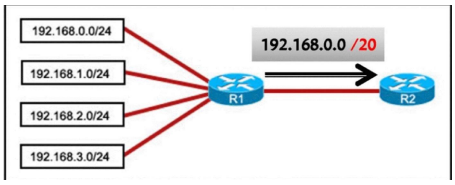
## SUMMARIZATION

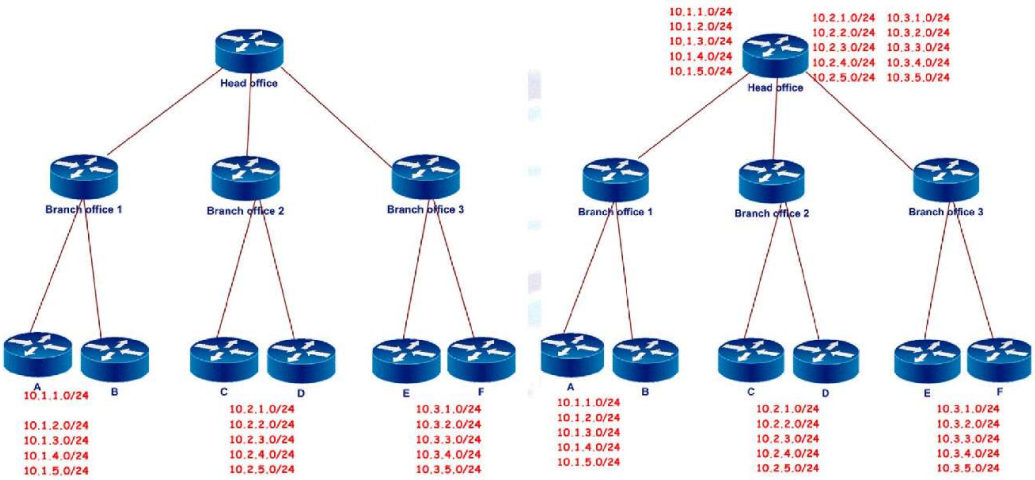
- It is the process of combining smaller networks in to single large sub-network
- Combining the contagious address into one and send to neighbor




**Advantages**

- Minimizing the routing table.
- Less use of resources like memory, processor, bandwidth.

## Routing table without summarization

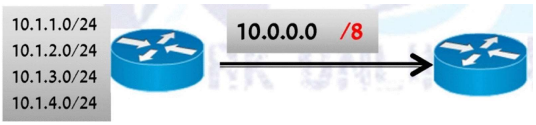



The more bigger the size of the network the more bigger the size of the routing table

- **Auto summary**
  - Summarization is done to a default class full boundary
    - A /8
    - B /16
    - C /24
  - Routing protocol like **RIPv2, EIGRP, BGPv4** support auto summary and can be disabled
  - Routing protocol like **OSPF and ISIS** doesn't support auto summary
- **Manual Summary**
  - Administrator manually configures Summarization to specific boundary
  - It is supported by all classless routing protocols
  - EIGRP , RIPV2 , OSPF , BGP support Manual summary.

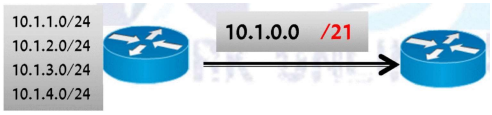
## Auto-summary

- **RIPv2, EIGRP, BGPv4** support auto summary by default
- Auto-summary can be disabled using **No auto-summary**
- **OSPF and ISIS** doesn't support auto summary



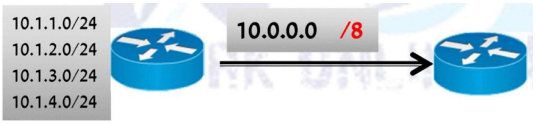

### Manual summary

- Administrator manually configures Summarization to specific boundary
- EIGRP , RIPV2 , OSPF , BGP support Manual summary.




