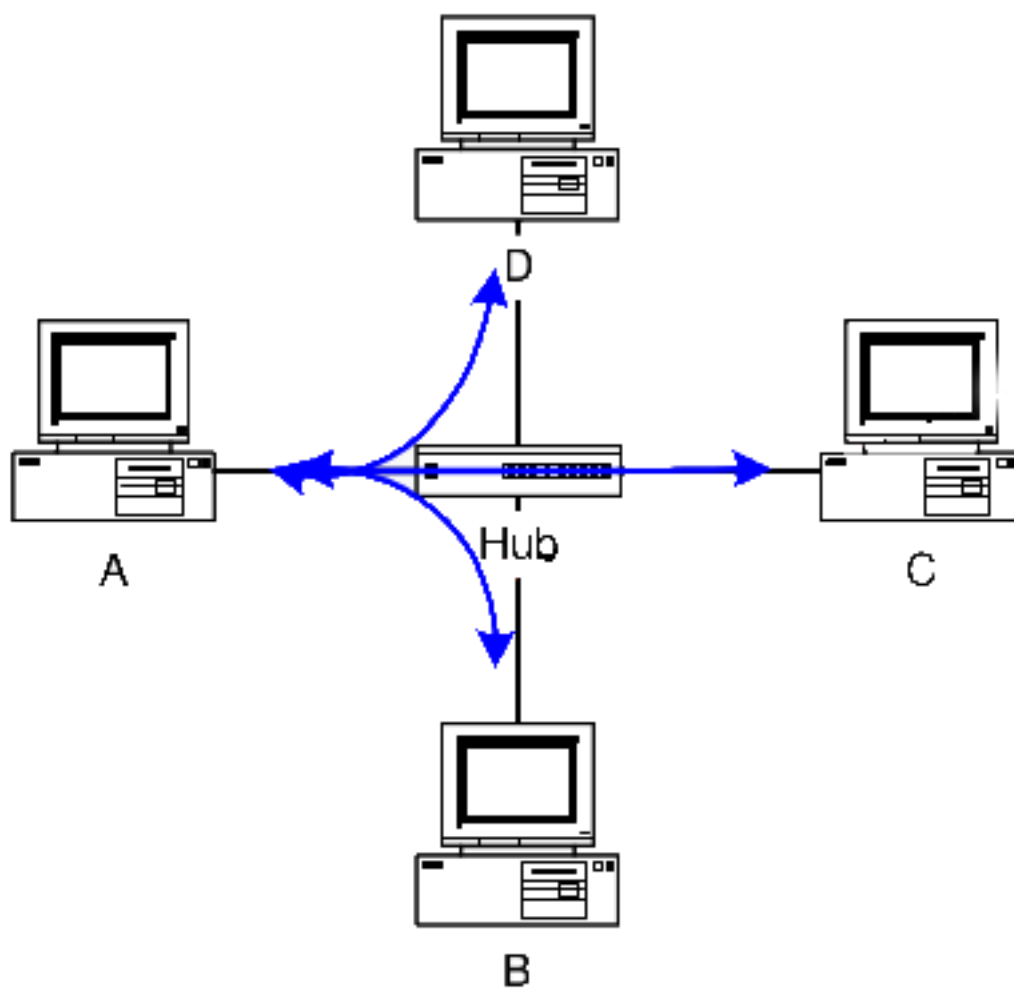


Figure 1: Exemplo de rede com Hub em estrela

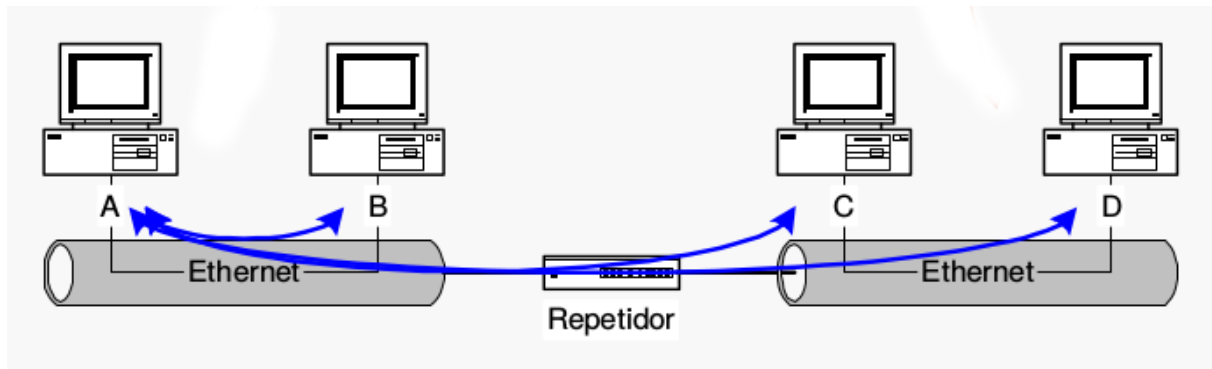


Figure 2: Exemplo de rede com Repeater em bus

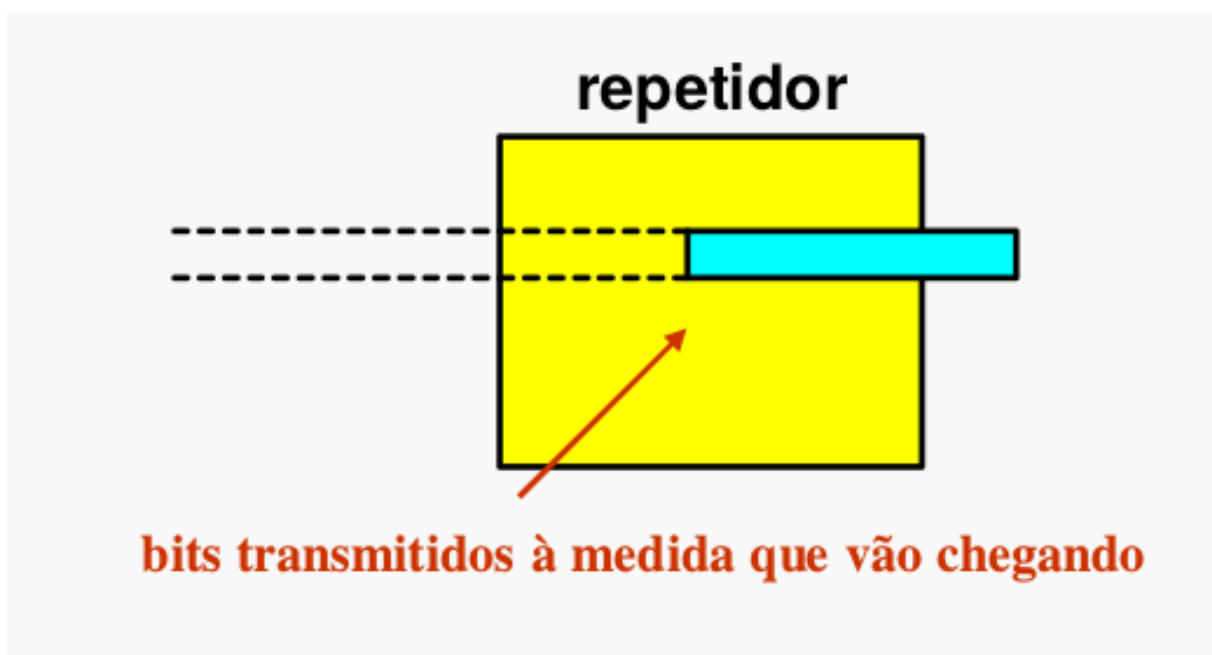


Figure 3: Operação de um repeater

- Um switch é uma *bridge multiple port*
- Um switch possui a noção de estados
- Tipo *store-and-forward*
- Opera ao nível da camada da ligação (camada 2)
- Interliga dois ou mais domínios de colisões
- Comuta com base nos endereços MAC

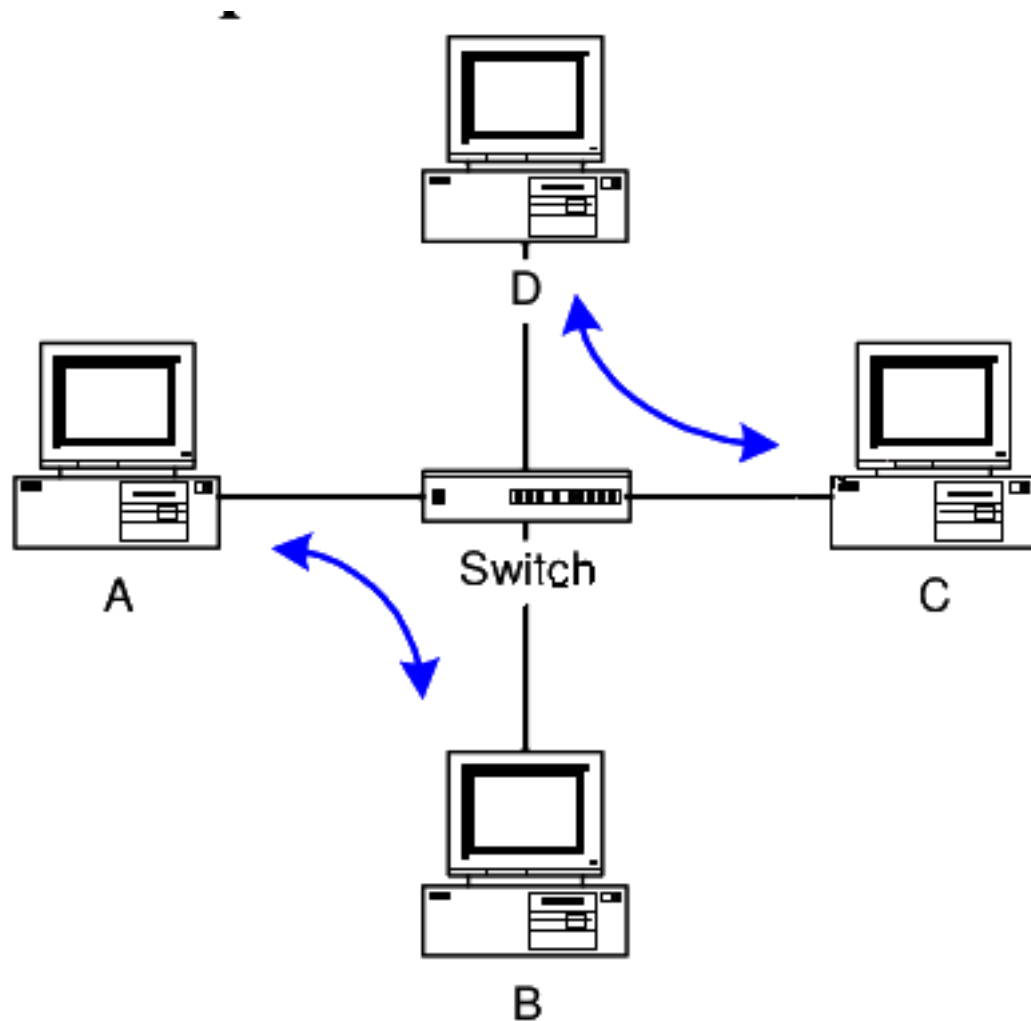


Figure 4: Exemplo de rede com Hub em estrela

1.3 Bridges/switches vs repeaters/hubs

Vantagens de usar bridges/switches:

- As colisões deixam de ser um problema

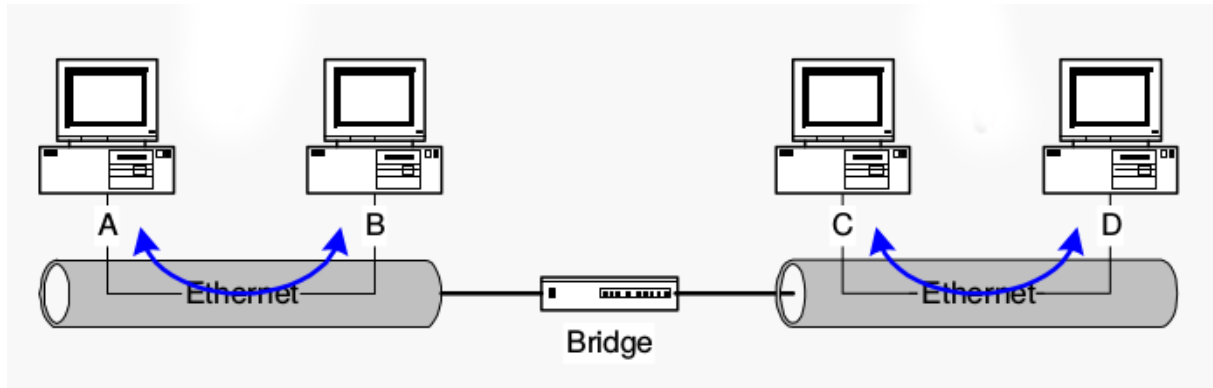


Figure 5: Exemplo de rede com Repeater em bus

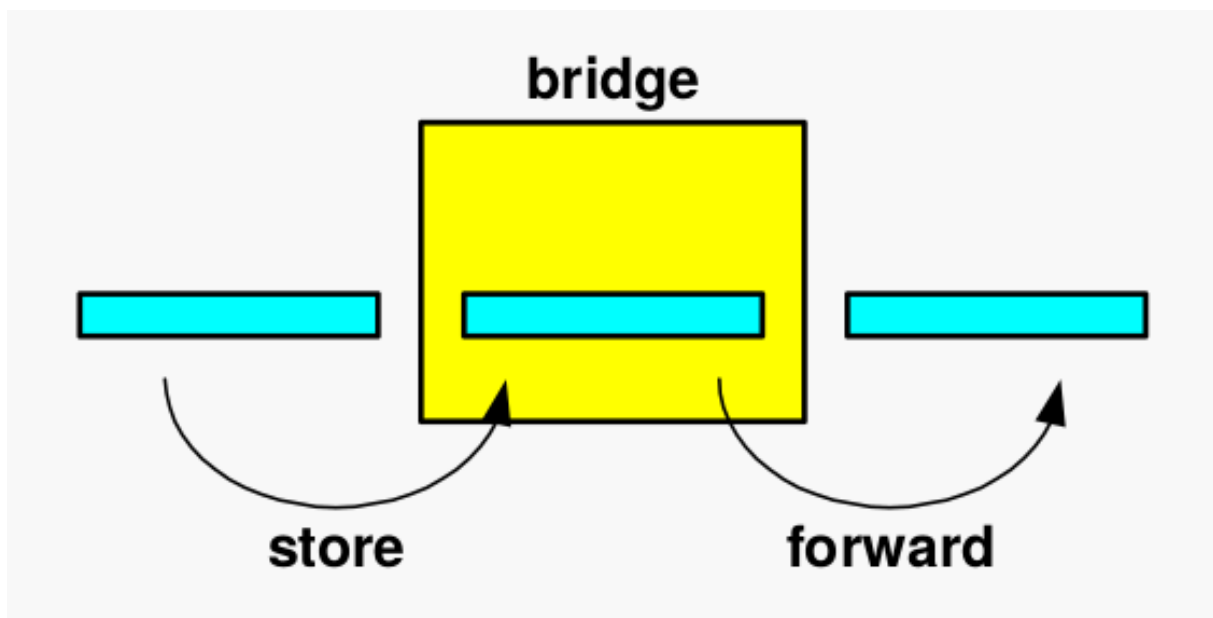


Figure 6: Operação de uma bridge

- A largura de banda é agregada e não é limitada pela taxa de transmissão das portas
 - Desde que não se conectem LANs, a largura de banda é multiplicada
- Várias estações podem transmitir em simultâneo se o seu emissor e destinatário forem diferentes

1.4 Bridges

1.4.1 Address Learning

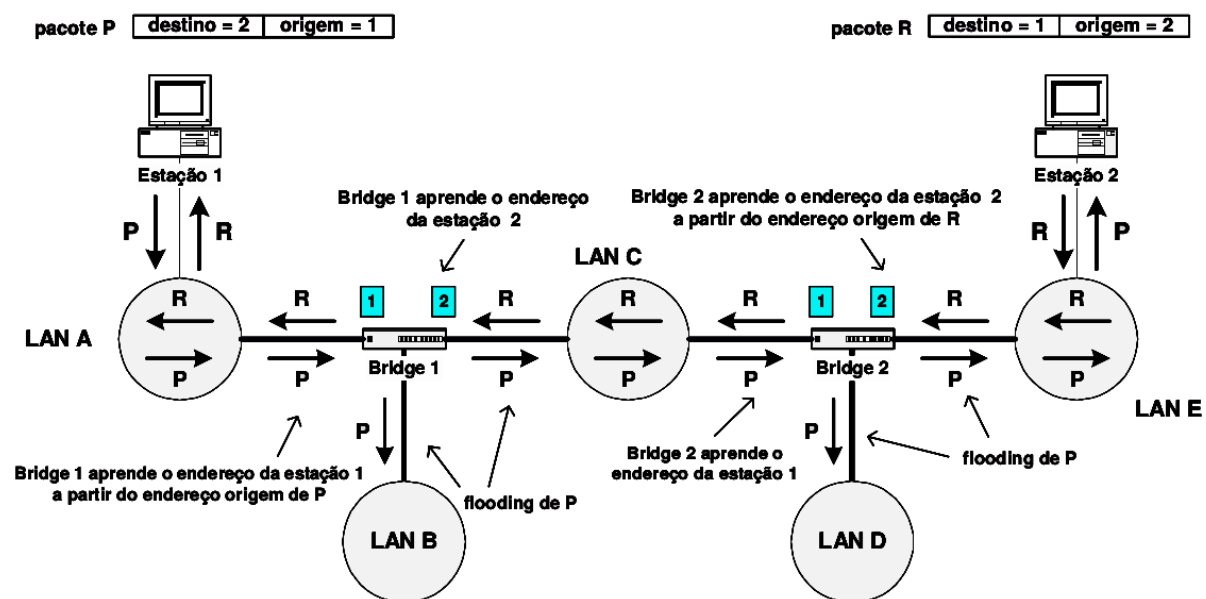


Figure 7: Exemplo ilustrativo do fenómeno de address learning em bridges

1. A estação 1 quer comunicar com a estação 2
2. Envia um pacote P para a LAN A
3. O pacote P chega à bridge 1
4. A bridge 1 aprende o endereço da estação 1, usando o **source MAC address** do pacote P
5. A bridge 1, que desconhece o endereço da estação 2, efetua um **flooding** do pacote P
6. O pacote P é enviado para a LAN B, de onde não obtém resposta
7. O pacote P é enviado para a LAN C, chegando à bridge 2
8. A bridge 2 aprende o endereço da estação 1, usando o **source MAC address** do pacote P
9. A bridge 2 não conhece o endereço de destino do pacote P
10. A bridge 2 efetua o flooding de P para a LAN D e para a LAN E
11. Na LAN E está ligada a estação de destino, que reconhece o seu endereço MAC
12. A estação 2 responde ao pacote P recebido, enviando o pacote R
13. A bridge 2, como já conhece o endereço de destino, faz o **forwarding** do pacote R da LAN E para a LAN C
 - A LAN D nunca vai receber o pacote R, porque a **bridge 2** sabe qual a porta para onde deve encaminhar os endereços para a estação 1

