Static Route

```
Router(config)#ip route network-address subnet-mask
{ip-address | exit-intf}
```

Parameter	Description
network-address	Destination network address of the remote network to be added to the routing table.
subnet-mask	• Subnet mask of the remote network to be added to the routing table. • The subnet mask can be modified to summarize a group of networks.
ip-address	• Commonly referred to as the next-hop router's IP address. • Typically used when connecting to a broadcast media (i.e., Ethernet). • Commonly creates a recursive lookup.
exit-intf	• Use the outgoing interface to forward packets to the destination network. • Also referred to as a directly attached static route. • Typically used when connecting in a point-to-point configuration.

Default Route

```
Router(config)#ip route 0.0.0.0 0.0.0.0 {ip-address | exit-intf}
```

Parameter	Description
0.0.0.0	Matches any network address.
0.0.0.0	Matches any subnet mask.
ip-address	• Commonly referred to as the next-hop router's IP address. • Typically used when connecting to a broadcast media (i.e., Ethernet). • Commonly creates a recursive lookup.
exit-intf	• Use the outgoing interface to forward packets to the destination network. • Also referred to as a directly attached static route. • Typically used when connecting in a point-to-point configuration.

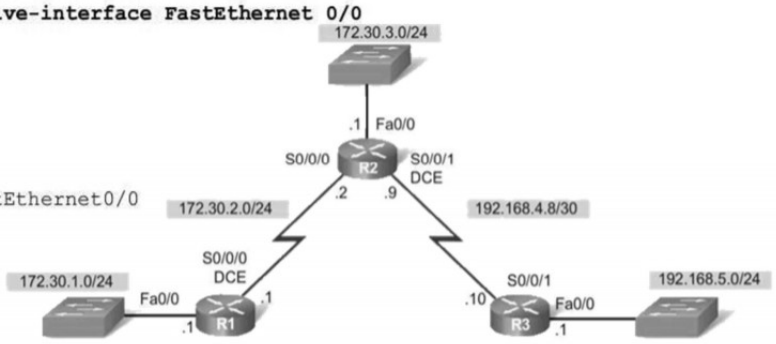
```
show ip route
show ip route static
show ip route network
Show running-config
show ip interface brief
```

RIPv1 (Auto Summarization)

```
R1(config)#router rip
R1(config-router)#network 172.30.1.0
R1(config-router)#network 172.30.2.0
R1(config-router)#passive-interface FastEthernet 0/0
R1(config-router)#end
R1#show run
 (**output omitted**)

!
router rip
 passive-interface FastEthernet0/0
 network 172.30.0.0
!
 (**output omitted**)

R1#
```



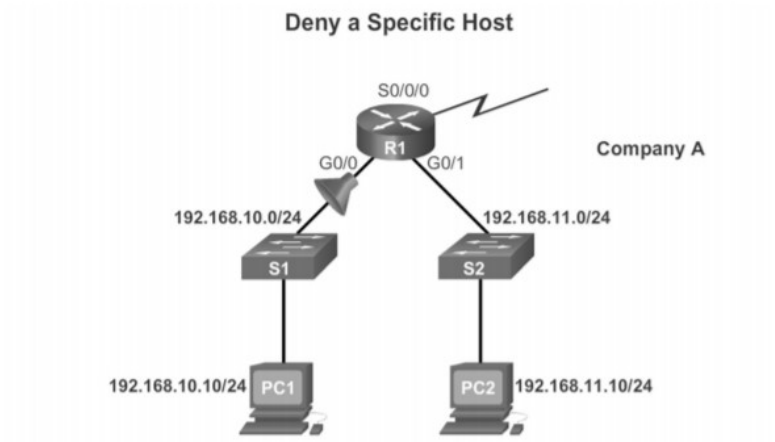
RIP update : `R2#debug ip rip`

RIPv2 (network = สายที่ออกจาก router)

```
router rip
version 2
network 192.168.1.80 {Subnet mask}
network 192.168.1.0   {Subnet mask}
network 192.168.1.32          {Subnet mask}
passive-interface fa0/0.10
passive-interface fa0/0.20
no auto-summary
```

