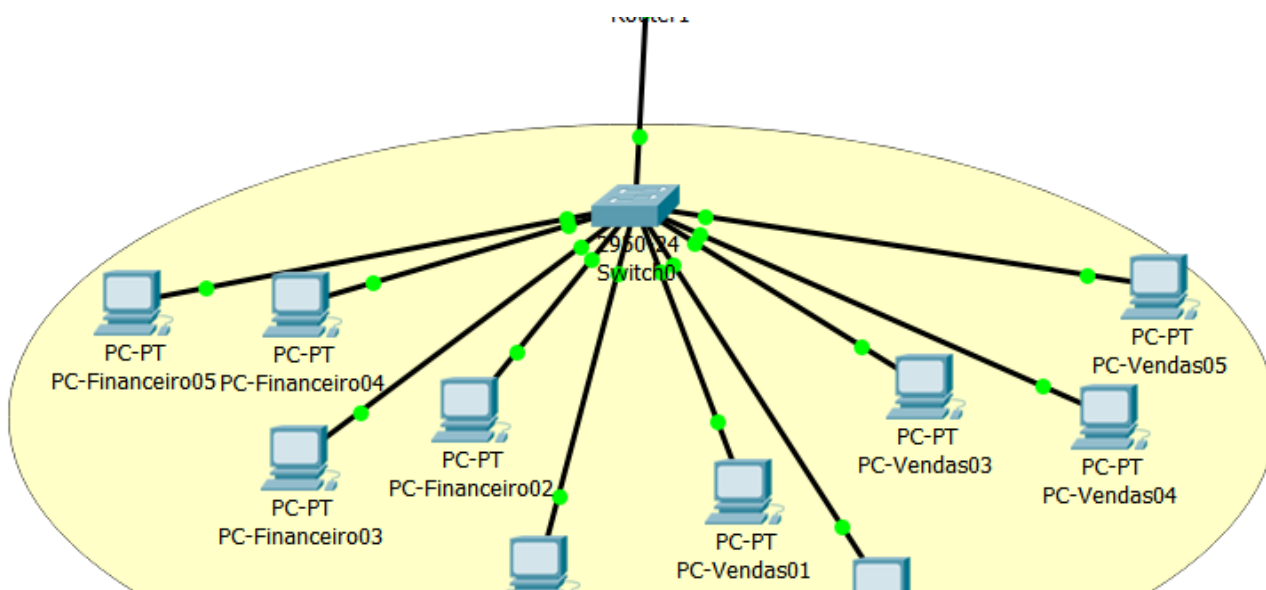


ARTIGOS > DEVOPS

Como configurar VLANs em dispositivos Cisco

**Yuri Matheus**

19/09/2017

COMPARTILHE



Esse artigo faz parte da
Formação DevOps

Estou configurando a rede em um prédio onde funcionará alguns departamentos da Alura: Financeiro e Vendas.

Poderíamos colocar os dois departamentos na mesma rede, porém com isso teríamos alguns problemas:

- Como todos os computadores estariam se comunicando pela mesma via, ao mesmo tempo, a rede poderia ficar mais lenta;
- Um computador conseguiria "ver" o outro. Isso poderia fazer com que um usuário malicioso direcionasse algum tipo de ataque;
- Caso ocorresse algum tipo de problema em alguma parte da rede, ambos departamentos seriam afetados.

Então temos que configurar duas redes diferentes, cada uma relacionada a um departamento.

Em uma primeira abordagem, podemos configurar essas redes fisicamente separadas. Isto é, com um switch para cada departamento, assim como uma interface no roteador. Porém o custo disso seria mais elevado, já que precisaríamos de mais switches e de um roteador com mais portas.

Outra maneira seria separar logicamente o switch e o roteador, pois dessa forma, além de separarmos a rede entre os departamentos, estamos economizando também. Esse tipo de configuração é conhecida como [VLAN](#).