## Auto-summary

- **RIPv2, EIGRP, BGPv4** support auto summary by default
- Auto-summary can be disabled using **No auto-summary**
- **OSPF and ISIS** doesn't support auto summary




## Steps for calculating manual summary :

1. Write Binary value of first and the last number.
2. Separate the portions in to two parts ( common and un-common)
   - ( 0 – 0 or 1- 1 are common)
3. Convert right side values of the first number in to zeros ( change in to decimal) and left side values should be same.
4. count the left side bits (to find the / value)

Example : 1

Summarize the following addresses to nearest subnet mask possible

10.1.0.0/24
10.1.2.0/24
10.1.3.0/24
10.1.4.0/24
10.1.5.0/24
10.1.6.0/24

1. Write Binary value of first and the last number.
2. Separate the portions in to two parts
   - 0 – 0 & 1-1 common
   - 0 – 1 not common

| | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10.1.0.0/24 | 10. | | 1. | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10.1.6.0/24 | 10. | | 1. | | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Example : 1

Summarize the following addresses to nearest subnet mask possible

10.1.0.0/24
10.1.2.0/24
10.1.3.0/24
10.1.4.0/24
10.1.5.0/24
10.1.6.0/24

- Convert right side values of the first number in to zeros (change in to decimal) and left side values should be same.

| | | | | Same | | | | Zero | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 10. | | 1. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10.1.0.0/24 | 10. | | 1. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10.1.6.0/24 | 10. | | 1. | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

Example : 1

Summarize the following addresses to nearest subnet mask possible

10.1.0.0/24
10.1.2.0/24
10.1.3.0/24
10.1.4.0/24
10.1.5.0/24
10.1.6.0/24

- count the left side bits (to find the / value)

Summary-Address 10.1.0.0

| | | | | Same | | | | Zero | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 10. | | 1. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10.1.0.0/24 | 10. | | 1. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10.1.6.0/24 | 10. | | 1. | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

Mask / 21

SUMMARIZE THE FOLLOWING ADDRESSES TO NEAREST SUBNET MASK POSSIBLE

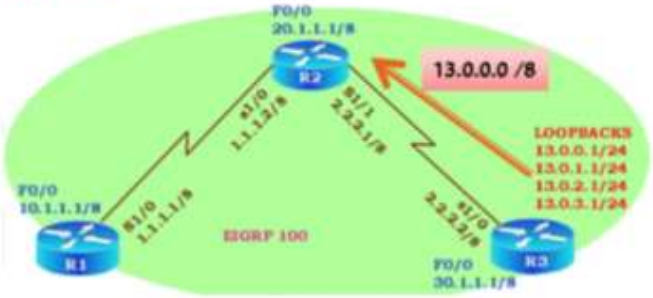
172.16.25.0/24
172.16.26.0/24
172.16.27.0/24
172.16.28.0/24
172.16.29.0/24

| | | |
|---|---|---|
| 172.16.25.0/24 | 172.16. | 0 0 0 1 1 0 0 1 00000000 |
| 172.16.29.0/24 | 172.16. | 0 0 0 1 1 1 0 1 00000000 |

Now will see the normal 1.default advertisement
2.Auto summarization and then
3.Manual summarization

