

### **Password CLI**

1. La 1° password è “cisco”.
2. scrivi “en” nel terminale.
3. La 2° password è “class”.

### **L'importanza di una Rete**

L'infrastruttura di rete è “critica”, ossia essenziale per svolgere le attività dell'azienda.

### **Dispositivi “Plug and Play”**

Dispositivi che dopo averli collegati funzionano correttamente, molto famosi nelle piccole reti private.

Noi prenderemo in esame dispositivi non “plug and play”, ossia non configurati e non funzioneranno.

I router vanno configurati.

Lo switch di base è plug and play, ma non possiamo darlo per scontato, perché esistono switch che operano anche come router (switch multilayer) e che quindi vanno configurati.

### **Affidabilità di una Rete**

Se si rompe un link la rete deve ancora funzionare.

Ci sono molte soluzioni (quasi tutte standard) che vanno ad aumentare la robustezza ai guasti.

### **Il Corso di “Progettazione di Reti Informatiche”**

Noi ci concentreremo sulle reti private/aziendali.

### **Telefoni IP**

I telefoni ora sono Host con interfacce IN/OUT (cassa, microfono e tastiera), ossia una macchina che usa il protocollo IP per trasmettere i sample di voce catturati dal microfono i quali verranno ricevuti e riprodotti da un altro telefono IP da qualche altra parte.

Vedi protocollo VoIP.

### **Batterie di Server**

Cloud aziendale, dove risiedono gli applicativi.

### **Reti a Livelli**

Le reti seguono un approccio “Divide et Impera”, divido il progetto in livelli, dove ogni livello ha un certo livello di ridondanza (legata strettamente all'affidabilità).

Problema del Percorso Chiuso tra Switch - Spanning Tree Protocol

La ridondanza nelle reti è “anti ridondante”, ossia una rete non può essere ridondante per permettere al protocollo ethernet di funzionare.

Ad esempio se collegassi due switch in un ciclo e venisse trasmesso un pacchetto broadcast congestionerei la rete.

Ma gli switch moderni hanno contromisure per evitare questa cosa (Spanning Tree Protocol).

Ma in certi casi la ridondanza è richiesta per aumentare l'affidabilità e quindi serve introdurre anche questa contromisura, la quale disabilità i link (le porte in realtà) che compongono il percorso chiuso.

### **First Hop Redundancy**

Posso mettere 2 gateway in una unica rete? Un host ha un solo gateway.

Potrei avere un router 4g e un router DSL nella solita rete e usare uno quando l'altro è inutilizzabile.

Di base il protocollo IPv4 non permette questa cosa (ossia automaticamente usare un gateway secondario se il primario cade).

Per cambiare gateway devo cambiarlo in tutti gli host.

IPv6 invece ha previsto questa cosa.

### **Link Aggregation**

Posso aggregare link per aumentare la capacità e quindi diminuire il collo di bottiglia?

Lo standard odierno è avere porte nell'ordine di Gigabit al secondo.

Uno switch di solito ha 24 porte, uno switch può gestire più di 24 Gigabit al secondo? dipende dalla qualità dello switch e quindi anche dal costo.

Lo spanning tree protocol permette di vedere come un unico link vari link differenti.

### **Gestione VLAN**

Vedremo nel concreto come si creano le VLAN e come gli host possono fare comunicazione tra VLAN diverse.

### **Controllo degli Accessi**

Fare in modo che gli switch abbiano la possibilità di stabilire delle regole che permettono di fornire servizi o meno.

### **Piani di Allocazione degli Indirizzi IP**

Gli indirizzi IP sono limitati, soprattutto in una rete privata.

Ogni sottorete deve avere un range preciso di indirizzi IP.

### **Troubleshooting**

Individuare il motivo all'origine di un problema.

### **NAT e configurazione**

Imparare a configurare un NAT.

### **DHCP e configurazione**

Imparare a configurare un DHCP.

### **Switch Multilayer**

Apparati utilizzati dovunque tranne che nel livello di accesso.

Al loro interno racchiudono le funzionalità di uno switch e di un router.

### **Internetworking - Routing**

OSPF e RIPv2 (Distance Vector)

OSPF è molto potente e a volte anche su reti aziendali risulta “overkill”.

Ogni protocollo di routing ha varie conseguenze di utilizzo, quando si sceglie un algoritmo di routing bisogna essere consci di come funzionano, altrimenti durante il troubleshooting si piange.

Ergo bisogna conoscere le tecnologie che si usano.

### **Protocolli Proprietari**

I router Cisco usano dei protocolli di routing tutti loro e che trovi solo nei router Cisco.

Gli algoritmi possono anche essere brevettati.

Non li studieremo in questo corso, ma quando si va a lavorare in un posto bisogna sapere che protocolli si usano.

### **Wide Area Networks**

#### **Filtraggio dei Pacchetti**

Dopo aver configurato la rete e quindi tutti possono comunicare con tutti, mi occupo del filtraggio.

OSPF è progettato per garantire che tutti possono comunicare con tutti.

Può interessarmi anche fare in modo che abbia meccanismi per NON dire a qualcuno come raggiungere certe destinazioni.

I provider fanno accordi tra di loro, ma se non hanno accordi, allora un pacchetto di un certo provider non dovrebbe poter raggiungere il secondo provider.

#### **Richiesta ARP**

Prima del primo ping fatto da un host ad un altro host, viaggia una richiesta ARP, effettuata dal primo per ottenere l'indirizzo MAC del secondo.

#### **Cablaggio Interfacce e file di configurazione**

Come configurare le interfacce del router e come fare il cablaggio tra apparati.

In Cisco IoS un'interfaccia è una porta fisica di ingresso-uscita di pacchetti.

Con il comando “interface” vado ad operare sulla porta.

Ogni porta ha un identificatore univoco e un tipo.

- Type → Protocollo di livello 2 usato in quella porta
  - Ethernet
  - Gigabit Ethernet
  - Seriale
  - ...
- Port → Identificatore (non per forza numerico).
  - L'identificatore è legato alla porta ed è riportato anche sul retro fisico del router.

Con shutdown verifico che la porta sia attiva o meno.

Un cavo può essere di 3 tipi:

- Fibra ottica.
- rame (doppino di rame).
  - Standard Ethernet, uno standard che resiste da più di 50 anni.
  - Il formato del frame non è mai cambiato, ma il mezzo fisico e quindi anche la velocità massima di trasmissione sono migliorati.
    - T → Twisted Pair
    - Numero → Modo in cui è fatta la codifica dei bit sul mezzo.
- wireless (antenna).

Una porta può avere una di queste 3.

### **Copper TP**

Termina con un connettore RJ-45.

Le varie versioni di questi cavi cambiano dal confezionamento.

Nella maggior parte dei casi, il confezionamento è fatto per avere certe garanzie riguardo la protezione dal rumore esterno.

Ci sono varie categorie e ogni categoria varia nel confezionamento, infatti se apriamo i cavi possiamo vedere notevoli differenze tra un Cat3 (25 anni fa, adatto a velocità bassissime e quindi non c'era l'esigenza di proteggere il segnale) e un Cat6.

In un cavo ho 4 coppie.

Ho una coppia per trasmettere, una per ricevere e 2 non usate.

Un cavo di questo tipo può arrivare fino al Gigabit al secondo.

Un cavo può essere di 2 tipologie:

- **Straight-Through Cable.**
  - I pin sono collegati direttamente (Pin1 → Pin1).
  - Il problema è che se uno trasmette l'altro può solo ricevere, poiché la coppia per trasmettere è collegata alla coppia per trasmettere del ricevitore.
- **Crossover Cable.**
  - I pin non sono collegati direttamente (Pin1 → Pin3).
  - Permette ai due di trasmettere contemporaneamente.
  - Ma a volte alcune interfacce fanno automaticamente l'incrocio, in quel caso devo usare un cavo straight, se invece questa cosa non è automatica devo usare un crossover (anche se ormai le interfacce sono intelligenti e l'incrocio può essere regolato aumentaticamente tramite una negoziazione tra le 2 parti).
    - Il numero di incroci deve essere dispari, altrimenti non funziona.
    - In genere tra apparati uguali si usa uno straight.

### **Fibra Ottica**

La potenza del segnale diminuisce quadraticamente con la distanza.

La distanza porta-porta deve essere massimo 100 metri.

Nella fibra il messaggio non viaggia su elettricità e non è affatto da disturbi dovuti a interferenze elettromagnetiche.

I connettori e i cavi in fibra sono complessi, delicati e più costosi, mentre i connettori in rame sono robusti.

#### File di Configurazione del Router

Contiene la lista dei comandi che sono stati applicati al router per portarlo in quello stato, considerata anche la configurazione di partenza.

La configurazione di partenza si trova su una memoria RAM persistente chiamata NVRAM.  
All'avvio il contenuto della NVRAM viene copiato in RAM.

Il file nella NVRAM si chiama "startup-config".

Il file in RAM si chiama "running-config", ogni volta che viene eseguito un comando il file viene aggiornato, ma allo shutdown scompare.

commit → Copiare "running-config" in "startup-config".

Se lo "startup config" non esiste, il router parte con un "running config" predefinito.  
Lo vediamo quando apriamo il running-config per la prima volta e vediamo linee non scritte da noi.

Le configurazioni (immagini) possono essere anche messi in un server TFTP.

#### Connessioni Seriali

Comunicazione standard del livello fisico.

### Lezione 13 Marzo

#### ROUTING

Per un router configurare significa assegnare le credenziali di accesso e degli indirizzi assegnati alle interfacce, questo vale per il compito scritto.

Oggi vediamo come si configurano le informazioni utili per instradare i pacchetti, ovvero come si configura la tabella di routing, la quale è anche detta tabella di forwarding, bisogna fare attenzione dato che si differenziano in alcuni casi, è una struttura dati che si trova nella memoria dinamica, ovvero nella RAM, è usata nel piano dati del router per prendere decisioni su come instradare i pacchetti, noi eseguiremo dei comandi ed il router trasferisce o no queste informazioni all'interno della tabella di routing. Il router ha dei database di routing nei quali un router colleziona informazioni scambiate con altri router, il router processando queste informazioni va a riempire la tabella, abbiamo più database dato che un router può utilizzare più protocolli di routing.

#### La funzione principale dei router in una rete multi-hop:

**Il protocollo IP** utilizza un routing detto routing ad indirizzo di destinazione, quando un router riceve un pacchetto IP, nell'intestazione c'è l'indirizzo di destinazione ed è l'unica informazione all'interno del pacchetto utilizzata dal router per decidere dove ritrasmetterlo.  
Non è l'unico meccanismo, ce ne sono altri due, **label-swapping**, ovvero i pacchetti vengono etichettati e le etichette non necessariamente corrispondono all'indirizzo di

destinazione (ad esempio il protocollo MPLS), c'è un'altra soluzione è il **segment routing** che utilizza il *source routing*, dentro il pacchetto ci viene scritto il percorso che il pacchetto deve seguire. In una rete aziendale/compressorio privato l'uso di queste tecniche avanzate sono raramente utilizzate, queste soluzioni sono adottate all'estremità della rete.

### Come funziona il routing IP?

Un host ha un indirizzo composto dal prefisso che identifica il link IP (rete locale) a cui l'host appartiene e un suffisso che identifica quell'host all'interno di quel link, il routing IP sfrutta la gerarchia per rendere la soluzione più scalabile, ovvero non posso avere una tabella con tutti gli indirizzi di destinazione, ho gli indirizzi dei link a cui l'host destinatario appartiene, quindi si fa una separazione, prima si instrada il pacchetto alla rete a cui appartiene il destinatario e dopo una volta raggiunta questa rete, il pacchetto raggiungerà l'host di destinazione.

### Tabella di Routing

La tabella di routing ha un elenco di righe, per ogni riga ho una rete di destinazione, in un'altra colonna abbiamo il next hop, ovvero l'interfaccia di uscita e un'altra colonna è il costo (numero di hop necessari per arrivarci). Abbiamo diversi percorsi per andare ad una destinazione ciascuna con costi diversi.

Destination prefix	Next Hop	Cost
192.168.4.0/24	Fa0/0	0
192.168.0.0/24	192.168.66.1	2
192.168.3.0/24	192.168.68.1	1
...	...	...

Andiamo in packet tracer, nella rete della slide, i prefissi in questo caso sono tutti a 24 bit, andiamo su router E, nel modo privilegiato chiedo al router qual è la tabella di routing utilizzata, si usa: show ip route, nella tabella c'è una lettera maiuscola che indica la sorgente da cui il router ha inserito la riga, con R significa che il router ha trovato quel percorso attraverso il protocollo RIP, C indica connected, ovvero una riga che indica che il router è connesso direttamente a quella rete:

L'ultimo parametro della riga indica la porta da cui far uscire il pacchetto.

Nella sesta riga abbiamo due alternative, ci dice che possiamo raggiungere il destinatario via 192.168.68.1 oppure via 192.168.66.1

```

RE#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route

Gateway of last resort is not set

R 192.168.0.0/24 [120/2] via 192.168.66.1, 00:00:03, Serial0/0/1
R 192.168.3.0/24 [120/1] via 192.168.68.1, 00:00:02, Serial0/1/0
C 192.168.4.0/24 is directly connected, FastEthernet0/0
R 192.168.5.0/24 [120/1] via 192.168.70.2, 00:00:03, Serial0/0/0
R 192.168.64.0/24 [120/1] via 192.168.66.1, 00:00:03, Serial0/0/1
R 192.168.65.0/24 [120/2] via 192.168.68.1, 00:00:02, Serial0/1/0
               [120/2] via 192.168.66.1, 00:00:03, Serial0/0/1
C 192.168.66.0/24 is directly connected, Serial0/0/1
R 192.168.67.0/24 [120/1] via 192.168.68.1, 00:00:02, Serial0/1/0
C 192.168.68.0/24 is directly connected, Serial0/1/0
R 192.168.69.0/24 [120/1] via 192.168.70.2, 00:00:03, Serial0/0/0
               [120/1] via 192.168.68.1, 00:00:02, Serial0/1/0
C 192.168.70.0/24 is directly connected, Serial0/0/0

```

### **Tipologia di Righe della Tabella di Routing**

Una riga può essere di 3 categorie:

- C - Rete direttamente connessa
- S - Route statica, l'ho detta io al router, e lui non sa perché c'è questa riga
- R - Route dinamica.

Il router in funzione del tempo acquisisce informazioni in seguito a cambiamenti dello stato della rete, dato che la rete è dinamica i percorsi possono variare per congestione o rottura. La C come lettera indica le reti direttamente connesse, la S indica una riga statica, non è una rete a cui è direttamente connesso, quindi o glielo dico io (staticamente), oppure la può ricavare dinamicamente attraverso un protocollo di routing, quindi nel caso del protocollo RIP che si usa per riempire la tabella dinamicamente avremo una R.

### **Metriche di routing**

La metrica è un modo per misurare la lunghezza di un percorso, voglio decidere il percorso con la metrica migliore (costo minore), la natura della metrica può essere diversa, posso misurare un percorso in base alla rete ed alla sua topologia, ad esempio il numero di hop (utilizzato da RIP), oppure potrei misurare un percorso sulla base della capacità dei link attraversati.

Questi metodi sono statici dipendono dalla struttura della rete, la capacità del link che sia congestionato o vuoto, è la stessa, il numero di hop basta conoscere la topologia; quindi, non serve conoscere il livello di utilizzo della rete.

Ci sono invece altre metriche che dipendono dalla congestione della rete, la quale a sua volta dipende dal tempo, e questo è difficile per il routing perché bisogna trovare un risultato che converga e garantire la consistenza delle informazioni che si conoscono.

- RIP utilizza hop-count.
- OSPF utilizza il concetto di costo.
- EIGRP è un protocollo di routing proprietario di Cisco, non si sa come funziona, però funziona molto bene.

Ritorniamo al packet tracer.

Nella tabella di routing nelle varie righe ho [120/1], il secondo numero indica il numero di router che il pacchetto ALMENO deve attraversare per arrivare a destinazione (quindi il costo minimo).

Nelle righe con codice C non c'è una metrica (il secondo numero tra le quadre), dato che la rete è direttamente connessa (praticamente è come se fosse 0).

### Distanza Amministrativa

Se avessimo aggiunto però la metrica staticamente?

Il router come fa a capire se quello che ha imparato lui dinamicamente o quello che ha appreso staticamente vale di più?

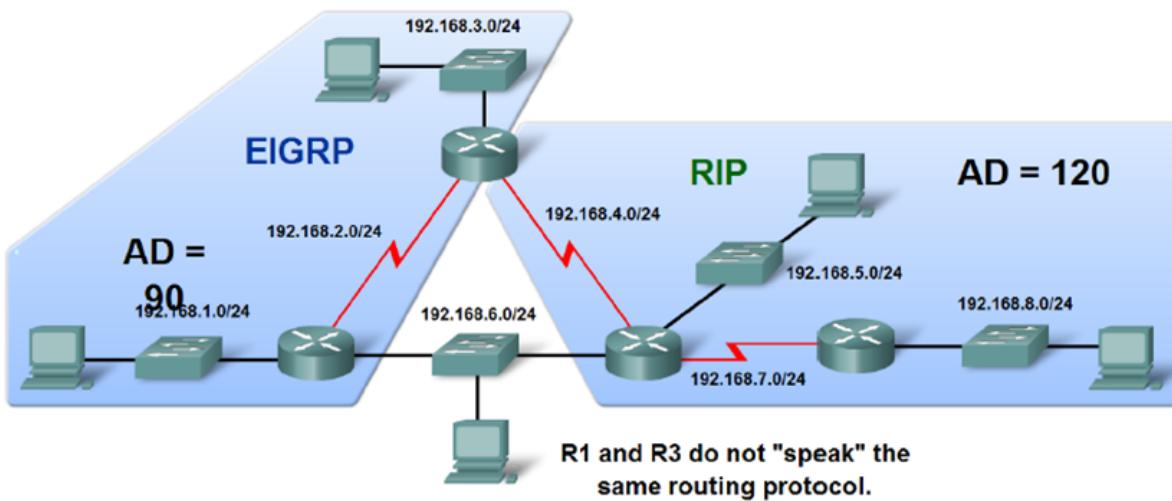
C'è la distanza amministrativa, numero intero da 0 a 255.

Più è piccolo più è considerato preferibile e misura la priorità delle sorgenti, quindi per ogni possibile codice (R, C, S...), c'è associata una distanza amministrativa in modo predefinito, quindi il confronto tra due alternative si fa su due livelli, prima si guarda la distanza amministrativa, vince chi ha minore distanza, a parità di distanza allora vado a vedere.

Ad esempio BGP che regola lo scambio al di fuori del mio *autonomous-system* deve valere di meno, deve prevalere una soluzione che mi fa rimanere all'interno della rete, tra RIP ed OSPF vince OSPF essendo più preciso.

Route Source	Administrative Distance
Connected	0
Static	1
EIGRP summary route	5
External BGP	20
Internal EIGRP	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
External EIGRP	170
Internal BGP	200

### Comparing Administrative Distances



La rete a destra utilizza RIP, tutti i pacchetti RIP in questa rete non li manda all'altra rete, la rete di sinistra utilizza EIGRP che conta di più rispetto a RIP, quindi il router di confine se vuole raggiungere il PC in basso, nonostante il percorso valga uguale per raggiungerlo (1 hop), utilizzerà il collegamento a sinistra che utilizza EIGRP dato che conta di più (ha una AD minore) di RIP.

### Parità di AD - Load Balancing

Se ho questo caso ho due percorsi che hanno lo stesso costo [120/2], se penso alla destinazione in quanto host non è vero che sono equivalenti, io non ho questo dettaglio di scelta, ho solo il prefisso di rete, in questa situazione il router utilizza entrambi i percorsi, fa un load balancing, ovvero bilanciamento del carico dei percorsi, questo load balancing può avere al massimo 4 percorsi, il router a turno utilizza uno dei due percorsi (due in questo esempio).

```
R 192.168.65.0/24 [120/2] via 192.168.68.1, 00:00:02, Serial0/1/0
[120/2] via 192.168.66.1, 00:00:03, Serial0/0/1
```

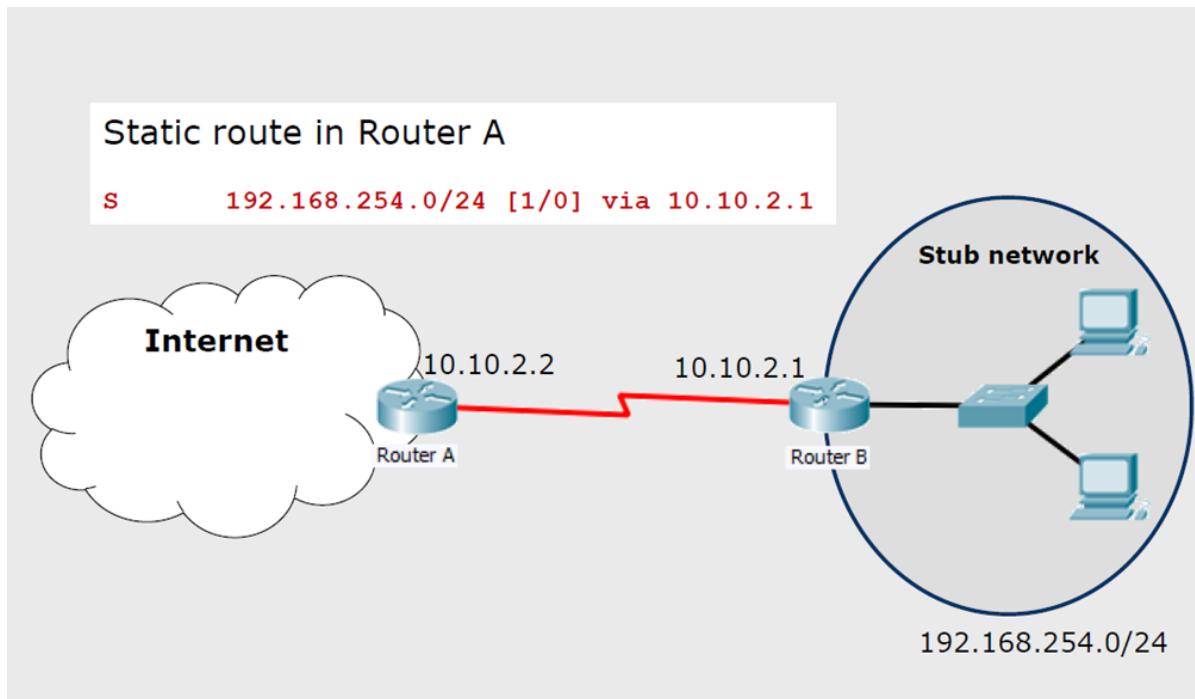
Il comando `show ip route`, ha anche un parametro per filtrare in modo da vedere solo alcune reti, ad esempio, `show ip route rip`, mostra solo le righe R.

### Come viene applicato il protocollo di Routing

1. Il router legge l'IP di destinazione del pacchetto che arriva.
2. Se l'IP coincide con una delle sue interfacce il pacchetto è già arrivato non deve essere trasmesso.
3. Se la destinazione invece è in una rete direttamente connessa, diventa un problema di livello 2.
4. Se la destinazione è remota devo trovare una riga che fa match nella tabella di routing.
  - Se ci sono più righe scegliere la più conveniente.

- Se non trova nessuna corrispondenza nella tabella, il pacchetto viene scartato, il router in questo caso informa il mittente con un pacchetto ICMP, detto destination unreachable.

## Routing statico

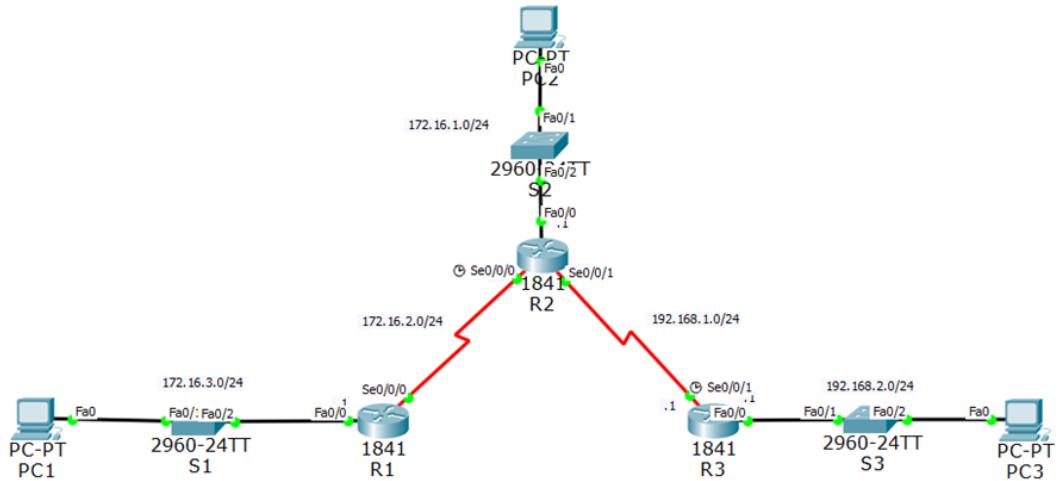


Abbiamo la rete alla quale è collegata la stub network, ha un indirizzo e ha un solo punto di ingresso/uscita, devo andare da router A al router B, o viceversa, quindi si aggiunge una riga statica in rosso, indica che per qualunque pacchetto che arriva ad A, il next-hop è il router B. Per configurare una route c'è il comando:

```
ip route network-address subnet-mask {ip-address | exit-interface} [Distance]
```

Il terzo parametro è l'indirizzo ip oppure l'interfaccia di uscita.

L'ultimo parametro è opzionale, indica la distanza amministrativa, di default vale 1, se vogliamo cambiarla si specifica.



Andiamo su packet tracer, vogliamo mandare dei pacchetti da PC2 a PC1 in questa rete:  
 Con il comando `R1#debug ip routing`, si abilitano messaggi di notifica che vengono mostrati nell'interfaccia a riga di comando in corrispondenza di eventi, in questo caso si mostrano informazioni riguardo la tabella di routing, adesso scriviamo:

```
R1(config)#ip route 172.16.1.0 255.255.255.0 172.16.2.2
```

In questo caso ho messo un indirizzo di una rete direttamente connesso, non è detto che debba essere direttamente connesso, in caso contrario il router fa le sue ricerche.  
 Visualizziamo la tabella di routing adesso:

```
R1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B
      - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS
      inter area
      * - candidate default, U - per-user static route, o - ODR
      P - periodic downloaded static route

Gateway of last resort is not set
```

```
172.16.0.0/24 is subnetted, 3 subnets
S 172.16.1.0 [1/0] via 172.16.2.2
C 172.16.2.0 is directly connected, Serial0/0/0
C 172.16.3.0 is directly connected, FastEthernet0/0
```

Aggiungiamo una nuova riga, e ci mostra anche delle righe di debug:

```
R1(config)#ip route 192.168.2.0 255.255.255.0 172.16.2.2
R1(config)#RT: SET_LAST_RDB for 192.168.2.0/24
    //righe di debug

NEW  pdb: via 172.16.2.2

RT: add 192.168.2.0/24 via 172.16.2.2, static metric [1/0]

RT: NET-RED 192.168.2.0/24
```

Quando mostriamo la tabella di routing, in questo caso essendo la rete direttamente connessa, non viene mostrata l'interfaccia relativa.

Una riga statica disorienta il router, poiché non gli dà la cognizione della distanza.

Avendo fatto questa configurazione su Router1 ora devo configurare ANCHE il Router2.

A casa fare l'esercizio 3.2

Vado nel modo di configurazione dell'interfaccia serial0/0/0:

```
R1(config)#interface serial 0/0/0
```

La spengiamo:

```
R1(config-if)#shutdown
```

```
R1(config-if)#
%LINK-5-CHANGED: Interface Serial0/0/0, changed state to
administratively down

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0/0/0,
changed state to down

RT: interface Serial0/0/0 removed from routing table
RT: del 172.16.2.0 via 0.0.0.0, connected metric [0/0]

RT: delete network route to 172.16.2.0

RT: NET-RED 172.16.2.0/24

RT: del 172.16.1.0 via 172.16.2.2, static metric [1/0]

RT: delete network route to 172.16.1.0

RT: NET-RED 172.16.1.0/24

RT: del 192.168.2.0 via 172.16.2.2, static metric [1/0]

RT: delete network route to 192.168.2.0
```

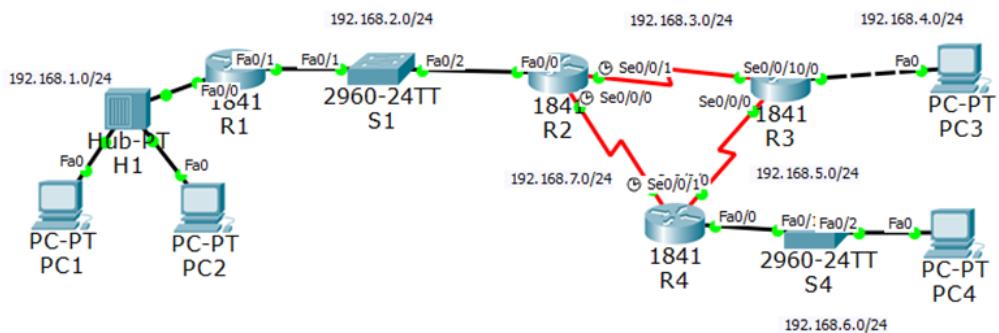
RT: NET-RED 192.168.2.0/24

Il router mantiene il contenuto della tabella di routing aggiornato in base alle informazioni che ha, andiamo nel file di configurazione e vediamo i comandi che abbiamo inserito, il routing ha rimosso qualunque riga in cui c'era scritto la seriale 0/0/0, anche quelle in cui c'era l'indirizzo IP.

Se faccio un ping, non funziona perché il router non sa cosa farci con quel pacchetto (caso citato in precedenza), se riaccendiamo l'interfaccia, la tabella di routing ritorna quella di prima, quindi con la modalità statica non si scrive direttamente nella tabella di routing del router, si scrive nel file di configurazione, quindi quando si riaccende l'interfaccia la configurazione viene ricaricata.

Quando si usa la modalità con l'interfaccia, questa va bene se l'interfaccia è seriale. se i router sono collegati con un collegamento ethernet, questo non va bene perché mandare un frame di livello 2 su una porta seriale non richiede di conoscere l'indirizzo MAC (di livello 2) del destinatario, invece nel caso di ethernet devo sapere l'indirizzo MAC e per saperlo devo conoscere anche il relativo IP perché con ARP mi consente di risolvere il relativo indirizzo MAC di destinazione.

Esercizio 3.2:



Se in PC1 si fa un ping in PC3 non funziona, allora andiamo su R1 è visualizziamo la routing table:

```
R1#show ip route
C 192.168.1.0/24 is directly connected, FastEthernet0/0
C 192.168.2.0/24 is directly connected, FastEthernet0/1
S 192.168.3.0/24 is directly connected, FastEthernet0/1
S 192.168.5.0/24 is directly connected, FastEthernet0/1
S 192.168.6.0/24 [1/0] via 192.168.2.2
S 192.168.7.0/24 [1/0] via 192.168.2.2
```

Nella tabella non c'è nessuna informazione riguardo la rete 192.168.4.0/24, aggiungiamo una route statica, e mettiamo come next-hop l'indirizzo dell'interfaccia FastEthernet 0/0 del router:

```
R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#ip route 192.168.4.0 255.255.255.0 192.168.2.2
Adesso vediamo se la riga è stata aggiunta nella tabella:
R1(config)#exit
R1#
%SYS-5-CONFIG_I: Configured from console by console

R1#show ip route
C 192.168.1.0/24 is directly connected, FastEthernet0/0
C 192.168.2.0/24 is directly connected, FastEthernet0/1
S 192.168.3.0/24 is directly connected, FastEthernet0/1
S 192.168.4.0/24 [1/0] via 192.168.2.2
S 192.168.5.0/24 is directly connected, FastEthernet0/1
S 192.168.6.0/24 [1/0] via 192.168.2.2
S 192.168.7.0/24 [1/0] via 192.168.2.2
```

Se facciamo il ping adesso non funziona ancora, andiamo in modalità simulazione di Cisco: Abilitiamo solo ICMP, facciamo il ping da PC1 a PC3, vediamo che il pacchetto da R1 va ad R2, da R2 ad R3, R3 inoltra il pacchetto a PC3, PC3 invia il pacchetto di echo reply, ad R3, quest'ultimo scarta il pacchetto.

Questo perché R3 non sa a chi mandarlo.

Quindi facciamo la stessa cosa ma su R3.

Andiamo in R3 e controlliamo la tabella di routing:

```
R3#show ip route
R 192.168.2.0/24 [120/1] via 192.168.3.1, 00:00:21, Serial0/0/1
C 192.168.3.0/24 is directly connected, Serial0/0/1
C 192.168.4.0/24 is directly connected, FastEthernet0/0
C 192.168.5.0/24 is directly connected, Serial0/0/0
R 192.168.6.0/24 [120/1] via 192.168.5.2, 00:00:15, Serial0/0/0
R 192.168.7.0/24 [120/1] via 192.168.3.1, 00:00:21, Serial0/0/1
[120/1] via 192.168.5.2, 00:00:15, Serial0/0/0
```

Notiamo l'assenza di una corrispondenza con la rete 192.168.1.0/24, inseriamo questa corrispondenza nella tabella di routing:

```
R3#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)#ip route 192.168.1.0 255.255.255.0 Serial 0/0/1
```

Come ultimo parametro abbiamo l'interfaccia da cui facciamo uscire il pacchetto, verso R2, in questo modo diminuisce il numero di hop, visualizziamo la tabella di routing e notiamo la presenza di una riga con l'indirizzo 192.168.1.0/24 inserito staticamente:

```

R3(config)#exit
R3#
%SYS-5-CONFIG_I: Configured from console by console

R3#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B
      - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS
      inter area
      * - candidate default, U - per-user static route, o - ODR
      P - periodic downloaded static route

```

Gateway of last resort is not set

```

S 192.168.1.0/24 is directly connected, Serial0/0/1
R 192.168.2.0/24 [120/1] via 192.168.3.1, 00:00:27, Serial0/0/1
C 192.168.3.0/24 is directly connected, Serial0/0/1
C 192.168.4.0/24 is directly connected, FastEthernet0/0
C 192.168.5.0/24 is directly connected, Serial0/0/0
R 192.168.6.0/24 [120/1] via 192.168.5.2, 00:00:14, Serial0/0/0
R 192.168.7.0/24 [120/1] via 192.168.3.1, 00:00:27, Serial0/0/1
[120/1] via 192.168.5.2, 00:00:14, Serial0/0/0

```

Adesso se facciamo il ping da PC1 a PC3 otteniamo echo reply, rifacciamo il ping anche con la modalità simulazione, creiamo un nuovo scenario, con new, inviamo un pacchetto ICMP da PC1 a PC3, adesso echo reply non viene scartato da R3, esso arriva a PC1.

## Lezione 16 Marzo

### Righe per Specifici Host

Nulla vieta che in una tabella di routing ci sia una riga in cui il prefisso dell'indirizzo di destinazione sia di 32 bit, ovvero, ci sia l'indirizzo di uno specifico host, questo è utile per indicare dei particolari host che in questo caso sono dei router stessi.

Possiamo vedere un indirizzo a 32 bit come una rete con un solo Host.

Il costo può essere misurato con diverse metriche, RIP utilizza il numero di hop, viene utilizzato anche il load balancing, ovvero se ho più strade con lo stesso costo i pacchetti vengono instradati in modo equo nelle due alternative di percorsi diverse fino ad un massimo di 4 alternative.

### Come viene usata la Distanza Amministrativa

Nel caso in cui ci sia da fare una scelta sulla strada da percorrere nel caso in cui le alternative hanno lo stesso costo entra in gioco la distanza amministrativa (AD), in questo caso un valore più piccolo della distanza indica maggiore priorità.

### **Porte di Tipo Broadcast**

Se il router ha una porta di uscita ethernet, la quale è broadcast, il router non può specificare solo la porta ethernet perché non può inviare a tutti un pacchetto, deve essere lui a scegliere a chi inviarlo, quindi in questo caso deve essere specificato l'IP del next-hop oltre all'interfaccia.