Entendendo as VLANs

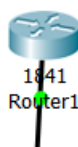
As VLANs, nada mais são do que redes locais (Lan) virtuais. Ou seja, são redes lógicas configuradas em um mesmo [ativo de rede](#) (roteadores e switches, por exemplo). Com ela é possível dividir nossa rede em quantas partes precisarmos para o projeto.

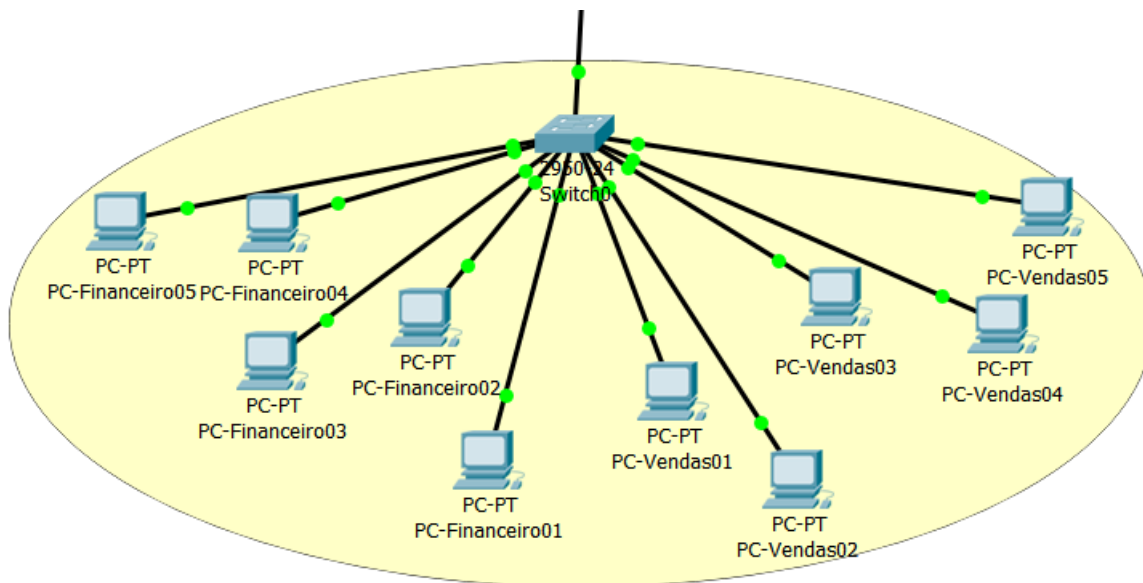
Além da divisão e economia da rede, temos as seguintes vantagens ao implantar esse tipo de rede:

- Mais desempenho, já que o [domínio de broadcast](#) fica dividido entre as redes;
- Mais segurança, apenas os computadores pertencentes a VLAN podem "ver" um ao outro;
- Conseguimos isolar melhor os problemas, como as redes estão separadas, o problema de uma não afeta a outra.

Bacana, agora que sabemos o que é uma VLAN e como ela pode nos ajudar, como podemos implementá-la na nossa rede?