## Lab : EIGRP Auto- Summarization

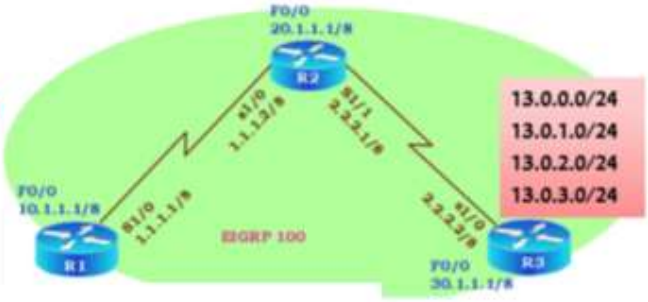
```
R1(config)#router EIGRP 100
R1(config-router)# network 10.0.0.0
R1(config-router)# network 1.0.0.0

R2(config)#router  EIGRP 100
R2(config-router)# network 20.0.0.0
R2(config-router)# network 2.0.0.0
R2(config-router)# network 1.0.0.0

R3(config)#router EIGRP 100
R3(config-router)# net 30.0.0.0
R3(config-router)# net 2.0.0.0
R3(config-router)# net 13.0.0.0
```




```
R2#sh ip route eigrp
D    10.0.0.0/8 [90/2172416] via 1.1.1.1, 00:05:07, Serial1/0
D    13.0.0.0/8 [90/2297856] via 2.2.2.2, 00:04:38, Serial1/1
D    30.0.0.0/8 [90/2172416] via 2.2.2.2, 00:04:38, Serial1/1.
```

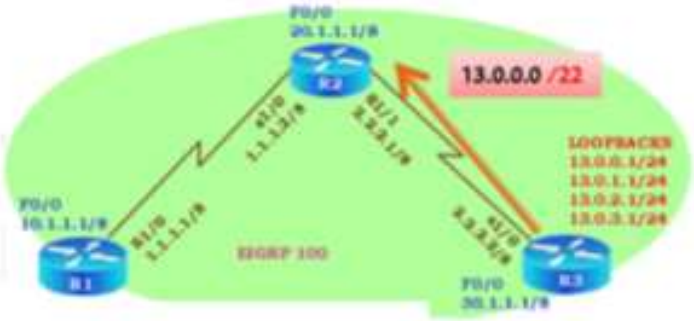
## Disable Auto-summary

```
R-X(config)#router EIGRP 100
R-X(config-router)#no auto-summary
```




```
R2#sh ip route eigrp
D    10.0.0.0/8 [90/2172416] via 1.1.1.1, 00:08:34, Serial1/0
     13.0.0.0/24 is subnetted, 4 subnets
D       13.0.1.0 [90/2297856] via 2.2.2.2, 00:00:10, Serial1/1
D       13.0.0.0 [90/2297856] via 2.2.2.2, 00:00:10, Serial1/1
D       13.0.3.0 [90/2297856] via 2.2.2.2, 00:00:10, Serial1/1
D       13.0.2.0 [90/2297856] via 2.2.2.2, 00:00:10, Serial1/1
D    30.0.0.0/8 [90/2172416] via 2.2.2.2, 00:08:05, Serial1/1
```

## Manual summary




```
R3(config)#int s1/0
R3(config-if)# ip summary-address EIGRP 100 13.0.0.0  255.255.252.0
```

```
R2#sh ip route eigrp
     10.0.0.0/8 [90/2172416] via 1.1.1.1, 00:10:49, Serial1/0
     13.0.0.0/22 is subnetted, 1 subnets
D       13.0.0.0 [90/2297856] via 2.2.2.2, 00:00:19, Serial1/1
D    30.0.0.0/8 [90/2172416] via 2.2.2.2, 00:10:20, Serial1/1
```

**********END OF SUMMARIZATION**********

********** RIP EIGRP OSPF AUTHENTICATION**********

## Routing protocol Authentication

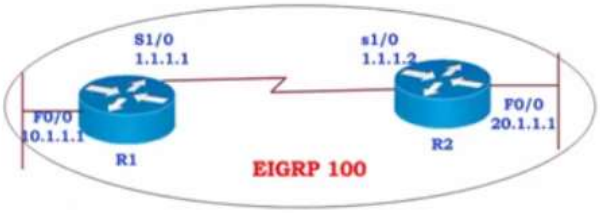
- *a router authenticates the source of each routing update packet that it receives.*
- *Many routing protocols support authentication .*

## EIGRP Authentication

*Router(config)#* ***key chain*** *< name-of-chain>*
*Router(config-keychain)#* ***key*** *<key-id>*
*Router(config-keychain-key)#* ***key-string*** *<text>*
*Router(config-keychain-key)#* ***exit***