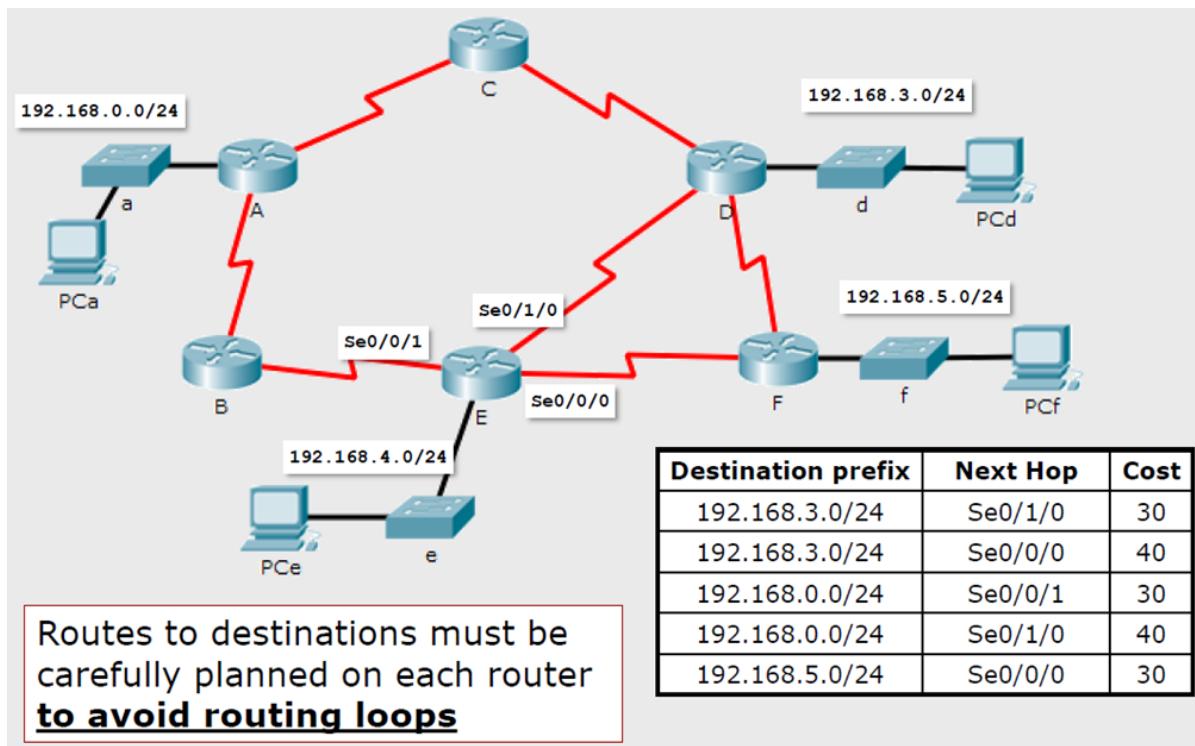
Il fatto che le righe siano statiche non è detto che non siano gestite dinamicamente, se una porta indicata in una riga statica venisse spenta, quindi non sarebbe più disponibile, l'informazione non rimane nella tabella, viene eliminata dinamicamente.

### **Vantaggi del Routing Statico**

Quando ha senso utilizzare una route statica? Abbiamo visto l'esempio della stub network, in generale però io posso utilizzare il routing statico per mantenere un controllo totale sulle strade da percorrere in modo da raggiungere la destinazione, il routing dinamico mantiene la connettività della rete in caso di variazioni/guasti della rete, io però vorrei poter rinunciare a questa affidabilità perché magari un guasto è prevedibile e gestibile in modo ordinario, quello che guadagno con il routing statico invece è un controllo completo, questo perché il routing dinamico sceglie la strada solo in funzione della minimizzazione del costo e non in base ad altri vincoli, quando decido io i percorsi posso mettere dei vincoli in più, dei quali i protocolli di routing non possono tenere conto, questo controllo dei flussi è detto *traffic engeneering*, (è come se una strada percorribile in entrambi i sensi la faccio diventare a senso unico), in pratica si forza il passaggio del traffico in determinati punti della rete dove ho messo delle funzioni di analisi, ad esempio un firewall.

### **Piano di Routing**

Io in questo caso devo progettare il piano di routing, ad esempio nella rete in figura prima di configurare i router devo fare un piano di routing, ovvero determinare il percorso end to end e fare in modo che sia consistente quando vado a configurare ciascun router, perché ogni router conosce solo il successivo router nel percorso e non tutti i router del percorso.



### Piano di Routing Corretto

Bisogna fare attenzione a non creare percorsi chiusi, ovvero avere dei pacchetti che circolano tra i router senza arrivare a destinazione finché il suo TTL non vale 0 e viene scartato.

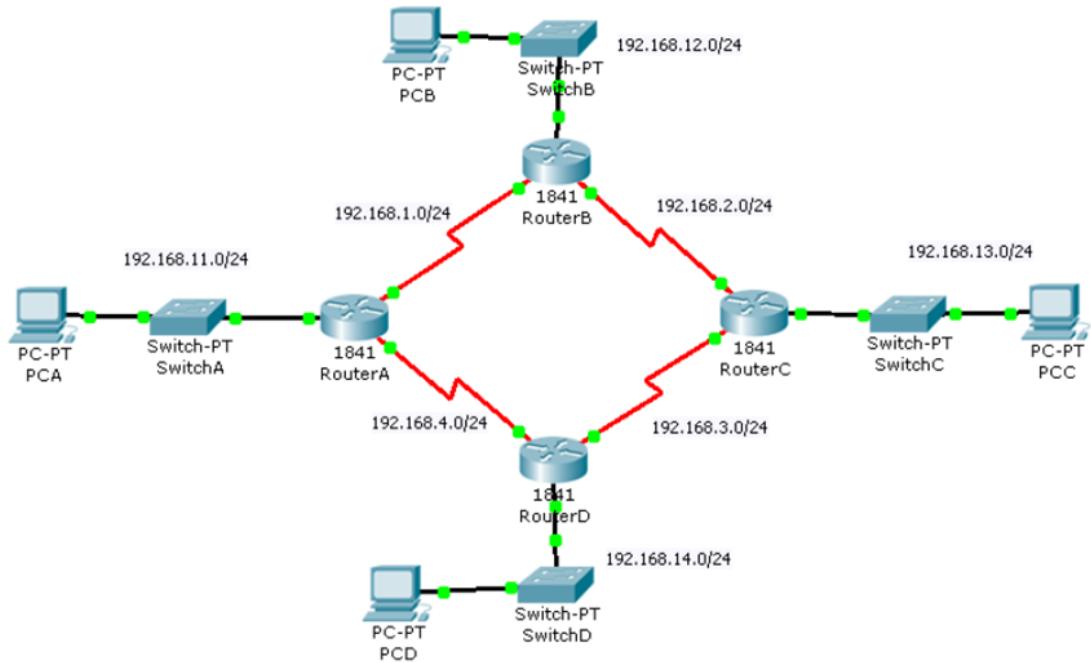
Per vedere se la configurazione è corretta bisogna vedere le loro tabelle di routing, ovvero bisogna avere next-hop diversi e le destinazioni che si sovrappongono.

### Determinazione dei Next-Hop

Dopo aver determinato le destinazioni (ossia le possibili destinazione che ogni host della rete può o non può raggiungere) bisogna determinare il next-hop in modo consistente.

Se sono nel router E, e devo instradare nelle altre 4 reti, nella prima (F) è direttamente connessa e non ci sono problemi, per 192.168.5.0/24 scelgo F, per 192.168.3.0/24 scelgo D, per 192.168.0.0/24 ho due alternative B oppure D, la scelta si fa in base al criterio che vogliamo ottimizzare, se il numero di hop, se il costo, queste scelte si configurano nei singoli router ma devono essere fatte in modo consistente e unitario.

Come si fa il piano di routing? Si può utilizzare un algoritmo che utilizzerebbe il routing dinamico, ad esempio Dijkstra e calcolare il percorso minimo in base a questo algoritmo.



Nella rete sopra, i router sono connessi in maniera circolare, quindi per raggiungere una rete ogni router ha due alternative, si può adottare una scelta di utilizzare il percorso di lunghezza minima, si può utilizzare una tabella come quella sotto che ha tante righe quante sono le destinazioni che devo configurare nel router (reti punto-punto e quelle da raggiungere).

Per ogni router della rete (colonna) e possibili Reti Destinazione (riga) indichiamo il next-hop.

Se la rete è direttamente connessa mettiamo i trattini (-), questa è la prima cosa che facciamo, dopo si riempie il resto della tabella.

Ad esempio in base al costo minimo, se sono in A voglio andare in 192.168.2.0/24, scelgo B, per ogni riga della tabella (rete), io vedo per ogni router (A, B, C, D), il next-hop, se guardando un riga c'è un trattino (-) non ho un percorso chiuso, se invece c'è una riga senza trattino ho un percorso chiuso come nella seconda tabella.

	A	B	C	D
192.168.1.0/24	—	—	B	A
192.168.2.0/24	B	—	—	C
192.168.3.0/24	D	C	—	—
192.168.4.0/24	—	A	D	—
192.168.11.0/24	—	A	B	A
192.168.12.0/24	B	—	B	A
192.168.13.0/24	B	C	—	C
192.168.14.0/24	D	A	D	—

	A	B	C	D
192.168.1.0/24	—	—	B	A
192.168.2.0/24	B	—	—	C
192.168.3.0/24	D	A	—	—
192.168.4.0/24	—	A	D	—
192.168.11.0/24	—	A	B	A
192.168.12.0/24	B	—	B	A
192.168.13.0/24	B	A	—	C
192.168.14.0/24	D	A	D	—

### Indirizzamento IPv4

Finora abbiamo visto che gli IP sono stati già inseriti nella rete, in realtà quando abbiamo a che fare con una rete, l'assegnazione degli indirizzi fa parte della progettazione della rete, devo decidere come gestire il pacchetto di indirizzi IP che mi è stato dato, bisogna stabilire un piano di configurazione della rete, gli indirizzi IPv4 sono numeri su 32 bit, i primi n bit identificano la sottorete a cui quell'host appartiene, originariamente questi n bit non erano a piacere, gli indirizzi erano classful, c'erano classi di prefissi di rete, le lunghezze potevano essere di 8, 16, 24, quindi le reti potevano avere rispettivamente  $2^{24}$  host,  $2^{16}$  host,  $2^8$  host.

La ricerca di una tabella di routing di un match con un prefisso di dimensione fissa era più facile, questa cosa è stata rilasciata perché il costo ovvero la forte inefficienza nell'allocazione aveva superato i vantaggi, perché quando alloco un indirizzo di rete ho bloccato  $2^{24}$  indirizzi,  $2^{16}$  indirizzi o  $2^8$  indirizzi, c'è un gap enorme tra reti di  $2^{16}$  e  $2^8$ , questo ha cambiato l'HW dei router, è stato introdotto il VLSM (Variable Length Subnet Mask), adesso con indirizzi classless devo dire anche la maschera (ad esempio /24), prima non serviva perché il numero di host dipendeva dalla classe della rete, nel routing un indirizzo del tipo a.c.d.e/x, la x è determinante perché a parità di indirizzi viene scelto quello con la x più grande.

### Indirizzi IP pubblici e Privati

Gli indirizzi IP sono divisi in pubblici e privati, quelli privati sono utilizzati nelle nostre reti a casa, sono inutili su internet, l'ente IANA, organizzazione sovranazionale alla quale è stato assegnato il compito di allocazione degli indirizzi IP, se voglio utilizzare degli IP al di fuori del range che ho, devo avere da qualcuno il permesso di farlo.

L'indirizzo pubblico, a differenza di quelli privati, bisogna chiedere a qualcuno di poterlo utilizzare, invece gli indirizzi privati li posso utilizzare senza chiedere ma non posso utilizzarli per andare in internet.

È una scelta di progetto comune utilizzare indirizzi dentro i range nella tabella seguente, altrimenti se voglio andare in internet dovrei rifare la configurazione degli indirizzi, però se la nostra rete è isolata da internet io potrei utilizzare gli indirizzi che voglio.

Indirizzi speciali

Class	RFC 1918 Internal Address Range	CIDR Prefix
A	10.0.0.0 - 10.255.255.255	10.0.0.0/8
B	172.16.0.0 - 172.31.255.255	172.16.0.0/12
C	192.168.0.0 - 192.168.255.255	192.168.0.0/16

- 0.0.0.0/8 è riservata e non si può allocare per un host;
- 127.0.0.1/8 è la rete di loopback, ovvero una rete virtuale che un host considera come direttamente connessa;
- 169.254.0.0/16 è il blocco link local, se non ho un servizio DHCP e non posso configurare gli host a mano si identifica un prefisso noto a tutti e quindi sono in grado di comunicare all'interno della mia rete, un host che si connette ad una rete se fa una richiesta DHCP, se dopo un po' non ottiene l'indirizzo, l'host si assegna un indirizzo in questo blocco, e nella parte di host (suffisso) utilizza un numero casuale da 0 a  $2^{16}$ , successivamente deve assicurarsi che nessun'altro abbia quell'indirizzo, in questo modo la rete si configura da sola con indirizzi con questo prefisso;
- 192.0.2.0/24, indirizzi utilizzati nei documenti RFC per gli esempi, sono stati scelti in modo che questi indirizzi non siano di qualcuno, ad esempio per evitare che negli esempi vengano utilizzati gli indirizzi di Google.

Con IPv4 il modo che abbiamo per documentare informazioni riguardanti gli indirizzi è la notazione decimale puntata, si scrivono 4 decimali separati da un punto, finché il prefisso era 16/24/8, questa notazione era perfetta, era allineata alle classi, quando questa /x è diventata variabile è in po' meno intuitivo capire l'indirizzamento.

Nell'esempio ho una rete /25, degli ultimi 8 bit il primo fa parte del prefisso di rete, i restanti 7 sono gli indirizzi per gli host, di questi 7, tutti 0 è riservato, tutti 1 identifica l'indirizzo broadcast locale di quel link, quindi non posso assegnarlo; quindi, tolti questi due posso assegnare tutte le combinazioni in mezzo ( $2^7 - 2$ ), quindi in decimale ho numeri che vanno da 1 a 126.

## Divisione in Blocchi - Subnetting

Io assegno un blocco di indirizzi all'amministratore della rete e lui deve suddividerlo (subnetting) in sotto-blocchi, in modo da avere un prefisso per ciascuna sottorete possibilmente della dimensione corrispondente al numero di host che si prevede di avere in questa rete, il problema duale è quello di determinare l'indirizzo di sottorete più piccolo sufficiente per quel piano di indirizzamento.

Ad esempio ho un blocco d'indirizzi 192.168.1.0/24 e tre sottoreti.

Devo dividere il blocco di partenza in almeno 3 sotto-blocchi.

Ciascun sotto-blocco deve essere una potenza di 2, quindi non si posso fare  $256/3$  perché gli indirizzi base di ogni sottorete devono essere allineati con una potenza di 2, per garantire questa cosa basta che divido in un numero di blocchi che sia potenza di 2, in questo caso posso dividere per 4.

Come cambia la subnet mask con questa procedura?

Con /24 → **11111111.11111111.11111111.00000000**

Il blocco principale ha 24 bit destinati alla subnet, ossia un blocco da 256 indirizzi.

Devo allungare il prefisso (quindi i bit destinati alla subnet) per dividere in sottoblocchi il blocco principale.

Se mi sposto a /25 → **11111111.11111111.11111111.10000000**

Ottengo 2 sotto-blocchi da 128 indirizzi, sono sufficienti ma troppi.

Con /25 ho 1 bit che indicano la subnet all'interno del blocco principale.

Se mi sposto a /26 → **11111111.11111111.11111111.11000000**

Ottengo 4 sotto-blocchi da 64 indirizzi, sono sufficienti e non serve che vada oltre.

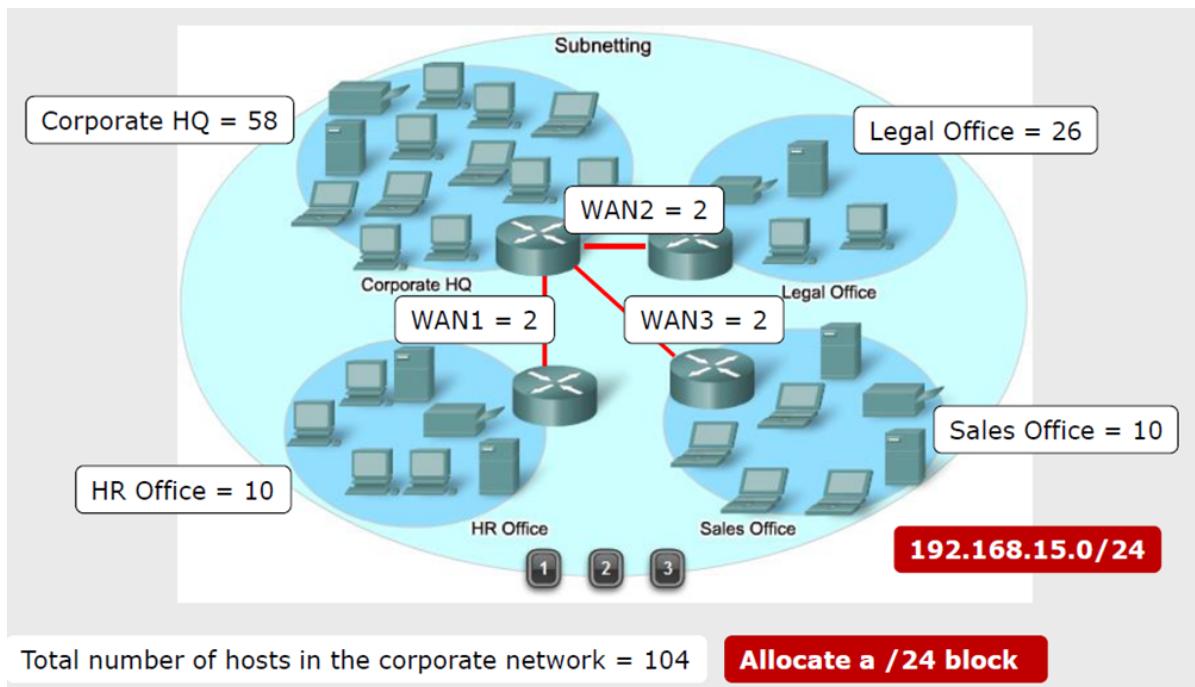
Con /26 ho 2 bit che indicano la subnet all'interno del blocco principale.

Con /26 ho 6 bit per indirizzare gli host di una subnet, quindi 64 indirizzi ciascuna.

Gli indirizzi effettivamente utilizzabili per gli host sono di meno:  $64 - 2 = 62$ .

“-2” perché togliamo il primo, quello che rappresenta la rete con numero di host 0 e l'ultimo indirizzo quello con tutti 1 che rappresenta l'indirizzo di broadcast).

**Esempio Subnetting:**



Total number of hosts in the corporate network = 104

**Allocate a /24 block**

Ho a disposizione 256 indirizzi con un indirizzo IP di partenza di 198.168.15.0 / 24. Nella Corporate Network devo avere 104 host, qual è il blocco di dimensione più piccola?

In generale mi basta un blocco da 128 indirizzi, quindi una subnet di tipo /25 (2<sup>7</sup> host).

Però dipende da come questi 104 host sono distribuiti (ossia dalla topologia della rete).

Se sono distribuiti in maniera uniforme mi servono sotto-blocchi uguali.

Se sono distribuiti in maniera diversa mi servono sotto-blocchi diversi impostati ad-hoc per la specifica situazione.

Adesso per ogni subnet vediamo quanti hos possiede e quindi il tipo di mask di cui ha bisogno.

1. La subnet HQ necessita di 58 host, quindi mi basta un blocco di /26 (*max 64 - 2 host*).
2. La subnet Legal Office necessita di 26 host, quindi mi basta un blocco di /27 (*max 32 - 2 host*).
3. La subnet HR Office necessita di 10 host, quindi mi basta un blocco di /28 (*max 16 - 2 host*).
4. La subnet Sales Office necessita di 10 host, quindi mi basta un blocco di /28 (*max 16 - 2 host*).
5. La subnet WAN1 necessita di 2 host, quindi mi basta un blocco di /30 (*max 4 - 2 host*).
6. La subnet WAN2 necessita di 2 host, quindi mi basta un blocco di /30 (*max 4 - 2 host*).
7. La subnet WAN3 necessita di 2 host, quindi mi basta un blocco di /30 (*max 4 - 2 host*).

Le reti WAN sono quelle che stanno “in mezzo” ai router, esse sono molto piccole (2 host ciascuna), quindi sono 3 blocchi di /30, ognuna delle quali “prende” 4 indirizzi IP.

Quindi in totale necessito di  $64 + 32 + 16 + 16 + 4 + 4 + 4 = 140$  indirizzi IP.

Quindi il blocco che mi serve è esattamente quello originale di /24, ovvero 256 host, poiché necessito di 140 indirizzi IP.

Come fare rapidamente questo esercizio.

Prima allochiamo il blocco più grande (da 64), poi quello da 32, dopo i due da 16 e infine i tre da 2.

Nome Subnet	N° Host Necessari	Blocco	N° Indirizzi	Indirizzo IP di Partenza
Corporate HQ	58	/26	64	192.168.15. <b>0</b> / 26
Legal Office	26	/27	32	192.168.15. <b>64</b> / 27
HR Office	10	/28	16	192.168.15. <b>96</b> / 28
Sales Office	10	/28	16	192.168.15. <b>112</b> / 28
WAN1	2	/30	4	192.168.15. <b>128</b> / 30
WAN2	2	/30	4	192.168.15. <b>132</b> / 30
WAN3	2	/30	4	192.168.15. <b>136</b> / 30

WAN3 non sfiora .256, quindi la soluzione è corretta.

.0				.0 (.1- .6)	.0 (.1- .2)
.4				.4 (.5- .6)	
.8				.8 (.9- .10)	
.12				.12 (.13- .14)	
.16				.16 (.17- .22)	.16 (.17- .18)
.20				.20 (.21- .22)	
.24				.24 (.25- .26)	
.28				.28 (.29- .30)	
.32				.32 (.33- .34)	
.36				.36 (.37- .38)	
.40				.40 (.41- .42)	
.44				.44 (.45- .46)	
.48				.48 (.49- .50)	
.52				.52 (.53- .54)	
.56				.56 (.57- .58)	
.60				.60 (.61- .62)	
.64				.64 (.65- .66)	
.68				.68 (.69- .70)	
.72				.72 (.73- .74)	
.76				.76 (.77- .78)	
.80				.80 (.81- .82)	
.84				.84 (.85- .86)	
.88				.88 (.89- .90)	
.92				.92 (.93- .94)	
.96				.96 (.97- .98)	
.100				.100 (.101- .102)	
.104				.104 (.105- .106)	
.108				.108 (.109- .110)	
.112				.112 (.113- .114)	
.116				.116 (.117- .118)	
.120				.120 (.121- .122)	
.124				.124 (.125- .126)	
.128				.128 (.129- .130)	
.132				.132 (.133- .134)	
.136				.136 (.137- .138)	
.140				.140 (.141- .142)	
.144				.144 (.145- .146)	
.148				.148 (.149- .150)	
.152				.152 (.153- .154)	
.156				.156 (.157- .158)	
.160				.160 (.161- .162)	
.164				.164 (.165- .166)	
.168				.168 (.169- .170)	
.172				.172 (.173- .174)	
.176				.176 (.177- .178)	
.180				.180 (.181- .182)	
.184				.184 (.185- .186)	
.188				.188 (.189- .190)	
.192				.192 (.193- .194)	
.196				.196 (.197- .198)	
.200				.200 (.201- .202)	
.204				.204 (.205- .206)	
.208				.208 (.209- .210)	
.212				.212 (.213- .214)	
.216				.216 (.217- .218)	
.220				.220 (.221- .222)	

## Summary routes

Operazione inversa al subnetting, in certe situazioni può essere utile fare il summarization, ovvero aggregazione d'indirizzi, io parto da 3 blocchi che individuo come contigui e mi chiedo qual è l'unico prefisso che mi comprende quei blocchi, si guarda l'insieme di bit di prefisso che accomuna quegli indirizzi di rete, nella figura seguente, i primi 22 bit sono a comune, ovvero (255.255.252.0).

Se vado nel file lab03\static:

Se vado in R3, se devo fare la configurazione delle route statiche, non ci sono percorsi chiusi, quindi è facile configurarle, sono 3 righe nella tabella di routing per ciascuna delle reti, per tutte quelle 3 reti, il next-hop è lo stesso, se dovessi condensare in un'unica riga nella tabella di routing torna comodo, per farlo scriviamo il comando ip route 172.16.0.0 255.255.252.0 192.168.1.2, abbiamo messo /22, in questo modo il prefisso è lo stesso

quindi per ciascuna di queste tre reti (172.16.1.0 - 172.16.2.0 - 172.16.3.0) il pacchetto viene inviato verso l'interfaccia 192.168.1.2 di R2.

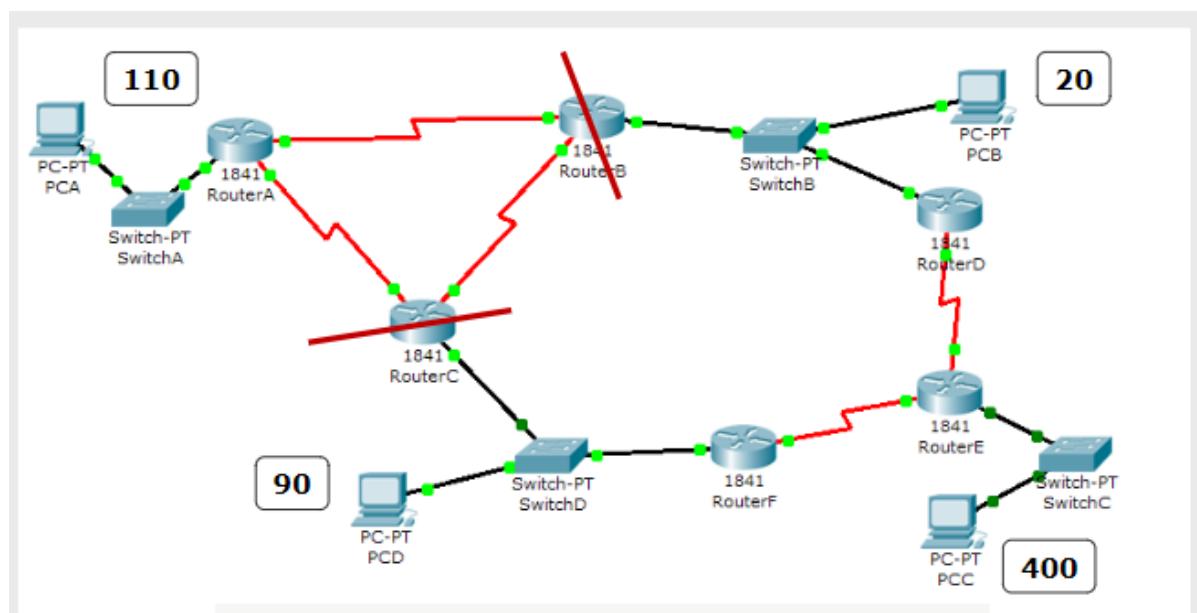
Nella summarization il percorso non deve essere per forza lo stesso, deve essere lo stesso il next-hop, le due tabelle di routing, quella con le 3 righe e quella summary sono equivalenti o no? Non necessariamente, se per equivalenti intendiamo che il comportamento di R3 è lo stesso indipendentemente dal pacchetto che gli arriva, allora non sono equivalenti, noi includiamo pacchetti che nel caso delle 3 righe non facevano match, adesso (caso summary) lo fanno, ad esempio pacchetti con destinazione 172.16.0.x, destinati alla rete 172.16.0.0, nel caso delle 3 righe il pacchetto viene scartato, nel caso del summary il router lo invia, e dopo non essendoci quella rete viene scartato, ho consumato risorse di rete in questo caso, la rete con il summary funziona correttamente, il modo però non è lo stesso introduco questa inefficienza.

Sapendo che posso fare il summary, esiste un modo per realizzare un piano d'indirizzamento più furbo? Esempio figura sotto, se alloco gli indirizzi per dimensione (tabella in alto), rende impossibile il summary perché dovrei mettere assieme A1 con A2 ed A3, non posso farlo perché ho il blocco B1 che lo separa, la prossima volta vediamo che è possibile non avere un indirizzamento che tiene conto del numero di host minimo per effettuare il summary.

### Esercizio 4.1

Il blocco di indirizzi è 172.10.0.0 / 22.

Ignora i trattini rossi sui router.



1. Fare un piano di indirizzamento tenendo conto dei numeri di host di ogni sottorete.
2. Fare la configurazione di base dei router e successivamente possiamo fare un piano di routing statico (soluzione nelle slide 3).

Quindi in totale gli host che devo allocare sono:

$$(400 + 1) + (110 + 1) + (90 + 2) + (20 + 2) + (5 * 2) = 636.$$

I +1 e i +2 sono inseriti per considerare le interfacce dei router all'interno della sottorete, che effettivamente corrispondono ad un host.

Quindi il blocco sarà almeno /22 → 1024 – 2 = 1022 host.  
Quindi sfrutto tutto il blocco a me dato.

Iniziamo a costruire la tabella per il piano d'indirizzamento (le reti si inseriscono in ordine di grandezza decrescente) che corrisponde alla soluzione dell'esercizio:

Subnet	N° Host Necessari	N° Indirizzi	Address	Mask	Dec. Mask	Assignable range	Broadcast
Lan C	400+1	512	172.16.0.0	/23	255.255.254.0	172.16.0.1 172.16.1.254	172.16.1.255
Lan A	110+1	128	172.16.2.0	/25	255.255.255.128	172.16.2.1 172.16.2.126	172.16.2.127
Lan D	90+2	128	172.16.2.128	/25	255.255.255.128	172.16.2.129 172.16.2.254	172.16.2.255
Lan B	20+2	32	172.16.3.0	/27	255.255.255.224	172.16.3.1 172.16.3.30	172.16.3.31
RA-RB	2	4	172.16.3.32	/30	255.255.255.252	172.16.3.33 172.16.3.34	172.16.3.35
RA-RC	2	4	172.16.3.36	/30	255.255.255.252	172.16.3.37 172.16.3.38	172.16.3.39
RB-RC	2	4	172.16.3.40	/30	255.255.255.252	172.16.3.41 172.16.3.42	172.16.3.43
RD-RE	2	4	172.16.3.44	/30	255.255.255.252	172.16.3.45 172.16.3.46	172.16.3.47
RE-RF	2	4	172.16.3.48	/30	255.255.255.252	172.16.3.49 172.16.3.50	172.16.3.51

Successivamente si assegnano gli indirizzi e si configurano i router.

Fatto questo si stabilisce il piano di routing con il quale si riempiono staticamente le tabelle di routing. Come prima cosa segniamo le reti direttamente connesse con un trattino (-).

Poi stabiliamo le route statiche facendo attenzione a non creare dei cicli (basta verificare che per qualsiasi pacchetto proveniente da una rete, ci sia un hop ad un router con -).

Network	RA	RB	RC	RD	RE	RF

<b>172.16.0.0/23</b>	RC	RD	RF	RE	-	RE
<b>172.16.2.0/25</b>	-	RA	RA	RB	RF	RC
<b>172.16.2.128/25</b>	RC	RC	-	RB	RF	-
<b>172.16.3.0/27</b>	RB	-	RB	-	RD	RC
<b>172.16.3.32/30</b>	-	-	RB	RB	RD	RC
<b>172.16.3.36/30</b>	-	RA	-	RB	RF	RC
<b>172.16.3.40/30</b>	RB	-	-	RB	RF	RC
<b>172.16.3.44/30</b>	RB	RD	RB	-	-	RC
<b>172.16.3.48/30</b>	RC	RC	RF	RB	-	-

Esercizio Terminato :D

### Lezione 20 Marzo 2023

Esiste una procedura che, dato un elenco di sottoreti avendo noti il numero di host al loro interno, mi permette di dividere efficacemente gli indirizzi IP del blocco a me dato.

#### Route Summarization

Metodo che si applica durante le definizione delle Route Statiche, serve per minimizzare il numero di righe della tabella di routing in una rete IP.

Meno righe ho nella tabella e più veloce risulta il router.

Con l'aumentare delle dimensioni della rete, aumenta anche il numero delle sottoreti elencate nella tabella di routing, così come la dimensione del pacchetto.

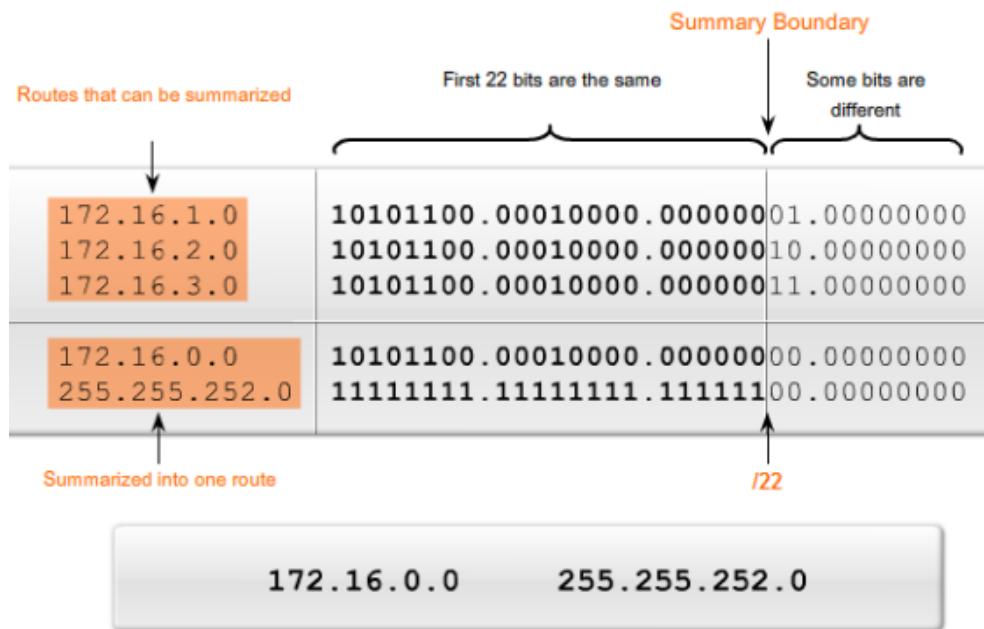
Le tabelle troppo grandi sono difficili da gestire e comportano ritardi.

Il metodo si può applicare solo se nella tabella "base" ho un insieme di indirizzi di rete contigui.

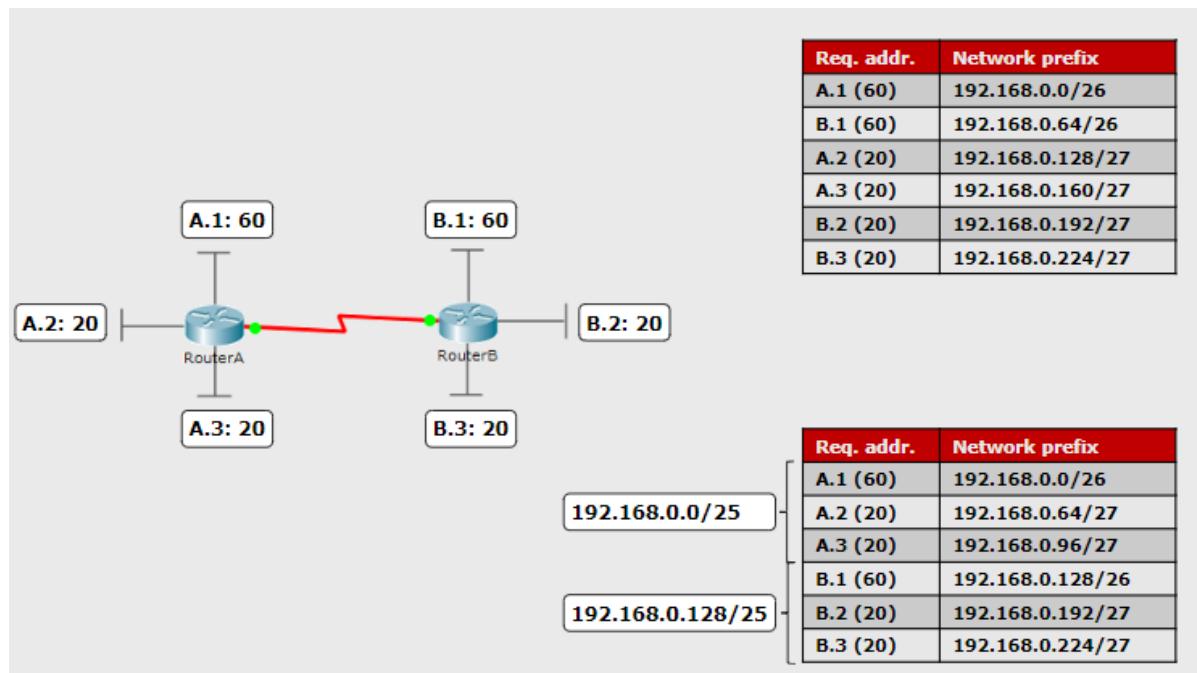
Il metodo mi permette di aggregare due o più righe della tabella in una sola.

Il "subnetting" (ossia dividere un blocco di indirizzi in sottoreti) è il processo inverso al Route Summarization, ossia prendo molteplici sottoreti e le tratto come se fossero un'unica grande rete.

Questo posso farlo se nella lista ho (nell'ordine che ho) delle reti con lo stesso "/x". E ovviamente conta anche l'aspetto topologico.



