**Rascunho - TP DD**

**Spanning Tree Protocols**

**Explicando os protocolos spanning tree**

O protocol STP foi o primeiro protocolo de spanning tree a ser desenvolvido. É um protocolo para equipamentos de nível 2 que corre em bridges e switches com o objetivo de contribuir para o sucesso da redundância em redes locais, evitando assim a existência de loops em VLAN.

Inicialmente foi desenvolvido pela Radia Perlman e mais tarde foi adotado e adaptado pelo IEEE a partir da Norma IEEE 802.1D.

Existem diversos protocolos: STP, RSTP (respetivas alternativas cisco PVST+ e Rapid PVST+) e ainda o MSTP.

Estes protocolos definem uma tabela de forwarding nos switches e bridges de modo que uma rede constituída por estes equipamentos fique livre de ciclos prejudiciais à disponibilidade de uma rede.

**Explicando o STP**

Para a redundância de caminhos existir e a criação de loops ser evitada em simultâneo, o STP define uma espécie de árvore que abrange todos os switches de uma rede.

Este protocolo força a que certos caminhos fiquem no estado de “standby” (blocked) e deixa outros no estado de “forwarding”. Se uma das ligações ficar indisponível, o protocolo adapta a rede de maneira que a porta que estava em estado “block” entre em ação, assegurando assim a continuidade dos serviços prestados pela rede.

## Funções das portas

Existem 3 funções em que as portas dos switches podem ficar perante este protocolo:

* **Designated Port:** interface eleita para ficar responsável por transportar o tráfego à root bridge em cada segmento. Estas ficam no estado de “forwarding”
* **Root Port:** A melhor interface de um switch (que não a root bridge) para conduzir o tráfego até à root bridge. Estas ficam no estado de “forwarding”
* **Alternate Port:** Portas bloqueadas que podem vir a ser ativadas quando necessário. Estas ficam no estado de “blocking”

**Estados das portas**

Existem 4 estados nos quais as portas podem estar. Em Blocking, que servem como backup e não encaminham tráfego, prevenindo assim loops indesejados e Listening e Learning que são estados de aprendizagem de MAC Address, sendo que é em Learning o estado em que todas as portas iniciam quando um switch é ligado. Nestes 3 estados, as interfaces conseguem receber e processar BDPUs. Por último, existe o estado de Forwarding, em que as portas encaminham pacotes BPDUs para outras interfaces e é por elas que passa todo o tráfego.

**Como funciona o STP?**

Todos os switches da rede elegem uma root bridge, ou seja, o switch que é o ponto focal da rede. É a partir daí que decisões como quais portas bloquear e quais meter no modo “designated port” ou “root port” vão ser tomadas. As mesmas vão ser escolhidas tendo em conta o posicionamento da root bridge.

Se existirem diferentes VLANs, as mesmas devem ter diferentes bridge root, visto que diferentes VLANs tem o seu próprio domínio de Broadcast.

**Como eleger então a root bridge?**

A eleição da root bridge é feita por duas maneiras: por decisão do gestor de rede ou pela escolha automática feita entre os diferentes switches que se baseia na escolha do switch com uma Bridge ID menor. Para isto, todos os switches vão trocando informações entre si que são transportadas pelas Bridge Protocol Data Units (BPDUs), enviando e recebendo aos vizinhos estes dados. Assim cada switch compara os parâmetros da BPDU que recebe com aqueles que envia.

**Como é que então os switches comunicam com as BPDUs?**

Todos os switches enviam ou recebem mensagens Hello, de 2 em 2 segundos, para um grupo multicast (01-80-C2-00-00-00), no qual todos os switches estão subscritos. Nestas mensagens são enviadas BPDUs que são geradas pela Root Bridge e os restantes switchs, quando recebem a BPDU pela root port, reenviam ao switch vizinho.

Estas mensagens servem para:

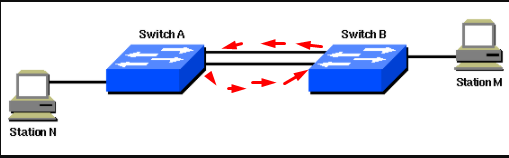
* + Eleger um switch como o Root Bridge.
  + Eleger as portas designated bridge, root port e as bloqueadas.
* Exemplo de uma mensagem:

Uma imagem com texto

Descrição gerada automaticamente

* Nestas mensagens BPDU conseguimos ver:
  + Root ID, identificação do switch que é Root Bridge - - contém a bridge priority, o extend system ID(usado para?) e ainda o mac address do switch
  + Root Path Cost, que apresenta o custo da ligação entre a Root Bridge e o switch que enviou o BPDU
  + Bridge ID da máquina que originou a mensagem - contém a bridge priority, o extend system ID(usado para?) e ainda o mac address do switch
  + Message age, que representa a distância em switchs até à Root Bridge. Cada switch que não seja root e que envie uma BPDU incrementa este valor em 1.
  + Max age, que representa o período máximo de tempo possível de existir antes que uma porta de uma bridge guarde as informações da BPDU.
  + Hello time – intervalo de tempo entre as mensagens enviadas
* Quando a eleição é feita automaticamente entre os switchs, no início, cada switch considera-se a Root Bridge e envia de 2 em 2 segundos estas mensagens BPDU com o respetivo Bridge ID e o root ID ao qual está associado.
* Se um switch A anunciar (para um switch B) um Root ID inferior ao Root ID ao que o B está a anunciar, então este último deixar de enviar aos seus vizinhos o Root ID que tinha, guarda o inferior que recebeu do A e começa a anunciar também este.
* Sendo assim, quando um switch recebe uma mensagem com um root ID inferior ao seu, atualiza o seu parâmetro do Root ID até todos os equipamentos chegarem a um acordo de qual é o switch com um identificador menor e aí consideram esse switch como a Root Bridge.

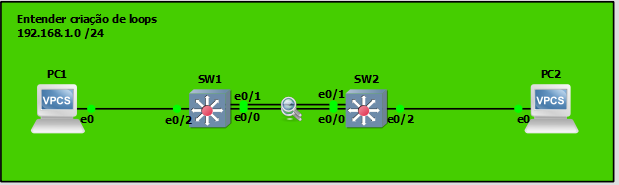
No entanto, é aconselhável que seja o gestor de rede a proceder a esta decisão tendo em conta fatores como o posicionamento do switch na rede em causa, que deve ser o mais central possível.

**Como são criados os loops?**

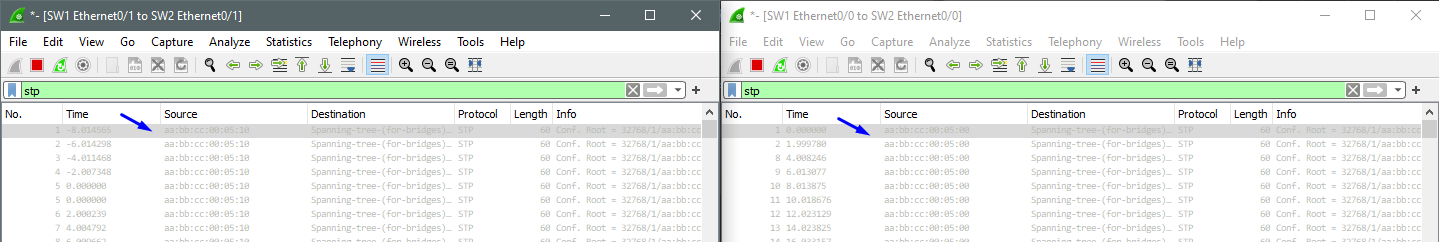
Considerando este esquema, conseguimos ver que existe uma ligação redundante entre o swA e o swB.