Standard ACL :

```
Router(config)# access-list access-list-number
   deny | permit | remark
   source [ source-wildcard ] [ log ]
```

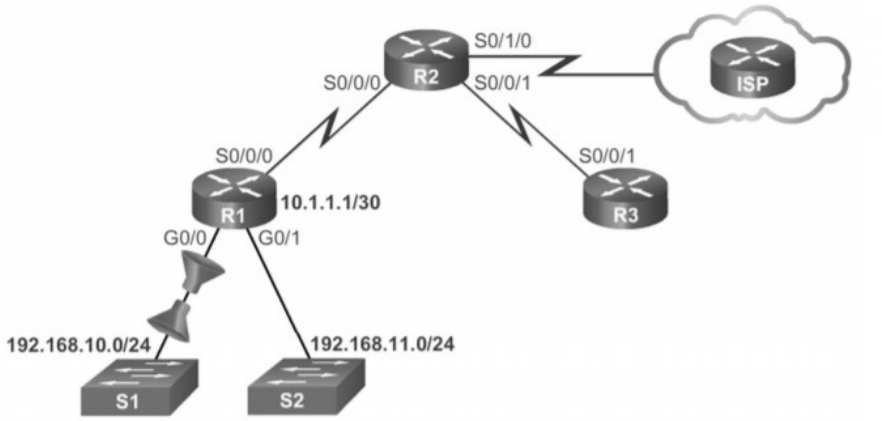



```
R1(config)#no access-list 1
R1(config)#access-list 1 deny host 192.168.10.10
R1(config)#access-list 1 permit any
R1(config)#interface g0/0
R1(config-if)#ip access-group 1 in
```

```
R1(config)#ip access-list standard NO_ACCESS
R1(config-std-nacl)#remark Do not allow access from Lab
workstation
R1(config-std-nacl)#deny host 192.168.11.10
R1(config-std-nacl)#remark Allow access from all other networks
R1(config-std-nacl)#permit any
R1(config-std-nacl)#interface G0/0
R1(config-if)#ip access-group NO_ACCESS out
R1(config-if)#
```

Extended ACL :

```
access-list access-list-number {deny | permit | remark}
protocol source [source-wildcard] [operator operand]
[port port-number or name] destination [destination-wildcard]
[operator operand] [port port-number or name][established]
```