Il mio obiettivo è minimizzare il numero di indirizzi usati per le sottoreti. Quindi metto in ordine decrescente per numero di indirizzi necessari e alloco i blocchi.



In questo caso ho ridotto a 2 le righe nella tabella di routing, invece che a 6. Uno svantaggio evidente di questa tecnica si vede nella figura precedente a quella sopra, potrei permettere l'instradamento per indirizzi IP non effettivamente utilizzati.

RIP può supportare la route summarization anche in versione classless. OSPF invece non supporta questo tipo di metodo.

### Tabella di Routing - Riga di Default (Default Route)

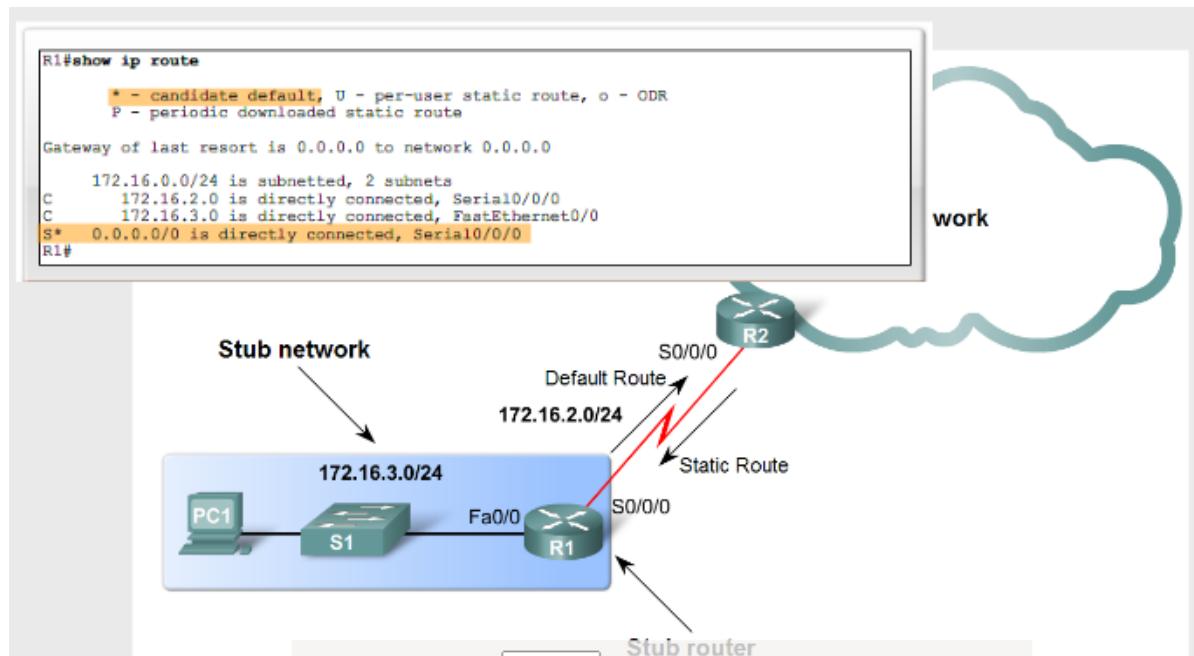
Quando si usa il routing dinamico, è normale che sia almeno una route statica.

La route punta alla destinazione `0.0.0.0/0`, il quale è l'indirizzo a cui si manda il pacchetto quando non fa match con nessuna riga della tabella di routing.

La riga di default fa sempre match con qualsiasi indirizzo di destinazione, poiché non devo confrontare nessun bit.

(Ricorda che la tabella di routing funziona con il match del numero di bit uguali, quindi con l'indirizzo più simile a quello target)

Se non ci fosse la regola di default, il pacchetto alla fine verrebbe scartato.



```
Router(config)#ip route 0.0.0.0 0.0.0.0 [exit-interface |ip-address]
```

### Protocolli di Routing - RIP

Usato in reti semplici e con pochi router.

Utile per far capire a ogni router la topologia della rete.

### Limiti del Routing Statico

Il routing statico perde di utilità quando la topologia della rete cambia oppure quando la rete è soggetta a fenomeni temporanei che causano un cambiamento di topologia anche per qualche istante.

Ossia quando un cavo viene tranciato oppure se un router oppure perché si guasta. L'hardware si guasta nel tempo oppure può funzionare male, ossia riceve il 50% dei pacchetti, ciò rende la rete instabile e fare troubleshooting diventa molto difficile.

Serve un sistema che sia capace di adattarsi alla situazione corrente, rendendo la rete più resiliente.

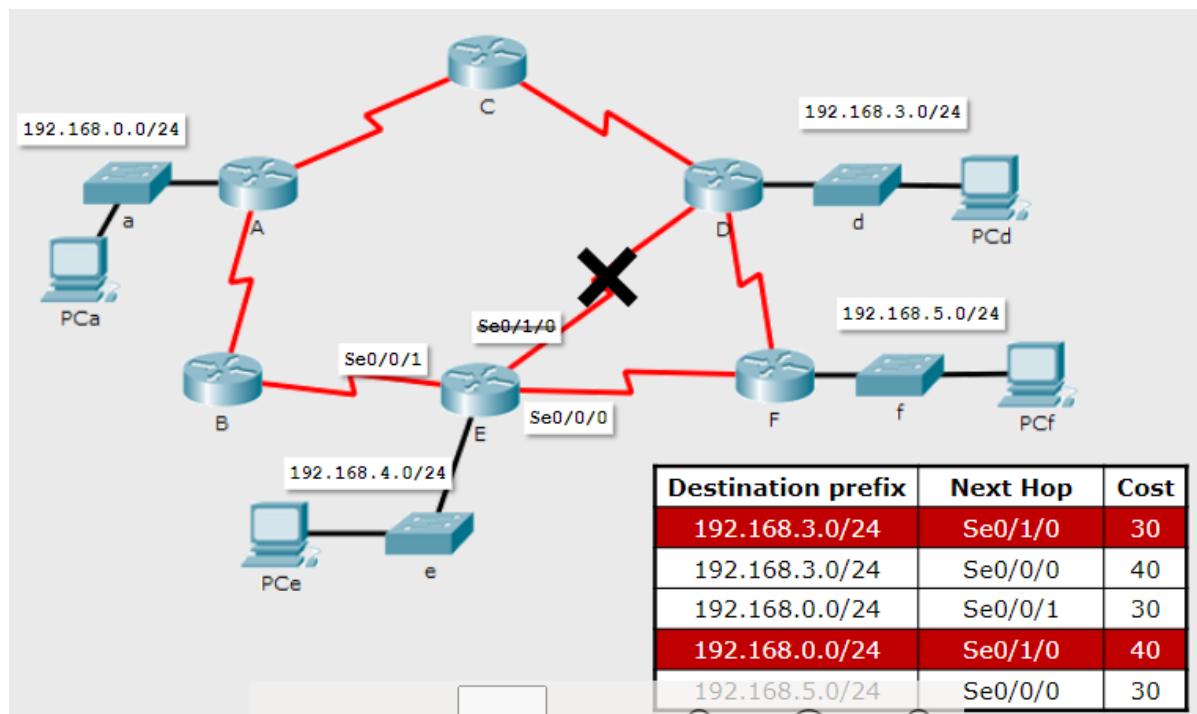
### Ridondanza

Metto più cose di quelle che servono (*2 router in più o 2 ink in più*) che mi permette di avere scelte multiple per una sola destinazione.

Oppure perché mi può convenire l'aumentare la capacità della rete e quindi poter spartire il traffico in più strade diverse.

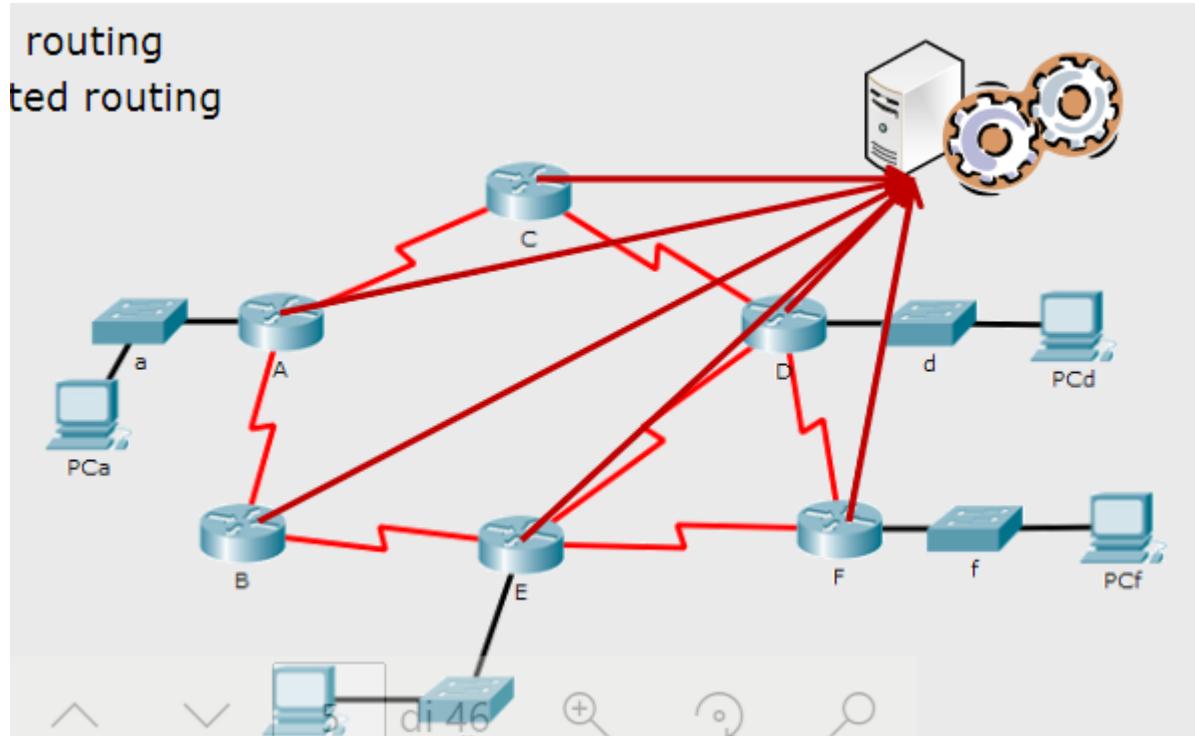
La ridondanza è gestibile anche con il routing statico, anche se in modo limitato e non può portare una soluzione definitiva, per i problemi descritti sopra.

Poiché il router E non saprà mai se il link AB è intatto o no, quindi il router E potrebbe comunque mandare i pacchetti verso B destinati ad A.



Il router si accorge (tramite il suo hardware) quando un link diventa inutilizzabile.

## Approccio Centralizzato



I router dicono ad un server quando i link diventano inutilizzabili e il server aggiorna le tabelle di routing.

Nel server gira un software che lavora sul grafo che tiene aggiornato.

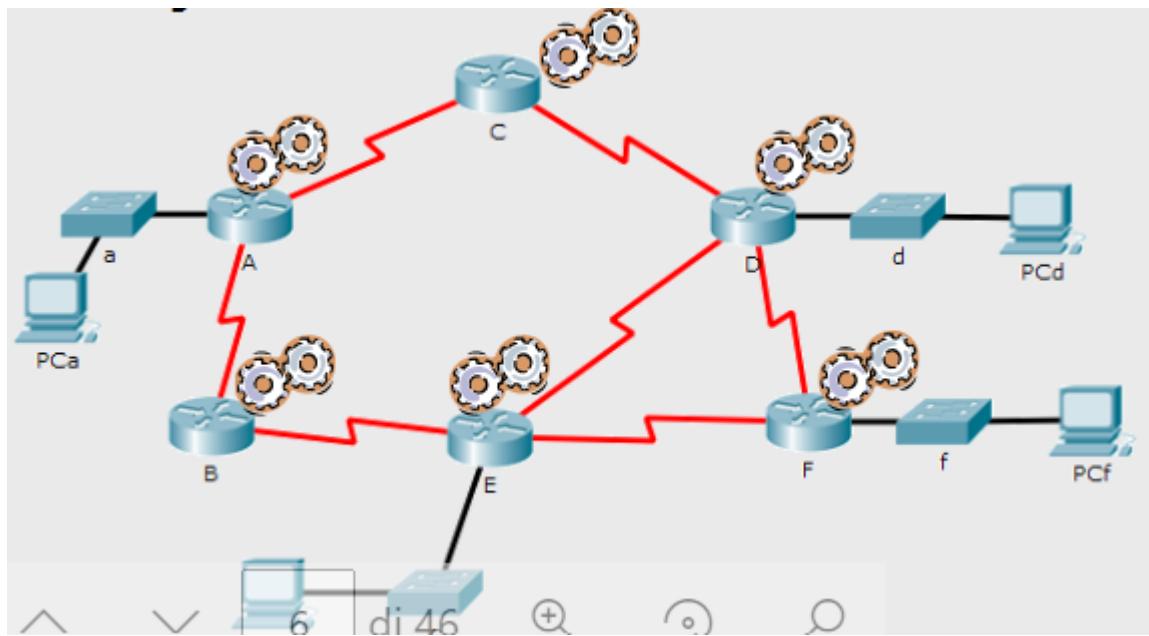
Questa soluzione, nonostante abbia vantaggi evidenti, ha il problema del single point of failure.

Poi queste cose sono state fatte durante la guerra fredda, dove tutto doveva essere distribuito...

Ora però si tende ad andare verso questa direzione con un approccio Software Defined Networking, dove la rete è controllata da un software chiamato “controller” che gira su un server.

SDN ora è adottata in reti di datacenter.

### **Routing Isolato**

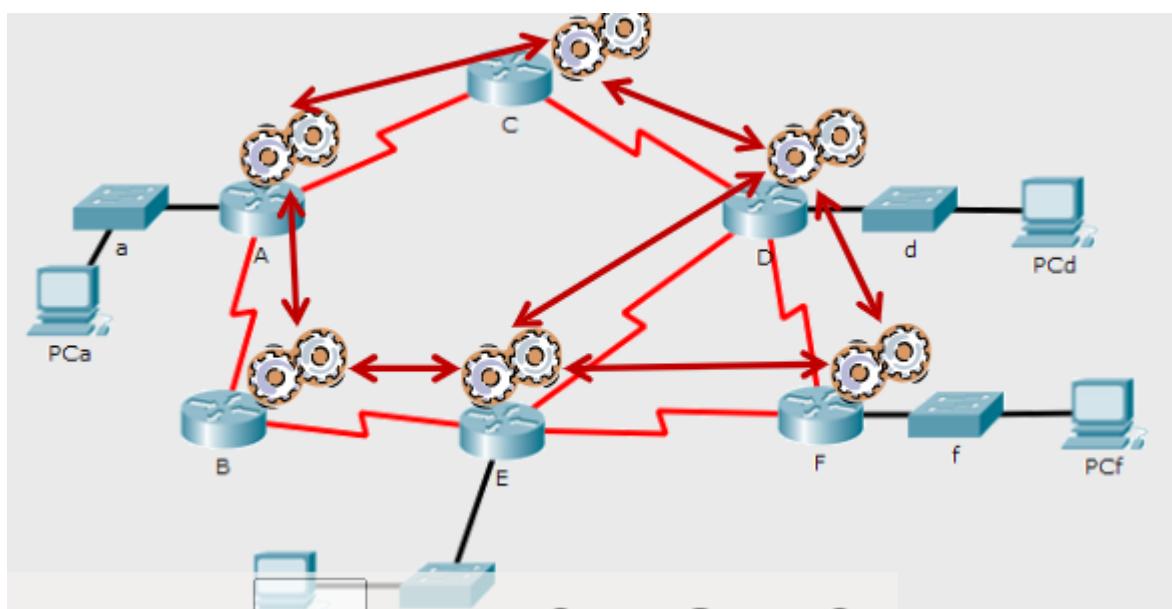


I router, senza scambiarsi i pacchetti tra di loro, compilano la loro tabella di routing.

Un esempio è il funzionamento della rete Ethernet, dove gli switch determinano indipendentemente da tutti gli altri il contenuto della sua tabella di switching.  
Ma ethernet non prevede i percorsi chiusi, quindi non va bene per le reti IP.

### Routing Distribuito

I router si scambiano informazioni tra di loro, e cooperando creano le loro tabelle di routing.



### RIP - Distance Vector

RIP si appoggia su l'algoritmo Distance-Vector, in RIP che informazioni vengono scambiate tra i router?

Vengono scambiati delle strutture chiamati Distance-Vector, il quale contiene parte della tabella di routing del router sender.

RIP si basa solo su informazioni topologiche.

I router ricevono il DV dai router vicini e trasformano la loro tabella di routing di conseguenza.

Prima o poi (forse, va dimostrato matematicamente) la rete si stabilizza e le tabelle di routing non cambiano più. (a meno di cambiamenti della topologia).

RIP funziona secondo la definizione di costo che gli diamo, di solito viene usato l'Hop Count.

- Direttamente connessa → 0.
- Non Raggiungibile → 16.

### Esempio di Distance Vector:

Il Sender crea il DV mettendo le info contenute nella sua tabella di routing.

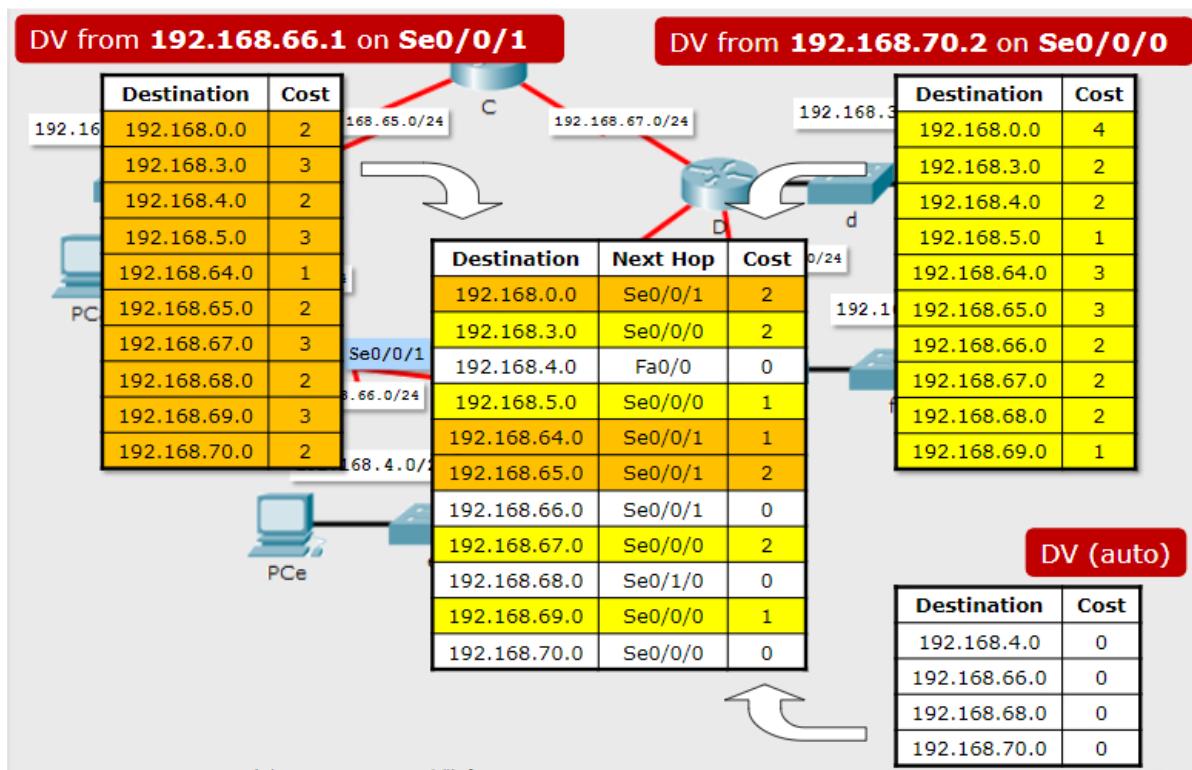
Prima di inviarlo ai suoi vicini incrementa di 1 ogni valore di Cost (aggiunge il fatto che il pacchetto dal punto di vista dei suoi vicini, deve attraversare se stesso).

Un DV è fatto in questo modo:

Destination	Cost
192.168.0.0	1
192.168.3.0	2
192.168.4.0	1
192.168.5.0	2
192.168.64.0	0
192.168.65.0	1
192.168.66.0	0
192.168.67.0	2
192.168.68.0	1
192.168.69.0	2
192.168.70.0	1

Il ricevitore, dopo aver ricevuto tutti i DV dai suoi vicini, fa un'operazione molto semplice chiamata "merge".

Per ogni rete destinazione prende la riga che costa di meno dai DV ricevuti.

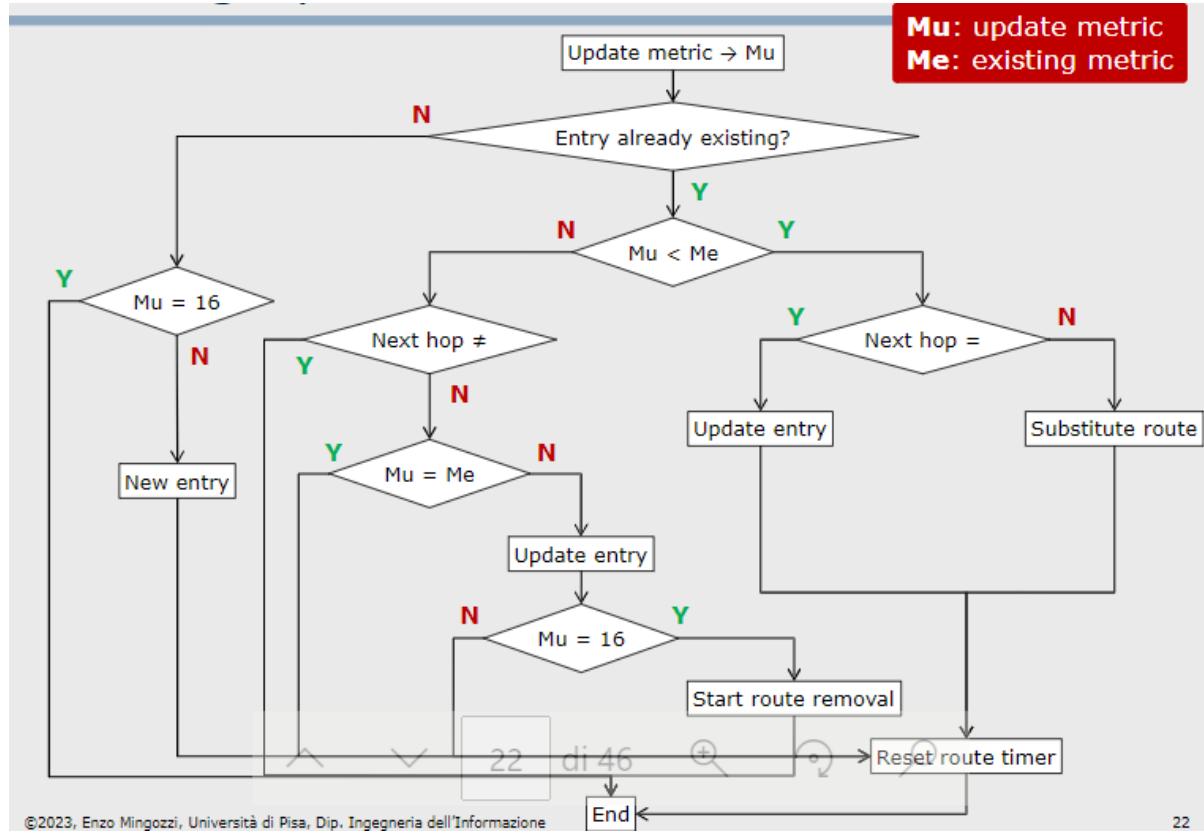


Questa cosa viene fatta periodicamente e i DV vengono mantenuti fino all'arrivo di quelli nuovi.

Quando un'interfaccia di un Router diventa inutilizzabile, la tabella di routing di quel router deve essere modificata.

La aggiorno eliminando tutte le info (quindi le righe relative al DV) che ho appreso dal router con cui ho perso il contatto.

### Algoritmo Distance Vector



Quando un link diventa a distanza 16, non è saggio rimuoverlo subito, ricordiamo che i link sono "Ballerini" e ci può stare che per qualche istante il link risulti spezzato.

Reagire in maniera troppo reattiva a situazioni di cambio di topologia, quando queste sono temporanee, può essere peggio che fare finta di niente mentre la rete non funziona in modo corretto in quell'intervallo di tempo, quindi se c'è un malfunzionamento aspetto a reagire nel piano dati, faccio finta di niente per un po' di tempo, invece nel piano di controllo mando le allerte, quindi mantengo separati i due livelli (dati e controllo).

Le interfacce dei router potrebbero anche disabilitarsi per qualche millisecondo.

Quando un cavo si rompe o un'interfaccia si disabilita, avviene un "cambio di topologia".

Da un link realmente rotto mi aspetto di non ricevere nulla al prossimo istante di update.

1. Se alla prima occasione non ricevo l'update, allora aspetto altri 180 secondi, quindi 6 update (*router timer*).
2. Se dopo il router timer, ossia dopo la sesta volta che quel router manca all'update dei DV dò per scontato che il router è morto e quindi imposto la distanza 16 verso quel router.
  - Quindi informo gli altri router che per me quel router è irraggiungibile.
3. Non cancello ancora la riga, la riga con distanza 16 la tengo per 120 secondi, quindi per 4 update (*Holddown Timer*).
4. Se durante lo *Holddown Timer* la situazione si ripristina allora cambia il valore nella riga della tabella e quindi si riaggiusta tutto.

- Allo scadere del *Holddown Timer*, se non è arrivato nulla, allora tolgo la riga dalla tabella
  - Cisco si prende altri 240 secondi per cancellarla, di norma viene cancellata dopo Holddown.

### **Posso modificare i Timer?**

I timer possono essere modificati, ma bisogna stare molto attenti, perché i loro valori non sono stati dati a caso.

A seconda della rete potrebbe convenire ridurli.

**MAI METTERE TIMER DIVERSI SU ROUTER DIVERSI**, altrimenti potrei avere situazioni imprevedibili tra cui anche attese infinite.

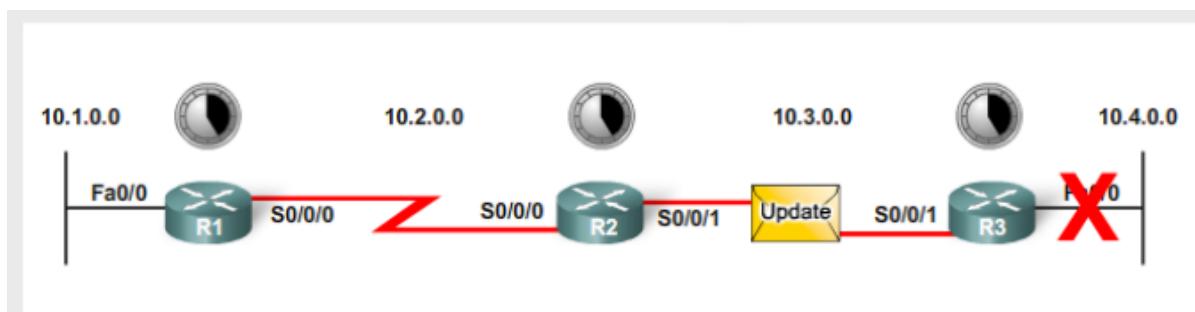
I router devono avere gli stessi timer.

Il route Timer serve per poter permettere la propagazione degli aggiornamenti delle route.

### **RIP - Triggered Update**

Supponiamo che subito dopo l'update si guasta un'interfaccia, il router non deve aspettare 30 secondi per dire ai vicini che per lui la distanza vale 16 (poiché nel frattempo i vicini potrebbero inviare messaggi e quindi sprecare tempo) ma esegue un Triggered Update e dice subito ai router vicini che tramite lui la distanza è 16.

Il Triggered Update è un update speciale, perché è fatto solo a seguito di un evento preciso (un cambio di topologia) ed è fatto (non del tutto il Distance Vector) ma solo della riga relativa alla porta guastata.



R3 dirà ai router vicini (quindi R2) che l'interfaccia a lui direttamente connessa è a distanza 16.

Il trigger update vale anche l'incontrario, quando la porta si risveglia, viene inviato un trigger update.

Per questo le righe non vengono tolte subito dalla tabella.

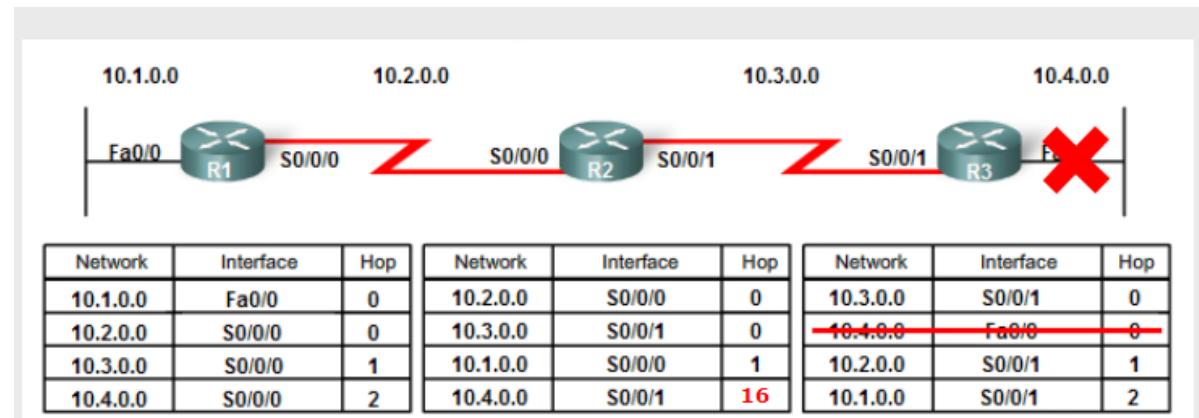
### **Perché si fa il Triggered Update?**

Se noi avessimo comunicazioni continue ed istantanee dei distance vector non avrei il problema seguente, però siccome devo aspettare un po' di tempo per inviare aggiornamenti

potrebbero esserci situazioni in cui certi aggiornamenti arrivano in ritardo e quindi vengano preceduti dagli aggiornamenti che dovrebbero seguirli.

Esempio in figura sopra, se sono su R3 e si rompe l'interfaccia Fa0/0, supponiamo di aver mandato un update, subito dopo la rete si guasta e oltre il periodico invio ogni 30 secondi, il router manda un update immediato (**triggered update**), ma non con tutto il distance vector, solo con l'informazione che è cambiata adesso, quindi R3 invia immediatamente ad R2 solo la riga del distance vector relativa alla rete guastata e come valore mette 16, in questo modo tutti imparano che quella destinazione non è raggiungibile.

### Cosa succede se NON si Invia il Triggered Update - “Count to Infinity”



1. Supponiamo che R3 non mandi il triggered update, ma che aspetti il regolare update ogni 30 secondi.
    - R3 invia un update ad R2 con scritto 16.
    - R1 invia un update ad R2 con scritto 3.
  2. R2 prenderà l'offerta di R1 e aggiorna la riga mettendo come distanza il valore 3.
    - R2 non è cosciente del fatto che in ogni caso dovrà passare da R3 poiché è l'unico direttamente connesso, e quindi R1 invierà ad R2 tutti i pacchetti destinati alla rete direttamente connessa ad R3.
    - Questo perché l'algoritmo DV non prevede lo scambio di questa informazione.
  3. Dopo 30 secondi R2 invia l'update ad R1, con distanza pari a  $4 = (3 + 1)$ .
  4. R1 lo riceve, vede che la riga è cambiata quindi la aggiorna mettendo come distanza il valore 4.
  5. Dopo 30 secondi R1 invia l'update ad R2, con distanza pari a  $5 = (4 + 1)$ .
  6. R2 lo riceve, vede che la riga è cambiata quindi la aggiorna mettendo come distanza il valore 5.
- ...

7. Dopo 30 secondi R2 invia l'update ad R1, con distanza pari a  $16 = (15 + 1)$ .
8. R1 lo riceve, vede che la riga è cambiata quindi la aggiorna mettendo come distanza il valore 16.
9. Dopo 30 secondi R1 invia l'update ad R2, con distanza pari a 16.
10. R2 lo riceve, vede che la riga è cambiata quindi la aggiorna mettendo come distanza il valore 16.

Ci si accorge che si va all'infinito (count to infinity) perché si va avanti così finché comando 1 non si arriva a 16.

Si spreca un sacco di tempo!

### **Split Horizon**

Tecnica per evitare il Count to Infinity.

Supponiamo di essere il router R1.

Se nella tabella di routing ho una riga con una certa destinazione (10.4.0.0 direttamente connessa a R3) e un certo Next Hop (in questo caso R2)

Split Horizon prevede che quando un router deve inviare i DV ai suoi vicini, esso NON RIMANDI ad un router R2 vicino le righe dove come next hop risulta esserci proprio R2 .

Ho lo svantaggio che la struttura dati che contiene il DV può avere una dimensione variabile.

Semplicemente Omette le righe "incriminate".

### **Split Horizon - Poisoned Reverse**

Funziona come il normale split horizon, ma con la differenza che nel DV non omette le righe che come next hop hanno il Router a cui invio il DV, ma ci mette un costo di 16.

In questo modo siamo sicuri che il router destinazione non aggiornerà la tabella di routing con quelle info.

Il vantaggio è che la struttura dati che contiene il DV continua a mantenere la stessa dimensione di prima e risolvo il problema senza effettuare grandi modifiche all'algoritmo.

R1 non vuole che R2 gli mandi pacchetti che come next hop hanno R2.

### **Note su RIP**

#### **RIP - Versioni**

Si divide in 2 versioni.

La prima è quella classica, con indirizzi IP Classful.

La seconda è più moderna, con indirizzi IP Classless.

In una rete, i router devono usare tutti la stessa versione di RIP, altrimenti potrei avere situazioni non prevedibili.

#### RIP - Tipo di Messaggi Scambiati

I messaggi di Update sono chiamati "Routing Updates" e vengono inviati periodicamente ogni 30 secondi.

Sono pacchetti di tipo UDP con porta 520.

#### RIP - Update Mancato

Se dopo 30 secondi non invia il DV, dò per scontato che il router sia morto e quindi eseguo la procedura di cancellazione dei suoi record.

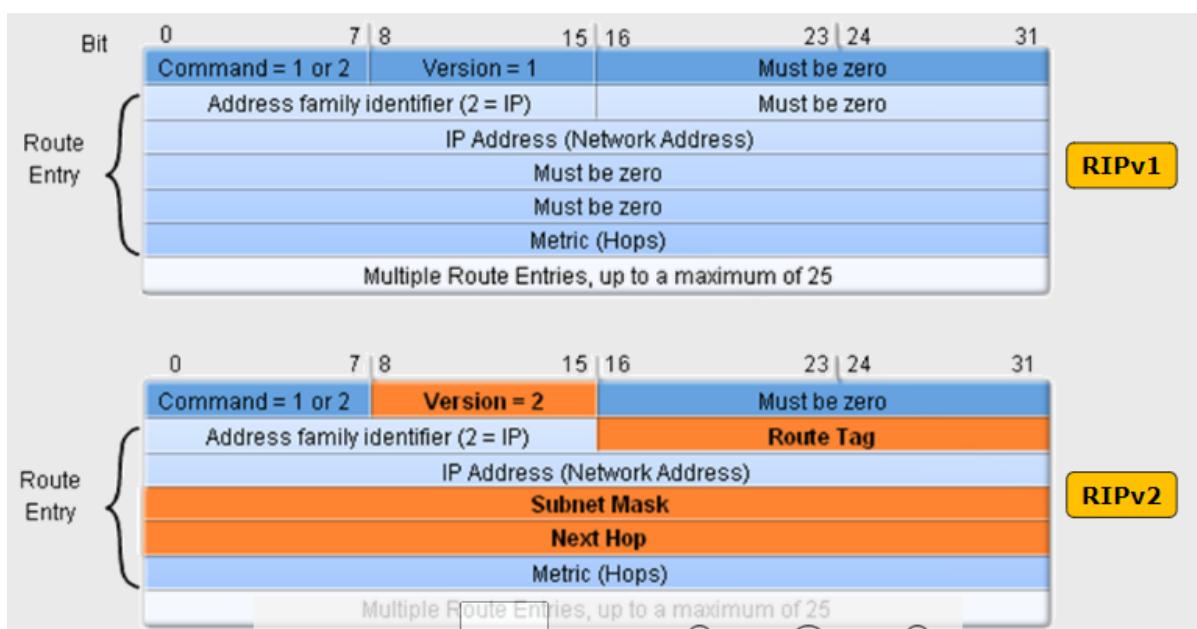
#### RIP - Costo massimo

Il costo massimo ammesso è 15.

#### RIP - Numero Massimo di Router

Quindi se ho 16 router, non posso usare RIP.