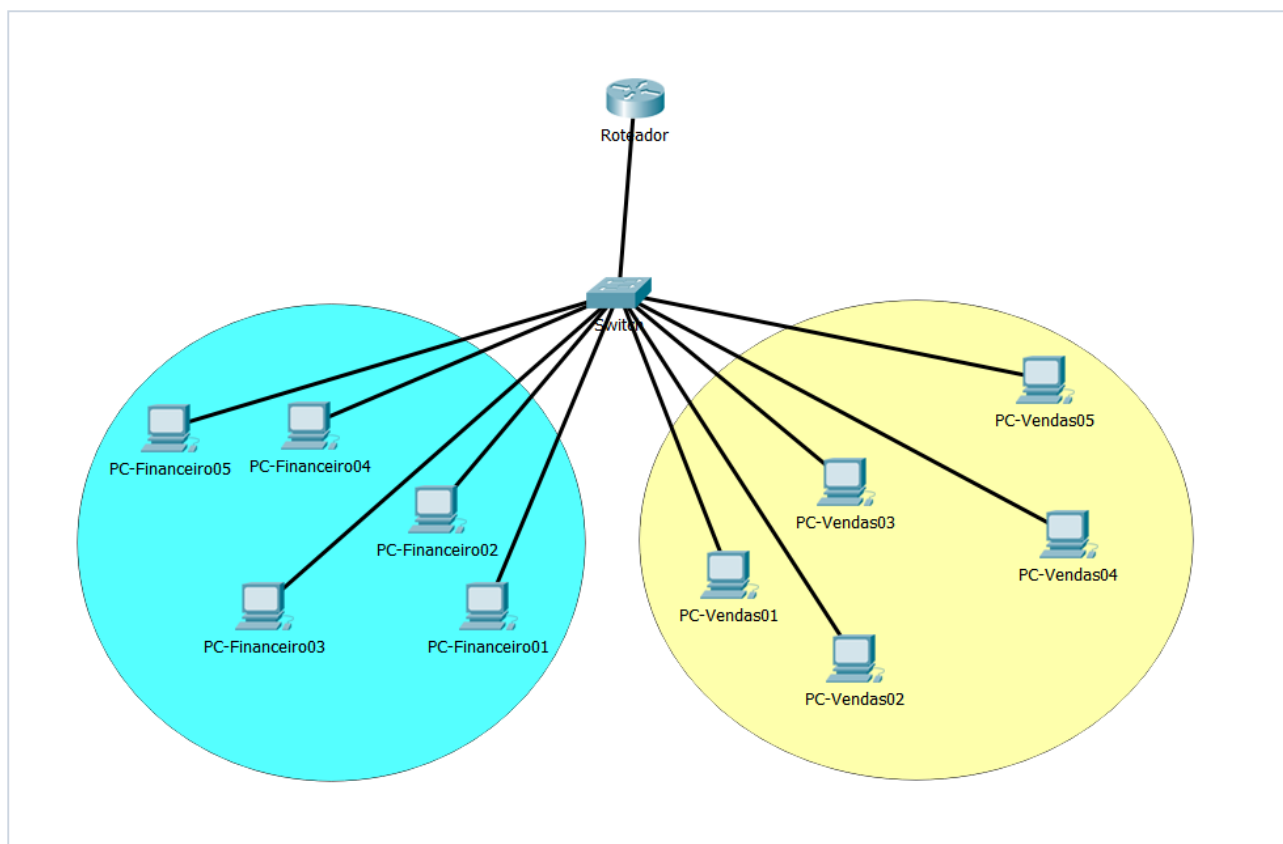
Para isso vamos dar uma olhada na nossa rede e ver como ela está distribuída:





Como vimos, nossa rede tem dez máquinas, cinco do departamento Financeiro e cinco de Vendas. Temos também um switch que concentra nossa rede em um único ponto, seguido de um roteador.

Nós podíamos dividir a rede para que cada departamento faça parte de uma VLAN diferente:



Tanto o nosso switch quanto o nosso roteador são da marca Cisco, em outras palavras, veremos como configurar VLANs em dispositivos Cisco. Mas por onde vamos começar?

Configurando o switch

Bem, como é nosso switch que ligará as máquinas, começaremos por ele.

Esse nosso switch tem 24 portas e precisamos decidir a qual rede elas vão pertencer. Como são dez máquinas, podemos usar as cinco primeiras portas para uma rede e as cinco seguinte para outra. Deixando as outras livres no caso do números de computadores aumentar.

//

Obs: Uma boa prática é usar a última porta do switch para se conectar ao roteador, e deixar uma margem entre uma porta e outra para o caso da rede crescer. Por exemplo, deixar das portas 1 à 10 para uma Vlan e da 11 à 20 para outra.

Também temos que conectar o switch ao roteador. Podemos usar uma das portas que sobraram, a onze, por exemplo.

O primeiro passo para configurar o switch é acessar o modo de usuário privilegiado. Para isso temos que digitar, no terminal, o comando: `enable`.

```
Switch>enable
Switch#
```

Viram que o símbolo `>` mudou para `#`? Isso indica que estamos como usuário privilegiado.