*Router(config)#* ***interace serial 1/0***
*Router(config-if)#* ***ip authentication mode EIGRP*** *<autonomous-system>* ***md5***
*Router(config-if)#****ip authentication key-chain EIGRP*** *< AS no > <name-of-chain>*

***R1#debug EIGRP packets***
***R1#show ip EIGRP neighbors***
***R1#show key chain***



First create key chain . Key ID / Key number both are same , key Number and Key string should be same on both the routers . If there is auth failure then there will not be any neighbourship between two routers .

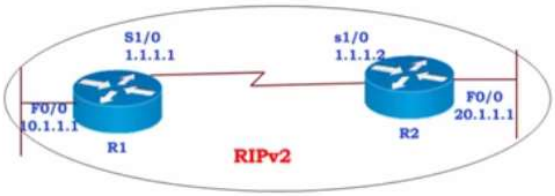
Use “debug eigrp packets” – to check the backend eigrp process .

```
R1(config)#Key chain CHAINR1
R1(config-keychain)#Key 1
R1(config-keychain-key)#Key-string cisco

R1(config)#int s1/0
R1(config-if)#  ip authentication mode EIGRP 100 md5
R1(config-if)#  ip authentication key-chain EIGRP 100 CHAINR1

R2(config)#Key chain CHAINR2
R2(config-keychain)#Key 1
R2(config-keychain-key)#Key-string cisco

R2(config-keychain-key)#int s1/0
R2(config-if)#  ip authentication mode EIGRP 100 md5
R2(config-if)#  ip authentication key-chain EIGRP 100 CHAINR2
R2(config-if)#end
```

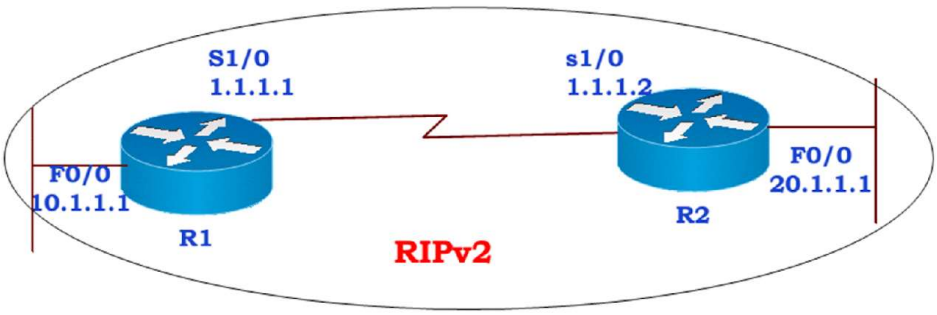
## RIPv2 Authentication

```
Router(config)# key chain <name-of-chain>
Router(config-keychain)# key <key-id>
Router(config-keychain-key)# key-string <text>
Router(config-keychain-key)# exit

Router(config)# interace serial 1/0
Router(config-if)# ip rip authentication mode <md5/text>
Router(config-if)#ip rip authentication key-chain <name-of-chain>
```

```
R1#debug ip rip events
R1#clear ip route *
R1#show key chain
```




LAB RIPv2 AUTHENTICATION



TASK

- Configure R1 and R2 to exchange the routes only after successful authentication

```
R1(config)#
R1(config)#Key chain CHAINR1
R1(config-keychain)#Key 1
R1(config-keychain-key)#Key-string cisco123

R1(config-keychain-key)#int s1/0
R1(config-if)#  ip rip authentication mode md5
R1(config-if)#  ip rip authentication key-chain CHAINR1

R2(config)#Key chain CHAINR2
R2(config-keychain)#Key 1
R2(config-keychain-key)#Key-string cisco

R2(config-keychain-key)#int s1/0


R2(config-if)#  ip rip authentication mode md5
R2(config-if)#  ip rip authentication key-chain CHAINR2
R2(config-if)#
```