No entanto, com estas ligações e sem nenhum protocolo implementado, existe uma grande probabilidade de existirem loops entre os dois switches, visto que por exemplo um pacote Broadcast ou Multicast transmitido do pcM para o pcN, entra num ciclo entre os dois switches.

**Fazendo esta experiência no gns3 para perceber os ciclos**



Inicialmente, com o STP ativado e os PCs a comunicarem entre si, **capturou-se tráfego nas duas ligações entre os switchs.**



Conseguimos capturar pacotes STP a serem enviados de 2 em 2 segundos em ambos os caminhos pelo switch 1, que era a Root Bridge.

De seguida, **desligou-se o stp em ambos os switchs** com o seguinte comando: no spanning-tree vlan 1-1014 (desativa o spanning tree de todas as vlans possíveis).

Uma imagem com texto

Descrição gerada automaticamente Uma imagem com texto

Descrição gerada automaticamente

**Realizou-se um ping entre o PC1 e o PC2 e iniciou-se a captura de tráfego nas duas ligações entre os switchs.**

A comunicação entre os dois pcs nunca ocorreu.

Uma imagem com texto

Descrição gerada automaticamente

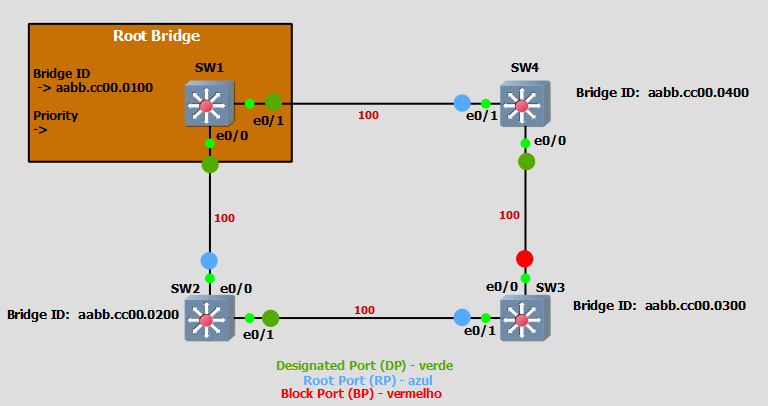
Analisando a captura de tráfego nas duas ligações, conseguimos perceber que existiu uma troca contínua de pacotes ARP, nos dois caminhos, enviadas pelo mesmo equipamento. Estas mensagens ocorrem com intervalos de tempo inferiores a 1 segundo e isto ocorre porque, uma vez que o STP foi desligado, o SW1 recebe o pedido ARP e envia-o para o SW2 por uma das interfaces. O SW2 recebe esse pedido e volta a reencaminhá-lo pela outra interface, ou seja, pela outra ligação.

Estas trocas de mensagem estão sempre e ocorrer e o SW2 nunca consegue entregar os pacotes ao PC2, visto que recebe por uma interface (de uma das ligações) e envia por outra que está também ligada ao SW1. Aqui se formam então os ciclos que congestionam uma rede e impossibilitam, por exemplo, a comunicação entre equipamentos.

Depois deste teste, os switchs ficaram de tal maneira sobrecarregados que as suas consolas deixaram de funcionar.

Uma imagem com texto, captura de ecrã, interior

Descrição gerada automaticamente

**Fazendo no gns3 uma topologia para perceber o stp – mudar esta foto depois de acabada**

Ao fazer a topologia de cima no gns3, o switch 1 ficou como Root Bridge, visto que:

* Todos os switches vinham com a prioridade igual (32769)
* O SW1 é o que tem o identificador com um número mais baixo (aabb.cc00.**0100)**
  + Quando assim é, o switch escolhido é o que tem um Bridge ID menor

Uma imagem com texto

Descrição gerada automaticamenteUma imagem com texto

Descrição gerada automaticamenteUma imagem com texto

Descrição gerada automaticamenteUma imagem com texto

Descrição gerada automaticamente

**As portas foram estabelecidas como a topologia indica:**

* As do SW1 são ambas Designated porque é a root bridge levando a que as interfaces dele fiquem com esta designação.
* No SW2 temos a e0/0 como Root Port porque esta é a porta do switch que acarreta menos custo (100) para chegar até ao SW1. A e0/1 ficou como designated pela mesma lógica, visto que é a melhor interface de entre as existentes na ligação SW2-SW3 para se obter menos custo (100) até chegar à Root Bridge.
* No SW4 a lógica é a mesma do SW2. A e0/1 é a interface que menos custo total leva para chegar à root bridge e a e0/0 é a melhor porta da ligação SW4-SW3.
* Por fim, no SW3, temos empate para eleger a Root Port, visto que ambas as interfaces levam o mesmo custo (200) para chegar ao SW1. Nesta situação, escolhemos a interface que estabelece uma ligação ao vizinho que tem menor Bridge ID. Visto que tanto o SW2 como o 4 têm a prioridade igual, então o aspeto a ter em conta é o switch com menor MAC da Bridge ID, que neste caso é o SW2. Portanto, ficamos com a e0/1 como Root Port e a e0/0 como porta bloqueada visto que é a única porta que não pode levar outra designação que não esta. Esta porta ficará como de backup para uma falha.

**Fazendo capturas de tráfego STP**

Foram feitas capturas de tráfego em: SW1-SW4 e SW4-SW3

**SE CALHAR METER FOTO da topologia, COM AS CAPTURAS**

Uma imagem com texto

Descrição gerada automaticamente**Captura de tráfego entre SW1 e SW4**

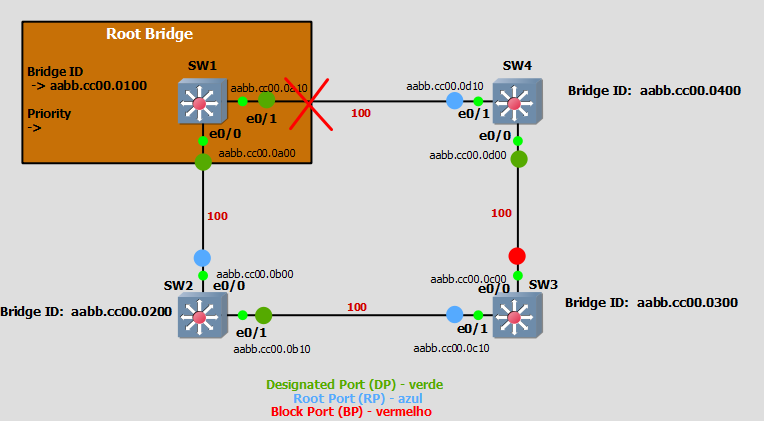
* A mensagem teve origem na interface e0/1 do SW1 e foi enviada para o grupo multicast.
* Contém o Root Id do switch 1 que é o Root Bridge da rede e também os identificadores do switch que enviou a mensagem, o sw1. Neste caso estes dois aspetos coincidem, visto que o pacote foi enviado pelo SW1 que é a Root Bridge.