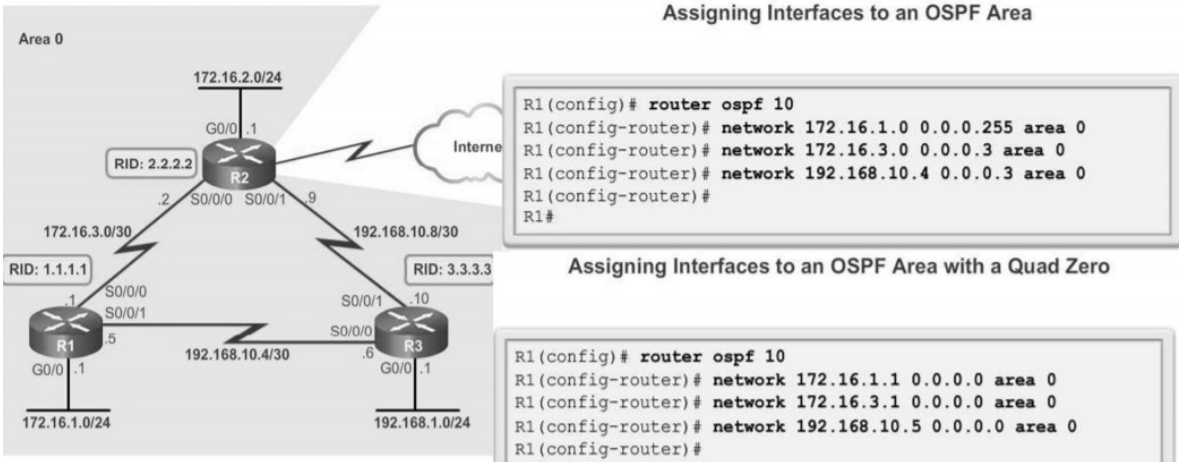



```
R1(config)#access-list 103 permit tcp 192.168.10.0 0.0.0.255 any eq 80
R1(config)#access-list 103 permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1(config)#access-list 104 permit tcp any 192.168.10.0 0.0.0.255 established
R1(config)#interface g0/0
R1(config-if)#ip access-group 103 in
R1(config-if)#ip access-group 104 out
```

```
R1(config)#access-list 101 deny tcp 192.168.11.0 0.0.0.255 192.168.10.0
0.0.0.255 eq ftp
R1(config)#access-list 101 deny tcp 192.168.11.0 0.0.0.255 192.168.10.0
0.0.0.255 eq ftp-data
R1(config)#access-list 101 permit ip any any
R1(config)#interface g0/1
R1(config-if)#ip access-group 101 in
```

```
R1(config)#ip access-list extended SURFING
R1(config-ext-nacl)#permit tcp 192.168.10.0 0.0.0.255 any eq 80
R1(config-ext-nacl)#permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1(config-ext-nacl)#exit
R1(config)#ip access-list extended BROWSING
R1(config-ext-nacl)#permit tcp any 192.168.10.0 0.0.0.255 established
R1(config-ext-nacl)#exit
R1(config)#interface g0/0
R1(config-if)#ip access-group SURFING in
R1(config-if)#ip access-group BROWSING out
```

OSPF :



Cost = 100,000,000 bps/interface bandwidth in bps

Serial 1.544 Mbps	100,000,000	÷	1,544,000	64

ǫ Bandwidth = `R1# show interfaces serial 0/0/0`

DHCP :

```
R1(config)# ip dhcp excluded-address 192.168.10.1 192.168.10.9
R1(config)# ip dhcp excluded-address 192.168.10.254
R1(config)# ip dhcp pool LAN-POOL-1
R1(dhcp-config)# network 192.168.10.0 255.255.255.0
R1(dhcp-config)# default-router 192.168.10.1
R1(dhcp-config)# dns-server 192.168.11.5
R1(dhcp-config)# domain-name example.com
R1(dhcp-config)# end
```

Switch :

Cisco Switch IOS Commands	
Enter global configuration mode.	S1# **configure terminal**
Enter interface configuration mode.	S1(config)# **interface fastethernet 0/1**
Configure the interface duplex.	S1(config-if)# **duplex full**
Configure the interface speed.	S1(config-if)# **speed 100**
Return to the privileged EXEC mode.	S1(config-if)# **end**
Save the running config to the startup config.	S1# **copy running-config startup-config**

Cisco Switch IOS Commands	
Display interface status and configuration.	S1# **show interfaces [interface-id]**
Display current startup configuration.	S1# **show startup-config**
Display current operating config.	S1# **show running-config**
Displays info about flash filesystem.	S1# **show flash**
Displays system hardware & software status.	S1# **show version**
Display history of commands entered.	S1# **show history**
Display IP information about an interface.	S1# **show ip [interface-id]**
Display the MAC address table.	S1# **show mac-address-table**

Sw with SSH

```
S1 # configure terminal
S1(config)# ip domain-name cisco.com
S1(config)# crypto key generate rsa
The name for the keys will be: S1.cisco.com
...
How many bits in the modulus [512]: 1024
...
S1(config)# username admin password ccna
S1(config)# line vty 0 15
S1(config-line)# transport input ssh
S1(config-line)# login local
S1(config)# end
```

- Static secure MAC addresses

```
Switch(config-if)#switchport mode access
Switch(config-if)#switchport port-security
Switch(config-if)#switchport port-security mac-address MAC-ADD
```

- Dynamic secure MAC addresses

```
Switch(config-if)#switchport mode access
Switch(config-if)#switchport port-security
Switch(config-if)#switchport port-security mac-address sticky
```