#### RIP - Formato dei Pacchetti



#### Problemi di Temporizzazione

Corse critiche → Eventi che portano il sistema in uno stato instabile.

Le reti sono un sistema, che percorrono una serie di stati e che quindi possono passare in uno stato instabile.

#### Tabella di Routing e Algoritmi di Routing

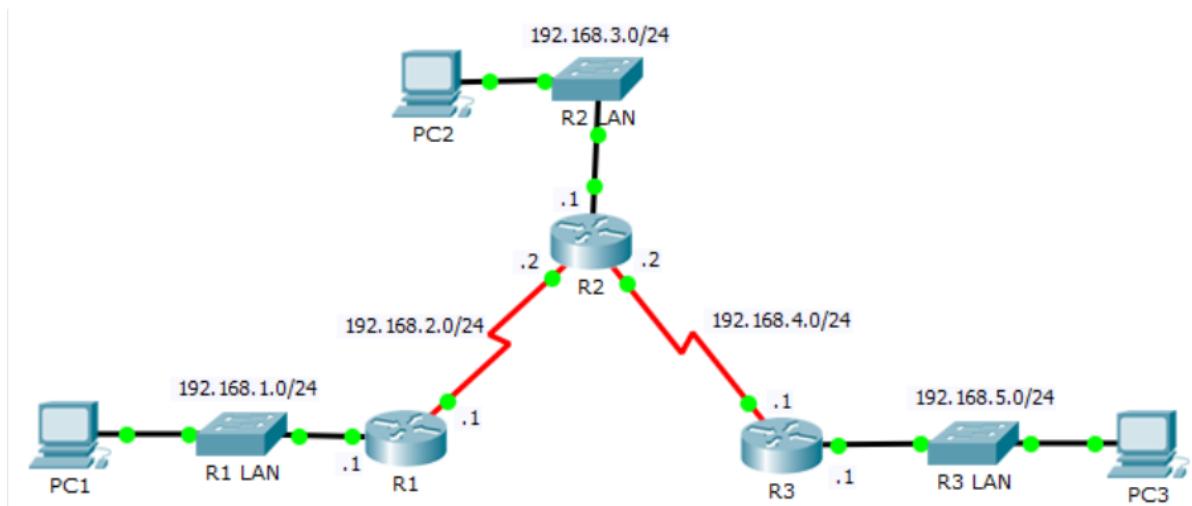
Un router può usare più protocolli di routing contemporaneamente, le informazioni dei protocolli vengono mantenute in un database e vengono poi utilizzate dal router.

La tabella di routing sarà un sunto delle informazioni contenute nel database di ogni algoritmo di routing che il router utilizza.

Il router in base alla distanza amministrativa decide il percorso da utilizzare, nel caso in cui ci siano diversi percorsi per la stessa destinazione anche ricavati con algoritmi diversi.

## Esercizio 5.1

Un router può usare più protocolli di routing contemporaneamente.



RIP è utile per capire la topologia della rete.

Sono io a decidere su quali porte deve usare RIP e per ogni porta quali reti annuncia.

Ad esempio io potrei non voler fare sapere a tutta la rete che una rete è raggiungibile mediante uno specifico router.

```
R1>enable // Entriamo in Modalità Privilegiata  
R1#conf t // Entriamo in Modalità di configurazione del router.
```

```
R1(config)#router ? //Algoritmi di Routing supportati dal Router.
```

bgp Border Gateway Protocol (BGP)  
eigrp Enhanced Interior Gateway Routing Protocol (EIGRP)  
ospf Open Shortest Path First (OSPF)  
rip Routing Information Protocol (RIP)

```
R1(config)#router rip
```

/\*

## Vogliamo utilizzare RIP.

Se non lo diciamo esplicitamente, di default usa la versione 1.

Quindi usiamo il seguente comando, con il quale entriamo nel modo di configurazione specifico per questo protocollo di routing:

Per assegnare le interfacce del router che dovranno utilizzarlo devo usare un altro comando.  
\*/

```
R1(config-router) #network 192.168.1.0
```

```
/*
```

Per configurare RIP si utilizza il comando `network`.

`network` vuole una quadrupla (a.b.c.d), non c'è la lunghezza del prefisso perché ragiona in termini di indirizzi classful.

Dopo il comando, il router controlla ogni interfaccia, se un'interfaccia ha l'indirizzo assegnato che fa match con quello specificato nel comando

- Su quell'interfaccia devo mandare gli update:
- Inoltre quella rete deve essere inclusa su tutti gli update, quindi il DV pubblicizza ai router vicini la rete a cui è connessa quella interfaccia.

Da questo momento gli update passano solo sul link di 192.168.1.0, e non sul link di 192.168.2.0.

```
*/
```

```
R1#show running-config // Vediamo il file di configurazione.
Current configuration : 799 bytes
!
version 12.3 // Non ci interessa al momento.
no service timestamps log datetime msec // Non ci interessa al momento.
no service timestamps debug datetime msec // Non ci interessa al momento.
no service password-encryption // Non ci interessa al momento.
!
hostname R1 // Router che Stai Configurando
!
...
!
ip cef // Non ci interessa al momento.
no ipv6 cef // Non ci interessa al momento.
!
...
!
spanning-tree mode pvst // Non ci interessa al momento.
!
...
!
interface FastEthernet0/0
```

```

mac-address 0001.4375.3b2a
ip address 192.168.1.1 255.255.255.0
duplex auto
speed auto
!
interface FastEthernet0/1
mac-address 0004.9a58.d25c
no ip address
duplex auto
speed auto
shutdown
!
interface Serial0/0/0
ip address 192.168.2.1 255.255.255.0
clock rate 64000
!
interface Serial0/0/1
no ip address
clock rate 2000000
shutdown
!
interface Vlan1
no ip address
shutdown
!
router rip // Elenco delle interfacce su cui uso RIP
network 192.168.1.0
!
ip classless

```

### **Cosa succede se in RIP v1 metto un indirizzo IP Classless?**

Se vado sulla configurazione di RIP e digito:

```
R1(config-router)#network 192.168.2.32
```

Lo esegue, non dà errore, però RIP in realtà è come se eseguisse:

```
R1(config-router)#network 192.168.2.0
```

Dà per scontato che sia un indirizzo di classe C, quindi ignora tutti i bit dell'ultimo byte.

Se andiamo nel file di configurazione infatti troviamo un altro comando:

```
router rip
network 192.168.1.0
network 192.168.2.0
```

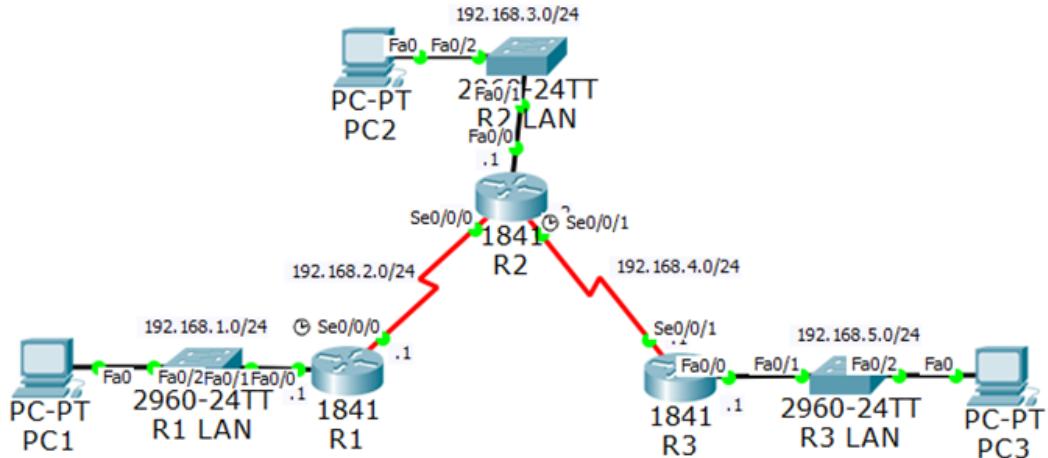
Salviamo la configurazione e andiamo su R2 a configurare RIP:

### **RIP Database**

```
R1# show ip rip database
```

Per ogni destinazione che ho imparato.  
Elenco chi mi ha inviato i dati e cosa mi ha inviato.  
Ossia mostra le info con cui ha fatto il merge.

### Esercizio 5.1



Indirizzi classful /24.  
Configuriamo la rete in modo da utilizzare RIP

Andiamo su R1.

Come ogni esercizio, osserviamo la tabella di routing di R1.

Per vedere se c'è qualcosa che manca.

```

R1#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B
      - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS
      inter area
      * - candidate default, U - per-user static route, o - ODR
      P - periodic downloaded static route
      Gateway of last resort is not set

C 192.168.1.0/24 is directly connected, FastEthernet0/0
C 192.168.2.0/24 is directly connected, Serial0/0/0
  
```

Le reti che abbiamo sono solo quelle direttamente connesse.  
Dobbiamo aggiungere le reti non direttamente connesse, ovvero:

- 192.168.1.3/24
- 192.168.1.4/24
- 192.168.1.5/24

Intanto configuro RIP su R1, dicendo quali interfacce usare e quindi quali reti pubblicizzare con RIP.

```
R1>enable
R1#conf t
R1(config)#router rip
R1(config-router)#network 192.168.1.0
R1(config-router)#network 192.168.2.32
R1(config-router)#exit
R1# copy r s
```

Andiamo su R2.

Dove faremo la stessa cosa, ma con le interfacce di R2.

```
R2(config)#router rip
R2(config-router)#network 192.168.2.0
R2(config-router)#network 192.168.3.0
R2(config-router)#network 192.168.4.0
R2#copy r s
```

Andiamo su R3.

```
R3(config)#router rip
R3(config-router)#network 192.168.4.0
R3(config-router)#network 192.168.5.0
R3#copy r s
```

Ora dobbiamo verificare se la configurazione è corretta.

Facciamo passare un pò di tempo (tipo 60 secondi) per permettere a RIP di lavorare.

Adesso vediamo la tabella di routing di R3.

Vediamo se ha aggiunto le reti non direttamente connesse, ha calcolato la distanza e c'è il tempo dall'inserimento della riga:

```
R3>enable
R3#show ip route

...
R 192.168.1.0/24 [120/2] via 192.168.4.2, 00:00:23, Serial0/0/1
R 192.168.2.0/24 [120/1] via 192.168.4.2, 00:00:23, Serial0/0/1
R 192.168.3.0/24 [120/1] via 192.168.4.2, 00:00:23, Serial0/0/1
C 192.168.4.0/24 is directly connected, Serial0/0/1
C 192.168.5.0/24 is directly connected, FastEthernet0/0
```

Lo stesso facciamo in R1:

```
R1>enable
R1#show ip route
...
C 192.168.1.0/24 is directly connected, FastEthernet0/0
```

```
C 192.168.2.0/24 is directly connected, Serial0/0/0
R 192.168.3.0/24 [120/1] via 192.168.2.2, 00:00:04, Serial0/0/0
R 192.168.4.0/24 [120/1] via 192.168.2.2, 00:00:04, Serial0/0/0
R 192.168.5.0/24 [120/2] via 192.168.2.2, 00:00:04, Serial0/0/0
```

Una condizione necessaria perché la configurazione di RIP sia corretta è che tutte le tabelle di routing di tutti i router elenchino le stesse destinazioni.

Se questa condizione è rispettata, allora ogni sottorete che intendiamo pubblicizzare risulta, in qualche modo, raggiungibile da ogni sottorete di uno qualsiasi dei router della rete.

#### RIP Database

Se uno volesse vedere questi messaggi di update possiamo passare allo strumento di simulazione oppure utilizzare il comando (il database contiene tutti i messaggi scambiati nel caso del protocollo RIP):

```
R1#show ip rip database
    192.168.1.0/24 auto-summary
        192.168.1.0/24 directly connected, FastEthernet0/0

    192.168.2.0/24 auto-summary
        192.168.2.0/24 directly connected, Serial0/0/0

    192.168.3.0/24 auto-summary
        192.168.3.0/24
            [1] via 192.168.2.2, 00:00:00, Serial0/0/0

    192.168.4.0/24 auto-summary
        192.168.4.0/24
            [1] via 192.168.2.2, 00:00:00, Serial0/0/0

    192.168.5.0/24 auto-summary
        192.168.5.0/24
            [2] via 192.168.2.2, 00:00:00, Serial0/0/0
```

[n] → Numero di Hop necessari

Il comando mostra per ogni destinazione che ho imparato, elenco che mi ha inviato i dati e cosa mi ha inviato.

#### RIP - Debug

Comando per abilitare le notifiche relative agli eventi del protocollo RIP:

**RIP: received**

[Versione del DV]

**update from**

[Indirizzo IP dell' Interfaccia da cui è Arrivato]

**on**

[Nome dell' Interfaccia da cui è Arrivato]

**RIP: sending**

[Versione del DV]

**update to**

[Indirizzo del ricevente del DV, può essere broadcast]

**via**

[Nome dell'Interfaccia in cui l'ho mandato]

([Indirizzo IP dell'Interfaccia in cui l'ho mandato])

```
R1#debug ip rip
    RIP protocol debugging is on
R1#debug ip rip

RIP: sending v1 update to 255.255.255.255 via FastEthernet0/0
(192.168.1.1)
// Update inviato alla rete direttamente connessa a R1
RIP: build update entries
    network 192.168.2.0 metric 1
    network 192.168.3.0 metric 2
    network 192.168.4.0 metric 2
    network 192.168.5.0 metric 3

RIP: received v1 update from 192.168.2.2 on Serial0/0/0
// Update ricevuto da R2
    192.168.3.0 in 1 hops
    192.168.4.0 in 1 hops
    192.168.5.0 in 2 hops

RIP: sending v1 update to 255.255.255.255 via Serial0/0/0
(192.168.2.1)
RIP: build update entries
    network 192.168.1
/*
Update inviato a R2
Vedo meno righe perché "Split Horizon" è attivo, quindi non invio a R2 le righe relative alle reti che mi ha pubblicizzato lui
*/
```

Anche i singoli host mantengono delle tabelle di routing al loro interno.

**Cosa succede se Taglio un Cavo?**

Adesso tagliamo il cavo tra R3 e lo switch della LAN3.

Su R3 arriva la notifica che quella rete sulla porta Fa0/0 è down, e la riga 192.168.5.0 viene cancellata dalla tabella di routing.

Andiamo su R2 e abilitiamo il debug.

Adesso eliminiamo quel collegamento, stoppiamo il tempo prima dei 30 secondi, R3 manda la triggered update, con 16 hops a indicare che quella rete è diventata irraggiungibile:

```
RIP: sending v1 update to 255.255.255.255 via Serial0/0/1  
(192.168.4.2)
```

```
RIP: build update entries  
      network 192.168.1.0 metric 2  
      network 192.168.2.0 metric 1  
      network 192.168.3.0 metric 1
```

// Vedo arrivare il trigger update da R3.

**RIP: received v1 update from 192.168.4.1 on Serial0/0/1  
192.168.5.0 in 16 hops**

R2 a questo punto modifica la sua tabella di routing inserendo 16.

```
RIP: received v1 update from 192.168.2.1 on Serial0/0/0  
192.168.1.0 in 1 hops
```

```
RIP: sending v1 update to 255.255.255.255 via Serial0/0/0  
(192.168.2.2)
```

```
RIP: build update entries  
      network 192.168.3.0 metric 1  
      network 192.168.4.0 metric 1  
network 192.168.5.0 metric 16
```

```
R2#no debug ip rip // Disabilito il Debug  
RIP protocol debugging is off
```

```
R2#show ip route
```

```
...  
R 192.168.1.0/24 [120/1] via 192.168.2.1, 00:00:07, Serial0/0/0  
C 192.168.2.0/24 is directly connected, Serial0/0/0  
C 192.168.3.0/24 is directly connected, FastEthernet0/0  
C 192.168.4.0/24 is directly connected, Serial0/0/1  
R 192.168.5.0/24 is possibly down, routing via 192.168.4.1,  
Serial0/0/1
```

Dice che quella rete è “possibly down”, non la toglie subito dalla tabella di routing per i motivi spiegati prima.

Se ricollegiamo la rete, R3 invia un triggered update anche in questo caso.

## RIP - Anomalie con le versioni di RIP

Cosa succede se uso indirizzi classless con RIP v1?

Applichiamo in 5.2 la stessa identica configurazione di 5.1.

Se uso "network 172.30.3.0", il router capisce "network 172.30.0.0".

Questo perché non specifichiamo il tipo di indirizzo, e quindi prende la versione classful dell'indirizzo.

- A.
- B. 172.30.x.x
- C. 192.168.5.x

Perché a R1 non arrivano gli update di 172.30.2.0, rete collegata con R2?

Questo perché qualsiasi rete del tipo 172.30.x.x sono tutte la stessa rete, ossia una rete con indirizzo IP di classe B, quindi R2 e R1 sembrano essere direttamente connessi alla stessa rete.

I router non annunciano ai loro vicini le reti a cui sono entrambi direttamente connessi, quindi lo split horizon toglie dai DV le righe relative alla rete 172.30.2.0.

### Boundary Router

Un router può autonomamente fare una route summarization.

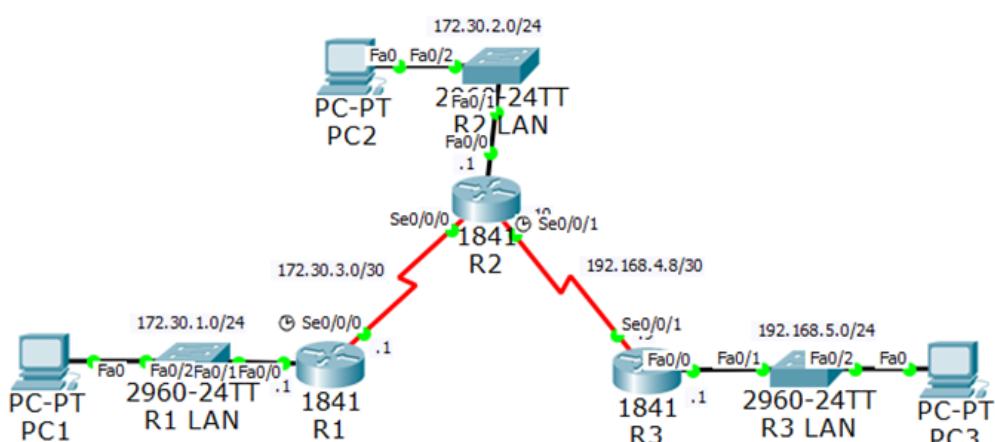
Posso disabilitarla con il comando **no auto summary**

### Comando passive interface

`passive interface fa0/0`

Fisicamente, su questa interfaccia non mandare gli update.

## Lab 5.2



Su R1 dovrei eseguire un comando `network` per 172.30.1.0/24 e uno per 172.30.3.0/30, però il comando `network` verrà interpretato da RIP con un indirizzo classful.

```
R1(config-router) #network 172.30.1.0
```

```
R1(config-router)#network 172.30.3.0
```

Però se vediamo il file di configurazione, vediamo che lui ha utilizzato questo indirizzo:

```
!  
router rip  
network 172.30.0.0  
!
```

Quindi configurare la rete 172.30.3.0/30 non ha senso visto che la interpreta allo stesso modo.

Andiamo su R2 e scriviamo:

```
R2(config-router)#network 172.30.0.0  
R2(config-router)#network 192.168.4.0
```

Andiamo su R3 e scriviamo:

```
R3(config-router)#network 192.168.4.0  
R3(config-router)#network 192.168.5.0
```

Se andiamo nella tabella di routing di R1 e vediamo che c'è 192.168.4.0/24 e 192.168.5.0/24, non c'è invece 172.30.2.0/24, la quale è direttamente connessa a R2.

Questo perché R1 è connesso ad una rete 172.30.1.0/24 e interpreta 172.30.2.0/24 come se fosse la stessa rete, quindi tutte le reti 172.30.x.x le interpreta come sottoreti di 172.30.0.0.

Le reti direttamente connesse non vengono comunicate, lo split horizon toglie le righe delle reti connesse al router a cui si manda il distance vector.

In R2 abbiamo queste reti:

```
C 172.30.2.0/24 is directly connected, FastEthernet0/0  
C 172.30.3.0/30 is directly connected, Serial0/0/0  
192.168.4.0/30 is subnetted, 1 subnets  
C 192.168.4.8 is directly connected, Serial0/0/1  
R 192.168.5.0/24 [120/1] via 192.168.4.9, 00:00:29, Serial0/0/1
```

Abbiamo solo 192.168.5.0/24 essendo di classe C, quindi il prefisso è diverso

R3 invece ha:

```
R 172.30.0.0/16 [120/1] via 192.168.4.10, 00:00:27, Serial0/0/1  
192.168.4.0/30 is subnetted, 1 subnets  
C 192.168.4.8 is directly connected, Serial0/0/1  
C 192.168.5.0/24 is directly connected, FastEthernet0/0
```

Dove la prima rete è un summary, quindi qualunque cosa diretta alla rete 1 o 2 va ad R2, però se faccio un ping da PC1 a PC2, il pacchetto arriva a PC3, ma non torna a PC1, si ferma nel router 2.

Quindi il funzionamento è casuale se uso indirizzi classless con la versione 1, quindi andiamo su R2 e si indica la versione 2:

**RIP - Cambio di Versione**

```
R1(config-router)#version 2
```

Se R2 rimanesse con la versione 1, esso ignorerà i pacchetti di R2 dato che R2 utilizza solo pacchetti versione 1.

R1 adesso vede solo pacchetti versione 2, e dato che ignora gli updates di v1, tra poco scadranno le destinazioni della rete 3.

Dopo 180 secondi, nella tabella di routing di R1 ci verrà scritto che quel l'interfaccia è possibly down.

Adesso andiamo su ogni router ed eseguiamo il comando di cambio versione.

In questo modo non cambiano gli indirizzi assegnati con network.

Il comando version 2, dovrebbe essere fatto come primissima cosa, prima di utilizzare il comando network, altrimenti rischio di avere informazioni non valide nelle tabelle per un pò di minuti.

### **Eliminare la Auto-Summarization**

```
R2(config-router)#no auto summary
```

### **RIP - Interfacce Passive**

Io conosco le interfacce della rete, se ho una stub network, sicuramente non riceverò update da essa, e non dato che non ci sono router, non serve che invio update.

Devo usare il comando network ugualmente, e successivamente su R1 uso il comando passive-interface.

In questo modo la rete è pubblicizzata ai vicini, ma non invierò ne mi aspetterò update da essa.

```
R1(config)#router rip  
R1(config-router)#passive-interface Fa0/0
```

Serve perché si consumavano risorse di rete per delle informazioni che non servono a nessuno, va fatto allo scritto, per le interfacce dei router dal lato della rete, quindi per la Fa0/0 nella rete 2, la Fa0/0 nel caso della rete 3.

Questo comando può essere utile anche per ignorare gli update di un router bersaglio o per non inviare gli update ad uno specifico router.

### **RIP - Default Route Propagation**

Si va nell'esercizio 5.3.

La route di default (0.0.0.0 0.0.0.0) va configurata in modo statico.  
NON VIENE CONDIVISA DA RIP.

Spesso in un router ci sono molteplici protocolli di routing in atto.

Di norma, le righe nella tabella immesse da un protocollo di routing non possono essere lette o modificate da altri protocolli di routing diversi da quello che le ha messe.

Le route statiche sono viste dai protocolli di routing come righe messe da un altro protocollo di routing.

Quindi o definisco la default route per ogni router, ma è statica e ha i limiti di ogni possibile route statica.

Io vorrei che sia RIP ad annunciare la default route.

Questa cosa la si fa con il comando `redistribute`, che consente a RIP di redistribuire le route statiche o informazioni apprese da altri protocolli di routing.

Ma a me interessa solo al default route, quindi con il comando `default-information originate`, la default route (quella con \*) viene distribuita anche ai router vicini.

A quel punto, i router che la ricevono, la condividono come se fosse una destinazione qualunque, perché nella loro routing table, essa sarà del tipo (R\*).

Con il comando `redistribute` si possono mischiare le informazioni dei router e propagarla a tutta la rete.

### Algoritmo OSPF

Oggi vediamo **OSPF (Open Shortest Path First)** che ha un livello di complessità maggiore rispetto a RIP.

E' simile ad **IS-IS (Intermediate System to Intermediate System)**, il quale è uno standard non definito dentro lo IETF, ma è stato definito all'interno di un insieme di protocolli che furono definiti in alternativa a **TCP/IP**, ovvero il modello **ISO/OSI**.

### Cos'è un Algoritmo di tipo Link State?

OSPF è un algoritmo di tipo *link state*, ossia: durante l'esecuzione dell'algoritmo, i nodi si interscambiano informazioni riguardo i link a loro direttamente connessi.

Supponiamo di essere un router che partecipa al routing.

In quanto tale mi serve una base di informazioni che mi permette di conoscere la topologia della rete.

Questa conoscenza la acquisisco grazie ai miei **vicini**, ovvero altri router connessi a me.

### Link State - Router Direttamente Connesso ad una Rete

Ogni router avrà almeno una rete direttamente connessa.

Un router è direttamente connesso ad una rete, solo se ha almeno un interfaccia connessa a quella rete.

### Link State - Messaggi di Link State & Flooding

In genere in una rete nessun router è direttamente connesso a tutti gli altri router della rete. Per far arrivare una info a tutti i router allora si utilizza il meccanismo del *flooding* (inondazione).

1. Router A invia il messaggio di link state ai suoi router vicini.

- I router vicini del router A inoltreranno a loro volta il messaggio a tutti i loro vicini, e così via.

Queste informazioni vengono codificate con un messaggio detto "link state".

Quindi un router non invia solo i Link State che ha creato lui ma inoltra ai suoi vicini anche quelli che arrivano da altri router (flooding).

Se la rete è interamente connessa (tutti i router in qualche modo possono raggiungere tutti gli altri router) ogni router della rete riceverà il messaggio di link state.

Quando un router riceve un messaggio di Link State, oltre a propagare l'informazione, esso conserva una copia di questo messaggio in un database interno.

I messaggi di link state hanno lo scopo di far capire la topologia della rete a tutti i router.

### Link State - Riempire la Tabella di Routing & Algoritmo di Dijkstra

Supponiamo di essere un router che ha partecipato al flooding di messaggi link state. Dopo un pò di tempo avrò del mio Database Interno tutti i messaggi scambiati da tutti i router e quindi a quel punto conosco tutte le destinazioni possibili e per ciascuna destinazione siamo in grado di calcolare il modo migliore per arrivarci.

Il router grazie ai messaggi di link state, crea un grafo, dove i nodi sono i router e gli archi sono i link tra i router, ogni arco ovviamente ha un peso (costo di inviare un messaggio attraverso questo link).

A quel punto basta trovare il percorso di costo minimo per ogni possibile destinazione della rete con l'algoritmo di **Dijkstra** (aka *Shortest Path First*).

Una volta concluso Dijkstra.

Data una qualsiasi destinazione della rete basta osservare il grafo (e sapendo quale nodo mi rappresenta) posso facilmente ricavare il next-hop a cui inviare quel pacchetto e quindi riempire la tabella di routing.

La garanzia che i router useranno sempre il percorso ottimo è data dall'algoritmo di Dijkstra, il quale è un algoritmo consolidato e molto famoso.

### Link State - Come è rappresentato il Grafo della Rete internamente?

Sender	Node/Metric
A	B/10 C/9 {192.168.0.0/24}/1
B	A/10 E/4
C	A/9 D/2
D	C/2 E/5 F/6 {192.168.3.0/24}/1
E	B/4 D/5 F/2 {192.168.4.0/24}/6
F	D/6 E/2 {192.168.5.0/24}/2

Questa tabella è una struttura dati che rappresenta un grafo, per ogni router ho le informazioni sui suoi router vicini e su eventuali reti di destinazione direttamente connesse ad essi.

Il numero in rosso è il costo del ramo, ad esempio la prima riga indica:

Tramite il router A posso raggiungere:

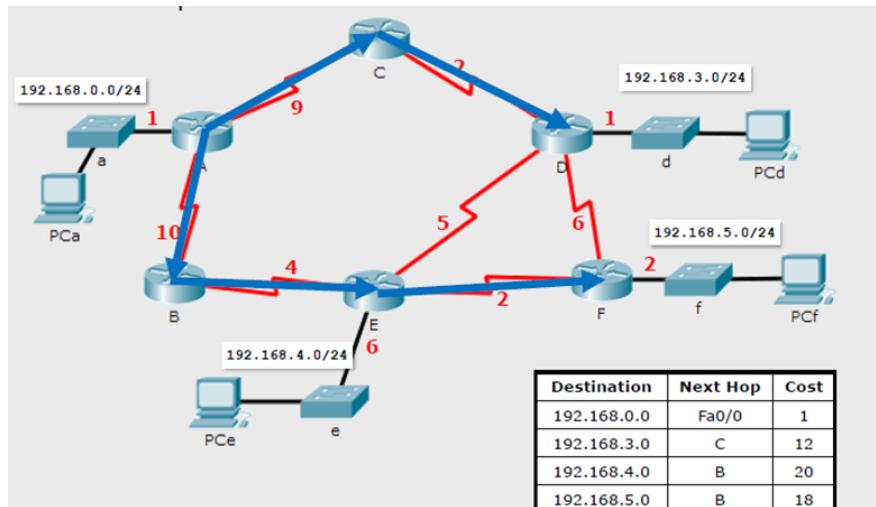
- Il router B con costo 10

- Il router C con costo 9
  - La rete 192.168.0.0/24 con costo 1.

## Link State - Come un router costruisce la sua tabella di routing

Ogni router calcola l'albero dei cammini minimi con radice lui stesso.

L'albero dei cammini minimi visita tutte le possibili destinazioni mantenendo il costo minimo possibile (percorso blu in figura).



### **Link State - Che vantaggi ho rispetto a RIP?**

Tramite questa struttura dati il router conosce tutto il percorso e non solo il primo hop (figura sotto).

Se cambia un percorso, cambiano alcune righe della tabella, non tutte.

## **Link State - Problema del Flooding Infinito**

Questo problema dipende anche dalla topologia della rete.

Se nella rete ci sono percorsi chiusi, può verificarsi un problema:

Supponiamo di essere un router coinvolto nel percorso chiuso e supponiamo di aver appena mandato un messaggio link state.

In un percorso chiuso può capitare che il messaggio che ho precedentemente inviato ad un mio vicino mi ritorni indietro e se non ho modo di riconoscere che era un mio messaggio continuerò a trasmetterlo all'infinito.

Questa cosa accade perché il flooding per ora definito segue la regola: “*se mi arriva un Link state copio e inoltro*”.

Perché dipende dalla topologia?

Se ho una rete ad albero non c'è problema, basta mandare un messaggio solo una volta e quando il messaggio arriverà alle foglie (ossia router con un solo vicino) queste ultime non lo trasmetteranno e quindi non ha modo di tornare indietro.

## **Link State - Soluzione al Flooding Infinito: “Selective Flooding”**

Il flooding deve ovviamente veloce ma anche selettivo (**Selettivo verso i link state che riceve**), ovvero: ogni router della rete che riceve un link state deve essere in grado di capire se questa è una copia di un messaggio che ha già ricevuto o che lui stesso ha generato.

Come nel flooding normale, ogni router ha un database interno contenente i messaggi link state che ha precedentemente ricevuto.

Supponiamo che un router riceva un messaggio di link state.

Ogni messaggio di link state, al suo interno ha un identificatore che lo identifica univocamente da tutti gli altri messaggi di link state della rete.

Questo ID lo uso come chiave primaria del database dei messaggi ricevuti.

A quel punto il router ricevente esegue una query nel suo database interno.

- Se questa informazione non c'è nel database, aggiorno il DB e inoltro il messaggio ai miei vicini tranne al vicino che me lo ha mandato.
- Se questa informazione è già presente nel database, non lo copio e non lo trasmetto.

### Link State - “Correggere l'errore di un vicino”

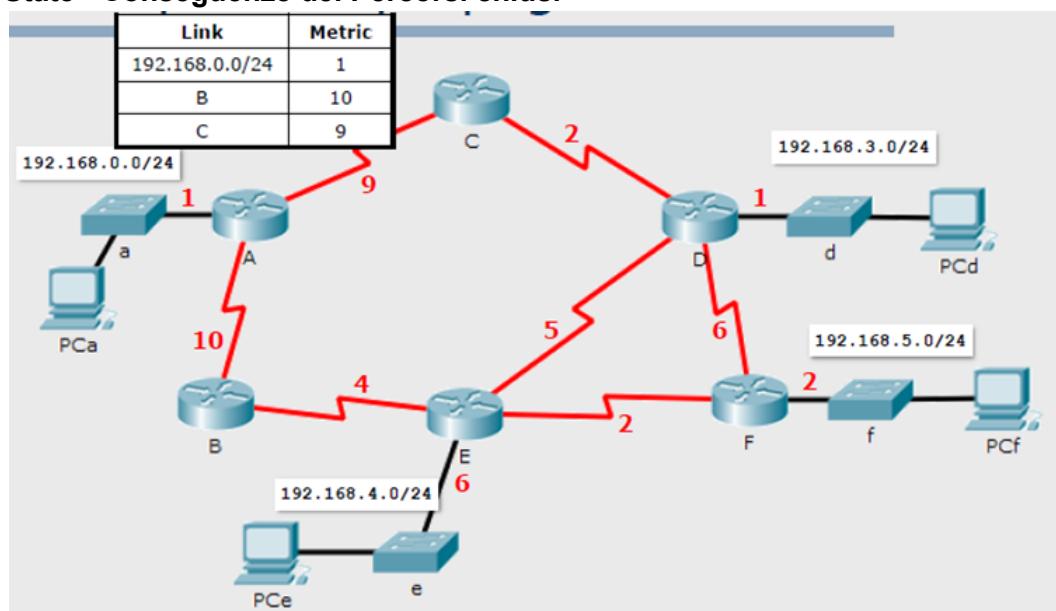
Supponiamo che un router invii un messaggio di link state vecchio (magari frutto di flooding infinito o altro) ad un suo vicino.

Il secondo router ovviamente non avrà questa informazione nel suo database ma avrà la sua versione aggiornata.

Quindi il secondo router non ignora del tutto il link state obsoleto, ma invia (solo al router mittente) l'informazione aggiornata dal mio database.

In questo modo il primo router aggiorna l'informazione.

### Link State - Conseguenze dei Percorsi chiusi



1. A genera un messaggio link state (*la tabella in figura*)
2. A lo trasmette ai suoi vicini, quindi i router B e C.
3. C lo trasmette a D.
4. B lo trasmette ad E.
5. D lo trasmette ad E ed F.
6. E lo trasmette a F.

Quale è il problema?

Alcuni router (E ed F) ricevono inutilmente la stessa informazione più volte.

Perché?

Nella rete ho un percorso chiuso e perché i router inoltrano il messaggio di link state ai vicini “senza pensare” (*poiché non possono saperlo, non circola questa informazione*) se quel vicino lo ha già ricevuto oppure no.

## **Link State vs Distance Vector (RIP)**

L'algoritmo Link state ha una serie di vantaggi rispetto a DV.

- Link State è più stabile.
- Link State ha un tempo di convergenza migliore.
  - Poiché il numero di messaggi link state dipendono logaritmicamente dal numero di nodi;

### **Periodo Transitorio**

Nel distance vector (RIP) ho un periodo di tempo “transitorio” (ossia il tempo che passa dall'inizio a quando i router sono tutti arrivati alla convergenza) che coinvolge tutti i router, i quali ricevono un'informazione, la elaborano e poi la trasmettono.

In RIP il transitorio finisce quando tutti i router arrivano allo stato comune (convergenza).

In RIP se c'è un router più lento di tutti gli altri, questo porterà l'intera rete ad avere un tempo transitorio maggiore (perché devono attendere che anche questo sia aggiornato).

In link state, per prima cosa i Router si scambiano tutte le informazioni necessarie.

Alla fine delle trasmissioni ognuno esegue l'algoritmo per conto suo e crea la sua struttura dati.

Se è presente un router più lento degli altri non compromette gli altri, i quali nel frattempo sono già operativi.

## **Algoritmo di Dijkstra - Percorsi Chiusi**

Se tutti i router utilizzano Dijkstra per calcolare l'albero dei cammini minimi.

Dijkstra garantisce che negli alberi di percorso minimo non ci siano percorsi chiusi.

L'unico rischio di avere un percorso chiuso è nell'intervallo brevissimo in cui dei router operano su una struttura dati inesatta, ossia operano su una topologia “virtuale” (ossia quella rappresentata nelle loro strutture dati) diversa da quella reale.

Questa cosa può accadere perché in quell'istante alcuni router non sono ancora del tutto aggiornati perché magari più lenti degli altri.

## **Link State - Note su Database e Tabella di Routing**

Il database interno che contiene i messaggi di link state deve essere identico per ogni router (ossia tutti hanno le stesse informazioni di base e quindi i router conoscono tutta la stessa topologia della rete).