Uma imagem com texto

Descrição gerada automaticamente**Captura de tráfego entre SW4 e SW3**

* A mensagem teve origem na interface e0/0 do SW4 e foi enviada para o grupo multicast.
* Contém o Root Id do switch 1 que é o Root Bridge da rede e também os identificadores do switch que enviou a mensagem, o sw4. Neste caso conseguimos ver que os mesmos são diferentes.

**Injeção de falhas e análise das consequências**

**Falha na e0/1 do sw1**

**TIRAR OUTRA FOTO DEPOIS DISTO**

**Estado das portas dos Switches antes da falha**

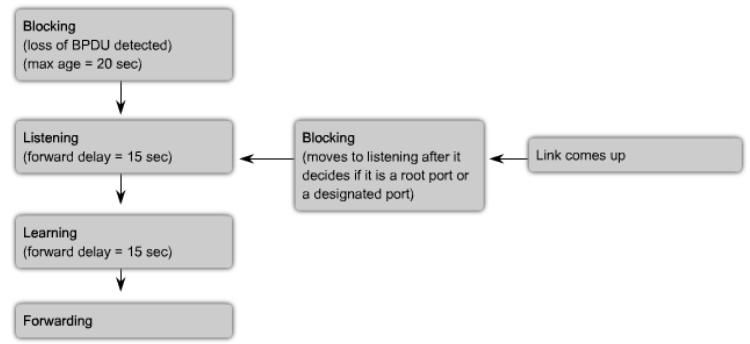
O sw1 está como em baixo restantes switches encontram-se com as portas como a figura da topologia indica

Uma imagem com mesa

Descrição gerada automaticamente

**Se calhar não é preciso meter esta foto, apenas dizer que tudo está igual**

**Quando se injeta a falha na porta referida**, ao fazer uma captura de tráfego entre o sw1 e o sw4, conseguimos ver que os pacotes STP deixam de ser enviados durante, aproximadamente, 20 segundos. Ou seja, um switch demora 20 segundos a perceber que não foi enviada nenhuma BPDU do vizinho. Depois, o switch que tem a porta de backup (a que está no estado de bloqueada), coloca a mesma em Listening durante 15 segundos e em seguida no estado de Learning também durante 15 segundos. Só após estes períodos é que a interface passa para o estado de Forwarding e o serviço é restabelecido. Podemos concluir que o STP demora 50 segundos a reagir a uma falha e a colocar a rede novamente disponível.

 fazer um gráfico tipo este na apresentação?

No último pacote enviado antes da injeção da falha, conseguimos ver que a flag “topology change” encontra-se no estado “no”, como podemos ver a baixo.

Uma imagem com texto

Descrição gerada automaticamente

No primeiro pacote enviado depois desses 20s, já conseguimos ver que a flag “topolgy change” encontra-se a “yes”, visto que houve uma alteração na topologia. Esta flag é colocada a yes pela Root Bridge quando esta recebe uma TCN (explicado a baixo).

Com isto, como cada Hello BPDU leva 2 segundos, percebemos que é ao final de 10 Hello BPDU’s não enviados que o protocolo percebe que algo na rede aconteceu.

Uma imagem com texto

Descrição gerada automaticamente

Se fizermos capturas nas restantes 3 ligações, conseguimos ver que é enviada, em todas elas, uma mensagem que segue os mesmos aspetos de origem e destino, mas que, no entanto, apenas contém a informação de que ocorreu uma mudança na topologia, para assim os restantes switches conseguirem reajustar as suas portas.

Esta mensagem é uma Toplogy Change Notification e é uma BPDU especial que não contém qualquer tipo de informação e é enviada em direção à Root Bridge pelo switch que deteta a mudança na rede.

Uma imagem com mesa

Descrição gerada automaticamente

**Estado das portas dos Switches antes da falha**

Depois da falha, o SW1 continua com as portas no mesmo estado, tal como o SW2.

No entanto o mesmo não se passa com os restantes:

* A root port do sw4 passa a ser a e0/0, visto que a anterior (e0/1) está ligada à porta da root bridge que se desligou.
* A e0/1 passa a ser designated, visto que agora o caminho tem de passar por essa porta e dar a volta à topologia.
  + **METER FOTO DO ANTES(TIRAR DESTE) E DEPOIS**

Uma imagem com texto

Descrição gerada automaticamente

* A interface e0/0 do SW3 que se encontrava bloqueada, passou a estar no estado de designated root
  + É que aqui que o STP trabalha, mantendo a redundância e assegurando a resolução de falhas na rede.
  + **METER FOTO DO ANTES(TIRAR DESTE) E DEPOIS**

Uma imagem com texto

Descrição gerada automaticamente

**Pingando do PC4 para o PC2 e injetar falha na e0/1 do sw1**

Inicialmente, o PC4 consegue pingar sem problemas o PC2. Os echo request e reply vão da interface e0/1 do sw4, visto que é esta a root port (conseguimos comprovar isso pois obtivemos esse mesmo tráfego entre o sw4 e o sw1) e por fim chegam à e0/0 do sw2.

Quando é efetuada uma falha na interface e0/1 do SW1, conseguimos ver que a primeira captura (a da esquerda) deixa de detetar tráfego ICMP e passado algum tempo, é a captura sobre a ligação entre o SW4 e o SW3 que interceta tráfego, visto que o SW4 muda a sua root port para a e0/0.

Uma imagem com mesa

Descrição gerada automaticamente

Como conseguimos ver, o PC4 durante 50 segundos sem conseguir comunicar com o PC2.

Uma imagem com texto

Descrição gerada automaticamentemeter no ppt apenas?

Nas imagens seguintes também conseguimos ver, com a ajuda do comando “sh spanning-tree”, a passagem pelos 2 estados (depois dos 20 segundos sem BPDUs), que a porta que estava bloqueada no SW3 sofreu.

Uma imagem com texto

Descrição gerada automaticamente Inicialmente Bloqueada.

Uma imagem com texto

Descrição gerada automaticamenteDepois passa para Listening

Uma imagem com texto

Descrição gerada automaticamenteDepois de Listening passa para Learning

Uma imagem com texto

Descrição gerada automaticamentePor fim, começa a encaminhar tráfego

**O que concluir depois da captura de tráfego e injeção de falhas?**

Conclui-se que o protocolo spanning tree consegue adaptar-se perante falhas na rede, mudando as configurações das portas de cada router perante a falha ocorrida. As portas que se encontravam bloqueadas, entram em ação para assumirem o papel de designated root para assim ser possível a continuação dos serviços.

Os custos dos caminhos de cada switch até à root bridge acabam por mudar, visto que as portas do switches mudam para outras funções.

**DEITAR SW3 E0/0 A BAIXO E VER O QUE ACONTECE NO WIRE E NO SH SPAN**

**GUARDAR TUDO E DEPOIS EXPERIMENTAR METER PRIORIDADES NOS SWITCHES E VER O QUE ACOTNECE DE DIFERENTE**

***SE CONTINUAR IGUAL, DEIXAR ESTAR E METER NA TOPOLOGIA AS PRIORIDADES***

**Explicando o RSTP**

<https://www.omnisecu.com/cisco-certified-network-associate-ccna/difference-between-stp-and-rstp.php>

O protocolo Rapid Spanning-Tree é visto mais como uma evolução do STP do que uma revolução e é definido no padrão IEE 02.1w. Este protocolo, melhora o tempo de convergência (QUE É ISTO?) do STP ao reduzir para apenas 3 Hello BPDUs para a rede perceber que tem de se adaptar perante uma alteração. Isto leva a uma interrupção de serviço de apenas 6 segundos e podemos concluir que o RSTP é um protocolo melhor que o STP.