Precisamos falar agora que ele será configurado (`configure`) via terminal (`terminal`). Portanto:

```
# configure terminal
```

```
Switch>enable
Switch#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#
```

Notaram que antes da hashtag apareceu escrito (`config`)? Os switches e roteadores Cisco nos mostram em que modo estão. Neste caso no modo de configuração.

Agora podemos configurar nossas VLANs, mas como identificá-las?

Para isto, cada VLAN deve possuir um número de identificação única. Esse número pode estar entre [2 à 1005](#). Já que nossos dois setores estão no térreo, vamos deixar os ids das VLANs terminados em 0, assim saberemos o andar que elas estão. Assim nossa primeira VLAN será `10`, seguido do `0`, ficando `100`.

Vamos dizer ao switch que queremos criar uma `vlan` que tem o id igual a `10`:

```
(config) # vlan 10 .
```

```
Switch(config)#vlan 10  
Switch(config-vlan)#
```

Perceba que ao criar a VLAN 10, o modo do switch mudou para o `config-vlan`. Ou seja, toda a configuração que fizermos servirá apenas para essa VLAN. Sendo assim, para criarmos a outra VLAN, precisamos sair do modo de configuração dessa que criamos. Para isso basta digitar: `exit` e já saímos desse modo.

```
Switch(config-vlan)#exit  
Switch(config)#
```

Assim podemos criar nossa outra VLAN. O comando é similar, basta substituir o número de identificação:

```
(config) # vlan 20 .
```

```
Switch(config)#vlan 20  
Switch(config-vlan)#
```

Maravilha! Criamos nossas duas VLANs! Mas como podemos ter certeza que elas foram mesmo criadas?

Podemos ver suas configurações com o comando `show vlan` no modo de usuário privilegiado. Para isso ou digitamos `exit` até voltarmos a esse ponto, ou então, utilizamos o atalho **Ctrl+C**, que retorna automaticamente a esse modo.

Feito isso, conseguimos listar nossas VLANs:

```
# show vlan
```

```
Switch#show vlan
```

VLAN Name	Status	Ports
1 default	active	Fa0/1, Fa0/2, Fa0/3, Fa0/4 Fa0/5, Fa0/6, Fa0/7, Fa0/8 Fa0/9, Fa0/10, Fa0/11, Fa0/12 Fa0/13, Fa0/14, Fa0/15, Fa0/16 Fa0/17, Fa0/18, Fa0/19, Fa0/20 Fa0/21, Fa0/22, Fa0/23, Fa0/24
10 VLAN0010	active	
20 VLAN0020	active	
1002 fddi-default	act/unsup	
1003 token-ring-default	act/unsup	
1004 fddinet-default	act/unsup	
1005 trnet-default	act/unsup	

VLAN	Type	SAID	MTU	Parent	RingNo	BridgeNo	Stp	BrdgMode	Trans1	Trans2
1	enet	100001	1500	-	-	-	-	-	0	0
10	enet	100010	1500	-	-	-	-	-	0	0
20	enet	100020	1500	-	-	-	-	-	0	0
1002	fddi	101002	1500	-	-	-	-	-	0	0

--More-- |

Viram? Nossas VLANs foram criadas com os ids que informamos, mas que nomes são esses que foram atribuídos as VLANs?

Quando não informamos nenhum nome para as VLANs, esses são atribuídos de maneira automática pelo sistema, neste o sistema nomeou VLAN seguido do id da mesma.

Mas esses nomes não possuem nenhum significado para nós, ou seja, seria mais intuitivo se fossem nomeados de acordo com o departamento, como Financeiro ou Vendas... Como podemos alterá-los?

