Table of Contents

CCIE Routing & Switching

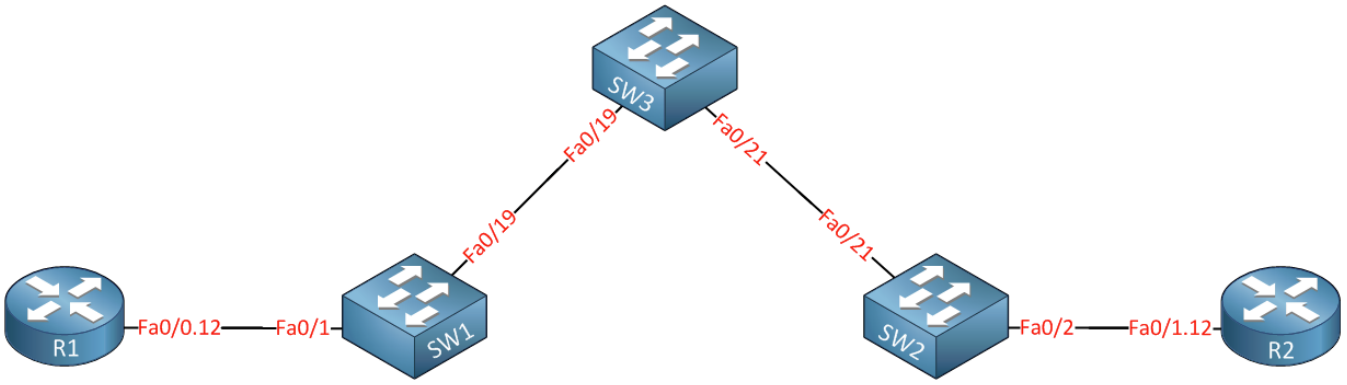
802.1Q Tunneling (Q-in-Q) Configuration Example


12 votes

802.1Q tunneling (aka Q-in-Q) is a technique often used by Metro Ethernet providers as a layer 2 VPN for customers. 802.1Q (or dot1q) tunneling is pretty simple...the provider will put a 802.1Q tag on all the frames that it receives from a customer with a unique VLAN tag. By using a different VLAN tag for each customer we can separate the traffic from different customers and also transparently transfer it throughout the service provider network.

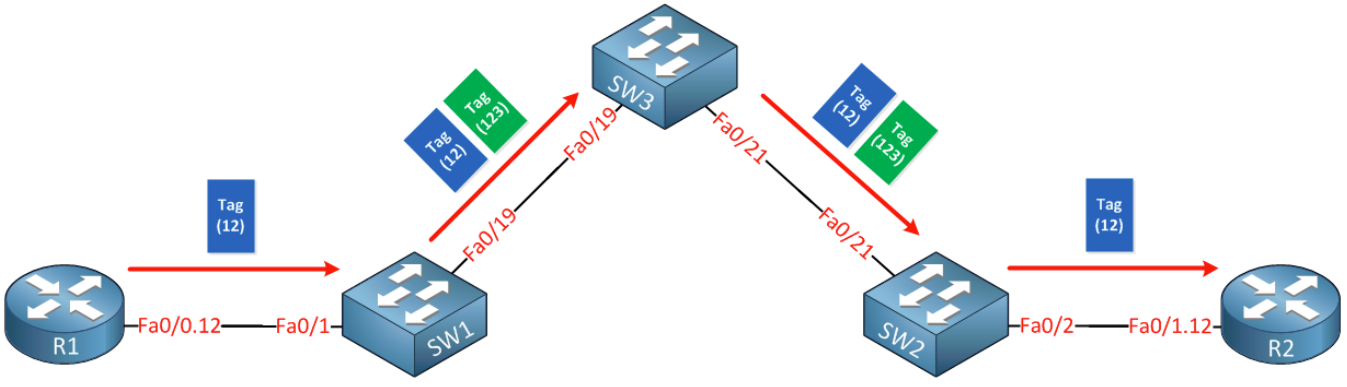
One of the advantages of this solution is that it's easy to implement, you don't need exotic hardware and we don't have to run any routing protocols between the service provider and customer (unlike MPLS VPN). From the customer's perspective, it's just like their sites are directly connected on layer 2.