14. A bridge 1, como já conhece o endereço de destino, faz o **forwarding** do pacote R para a LAN A
15. A estação 1, ligada à LAN A, recebe o pacote de destino

Quando uma bridge recebe uma trama MAC numa porta de entrada:

- regista na sua tabela de encaminhamento a **porta** em que recebeu a trama e o **endereço MAC** de origem da trama
- procura o endereço MAC de destino na **trama** da tabela de encaminhamento, para reencaminhar a trama
- se não tiver o endereço de destino efetua **flooding**
- Se tiver o endereço MAC
 - Se for na mesma interface, descarta o pacote
 - Se for noutra interface, efetua **forwarding**
- **flooding**: broadcast de um pacote para todas as portas do **switch**, exceto a de origem do pacote, quando este desconhece a porta que deve usar para atingir o endereço de destino. É um processo que é efetuado para o **switch** aprender a que porta se encontra ligada a LAN que contém uma estação com o endereço MAC de destino do pacote. Só acontece quando a tabela de encaminhamento do switch não possui o endereço MAC de destino na sua tabela de encaminhamento. Comporta-se como um repetidor
- **forwarding**: encaminhamento de um pacote de uma porta do **switch** para outra porta do **switch**, usando a porta indicada na tabela de encaminhamento do switch. A bridge envia apenas a trama pela porta registada na tabela.

Tabela de Encaminhamento

endereço MAC interface tempo de vida ——— ——— ——— ———

: Exemplo da organização das tabelas de encaminhamento nos switches

TTL - Time to Live of a MAC Address Table ENtries

Após a troca da estação 2 da LAN E para a LAN D, a estação 2 deixa de receber pacotes porque as entradas das tabelas de switch das bridges continuam a indicar que devem enviar os pacotes para a LAN E.

As tabelas de encaminhamento só vão ser atualizadas se:

- a estação 2 transmitir
- o tempo de vida (**TTL**) das entradas na tabela de encaminhamento expirar
 - introduz-se mecanismos de aging no sistema
 - é uma forma de evitar bloquear um sistema que tem a noção de estado
 - se a bridge não receber nada da estação durante 30 segundos, remove-a

2 Interconexão de LANs

- A LAN 1 e a LAN 2 pertencem ao mesmo domínio de colisão
 - Estão na mesma camada de nível 2

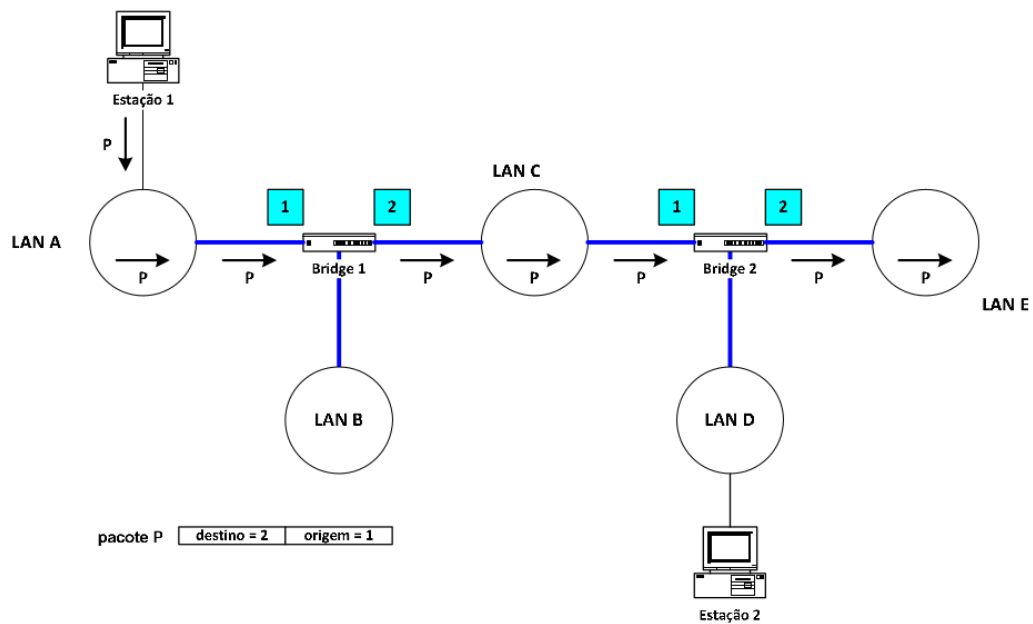


Figure 8: Tempo de vida na tabela de swtiching

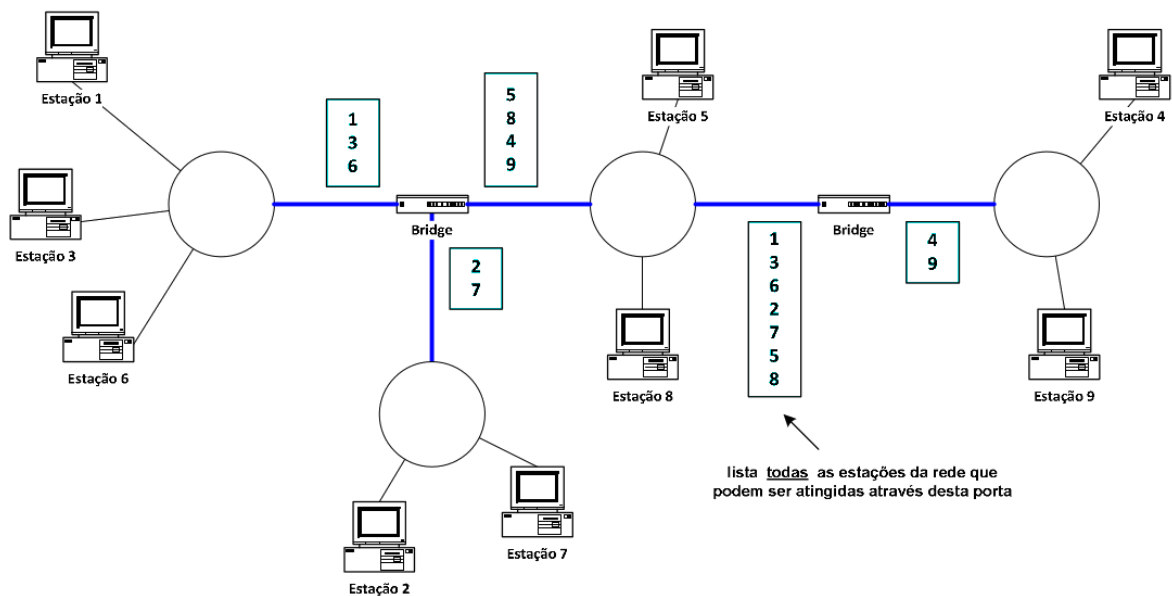


Figure 9: Exemplo das portas conhecidas para uma rede de switches

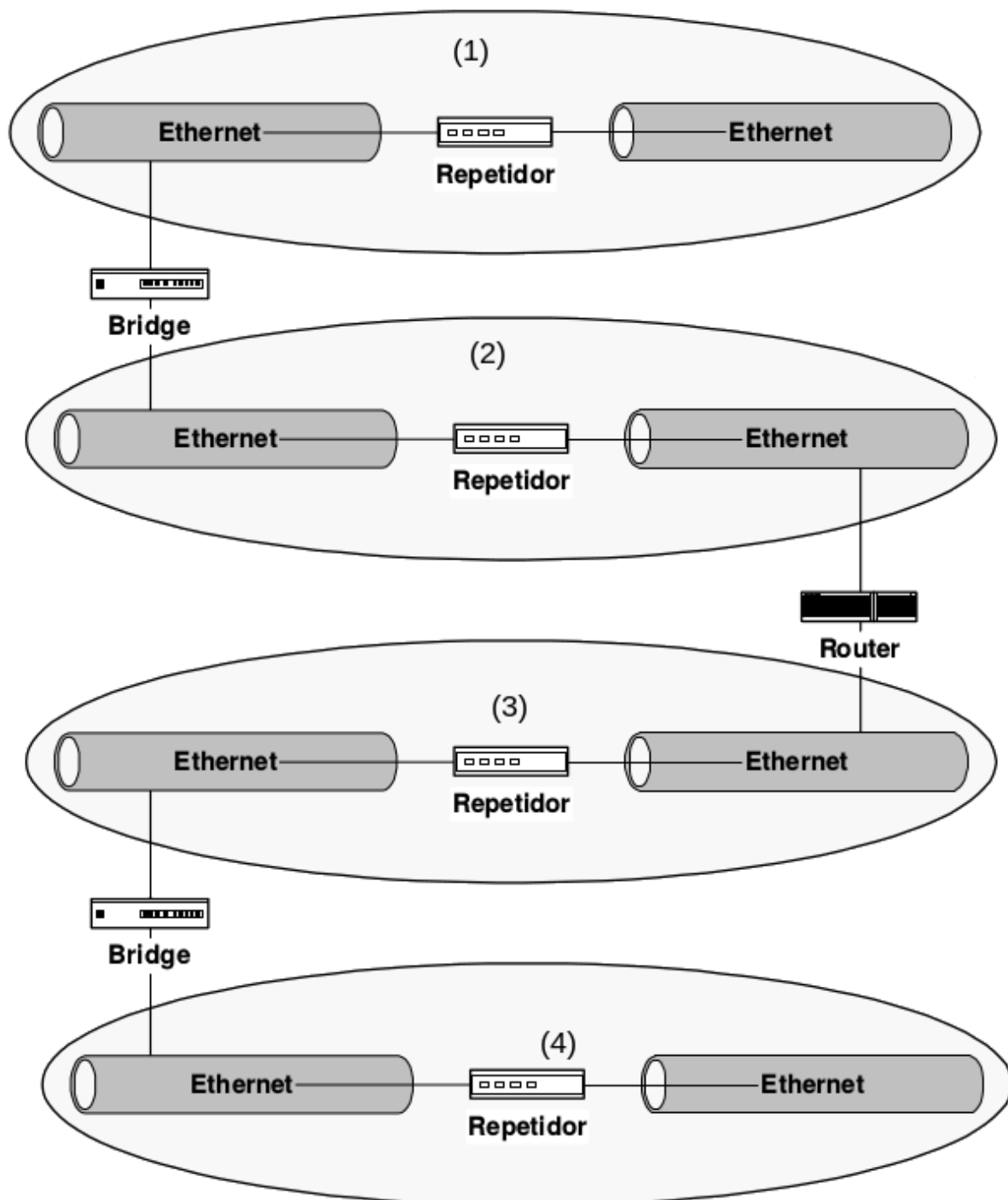


Figure 10: Exemplo de Interconexão de LANs

- A LAN 3 e a LAN 4 pertencem ao mesmo domínio de colisão
 - Estão na mesma camada de nível 2
- Os segmentos de Ethernet, dentro de cada LAN, estão ligados por um repetidor, por isso recebem os mesmos pacotes
- O router separa duas redes IP diferentes
 - com diferentes domínios de **broadcast**

Através da figura 11, são apresentadas as camadas do protocolo OSI em que operam as bridges e os routers.

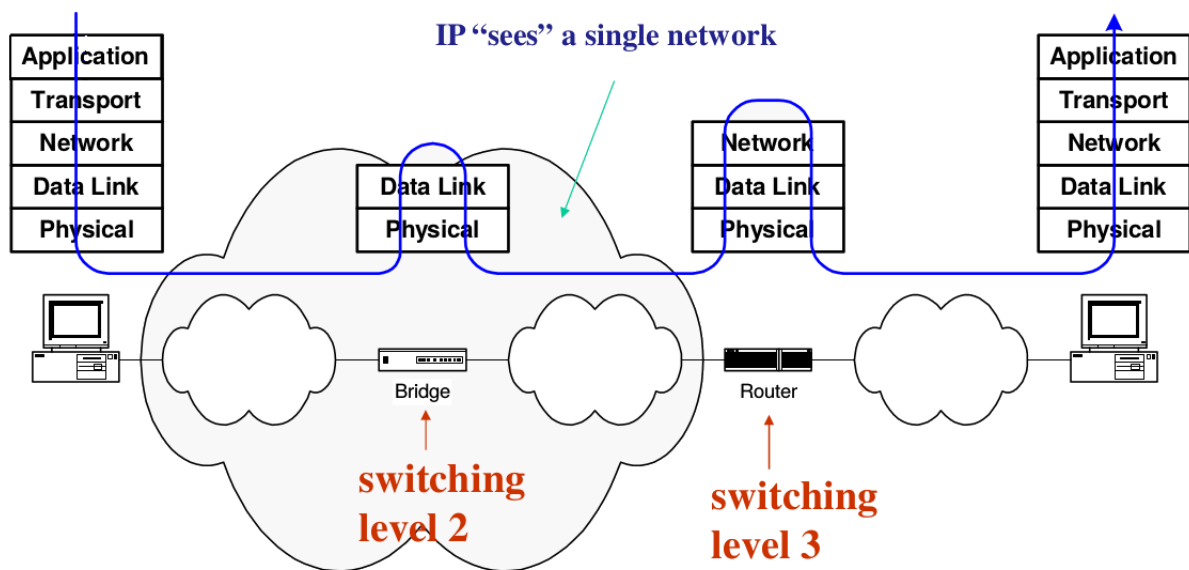


Figure 11: Comparação entre um router e switch

2.1 Routers

- Opera na camada 3
 - Camada IP
 - Nível da rede
- Efetua **level 3 "switching"**
 - O termo correto é **routing**
- tipo **store and forward**
- Comuta com base nos endereços de nível 3
 - IP
 - IPX
 - AppleTalk
 - etc.