Alterando o nome das VLANs

Para alterar os nomes das VLANs, temos que ir até seu modo de configuração. Como estamos no modo de usuário privilegiado, temos que digitar: `# configure terminal`. Em seguida, digitamos a VLAN que queremos configurar, no caso `vlan 10`:

```
(config) # vlan 10
```

Neste modo, falamos que esta VLAN terá o nome (`name`), neste caso, `Financeiro`:

```
Switch(config-vlan)#name Financeiro
```

```
Switch(config-vlan)#name Financeiro
Switch(config-vlan)#
```

Para renomear a outra VLAN, podemos fazer da mesma forma como fizemos anteriormente, a diferença é que entraremos no modo de configuração da vlan de id 20:

```
Switch(config-vlan)#exit
Switch(config)#vlan 20
Switch(config-vlan)#name Vendas
Switch(config-vlan)#
```

Vamos listar as VLANs novamente e ver se a alteração funcionou:

```
Switch#show vlan
```

VLAN	Name	Status	Ports
1	default	active	Fa0/1. Fa0/2. Fa0/3. Fa0/4

			Fa0/5, Fa0/6, Fa0/7, Fa0/8
			Fa0/9, Fa0/10, Fa0/11, Fa0/12
			Fa0/13, Fa0/14, Fa0/15, Fa0/16
			Fa0/17, Fa0/18, Fa0/19, Fa0/20
			Fa0/21, Fa0/22, Fa0/23, Fa0/24
10	Financeiro	active	
20	Vendas	active	
1002	fdi-default	act/unsup	
1003	token-ring-default	act/unsup	
1004	fdinet-default	act/unsup	
1005	trnet-default	act/unsup	

Nossas VLANs já estão criadas e com um nome que nos mostra o lugar a que elas pertencem.

Mas e essas outras VLANs que estão aparecendo, para que são?

Essas são outras VLANs padrões usadas pelos roteadores e switches Cisco, sendo a VLAN 1 a padrão. Como podemos ver, todas as portas do switch pertencem a ela.

Mas para nossa configuração funcionar, as porta precisam estar pertencendo as nossas VLANs, logo precisamos atribuir algumas dessas portas a nossa VLAN... Como vamos fazer isso?

Atribuindo as VLANs às portas do switch

Para isso, no modo de configuração, temos que acessar a porta que queremos configurar. Ou seja, vamos ter que acessar a interface de rede e atribuir uma VLAN a mesma.

Como queremos acessar uma interface do switch, digitamos `interface` seguido do seu tipo ([Ethernet](#), Fast Ethernet), no nosso caso é uma `fastEthernet` e a porta que queremos acessar, `0/1`.

```
Switch(config)#interface fastEthernet 0/1
Switch(config-if)#
```

Precisamos falar que essa porta está conectada a um computador. Isto é, que essa porta do switch (`switchport`) ficará no modo (`mode`) de acesso (`access`) a um dispositivo final (computador, notebook, servidor...).

```
(config-if) # switchport mode access
```

E também que essa porta (`switchport`) estará acessando a VLAN `10`

```
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 10
Switch(config-if)#
```

Com isso a porta 1 já está acessando a VLAN 10, podemos configurar as outras da mesma maneira. Mas você pode se perguntar:

//

"Temos que configurar porta por porta?"

Além do modo manual, conseguimos informar para o switch uma variação. Ou seja, um range de portas que terão a mesma configuração.

Basta dizer que as interfaces dentro da variação (range), do tipo fastEthernet de 0/2 até (-) a porta fastEthernet 0/5 :

```
(config) # interface range fastEthernet 0/2 - fastEthernet 0/5
```

Agora informamos que essas portas estão conectadas a um dispositivo final e que acessarão a VLAN 10:

```
Switch(config)#interface range fastEthernet 0/2 - fastEthernet 0/5
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 10
Switch(config-if-range)#
```

Temos que fazer o mesmo com as portas que pertencerão a outra VLAN:

```
Switch(config)#interface range fastEthernet 0/6 - fastEthernet 0/10
Switch(config-if-range)#switchport mode access
Switch(config-if-range)#switchport access vlan 20
Switch(config-if-range)#
```

Atribuímos nossas portas as suas respectivas VLANs. Podemos conferir usando aquele comando já conhecido: # show vlan

```
Switch#show vlan
```