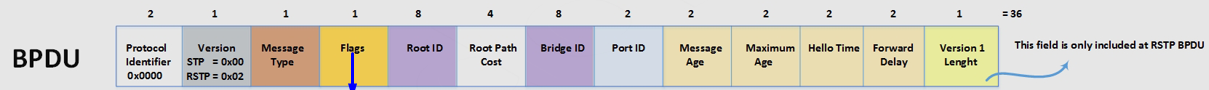
**Diferenças entre o STP**

As características do protocolo RSTP não diferem muito do STP. Temos novos tipos de papéis para as portas: Alternate e Backup. As mesmas podem estar nos estados de “Learning”, ”Forwarding” e “Discarding”(que substitui os estados de Disabled, Blocking e Listening),.

Sendo assim o RSTP é compatível com o STP, no entanto, temos diferenças no preenchimento do campo *Flag* e no envio das BPDUs.

O que torna este protocolo mais rápido, é que o mesmo é capaz de confirmar rapidamente que uma porta pode passar para o estado de forwarding sem depender de qualquer temporizador. Além disso, para ser alcançada uma convergência mais rápida numa porta, existem as Edge Ports e os Link Type.

**Mudanças no campo Flag da BPDU**



Uma imagem com mesa

Descrição gerada automaticamente

**Isto é só para o ppt?**

**Mudança na utilização das BPDUs**

Com o RSTP, um switch envia uma BPDU com as suas informações atuais a cada hello-time mesmo se não receber nenhuma da Root Bridge. Com isto, um switch consegue perceber se o seu vizinho entrou em falência se o mesmo não lhe enviar nenhuma BPDU.

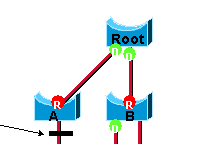
Este aspeto é diferente do STP, visto que se um switch vizinho não enviar nenhuma BPDU, não é certo que tenha existido uma falência do mesmo. Como as BPDUs partem da Root Bridge, pode ter existido uma falha numa ligação desde a Root Bridge até ao switch vizinho em causa e este ainda se encontrar ativo.

## Funções das portas

As portas com o papel de designated port e root port continuam a desempenhar a mesma função que no STP. No entanto, a alternate port que tínhamos no protocolo anterior, pode ser backup port ou alternate port. Estas funções, tal como o STP, são determinadas perante o conteúdo trocado nas BPDUs, entre os switchs.

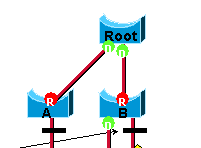
Alternate **Port**

Porta alternativa à Root Port, ou seja, fica de backup num switch para caso a Root Port desse mesmo switch entrar em falência. Se isto acontecer, a Alternate Port passa a Root Port e oferece outro caminho para a Root Bridge.



**Backup Port**

Porta que fica de backup à Designated Port. Se a porta Designated falhar, esta assume esse papel e oferece redundância na conectividade.



## Edge Ports

As portas deste tipo não são usadas em ligações para com switchs, mas sim para por exemplo switch-pc e não geram avisos de alterações na topologia. Correspondem às portas FastPort. A porta que fica com esta característica consegue passar imediatamente para o estado de forwarding, saltando o de listening e learning. Para serem configuradas: spanning-tree portfast.

## Estados possíveis das ligações

Para além das edge ports, para existir uma transição mais rápida de uma porta para o estado de encaminhamento, o RSTP tem em conta também o tipo de ligação dessa porta. Se uma porta estiver configurada em full-duplex, ela é considerada do tipo point-to-point. Se estiver em half-duplex já é do tipo shared port. Hoje em dia, nas redes entre switchs, a maioria das ligações trabalha em full-duplex e são consideradas ligações do tipo point-to-point, o que possibilita uma transição para os estados de encaminhamento mais rápida.

## Estados das portas

Discarding: impede o encaminho de pacotes entre o switch com a porta nesse estado e o switch vizinho. Corresponde ao estado de Blocking do STP e de Listening, visto que o RSTP usa o Discarding para também “ouvir” o MAC Address.

Learning: estado em que a interface aprende o seu MAC Address.

Forwarding: estado de encaminhamento de pacotes

# Diferenças na convergência entre STP e RSTP

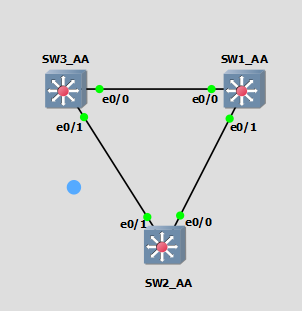
## Convergência STP – mudar fotos, depois

## 

Na presente topologia, colocou-se o comando “spanning-tree vlan 1 root primary” e temos assim o SW1 como a Root Bridge. Sem (ainda) uma ligação entre o SW2 e o SW3, não temos a possibilidade de existirem loops. Com isto, ambas as portas do SW1 encontram-se na função de designated e as do SW2 e 3 que se ligam à Root Bridge, encontram-se em root

## Ao ser adicionada a ligação entre SW2 e SW3- mudar fotos, depois

Foi adicionada uma ligação entre o switch 2 e 3, capturou-se tráfego nas 3 ligações e usou-se o comando “sh spanning-tree” nos 3 switchs, logo no momento da criação da nova ligação.



Na captura realizada entre o SW2 e o SW1, conseguimos ver que o SW2 envia uma TCN para o SW1, visto que foi o switch 2 que detetou a mudança na rede e avisa então a Root Bridge.

Uma imagem com texto

Descrição gerada automaticamente

A Root Bridge recebe este TCN e responde logo de seguida com uma BPDU onde a flag da topology change acknowledgment vem a yes, para quem enviou o avisou de que a rede mudou, saber que a Root Bridge já tem conhecimento disso.

Uma imagem com texto

Descrição gerada automaticamente

A porta e0/1 do SW3 ficou como designated e a e0/1 do SW2 ficou como alternate, ou seja, bloqueada. Isto aconteceu porque o SW3 mandou BPDUs para o SW2 como informações melhores na Bridge ID, neste caso com um identificador mais baixo (como podemos ver na imagem seguinte). Sendo assim, o SW2 percebe que “perde” em relação ao switch 3 e bloqueia a sua porta.

Uma imagem com mesa

Descrição gerada automaticamente

Neste aspeto, o STP resolve perfeitamente o problema e consegue calcular, eficientemente, as novas alterações na rede. No entanto, este processo demora sempre imensos segundos e nisto, o RSTP é mais rápido.