Le tabelle di routing, ovviamente, devono essere diverse perché ogni router calcola l'albero dei cammini minimi usando una radice diversa (ossia loro stessi).

Invece in RIP il database interno è diverso per ciascun router, perché ricevono solo messaggi dai vicini e non da tutti i router.

## **OSPF - Versioni**

Noi vediamo la versione 2 di OSPF, la quale è capace di gestire indirizzi IPv4 di tipo classless.

La versione 3 supporta sia indirizzi IPv4 che IPv6, supporta il CIDR (Classless Inter Domain Routing, ovvero nella tabella di routing deve esserci il /x).

## **Dominio di Routing**

Chiameremo dominio di routing, una “zona” in cui sono presenti n router che si scambiano messaggi.

## **Autonomous System (AS)**

Un dominio di routing gestito con OSPF è detto Autonomous System (AS).

In internet ogni AS possiede un AS number univoco.

Un provider può avere più AS, ciascuno dei quali con un proprio identificatore (AS number).

## **AS Number**

Un AS è una rete connessa, e ha un AS number solo se fa routing a livello di internet.

Una rete aziendale è un AS, ma non ha un AS number perché non fa routing a livello di internet, fa routing solo internamente alla rete aziendale.

## **Compito di OSPF**

Il compito di OSPF è definire il routing all'interno dell'AS.

Quindi costruirà una riga all'interno della tabella di routing, se questo AS è connesso ad altri AS.

OSPF si fermerà al confine (non fa a fare routing fuori dall'AS).

Però OSPF definisce il percorso fino al router per uscire dall'AS.

## **AS - Identificatore di Router**

Dentro un AS ciascun router ha il proprio identificatore univoco ed è codificato su 32 bit (router ID) (è diverso dall'IP anche se è su 32 bit).

Nelle reti private si può utilizzare un blocco d'indirizzi privati per gli identificatori di questi router.

Vedremo nello specifico di cosa si tratta.

## **Limiti di OSPF**

### **Scalabilità**

Teoricamente OSPF funziona indipendentemente dal numero di router all'interno dell'AS, ma nella realtà no, OSPF si trova in difficoltà se ho troppi router.

Questo a causa della quantità d'informazione (messaggi link state) che deve circolare con il flooding.

Se ho 1000 router, un router deve fare in modo che l'informazione venga trasmessa almeno una volta su tutti i link per essere sicuro di aver raggiunto gli altri 999 router.

### **Costo Computazionale per il Singolo Router**

Distance vector parte dall'idea di distribuire il calcolo tra i router, il calcolo fatto da un singolo router è di fatto molto semplice: il merging dei DV (distance vector) che sono pochi dato che l'informazione è distribuita tra tutti i router.

In link state, quando ho ricevuto l'informazione devo calcolarmi Dijkstra da solo, quindi tutti i 1000 router devono calcolarsi Dijkstra su un grafo di 1000 nodi.

### Suddivisione di un AS in aree

Per risolvere questo problema partiziono la rete in Aree, ossia AS più piccoli.

L'AS viene separato in aree più piccole e link state opera all'interno di ogni area indipendentemente dalle altre, in questo modo i router eseguono calcoli solo basandosi sulla propria area e quindi molto più facili.

Per connettere tra loro i singoli AS uso l'algoritmo BGP (Border Gateway Protocol) che vede gli AS come singoli router.

Oppure un'altra scelta è includere un meccanismo in OSPF che implementa la scalabilità ed è la soluzione che è stata adottata.

### Arese - Da cosa è composta?

Un'area è un sottoinsieme di link e router appartenenti all'AS originale.

L'appartenenza di un router ad un'area dipende dalle sue interfacce (link).

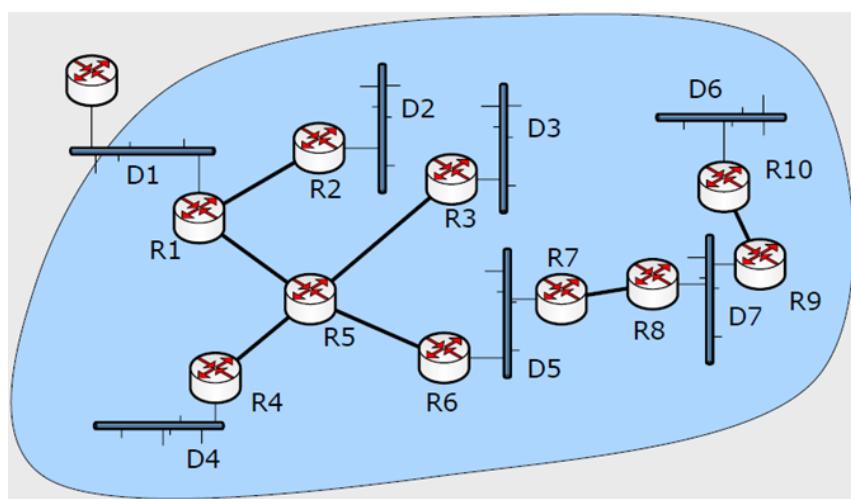
Quindi di fatto, sono le interfacce che vengono suddivise nelle aree e non tutto il router.

### Arese - Router di Bordo Area - Area Border Router (ABR)

Supponiamo di avere un router con un'interfaccia su un'area e un'interfaccia su un'altra area.

Questo router appartiene a più aree ed è detto **router di bordo area**.

Le aree hanno un ID a 32 bit per essere identificate.



### Arese - Boundary Router

Esempio figura sopra: R1 che ha un'interfaccia su un link su cui è collegato un router esterno all'AS.

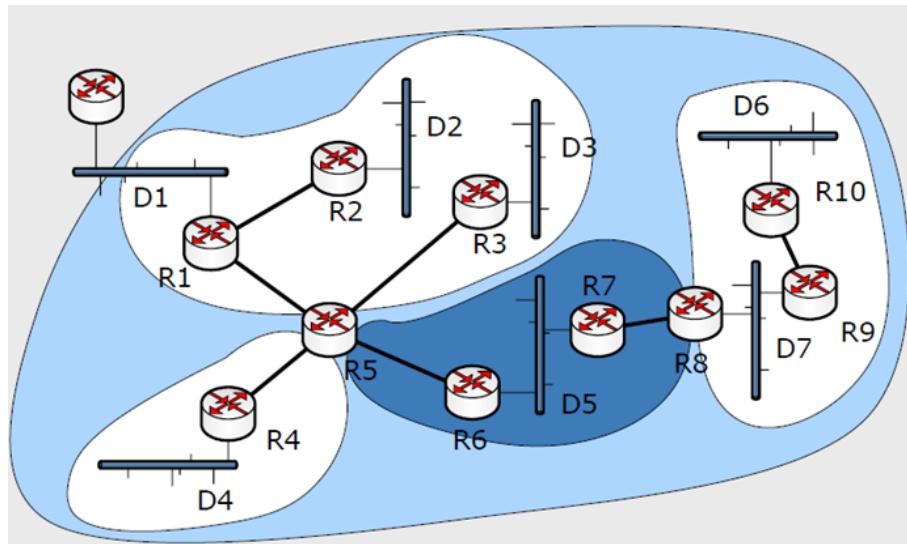
R1 è detto **router di boundary**, ciascuna di queste reti è destinazione oppure una rete che serve a due router per comunicare tra di loro.

### Divisione in Arese

Divido in aree l'AS.

Ciascuna area è connessa.

Se sono su R5 e voglio andare ad R1, R2, R3, D2 o D3 rimango all'interno dell'area senza passare da router esterni all'area.



### Ruoli di un Router in una AS

Un router può essere di 3 tipi:

- **Router interno:** ha tutte le interfacce all'interno dell'area.
- **Area Border Router (ABR):** router che ha interfacce connesse a più aree
- **AS Boundary Router:** router che ha almeno un'interfaccia collegata verso l'esterno

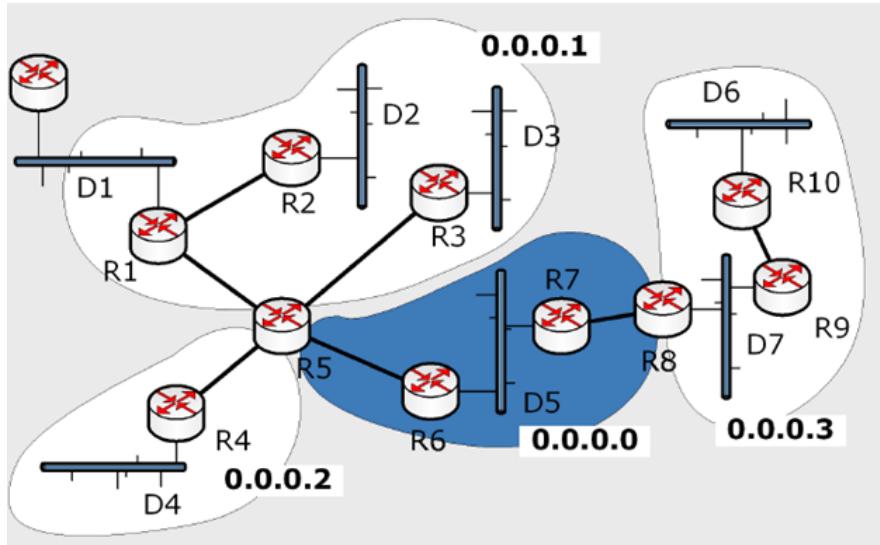
### Area di Backbone

Tutti gli ABR devono avere almeno un'interfaccia connessa ad un'area detta "area backbone".

Quindi tutte le aree di un AS devono essere connesse all'area backbone.

Imponendo questo vincolo, la topologia diventa molto semplice: ottengo un albero, dove la radice è l'area di backbone e le foglie sono le altre aree.

Nella figura, l'area di backbone è quella blu e ha come identificatore 0.0.0.0.



### Link State dentro un Area

Se è un router interno: Nella tabella ha le informazioni solo riguardanti la sua Area.

Se è un router ABR: Nella tabella ha le informazioni riguardanti le aree a cui è collegato

All'interno di ogni area funziona link state così come lo abbiamo descritto precedentemente. Se sono in un router e devo raggiungere una destinazione all'interno della mia area, calcolo il percorso con Dijkstra (**intra-area routing**).

### Link State tra Aree diverse

Qui entra in gioco il distance vector e i router ABR.

Supponiamo che R6, un router nella backbone, voglia raggiungere la rete D6 che si trova nell'area 0.0.0.3.

R8 è il router ABR tra l'area di backbone e l'area in cui si trova la rete D6.

R8 al suo interno ha gli alberi dei cammini minimi per l'area di backbone e l'area 0.0.0.3 (in realtà non gli basta molto meno, vedremo dopo)

1. Supponiamo che R8 possa raggiungere D6 con un percorso di distanza 15.
2. R8 comunica alla rete 0.0.0.0 che lui conosce la strada per D6 con distanza 15.
3. R6 nella backbone, calcola il percorso migliore per raggiungere R8 e invia il pacchetto.
4. R8 riceve il pacchetto e si occuperà di raggiungere D6 con il percorso migliore.

In parole poche:

Se devo raggiungere un router in un'altra area dell'AS, calcolo il percorso migliore per raggiungere l'ABR e poi l'ABR completerà il lavoro.

### Vantaggi della Suddivisione in Aree

Questa soluzione gerarchica è dovuta per questioni di scalabilità, i calcoli si devono fare con frequenza ridotta.

Se in un'area cambia la topologia o altro, nelle altre aree, le tabelle di routing rimangono praticamente inalterate devo solo aggiornare il costo.

L'aggiornamento verrà fatto con il prossimo flooding di link state o con eventuali trigger update.

## **Regole di Design di una Rete**

1. Fare in modo che non ci siano più di 6 hop tra una qualsiasi sorgente ad una qualsiasi destinazione.
2. Usare dai 30 ai 100 routers per area.
3. Non permettere che un ABR sia connesso a più di 2 aree contemporaneamente (conta che una delle 2 è per forza la backbone).
  - Altrimenti un ABR ha troppi Database da gestire..

## **Gli ABR hanno pochissime righe nella Tabella di Routing**

Negli ABR ho una sola riga per tutta l'area (si esegue un summary) invece che una riga per ogni destinazione dentro l'area.

## **Routing tra Aree**

Dal punto di vista del routing tra aree, se la mia destinazione è in un'area diversa (e non sono nella backbone) devo trovare il percorso ottimale tra le aree per raggiungere la destinazione.

## **Pacchetti che girano all'infinito tra vari AS**

Come per il routing classico, potrebbe esserci il problema dei pacchetti che girano all'infinito tra i vari AS.

BGP lo risolve con il **path vector**.

## **Percorsi chiusi tra AS**

Come per il routing classico, potrebbe esserci il problema di percorsi chiusi tra aree.

Si risolve evitando di avere percorsi chiusi tra aree, si può fare in tanti modi uno dei quali lo abbiamo già introdotto: la **backbone** alla quale sono connesse tutte le altre aree.

Grazie ad essa la topologia delle aree diventa ad Albero e la backbone diventa l'unico punto di uscita e il punto di entrata per ogni area.

Se ho un pacchetto che non è destinato all'interno della mia stessa area, questo viene inviato alla backbone e sarà essa a inviarlo alla destinazione (se la destinazione è la backbone stessa il pacchetto non viene instradato dalla backbone).

## **Divisione in Aree - Conseguenze nella Tabella di Routing**

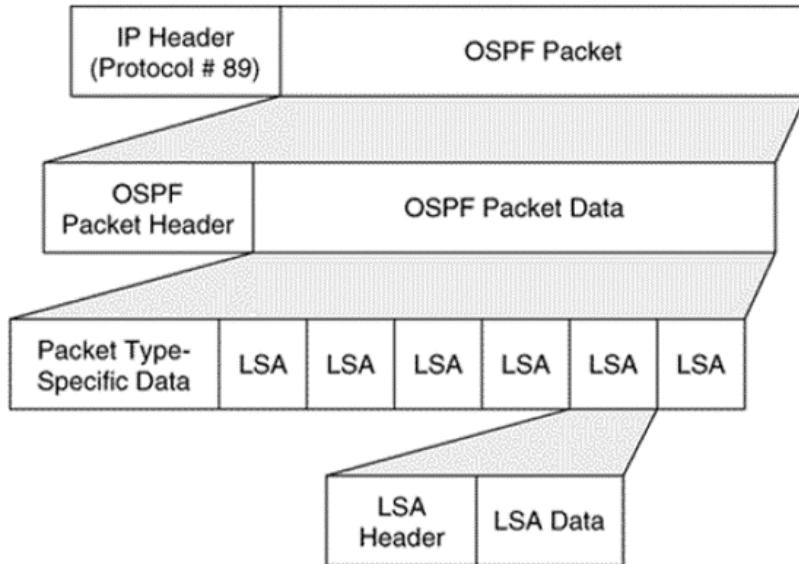
Nei router, la tabella di routing contiene tutte le destinazioni in tutte le aree, quando sono **nel piano dati** questa ripartizione in aree non la vedo più, la distinzione è **nel piano di controllo**, che riguarda il come riempire la tabella di routing.

Nelle tabelle di routing dei Router interni non avrò tutte le informazioni riguardo destinazioni fuori dalla mia Area.

## **Pacchetto OSPF**

Ci sono **5 tipi** di pacchetti, sono scambiati utilizzando il protocollo IP.

Hanno un numero di protocollo all'interno dell'header IP, che è il numero 89, il router quando riceve un messaggio del genere, toglie l'intestazione IP e prende il pacchetto OSPF.



All'interno di esso c'è un header OSPF nella quale ci sono delle informazioni che indicano al router che cosa farci.

In figura è un esempio di pacchetto che contiene i link state (**LSA Link State Advertisement**), contiene più link state.

Per motivi di efficienza OSPF non invia un LSA alla volta ma ne invia molteplici in un colpo solo.

### Header OSPF

- **versione**
  - Di OSPF (versione 2 per IPv4, 3 per IPv6).
- **tipo**
  - un numero (1-2-3-4-5) a cui corrisponde uno dei 5 tipi di pacchetti OSPF.
- **lunghezza**
  - Espressa in numero di byte, di tutto il pacchetto.
- **checksum**,
  - OSPF ne mette uno suo perché il checksum del pacchetto IP non mi garantisce al 100% il controllo dell'errore.
- **AuthType**:
  - Serve per autenticare il pacchetto, ovvero per verificare se il mittente è affidabile o meno.
  - Non vogliamo che utenti cattivi danneggino l'operato di OSPF con informazioni false o malevoli.
- **Router ID**:
  - Identificatore univoco del router all'interno dell'AS, indica il mittente del pacchetto che ha confezionato il messaggio e che me lo inviato,
- **Area ID**:
  - Indica l'area in cui il router mittente risiede.

Version	Type	Length
Router ID		
Area ID		
Checksum		AuthType
AuthData		
AuthData		

### Ruolo delle 5 tipologie di pacchetti

#### Tipo 1: Hello

Serve per scoprire chi sono i miei vicini e per monitorare lo stato della rete.

Viene trasmesso periodicamente da ciascun router in modo da verificare che i propri vicini siano ancora funzionanti (è l'implementazione di un meccanismo di **keep alive**).

Non contiene informazioni di routing.

In OSPF c'è una procedura particolare in cui se ci sono più router connessi in una LAN su un canale broadcast, viene eletto un router che svolge delle funzioni particolari di **backup** (trattato avanti);

#### Tipo 2: Database Description.

Il database è l'insieme degli LSA che un router ha ricevuto.

Un router costruisce il suo DB nel tempo man mano che riceve gli LSA dai vari router, finché non

si raggiunge una situazione di stabilità,

Questo pacchetto è introdotto per l'efficienza, consente ad un router di scambiare con il vicino tutto il contenuto del database e quindi mandargli in una volta sola tutta la cronologia dei messaggi LSA, in modo che anche lui possa ricostruire il grafo.

Il pacchetto non contiene tutto il database vero e proprio ma solo le chiavi primarie dei record.

Chi lo riceve, lo confronta con il suo database e richiede (tramite un pacchetto di tipo 3) le chiavi che gli mancano.

Ricordiamo che le chiavi (ossia gli id degli LSA) sono uniche in tutta la rete.

Questa cosa è utile, se ad esempio un router viene spento e riacceso.

Appena acceso esso vorrà scoprire i suoi vicini.

Ma la sua tabella di routing ed il suo DB sono vuoti, e per riempirli e quindi portarsi a regime occorre aspettare decine di minuti.

Per evitare questa attesa, i suoi vicini gli inviano un pacchetto di tipo 2.

Se al router manca qualcosa, non aspetto che arrivi, lo chiedo al mio vicino con un pacchetto di tipo 3.

#### Tipo 3: Link State Request.

Indica la richiesta di uno specifico set di LSA ad un router adiacente.

La richiesta è fatta indicando la chiave primaria.

#### **Tipo 4: Link State Update**

Trasmesso in risposta ad una richiesta di tipo 3.

Trasmesso anche per durante il flooding LSA.

#### **Tipo 5: Link State Acknowledgment**

Il router che riceve un LSA invia questo pacchetto di ACK per indicare al vicino che lo ha ricevuto con successo.

### **OSPF - startup**

Se sono un router ed ho un link seriale che mi collega ad un altro router, io posso comunicare con quel router (se mando un ping la cosa funziona normalmente) ma non posso inoltrare qualcosa attraverso lui, questo perché diventiamo vicini solo dopo che è terminata questa procedura.

Se questa procedura fallisce, noi possiamo inviarci pacchetti, ma per OSPF non siamo vicini, e quindi nel calcolo dell'albero non risulteremo vicino a quel router, e quindi non posso usarlo come next-hop.

#### **OSPF - startup - Neighbor Discovery**

In questa fase invio e ricevo i pacchetti di Hello per scoprire i vicini e per farmi scoprire da loro.

Nel pacchetto di Hello ogni router mette il suo ID nell'header.

Quando un router riceve un pacchetto di Hello da un suo vicino esso prende il router ID contenuto in esso e lo mette nella lista dei vicini.

La lista dei vicini di un router è stampata interamente nei messaggi di Hello che invierà da qui in avanti.

Quando un router riceve un hello, esso risponde inviando a sua volta un hello.

#### **OSPF - startup - Bidirectional Communication**

Questa fase riguarda 2 specifici router.

Questa fase è stabilita quando due router vicini mostrano (nella lista dei vicini dei loro pacchetti di Hello) l'ID di uno nell'Hello dell'altro.

Tramite questo match capiscono di essere vicini.

#### **OSPF - startup - Database Synchronization**

Prima di scambiare gli LSA ci assicuriamo che i nostri database siano identici.

Se non lo sono si fa la sincronizzazione: uno dei due si elegge master e l'altro lo slave.

I due si scambiano le parti mancanti del DB.

#### **OSPF - startup - Full Adjacency**

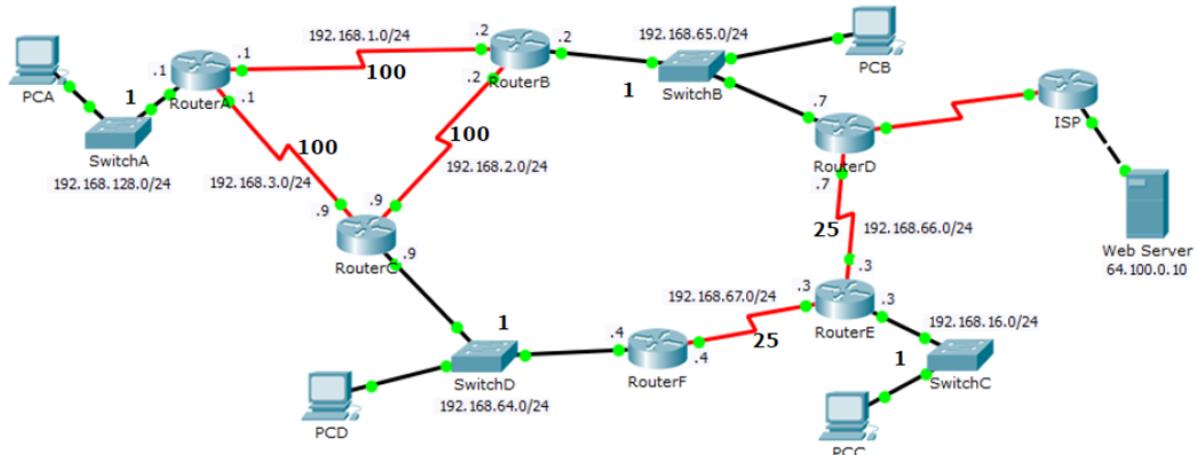
I due router sono completamente adiacenti.

Una volta in questo stato i router sono diventati "vicini" e si possono scambiare gli LSA.

### **I Router ABR**

Gli LSA sono trasportati dai link state update e contengono informazioni topologiche sulla rete.

Un router ABR ha un database separato per ogni area a cui è connesso.



Nella rete sopra c'è una sola area grande quanto l'AS che ha un suo ID e c'è un unico AS boundary router, ovvero routerD.

I numeri in grassetto sono i costi, qua sono associati ai link, in realtà i costi si associano alle **interfacce**, per semplicità le interfacce di un link hanno costi uguali in entrambe le direzioni, nella realtà possono avere costi differenti.

Ad esempio nel link tra router A e C, l'interfaccia di A potrebbe costare 5 e l'interfaccia di C potrebbe costare 10 (nel nostro caso abbiamo 100 in entrambe le interfacce) anche se è lo stesso link.

### Come vengono inviati i pacchetti di Hello?

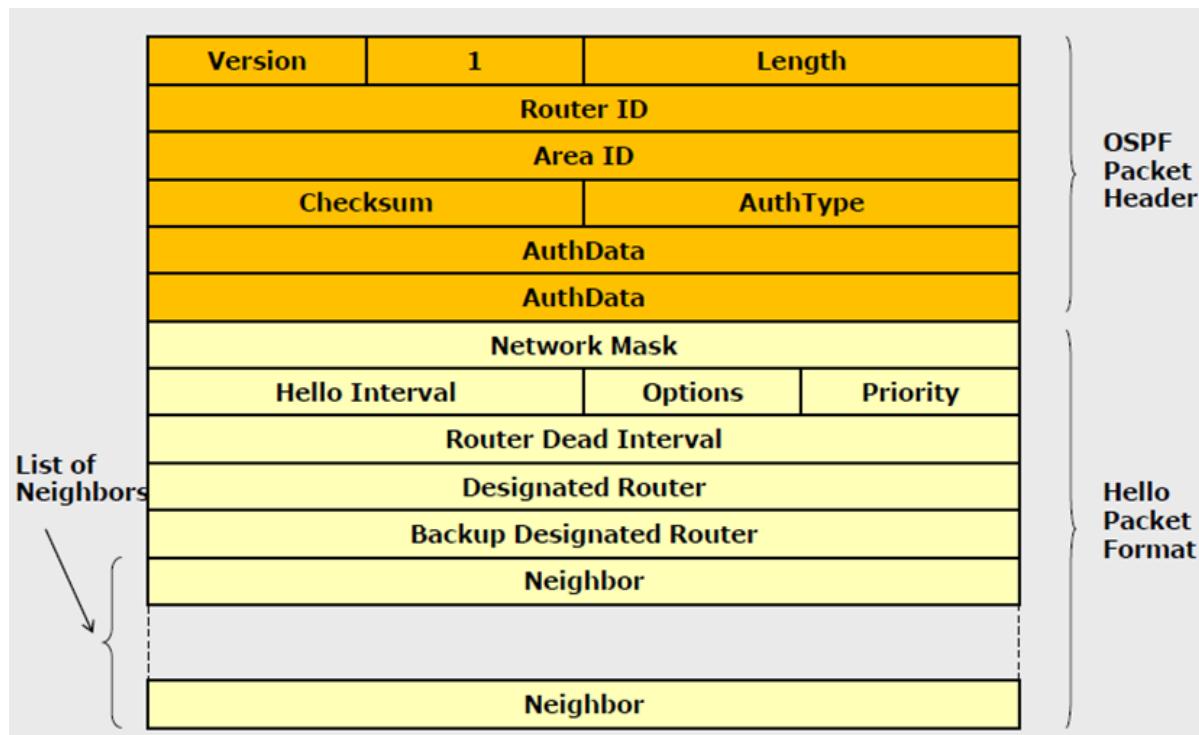
Nella figura sotto abbiamo il pacchetto di Hello, la parte arancione è l'header, in questo caso è il tipo 1.

Nella parte in giallo trovo il body.

Se decido di usare OSPF in una certa interfaccia, da questa interfaccia ogni 10 secondi mando una copia del pacchetto di Hello.

Ci scrivo la maschera che uso sull'interfaccia su cui ti sto trasmettendo questo pacchetto.

Io invio questo pacchetto nell'indirizzo multicast (224.0.0.5) e lo ricevono tutti i router che sono su quel link, perché se sono in un link punto-punto con un altro router, non si forma l'adiacenza se l'altro router non ha la stessa maschera.



Il pacchetto ha i seguenti campi:

- **Hello interval**, ogni 10 secondi invio un hello;
- **Router Dead Interval**, intervallo di 40 secondi dopo il quale se il destinatario non risponde capisco che non è più attivo;
- **Designated router**, router da cui ho ricevuto messaggi di hello su questa interfaccia, se non ricevo nulla vuol dire che sono collegato ad una stub network, è il router che viene eletto a master;
- **Backup designated Router** serve per implementare algoritmi di distribuzione tra router, è il master secondario, nel caso non funzioni il designated router (backup);
- **Priority** indica quanto un router vuole essere eletto master durante la sincronizzazione dei Database.

Sullo stesso link dobbiamo utilizzare tutti gli stessi valori di invio, e di scadenza, per evitare situazioni anomale.

Esempio figura sotto, supponiamo di avere un'unica area.

Sono su router C, e infatti nel pacchetto ho l'ID del RouterC che è 9.9.9.9.

Anche se non abbiamo una divisione in aree, in OSPF l'area di backbone deve esserci sempre, quindi l'area di backbone risulta essere tutta la rete, quindi tutti i router appartengono alla backbone.

Quindi come identificativo si mette 0.0.0.0.

Come maschera abbiamo 255.255.255.0, essendo una rete /24.

Come **hello interval** abbiamo 10 secondi e come **designated dead interval** abbiamo 40 secondi, (i tempi sono stabiliti così per l'efficienza, e a differenza di RIP, il tempo è minore perché il messaggio è molto più piccolo ed è molto più veloce da processare, in RIP se mandassi i messaggi ogni 10 secondi, non avrei abbastanza tempo per processare).

Come **neighbor** indica da chi ha ricevuto il messaggio di hello da quell'interfaccia (Se 0/0/0) ed è 1.1.1.1, ossia il RouterA.

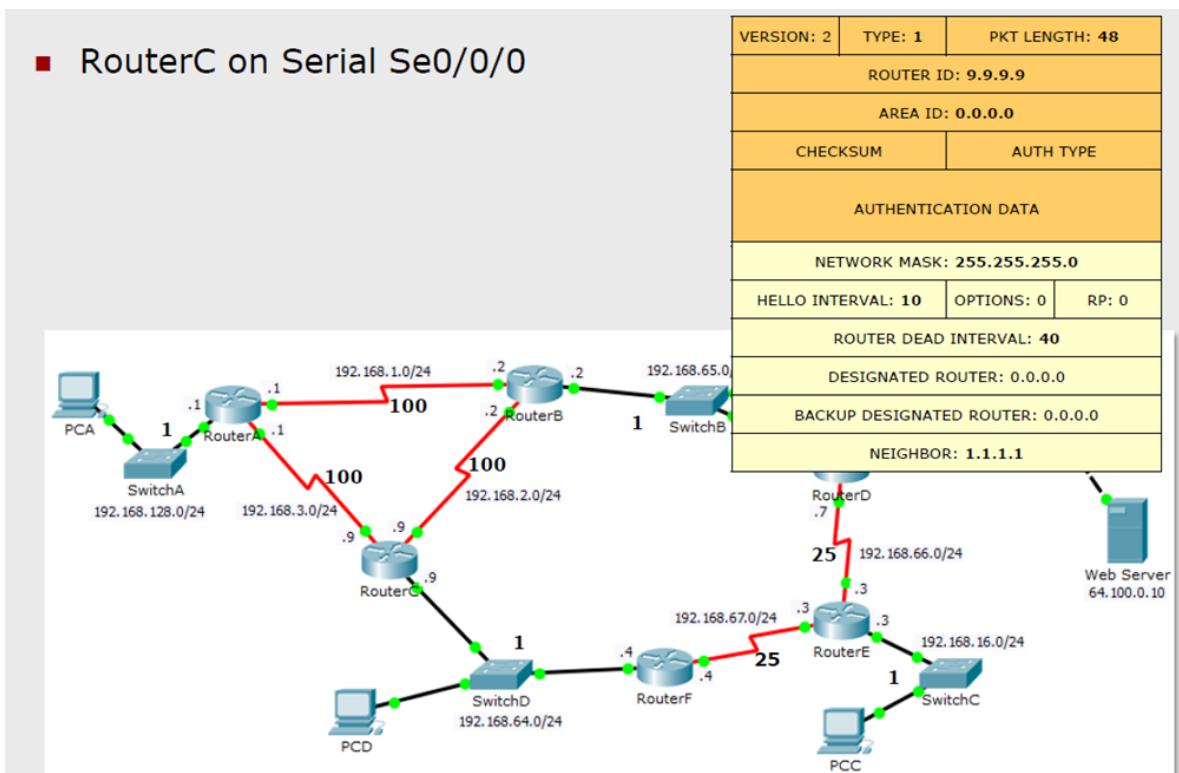
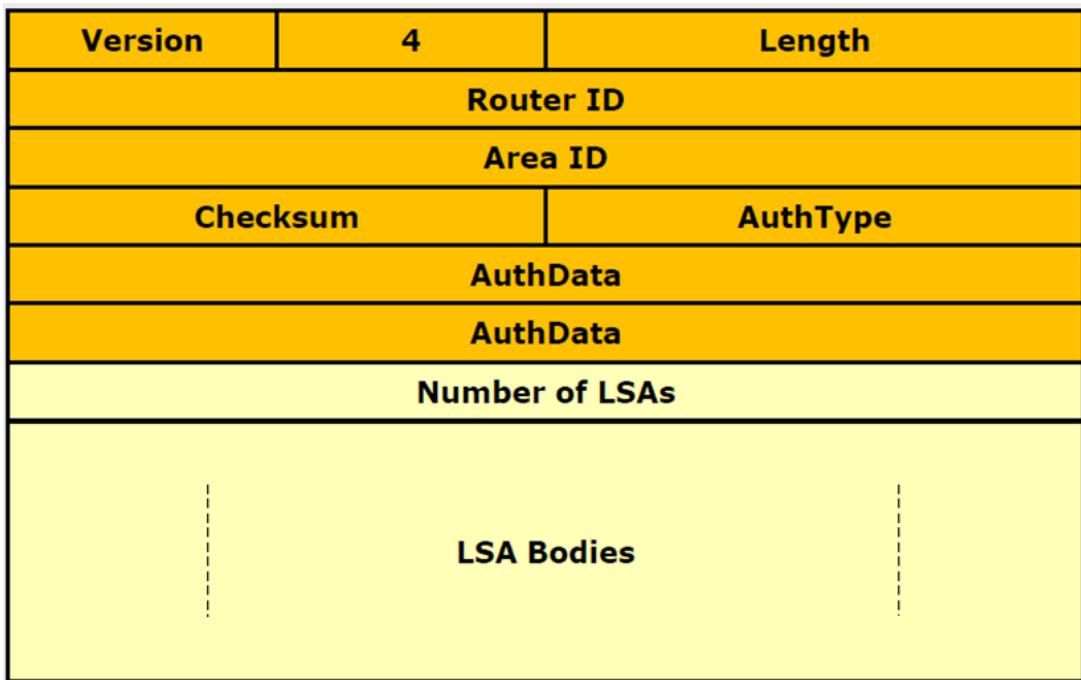


Figura sotto, il pacchetto di tipo 4 è quello che implementa il **selective flooding**, l'header è lo stesso, cambia la parte d'informazione (giallino), c'è una riga che indica quanti LSA sono contenuti nel pacchetto e a seguire ci sono gli LSA contenuti nel pacchetto, uno dopo l'altro.



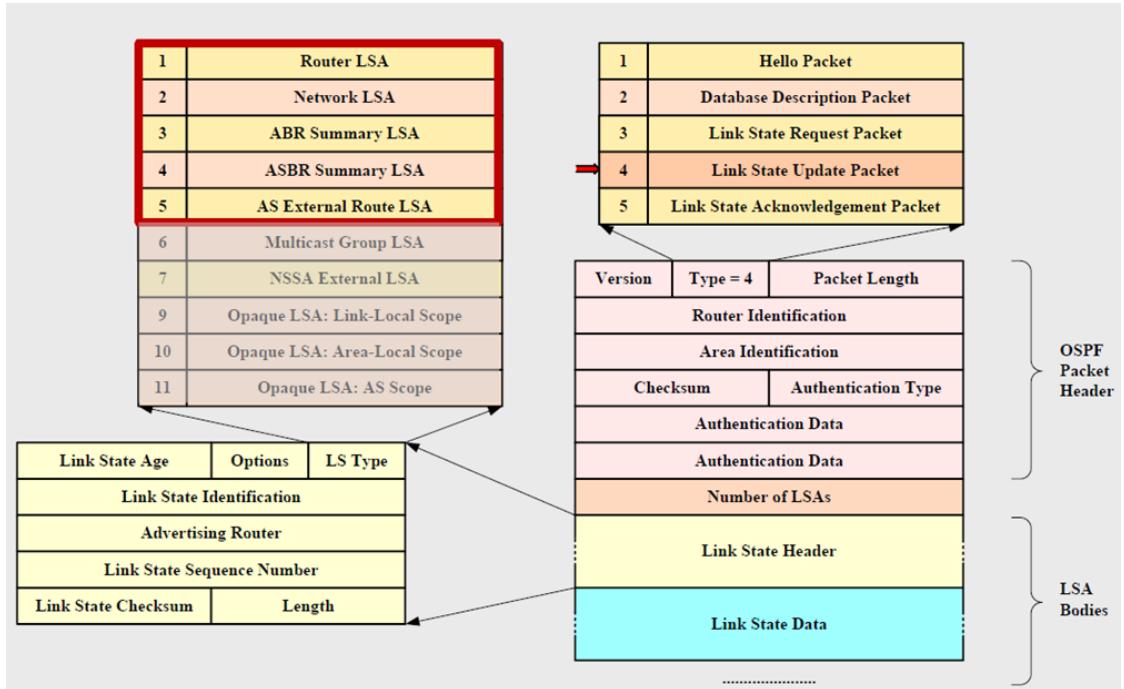
### Pacchetto LSA

Figura sotto, è lo stesso pacchetto e mostra che ciascun LSA a sua volta contiene un'intestazione(che identifica in modo univoco l'informazione) ed un campo dati.

Gli LSA sono oggetti che qualcuno ha generato e rappresentano un pezzo della topologia della rete.