- Maximum MAC addresses

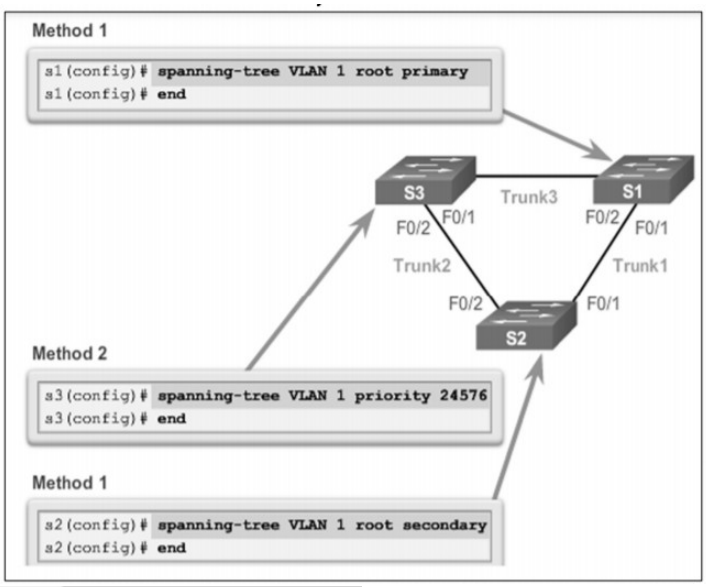
```
Switch(config-if)#switchport port-security maximum MAX
```

- Violation mode

```
Switch(config-if)#switchport port-security violation ?
  protect   Security violation protect mode
  restrict  Security violation restrict mode
  shutdown  Security violation shutdown mode
```

Cisco IOS CLI Commands	
S1(config)#**interface fastethernet 0/18**	Specify the interface to be configured for port security.
S1(config-if)#**switchport mode access**	Set the interface mode to access.
S1(config-if)#**switchport port-security**	Enable port security on the interface.
S1(config-if)#**switchport port-security maximum 50**	Set the maximum number of secure addresses allowed on the port.
S1(config-if)#**switchport port-security mac-address sticky**	Enable sticky learning.

Spanning Tree




```
S3# show spanning-tree
```

```
S1# configure terminal
S1(config)# spanning-tree mode rapid-pvst
S1(config)# interface f0/2
S1(config-if)# spanning-tree link-type point-to-point
S1(config-if)# end
S1# clear spanning-tree detected-protocols
```

```
S3(config)# spanning-tree vlan 20 root primary
```

This command forces S3 to be the primary root for VLAN 20.

```
S3(config)# spanning-tree vlan 10 root secondary
```

This command forces S3 to be the secondary root for VLAN 10.

```
S1(config)# spanning-tree vlan 10 root primary
```

This command forces S1 to be the primary root for VLAN 10.

```
S1(config)# spanning-tree vlan 20 root secondary
```

Configure PVST+

```
S3(config)# spanning-tree vlan 20 priority 4096
```

This command sets the priority for S3 to be the lowest possible, making it most likely hat S3 will be the primary root for VLAN 20.

```
S1(config)# spanning-tree vlan 10 priority 4096
```

This command sets the priority for S1 to be the lowest possible, making it most likely hat S1 will be the primary root for VLAN 10.

VLAN

Cisco Switch IOS Commands	
Enter global configuration mode.	S1# **configure terminal**
Create a VLAN with a valid id number.	S1(config)# **vlan** vlan_id
Specify a unique name to identify the VLAN.	S1(config)# **name** vlan_name
Return to the privileged EXEC mode.	S1(config)# **end**

```
s1# configure terminal
s1(config)# interface F0/18
s1(config-if)# switchport mode access
s1(config-if)# switchport access vlan 20
s1(config-if)# end
```

```
S1(config)# interface FastEthernet0/1
S1(config-if)# switchport mode trunk
S1(config-if)# switchport trunk native vlan 99
S1(config-if)# switchport trunk allowed vlan 10,20,30
S1(config-if)# end
```