In this tutorial I'm going to show you how to configure 802.1Q tunneling and I'll explain how it works. I'll be using the following topology for this:



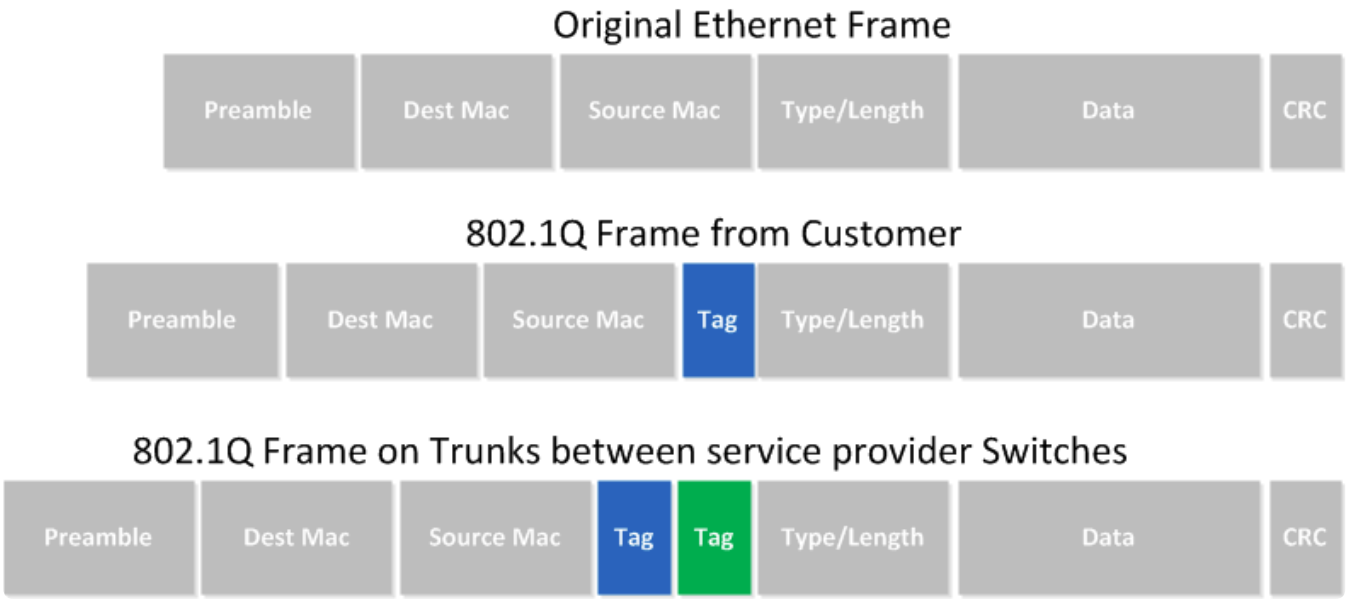
Above you see two routers called R1 and R2, imagine these routers are the customer sites that we want to connect through the service provider network which consists of SW1, SW2 and SW3. Our customer wants to use VLAN 12 between the two sites and expects our service provider to transport this from one site to another.

In my example our customer will be using VLAN 12 for traffic between their sites. The service provider has decided to use VLAN 123 to transport everything for this customer. Basically this is what will happen when we send frames between R1 and R2:

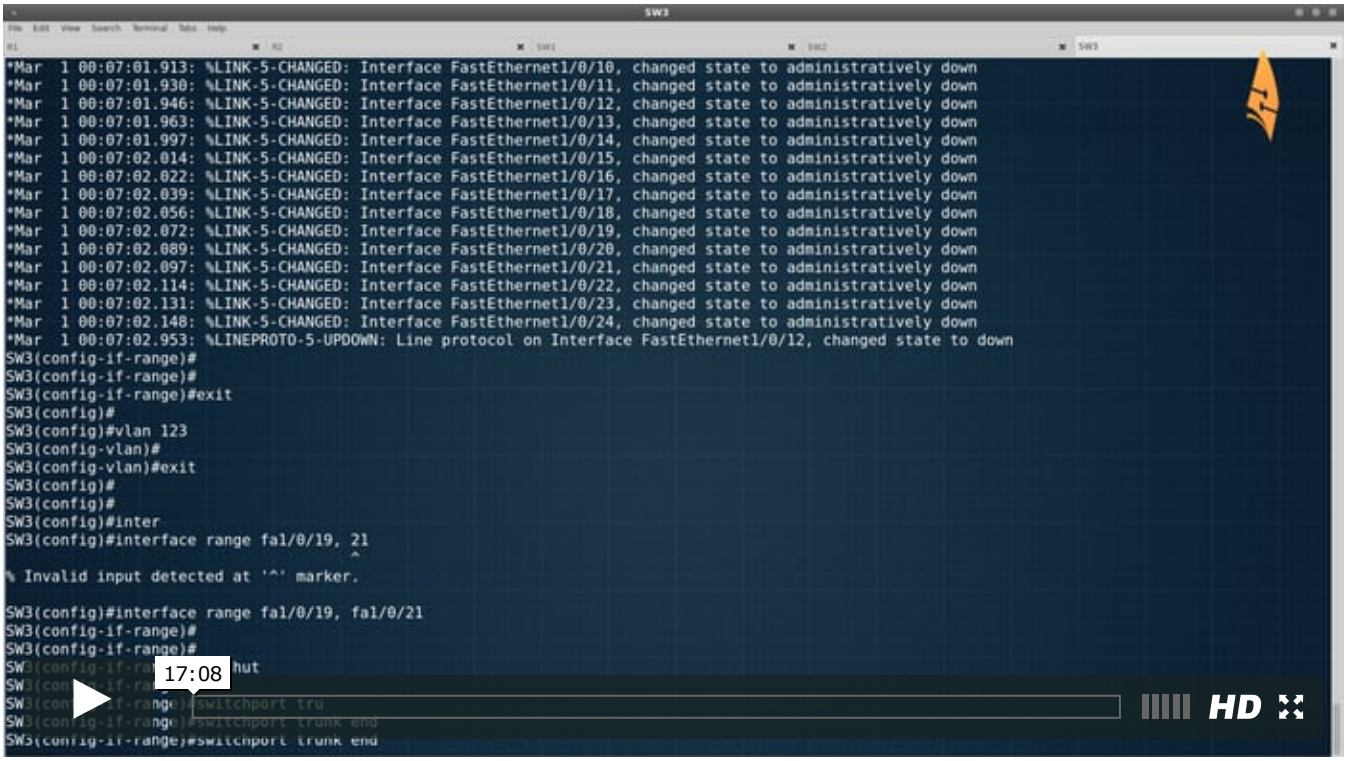


Whenever R1 sends trafffic it will tag its frames for VLAN 12. Once it arrives at the service provider, SW1 will add an additional VLAN tag (123). Once SW2 forwards the frame towards R2 it will remove the second VLAN tag and forwards the original tagged frame from R1.

Here is another way to visualize this:



Enough talk…let's take a look at the configuration.



Here's what the router configs look like:

```
R1(config)#interface fastEthernet 0/0
R1(config-if)#no shutdown
R1(config-if)#interface fastEthernet 0/0.12
R1(config-subif)#encapsulation dot1Q 12
```

```
R1(config-subif)#ip address 192.168.12.1 255.255.255.0
```

```
R2(config)#interface fastEthernet 0/0
R2(config-if)#no shutdown
R2(config-if)#interface fastEthernet 0/0.12
R2(config-subif)#encapsulation dot1Q 12
R2(config-subif)#ip address 192.168.12.2 255.255.255.0
```

R1 and R2 are both configured with sub-interfaces and use subnet 192.168.12.0 /24. All their frames are tagged as VLAN 12.

On the service provider network we'll have to configure a number of items. First I will configure 802.1Q trunks between SW1 – SW3 and SW2 – SW3:

```
SW1(config)#interface fastEthernet 0/19
SW1(config-if)#switchport trunk encapsulation dot1q
SW1(config-if)#switchport mode trunk
```

```
SW2(config)#interface fastEthernet 0/21
SW2(config-if)#switchport trunk encapsulation dot1q
SW2(config-if)#switchport mode trunk
```

```
SW3(config)#interface fastEthernet 0/19
SW3(config-if)#switchport trunk encapsulation dot1q
SW3(config-if)#switchport mode trunk

SW3(config)#interface fastEthernet 0/21
SW3(config-if)#switchport trunk encapsulation dot1q
SW3(config-if)#switchport mode trunk
```