2.2 Domínios de colisão e broadcast

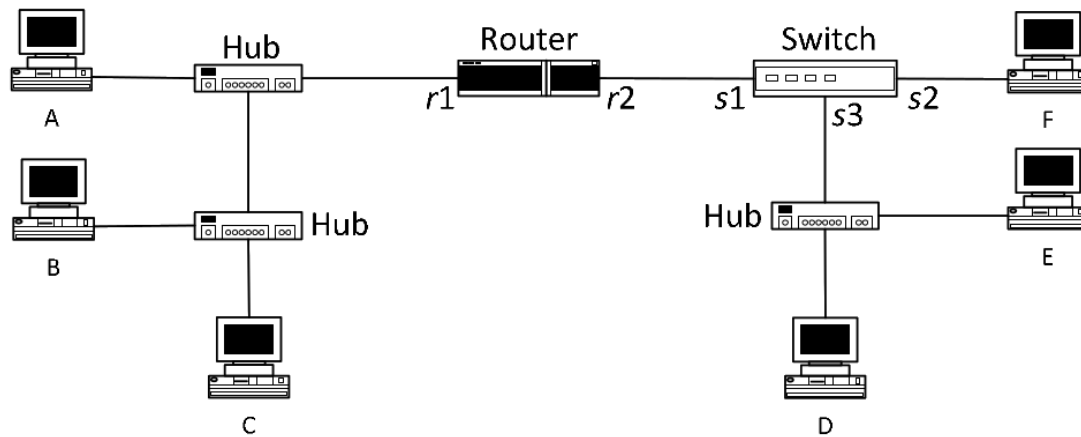


Figure 12: Domínios de colisão e de Broadcast

Na figura ?? existem

- Dois domínios de **broadcast**:
 - A, B, C e r_1
 - D, E, F e r_2
- Dois domínios de colisão:
 - interfaces **full-duplex**
 - * A, B, C, e r_1
 - * D, E e s_3
 - interfaces **half-duplex**
 - * A, B, C, r_1 , D, E, s_3
- Não existem colisões entre as interfaces
 - se forem **full-duplex**
 - * (r_2, s_1)
 - * (s_2, F)

Os pacotes só atravessam o switch se:

- forem **unicast** para a **bottom layer**
- forem **broadcast**

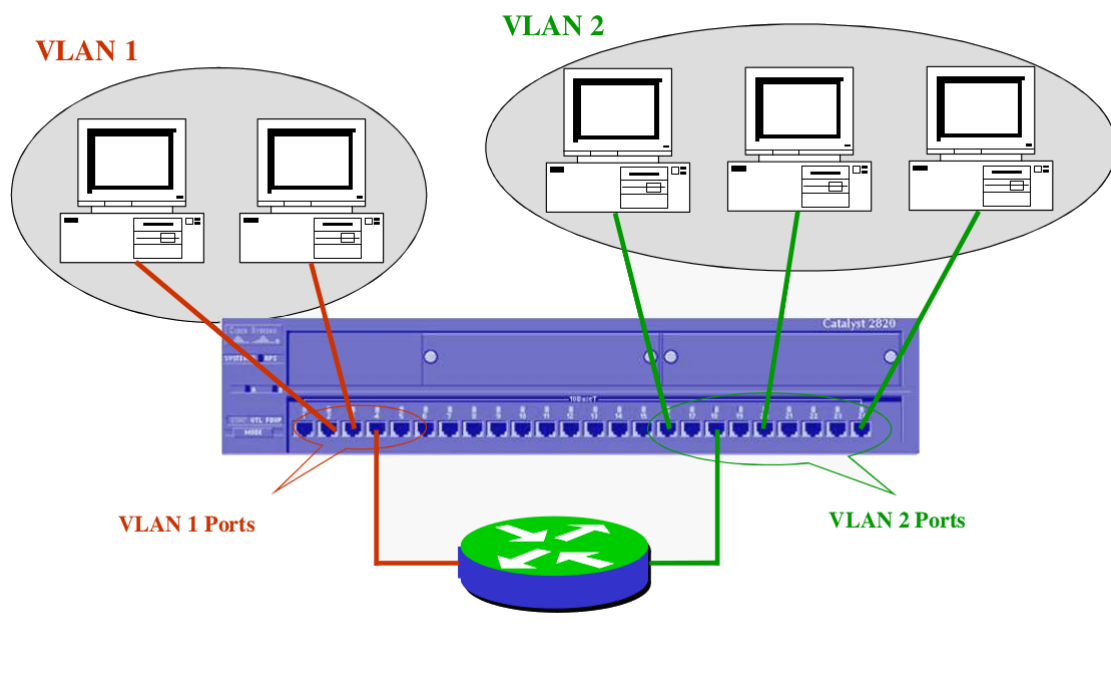
2.2.1 Como é que os switches respondem a colisões?

- **Full-duplex**:

- As entidades são capazes de comunicar ao mesmo tempo
- não ocorre conflito
- **half-duplex**:
 - Sou capaz de ou ouvir ou falar
 - Se um dispositivo fala, os outros ouvem
 - Podem ocorrer conflitos
 - * Duas estações a tentarem comunicar para o mesmo destino ao mesmo tempo
- Se a linha estiver cheia, os pacotes não são enviados
 - \Rightarrow Saturação do link na rede
 - \Rightarrow buffer enche
 - \Rightarrow Data é perdida

3 VLANs - Virtual LANs

- No mesmo **switch** posso declarar a existência de diferentes VLANs
- Duas VLANs diferentes nunca comunicam entre si
 - são totalmente independentes
 - não estão na mesma rede
 - não é possível fazer ping
- é necessário criar ligações físicas entre as VLANs, usando um router
- Se nunca quiser ligar dispositivos entre VLANs diferentes, posso possuir os mesmos endereços locais



3.1 Ligação entre switches

- ou usa uma porta física, figura ??
- requer uma porta física especial, *interswitch port*, figura ??
 - protocolo IEEE802.1Q
 - requer o uso de *VLAN tags*
 - * permitem identificar diferentes VLANs, através do VLAN ID
 - * permite multiplexar temporalmente a informação no mesmo cabo

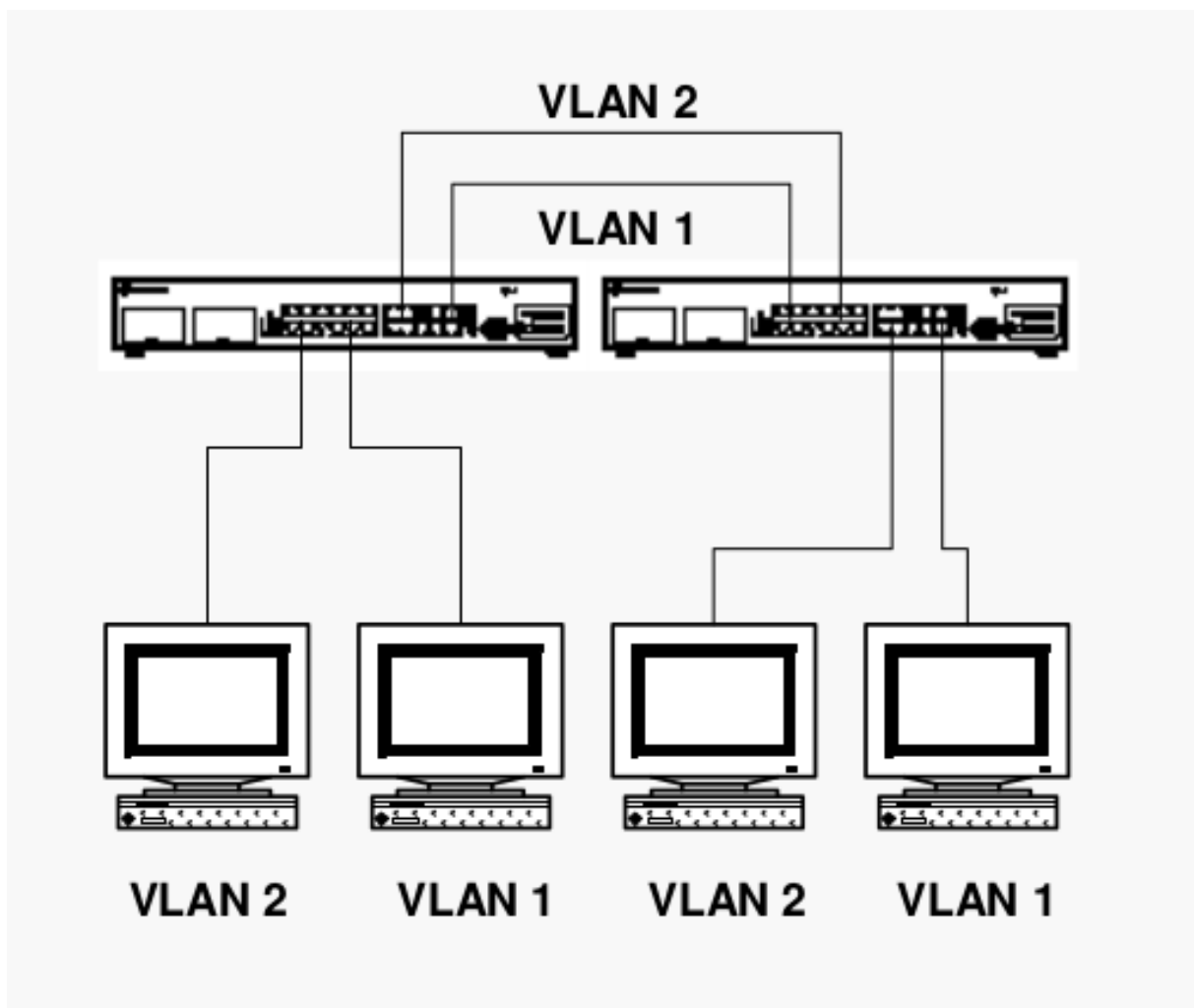


Figure 13: Ligação física entre duas VLANs em dois switches diferentes

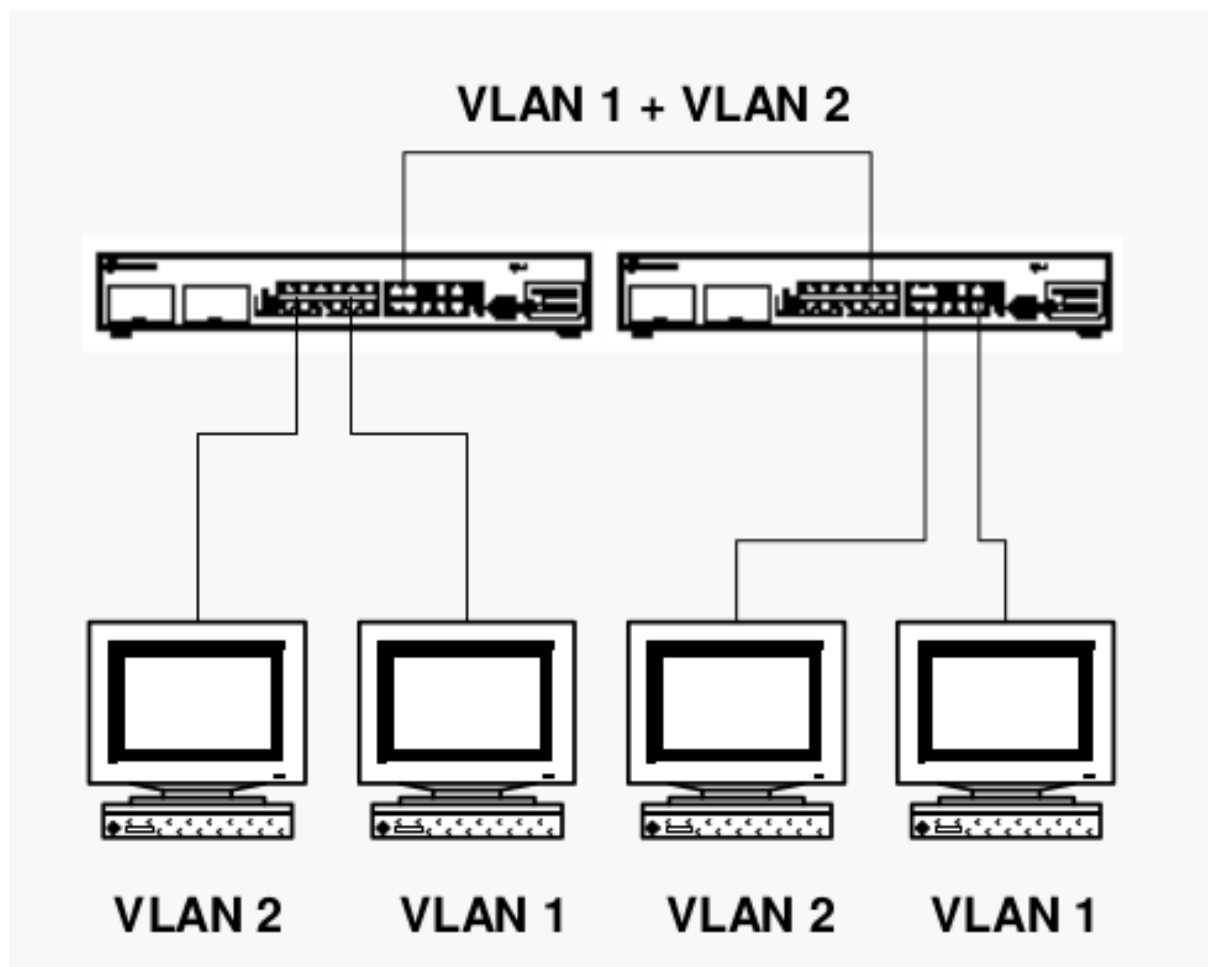


Figure 14: Ligação usando uma porta interswitching entre dois switches

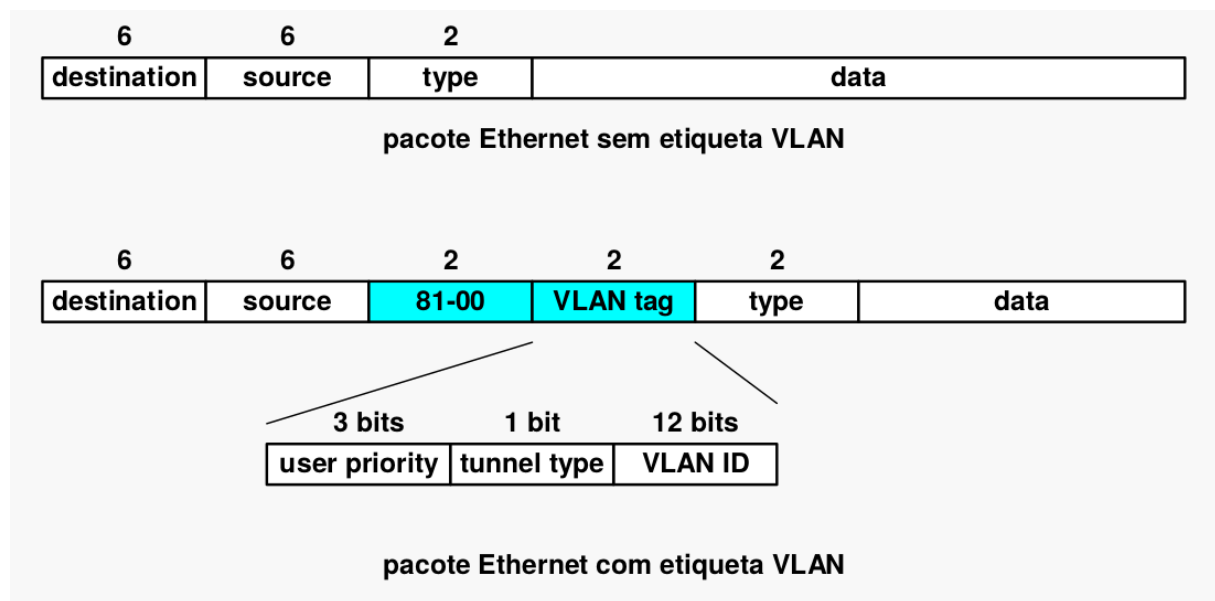


Figure 15: Pacote Ethernet com e sem etiqueta VLAN

3.1.1 IEEE802.1Q Standard

3.1.2 Redudância

A interligação de bridges/switches com redudância permite dotar a rede de mecanismos de **recuperação de falhas na rede**.

No entanto, pode trazer problemas ao criar mecanismos de feedback positivo na rede, como por exemplo, o colapso das ligações devido à saturação na rede se ocorrer um broadcast

4 Spanning tree

- **Objetivo:** Numa rede que pode ter ligações redundantes entre switches, escolher que portas usar para efetuar as ligações entre várias LANs

1. A estação 1 quer comunicar com a estação 2
2. Envia um pacote P para a LAN A
3. O pacote P chega à bridge 1
4. A bridge 1 aprende o endereço da estação 1, usando o **source MAC address** do pacote P
5. A bridge 1, que desconhece o endereço da estação 2, efetua um **flooding** do pacote P
6. O pacote P é enviado para a LAN B
7. Na LAN B, o pacote não é enviado pela bridge 3 porque a porta ligada à LAN B está inativa, em resultado do algoritmo da Spanning Tree
8. O pacote P é enviado para a LAN C, chegando à bridge 2
9. A bridge 2 aprende o endereço da estação 1, usando o **source MAC address** do pacote P