VLAN Name	Status	Ports
1 default	active	Fa0/11, Fa0/12, Fa0/13, Fa0/14 Fa0/15, Fa0/16, Fa0/17, Fa0/18 Fa0/19, Fa0/20, Fa0/21, Fa0/22 Fa0/23, Fa0/24
10 Financeiro	active	Fa0/1, Fa0/2, Fa0/3, Fa0/4 Fa0/5
20 Vendas	active	Fa0/6, Fa0/7, Fa0/8, Fa0/9 Fa0/10
1002 fddi-default	act/unsup	
1003 token-ring-default	act/unsup	
1004 fddinet-default	act/unsup	
1005 trnet-default	act/unsup	

A configuração do switch foi finalizada. Agora precisamos configurar o roteador para

A configuração do switch foi finalizada. Agora precisamos configurar o roteador para permitir a comunicação da nossa rede.

Configurando o Roteador

Assim como no switch, temos que acessar o modo de usuário privilegiado (`enable`), para começar nossa configuração. Em seguida digitamos `configure terminal` para dizer que ele será configurado:

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
```

Os roteadores Cisco vem com suas portas desabilitadas. Ou seja, para utilizá-las precisamos antes habilitá-las. Para isso, temos que acessar essa porta. Podemos fazer isso da mesma maneira que acessamos a porta do switch.

Dizendo que desejamos acessar a `interface` do tipo `fastEthernet` , cada roteador tem uma forma de nomear suas porta, no nosso caso essa porta é a `0/0` , mas poderia ser outro valor, como `2/1` , por exemplo.

```
(config) # interface fastEthernet 0/0
```

Já que queremos ligar essa interface, temos que falar para ela não (`no`) ficar desligada (`shutdown`). Com isso nossa porta já está habilitada.

```
(config-if) # no shutdown
```

```
Router(config)#interface fastEthernet 0/0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
```

Para que exista a comunicação entre os computadores da nossa rede, precisamos informar um [endereço IP](#). Um para cada rede.

//

"Mas o switch só se conecta a uma porta no roteador, como vamos configurar dois endereços diferentes em um mesmo lugar?"

Dividindo a interface do roteador

Da mesma forma que conseguimos dividir o switch, conseguimos dividir a interface do roteador em partes menores, criando subinterfaces que responderão a redes distintas.

Para acessar essa subinterface, temos que informar a interface que queremos acessar: (config) # interface fastEthernet 0/0 . E, para informar a subinterface, digitamos . (ponto) e o seu número, 10 por exemplo.

```
(config) # interface fastEthernet 0/0.10
```

```
Router(config)#interface fastEthernet 0/0.10
Router(config-subif)#
%LINK-5-CHANGED: Interface FastEthernet0/0.10, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0.10, changed state to up
```

Precisamos informar que a subinterface (0/0.10, neste caso) está associada a uma VLAN. Logo, precisamos dizer que ela está encapsulada (encapsulation) em uma VLAN (dot1Q) seguido do id da VLAN.

```
(config-subif) # encapsulation dot1Q 10
```

```
Router(config-subif)#encapsulation dot1Q 10
Router(config-subif)#|
```

O próximo passo é informar o endereço IP dessa subinterface. Podemos informar qualquer endereço [IP privado](#), no nosso caso, vamos utilizar **192.168.0.0/24** para uma rede e **192.168.1.0/24** para outra.

Como o que queremos informar é um endereço IP basta digitarmos ip address seguido do número e da máscara:

```
(config-subif) # ip address 192.168.0.1 255.255.255.0
```

```
Router(config-subif)#ip address 192.168.0.1 255.255.255.0
Router(config-subif)#|
```

Vamos fazer o mesmo para a outra subinterface. Vamos acessá-la: (config) # interface fastEthernet 0/0.20

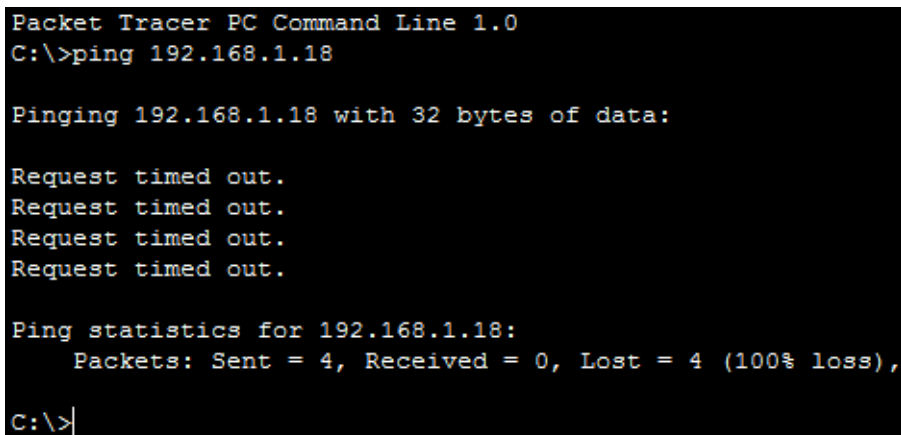
Dizer que ela está associada a VLAN 20: (config-subif) # encapsulation dot1Q 10

E terá como endereço IP: (config-subif) # ip address 192.168.1.1 255.255.255.0 :

```
Router(config)#interface fastEthernet 0/0.20
```

```
Router(config-subif)#  
%LINK-5-CHANGED: Interface FastEthernet0/0.20, changed state to up  
  
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0.20, changed state to up  
  
Router(config-subif)#encapsulation dot1Q 20  
Router(config-subif)#ip address 192.168.1.1 255.255.255.0  
Router(config-subif)#
```

Endereços atribuídos. Como podemos ver se nossa configuração está funcionando? Testando! Vamos abrir o terminal em um computador de um setor e tentar pingar a máquina de um outro setor:



```
Packet Tracer PC Command Line 1.0  
C:\>ping 192.168.1.18  
  
Pinging 192.168.1.18 with 32 bytes of data:  
  
Request timed out.  
Request timed out.  
Request timed out.  
Request timed out.  
  
Ping statistics for 192.168.1.18:  
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),  
C:\>
```

Hum... por quê não funcionou?

Lembra do nosso switch? Nós atribuímos a cada porta uma rede distinta, certo? Logo as interfaces só se comunicam com a rede da qual fazem parte. Ou seja, a porta que conecta o switch ao roteador precisa transmitir os dados das duas VLANs para ocorrer a comunicação. Como podemos fazer isso?

Configurando uma porta trunk no switch

Uma interface que transmite os dados de todas as VLANs é chamada de trunk. Para conseguirmos configurá-la temos que acessar o modo de configuração do switch.

```
Switch>enable  
Switch#configure terminal  
Enter configuration commands, one per line.  End with CNTL/Z.  
Switch(config)#
```

Precisamos acessar a interface que conecta ao roteador. No nosso caso essa é a interface 0/11

```
(config) # interface fastEthernet 0/11
```

```
Switch(config)#interface fastEthernet 0/11
Switch(config-if)#
```

Agora vamos falar que essa porta do switch (`switchport`) não está no modo (`mode`) de conexão a um dispositivo final, e sim em um modo em que junta todas as VLANs (`trunk`).

```
(config-if) # switchport mode trunk
```

```
Switch(config-if)#switchport mode trunk

Switch(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/11, changed state to down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/11, changed state to up
```

Vamos testar novamente e ver se nossas duas máquinas conseguem conversar entre si:

```
C:\>ping 192.168.1.18

Pinging 192.168.1.18 with 32 bytes of data:

Reply from 192.168.1.18: bytes=32 time<1ms TTL=127
Reply from 192.168.1.18: bytes=32 time=1ms TTL=127
Reply from 192.168.1.18: bytes=32 time<1ms TTL=127
Reply from 192.168.1.18: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.1.18:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>
```

Sucesso! Agora temos duas redes separadas logicamente que conseguem se comunicar.

Para saber mais

Veja que neste caso foi interessante juntar todas as VLANs na porta trunk, porém existem casos em que nós precisamos isolar essas VLANs escolhendo quais passarão pela porta trunk. Apesar de esse não ser a proposta desse post, você pode conferir [aqui](#) essa configuração..

Conclusão

Vimos nesse post o problema que tínhamos ao conectar duas redes distintas e como as VLANs seriam uma solução. Vimos como configurá-las nos switches e nos roteadores fazendo existir comunicação entre elas.

Gostou de redes? Essa é só uma pequena parte de muitas outras. Aqui na **Alura** temos uma [formação em redes de computadores](#) onde você aprenderá sobre VLANs, protocolos, como configurar listas de acesso, segurança e muito mais!



Yuri Matheus

Yuri é desenvolvedor e instrutor. É estudante de Sistemas de Informação na FIAP e formado como Técnico em Informática no Senac SP. O seu foco é nas plataformas Java e Python e em outras áreas como Arquitetura de Software e Machine Learning. Yuri também atua como editor de conteúdo no blog da Alura, onde escreve, principalmente, sobre Redes, Docker, Linux, Java e Python.

Leia também:

- [Como calcular máscaras de sub-rede de forma simples](#)
- [Configurando DHCP em roteadores Cisco](#)
- [O que enfraquece o sinal do Wi-Fi](#)
- [Entendendo o IPv6](#)
- [Criando listas de acesso nos roteadores Cisco](#)

Veja outros artigos sobre
[DevOps](#)

Quer mergulhar em tecnologia e aprendizagem?

Receba a newsletter que o nosso CEO escreve pessoalmente, com insights do mercado de trabalho, ciência e desenvolvimento de software

Escreva seu email

ME INSCREVA

NAVEGAÇÃO

PLANOS
TODOS OS CURSOS
ALURA CASES
GUIA DE CARREIRA
INSTRUTORES
TRABALHE CONOSCO
COMO VIRAR INSTRUTOR
ARTIGOS
PARA ESCOLAS
PODCASTS
POLÍTICA DE PRIVACIDADE
TERMOS DE USO
SOBRE NÓS
COMO FUNCIONA
DEV EM <T>
PERGUNTAS FREQUENTES
STATUS

FALE COM A GENTE

WHATSAPP
EMAIL E TELEFONE

BLOG

PROGRAMAÇÃO
FRONT-END
DATA SCIENCE
DEVOPS
UX & DESIGN
MOBILE
INOVAÇÃO & GESTÃO

AOVS Sistemas de Informática S.A
CNPJ 05.555.382/0001-33

NOSSAS REDES E APPS



ALURA NA SUA EMPRESA

Desenvolva seu time com apoio da nossa solução

CASES

COMUNIDADE

GUIA DE CARREIRA

NA MÍDIA

CONHEÇA MAIS

COMUNIDADE

SCUBA DEV

IMERSÕES

DEPOIMENTOS

NOSSOS INSTRUTORES

PARCEIROS



Nós, da Alura, somos uma das Scale-Ups selecionadas pela Endeavor, programa de aceleração das empresas que mais crescem no país.



Fomos uma das 7 startups selecionadas pelo Google For Startups a participar do programa Growth Academy em 2021.

CURSOS

Cursos de Programação

Lógica | Python | PHP | Java | .NET | Node JS | C | Computação | Jogos | IoT

Cursos de Front-end

HTML, CSS | React | Angular | JavaScript | jQuery

Cursos de Data Science

Ciência de dados | BI | SQL e Banco de Dados | Excel | Machine Learning | NoSQL | Estatística

Cursos de DevOps

AWS | Azure | Docker | Segurança | IaC | Linux

Cursos de UX & Design

Photoshop e Illustrator | Usabilidade e UX | Vídeo e Motion | 3D

Cursos de Mobile

React Native | Flutter | iOS e Swift | Android, Kotlin | Jogos

Cursos de Inovação & Gestão

Métodos Ágeis | Softskills | Liderança e Gestão | Startups | Vendas