Inter-VLAN

```
Router(config)#interface fastethernet 0/0
Router(config-if)#no shutdown
Router(config-if)#interface fastethernet 0/0.10
Router(config-subif)#description vlan 10
Router(config-subif)#encapsulation dot1q 10
Router(config-subif)#ip address 192.168.a.254 255.255.255.0
Router(config-subif)#exit
Router(config-if)#interface fastethernet 0/0.20
Router(config-subif)#description vlan 20
Router(config-subif)#encapsulation dot1q 20
Router(config-subif)#ip address 192.168.b.254 255.255.255.0
Router(config-subif)#exit
```

VTP

VTP Configuration in global configuration mode:

```
Switch#config terminal
Switch(config)#vtp version 2
Switch(config)#vtp mode server
Switch(config)#vtp domain cisco
Switch(config)#vtp password mypassword
```

VTP Configuration in VLAN configuration mode:

```
Switch#vlan database
Switch(vlan)#vtp v2-mode
Switch(vlan)#vtp server
Switch(vlan)#vtp domain cisco
Switch(vlan)#vtp password mypassword
```

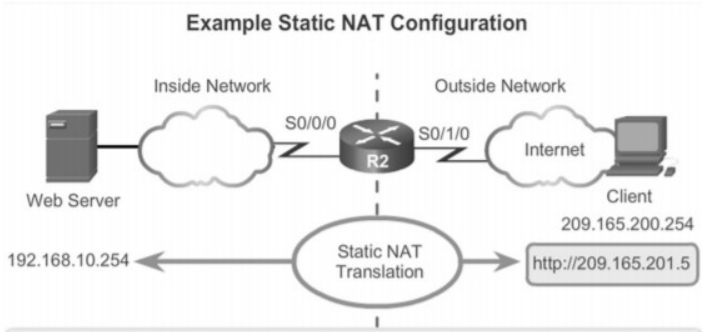
```
Switch#show vtp staus
MDF_Switch#show vtp counters
```

```
Switch(vlan)#vtp pruning
To make specific VLANs pruning ineligible
Switch(config)#interface fastethernet 0/3
Switch(config-if)#switchport trunk pruning vlan
                  remove vlan-id
```

Static NAT

```
ip nat inside source static 192.168.1.11
ip nat inside source static 192.168.1.12
ip nat inside source static 192.168.2.11
ip nat inside source static 192.168.2.12
int fa0/0.10
ip nat inside
exit
int fa0/0.20
ip nat inside
exit
int se0/0
ip nat outside
```

Step	Action
1	Establish static translation between an inside local address and an inside global address. `Router(config)# ip nat inside source static local-ip global-ip`
2	Specify the inside interface. `Router(config)# interface type number`
3	Mark the interface as connected to the inside. `Router(config-if)# ip nat inside`
4	Exit interface configuration mode. `Router(config-if)# exit`
5	Specify the outside interface. `Router(config)# interface type number`
6	Mark the interface as connected to the outside. `Router(config-if)# ip nat outside`




```
Establishes static translation between an inside local address and
an inside global address.
R2(config)# ip nat inside source static 192.168.10.254 209.165.201.5

R2(config)# interface Serial0/0/0
R2(config-if)# ip address 10.1.1.2 255.255.255.252
Identifies interface serial 0/0/0 as an inside NAT interface.
R2(config-if)# ip nat inside
R2(config-if)# exit

R2(config)# interface Serial0/1/0
R2(config-if)# ip address 209.165.200.225 255.255.255.224
Identifies interface serial 0/1/0 as the outside NAT interface.
R2(config-if)# ip nat outside
```

```
R2# show ip nat translations
```