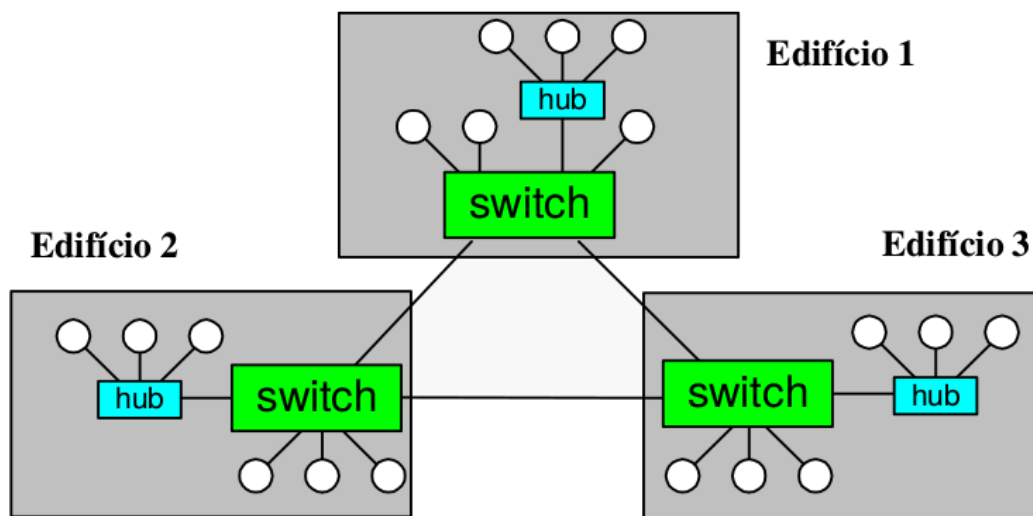


Figure 16: Exemplo de rede com ligações redundantes entre switches

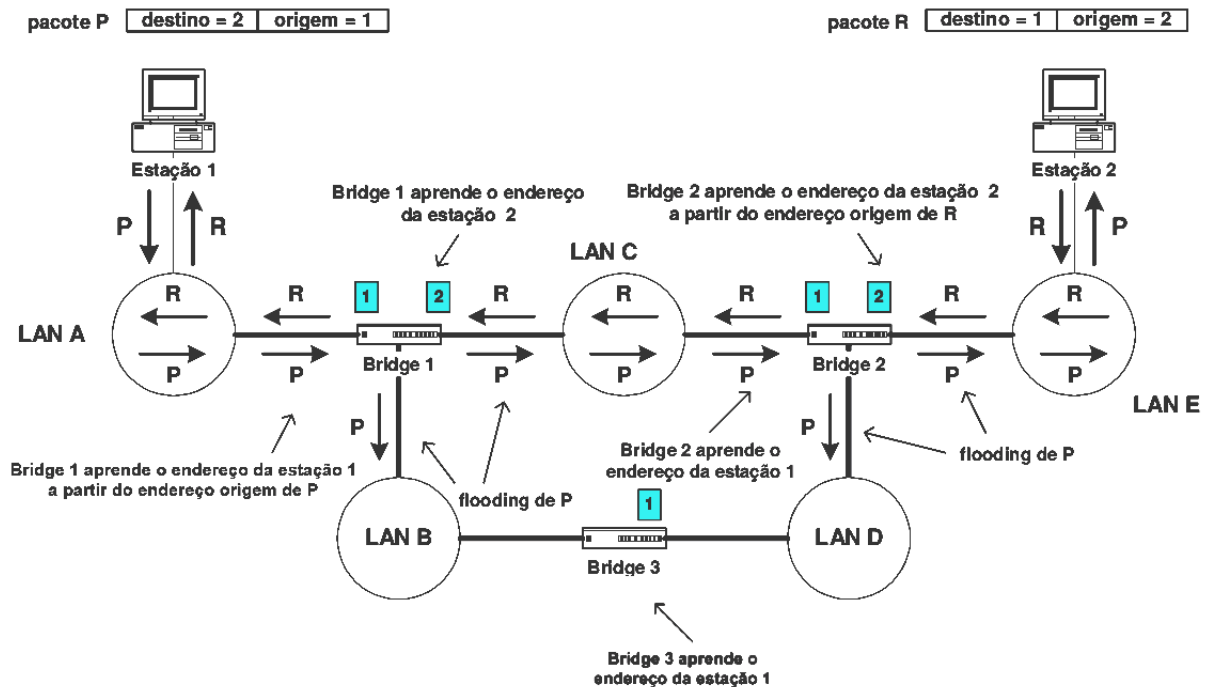


Figure 17: Exemplo do funcionamento de uma rede com Spanning Tree ativo

10. A bridge 2 não conhece o endereço de destino do pacote P
11. A bridge 2 efetua o flooding de P para a LAN D e para a LAN E
12. Na LAN D, a bridge 3 aprende o endereço da estação 1
13. Na LAN E está ligada a estação de destino, que reconhece o seu endereço MAC
14. A estação 2 responde ao pacote P recebido, enviando o pacote R
15. A bridge 2, como já conhece o endereço de destino, faz o **forwarding** do pacote R da LAN E para a LAN C
 - A LAN D nunca vai receber o pacote R, porque a **bridge 2** sabe qual a porta para onde deve encaminhar os endereços para a estação 1
16. A bridge 1, como já conhece o endereço de destino, faz o **forwarding** do pacote R para para a LAN A
17. A estação 1, ligada à LAN A, recebe o pacote de destino

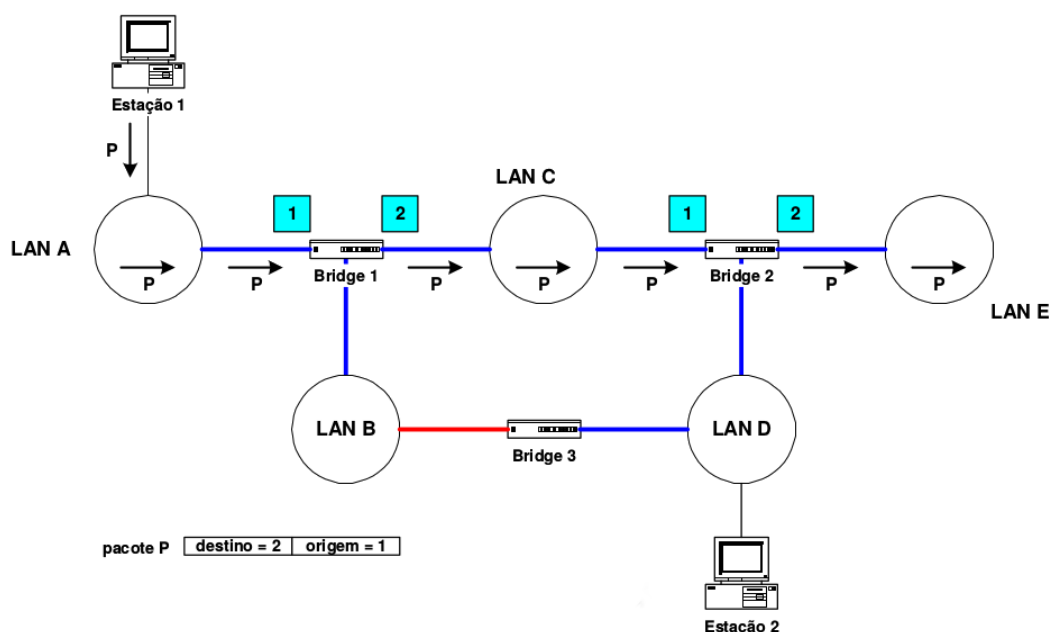


Figure 18: Exemplo da ação do spanning tree após ações na rede

Após a troca da estação 2 da LAN E para a LAN D, a estação 2 deixa de receber pacotes porque as entradas das tabelas de switch das bridges continuam a indicar que devem enviar os pacotes para a LAN E.

O protocolo de spanning tree vai entrar em ação, indicando que a entrada na rede deixou de ser válida para atingir o endereço pretendido, desencadeando os mecanismos necessários para ocorrer as mudanças nas tabelas de encaminhamento.

As tabelas de encaminhamento só vão ser atualizadas se:

- a estação 2 transmitir
- o tempo de vida (TTL) das entradas na tabela de encaminhamento expirar por **time-out** do contador do spanning tree

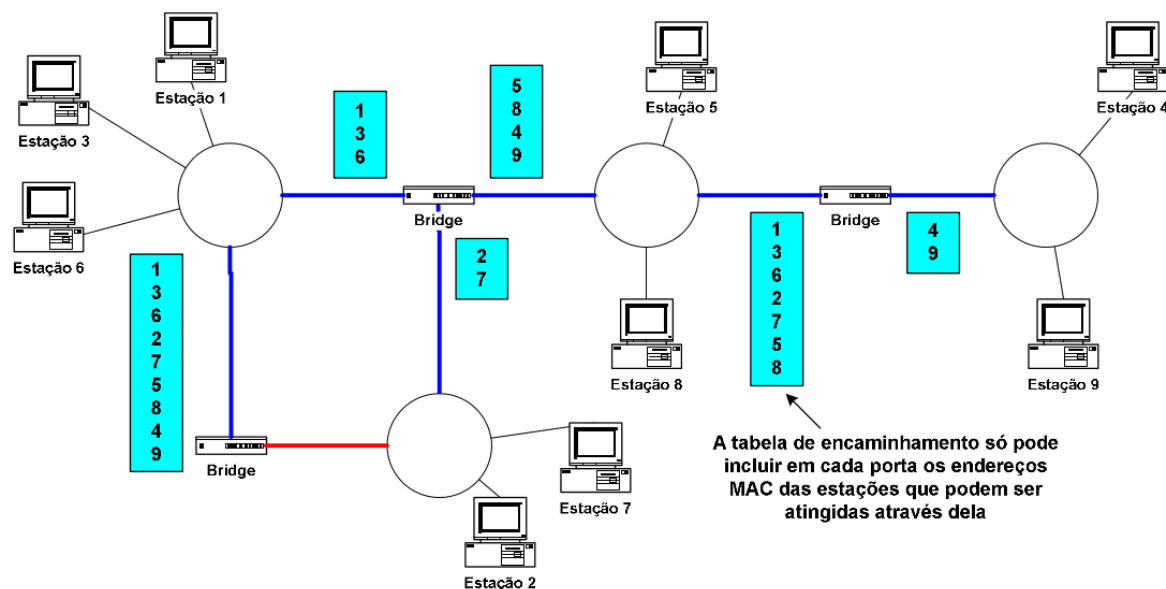


Figure 19: Exemplo de bridge-forwarding

No protocolo Spanning Tree, os nós da rede estão constantemente a aprender o estado da rede. As entradas da tabela de encaminhamento tem uma validade que força a atualização da informação sobre a rede.

A validade das entradas pode ser curta ou longa:

- validade curta
 - quando recomeça a transmitir após um intervalo de tempo, irá ser necessário ocorrer **flooding**
- validade longa
 - quando se efetua mudanças da máquina entre LANs, esta vai ficar sem coenctividade

Tipicamente, usam-se dois **timings**:

- Um para situações de atualização da **spanning-tree**
- Outro para situações de timeout normais da **spanning tree**

Numa situação normal de **time-out**, normalmente as entradas tem um timeout de ≈ 5 minutos. Na prática, o valor pode ser diferente, mas é raro obter valores inferiores a 2 min.

Quando a **spanning tree** está em configuração, as spanning tree sao caluldas de forma diferente em função da raiz da mesma. Não ocorre mudança das máquinas, mas a forma como chega às várias estações é diferente.

A validade de cada uma das entradas é mais curto do que o habitual (normalmente 15 seg). Quando ocorre uma mudança existe um pacote próprio:

- TCN - Topology Change Notification
 - 5 min - as tabelas demoram algum tempo a serem limpas
 - em termos práticos é difícil ver menos de 2 min

Podemos ter uma spanning tree que é “ineficiente”. A *spanning tree* procura o caminho mais curto, não o mais estável.

- por vezes o caminho mais curto não é o mais estável
- não é objetivo do SPT, (Spanning tree protocolo) descobrir o caminhos mais rápido

4.1 Bellman Equations

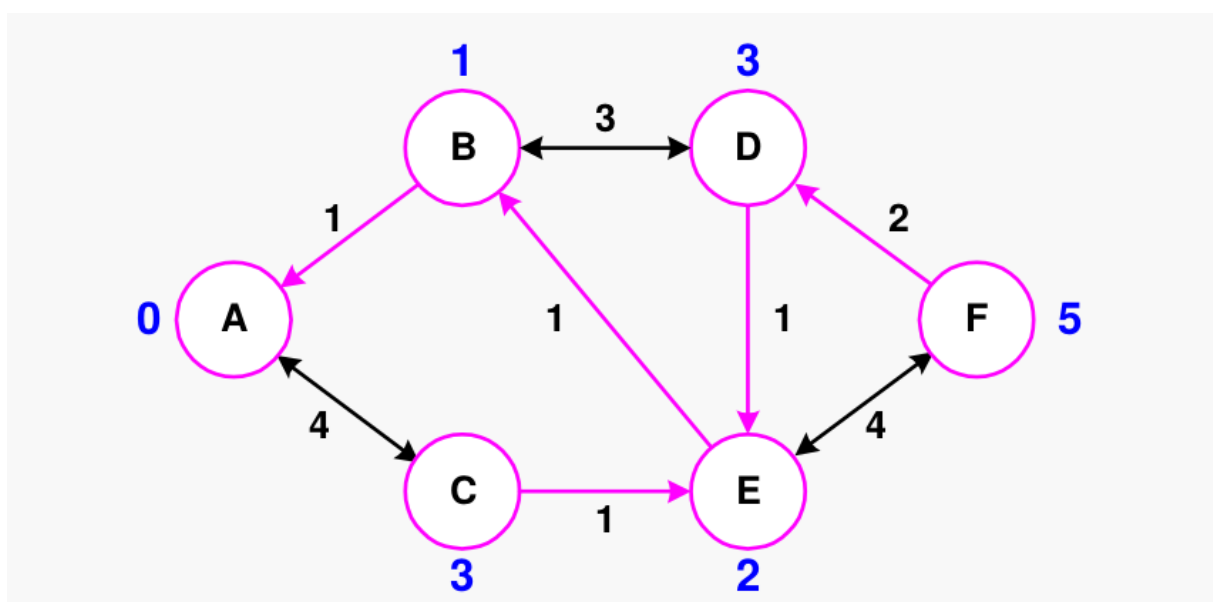


Figure 20: Grafo com os custos de uma rede *spanning tree*

Quando os custos das ligações são não-negativos, temos:

$$\text{Comprimento do percurso mnimo de um } n \text{ X para } A = \text{Comprimento mnimo de } X \text{ at ao } n \text{ que liga esse } n \text{ ao } n \text{ A} \\ + \text{ Comprimento do percurso minimo desse } n \text{ para } A$$

O algoritmo de Bellman-Ford caracteriza-se por ser:

- Distribuído
- Assíncrono
- Cada nó transmite a sua estimativa de custo, D_j , para os seus vizinhos, de um intervalo de tempo definido em um intervalo de tempo definido

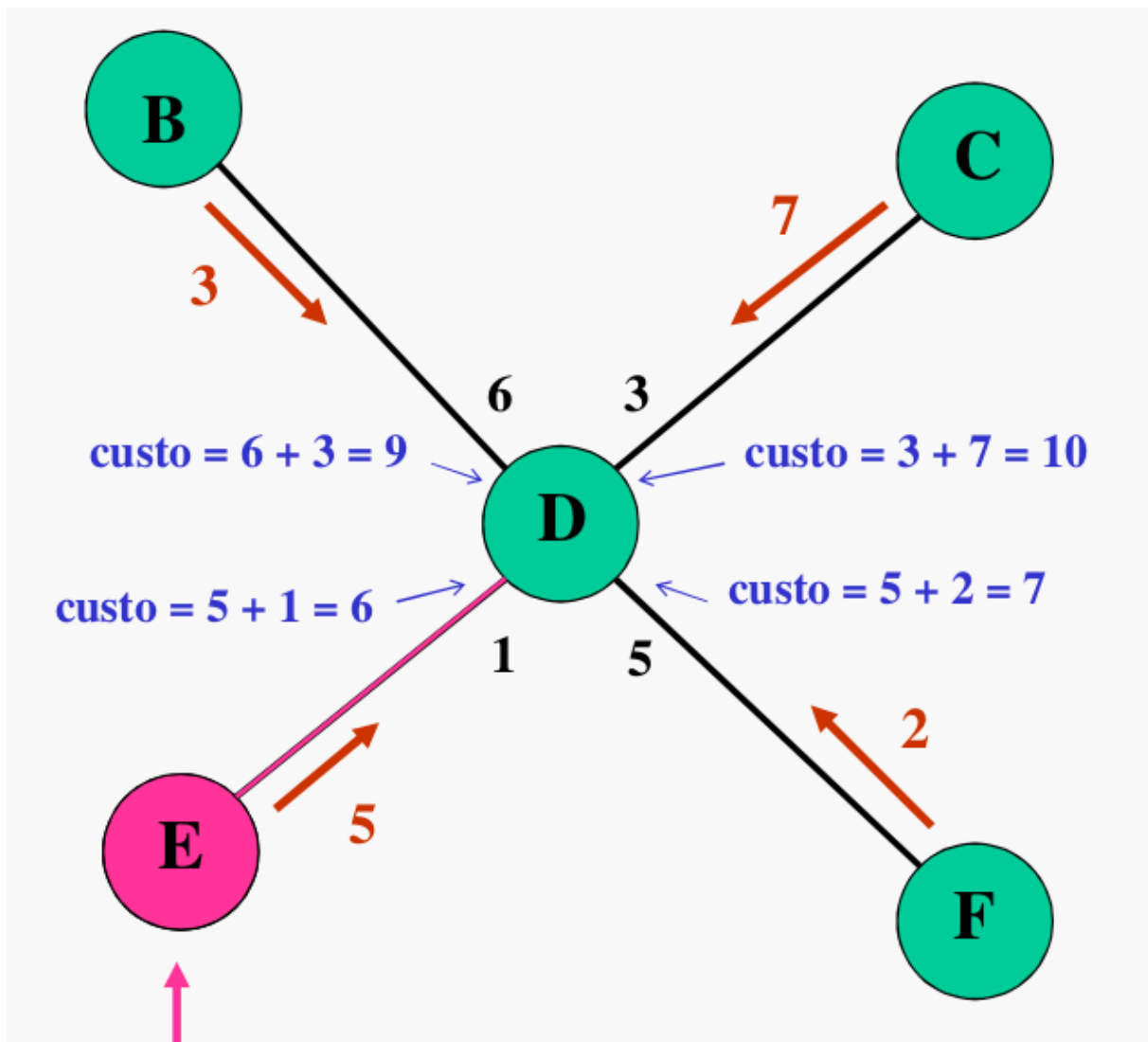


Figure 21: Exemplo do algoritmo de Bellman-Ford

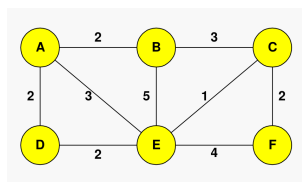
- Com uma cadência definida, cada nó $i \neq 1$, a cada iteração, usando as suas últimas estimativas para D_j e a informação mais recente sobre o seu estado e o comprimento das suas ligações de saída, calcula

$$D_i = \min(d_{ij} + D_j)$$

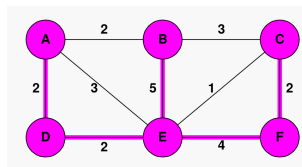
4.2 Forwarding baseado em Spanning Trees

- É escolhida uma bridge como nó de origem
- As outras bridges usam o algoritmo de Bellman-Ford assíncrono e distribuído
 - Calculam para decidir qual o nó vizinho que devem usar
 - O percurso escolhido deve ter o custo mínimo para o nó origem
- As ligações compostas pelos percursos de **custo mínimo** (de todas as bridges para a origem), definem uma spanning tree ¹
- Só as portas ativas é que compõem as ligações da **spanning tree**
- É necessário um critério para desempatar quando há múltiplos percursos de custo mínimo