The next part is where we configure the actual "Q-in-Q" tunneling. The service provider will use VLAN 123 to transfer everything from our customer. We'll configure the interfaces towards the customer routers to tag everything for VLAN 123:

```
SW1(config)#interface fastEthernet 0/1
SW1(config-if)#switchport access vlan 123
```

```
SW1(config-if)#switchport mode dot1q-tunnel
```

```
SW2(config)#interface fastEthernet 0/2
SW2(config-if)#switchport access vlan 123
SW2(config-if)#switchport mode dot1q-tunnel
```

The **switchport mode dot1q-tunnel** command tells the switch to tag the traffic and **switchport access vlan** command is required to specify the Q-in-Q VLAN of 123. Make sure that VLAN 123 is available on SW1, SW2 and SW3. By assigning the interfaces above to this VLAN it was automatically created on SW1 and SW2 but I also have to make sure that SW3 has VLAN 123 in its database:

```
SW3(config)#vlan 123
```

Everything is now in place, let's do a quick test to see if R1 and R2 can reach each other:

```
R1#ping 192.168.12.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

Great! Our ping is working! Let's take a look at some commands to verify our work:

```
SW1#show dot1q-tunnel

dot1q-tunnel mode LAN Port(s)
-----------------------------
Fa0/1
```

```
SW2#show dot1q-tunnel

dot1q-tunnel mode LAN Port(s)
-----------------------------
Fa0/2
```

The show **dot1q-tunnel** command doesn't give me a lot of information. The only thing we see are the interfaces that are configured for dot1q tunneling. A good way to prove that the service provider switches are really tunneling the frames from the customer is by looking at the trunks between SW1, SW2 and SW3:

```
SW1#show interfaces fa0/19 trunk

Port        Mode             Encapsulation  Status        Native vlan
Fa0/19      on               802.1q         trunking      1

Port        Vlans allowed on trunk
Fa0/19      1-4094

Port        Vlans allowed and active in management domain
Fa0/19      1,123

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/19      1,123
```

```
SW2#show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Fa0/21      on               802.1q         trunking      1

Port        Vlans allowed on trunk
Fa0/21      1-4094

Port        Vlans allowed and active in management domain
Fa0/21      1,123

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/21      1,123
```

```
SW3#show interfaces trunk

Port        Mode             Encapsulation  Status        Native vlan
Fa1/0/19    on               802.1q         trunking      1
Fa1/0/21    auto             n-802.1q       trunking      1
```

```
Port        Vlans allowed on trunk
Fa1/0/19    1-4094
Fa1/0/21    1-4094

Port        Vlans allowed and active in management domain
Fa1/0/19    1,123
Fa1/0/21    1,123

Port        Vlans in spanning tree forwarding state and not pruned
Fa1/0/19    1,123
Fa1/0/21    1,123
```

As you can see above the only VLAN that is active (besides VLAN 1) on these trunk links is VLAN 123. You won't see VLAN 12 here because that's the customer traffic and it's encapsulated with VLAN 123. Another good way to prove this is by looking at spanning-tree:

```
SW1#show spanning-tree vlan 12

Spanning tree instance(s) for vlan 12 does not exist.
```

```
SW2#show spanning-tree vlan 12

Spanning tree instance(s) for vlan 12 does not exist.
```

```
SW3#show spanning-tree vlan 12

Spanning tree instance(s) for vlan 12 does not exist.
```

Our switches don't have a spanning-tree topology for VLAN 12, they don't care what VLAN the customer is using...they only care about VLAN 123.

So far so good! 802.1Q tunneling has some more tricks up its sleeve, one of the things it can do is tunnel some layer 2 protocols. Take a look below:

```
SW1(config)interface fastEthernet 0/1
SW1(config-if)#l2protocol-tunnel ?
```

```
  cdp                 Cisco Discovery Protocol
  drop-threshold      Set drop threshold for protocol packets
  point-to-point      point-to-point L2 Protocol
  shutdown-threshold  Set shutdown threshold for protocol packets
  stp                 Spanning Tree Protocol
  vtp                 Vlan Trunking Protocol
  <cr>
```

If you want it can tunnel CDP, VTP, STP and even point-to-point protocols like PAgP or LACP (Etherchannel). Let me show you what happens when you tunnel CDP traffic:

```
SW1(config)#interface fastEthernet 0/1
SW1(config-if)#l2protocol-tunnel cdp
```

```
SW2(config)#interface fastEthernet 0/2
SW2(config-if)#l2protocol-tunnel cdp
```

I'll tell SW1 and Sw2 to tunnel all CDP traffic between the interfaces that are connected to R1 and R2. Take a look below for the result:

```
R1#show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
R2               Fas 0/0           171          R S I     2811      Fas 0/1
```

By tunneling CDP packets between R1 and R2 they see each other as directly connected. That's a great example of a "transparent" network.

The last thing we have to discuss are MTU (Maximum Transmission Unit) problems.

The ethernet frame of the customer with the 802.1Q tag will have a MTU of 1500 bytes but since we are adding another 802.1Q tag the total MTU will be 1504 bytes in the service provider network. By default most switches will only allow a maximum MTU of 1500 bytes so you will run into problems with large packets. Below you can see the actual problem:

```
R1#ping 192.168.12.2 size 1500 df-bit

Type escape sequence to abort.
Sending 5, 1500-byte ICMP Echos to 192.168.12.2, timeout is 2 seconds:
Packet sent with the DF bit set
.....
Success rate is 0 percent (0/5)
```

Because of second tag this ping will be dropped because the MTU is too small. To solve this you should increase the maximum MTU size of your switches:

```
SW1(config)#system mtu 1504
```

```
SW2(config)#system mtu 1504
```

```
SW3(config)#system mtu 1504
```

After configuring the MTU you have to reboot your switches. You can see the MTU size like this:

```
SW1#show system mtu

System MTU size is 1504 bytes
System Jumbo MTU size is 1504 bytes
Routing MTU size is 1504 bytes
```

Our ping should now work:

```
R1#ping 192.168.12.2 size 1500 df-bit

Type escape sequence to abort.
Sending 5, 1500-byte ICMP Echos to 192.168.12.2, timeout is 2 seconds:
Packet sent with the DF bit set
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

There we go, problem solved! This is all I have for now on 802.1Q tunneling, I hope this has been helpful to you. If you have any questions feel free to leave a comment!

Want to take a look for yourself? Here you will find the configuration of each device.

R1

```
hostname R1
!
interface FastEthernet0/0.12
 encapsulation dot1Q 12
 ip address 192.168.12.1 255.255.255.0
 !
end
```

R2

```
hostname R2
!
interface FastEthernet0/1.12
 encapsulation dot1Q 12
 ip address 192.168.12.2 255.255.255.0
 !
end
```

SW1

```
hostname SW1
!
vlan 123
!
interface FastEthernet0/1
 switchport access vlan 123
 switchport mode dot1q-tunnel
 no cdp enable
!
interface FastEthernet0/19
```

```
 switchport trunk encapsulation dot1q
 switchport mode trunk
!
end
```

SW2

```
hostname SW2
!
interface FastEthernet0/2
 switchport access vlan 123
 switchport mode dot1q-tunnel
 no cdp enable
!
interface FastEthernet0/21
 switchport trunk encapsulation dot1q
 switchport mode trunk
!
end
```

SW3

```
hostname SW3
!
vlan 123
!
interface FastEthernet1/0/19
 switchport trunk encapsulation dot1q
 switchport mode trunk
!
interface FastEthernet1/0/21
 switchport trunk encapsulation dot1q
 switchport mode trunk
!
end
```

Rate this Lesson:

« Previous Lesson
Cisco DTP (Dynamic Trunking Protocol) Negotiation

Next Lesson
Etherchannel over 802.1Q Tunneling »

Home › Forums › 802.1Q Tunneling (Q-in-Q) Configuration Example

This topic contains 26 replies, has 8 voices, and was last updated by Rene Molenaar 1 month ago.

Viewing 15 posts - 1 through 15 (of 26 total)
1 2 →

- Author
  Posts | Subscribe
- March 13, 2014 at 09:12 #11955 Reply

Richie
Great post mate.

First came across this trying to get a CCIE lab set up. Was very useful for that.

Is this still a tech widely used in service providers ?

March 17, 2014 at 12:45 #11956 Reply

Rene Molenaar
Keymaster
Thanks Richie,

I haven't seen it on a real network before and I doubt that it's still used a lot. A lot of service providers use MPLS which can also transport L2 if you want.