Dynamic NAT

Dynamic NAT Configuration Steps	
Step 1	Define a pool of global addresses to be used for translation. **ip nat pool** *name start-ip end-ip* { **netmask***netmask* | **prefix-length** *prefix-length* }
Step 2	Define a standard access list permitting the addresses that should be translated. **access-list** *access-list-number* **permit** *source [source-wildcard]*
Step 3	Establish dynamic source translation, specifying the access list and pool defined in prior steps. **ip nat inside source list** *access-list-number* **pool** *name*
Step 4	Identify the inside interface. **interface** *type number* **ip nat inside**
Step 5	Identify the outside interface. **interface** *type number* **ip nat outside**

PAT

conf t
ip nat pool net 161.246.1.101 161.246.1.103 netmask 255.255.255.0
access-list 1 permit 192.168.1.0 0.0.0.255
access-list 1 permit 192.168.2.0 0.0.0.255
ip nat inside source list 1 pool net overload

end

Step 1	Define a pool of global addresses to be used for overload translation. **ip nat pool** *name start-ip end-ip* {**netmask** *netmask* | **prefix-length** *prefix-length*}
Step 2	Define a standard access list permitting the addresses that should be translated. **access-list** *access-list-number* **permit** *source* [*source-wildcard*]
Step 3	Establish overload translation, specifying the access list and pool defined in prior steps. **ip nat inside source list** *access-list-number* **pool** *name* **overload**
Step 4	Identify the inside interface. **interface** *type number* **ip nat inside**
Step 5	Identify the outside interface. **interface** *type number* **ip nat outside**

Step 1	Define a standard access list permitting the addresses that should be translated. **access-list** *access-list-number* **permit** *source*[*source-wildcard*]
Step 2	Establish dynamic source translation, specifying the ACL, exit interface and overload options. **ip nat inside source list***access-list-number***interface** *type number* **overload**
Step 3	Identify the inside interface. **interface** *type number* **ip nat inside**
Step 4	Identify the outside interface. **interface** *type number* **ip nat outside**

EIGRP

```
conf t
router eigrp 100
eigrp router-id 1.1.1.1
network 192.168.1.80
network 192.168.1.76
network 192.168.1.56
no auto-summary
end
```

— We can determine the EIGRP metric as follows:

1. Determine the link with the slowest bandwidth and use that value to calculate bandwidth (10,000,000/bandwidth).
2. Determine the delay value for each outgoing interface on the way to the destination and add the delay values and divide by 10 (sum of delay/10).
3. This composite metric produces a 24-bit value which EIGRP multiplies with 256.