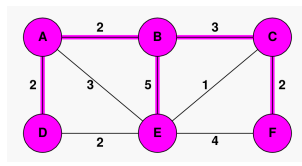
¹árvore abrangente



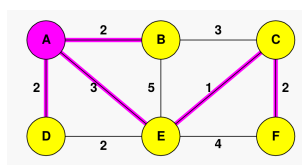
(a) Gráfico da rede



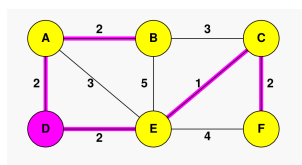
(b) 1º exemplo de uma spanning tree possível



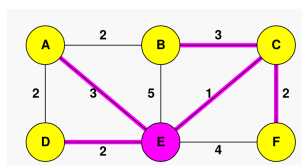
(c) 2º exemplo de uma spanning tree possível



(d) 3º exemplo de uma spanning tree possível



(e) 4º exemplo de uma spanning tree possível



(f) 5º exemplo de uma spanning tree possível

Figure 22: Exemplo de uma rede e múltiplas spanning trees possíveis

4.3 Funcionamento da Spanning Tree

Exemplo do funcionamento da Spanning Tree

- **Bridge ID**: cada bridge é identificada por um endereço que contém:
 - 2 octetos de prioridade
 - * configuráveis pelo gestor de rede
 - * 6 octetos fixos
 - Um dos endereços MAC das portas da bridge
 - * Qualquer outro endereço único de 48 bits
 - A prioridade possui precedência sobre o campo de octetos fixos
- **Bridge raiz** (*root bridge*):
 - bridge que está na raiz da spanning tree
 - bridge com menor bridge ID
- **Path cost**:
 - custo associado a cada porta da bridge
 - pode ser configurado pelo gestor de rede
- **Designated Bridge** (*bridge designada*):
 - bridge que numa LAN é responsável pelo envio de pacotes da LAN para a bridge raiz e vice-versa
 - a bridge raiz é a bridge designada em todas as LANs a que está ligada
 - ver figura 24
- **Designated Port** (*porta designada*):
 - porta que numa LAN é responsável pelo envio de pacotes da LAN para a bridge raiz e vice-versa
 - uma das portas da bridge designada
 - ver figura 24
 - **em cada LAN**, a porta que fornece o **melhor percurso** (menor custo) para a raiz
- **root port** (*porta raiz*)
 - porta que numa bridge é responsável pela receção/transmissão de pacotes de/para a bridge raiz
 - figura ??
 - **em cada bridge**, a porta que fornece o **melhor percurso** (menor custo) para a raiz

Cada bridge tem associado a si um **custo do percurso para a raiz** (**Root Path Cost**) igual à soma dos custos das portas que recebem pacotes enviados pela raiz (portas raiz) no percurso de menor custo para a bridge. Ver figura ??.

As **portas ativas** em cada bridge são a **porta raiz** e as **portas designadas**. As restantes portas estão **inativas** (*blocking*)

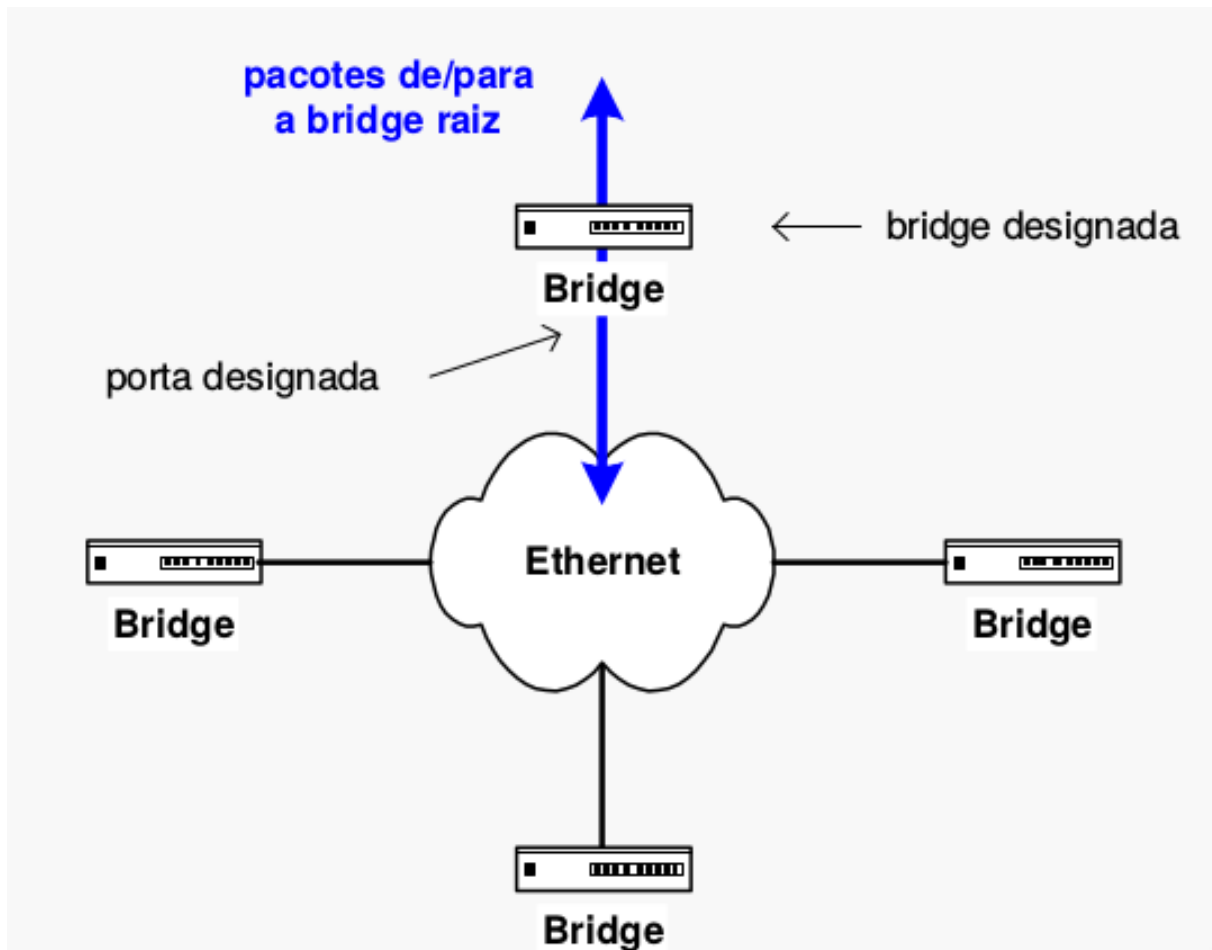


Figure 23: Conceitos de bridges - pt.1

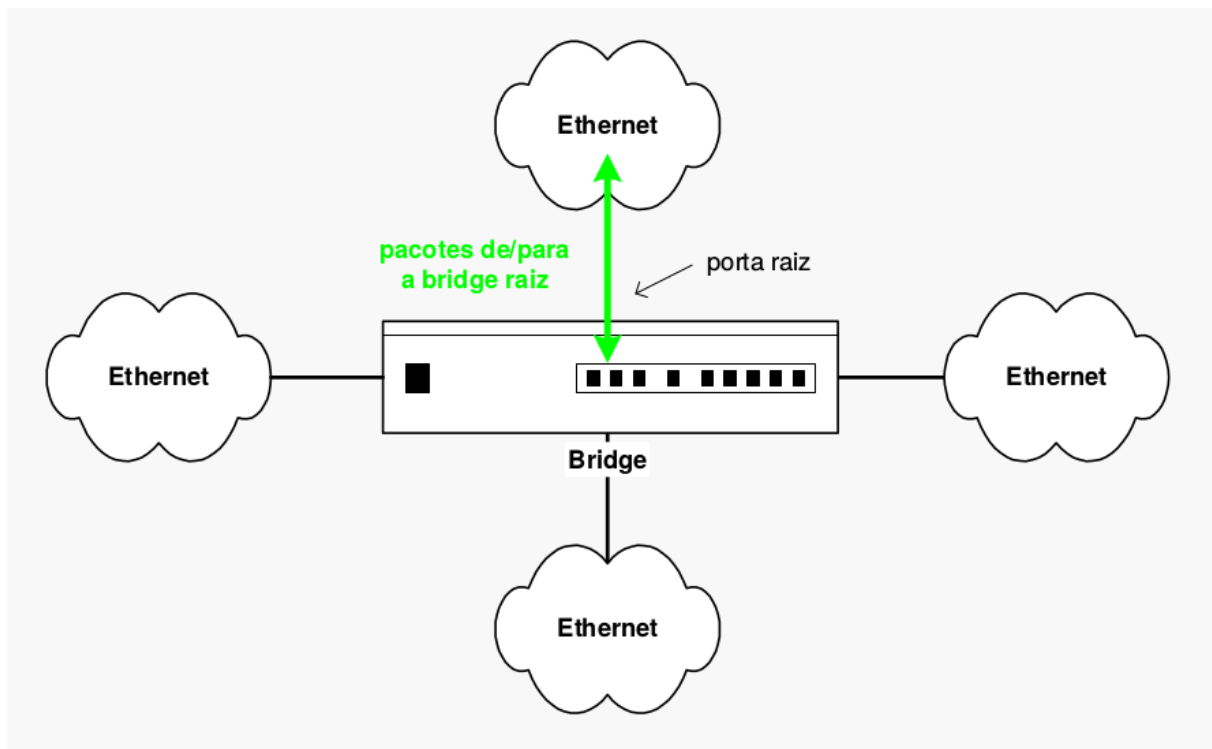


Figure 24: Conceitos de bridges - pt.2

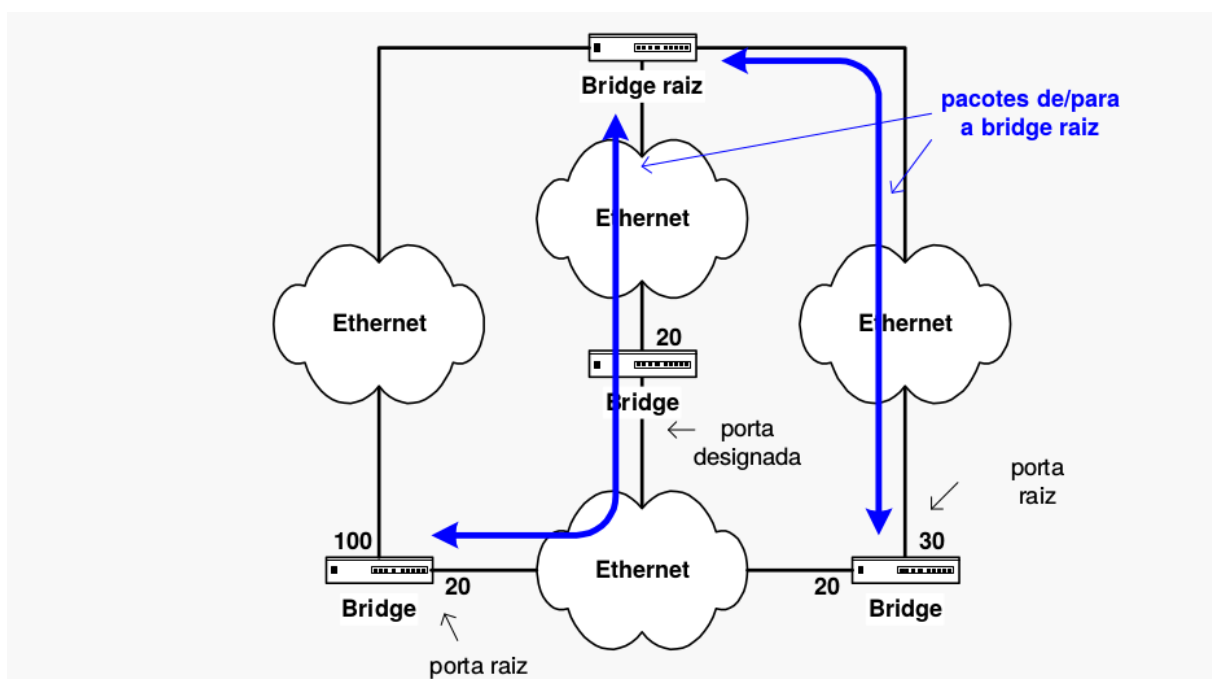


Figure 25: `label{root_path_cost}` Exemplo de custo para a raiz

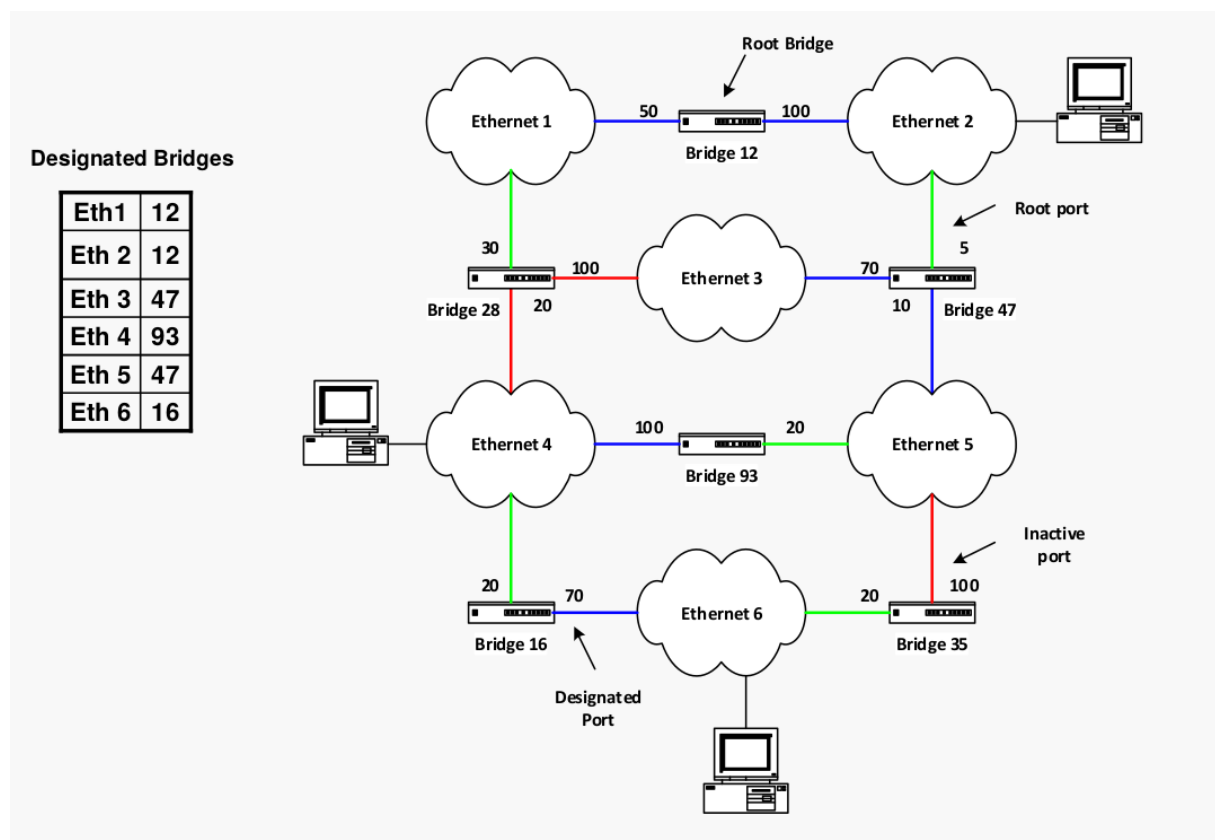


Figure 26: Ligações numa rede antes de as portas `blocking` serem removidas

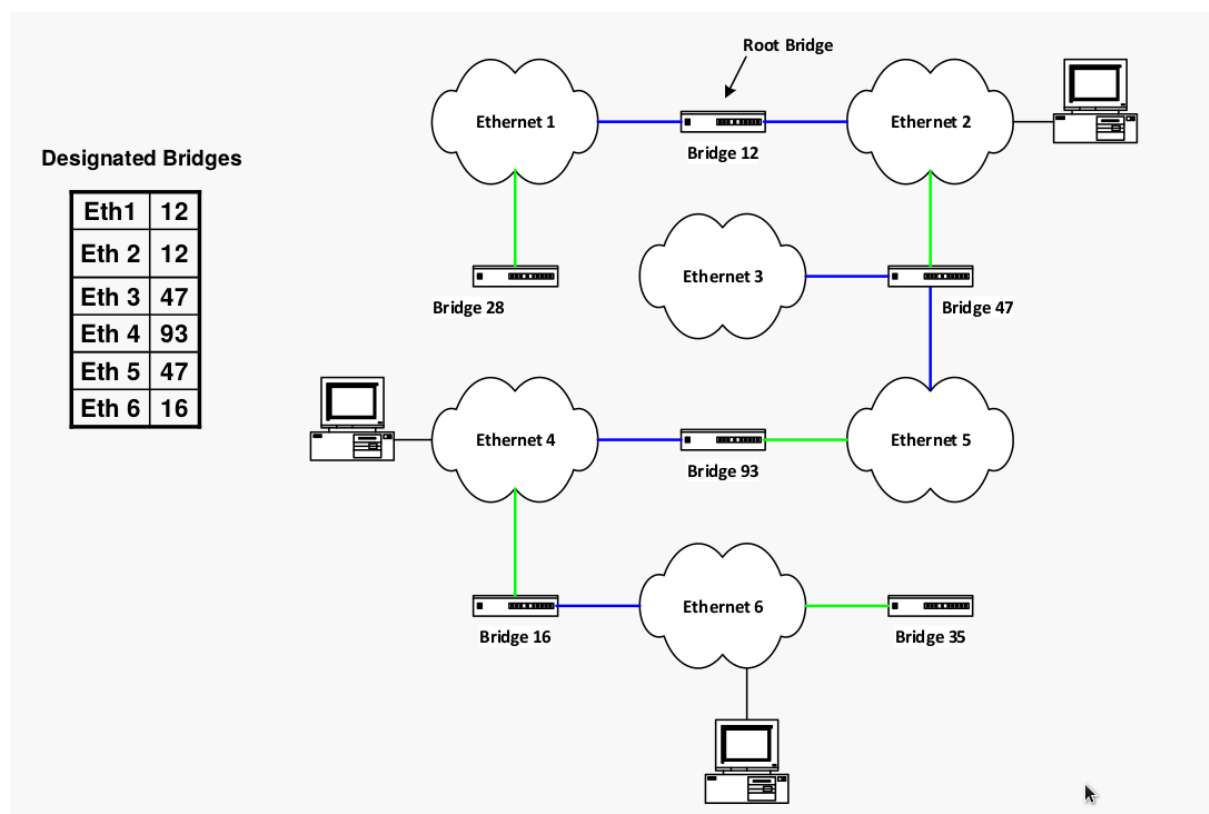


Figure 27: Ligações numa rede após as portas `blocking` terem sido removidas

4.3.1 Exemplo de Mudança

1. Uma porta passa ao estado blocking
2. A bridge que perdeu conectividade envia um TCN
3. A partir desse momento, a bridge acima recebeu o TCN
4. Envia um TCA - Topology CHange Ack para a outra bridge, a informar que já sabe que a ligação não existe
5. Envia um TCN na sua root port para outra bridge
6. O processo repete-se até chegar à raiz da rede
7. A raiz envia um BPDU com um TC - Toplogy Change
8. Os bridges mudam o timeout de reconfiguração
9. A reconfiguração acontece
10. As spanning trees são atualizadas