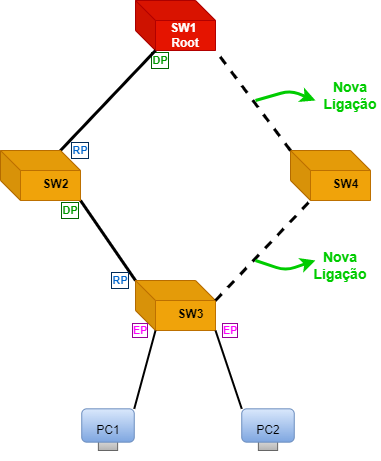
## Convergência RSTP – mudar fotos, depois

Uma imagem com texto, eletrónica, estacionamento, computador

Descrição gerada automaticamente

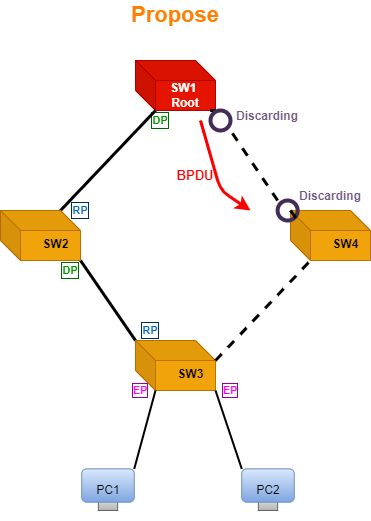
Considerando uma topologia como a da imagem anterior, temos 3 switchs onde o SW1 é considerado a Root Bridge. As portas de cada switch têm as funções tal como é sugerido na imagem.

De seguida, como é visível na imagem seguinte, se adicionarmos um novo Switch à rede são criadas duas novas ligações e isto vai causar uma alteração à mesma.

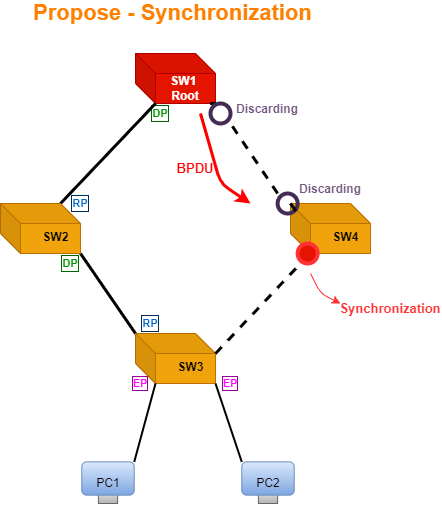


É aqui que o processo de negociação entre o SW1 e o SW4 começa, o proposal/agreement. No entanto, antes disso, ambos os switchs colocam as portas que fazem parte dessa nova ligação no estado de “discarding”

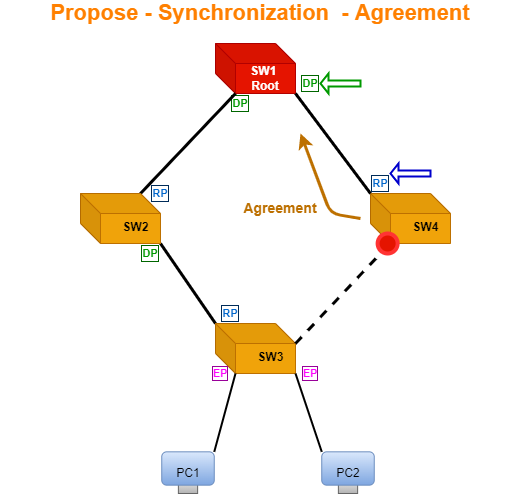
De seguida, o processo de propose inicia e o SW1 envia uma proposta (ao SW4) em forma de uma mensagem BPDU com as suas informações (Bridge ID, path cost, etc). Este propose tem como intuito o SW1 mostrar que é a Root Bridge e propor colocar a sua porta no modo Designated e ainda e indica ao SW4 que deve colocar a sua no modo Root.



Quando o SW4 recebe esta BPDU, percebe que é melhor do que aquela que ele poderia oferecer. No entanto, antes de aceitar a proposta vinda da Root Bridge, o SW4 passa primeiro pelo processo de sychronization. Aqui, o switch tem o cuidado de colocar todas as suas portas, que não sejam edge ports, bloqueadas. Este passo serve para assegurar que nenhum loop seja criado durante a negociação entre os switchs, ou seja, durante o processo de proposal/agreement.

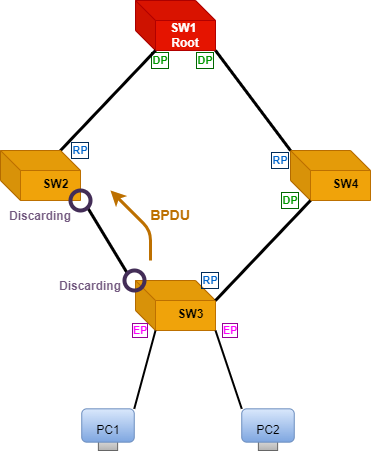


Depois de feita a sincronização, o SW4 vai então aceitar a proposta que foi feita pelo SW1 e envia a mensagem de agreement ao SW1 e quando o mesmo a receber, a nova ligação fica estabelecida com as decisões que foram concordas entre ambos: designated port do lado do SW1 e Root Port do lado do SW4.

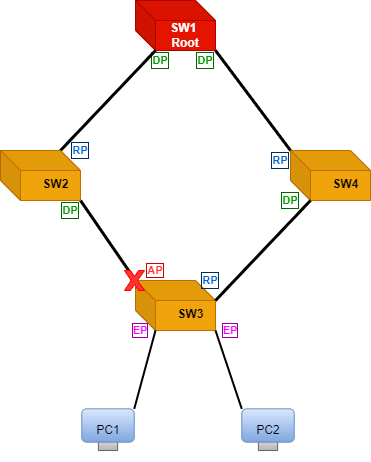


Com este conjunto de processos, esta ligação fica pronta e os processos são iguais para a outra ligação (entre o SW4 e SW3). Contudo, mesmo as ligações que já existiam, podem vir a sofrer alterações e para isso, também o processo de proposal/agreement é usado. A ligação entre o SW2 e o SW1 acaba por ser calculada de igual forma como a que foi explicada anteriormente, mas a que existe entre o SW3 e o SW2 tem algumas diferenças.

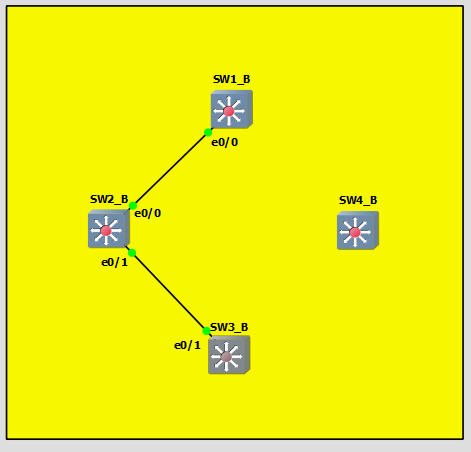
Assim, no caso da ligação entre o SW3 e o SW2, o switch 3 vai enviar uma BPDU ao SW2 com as suas características.



O SW2, ao receber isto, percebe que tem uma BPDU melhor e que não pode ser do lado do SW3 que ficará a Designated Port, não aceitando assim a proposta que lhe foi enviada. Com isto, os switchs percebem imediatamente que o SW3 tem de bloquear a sua porta, evitando assim a criação de loops. Assim, do lado do SW2 fica uma Designated Port e do lado do SW3 fica uma Alternate Port. Sendo assim, a topologia adapta-se rapidamente às novas alterações implementadas, sem o uso de timers para estes processos, resultando no que a imagem seguinte apresenta.



# Vendo a convergência do RSTP no GNS3 – mudar foto depois, tirar mesmo a ligação que penso q não faz mal



## Configuração do protocolo RSTP

Meter aqui os comandos

## Capturando tráfego entre os switchs

Foi feita uma captura de tráfego em cada uma das duas ligações da topologia.