[K1 * bandwidth + K3 * delay] * 256 = Metric

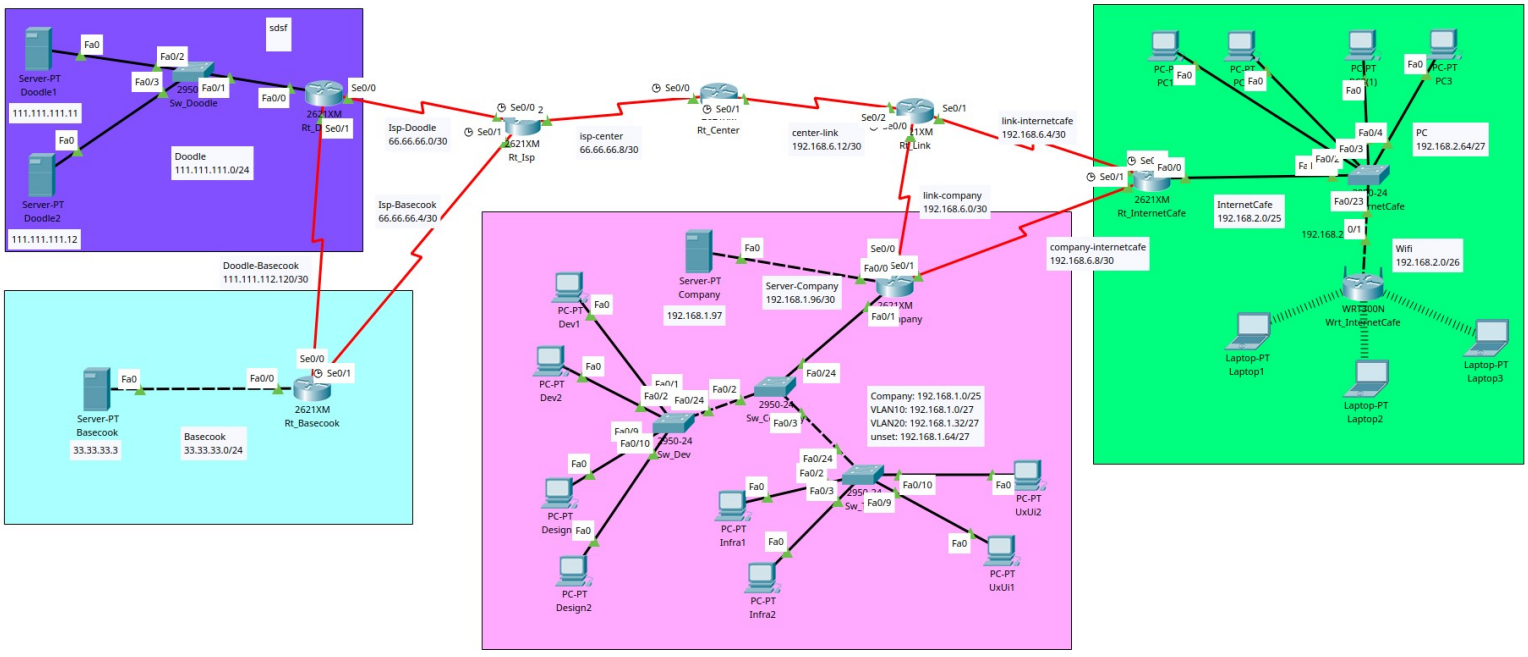
Because K1 and K3 both equal 1, the formula becomes:

(Bandwidth + Delay) * 256 = Metric

((10,000,000 / bandwidth) + (sum of delay / 10)) * 256 = Metric

```
R2# show ip route
D 192.168.1.0/24 [90/3012096] via 192.168.10.10, 00:12:32, Serial0/0/1
```

Subnet Mask	CIDR Value	Subnet Mask	CIDR Value
255.0.0.0	/8	255.255.240.0	/20
255.128.0.0	/9	255.255.248.0	/21
255.192.0.0	/10	255.255.252.0	/22
255.224.0.0	/11	255.255.254.0	/23
255.240.0.0	/12	255.255.255.0	/24
255.248.0.0	/13	255.255.255.128	/25
255.252.0.0	/14	255.255.255.192	/26
255.254.0.0	/15	255.255.255.224	/27
255.255.0.0	/16	255.255.255.240	/28
255.255.128.0	/17	255.255.255.248	/29
255.255.192.0	/18	255.255.255.252	/30
255.255.224.0	/19	255.255.255.254	/31 Not valid



Sw_Doodle

```
en
vlan database
vlan 10 name Doodle
exit
conf t
hostname Sw_Doodle
int range f0/1-3
sw mo ac
sw ac v 10
```

Rt_Doodle

```
en
conf t
hostname Rt_Doodle
int f0/0
ip add 111.111.111.1 255.255.255.0
no sh
int s0/0
ip add 66.66.66.2 255.255.255.252
no sh
int s0/1
ip add 111.111.112.121 255.255.255.252
no sh
ip route 33.33.33.0 255.255.255.0 s0/1
router eigrp 100
no auto-summary
network 111.111.111.0
network 66.66.66.0
ip address 192.168.2.73 255.255.255.252
clock rate 56000
no shutdown
exit
```

Rt_Basecook

```
en
conf t
hostname Rt_Basecook
int f0/0
```

ip add 33.33.33.1 255.255.255.0
no sh
int s0/0
ip add 66.66.66.6 255.255.255.252
no sh
int s0/1
ip add 111.111.112.122 255.255.255.252
no sh
ip route 111.111.111.0 255.255.255.0 s0/1
router eigrp 100
no auto-summary
network 33.33.33.0
network 66.66.66.4