4.4 Protocolo IEEE 802.1D

- Para construir e manter a spanning tree as bridges trocam mensagens entre si
- Estas mensagens são desingadas de BPDUs - Bridge Protocol Data Units
- Existem dois tipos:
 1. Configuration
 2. Topology Change Notification

O Header do pacote é apresentado na figura 28

destination	source	DSAP	SSAP	BPDU
-------------	--------	------	------	------

Figure 28: Campos do pacote IEEE802.1D

- **Destination:** endereço multicast atribuído a todas as bridges
- **Source:** endereço MAC da porta que enviou o BPDU
- **DSAP = SSAP = 0100010 = 0x42**

Os campos mais importantes são:

- **Root ID:** estimativa atual do endereço da bridge raiz
- **Root Path Cost:** estimativa atual do custo para a bridge raiz
- **Bridge ID:** endereço da bridge que envia a mensagem de configuração
- **Port ID:** endereço da porta que envia a mensagem de configuração

4.5 Manutenção da Spanning Tree

- Convenção para Configuration-BPDUs:
 - **root bridge**



The image shows a packet capture analysis in NetworkMiner. The selected packet is a Bridge Protocol Data Unit (BPDV) with the following details:

- 802.2: Address: 00-E0-B0-64-48-77 ---> 01-80-C2-00-00-00
- LLC: Sap 0x42 ---> 0x42 (Command)
- Bridge Protocol Data Unit
 - Protocol ID: 0
 - Version: 0
 - Type: 0 (Configuration)
 - Flags: 0x00
 - Root ID: 40000800021EA1F1
 - Priority: 0x4000
 - MAC Address: 08-00-02-1E-A1-F1
 - Root Path Cost: 100
 - Bridge ID: 800000E0B0644876
 - Priority: 0x8000
 - MAC Address: 00-E0-B0-64-48-76
 - Port ID: 0x8003
 - Message Age: 1
 - Max Age: 20
 - Hello Time: 2
 - Forward Delay: 15
 - Frame Padding : (8 bytes)
- Calculate CRC: 0xb209dfee

Figure 29: Exemplo de uma mensagem BPDV num analisador de redes

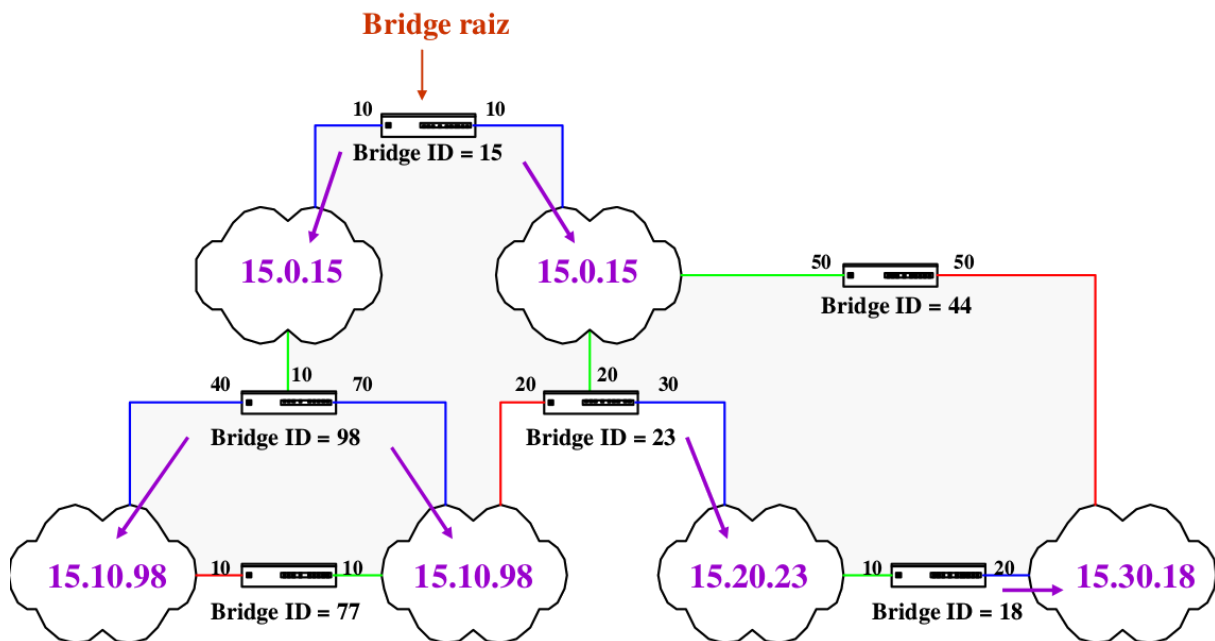


Figure 30: Exemplo de uma rede com a spanning tree configurada

- root path cost
- bridge ID
- Periodicidade das Configuration-BPDUs = *hello time*
 - *hello time* recomendado: 2 seg

4.5.1 Order of configuration messages

Uma mensagem de configuração C_1 diz-se melhor que outra, C_2 , se:

1. o root ID de C_1 for inferior ao de C_2
2. Se os dois root IDs forem iguais, o root path cost de C_1 for inferior ao de C_2
3. Se os root IDs e Root Path Cost forem idênticos, o bridge ID de C_1 for inferior ao de C_2
4. Sendo idênticos os root IDs, root path cost e bridge ID, o Port ID de C_1 for inferior ao de C_2

Table 1: Table que mostra a ordenação, do melhor para o pior, das mensagens de configuração

Root ID	Root Path Cost	Bridge ID	Port ID
18	27	32	2
18	27	32	4

Root ID	Root Path Cost	Bridge ID	Port ID
18	27	43	1
18	35	23	3
18	31	45	2

4.6 Construção da Spanning Tree

1. Cada bridge assum inicialmente que é a bridge raiz
 - `root path cost` = 0
 - Envia mensagens de configuração em todas as suas portas
 - figura 31
2. As melhores mensagens recebidas pela bridge que enviou as mensagens de configuração permitem decidir qual a `root port`
 - Usam a estimativa do `root path cost` enviada por cada uma das bridges
 - o `root path cost` é igual à soma do custo da:
 - porta raiz da bridge
 - `root path cost` da melhor mensagem de configuração recebida até ao instante
 - Através da figura 31, podemos concluir que
 - O `root ID` mais baixo é 41, mas existem 4 bridges com este ID
 - O `root path cost` mais baixo é o 12, mas existem 2 bridges com este `root path cost`
 - O `bridge ID` mais baixo é 111
 - Assim, na figura:
 - * **bridge raiz** = 41
 - * **porta raiz** = 4
 - * **custo para a raiz** = $12 + 1 = 13$, supondo que os custos das portas são todos unitários
 - Quando um bridge recebe, na que julga ser a sua **porta raiz**, uma mensagem de configuração **melhor** que a mensagem de configuração que iria transmitir, com base nas suas estimativas actuais de `root ID` e `root path cost`, transmite essa mensagem de configuração em todas as portas em que **bridge designada**
 - Quando uma bridge recebe **numa porta** uma mensagem de configuração **melhor** do que aquela que iria transmitir (com base nas suas estimativas actuais de **Root ID** e **Root Path Cost**), a bridge **cessa de transmitir mensagens de configuração nessa porta**
 - a porta passa ao estado `blocking`
 - Quando uma bridge recebe numa porta uma mensagem de configuração **pior** do que aquela que iria transmitir (com base nas suas estimativas actuais de `Root ID` e `Root Path Cost`), a bridge **transmite a sua mensagem de configuração nessa porta**
 - este acontecimento pode ser visto na figura ??

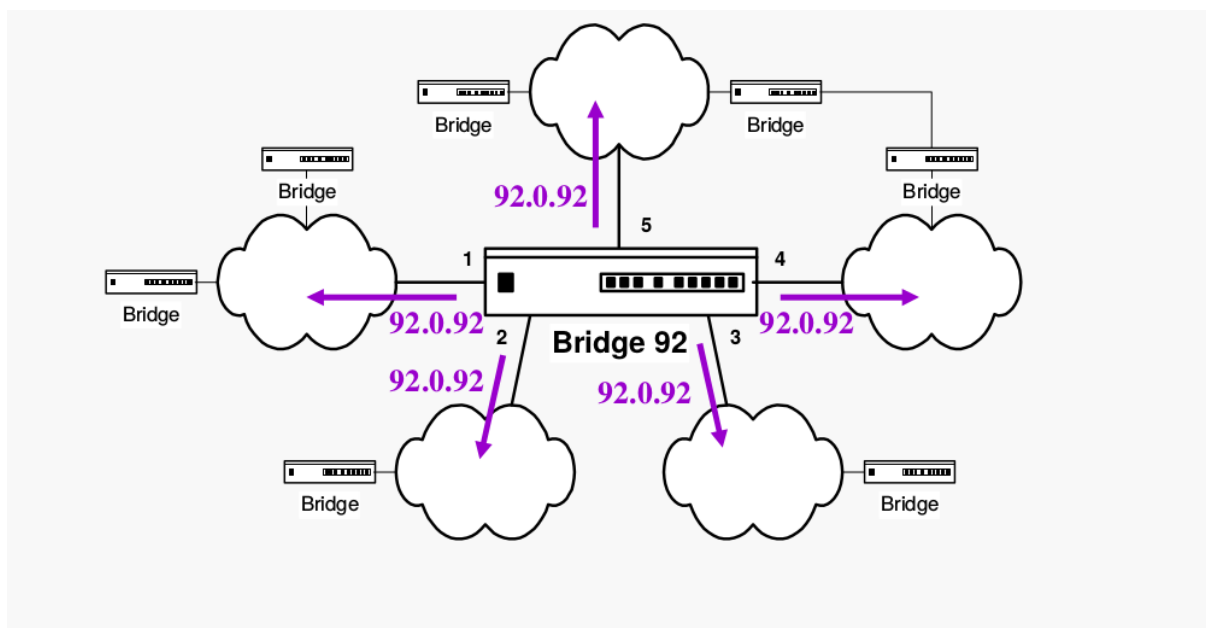


Figure 31: Mensagens enviadas pela Bridge 92

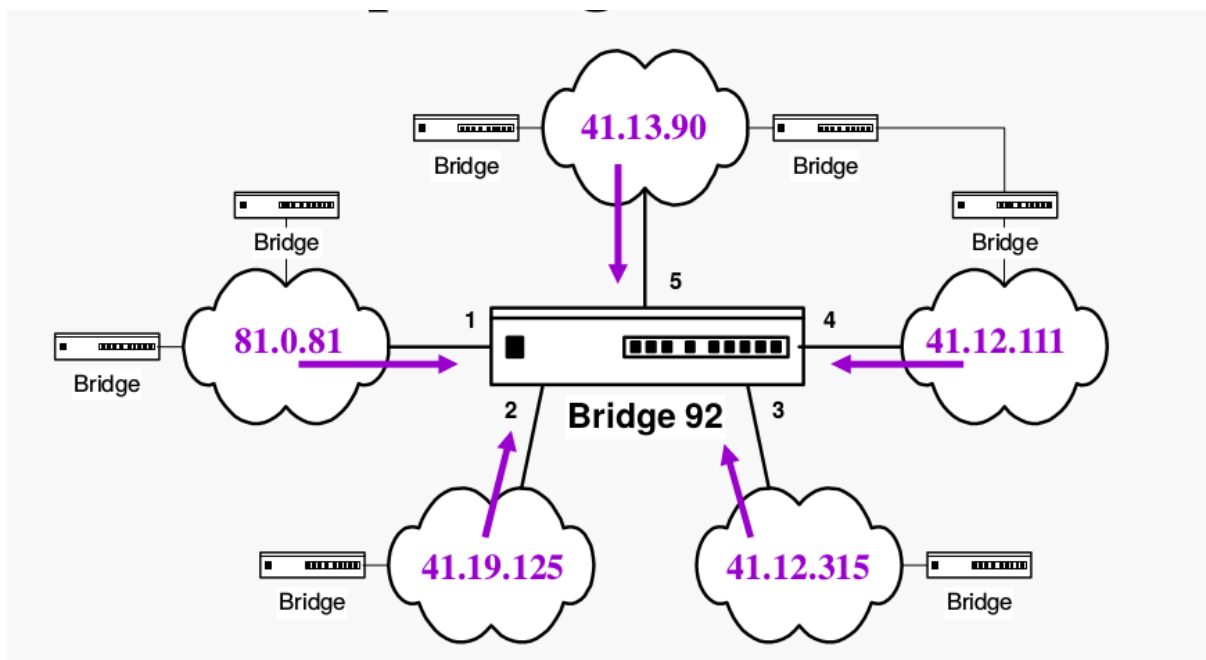


Figure 32: Mensagens recebidas pela Bridge 92

4.6.1 Exemplo

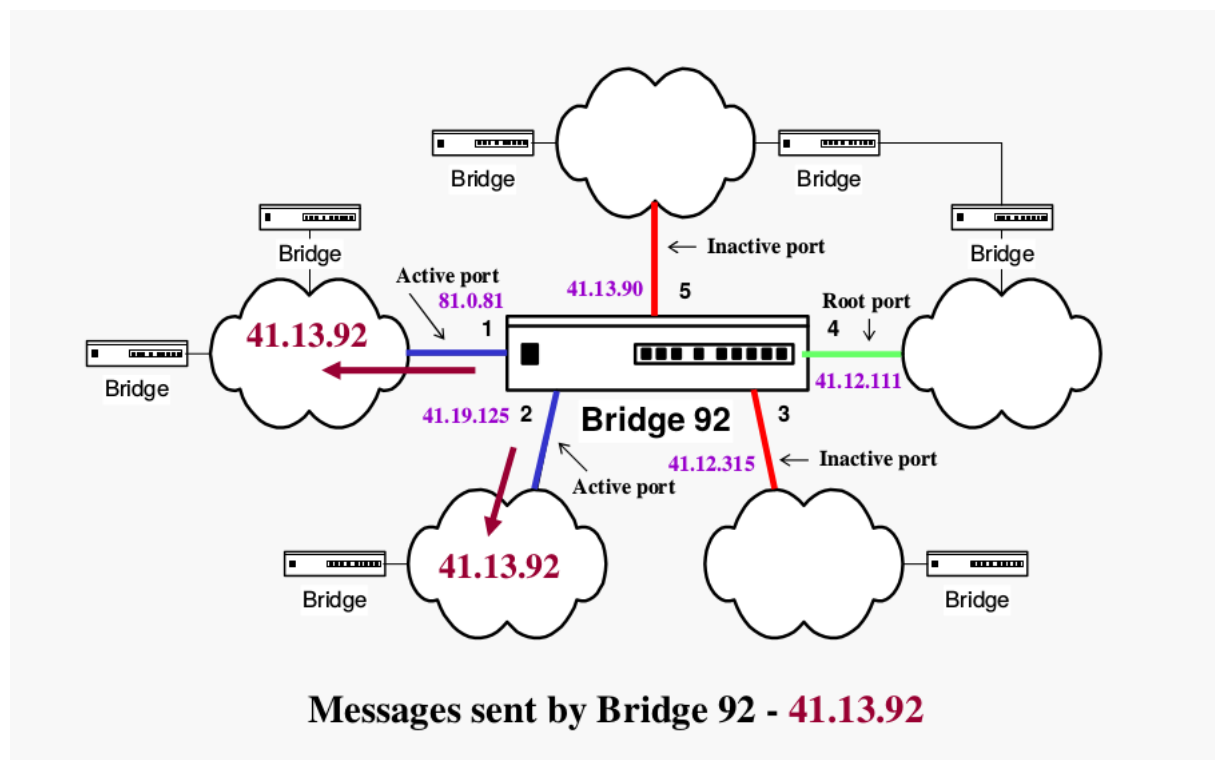


Figure 33: Exemplo de construção da Spanning Tree

1. A bridge 83 envia 83.0.83 em Eth1, Eth2 e Eth3
2. A bridge 21 envia 21.0.21 em Eth2 e Eth3 (83 fica a saber que não é raíz)
3. A bridge 83 envia 21.20.83 em Eth1 (porta raíz = 2)
4. A bridge 18 envia 18.0.18 em Eth1 e Eth3 (21 fica a saber que não é raíz)
5. A bridge 21 envia 18.20.21 em Eth2
6. Mas a bridge 83 tem uma mensagem melhor para a Eth2
7. A bridge 83 envia 18.10.83 em Eth2 (porta raíz = 1, Path Cost = 10)

O Algoritmo convergiu. No entanto, continuam a ocorrer um conjunto de operações periódicas.