Uma imagem com texto

Descrição gerada automaticamente

Segundo a imagem anterior conseguimos ver que são enviados de cada vez, dois pacotes STP ao mesmo tempo. Um deles continua com o mesmo destino que o protocolo normal de spanning-tree. O outro pacote é enviado também para um grupo multicast/Broadcast do pvst+(protocolo explicado mais à frente). A mensagem teve origem na e0/1 do SW2.

Estes dois pacotes apresentam o mesmo conteúdo. Inicialmente conseguimos ver que a versão RSTP está identificada.

Tal como referido anteriormente, este protocolo apresenta mais flags. Como a topologia ainda não sofreu nenhuma alteração, as flags de topology change (as comuns ao STP) encontram-se a “no” e as de Agreement e Proposal também se encontram a “no”. Já a flag que indica a função da porta por onde foi enviada a mensagem, encontra-se em designated, visto ser esta a função da e0/1 do switch que enviou o pacote. As flags de Learning e Forwarding encontram-se a “yes” e assim se mantiveram enquanto não houve uma alteração à rede.

Na imagem seguinte conseguimos ver o que foi referido.

Uma imagem com texto

Descrição gerada automaticamente

Continuando a analisar o pacote, conseguimos ver a identificação da Root Bridge e da Bridge que enviou a BPDU. Vemos o custo do caminho até ao switch principal, vemos message age e o max age e o Hello Time que é igual ao protocolo estudado anteriormente.

## Estado da topologia

O SW1 é a Root Bridge. Como ainda não estabelecemos ligações do SW4 para o SW1 e SW3, ainda não temos a possibilidade existirem loops. Assim, a interface do SW1 encontra-se na função de designated, tal como a e0/1 do SW2. Já a e0/0 do 2º switch está em root e a do SW3 também.

## Ao ser adicionada o SW4 - mudar fotos, depois

Foi adicionado um novo switch, o SW4, que se ligou ao SW1 e 3.

No momento da adição desse novo equipamento, foi introduzido o comando “sh spanning-tree” em todos os switchs e capturou-se tráfego STP em todas as ligações.

## Analisando o sw3 para ver a negociação e sincronização

Analisando o sw3 conseguimos ver que, inicialmente, como a interface e0/0 ainda não se encontrava com uma ligação estabelecida, a mesma passa para um estado de blocking, enquanto a Root port ainda se mantinha do lado da e0/1.

Uma imagem com texto

Descrição gerada automaticamente

Seguidamente, a negociação com o sw4 resultou em que DP ficasse do lado deste último switch. Portanto a e0/0 do sw3 ficou com o papel de Root port.

Como o SW3 também teve de fazer o proposal/agreemente com o SW2, conseguimos ver que o switch 3 fez a sincronização durante a negociação. Com isto, bloqueou todas as portas que não fossem edge ports (e0/2-3).

Com isto, visto que o sw2 apresentava uma BPDU melhor, ficou então decidido que a DP ficaria do lado desse switch e a porta alternate ficaria do lado do sw3, para impedir a criação de loops.

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

**Analisando os pacotes no GNS3**

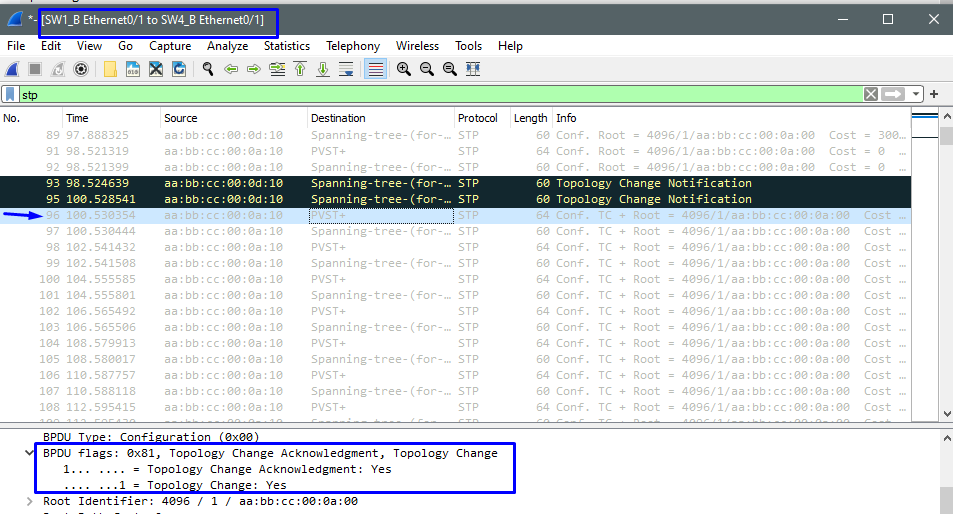
## Ligação SW1-SW4 e SW4-SW3

Aquando da introdução do novo switch nada rede, conseguimos ver a presença de pacotes STP do tipo Topology Change Notification em duas capturas. Estes pacotes são enviados para a Root Bridge por quem deteta a alteração na rede. Neste caso, foi o SW3 primeiramente a detetar essa mudança e enviou a TCN para o SW4 que posteriormente enviou para a Root Bridge.

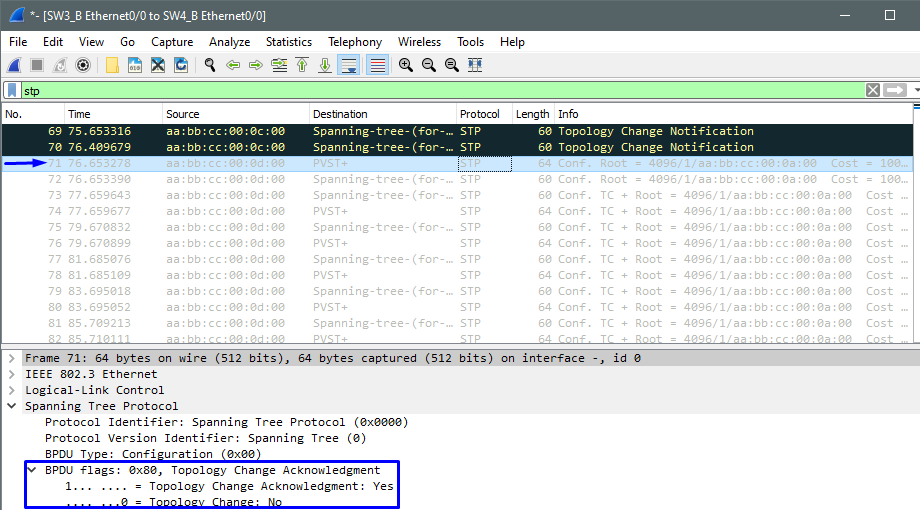
Uma imagem com texto

Descrição gerada automaticamente

Analisando o pacote imediatamente a seguir ao TCN no link SW1-4, vemos que o mesmo já é enviado pela Root Bridge. Esta BPDU serve para o SW1 avisar quem detetou a mudança que já sabe da alteração na rede. Assim, esta mensagem vem com a flag “topology change acknowledgment” ativa, tal como a flag “topology change”.