NOTE :

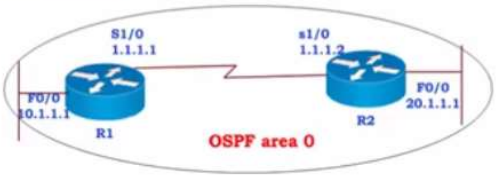
- key no and the key string should be same on both routers but the above configuration have mismatch of passwords.

```
R1#debug ip rip events
    RIP event debugging is on

R1#clear ip route *
    *Mar  1 00:24:24.751: RIP: sending request on FastEthernet0/0 to 224.0.0.9
    *Mar  1 00:24:24.755: rip_route_adjust for FastEthernet0/0 coming up
    *Mar  1 00:24:24.755: RIP: sending request on FastEthernet0/0 to 224.0.0.9
    *Mar  1 00:24:24.763: RIP: sending request on Serial1/0 to 224.0.0.9
    *Mar  1 00:24:24.763: rip_route_adjust for Serial1/0 coming up
    *Mar  1 00:24:24.767: RIP: sending request on Serial1/0 to 224.0.0.9
    *Mar  1 00:24:38.775: RIP: ignored v2 packet from 1.1.1.2 (invalid authentication)
R1#undebug all
    All possible debugging has been turned off

R1#sh key chain
        Key-chain CHAINR1:
            key 1 -- text "cisco123"


R2#sh key chain
        Key-chain CHAINR2:
            key 1 -- text "cisco"
```

# OSPF Authentication

### Clear Text Authentication

```
Rx(config)#int s1/0
Rx(config-if)# ip ospf authentication
Rx(config-if)# ip ospf authentication-key cisco123
Rx(config-if)# end
```




### MD5 Authentication

```
Rx(config)#int s1/0
Rx(config-if)# ip ospf authentication message-digest
Rx(config-if)# ip ospf message-digest-key 1 md5 cisco123
Rx(config-if)#end
```

- Use clear text authentication

```
R1(config)#int s1/0
R1(config-if)#  ip ospf authentication
R1(config-if)#  ip ospf authentication-key cisco
R1(config-if)# end

R2(config)#int s1/0
R2(config-if)#  ip ospf authentication
R2(config-if)#  ip ospf authentication-key cisco
```

TASK

- Remove clear text authentication and configure md5 authentication

On R1, R2

```
Rx(config)#int s1/0
Rx(config-if)#  ip ospf authentication message-digest
Rx(config-if)#  ip ospf message-digest-key 1 md5 cisco123
Rx(config-if)#end
```

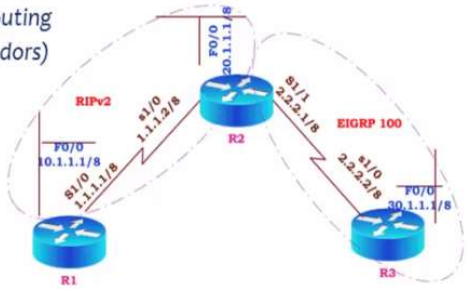
**********END OF RIP EIGRP OSPF AUTHENTICATION**********

## Redistribution

*The process of exchanging routing information between different routing protocols*

***When we use multiple protocol***

1. *migrating to a more advanced routing*
2. *Mismatch between devices (Vendors)*
3. *Political boundaries*




***Internal routes*** *are routes advertised with in the same protocol*

***External routes*** *are routes which gets redistributed .*

Migrating to a more advanced routing – lets say I m running very old protocol in my network now I want to implement some advanced routing which has advanced features – now i will do redistribution I will tell the router that take the routes from rip and convert it in to eigrp , accept eigrp routes and translate it in to RIP routes , it can be any protocols the main goal of redistribution is exchanging the info of one protocol to another protocol

But the question is when it is more appropriate to implement redistribution

## To configure Redistribution

Router must be running both protocols (at least one interface) (R2)

Change metric.