- A bridge raíz envia periodicamente 18.0.18
- A bridge 83 retransmite com 18.10.83 em Eth2
- A bridge 83 é a bridge designada em Eth2 e a sua porta raíz é a 1.
- A porta 3 está inactiva.
- A porta 1 da bridge 21 está inactiva.
- A bridge 18, sendo a raíz, é a bridge designada em Eth1 e Eth3.

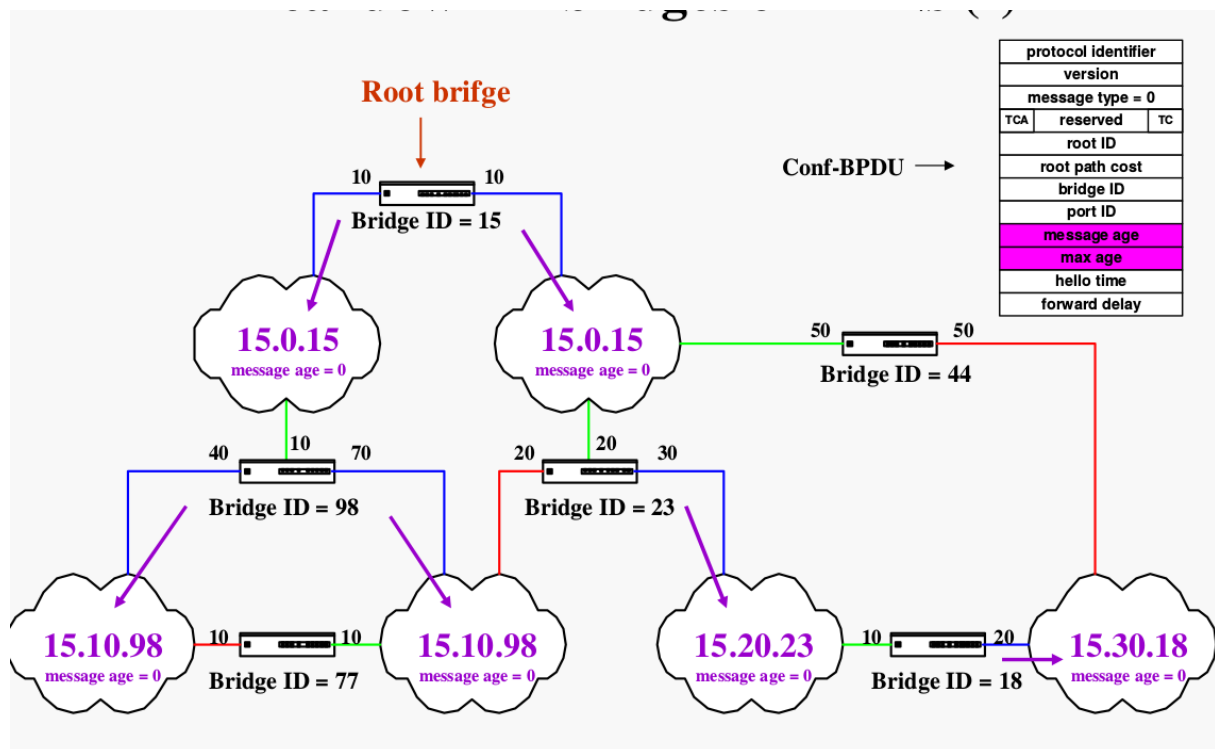


Figure 34: Estado original das bridge

4.7 Breakdown em bridges ou LANs

4.7.1 Ciclos temporários

Após a alteração da topologia da rede, pode ocorrer:

- **perda temporária de conectividade:**
 - uma porta estava inativa na topologia antiga deve estar ativa na nova topologia
- **ciclos temporários:**
 - uma porta que estava ativa na topologia antiga deve estar inativa na nova topologia

De modo a evitar os ciclos temporários que se podem estabelecer momentaneamente e fazer com que a rede crashe, quando ocorre a troca de topologias, é necessário que as portas esperem algum tempo antes de permitirem que uma das suas portas passe do estado inativo para ativo. Este tempo de espera denomina-se *forward delay*.

4.8 Bridge Port States

- **estado blocking**
 - os processos de aprendizagem e de expedição de pacotes estão **inibidos**

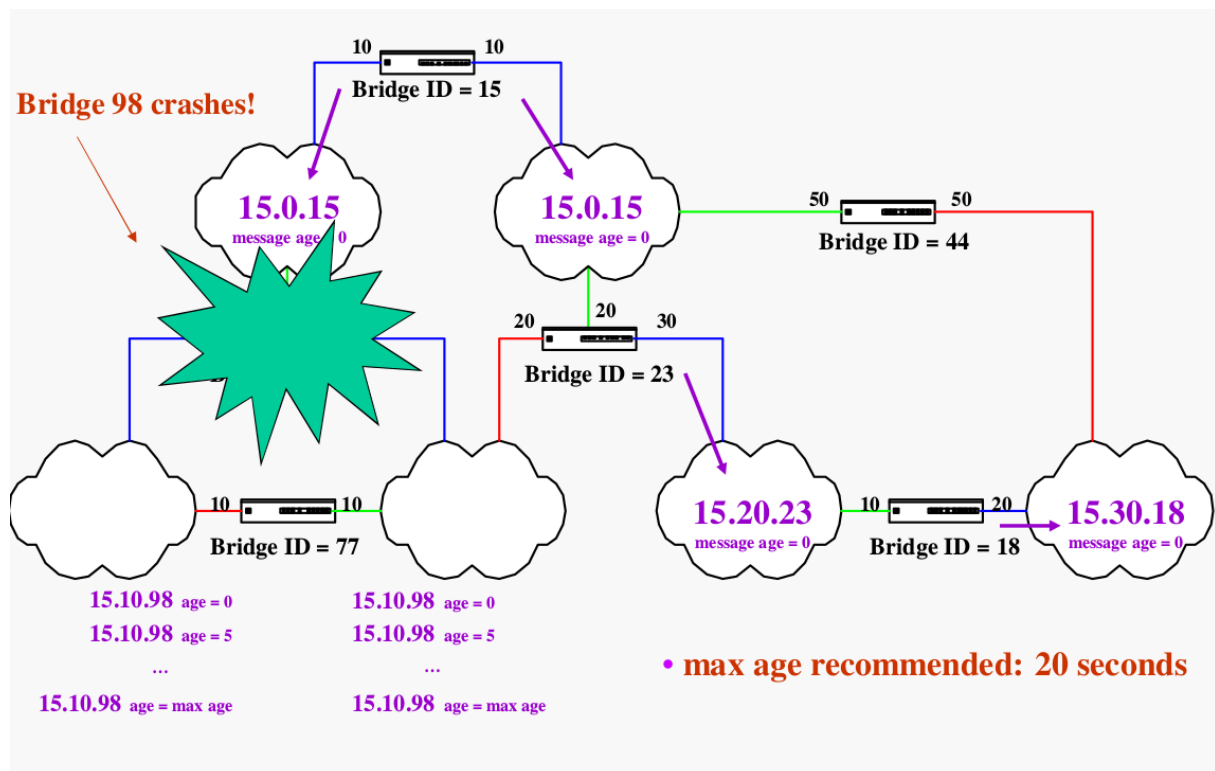


Figure 35: Bridge crash

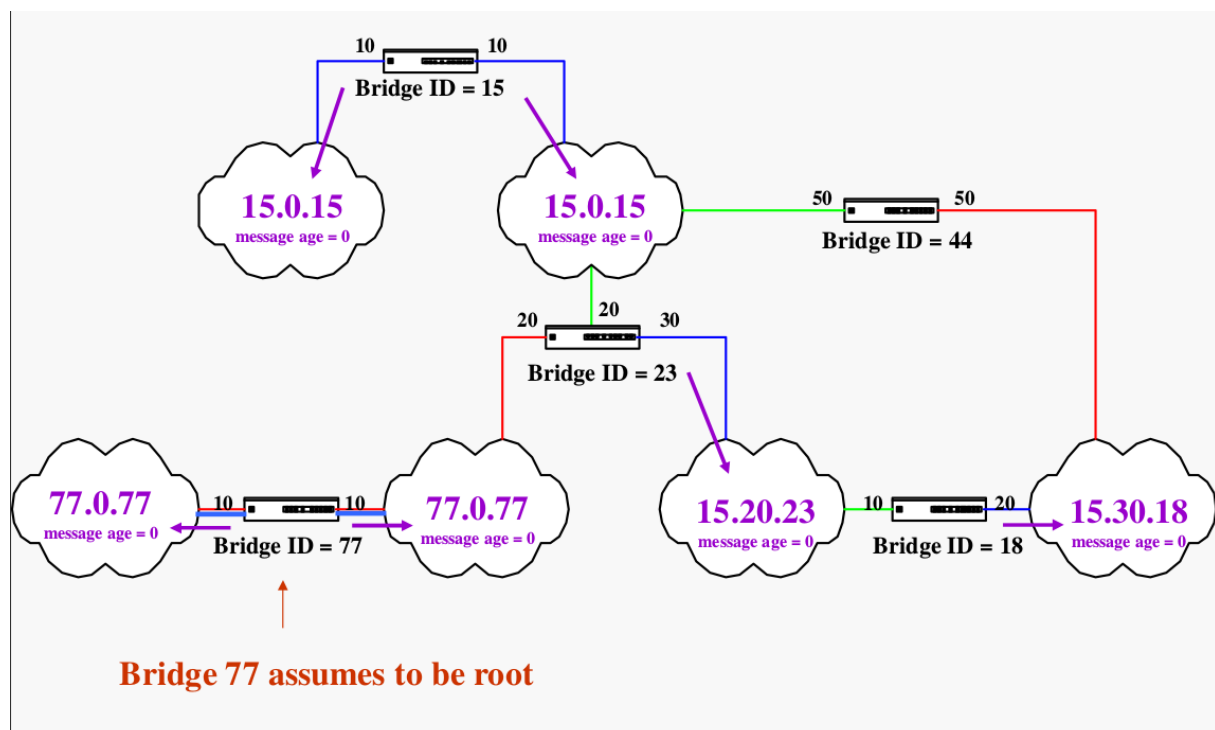


Figure 36: A Bridge 77 assume ser a bridge root e envia mensagens de configuração

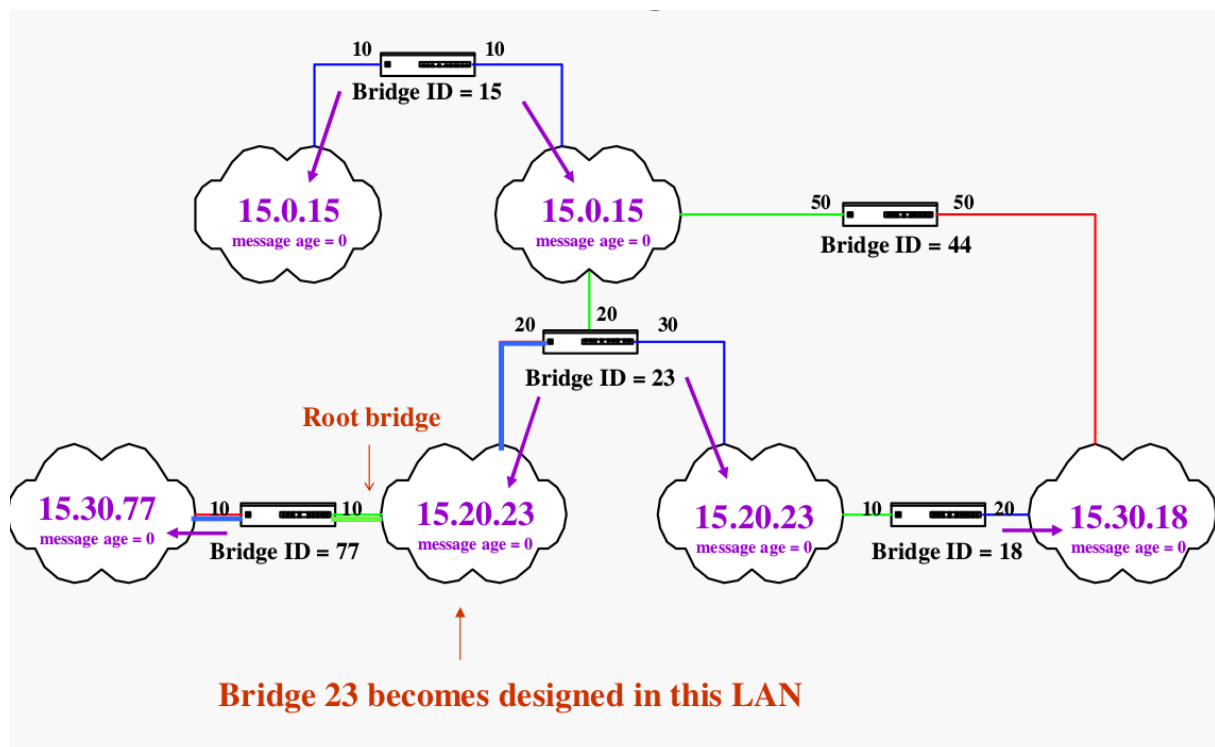


Figure 37: A bridge 23, torna-se a bridge designada na LAN

- recebe e processa **mensagens de configuração**
- **estado listening**
 - o processo de aprendizagem e expedição de pacotes estão **inibidos**
 - transita para o estado **learning** após um tempo de permanência neste estado ligado a **forward delay**
 - recebe e processa mensagens de configuração
- **estado learning**
 - o processo de aprendizagem está **activo**
 - o processo da expedição de pacotes está **inibido**
 - transita para o estado forwarding após um tempo de permanência neste estado igual a **forward delay**
 - recebe e processa mensagens de configuração
- **estado forwarding**
 - **estado activo**
 - * processo de aprendizagem activo
 - * processo de expedição de pacotes activo
 - recebe e processa mensagens de configuração
- **estado disabled**

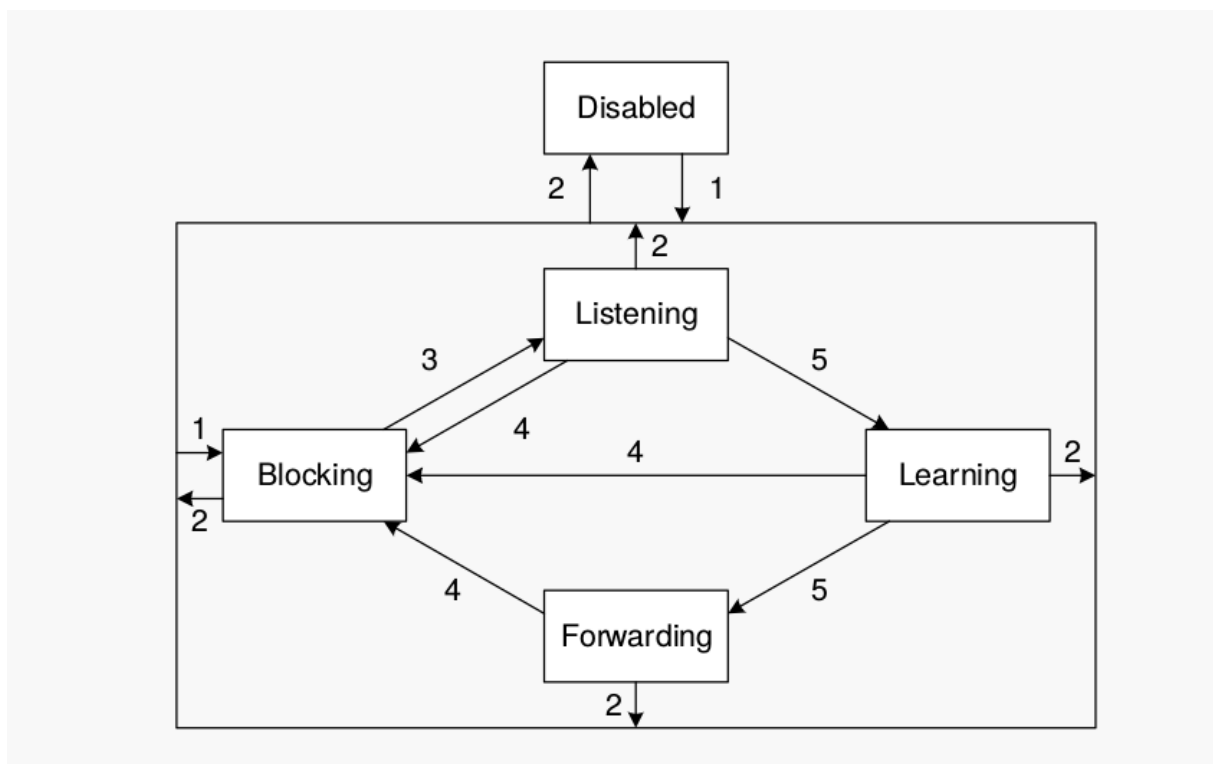


Figure 38: Estados das portas de uma bridge. Na imagem, (1) indica uma porta ativa quer pelo gesto de rede quer pela inicialização da bridge. (2) Indica uma porta desativa, por gestão ou falha. (3) Indica uma porta seleccionada como sendo porta designada ou porta raiz. (4) Indica uma porta seleccionada como não sendo uma porta designada ou porta raiz. (5) Forwarding time expira