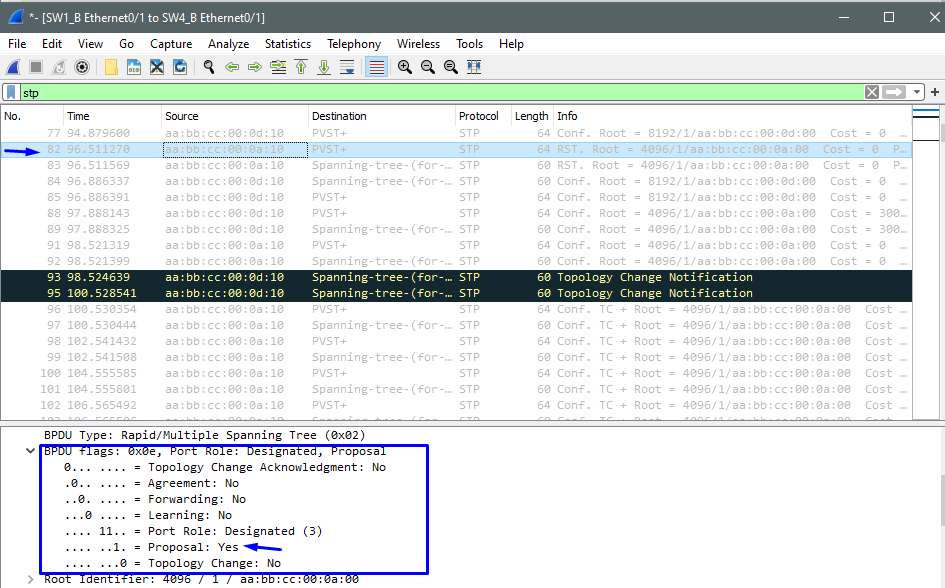


Olhando para a ligação entre o SW4 e o SW3 percebemos que o 4º switch, depois de receber pela Root Bridge a BPDU referida anteriormente, vai enviar também uma BPDU com as mesmas características para o SW3, visto que foi este a detetar a alteração na rede.



## Pacotes Proposal

No tráfego entre o SW1 e o SW4, antes da mensagem TCN, é possível ver que o SW1 envia uma BPDU ao SW4 com a flag “proposal” ativa. Isto demonstra o processo de proposal/agreement entre estes dois switchs. Neste caso, é o SW1 que está a propor ao switch 4 que a porta designated fique do lado do switch 1.



## Pingar do PC1 para o PC3 e injetar falha

Foi realizado um ping do PC1 para o PC3 e, de seguida, foi injetada uma falha na interface e0/0 do SW3, visto ser esta a Root port do switch a que o PC1 se encontra ligado.

Como é possível de ser confirmado pela imagem seguinte, o protocolo RSTP é tão rápido a responder a uma falha que nem se deteta nenhuma alteração no ping entre os PCs.

Uma imagem com texto

Descrição gerada automaticamente

O SW3 rapidamente muda a sua porta root para porta que estava como backup, com o papel de Alternate. Assim, a e0/1 passa para Root Port.

Uma imagem com texto

Descrição gerada automaticamente

# Explicando o MSTP

O protocolo Multiple Spanning Tree foi criado segundo a IEEE…. O mesmo é baseado no protocolo RSTP mas tem em conta o tratamento de várias Vlans. Enquanto que nos protocolos anteriores (STP e RSTP) cada vlan representa uma instância, no MSTP podemos dividir x Vlans num número menor de instâncias. Ou seja, se tivermos 10 vlans, em vez de termos o processamento de um protocolo para cada uma das vlans, agrupamos as mesmas numa instância que trata da vlan 1 até à 5 e ou tra instância que trabalha com a vlan 6 até à 10. Com várias instâncias, não é obrigatório que seja apenas um switch a ser a Root Bridge de cada uma das instâncias. Diferentes instâncias podem ter diferentes switchs root. Este protocolo, causa um impacto menos prejudicial à computação dos equipamentos.

O Multiple Spanning Tree Protocol (IEEE 802.1s) é uma evolução do RSTP, tendo como objetivo principal permitir múltiplas instâncias RSTP e assim reduzir o tempo de convergência da spanning tree. O padrão foi lançado visando a utilização do RSTP em ambientes com múltiplas VLANs. Ao invés de calcular uma instância para cada VLAN, o MSTP reduz o total de instâncias RSTP.

Cada instância MSTP está associada a um conjunto de VLANs que compartilham a mesma topologia lógica (spanning tree) e que pertencem a uma mesma região. Uma instância é independente de outras instâncias MSTP, sendo assim, o MSTP permite distribuir a carga (de serviço) (load balancing) das instâncias, de forma que o tráfego das VLANs contidas em uma determinada região possa fazer uso de caminhos independentes de outras regiões.

As instâncias, pertencentes a suas respectivas regiões, são conectadas por uma spanning tree comum (CST), possibilitando que diferentes regiões se comuniquem. Numa visão macroscópica, as regiões podem ser vistas como bridges que fazem parte de uma spanning tree (comum).

# MST Region

O protocolo MST opera em regiões. Estas, são grupos de switchs que partilham a mesma configuração que consiste num nome, num revision number que é um número administrativo escolhido pelo gestor e na criação das instâncias dessa região e respetiva atribuição das vlans a essas instâncias. Por defeito, a região atribui a instância número 0 com a revision number igual a 0. Se um conjunto de switchs tiver estas características em comum, eles pertencem à mesma região.

# Topologia MSTP - experimentar regiões

Uma imagem com texto

Descrição gerada automaticamente

Meter depois as características da região de cima e de baixo na topologia mesmo

NO SW1

-» colocar no modo mst

-> conf t, spanning-tree mode mst

-» para configurar o mstp na mesma região

-> Conf t, spanning-tree mst configuration

-> name regiao1, revision 1

-» as configurações só são aplicadas quando fazemos exit

-» configurar mode trunk e access

-> int e0/0, switchport mode access, spanning-tr portfast

-> int range e0/1-2, switchport trunk encapsulation dot1q, switchport mode trunk

-» ao começar por se configurar apenas um switch, sendo que o mesmo estabelece ligações a outros switchs, as portas começam no estado de “learning” e com a designação de Bound. Isto acontece porque as mesmas estão ligadas a equipamentos que correm protocolos de spanning-tree que não o MST. Podemos ver, na imagem seguinte, que nas interfaces em questão aparece Bound(STP).

Uma imagem com mesa

Descrição gerada automaticamente

NO SW2

-» colocar no modo mst

-> conf t, spanning-tree mode mst

-» para configurar o mstp-> Conf t, spanning-tree mst configuration

-> name regiao1, revision 1

-» Agora, se formos ver como está a porta do SW1 (e0/0) que se liga ao SW2, já conseguimos ver que a mesma se encontra sem o Bound(STP), visto que ambos os lados operem o protocolo MST.

Uma imagem com texto

Descrição gerada automaticamente

# Capturando tráfego entre o SW1 e SW2

Analisando tráfego entre o Switch 1 e 2, conseguimos perceber como são as BPDUs do MSTP.

Visto que o Hello Time deste protocolo é igual ao dos restantes, as mensagens são trocadas a cada 2 segundos. Podemos ver que a estrutura do pacote BPDU é muito semelhante ao que já foi estudado até agora, principalmente em comparação ao RSTP, visto que temos também presentes as flags desse protocolo.