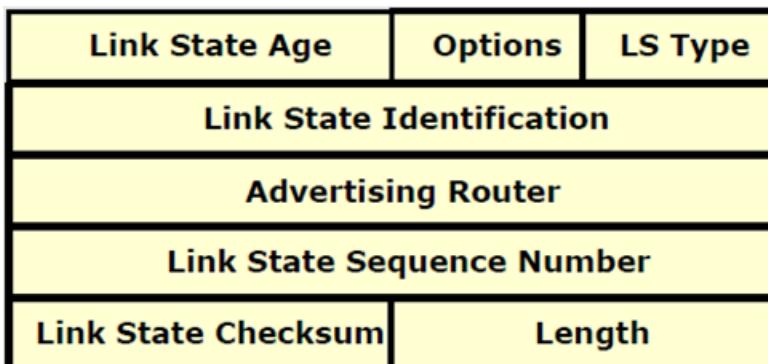
Sono inviati a tutti.

Nell'intestazione abbiamo un campo **LS type**, nel quale viene indicato il tipo di LSA che è stato propagato, questi tipi sono quelli in alto a sinistra, dove i primi 5 sono rilevanti per consentire d'imparare la topologia della rete, funzionare con più aree, e scambiare informazioni con l'esterno dell'AS.



L'intestazione è comune a tutti e i dati invece sono specifici a seconda del Link State Header.

Il tipo nell'header mi consente di specificare com'è fatto il campo dati del singolo LSA.



In figura sopra abbiamo il Link state header costituito da:

- **checksum**: per proteggere l'informazione del singolo LSA (gli LSA sono unità d'informazione indipendenti che per questioni di efficienza si mettono in un unico messaggio);
- **length**: lunghezza di tutto il pacchetto;
- **LS type**: visto sopra;
- **Link State Age**
  - Quando è stata prodotta questa informazione, serve ovviamente per capire la sua età.
  - Noi ad un certo punto copiamo una entry del database e la inviamo nel messaggio, noi nel DB copiamo LSA con header e dati)
  - Gli LSA scadono dopo 1 ora.

- Serve anche a capire se ho un'informazione non aggiornata, se ad esempio ho un LSA nel mio DB vecchio di 20 minuti e mi arriva una copia vecchia 10 minuti dello stesso LSA, viene aggiornato il database;
- **Advertising Router**, Il router che ha generato questo LSA, non è detto che coincida col router che ha inviato questo pacchetto (all'interno di un pacchetto ci sono diversi LSA), è per questo che anche qua abbiamo questo campo;
- **Link State Identification**, funge da chiave primaria del database assieme all'advertising router, e cambia in base al tipo;
- **Link State Sequence Number**, indica il numero di sequenza dell'LSA.

### OSPF - Irraggiungibilità

In RIP il valore di distanza 16 viene interpretato come infinito e serve dire che una certa destinazione è irraggiungibile.

In OSPF invece, l'informazione va cancellata dal DB.

Se aspetto 1 ora, il record LSA si cancella automaticamente, però se non voglio aspettare c'è un trucchetto:

Mando un LSA con sequence number aggiornato (quindi risulta nuovo), e come età metto 1 ora, in modo che subito dopo essere stato copiato si capisce che è scaduto, e quindi si elimina dal DB.

Quindi alla fine non rimane nulla nel DB.

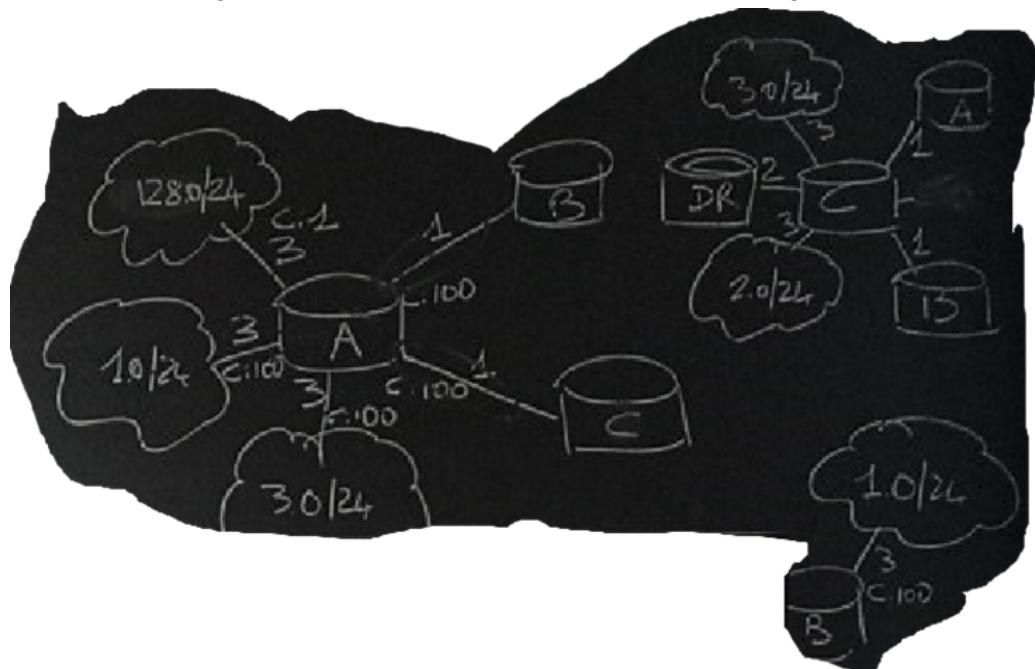
### Router LSA

Esempio figura sotto, ciascun router genera un solo LSA di tipo 1.

Cosa descrive ciascun router con questo LSA che si genera?

Descrive ciò che è adiacente a lui.

Cosa devo codificare in questo LSA in modo che un qualsiasi altro router della rete abbia tutto quello che gli serve per costruire la sua tabella di routing?



Essendo di tipo 1, il router A deve comunicare tutto ciò che ha intorno, quindi che è connesso alla rete 192.168.128.0/24, alla 192.168.1.0/24, e a 192.168.3.0/24.

Il router F però non sa la sequenza di router a cui deve mandare pacchetti per raggiungere il router A, è una cosa indipendente dal fatto che A può raggiungere quelle tre reti perché non è detto che i router collegati facciano da next-hop se fallisce l'adiacenza.

Nella figura vediamo che B e C sono adiacenti ad A.

A comunica che B e C sono i suoi vicini (ovvero sono adiacenti).

A dice che vicino a lui ci sono 5 link:

- I primi 3 sono le stub network e si codificano con il numero 3 sul link.
- Le altre 2 sono reti punto-punto con i router e si codificano col numero 1.

### **Costruzione di un LSA**

Quando il router B costruisce il suo LSA metterà nel suo elenco la rete 192.168.1.0/24.

Oltre al tipo (ovvero 3 o 1) viene indicato anche il costo che abbiamo nei vari percorsi.

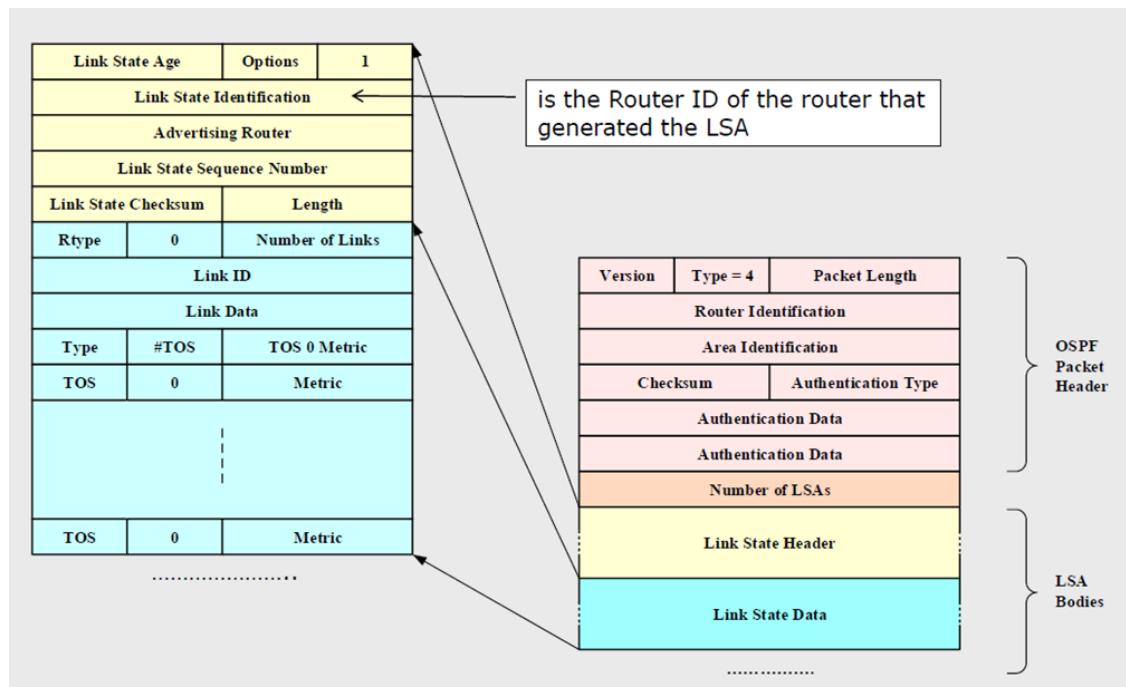
Il routerC, marcherà come 3 (stub network) le reti 192.168.3.0/24, 192.168.2.0/24.

Il routerC, marcherà come 1 (router punto-punto) i router A e B.

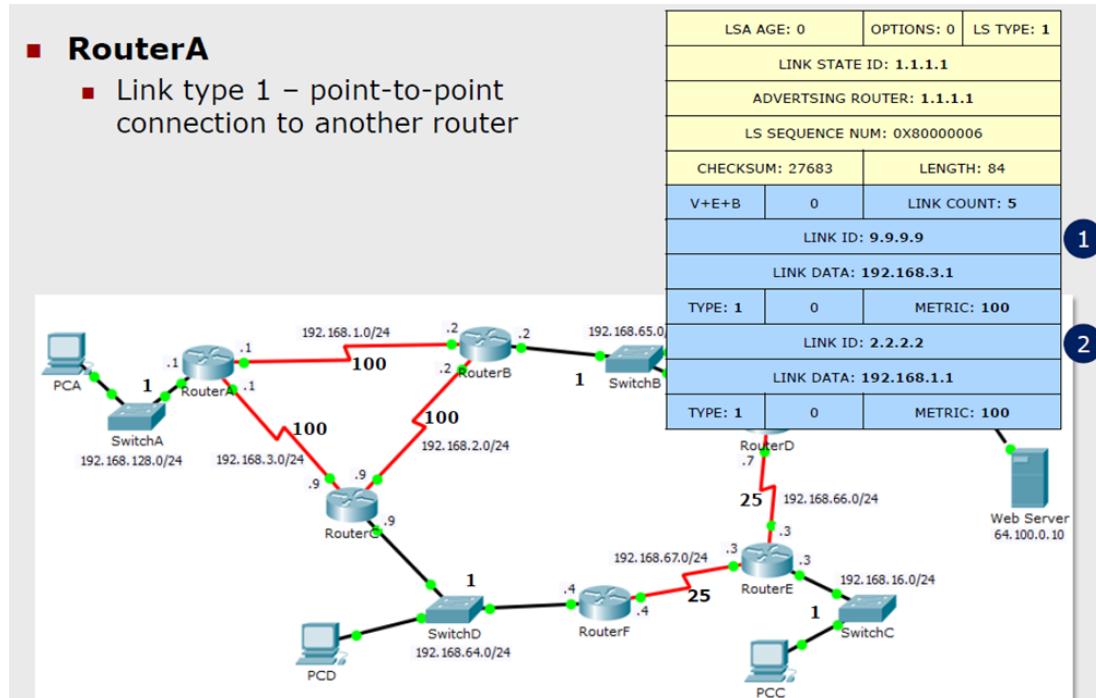
Il routerC ha un'interfaccia che riconosce come di tipo broadcast (interfaccia verso lo switch) alla quale possono essere collegati dei route.

In figura sotto, nel campo dati abbiamo un elenco di link di tipo diverso a seconda di cosa c'è dall'altra parte del link, abbiamo il **numero dei link** (nel caso di A abbiamo 5), c'è il **tipo di router** (c'è un bit per dire che sono un ABR, uno per l'AS boundary router, se non c'è niente sono un router interno), abbiamo il **link ID** e il **link data**, c'è il tipo, il campo **TOS (Type Of Service)**, per ogni classe devo indicare la **metrica**, se però in TOS metto 1, nella metrica metto **best effort**.

Quando il tipo è 1 basta leggere il tipo e l'advertising router per sapere che cos'è questo messaggio, essendo univoco per ogni router, il messaggio di tipo 1.

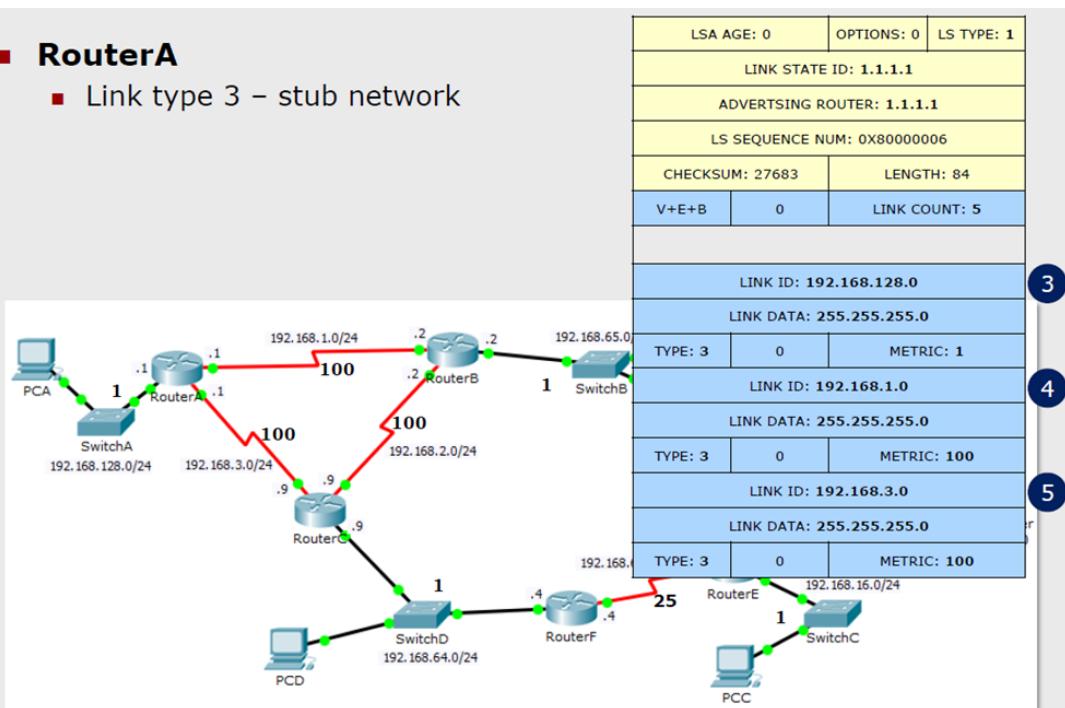


Nelle figure sotto, sono su router A e abbiamo i campi V+E+B (V sta per virtuale, se V=1 è un router virtuale, se B =1 è un ABR, se E =1 è un AS boundary router), il campo **data** contiene l'IP della mia interfaccia con cui raggiungo router C (nel caso del primo link), con cui raggiungo router B; invece, gli altri 3 sono di tipo 3, sono stub network, e come metrica abbiamo la metrica per arrivare a destinazione.



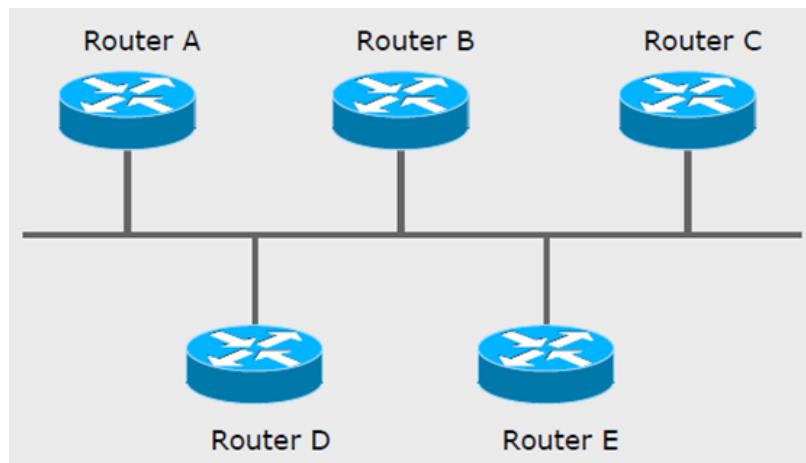
## ■ RouterA

- Link type 3 – stub network



## Link broadcast

Ho una sola interfaccia con cui posso raggiungere n router.



Il primo problema è il meccanismo del flooding (esempio figura sotto) se sono il router E, e ricevo un LSA da qualcuno che non è uno di quei router, essendo un link broadcast in un colpo solo lo invio ai miei vicini.

Il router C lo riceve, aggiorna il DB e lo invia ai suoi vicini nel link broadcast, ma che sono gli stessi che lo avevano appena ricevuto dal router E.

Bastava solo la sua trasmissione, ma ne ho avute molteplici dello stesso LSA.

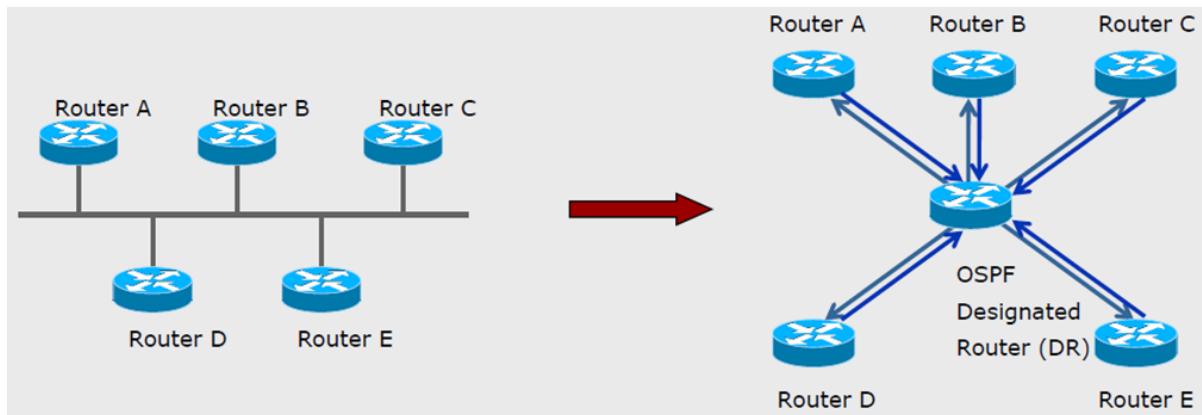
Il secondo problema è che Dijkstra non è ottimale, se rappresento questa rete con collegamenti punto-punto tra coppie di router, questa cosa viene rappresentata come una rete a maglia completa, quindi l'algoritmo vedrebbe  $(5 * 4) = 20$  collegamenti.

Aumento inutilmente il tempo di processazione dell'algoritmo.

Inutilmente perché in realtà la topologia della rete è molto più semplice di come sembra.

## Router Virtuale (Designated Router)

Allora si risolve aggiungendo un **router virtuale**, e la topologia diventa come quella in figura:



Dal punto di vista del router D, quello adiacente è solo il router virtuale, quindi gli LSA li manda solo a lui, e lui li manda a tutti gli altri.

Ciascun router vede una topologia più semplice.

Si occupa di gestire la sincronizzazione dei Database dei router nella rete.

Nello specifico, ogni router si sincronizza con il designated router, quindi per la proprietà transitiva, tutti sono sincronizzati alla fine.

### “Elezioe” del Router Virtuale

Il router virtuale viene impersonificato da uno dei router reali della rete originaria.

E' necessario il processo di elezione del router virtuale.

Ci si scambiano dei numeri, se io voglio diventare quello eletto, mando il numero massimo, se non voglio essere eletto invio il numero più piccolo.

Se accade che due numeri sono uguali, si usa l'identificatore univoco del router, si elegge quello con numero ID maggiore.

### Designated Router e Backup Designated Router

Figura sotto, ciascun router nel link broadcast (rete 192.168.64.0/24), essendo tutti nello stesso link mette gli IP per identificarsi e mette il **designated router** e anche il secondo (**backup designated router**).

Questo router serve nel caso si guasti il designated router.

Quando tutti sono d'accordo, ovvero hanno le stesse informazioni sul designated router uguali allora si può comunicare.

C'è il modo per andare sui router ed indicare a mano che è collegato ad un link broadcast.

## ■ RouterC on Fa0/0

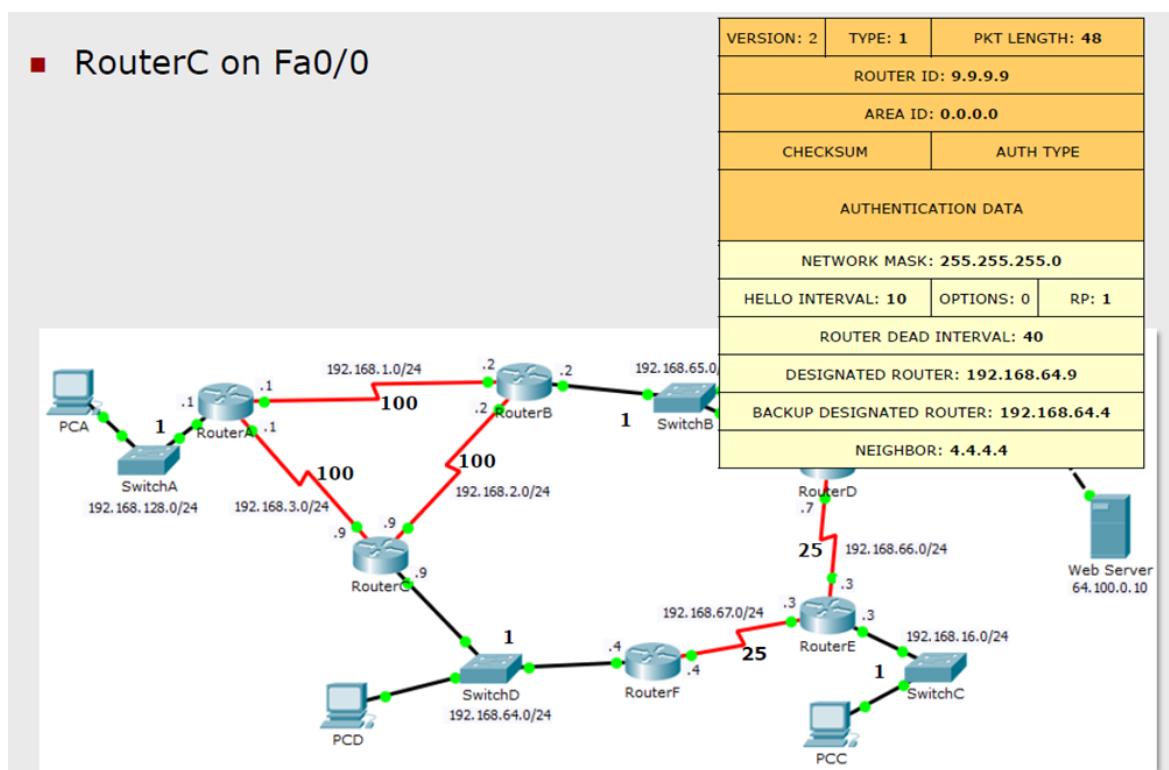
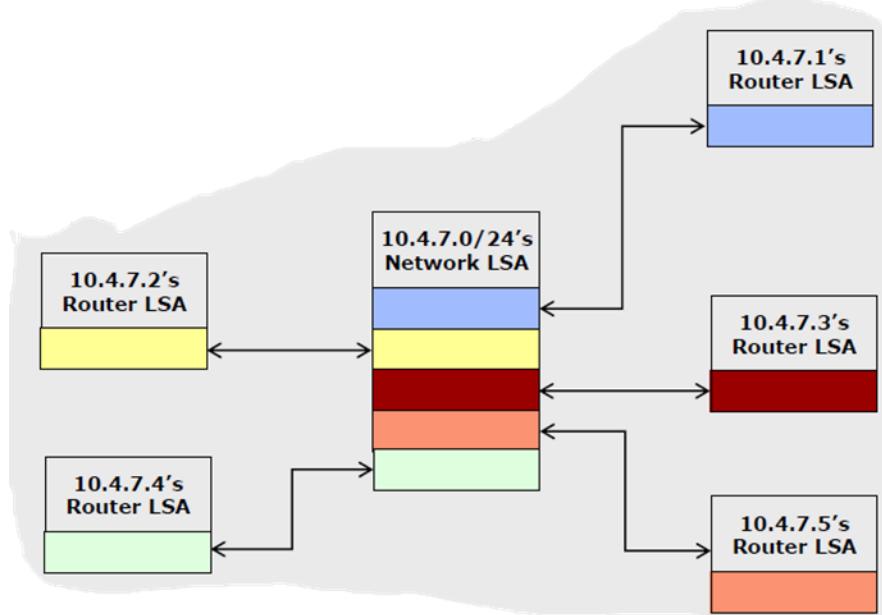


Figura sotto, questa informazione come viene rappresentata nel LSA?

Ciascun router nell'LSA indica il designated router a cui è connesso.

Il designated router invia un “network LSA” dove indica i vari router connessi, ovvero descrive tutti i link verso tutti gli altri.

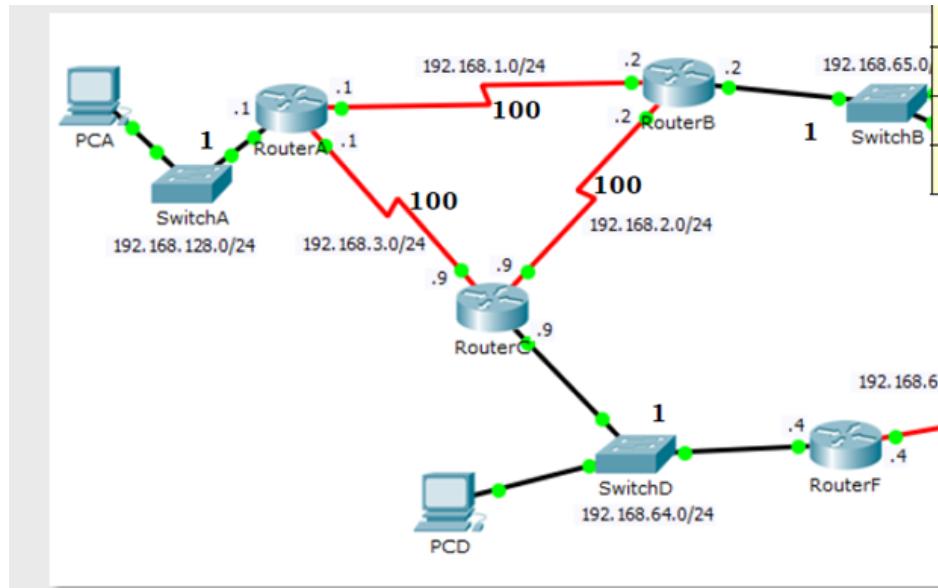


### Pacchetti LSA - Tipo di Link

Il router LSA (Tipo 1) serve a descrivere i link a cui il router è connesso.

Ogni link ha un tipo:

- **Tipo 1:** Link punto-punto verso un altro router, specifica anche l'IP dell'interfaccia del Router che comunica l'informazione e la matrica (quanto costa trasmettere su questo link).
- **Tipo 2:** Indica una rete di transito, ossia un link a cui possono essere connessi più di due router.
  - In questo tipo di rete occorre eleggere il designated router.
- **Tipo 3:** Link identificati da un indirizzo di Rete e da una maschera e sono Link su cui mi aspetto di esserci una stub network, quindi destinazioni finali (Host), indica anche la metrica.



routerC a quanti link è connesso? 5 link.

1. punto-punto verso Router A → Tipo 1.
2. punto-punto verso Router B → Tipo 1.
3. stub verso 192.168.3.0 → Tipo 3.
4. stub verso 192.168.2.0 → Tipo 3.
5. Transito verso 192.168.64.0 → Tipo 2.

Nella rete di esempio (slide 48) che tipo di LSA produce il routerC?

Il routerC è connesso direttamente a due router (routerA e routerB), quindi ho due link di tipo 1.

Il routerC è connesso ad una stub network, quindi ho un link di tipo 3? No.

**Come capisce che un link è di tipo 1?**

La tipologia di interfaccia è seriale.

**Come capisce che un link è di tipo 2?**

Se il link è di tipo broadcast (Ethernet) e se riceve almeno un Hello di altri router da esso.

Da quella interfaccia però, il routerC riceve gli Hello dal routerF.

E' una rete di transito perché l'interfaccia è Ethernet, quindi di tipo Broadcast.

Il link poi è di tipo 2 perché a quel link è connesso anche il routerF, il quale può ricevere un pacchetto da routerC e mandarlo altrove, quindi il link è di tipo 2 perché da quel link i pacchetti possono entrare e poi uscire da un altro router.

### Come capisce che un link è di tipo 3?

Se il link è di tipo broadcast (Ethernet) e se non riceve Hello di altri router da esso. Nelle stub network, i pacchetti nascono (quindi escono e non rientrano più) o finiscono la corsa (entrano e non escono più).

### Quando viene eletto il Designated Router?

Dato che ho un link di tipo 2, occorre eleggere un designated router.

### network LSA (Tipo 2)

Il designated router invia ai router attaccati al mezzo broadcast una LSA di tipo 2 (network LSA).

LSA tipo 2 ha lo stesso scopo di un LSA di tipo 1 ma è adatto a mezzi di tipo broadcast.

Dove è mostrata la maschera della rete (Link State ID) e la lista degli indirizzi IP dei router attaccati ad essa.

Il router che manda questo network LSA è Advertising Router.

La metrica (costo per trasmettere verso il designated router) non è mostrata, perché implicitamente è quella minima.

I router non designated ricevono questa cosa come se fosse un LSA di tipo 1 e non la inoltrano.

LSA AGE: 0	OPTIONS: 0	LS TYPE: 2
LINK STATE ID: 192.168.64.9		
ADVERTISING ROUTER: 9.9.9.9		
LS SEQUENCE NUM: 0X80000001		
CHECKSUM: 45610	LENGTH: 32	
NETWORK MASK: 255.255.255.0		
ATTACHED ROUTER: 4.4.4.4		
ATTACHED ROUTER: 9.9.9.9		

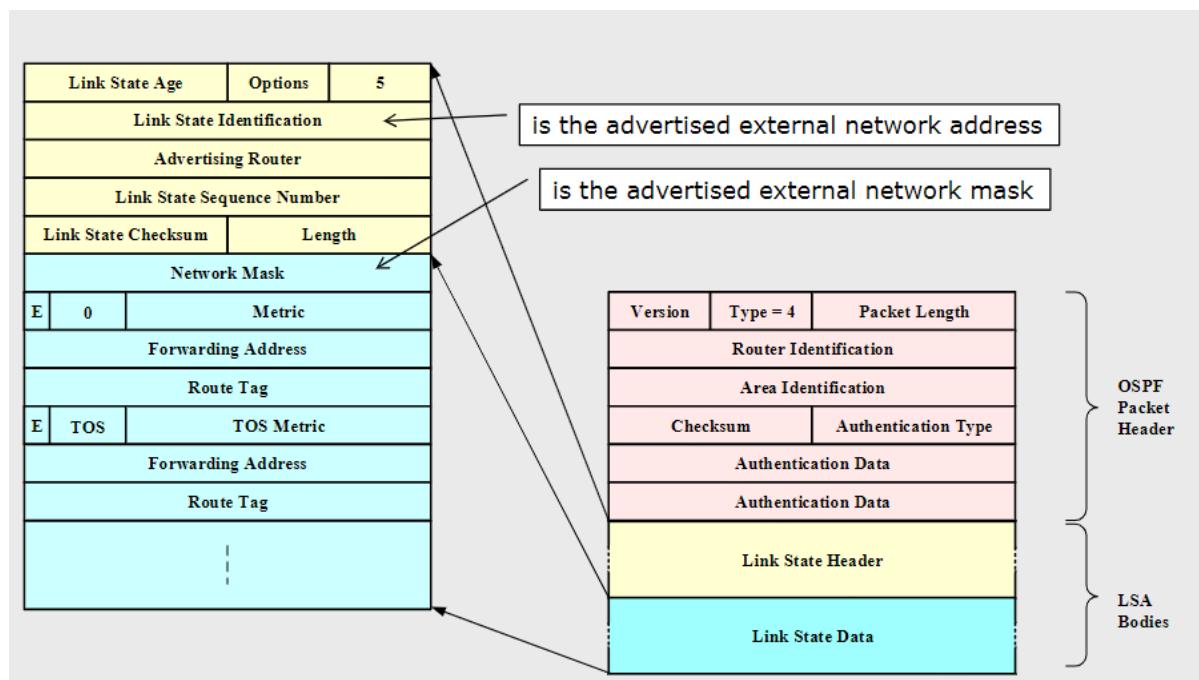
### AS External Link LSA (Tipo 5)

Generati dai router ASBR (router che hanno un'interfaccia all'esterno dello AS) per pubblicizzare reti esterne allo AS.

Propagati con selective flooding dentro allo AS come un normale router LSA.

Oltre alla rete esterna è indicato anche il costo (il costo non è sempre significativo).

Un LSA di tipo 5 ha lo stesso schema di un LSA di tipo 1, ma con qualche differenza:



Importante è il bit “e”, che serve ad interpretare la metrica.

La metrica va interpretata perché la rete è esterna allo AS.

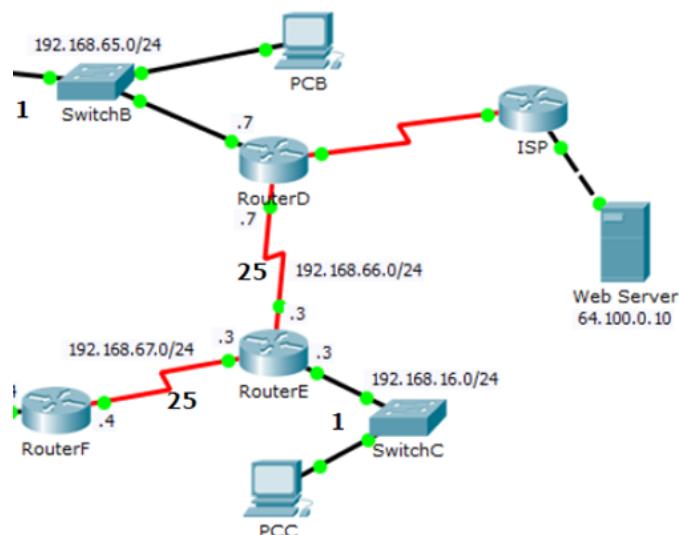
Dentro la AS i costi sono tutti omogenei, si ottengono sommando i costi dei link espressi con la stessa unità di misura.

LSA di tipo 5 serve a dire a tutta la rete che tramite lui si può raggiungere la rete esterna indicata in Forwarding Address.

Ma tra AS diverse, la definizione di costo potrebbe cambiare, magari si passa da Numero di Hop a metri o miglia.

Il bit “e” serve a dire se la metrica annunciata è consistente alla metrica interna allo AS o meno.

Se e = 1, la metrica espressa nel pacchetto LSA di tipo 5 non è consistente.



Advertising Router è il router interno allo AS che consente di arrivare alla rete esterna, nell'esempio è il Router D.

### OSPF in Laboratorio - 6.1

```
C      192.168.1.0/24 is directly connected, Serial0/0/0
O      192.168.2.0/24 [110/200] via 192.168.1.2, 00:07:08,
Serial0/0/0
                                         [110/200] via 192.168.3.9, 00:07:08,
Serial0/0/1
C      192.168.3.0/24 is directly connected, Serial0/0/1
O      192.168.16.0/24 [110/127] via 192.168.1.2, 00:06:33,
Serial0/0/0
                                         [110/127] via 192.168.3.9, 00:06:33,
Serial0/0/1
O      192.168.64.0/24 [110/101] via 192.168.3.9, 00:06:33,
Serial0/0/1
O      192.168.65.0/24 [110/101] via 192.168.1.2, 00:06:33,
Serial0/0/0
O      192.168.66.0/24 [110/126] via 192.168.1.2, 00:06:33,
Serial0/0/0
O      192.168.67.0/24 [110/126] via 192.168.3.9, 00:06:33,
Serial0/0/1
C      192.168.128.0/24 is directly connected, FastEthernet0/0
O*E2 0.0.0.0/0 [110/1] via 192.168.1.2, 00:06:33, Serial0/0/0
```

Con `show ip route`, abbiamo le reti connesse al router A.