Rene

March 18, 2014 at 10:58 #11957 Reply

Adinor
Great post.

Thanks

April 9, 2014 at 05:26 #11958 Reply

gov r
Member
Great site....to learn new things...

Thanks for sharing your knowledge...

Take care

April 14, 2014 at 11:16 #11959 Reply

Rene Molenaar
Keymaster
You are welcome.

April 23, 2014 at 11:53 #11960 Reply

parastoo
excellent post. tx alot.

May 1, 2014 at 12:11 #11961 Reply

Rene Molenaar
Keymaster
You are welcome.

May 15, 2014 at 10:41 #11962 Reply

Royala
Nice article…. thanks so much

May 22, 2014 at 17:46 #11963 Reply

Ashenafi
Excellent post! I like the simple flow with which you explained it.
Thank you!

May 23, 2014 at 14:05 #11964 Reply

Anonymous
Glad you like it!

June 13, 2014 at 09:21 #11965 Reply

autocrat
I have a question.

What if I want to have WAN ports in between the customer sites. I mean, image SW3 is not there and fa0/19 of SW1 and fa0/21 of SW2 are replaced with WAN ports. How do we implement QinQ? Is that possible and accepted?

June 25, 2014 at 08:10 #11966 Reply

Rene Molenaar
Keymaster
You can't do it with Q-in-Q because it's not available on the switches, you could however use something like any transport over MPLS to get your VLANs from one side to the other.

June 25, 2014 at 12:54 #11967 Reply

autocrat
But I have topology something like this.

Router1 (F1) —— (F3) Router2 (F8) ——– (F4) Router3 (F3) ———- (F1) Router4

In above topology,

F1 from Router1 and Router4 are switchports.
F3 from Router2 and Router3 are switchports.
F8 from Router2 and F4 from Router3 are WAN ports.

Configured 'dot1q-tunnel' at F3 of Router2 and Router3.
Configured 'encapsulation dot1q ' at F8 of Router2 and F4 of Router3. (The only difference from your above post. Instead of trunk between Router2 and Router3, I have WAN with encapsulation dot1q config.)

Also I have bridged the F8 and Vlan35 under F3 of Router2 and also F4 and Vlan35 of F3 of Router3.

Ping is failing from Router1 to Router4. This I expect to work. 🙂

July 10, 2014 at 09:57 #11968 Reply

autocrat
So nobody knows answer for the above question? Even to say it right or wrong!!!

October 8, 2014 at 18:57 #11969 Reply

Matt McClellan
Rene,
Great article, but you might want to edit this little typo as Fa0/20 is not in your example diagrams:

SW3(config)#interface fastEthernet 0/20
SW3(config-if)#switchport trunk encapsulation dot1q
SW3(config-if)#switchport mode trunk

I knew what you meant, though, and again, great article!

- Author
  Posts

Viewing 15 posts - 1 through 15 (of 26 total)
1 2 →
Reply To: 802.1Q Tunneling (Q-in-Q) Configuration Example

b / link b-quote del img ul ol li code close tags

Please put code in between `backticks` or use the CODE button.
To place inline images, please use any image share service (such as TinyPic or Imgur) and use the IMG button!

Notify me of follow-up replies via email

Maximum file size allowed is 2048 KB.

Attachments:
Выберите файл Файл не выбран
Add another file

Submit

About NetworkLessons.com

Hello There! I'm René Molenaar (CCIE #41726), Your Personal Instructor of Networklessons.com. I'd like to teach you everything about Cisco, Wireless and Security. I am here to Help You Master Networking!

Read my story

Social Fans

14,267
FANS

7,929
FOLLOWERS

1,589
SUBSCRIBERS

Highest Rated Lessons

MPLS Layer 3 VPN Configuration
(25 votes)

VRF Lite Configuration on Cisco IOS
(23 votes)

Cisco Portfast Configuration
(20 votes)

IPv6 Address Types
(18 votes)

EIGRP Stub Explained
(17 votes)

New Lessons

Introduction to Cisco IOS XE
ERSPAN Configuration on Cisco IOS XE
IGMP Filter
IGMP Snooping without Router
Cisco Group Management Protocol (CGMP)

802.1Q Tunneling (Q-in-Q) Configuration Example written by Rene Molenaar average rating 5/5 - 12 user ratings