Conseguimos comprovar isto, atendendo ao facto de que para além da versão aparecer identificada como “Multiple Spanning Tree”, a BDPU Type vem com “Rapid/Multiple Spanning Tree”, como podemos ver na imagem seguinte.

Uma imagem com texto

Descrição gerada automaticamente

No entanto, estas BPDUs têm um parâmetro muito importante que leva a informação da região para a qual o switch está configurado. Esse parâmetro é o MST Extension.

Assim, no MST Extension temos o nome da região e o número da revisão. Para além disso temos o MST Config digest, que é um código que o switch que recebe a BPDU lê para perceber se está na mesma região que a BDPU recebida. É portanto, com este MST Config Digest, que é definida na BPDU a região onde se enquadra o switch que enviou esta mensagem.

Uma imagem com texto

Descrição gerada automaticamente

Se mudarmos o revision number ou o nome da região, o MST Digest não vai sofrer alterações, como conseguimos ver nas imagens seguintes ao alterar o revision number para 3 e o nome para REGIAO1.

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

Este valor não é alterado (nem com o nome, nem com o número) porque apenas sofre alterações quando, numa região, alteramos as vlans a mapear para as instâncias que pretendermos.

Assim, se criarmos uma nova instância no SW1 e associarmos a mesma a uma vlan (por exemplo a 10), conseguimos ver que o MST Config Digest muda de valor.

-» criar uma instância no SW1 e mapear a vlan10

-> conf t, spanning-tree mst configuration

-> instance 1 vlan 10

-» de notar que a vlan não precisa de estar criada previamente. Apenas estamos a preparar uma instância que irá mapear essa vlan caso a mesma seja criada.

Nas imagens seguintes, conseguimos ver criação da nova instância e a diferença de valores no MST Digest

Uma imagem com mesa

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

# Experiencia usando apenas uma ligação como forwarding entre SW1-SW2

Como na presente topologia, temos duas ligações entre o SW1 e o SW2, podemos usar apenas uma para encaminhar o tráfego das duas instâncias entre os dois switchs. **NO SW1**

-» colocar no modo mst

-> conf t, spanning-tree mode mst

-» para configurar o mstp na mesma região que o SW2 com 2 instancias

-> Conf t, spanning-tree mst configuration

-> name regiao1, revision 1

-> instance 0 vlan 1-5,11-4094

-> instance 1 vlan 6-10

-» configurar mode trunk e access

-> int e0/0, switchport mode access, spanning-tr portfast

-> int range e0/1-2, switchport trunk encapsulation dot1q, switchport mode trunk

**NO SW2**

-» colocar no modo mst

-> conf t, spanning-tree mode mst

-» para configurar o mstp na mesma região que o SW1 com 2 instancias

-> Conf t, spanning-tree mst configuration

-> name regiao1, revision 1

-> instance 0 vlan 1-5,11-4094

-> instance 1 vlan 6-10

-» configurar mode trunk e access

-> int e0/0, switchport mode access, spanning-tr portfast

-> int range e0/1-2, switchport trunk encapsulation dot1q, switchport mode trunk

Uma imagem com mesa

Descrição gerada automaticamente

**Segundo a imagem anterior, conseguimos ver que o SW1 é a Root Bridge de ambas a instâncias.**

Uma imagem com mesa

Descrição gerada automaticamente

**Já pela imagem anterior, conseguimos ver que a porta e0/1 é a Root port do SW2 para ambas as instâncias. Ou seja, está a ser usada apenas uma ligação para duas diferentes. No entanto isto pode ser alterado.**

# Experiencia usando uma ligação para cada instância



Podemos aproveitar quando temos ligações redundantes numa topologia para dividir as instâncias pelas diferentes ligações, não colocando tanto tráfego numa ligação. Para isso, podemos por exemplo aumentar o custo da e0/1 (porta root) numa das instâncias (por exemplo MTS1) e assim, passará a ser a e0/2 que encaminha o tráfego para a Root Bridge

-» Aumentar custo na e0/1 do SW2

-> conf t, int e0/1, spanning-tree mst 1 cost 4500000

Isto vai fazer com que a porta que tenha uma ligação que implica menos custo para a instância 1 seja a e0/1. Assim, como podemos ver segundo a imagem seguinte, o SW2 passa a encaminhar o tráfego das vlans da instância 0 pela porta e0/1 e as vlans da instância 1 pela e0/2. Com isto, estamos a tirar proveito das duas ligações que existem entre os dois switchs.

Uma imagem com mesa

Descrição gerada automaticamente

# Pingando do PC1 para o PC2 e injetar falha

Para se perceber se a rapidez deste protocolo é igual à do RSTP, foi iniciado um ping entre o PC1 e o PC2. Durante o mesmo, foi injetada uma falha na porta Root do SW2 da primeira experiência, a e0/1.

Uma imagem com texto

Descrição gerada automaticamente

Tal como o RSTP, com pings de 1 em 1 segundo, não se nota a falha que existiu na rede, visto que o MSTP usa o protocolo de proposal/agreement do RSTP e acaba por ser tão eficaz quan

**VER O QUE ESTÁ NO PDF DE TL A PARTIR DA 148 e fazer parecido**

**FAZER DEPOIS A CONVERGENCIA DO RSTP COM O PROPOSAL OU O CRL**

**METER OS DOIS PCS A PINGAR, METER UMA FALHA E VER QUANTO TEMPO DEMORA**

**Ver os estados os tempos de cada estado STP porque isso dos 20s não está lá muito bem**

**FAZER A CONVERGENCIA NO RSTP**

**DEPOIS FALAR DOS ESTADOS TANTO NO RSTP COMO NO STP**

**DAR UMA VISTA DE OLHOS NO QUE ANDEI AÍ A MAISCULAR**

**FAZER UM RESUMO ENTRE AS CONVERGENCIAS DO RSTP E O STP?**

**VER DEPOIS EDGE (isto já falei mas está MEH) E ESTADO DAS LIGAÇÃO RSTP NO PPT DE TL – NÃO GRAVE**

**SÓ DEPOIS AVANCAR PARA O MSTP**

**SVER DE NOVO O VÍDEO DO HOMEM – MSTP**

**FAZER TOPOLOGIA DO HOMEM**

**VER SE DÁ PARA FAZER UNS DESENHOS A EXPLICAR**

**METER PCS NA REDE**

**COLOCAR UMA FALHA NA REDE E VER SE É COMO O RSTP**

**NO FIM FALAR NO RELATÓRIO/APRESENTAÇÃO DO QUE O VIDEO DO HOMEM FALA NO FIM**

**No relatório final ir metendo mais imagens de quando se adicionam ligações e assim**

**Ver introdução do guião de dezembro**

[**https://weekly-geekly-es.imtqy.com/articles/pt419785/index.html**](https://weekly-geekly-es.imtqy.com/articles/pt419785/index.html) **🡪 ver melhor isto** para o MSTP no relatório final

**No guião de dexembro substituir fotos por fotos onde tenham o simbolo da root bridge e assim**

**As fotos do mstp deviam ir com as caracteristicas**

**No relatório meter isto melhor**

É um *upgrade* do STP que providencia uma *spanning-tree* separada para cada *VLAN*, configurada na rede. Esta instância suporta as seguintes características:

* *PortFast*
* *UplinkFast*
* *Backbone Fast*
* *BPDU guard*
* *BPDU filter*
* *Root Guard*
* *Loop Guard*