Oltre alle reti Direttamente Connesse C, vediamo anche le reti pubblicizzate O le posso raggiungere da next-hop diversi.

```
192.168.2.0/24 [110/200] via 192.168.1.2, 00:07:08, Serial0/0/0
                                         [110/200] via 192.168.3.9, 00:07:08, Serial0/0/1
```

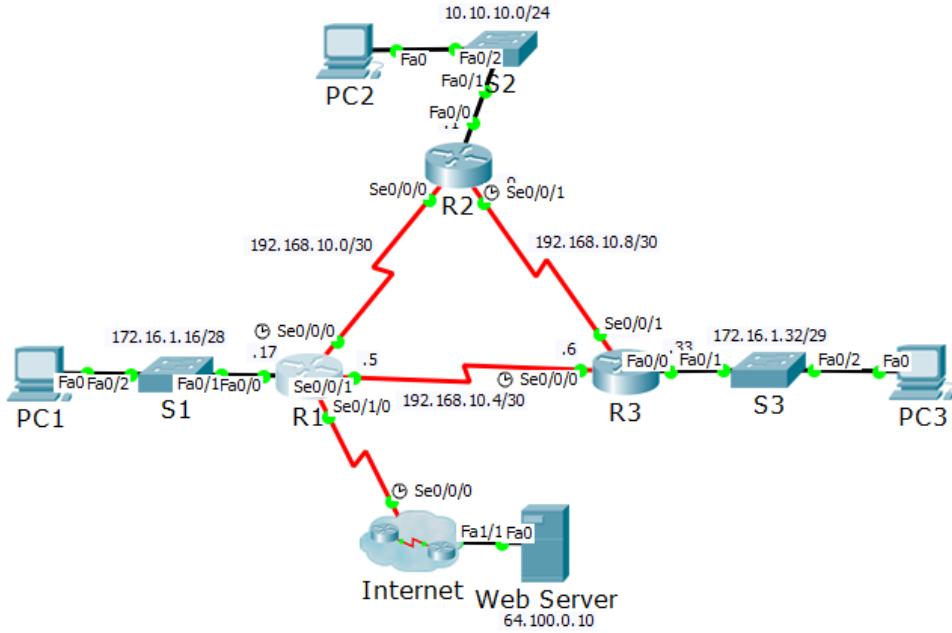
Vedo che con OSPF i tempi delle righe superano i 6 minuti, in RIP al più 30 secondi.  
A differenza di RIP, OSPF aggiorna le righe ogni 30 minuti (se non cambia la topologia).

Le righe con O sono quelle messe da OSPF.

La riga con O\*E2 significa “OSPF DEFAULT EXTERNAL TYPE 2”, ossia indica la riga di default ricevuta con OSPF.

Il E2 significa “Flag e Settato”.

### OSPF in Laboratorio - 7.1



Il router 1 è un ASBR, perché ha un link con internet.

La versione 2 di OSPF usa indirizzi classless.

Come dico al router di usare OSPF?

```
R1(config)#router ospf 1
```

1 è il PID che il processo all'interno del router 1 avrà non appena premiamo "invio".

La versione di default è la versione 2, quella classless.

La prima cosa che devo fare è conoscere i vicini, e quindi inviare i messaggi di Hello.  
Ma per farlo devo avere un router ID.

### Impostare il Router ID

La prima cosa è assicurarsi che il router abbia un ID.

La cosa più semplice è eseguire un comando per assegngarglielo manualmente.

Se non lo facciamo si arrangia da solo (*opzione non consigliata*).

Molto importante notare che l'ID poi non può essere cambiato in "run time", altrimenti potrei danneggiare l'operato di OSPF su tutta la rete, nello specifico: per un pò di tempo risultano esserci 2 router che in realtà sono lo stesso router, ma che virtualmente risultano essere 2. Solo che uno dei 2 router non può ricevere nulla e ciò che gli viene mandato è perso.

Il comando (quello che useremo all'esame e nei laboratori) per assegnare un ID manualmente è:

```
R1(config-router)#router-id 1.1.1.1
```

Dove #num è un intero a 32 bit.

Prima di continuare, assegno un ID a tutti i router e poi solo dopo avvio OSPF in ogni router, in modo da minimizzare il tempo in cui ci sono alcuni router che stanno facendo girare OSPF correttamente e gli altri ancora “immobili”.

### Interfaccia di LoopBack

Un altro modo per dare l'ID al router, è quello di dare un indirizzo all'interfaccia di loopback del router.

```
R3(config-router)#interface Loopback 0
```

L'interfaccia di loopback, è un modo per inviare messaggi a se stesso.

Nei pc tradizionali conosciamo l'indirizzo di loopback del tipo 127.0.0.0/8.

L'interfaccia di loopback è virtuale, in realtà non esiste, ma gli assegno un indirizzo e il router la tratta come un'interfaccia qualunque.

Essa è utilizzata sia per inviare messaggi a se stessi, ma anche per ricevere pacchetti come tutte le altre interfacce.

```
R3(config-if)#ip address 3.3.3.3 255.255.255.255
```

La rete ha come maschera tutti 1, perché c'è solo un indirizzo, ossia l'interfaccia del router.

Ora il router ha un interfaccia attiva Loopback con indirizzo 3.3.3.3 255.255.255.255, se riceve un pacchetto a quella interfaccia, lo tratta come sempre.

Il pacchetto viene inviato a questa interfaccia e per come è fatta l'interfaccia di loopback, esso viene consegnato al router stesso.

Con `show ip route` vedo:

```
3.0.0.0/32 is subnetted, 1 subnets
C      3.3.3.3 is directly connected, Loopback0
```

Perché uso la loopback?

Se usassi una delle interfacce effettivamente vere ho uno svantaggio, se va giù, il router diventa non più raggiungibile.

La loopback non va mai giù perché non è fisica e quindi non è soggetta ai problemi che ne derivano.

La loopback deve essere pubblicizzata come le altre interfacce.

### Come il Router determina il suo Router ID

1. Prendo l'ID assegnato con il comando `router-id`.
2. Se non è stato configurato manualmente, prendo l'indirizzo IP più alto tra le sue interfacce di LoopBack (la pratica più comune).
3. Se non ha interfacce di LoopBack, prende l'indirizzo IP più alto tra le sue interfacce fisiche (*questa opzione non va bene, va evitata*).

## Configurare le interfacce di OSPF

Indico al router su quali interfacce deve usare OSPF.

Per dire a OSPF di usare una interfaccia si usa, come in RIP, il comando `network`.

```
R1(config-router) #network 172.16.1.16 0.0.0.15 area 0
```

### Nota: wildcard-mask

Al posto della subnet-mask, si inserisce una `wildcard-mask`.

La wildcard-mask si ricava facilmente dalla subnet-mask: **basta prendere la subnet-mask della rete a cui appartiene l'indirizzo IP dell'interfaccia di riferimento, e negare tutti i 32 bit.**

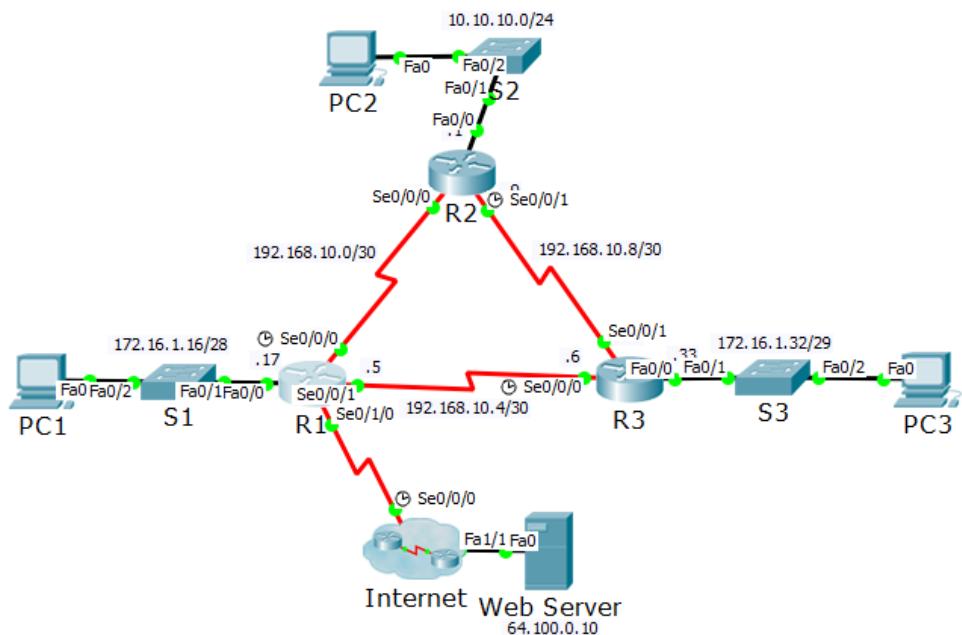
### Nota: area #num

Serve per specificare a quale area della rete appartiene l'interfaccia del router.

Devo specificare questa cosa perché è una informazione che può sapere solo il progettista della rete e perché (come detto prima) OSPF è un algoritmo che (di norma) gestisce il routing solo all'interno di una determinata area.

Se in un esercizio è specificato che la rete non è divisa ad aree (*single area*) allora specifico che ogni interfaccia di ogni router si trova sempre nella stessa area (ossia *ho un'unica grande area che guarda caso include tutta la rete*).

## Pratica: Configurare una rete con OSPF



Io ho usato questi comandi e la simulazione di Packet Tracer ha mostrato che tutti gli Host riuscivano a pingare qualsiasi altro host, anche il server Esterno.

Configuriamo R1 con il metodo più "grossolano".

```

R1(config)#router ospf 1
R1(config-router)#router-id 1.1.1.1
R1(config-router)#default-information originate // Propago la Default
Route
R1(config-router)#network 172.16.1.16 0.0.0.15 area 0
R1(config-router)#network 192.168.10.0 0.0.0.3 area 0
R1(config-router)#network 192.168.10.4 0.0.0.3 area 0
R1(config-router)#exit
R1(config)#ip route 0.0.0.0 0.0.0.0 Serial0/1/0 // Default Route
R1(config)#exit
R1#copy r s

```

La routing table di R1 sarà del tipo:

```

      10.0.0.0/24 is subnetted, 1 subnets
O          10.10.10.0 [110/65] via 192.168.10.2, 00:00:30,
Serial0/0/0
      172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C          172.16.1.16/28 is directly connected, FastEthernet0/0
O          172.16.1.32/29 [110/65] via 192.168.10.6, 00:00:30,
Serial0/0/1
      192.168.10.0/30 is subnetted, 3 subnets
C          192.168.10.0 is directly connected, Serial0/0/0
C          192.168.10.4 is directly connected, Serial0/0/1
O          192.168.10.8  [110/128] via 192.168.10.2, 00:00:30,
Serial0/0/0
                           [110/128] via 192.168.10.6, 00:00:30,
                           Serial0/0/1
      209.165.200.0/30 is subnetted, 1 subnets
C          209.165.200.224 is directly connected, Serial0/1/0
S*        0.0.0.0/0 is directly connected, Serial0/1/0

```

Configuriamo R2, però stavolta uso il metodo “chirurgico”.

```

R2(config)#router ospf 1
R1(config-router)#router-id 2.2.2.2
R2(config-router)#network 192.168.10.2 0.0.0.0 area 0
R2(config-router)#network 192.168.10.9 0.0.0.0 area 0
R2(config-router)#network 10.10.10.1 0.0.0.0 area 0
R2(config-router)#exit
R2(config)#exit
R2#copy r s

```

#### **Nota: “Metodo Chirurgico”**

Per “*Metodo chirurgico*” intendo che nel comando `network` specifico l’indirizzo esatto dell’interfaccia del router e metto `0.0.0.0` come wildcard-mask.

Il valore della wildcard-mask è messo in modo tale da specificare una rete composta da un solo indirizzo.

Il problema del metodo “*grossolano*” è che se metto nel comando `network` un indirizzo IP e una mask troppo generiche rischio di includere anche interfacce di rete che in realtà non vorrei fossero pubblicizzate da OSPF.

Invece con il metodo “*chirurgico*” non rischio di creare questa ambiguità.

Dopo qualche secondo, vediamo che R1 ha incontrato R2, ossia è avvenuta l’adiacenza.

```
00:34:49: %OSPF-5-ADJCHG: Process 1, Nbr 1.1.1.1 on Serial0/0/0  
from LOADING to FULL, Loading Done
```

Configuriamo R3 sempre con il metodo “*chirurgico*” (*Io preferisco perché è più difficile sbagliare*).

```
R3(config)#router ospf 1  
R3(config-router)#router-id 3.3.3.3  
R3(config-router)#network 192.168.10.6 0.0.0.0 area 0  
R3(config-router)#network 192.168.10.10 0.0.0.0 area 0  
R3(config-router)#network 172.16.1.33 0.0.0.0 area 0  
R3(config-router)#exit  
R3(config)#exit  
R3#copy r s
```

#### **Nota: Reset di OSPF**

Per resettare OSPF digito:

```
R3#clear ip ospf 1
```

Con questo comando solo il router da OSPF (drop del database e reset di ogni setting di OSPF), gli altri router ci metteranno un pò per capire che questo router non esiste più, pulisce tutto lo stato del processo OSPF con pid 1, il router non partecipa più al protocollo OSPF, a questo punto può essere riavviato con un ID diverso (questo è l’unico modo per cambiare l’ID di un router, se si utilizza di nuovo il comando `router-id` non funziona).

#### **Nota: Modifica dei Timer di OSPF**

Posso modificare anche i timer di OSPF delle singole interfacce (operazione sconsigliata, se configurati male possono accadere fenomeni anomali nel transitorio):

```
R3(config-if)#ip ospf hello-interval seconds  
R3(config-if)#ip ospf dead-interval seconds
```

#### **Nota: Mostrare la Lista dei Vicini**

Per ogni router posso visualizzare tutti i suoi vicini:

```
R3#show ip ospf neighbor
```

**Definizione:** Se almeno una interfaccia non invia messaggi di Hello, OSPF lo segna come ASBR.

#### **Verifica di OSPF**

Vediamo come assicurarci che OSPF sia impostato in modo corretto su un router.

## **OSPF - Tabella di Routing - Caratteri delle Righe**

Ogni riga ha un carattere che mostra delle caratteristiche della riga stessa.

- ○ → Destinazione appresa tramite OSPF intra-area.
- IA → Destinazione appresa tramite OSPF inter-area.
- N1 & N2 → Destinazioni per reti o aree segnalate come speciali.
  - *In questo corso non incontreremo casi del genere.*
- R1 & R2 → Destinazione appresa tramite algoritmi o metodi diversi da OSPF ma comunque distribuita con OSPF.
  - *Se due router mi danno due destinazioni uguali, uso la distanza amministrativa, perché non posso confrontare la metrica di due algoritmi diversi.*

## **OSPF - Segnare i Boundary Routers (ASBR) della rete**

Definizione: Se almeno un interfaccia del Router non è impostata per inviare messaggi di Hello, OSPF lo segna come ASBR.

Un router viene eletto come ASBR in modo automatico da OSPF stesso,  
La definizione è fatta in questo modo perché le interfacce rivolte verso internet non inviano hello.

### **Comando R1#show ip protocols**

Mi da informazioni complessive sullo stato di esecuzione dei protocolli di routing.

### **Comando R1#show ip ospf**

Mostra il Router ID.

Mostra anche SPF schedule Delay 5 seconds.

"*Ogniqualvolta ricevo un LSA che mi segnala un cambio di topologia aspetterò 5 secondi.*

### **Nota: SPF schedule delay**

Questa attesa è una forma di protezione verso cambi di topologia dovuti a glitch (fenomeni temporanei) che potrebbero essere contraddetti tra pochissimo tempo da una informazione giusta.

Come in RIP, a fronte di cambi di topologia vengono inviati LSA a prescindere dal timer di invio degli LSA.

Questa cosa ci permette di non eseguire inutilmente Dijkstra e quindi tenere occupata la CPU del router e quindi introdurre ritardi nella rete.

L'attesa di 5 secondi può essere modificata (in modo opportuno, se troppo grande la rete diventa poco reattiva ai cambi di topologia, se troppo piccola la rete diventa troppo reattiva e comunque perdere tempo con computazioni inutili).

### **Comando R1#show ip ospf interface Fa0/0**

Mostra come OSPF lavora su quella particolare interfaccia.

- Indirizzo IP e area ID.
- Timers di OSPF.
- PID del processo OSPF che lavora su quella interfaccia.
- Costo usato da OSPF per calcolare i percorsi migliori su questa interfaccia.
  - *Su un router posso avere più processi OSPF in esecuzione contemporaneamente (con PID diverso ovviamente), ma una interfaccia può essere gestita solo da un processo OSPF.*
  - *Avere due processi OSPF contemporaneamente può servire per dividere i domini di routing.*
- Designated Router su questa interfaccia.
- Come scambia messaggi di Hello, ossia i 2 timer (hello e dead)
- Se passiva o meno ( se è passiva, su questa interfaccia non vengono inviati Hello).

Mostra i vicini.

Mostra tutti i processi OSPF in esecuzione su un router.

#### **Comando R1#show ip ospf database**

Mostra il contenuto del database OSPF del router.

Ossia tutta la lista degli LSA ricevuti dal router, ogni LSA è identificato da un ID (contenuto nell'header del pacchetto LSA).

**Il Database è lo stesso in tutti i router che appartengono alla stessa area (stato di convergenza).**

Ciò che mi mostra il comando è la lista degli LSA (non il contenuto, mostra solo la chiave e il contenuto dell'header).

L'elenco è diviso per Tipo di LSA (tipo 1, tipo 2 ...).

**Di LSA di tipo 1 (Router) me ne aspetto uno solo per ogni router della rete.**

Nella lista c'è anche LSA di tipo 1 del router stesso dove eseguo il comando.

Age e Sequence Number (Seq#, espressa in secondi) sono due campi contenuti nell'header dell'LSA.

Il sequence number esprime quante volte è stato aggiornato quell'LSA.

**Link ID (router ID del router che ha generato questo LSA) e ADV Router (router ID del router che ha inviato questo LSA) sono i campi che costituiscono la chiave dell'LSA.**

Se la Age di quello appena ricevuto è minore aggiorna la riga del Database.

Nell'header c'è anche la lunghezza in byte, ma non è rilevante per identificare questo LSA.

Nel body dello LSA c'è il Link Count, è mostrato anche nel database.

Link Count dice quanti link sono annunciati del Router con questo LSA di tipo 1 (ossia quanti link di tipo 1, 2 o 3).

#### **Comando R1#show ip ospf database router (LSA di tipo 1)**

Il comando mostra il contenuto di ciascun LSA di tipo 1 contenuto del Database.  
Nelle prime righe mostra in dettaglio il contenuto di ogni campo del header.

Nel dettaglio:

Rtype ci sono dei flag dove il router specifica che tipo di router è (ASBR o altro).

A seguire vengono mostrati i link pubblicizzati nell LSA, per ogni link viene visualizzato:

- **Tipo di Link.**
  - Tipo 1: “*Another Router*”
  - Tipo 2: “*a Transit Network*”
  - Tipo 3: “*a Stub Network*”
- **Link ID:** indirizzo IP della rete pubblicizzata.
- **Link Data:** la maschera della rete.
- **Number of ToS (#ToS):** ???

**Comando R1#show ip ospf database external (LSA di tipo 5)**

Il comando mostra il contenuto di ciascun LSA di tipo 5 contenuto del Database.

Ossia LSA relativi a destinazioni esterne all'area di lavoro.

Le reti esterne sono indicate con E# nella tabella di routing.

- E1 → Rete esterna con metrica significativa (*bit e spento*).
- E2 → Rete esterna con metrica non significativa (*bit e acceso*).

Contenuto:

- Link state ID è l'indirizzo IP della rete (la rete di default ha 0.0.0.0).
- Advertising Router è il router che sta annunciando la rete esterna.
- Metric Type:
  - 2 → “Distanza Infinita”, ossia è considerato più distante di qualunque metrica all'interno dell'area.

#### **Nota: Metric Type**

In RIP la metrica è il numero di hop e il valore 16 è infinito; In OSPF la metrica è un costo e non ha un valore massimo, quindi l'infinito è rappresentato come un bit.

Perché se e = 1, si mette come costo l'infinito?

Supponiamo che per la rete esterna ho 2 ASBR (1 e 2) e ci sia un router interno che vuole raggiungere una destinazione esterna pubblicizzata da entrambi gli ASBR.

Se Metric Type è pari a 2, allora il router all'interno dell'area ignora il costo all'esterno dello AS e si basa solo su come raggiungere lo ASBR più vicino (quindi si basa solo sul costo all'interno dell'area).

Invece se ho Matric Type pari a 1, allora il router interno considera anche il costo all'esterno dello AS perché effettivamente risulta significativo.

#### **Esercizio 6.1 (single-area)**

Finire a casa.

Quanti LSA ho e motivare le risposte.

- Ho 6 LSA di tipo 1, perché ho 6 router.
- Ho 2 LSA di tipo 2, perché ho 2 situazioni in cui due router sono collegati tramite uno switch (quindi canale broadcast).
- LSA di tipo 5?

### Distanza Amministrativa

Regole nel caso una stessa rete di destinazione fosse pubblicizzata da più righe della tabella di routing.

- Una route Statica ha una DA pari a 1, ossia è la più potente.
- OSPF ha una DA pari a 110, potenza media.
- RIP ha una DA pari a , peggiore di OSPF.

### Costo di un Link

Cisco IoS calcola il costo di un link OSPF con la formula:

$$cost = \frac{\text{reference-bandwidth}}{\text{interface-bandwidth}}$$

#### Costo di un Link - Reference-Bandwidth

Esprime la potenza di un link di trasmissione.

Di default è pari a 100 Mbps.

Comunque può essere cambiato con un comando.

#### Costo di un Link - Interface-Bandwidth

Esprime la massima capacità di trasmissione di una interfaccia.

Dipende dall'interfaccia di riferimento.

- Gigabit-Ethernet → 1
- Ethernet → 10
- Serial (T1) → 64

Comunque può essere cambiato con un comando.

Con il comando `show interfaces se0/0/0` vedo varie informazioni, tra cui:

- BW 1544 Kbit → Interface-Bandwidth.

Per cambiare questo valore:

Di default la BW di una seriale vale 1544 Kbit e costo 64.

Per cambiarlo (non cambio il rate di trasmissione reale, cambio solo il valore nominale, ossia solo il costo usato da OSPF) digito:

```
R2(conf)#int se0/0/0
```

Per cambiare il costo di una interfaccia posso fare 2 cose:

- Imposto manualmente il costo a prescindere dalla banda: R2 (config-if) #ip ospf cost 50
  - *Ossia cambio direttamente il costo usato su questa interfaccia da OSPF indipendentemente dalla banda nominale.*
- Cambio direttamente la banda: R2 (config-if) #bandwidth 2000

Quando cambio il costo di un'interfaccia la rete reagisce ed entra in un transitorio.

Il router su cui ho eseguito il comando invierà una nuova versione dello LSA di tipo 1, quindi la variazione non è istantanea in tutta la rete.

Cambiare il costo può provocare ripercussioni su tutta la rete.

Tramite questa cosa posso creare dei percorsi preferenziali e manipolare la rete.

Questa cosa si chiama “Ingegneria del Traffico”, su grandi reti esso diventa un problema di Ricerca Operativa.

## Esercizio 7.2

Vediamo come sono configurate le interfacce di R1:

```
R1#show interface
```

Gli indirizzi IP delle interfacce di R1 sono settati, facciamo la stessa cosa per R2 e R3.

R1

- Fa0/0 172.16.1.17 255.255.255.240
- S0/0/0 192.168.10.1 255.255.255.252
- S0/0/1 192.168.10.5 255.255.255.252
- Lo0 10.1.1.1 255.255.255.255

R2

- Fa0/0 10.10.10.1 255.255.255.0
- S0/0/0 192.168.10.2 255.255.255.252
- S0/0/1 192.168.10.9 255.255.255.252
- Lo0 10.2.2.2 255.255.255.255

R3

- Fa0/0 172.16.1.33 255.255.255.248
- S0/0/0 192.168.10.6 255.255.255.252
- S0/0/1 192.168.10.10 255.255.255.252
- Lo0 10.3.3.3 255.255.255.255

Controllo se OSPF è stato configurato, lo vedo anche perché nella linea di comando di qualsiasi dei 3 router vedo i messaggi di adiacenza.

Vediamo come OSPF è stato settato nelle interfacce di R1.

```
R1#show ip ospf interface
```

```
FastEthernet0/0 is up, line protocol is up
  Internet address is 172.16.1.17/28, Area 0
```

```

Process ID 1, Router ID 10.1.1.1, Network Type BROADCAST, Cost: 1
Transmit Delay is 1 sec, State DR, Priority 1
Designated Router (ID) 10.1.1.1, Interface address 172.16.1.17
No backup designated router on this network
...
Neighbor Count is 0, Adjacent neighbor count is 0
Suppress hello for 0 neighbor(s)

Serial0/0/0 is up, line protocol is up
Internet address is 192.168.10.1/30, Area 0
Process ID 1, Router ID 10.1.1.1, Network Type POINT-TO-POINT, Cost: 64
Transmit Delay is 1 sec, State POINT-TO-POINT, Priority 0
No designated router on this network
No backup designated router on this network
...
Adjacent with neighbor 10.2.2.2
Suppress hello for 0 neighbor(s)

Serial0/0/1 is up, line protocol is up
Internet address is 192.168.10.5/30, Area 0
Process ID 1, Router ID 10.1.1.1, Network Type POINT-TO-POINT, Cost: 64
Transmit Delay is 1 sec, State POINT-TO-POINT, Priority 0
No designated router on this network
No backup designated router on this network
...
Adjacent with neighbor 10.3.3.3
Suppress hello for 0 neighbor(s)

```

Dall'output di questo comando scopriamo varie cose molto utili sulla rete:

- Indirizzo IP di ogni interfaccia di R1 su cui lavora OSPF.
- Aree su cui si trovano le interfacce di R1.
- PID del processo relativo a OSPF che gira su R1.
- Router ID di R1.
- Tipologie di Link a cui è attaccato R1.
  - Nei link broadcast vedo chi è il DR e il BDR.
  - Vedo anche la priorità di R1 su una specifica interfaccia (ossia quanto vuole essere DR).
- Costo dei link a cui è attaccato R1.
- Vedo se R1 ha dei vicini e se li ha, vedo i loro Router ID e a quale interfaccia posso raggiungerli

OSPF è settato su R1, facciamo la stessa cosa per R2 e R3.

Ora entriamo sulle interfacce s0/0/0 e s0/0/1 di R1 e modifichiamo il costo dei link da 64 a rispettivamente 1562 e 390 con il comando ip ospf cost.

```
R1#conf t
```

```
R1(config)#int se0/0/0
R1(config-if)#ip ospf cost 1562
```

```

R1(config-if)#exit

R1(config)#int se0/0/1
R1(config-if)#ip ospf cost 390
R1(config-if)#exit

R1(config)#exit
R1#copy r s

```

Ora entriamo sulle interfacce `s0/0/0` e `s0/0/1` di R2 e R3 e modifichiamo il costo dei link modificando la bandwidth.

La bandwidth finale deve essere quella specificata nello schema di rete sui link seriali.

Nota che il costo si ottiene dalla formula  $cost = \frac{reference-bandwidth}{interface-bandwidth}$

Dove la reference-bandwidth ha un valore di default pari a 100Mbps.

```

R2(config)#int se0/0/0
R2(config-if)#band 64

R2(config)#int se0/0/1
R2(config-if)#band 128

```

```

R3(config)#int se0/0/0
R3(config-if)#band 256

```

```

R3(config)#int se0/0/1
R3(config-if)#band 128

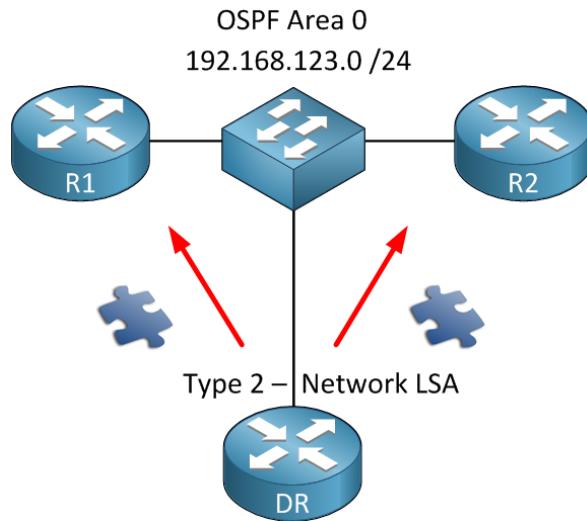
```

### Conclusione

Se adesso digitiamo il comando `show ip ospf interface` su R2 o R3 vedremo che il costo nelle interfacce seriali è cambiato notevolmente.

### Reti Broadcast (Multiaccess Networks)

In una situazione del genere, per questioni di efficienza, OSPF elegge un Designated Router (e un Backup Designated Router), il quale ha il compito di assicurare la sincronizzazione dei database dei router attaccati a questa rete broadcast.



Ciascun altro router (che non sia il DR o BDR) gestisce l'adiacenza con il DR e il BDR.  
Ossia scambia LSA e sincronizza Database solo con loro.

Il DR e BDR fanno questa cosa con tutti gli altri (DROTHER).

Due altri router tra loro non hanno una Adiacenza Full, ossia possono comunicare ma non si scambiano LSA.

#### **Come si elegge il DR?**

Ogni router ha (sulla interfaccia attaccata alla rete broadcast) una priorità (ossia un naturale a 8 bit).

Quello con priorità maggiore diventa un DR, il secondo diventa BDR.

Se la priorità coincide, si usa il Router ID maggiore.

La regola non è sempre rispettata.

La logica è “*io voglio che il processo di elezione sia il più veloce possibile, non conta molto rispettare tutte le regole ma serve che sia veloce*”.

Io voglio che la situazione sia più stabile possibile (ossia voglio che il DR e il BDR cambino il meno possibile, quindi limitare il numero di transitori)

Supponiamo che l’elezione è già avvenuta, se si aggiunge un altro router alla rete broadcast, non viene indetta una nuova elezione, ma il router nuovo si adeguà alla situazione già riconosciuta.

Questo perché non è molto importante che il DR sia sempre quello con Priorità o ID più alto, ma l’importante è che ci sia e che sia operativo.

Se il DR cade: automaticamente il BDR diventa DR e si esegue una elezione per il nuovo BDR.

Se il precedente DR torna sù, esso risulta come un nuovo router, quindi si adeguà come se non ci fosse mai stato.

Quindi, se voglio che sia uno specifico router ad essere il DR, devo fare in modo che nell'ordine di accensione dei Router sia il DR voluto a diventare effettivamente il DR.

#### Adiacenza con DR, BDR e altri DROTHER

Il DR e il BDR ha una adiacenza full (sincronizzati) con tutti i DROTHER.

I DROTHER hanno una adiacenza full con il DR e il BDR, invece con gli altri DROTHER non ha una adiacenza dello stesso genere.

Un nuovo arrivato scopre il DR e BDR dagli Hello ricevuti dai router della rete (Sia DR, BDR e DROTHER).

Vede chi sono il DR e BDR ed esegue l'adiacenza full (FULL/DR) e (FULL/BDR) solo con loro.

Con gli altri DROTHER esegue una adiacenza "non full" (2WAY/DROTHER), ossia sa che sono vicini ma non hanno eseguito la sincronizzazione dei database.

Ossia ci siamo fermati alla fase di Two-Way State, ossia ci fermiamo prima della sincronizzazione dei Database

#### Esercizio 7.4

Notare che i router ID sono messi tramite le interfacce di Loopback.

Chi è il DR e BDR? per scoprirlo uso il comando:

```
RouterA#show ip ospf interface
FastEthernet0/0 is up, line protocol is up
Internet address is 192.168.1.1/24, Area 0
Process ID 1, Router ID 192.168.31.11, Network Type BROADCAST,
Cost: 1
Transmit Delay is 1 sec, State DROTHER, Priority 1
Designated Router (ID) 192.168.31.33, Interface address 192.168.1.3
Backup Designated Router (ID) 192.168.31.22, Interface address
192.168.1.2
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit
5
...
Neighbor Count is 2, Adjacent neighbor count is 2
Adjacent with neighbor 192.168.31.33 (Designated Router)
Adjacent with neighbor 192.168.31.22 (Backup Designated
Router)
Suppress hello for 0 neighbor(s)
```

Oppure usa il comando:

```
RouterA#show ip ospf neighbor
```

Neighbor ID	Pri	State	Dead Time	Address	Interface
<b>192.168.31.22</b>	<b>1</b>	<b>FULL/BDR</b>	00:00:34	192.168.1.2	FastEthernet0/0
<b>192.168.31.33</b>	<b>2</b>	<b>FULL/DR</b>	00:00:34	192.168.1.3	FastEthernet0/0

Quindi RouterC è il DR e RouterB è il BDR.

Disattivo l'interfaccia di RouterC (`R (conf-if) shutdown`), cosa succede?

Avviene una nuova elezione per il DR.

Il BDR diventa DR e poi RouterA viene eletto BDR.

Riattivo l'interfaccia del RouterC (no shutdown), cosa succede?

Vedo che vengono scambiati i router LSA e il RouterB (il DR) invia un Network LSA.

Per cambiare la priorità:

```
RouterA(config)#
RouterA(config)#int fa0/0
RouterA(config-if)#ip ospf priority 200
```

Cosa vedo?

I tre router all'inizio si scambiano gli Hello.

Negli Hello è segnata la priorità.

Quando hanno fatto l'adiacenza con tutti, tutti e tre eleggono il designated router, ossia quello con priorità maggiore.

Quindi dopo il primo scambio di hello, il DR e BDR sono stati eletti entrambi.

## OSPF Multiarea

Il motivo delle aree è la scalabilità limitata di OSPF.

Ad un'area non appartiene il router, ma le sue interfacce.

- Se le sue interfacce sono tutte nello stesso router, allora è un Router Interno.
- Se ha almeno 2 interfacce in 2 aree diverse, allora è un ABR.
- Se ha almeno una interfaccia non connessa a nessuna area, allora è un ASBR.
  - *Di solito gli ASBR si mettono nell'area di Backbone, per diminuire al minimo il numero di aree attraversate.*

In un caso multiarea i database dei router sono sincronizzati all'interno della singola area.

Quindi quando ho più aree, OSPF mantiene un Database separato per ogni area.

OSPF sa in quale database mettere uno specifico LSA perché la destinazione è scritta nel header dello LSA stesso, nello specifico nel campo `Area ID`.

Un router interno ad un'area si comporta allo stesso modo che abbiamo definito finora, il suo database interno descrive solo ciò che è presente nella sua area.

Invece un Router ABR riceve LSA da router in più aree diverse, in tal caso, il router ABR mantiene altrettanti database separati, uno per ogni area a cui è collegato.

Un ABR genera 2 LSA di tipo 1, uno per ogni area a cui è collegato.

Quindi un router ABR, in ogni area a cui è collegato, invia un LSA di tipo 1 dove annuncia i link a cui è connesso in quell'area, ignorando le altre aree.

### **LSA di tipo 3 (Summary Link)**

I router di un'altra area, per conoscere le destinazioni in altre aree, devono ricevere un LSA di tipo 3 dal ABR.

Ossia, il ABR manda LSA di tipo 3 in un area per pubblicizzare le destinazioni di un'altra area, questa cosa lo fa anche le le destinazioni che apprende tramite l'area di backbone.

Gli LSA di tipo 3 sono uno per ciascuna destinazione, non come gli LSA di tipo 1 che accumulavano più destinazioni in uno solo.

Un router interno può scegliere eventualmente uno tra più ABR per raggiungere una destinazione in un'altra area.

Un LSA di tipo 3 dice:

*"Tramite me, un router ABR, puoi raggiungere questa rete con un costo DA ME IN POI pari a 25".*

Questa info la riceve anche da altri ABR, un router interno quindi prende la strada migliore.

La metrica è sempre valida perché comunque siamo nella stessa rete.

Gli LSA di tipo 3 sono tanti, quante le possibili destinazioni Inter-Area.

### **LSA di tipo 4 (ASBR summary)**

Un LSA di tipo 4 a cosa serve?

Viene generato e inviato da un ABR e serve per pubblicizzare un ASBR (ossia un router connesso all'esterno, "che non vede i confini di area").

Un router C (un ABR) sa quali router sono ASBR o meno.

Un LSA di tipo 4 dice a tutto lo AS:

*"Tramite me, un router ABR, puoi raggiungere questo ASBR con un costo DA ME IN POI pari a 25".*

## DISEGNO RETE CON AREE

Perché R<sub>c</sub> invia LSA di tipo 4 per pubblicizzare un ASBR anche nell'area stessa in cui si trova l'ASBR?

Perché R<sub>c</sub> riceve un LSA di tipo 4 da R<sub>b</sub> dove pubblica l'ASBR, quindi R<sub>c</sub> pubblica l'informazione ricevuta da R<sub>b</sub>.