- **estado inativo**
 - * processo de aprendizagem inativo
 - * processo de expedição de pacotes inativo
- Não participa no algoritmo da spanning tree

4.8.1 TTL of entries of the forwarding tables

- Tradeoff:
 - **Tempo de vida demasiado longo:** pode haver um número exagerado de pacotes perdidos quando a estação muda de localização
 - **Tempo de vida demasiado curto:** o tráfego na rede pode ser exagerado devido ao processo de *flooding*

Existem dois tempos de vida usado:

- *Longo*:
 - usado por defeito
 - valor recomendado: 5 minutos
- *Curto*:
 - usado quando a spanning tree está em reconfiguração
 - valor recomendado: 15 segundos
 - exige processo de notificação de alterações da topologia da rede
- (1) Porta passa ao estado blocking
- (2) A bridge 2 envia um TCN-BPDU, com a periodicidade de *hello time*
- (3) A bridge 2 envia um Conf-BPDU com TCA = 1 até deixar de receber o TCN-BPDU da bridge 2
- (4) A bridge 1 envia um TCN-BPDU, com periodicidade *hello time*.
- (5) A bridge raiz envia um Conf-BPDU com TC=1 (Topology Configuration), durante um tempo *forward delay + max age*
 - inicia o processo de reconfiguração.
 - a bridge 1 e 3 passam a usar o tempo de vida curto nas suas tabelas de encaminhamento durante esse período
- (6) A bridge 1 repete o Conf-BPDU com TC=1
 - a bridge 2 passa a usar o tempo de vida curto nas suas tabelas até voltar a ouvir TC = 0

Exemplo de uma spanning tree ineficiente

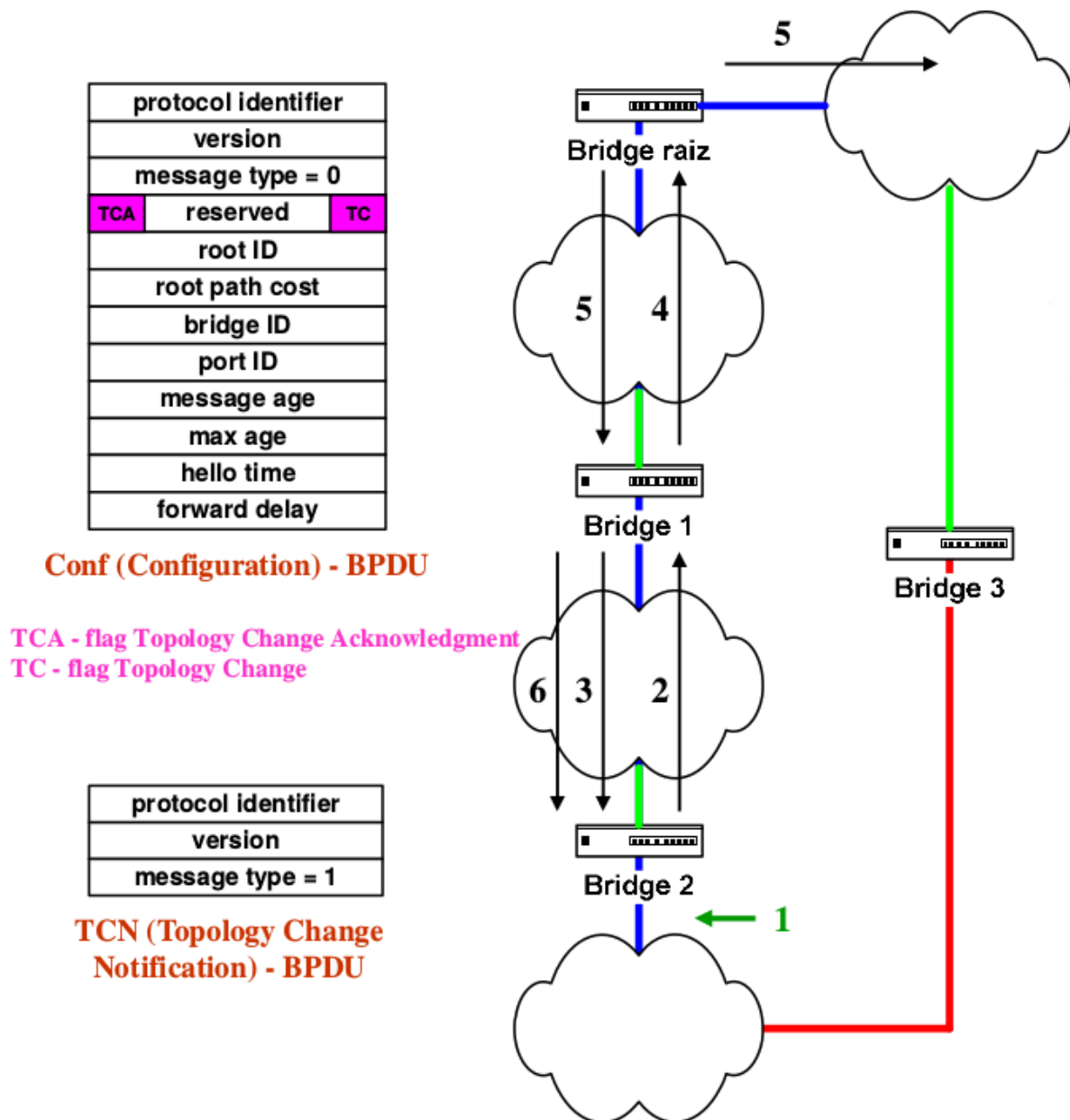


Figure 39

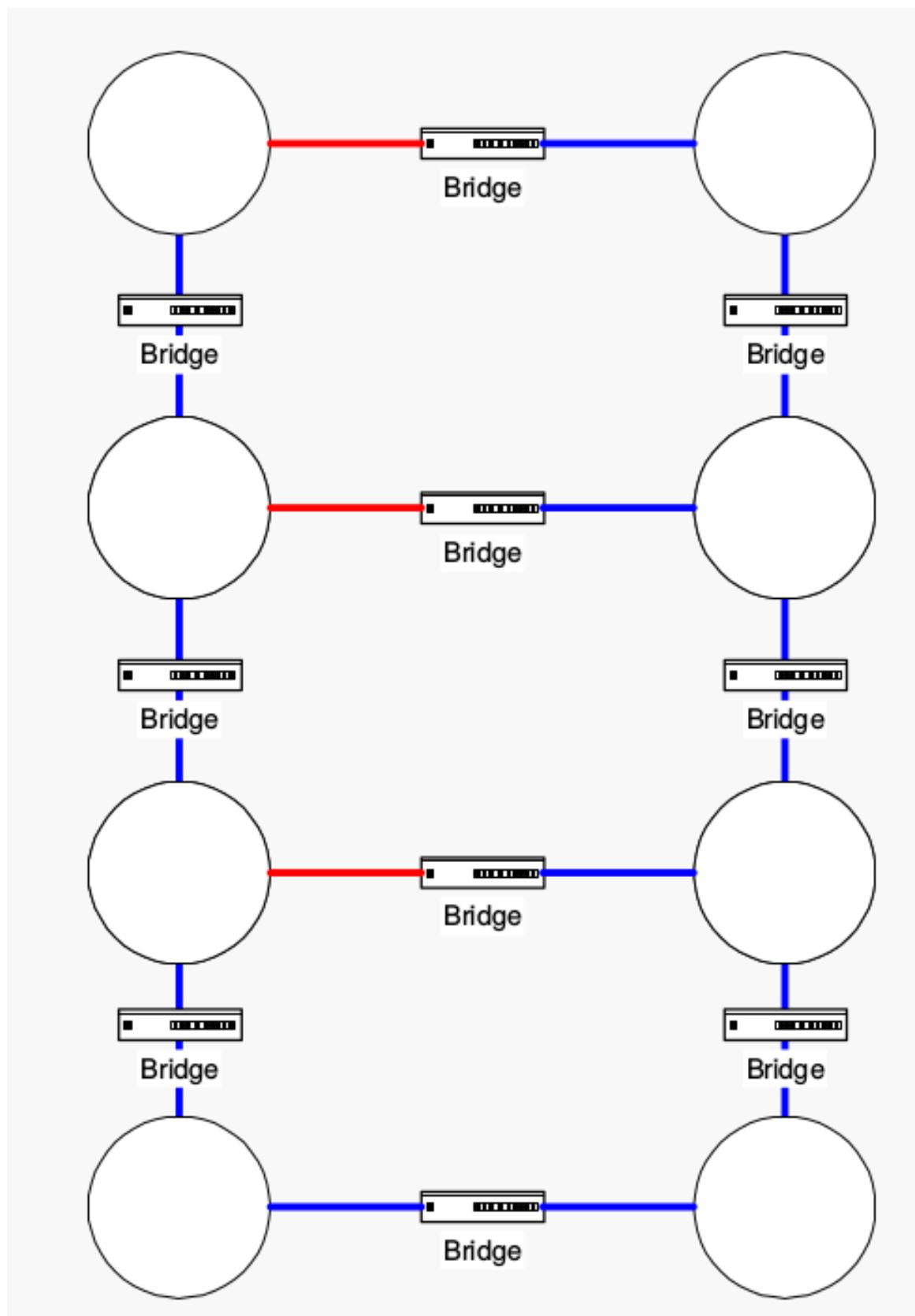


Figure 40: Exemplo de uma spanning tree ineficiente

4.9 Protocolos

4.9.1 802.1p

- extensão do IEEE 802.1Q
- permite mapear nos bit da ethernet prioridades
 - pode ser usado para qualidade de serviço
 - Campo **User Priority** (3 bits)
 - * 8 níveis de prioridade

Table 2: Recomendações do uso de prioridades da norma

Prioridade	Descrição
7	tráfego crítico de gestão de rede
5-6	tráfego sensível ao atraso (e.g., voz e vídeo)
1-4	tráfego sensível à variação do atraso (streaming)
0	restante tráfego (e.g., transferência de ficheiros)

4.9.2 802.1w Rapid Spanning Tree Protocol

- extensão do IEEE 802.1D
- Acelera o ponto de convergência relativamente ao 802.1D
 - mais rápido a convergir
- Altera os estados e funções de cada porta
- Introduce um mecanismo de negociação entre bridges
- usa os bits entre as flags TCA e TC
 - bits entre o **topology change** e o **topology change advertisement** que permite às bridges convergir mais rapidamente
- Mais informação: Understanding Rapid Spanning Tree Protocol (802.1w)

4.9.3 802.1s - Multiple Spanning Tree Protocol

- O STP e RSTP não reconhecem VLANs
 - Primeiro é necessário estabelecer as spanning tree
 - Ao criar as VLANs fecha portas nas spanning trees
 - Se existir tráfego nas VLANs não devia haver problema
 - Se as bridges não tiverem tráfego na altura de criação das VLANs, vamos ter problemas
 - Uma vez que as VLANs estão separadas, não vai ser possível criar uma spanning tree unica

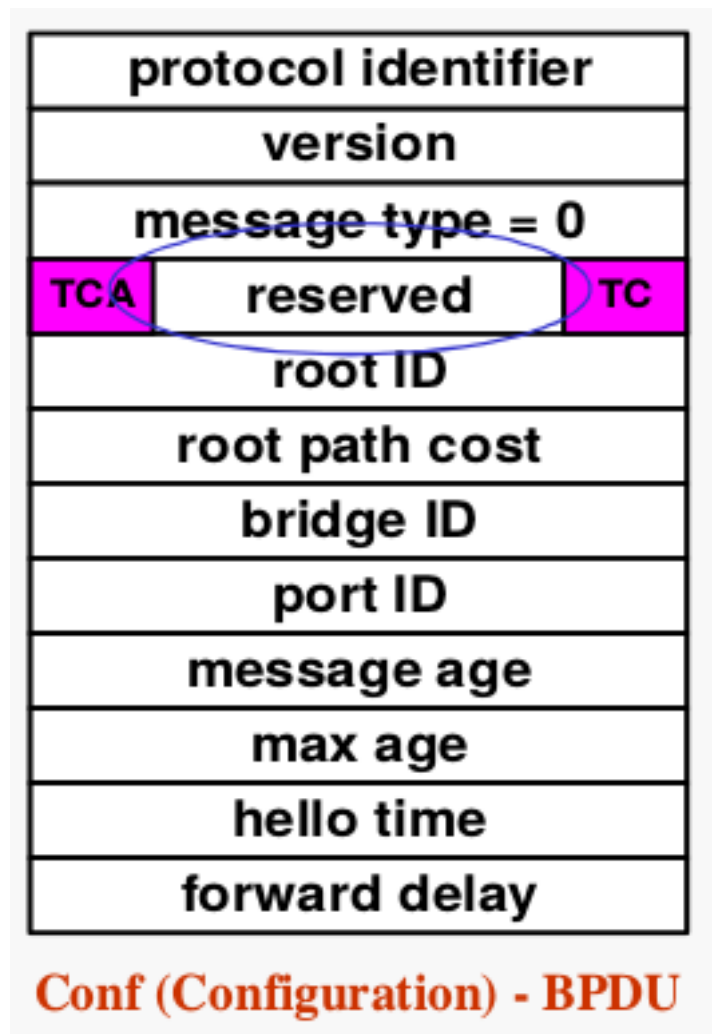


Figure 41: Pacote de configuração BPDU

- este protocolo trata de criar as spanning trees associadas as vlans
- Permite criar múltiplas Spanning Trees e atribuir cada VLAN a uma das Spanning Trees criadas
- Usa os protocolos IEEE 802.1W (VLANs) e IEEE 802.1w (RSTP)
- Permite criar **múltiplas regiões**
 - Fora das regiões funciona como o protocolo IEEE 802.1D
- Mais informação: Understanding Multiple Spanning Tree Protocol (802.1s)

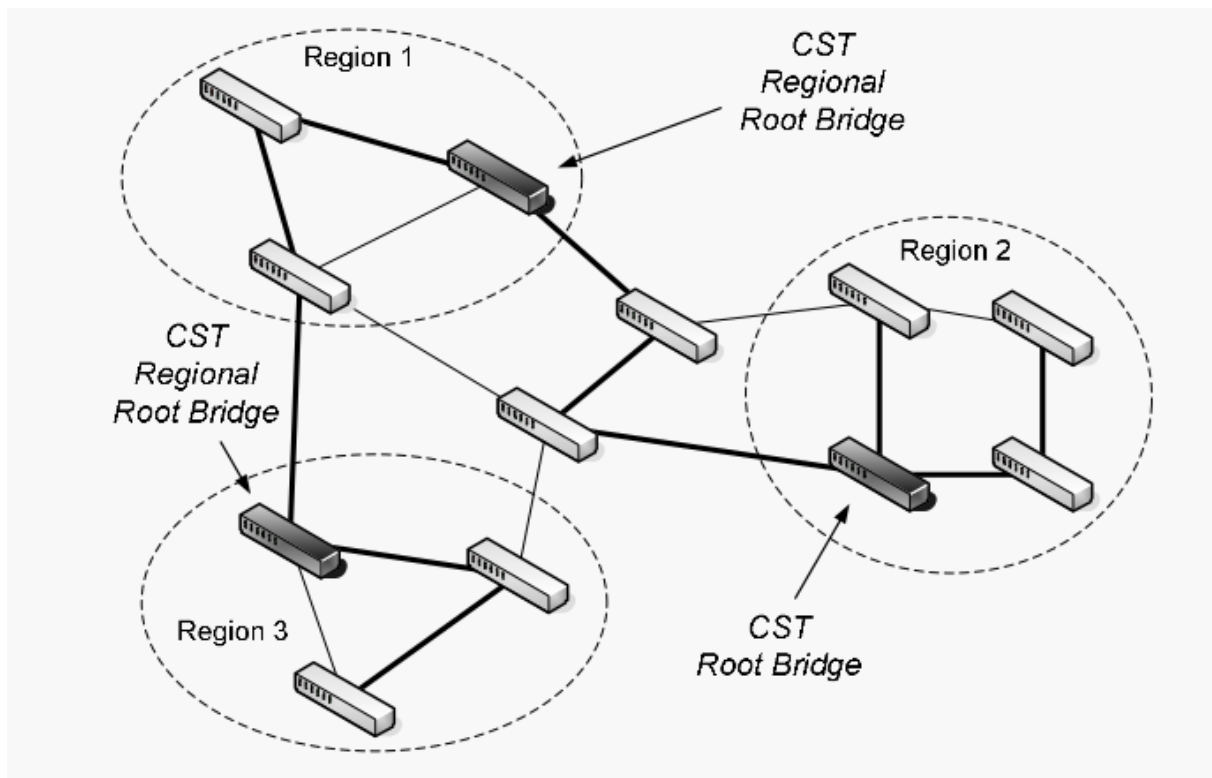


Figure 42: Multiple Spanning Tree protocol