- RIPv2 : hop counts
- OSPF : cost
- EIGRP : BW+ delay+ load + MTU+ reliability

## RIPv2 –EIGRP Redistribution

```
R1#sh ip route rip
R    2.0.0.0/8 [120/3] via 1.1.1.2, 00:00:17, Serial1/0
R    20.0.0.0/8 [120/1] via 1.1.1.2, 00:00:17, Serial1/0
R    30.0.0.0/8 [120/3] via 1.1.1.2, 00:00:17, Serial1/0
```


```
R2(config)#router eigrp 100
R2(config-router)#redistribute rip metric 1000 2000 255 1 1500
R2(config-router)#exit
```

```
R2(config)#router rip
R2(config-router)#redistribute eigrp 100 metric  3
```

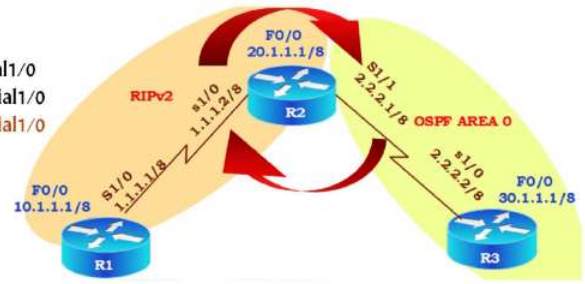
```
R3#sh ip route eigrp
D EX 1.0.0.0/8 [170/3584000] via 2.2.2.1, 00:00:46, Serial1/0
D EX 20.0.0.0/8 [170/3584000] via 2.2.2.1, 00:00:46, Serial1/0
D EX 10.0.0.0/8 [170/3584000] via 2.2.2.1, 00:00:46, Serial1/0
```

Internal and external routes will be advertised as “R” in case of RIP
But in case of EIGRP internal routes will be defined as “D” external as “D EX”

In EIGRP if I don’t mention the metrics while adv it inside RIP it will take the metric as “infinite”
And external routes AD value  will be 170

## RIPv2-OSPF Redistribution

```
R1#sh ip route rip
R    2.0.0.0/8 [120/3] via 1.1.1.2, 00:00:04, Serial1/0
R    20.0.0.0/8 [120/1] via 1.1.1.2, 00:00:04, Serial1/0
R    30.0.0.0/8 [120/5] via 1.1.1.2, 00:00:04, Serial1/0
```




```
R2(config)#router rip
R2(config-router)#redistribute ospf 1 metric  5
R2(config-router)#end
```

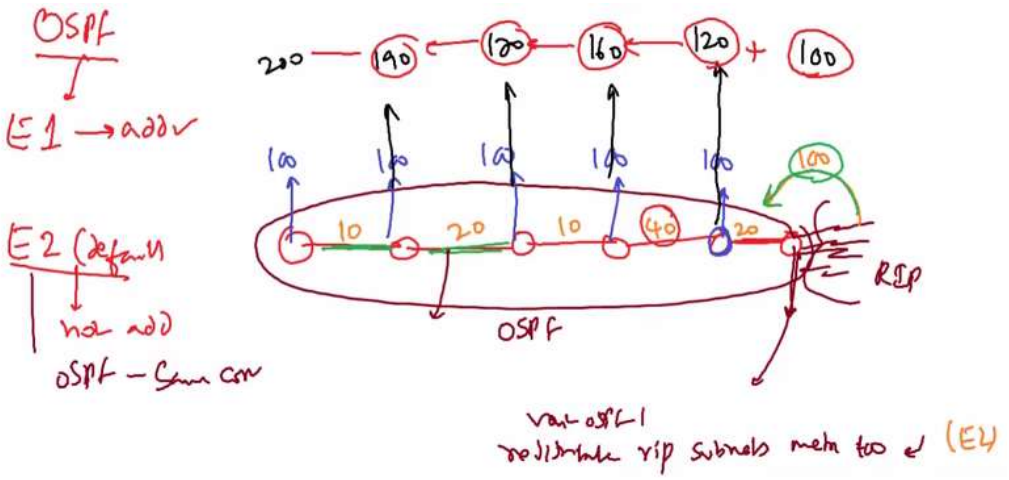
```
R2(config)#router ospf 1
R2(config-router)#redistribute rip subnets metric 100 metric-type 1
R2(config-router)#exit
```

```
R3#sh ip route ospf
O E1 1.0.0.0/8 [110/164] via 2.2.2.1, 00:00:31, Serial1/0
O E1 20.0.0.0/8 [110/164] via 2.2.2.1, 00:00:31, Serial1/0
O E1 10.0.0.0/8 [110/164] via 2.2.2.1, 00:00:31, Serial1/0
```