Rt_Isp

en
conf t
hostname Rt_Isp
int s0/0
ip add 66.66.66.1 255.255.255.252
no sh
int s0/1
ip add 66.66.66.5 255.255.255.252
no sh
int s0/2
ip add 66.66.66.9 255.255.255.252
no sh
router eigrp 100
no auto-summary
network 66.66.66.0
network 66.66.66.4
network 66.66.66.8

Rt_Center

en
conf t
hostname Rt_Center
int s0/0
ip add 66.66.66.10 255.255.255.252
no sh

int s0/1
ip add 192.168.6.14 255.255.255.252
no sh
router eigrp 100
redistribute ospf 100 metric 1000 100 255 1 1500
network 66.66.66.8
router ospf 100
log-adjacency-changes
redistribute eigrp 100 metric 1 subnets
network 192.168.6.12 0.0.0.3 area 0

Rt_Company

en
conf t
hostname Rt_Company
int f0/1
no sh
int f0/1.10
description Vlan-Dev
en do 10
ip add 192.168.1.30 255.255.255.224
int f0/1.20
description Vlan-Design
en do 20
ip add 192.168.1.62 255.255.255.224
int s0/0
ip add 192.168.6.2 255.255.255.252
no sh
int s0/1
ip add 192.168.6.9 255.255.255.252
no sh
int f0/0
ip add 192.168.1.98 255.255.255.252
no sh
router ospf 100
network 192.168.6.0 0.0.0.3 area 0
network 192.168.6.8 0.0.0.3 area 0
network 192.168.1.96 0.0.0.3 area 0

Sw_Company

```
en
vlan database
vlan 10 name Dev
vlan 20 name Design
exit
conf t
hostname Sw_Company
vtp do vtp
vtp mo t
vtp ve 2
int f0/2
sw mo tr
int f0/3
sw mo tr
```

Sw_Dev

```
en
vlan database
vlan 10 name Dev
vlan 20 name Design
exit
conf t
hostname Sw_Dev
vtp do vtp
vtp mo s
vtp ve 2
int f0/24
sw mo tr
no sh
int range f0/1-8
sw mo ac
sw ac v 10
int range f0/9-15
sw mo ac
sw ac v 20
```

Sw_Tech

```
en
vlan database
vlan 10 name Infra
```

```
vlan 20 name UxUi
exit
conf t
hostname Sw_Tech
vtp do vtp
vtp mo c
vtp ve 2
int f0/24
sw mo tr
no sh
int range f0/1-8
sw mo ac
sw ac v 10
int range f0/9-15
sw mo ac
sw ac v 20
```

Rt_InternetCafe

```
en
conf t
hostname Rt_InternetCafe
int s0/0
ip add 192.168.6.6 255.255.255.252
no sh
int s0/1
ip add 192.168.6.10 255.255.255.252
no sh
int f0/0
ip add 192.168.2.126 255.255.255.128
no sh
router ospf 100
network 192.168.6.4 0.0.0.3 area 0
network 192.168.6.8 0.0.0.3 area 0
network 192.168.2.0 0.0.0.127 area 0
```

Sw_InternetCafe

```
en
conf t
hostname Sw_InternetCafe
int f0/24
```

no sh

Wrt_InternetCafe

IP Address 192.168.2.126
Network: 192.168.2.0/26
Subnet Mask 255.255.255.128

SSID: InternetCafe
WPA2-PSK: InternetCafe

StartIp: 192.168.2.1
EndIp: 192.168.2.62

Rt_Link

```
en
conf t
hostname Rt_Isp
int s0/0
ip add 192.168.6.1 255.255.255.252
no sh
int s0/1
ip add 192.168.6.5 255.255.255.252
no sh
int s0/2
ip add 66.66.66.62 255.255.255.252
no sh
ip route 192.168.6.0 255.255.255.252 66.66.66.61
ip route 192.168.6.4 255.255.255.252 66.66.66.61
router ospf 100
redistribute static subnets
network 192.168.6.0 0.0.0.3 area 0
network 192.168.6.4 0.0.0.3 area 0
```