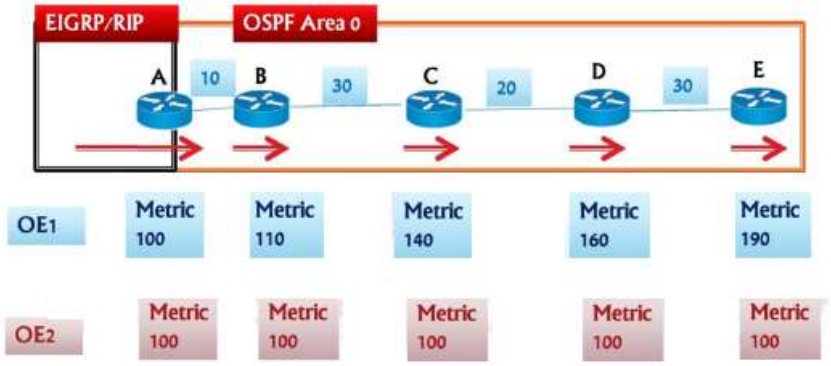
For external routes in OSPF

- Default Metric = 20
- Default metric-type = E2

If I dont use subents keyword then it will take classfull routes
Two metric types – metric type 1 (E1) and metric type 2 (E2)
Metric and Metric type both are optional
E1 will add the indivisual link cost while E2 – will not add induvisual cost




## OSPF metric-type (OE1 / OE2)




## OSPF metric-type (OE1 / OE2)

### OE2

- will not add the individual metric
- it remains same metric for al l the routers.
- Default for external routes

### OE1

- will add the individual metric
- it changes as move from router to router.

## RIPv2 –EIGRP Redistribution

```
R2(config)#router eigrp 100
R2(config-router)#redistribute rip metric 1000 2000 255 1 1500
R2(config-router)#exit
R2(config)#router rip
R2(config-router)#redistribute eigrp 100 metric  3
```

# RIPv2-OSPF Redistribution

```
R2(config)#router rip
R2(config-router)#redistribute ospf 1 metric  5
R2(config-router)#end

R2(config)#router ospf 1
R2(config-router)#redistribute rip subnets metric 100 metric-type 1
R2(config-router)#exit
```

```
R3#sh ip rou
O E1 1.0.0.0/
O E1 20.0.0.0
O E1 10.0.0.0
```

# EIGRP –OSPF Redistribution

```
R2(config)#router ospf 1
R2(config-router)#redistribute eigrp 100 subnets
R2(config-router)#exit

R2(config)#router eigrp 100
R2(config-router)#redistribute ospf 1 metric 1000 20000 255 1 2500
R2(config-router)#exit
```

```
R3#sh ip route
O E2 1.0
O E2 20.
O E2 10.
```

## EIGRP different AS- Redistribution

```
R2(config)#router eigrp 200
R2(config-router)#redistribute eigrp 100
R2(config-router)#end

R2(config)#router eigrp 100
R2(config-router)#redistribute eigrp 200
R2(config-router)#exit
```