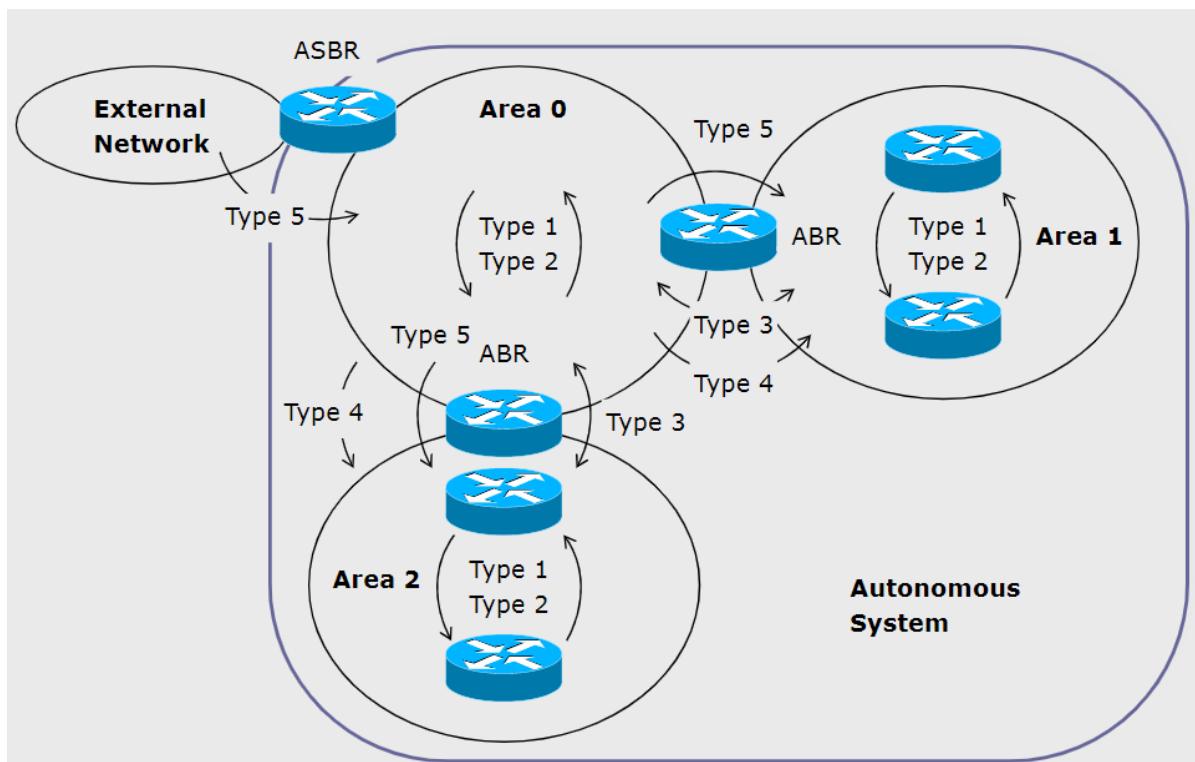
Questa informazione non è ridondante o inutile, anzi è molto importante, perché mi dà un percorso alternativo per raggiungere l'ASBR.

## Riassunto degli LSA

- **Tipo 1**

- Il loro scopo è quello di:
  - Conoscere i Neighbor.
  - Pubblicizzare le Reti a cui si è connessi e che si può raggiungere.

- Eleggere il DR/BDR nel caso in cui si fosse in un canale broadcast.
- Vengono generati dai router di un'area e viaggiano solo all'interno dell'area in cui sono generati.
- **Tipo 2**
  - Il loro scopo è quello di:
    - Far conoscere i Neighbor connessi al canale broadcast.
    - Pubblicizzare le Reti a cui si è connessi e che si può raggiungere.
  - Vengono generati dai DR di un canale broadcast e viaggiano solo all'interno del canale broadcast.
- **Tipo 3**
  - Il loro scopo è quello di:
    - Pubblicizzare una destinazione situata in un'altra area.
    - *"Tramite me, un router ABR, puoi raggiungere questa rete con un costo DA ME IN POI pari a 25".*
  - Vengono generati solo dagli ABR e vengono trasmessi dagli ABR
- **Tipo 4**
  - Il loro scopo è quello di:
    - Pubblicizzare ASBR situato in un'altra area.
    - *"Tramite me, un router ABR, puoi raggiungere questo ASBR con un costo DA ME IN POI pari a 25".*
  - Vengono generati solo dagli ABR e vengono trasmessi dagli ABR.
- **Tipo 5**
  - Viene generato da un ASBR e circola all'interno della sua area.
  - Il suo scopo è quello di far conoscere a tutti i router della sua area che tramite lui si può uscire dallo AS.
  - verrà pubblicizzato nelle altre aree tramite gli ABR e LSA di tipo 4.



## Esercizio 7.5 - OSPF Multiarea

Abbiamo 3 aree, quella di backbone e altre 2.

Ogni area ha 2 LAN (ossia 2 switch), ognuna con il suo indirizzo di sottorete.

Ognuna di esse non contiene un router interno (questo esempio è un caso particolare).

Le aree direttamente collegate ad un ABR e che non hanno router interni sono dette “*Stub Areas*”.

I router ID sono già configurati.

Appena si parte, la rete è configurata in modo che ci sia un'unica area, quindi l'esercizio consiste nel cambiare area a certe interfacce di R2 e R3.

Partiamo da R2.

Uso il comando “`show running config`”

Vedo che in R2 non è stata configurata l'interfaccia direttamente connessa ad una delle LAN (*la LAN di destra*).

Quindi digito il comando `network` per far pubblicizzare la LAN da OSPF.

La stessa cosa va fatta in R3, anche lì manca una LAN.

Notare che quando introduciamo molteplici aree le tabelle di routing dei router non cambieranno (nel piano dati, a meno di vincoli, le aree sono irrilevanti).

Ciascun router deve essere in grado di calcolare il percorso di costo minimo a prescindere dal fatto che ci sia una sola area o molteplici.

Quando mediante il comando `network` specifico che un'interfaccia è in un'area diversa, automaticamente, il router capirà di essere un ABR e si comporterà come tale.

Quando cambio di area ad un'interfaccia avviene un transitorio, perché avviene un cambio di topologia.

Non ci interessa studiare questo transitorio.

Per cambiare di area ad una interfaccia basta digitare il comando `network` indicando una area diversa

Dopo aver cambiato le aree, eseguo un “*Power Cycles Devices*” per spegnere e riaccendere tutto.

### Parti Importanti - Esercizio 7.5

Nell'area 1 vedo che R2 pubblicizza anche le aree raggiungibili tramite R3 tramite un LSA di tipo 3.

Le reti dentro l'area 0, R2 le impara tramite gli LSA di tipo 1.

### Definizione di ABR

La condizione necessaria per cui in OSPF un Router è considerato ABR se:

- Ha almeno due interfacce in due aree diverse.
  - *Perché un ABR connette più aree.*
- Ha almeno una interfaccia connessa all'area di backbone.
  - *In modo da evitare percorsi chiusi tra le aree.*
  - *Una rete multi area deve sembrare una stella dove in mezzo ho l'area di Backbone, in modo che la rete sia schematizzabile come un albero di profondità 1, quindi senza percorsi chiusi.*

### **Inter-area Route Summarization**

OSPF non esegue nessuna auto-summary sugli ABR.

Se la vogliamo, dobbiamo farla manualmente tramite uno specifico comando:

```
R2(config)#router ospf 1
R2(config-router)#area 1 range 10.10.0.0 255.255.254.0
```

Ossia specifico il range di indirizzi che un'area contiene.

Tramite questo comando posso permettere ad un ABR di fare un summary e quindi inviare meno LSA di tipo 3 ma che comprendono più destinazioni.

L'ABR annuncerà solo l'indirizzo che ho specificato nel comando Area.

Un altro vantaggio è lo snellimento delle tabelle di routing.

Cosa succede se nelle sottoreti ho costi diversi per raggiungerle? Che costo avrà il summary? Nel summary associo il costo più piccolo tra tutte le sottoreti.

### **Stub Areas**

Aree direttamente connesse agli ABR.

- I pacchetti nascono in quella rete oppure sono destinati in essa, ossia non è un area di transito.
- Non contengono degli ASBR.

Gli ABR possono essere configurati per indicare le Stub Areas.

Nelle stub area non devo inviare LSA di tipo 4 o 5.

Per la default route mette se stesso come destinatario in un LSA di tipo 3.

### **NSSA (Not So Stubby Areas) Areas**

Stub Networks ma con un ASBR all'interno.

Non le vediamo.

**Fine di OSPF, meno male eh?**

**“OSPF è proprio un bel protocollo”**

### **Hub**

Gli HUB sono dei semplici ripetitori di segnale.

- Funzionano a livello 1.
  - Quindi vedono il segnale per quello che è, ossia una sequenza di generici impulsi elettrici.
- Sono componenti attivi e rigenerano il Segnale.
- Non prevede una gestione delle collisioni.
  - La loro gestione è delegata allo standard Ethernet e quindi agli Host della LAN.

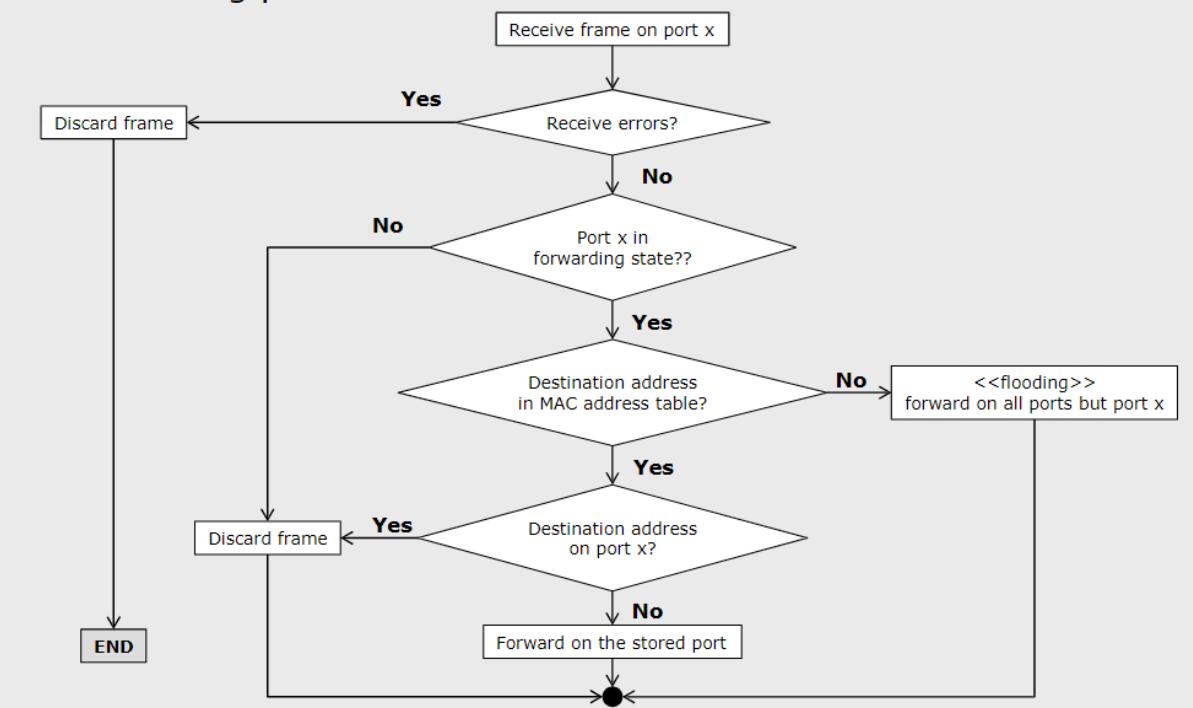
Dal punto di vista della rete, un Hub inoltra il segnale su tutte le sue porte (a parte quella sorgente) a prescindere da chi cosa ci sia al di là di esse.

Il dominio di collisione coincide con il dominio di broadcast.

## Switch

Lo switch ha il compito di ricevere dei frames (pacchetti di livello 2) e inoltrarli verso la porta destinataria dove si presume ci sia il ricevente del frame.

### ■ Forwarding process



Per fare questa operazione di forwarding lo switch (come il router) usa una tabella che in questo contesto è chiamata “Tabella di Switching” o anche “Tabella di Forwarding”.

Le righe di questa tabella mostrano delle associazioni del tipo:

< Indirizzo MAC , Porta >

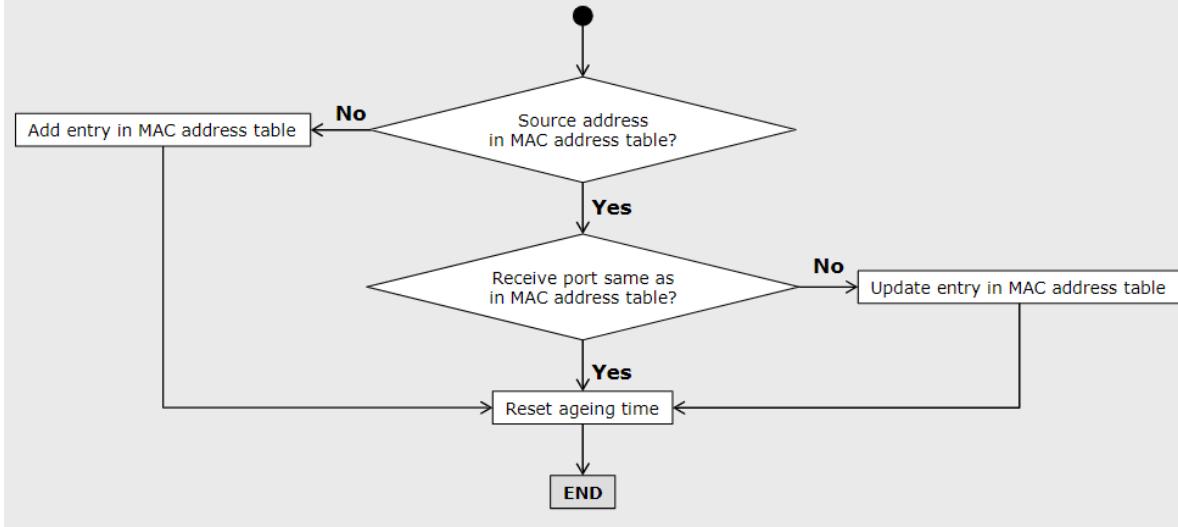
Come si riempie questa tabella?

A differenza dei router, negli switch l'intervento umano è ridotto al minimo, infatti la tabella di switching viene riempita automaticamente dallo switch stesso attraverso un semplice meccanismo di Self-Learning.

Si può dire che lo switch è un dispositivo “*Plug and Play*”.

Lo switch impara le destinazioni dai messaggi che arrivano leggendo il campo contenente l’indirizzo MAC sorgente, a quel punto inserisce nella tabella una riga con la corrispondenza.

### ■ Learning process



In una rete switched (a meno di percorsi chiusi) c’è sempre un unico percorso per raggiungere una specifica destinazione.

### Switch vs Hub - Collisioni

Come visto in Reti Informatiche, uno dei problemi delle LAN erano le collisioni tra trasmissioni, in passato furono in parte risolte tramite dei protocolli di gestione e recupero delle collisioni (tipo CSMA).

Lo switch fa in modo che il dominio di collisione non sia più coincidente al dominio di broadcast ma che sia ristretto alla singola porta.

Quindi è sempre una comunicazione punto-punto.

Quindi se avviene una collisione essa è limitata allo switch e al singolo Host.

Gli altri Host della LAN non vengono influenzati dalla collisione.

Il dominio di collisione non coincide con il dominio di broadcast.

Il dominio di collisione è confinato alla singola porta.

Con uno switch non c’è mai una collisione tra due trasmissioni.

Lo switch a differenza del routing, deve riempire la tabella in automatico aspettando che arrivino i pacchetti.

### Esercizio 7.7

Suddividere il blocco 172.20.0.0/16

Subnet	N° Host Richiesti	N° Indirizzi Necessari	Address	Mask	Dec. Mask	Assignable range	Broadcast
HQ	8000 + 1	8192	172.20.0.0	/19	255.255.224.0	172.20.0.1 172.20.31.254	172.20.31.255
Branch 1	4000 + 1	4096	172.20.32.0	/20	255.255.240.0	172.20.32.1 172.20.47.254	172.20.47.255
Branch 2	2000 + 1	2048	172.20.48.0	/21	255.255.248.0	172.20.48.1 172.20.55.254	172.20.55.255
HQ - B1	2	4	172.20.56.0	/30	255.255.255.252	172.20.56.1 172.20.56.2	172.20.56.3
HQ - B2	2	4	172.20.56.4	/30	255.255.255.252	172.20.56.5 172.20.56.6	172.20.56.7
B1 - B2	2	4	172.20.56.8	/30	255.255.255.252	172.20.56.9 172.20.56.10	172.20.56.11

**20 / 04 / 2023**

#### **Switch - VLAN “Awareness”**

Quando uno switch inoltra un frame, per definizione ha deciso in qualche modo a quale VLAN appartiene quel frame.

Per questo uno switch ha una tabella di switching per ogni VLAN.

#### **Switch - Configurazione iniziale delle Porte**

Quando collego due porte avvengono delle negoziazioni (full duplex, velocità etc) a livello fisico previste dallo standard Ethernet.

Questa cosa è fatta in modo automatico, anche se è controllabile tramite CLI.

#### **Switch - SSH**

Su uno switch (e anche su un router) si può eseguire le configurazione tramite SSH (terminale remoto).

Però SSH ha bisogno di un indirizzo IP ma lo switch è un dispositivo a livello 2!

Uno switch (di base) non ha un indirizzo IP...non è sempre vero.

#### **Switch - Switch Virtual Interfaces - SWI**

Posso assegnare un indirizzo IP ad una porta di uno switch.

Ma questa cosa se ci si pensa non ha molto senso, perché uno switch non può essere il destinatario finale di un pacchetto.

Per gestire questa cosa si fa un “*trucco istituzionalizzato*”.

Si immagina che lo switch al suo interno abbia delle porte virtuali (stesso concetto delle interfacce di Loopback) dette “Switch Virtual Interfaces” SWI con un indirizzo MAC, indirizzo IP e subnet mask.

Queste interfacce virtuali sono Interfacce di Livello 3 (ossia operano a livello 3, quindi hanno un indirizzo IP).

**Quante SWI ha uno switch?**

Queste interfacce virtuali sono una per ogni VLAN configurate su quello switch.

Di default, uno switch ha una sola VLAN (la VLAN 1) e quindi se non vogliamo averne di più, l’interfaccia virtuale la associamo alla VLAN 1.

```
S1#conf t  
S1(config)#interface vlan 1  
S1(config-if)#ip address 192.168.1.2 255.255.255.0  
S1(config-if)#no shutdown
```

La porta di livello 3 è detta “*Terminale*”, ossia i pacchetti che entrano in questa porta sono destinati propriamente allo switch.

Invece le altre porte sono di livello 2 dette “*Switched*”, i frame ricevuti su queste porte verranno mandati al fabric e subiranno lo switching come da norma.

### **Switch - VLAN**

Prendiamo una LAN switched priva di VLAN (quindi ce n’è una sola che include tutte le porte dello switch).

Lo switch divide il dominio di collisione, però non divide il dominio di broadcast.

Quando uno switch riceve un frame con indirizzo broadcast a quel punto lo switch lo inoltra su tutte le porte.

Quello che vorrei è dividere anche il dominio di broadcast, confinandolo a sottoinsiemi di host.

Vantaggi:

- **Economico.**
  - Permette di separare il traffico senza acquistare molteplici switch.
- **Sicurezza.**
  - Posso evitare che certi host ricevano certi frame broadcast.
- **Efficienza.**
  - Evita il “Broadcast Storm”.

Tramite le VLAN posso dividere lo switch in più switch separati.

In genere tutti gli switch funzionano con le VLAN (“*a parte gli switchini da 10 euro usati solo per collegare pochissimi dispositivi in un tavolo*”).

### **VLAN - Problema delle VLAN in un contesto inter-switch - Trunking**

Ci sono dei Problemi nelle VLAN? Purtroppo sì!

Nel caso in cui una VLAN sia distribuita su più switch fisici.

In questo caso il frame deve viaggiare su molteplici switch e come fanno gli switch a capire a quale VLAN appartiene un certo frame?

Per permettere questa cosa serve un'informazione per l'inoltro che va oltre al sapere semplicemente quale è la porta di uscita di una certa destinazione, mi serve sapere la VLAN a cui appartiene quel frame.

Nota che di norma gli switch per fare ciò che fanno guardano solo 2 cose (entrambe contenute nella tabella di switching):

- Indirizzo destinazione
- Porta di uscita

Per fare questa cosa ho 2 possibili soluzioni, lo standard ha deciso di adottarne solo una.

Cominciamo dalla soluzione non usata.

### Frame Filtering

Ogni switch ha un database dove associa <Indirizzo MAC sorgente , VLAN ID>.

In questo modo so a quale VLAN appartiene ogni host della LAN.

Questo database deve essere sincronizzato su tutti gli switch, per fare questa cosa potrei definire un protocollo oppure un database centralizzato su un server.

#### Pro

- In maniera naturale supporta la mobilità degli Host, se cambio porta di accesso il mio MAC resta lo stesso e quindi a prescindere su quale switch sono, la mia associazione con la VLAN non ne è intaccata perché è segnata sul database.
- Non devo cambiare il protocollo, nello specifico, non devo cambiare il formato del pacchetto di livello 2.

#### Contro

- Meno Scalabile, se ho grande mobilità di host allora devono essere fatte altrettante query sul database che ricordiamo deve restare sincronizzato tra tutti gli switch.

### Frame Tagging

Il VLAN ID è scritto nel frame di livello 2.

#### Pro

- Molto più scalabile, perché ho meno roba da gestire dentro gli switch.

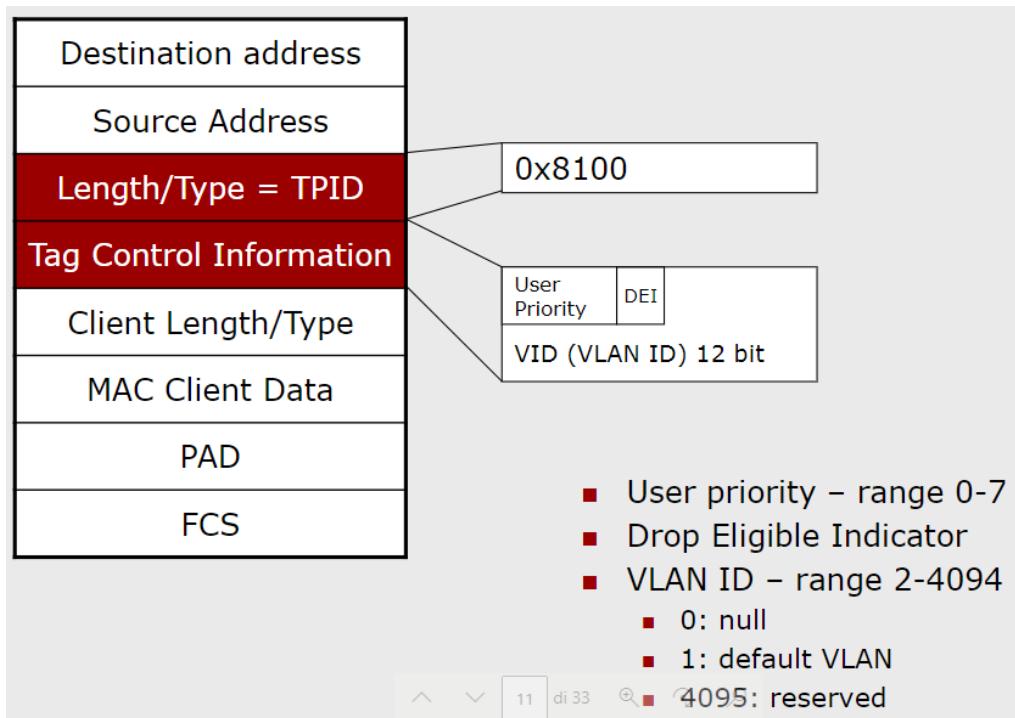
#### Contro

- L'associazione con le VLAN è fatta mediante le porte dello switch, quindi se cambio porta potrei cambiare VLAN.
- Devo cambiare il protocollo, nello specifico cambiare il formato del frame.

### VLAN - Frame Tagging - Pacchetti Tagged

Aggiungo 2 campi che in tutto sono al più 32 bit.

- **Length / Type | (TPID - Tag Protocol Identifier)** [ 16 ] Bit
  - Si sfrutta questo campo per stabilire la lunghezza del frame, quando quest'ultimo è inferiore a 0x8100 (1500).
  - Se TPID > 0x8100 , questo campo esprime il tipo di frame.
  - Il valore 0x8100 corrisponde alla lunghezza massima del frame.
- **Tag Control Information (TCI)** [ ≤ 16 ] Bit
  - **User Priority** [0 , 7] Bit
    - Non riguarda propriamente le VLAN.
    - Da la possibilità di trattare i frame con una priorità
  - **Drop Eligible Indicator (DEI)** [ 1 ] Bit
    - Campo di 1 bit che indica la possibilità di ignorare il frame in caso di congestione.
  - **VLAN ID (VID)** [12] Bit
    - Quindi posso avere al più  $2^{12} = 4096$  VLAN in una sola LAN, in realtà quelle effettivamente utilizzabili sono di meno (4094) perché la prima e l'ultima sono riservate.
    - Questo ai nostri giorni è un limite, soprattutto in un Datacenter.
    - Il limite di 4096 VLAN è un limite fisico, in quanto logicamente funziona.



### Standard che regola le VLAN

Se ne vuoi sapere di più, leggi lo standard!

Le VLAN sono definite dallo standard IEEE 802.1Q.

### Dispositivi “VLAN-aware” & “VLAN-unaware”

Lo standard contempla l’eterogeneità dei dispositivi: nella stessa rete possono esserci dispositivi che conoscono questo standard detti “VLAN-aware” e dispositivi che invece non lo conoscono “VLAN-unaware”.

I primi sanno gestire i TAG che si trovano in un frame tagged, i secondi invece non possono leggerli perché il formato risulta a loro sconosciuto.

Gli host finali in genere sono *VLAN-unaware*, perché non è di loro competenza sapere a quale VLAN appartengono.

Ma non è una regola ferrea, anzi è molto spesso violata, posso avere anche Host *VLAN-aware*.

Un esempio di Host VLAN-aware è il Router o alcuni Server.

I router sono semplici host dal punto di vista dello switch ma comunque sono in grado di leggere i tag.

## VLAN - Porte e Link

### 1. Access Port

- I frame che viaggiano su questa porta (*per configurazione*) sono sicuro che appartengono ad una specifica VLAN.
  - La VLAN a cui appartiene la porta dello switch.
- Non serve specificare il Tag nel frame.

### 2. Trunk Port

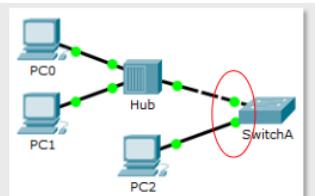
- I frame che viaggiano su una porta trunk devono sempre avere il Tag.
- La VLAN a cui appartengono i frame su una porta trunk è scritta sul Frame.

### 3. Hybrid Port

- I frame che viaggiano su una porta ibrida possono avere o non avere il tag.
- I frame senza tag sono inoltrati verso una “VLAN Nativi” che tutti gli switch della LAN conoscono.

#### ■ Access port – Access link

- Tx/Rx untagged frames



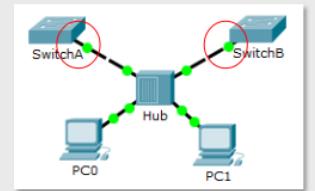
#### ■ Trunk port – trunk link

- Tx/Rx tagged frames



#### ■ Hybrid port – hybrid link

- Tx/Rx both tagged and untagged frames
- Untagged frames are forwarded to a configured link *native VLAN*



## VLAN ID Range

In teoria, in una singola LAN posso avere al più 4096 VLAN diverse.

Questo range di 4096 ID è stato diviso in 2 categorie.

Gli switch sono in grado di capire a quale categoria appartengono degli ID e quindi possono gestirli in due modi diversi.

Data questa differenza, l'ID di una VLAN non può essere dato a caso.

- **Normal Range** [ 0001 – 1005 ]
  - Usati in contesti di reti piccole/medie.
  - Gli ID 0001, 1002, 1003, 1004 e 1005 sono riservati.
- **Extended Range** [ 1006 – 4094 ]

Le configurazioni sono memorizzate in un Database, conservato in un file chiamato `vlan.dat`, nella memoria dello switch.

#### **VLAN di Default [ID 0001]**

In uno switch che contempla le VLAN una porta non può non essere assegnata ad una VLAN.

Quando uno switch esce dalla fabbrica o al riavvio (*a meno di configurazione iniziale cambiata*) tutte le sue porte sono incluse nella VLAN 1.

Non può essere rinominata o cancellata.

Se non salvo la configurazione di uno switch, tutte le sue porte tornano nella 0001.

#### **VLAN Nativa**

Serve configurarla in caso di porte trunk.

Default in Cisco: Negli switch Cisco le porte trunk sono ibride con VLAN Nativa 0001.

Ma comunque la posso cambiare.

#### **VLAN di Management**

La VLAN di Management separa gli indirizzi IP degli Switch da tutte le altre VLAN.

Per motivi ovvi legati alla sicurezza e alla separazione, di norma tutti gli switch in quanto apparati raggiungibili come apparati di rete (dato che hanno indirizzi IP dati dalle SWI) ha senso metterli in una VLAN dedicata dove gli Host normali (e quindi possibilmente malevoli) possano raggiungerli senza una opportuna autorizzazione.

La VLAN di management NON DOVREBBE essere (per motivi di sicurezza) la 0001.

Negli esercizi la VLAN di Management sarà la 0099.

#### **VLAN - 1° Laboratorio**

Mostrare le info sulle VLAN su uno Switch:

```
Switch#show vlan brief
```

Mostra per ogni VLAN

- ID e Nome della VLAN.
- Stato (Attiva o Non Attiva)
- Porte di Accesso assegnate su questa VLAN.
  - Ogni frame che esce o entra da queste porte è nella VLAN riportata.

Per creare una VLAN:

```
Switch(config)#vlan 20
Switch(config-vlan)#name studenti
Switch(config-vlan)#end
```

Appena creata la VLAN è aggiunta alla lista, è attiva ma non ha porte incluse.

Per aggiungere le porte alla VLAN.

Per ogni porta specifico:

- Tipologia di Porta (Accesso o Trunk).
  - Di default sono di Accesso e quindi compare nella lista, se non appare nella lista è Trunk.

```
Switch(config)#interface Fa0/1
Switch(config-if)#switchport mode access // Sei una porta Access
Switch(config-if)#switchport access vlan 20 // Sei nella VLAN 20
```

Ora nella lista delle VLAN, la Fa0/1 sarà nella VLAN 20.

Ma farlo per ogni interfaccia è lungo e noioso, posso fare un comando per tutte? Certo!

```
Switch(config)#interface range fa0/2 - 10 // Le Fa dalla 0/2 alla 0/10
Switch(config-if-range)#switchport mode access // Siete delle porta Access
Switch(config-if-range)#switchport access vlan 20 // Siete nella VLAN 20
```

Quindi dividere le porte in VLAN si tratta di togliere le porte dalla VLAN 0001 e spostarle in altre VLAN.

```
Switch#show interfaces switchport
```

Mi mostra info riguardo l'operato di una porta in quanto switchport (porta di livello 2).

Le porte di accesso sono fatte per collegare Host VLAN-unaware.

Se una porta so che è collegata ad un altro switch, essa deve essere di tipo Trunk.

Sull'altro switch devo definire la VLAN 20.

L'ID deve essere lo stesso, il nome può cambiare.

```
Switch_altro(config)#vlan 20
Switch_altro(config-vlan)#name studenti_altro

Switch_altro(config)#interface range fa0/2 - 10
Switch_altro(config-if-range)#switchport mode access
Switch_altro(config-if-range)#switchport access vlan 20
```

Ora devo collegare i due switch, uso un cavo crossover (quello tratteggiato).

Le porte in questione di default sono nelle VLAN 0001 e sono porte di accesso.

Devo configurarlo come trunk.

*"Prendi questa porta, togila dalla VLAN 1 e considerala come trunk, ossia come una porta da dove arrivano frames da qualsiasi VLAN"*

```
Switch(config)#interface Gi0/1
Switch(config-if)#switchport mode trunk // Sei una porta Trunk
Switch(config-if)#switchport trunk native vlan 1
```

L'ultimo comando non è obbligatorio, di default la VLAN nativa è la 0001.

L'importante è che dall'altra parte la VLAN nativa sia la stessa.

```
Switch_altro(config)#interface Gi0/1
Switch_altro(config-if)#switchport mode trunk // Sei una porta
Trunk
Switch_altro(config-if)#switchport trunk native vlan 1
```

Configurare la porta trunk nel secondo switch in realtà non è necessario

Perché gli switch automaticamente si scambiano informazioni di configurazione, il primo switch ha detto al secondo che la porta che li collega è di tipo trunk e quindi anche il secondo si è adattato.

Quando configuro uno switch come trunk posso dargli info aggiuntive.

*"Su questo link trunk devi considerare solo le VLAN 0010, 0020 e 0023"*, ossia posso mettere un filtro nello switch.

Posso escludere una certa vlan.

```
Switch_altro(config)#interface Gi0/1
Switch_altro(config-if)#switchport mode trunk allowed vlan except
20
```

Oppure toglierle tutte e poi mettere solo quelle buone.

```
Switch_altro(config)#interface Gi0/1
Switch_altro(config-if)#switchport mode trunk allowed vlan except
all
Switch_altro(config-if)#switchport mode trunk allowed ???
```

#### Cisco Dynamic Trunking Protocol

Protocollo che esegue le negoziazioni riguardo le porte trunk.

```
sw(config-if)# switchport mode dynamic auto
```

Comportamento di default.

La porta è in grado di essere trunk, ma non richiede alla porta dall'altra parte di essere trunk.

Per questo motivo dopo aver impostato come trunk la porta del primo switch a quel punto anche la porta nel secondo è diventata trunk.

Perché la prima ha chiesto alla seconda di diventare trunk.

```
sw(config-if)# switchport mode dynamic desirable
```

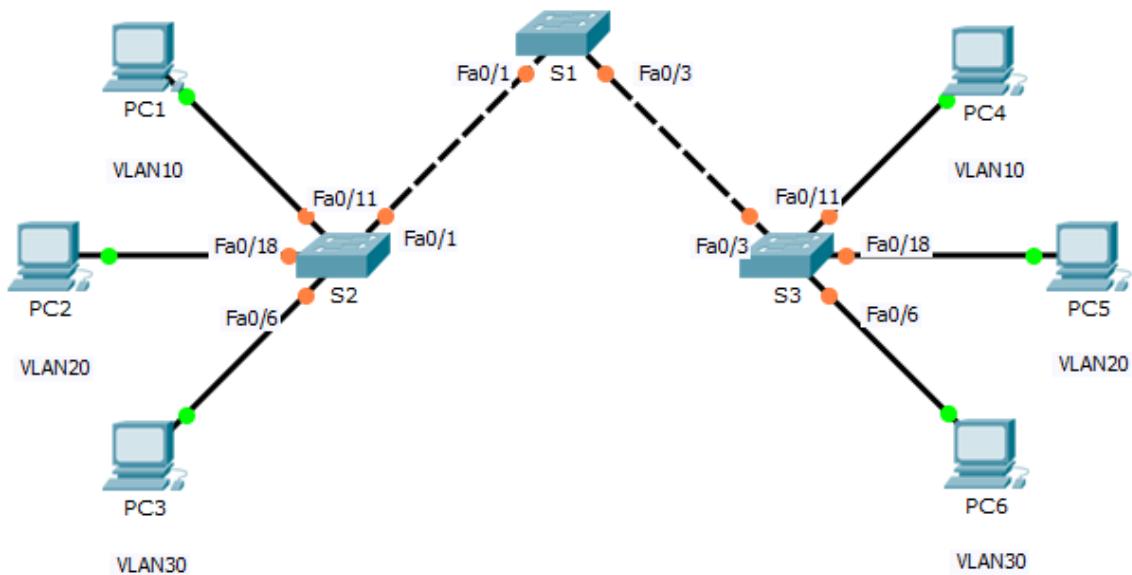
La porta è in grado di essere trunk, e può chiedere alla porta dall'altra parte di essere trunk ("Io voglio essere trunk, puoi esserlo anche tu?").

### VLAN Registration Protocol

Permette agli switch di rilevare automaticamente le VLAN.

Non li guardiamo.

### Esercizio 9.2



Usa il comando show vlan per vedere le vlan di uno switch.

```
S2#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 Gig0/1, Gig0/2
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
1002	fddi	101002	1500	-	-	-	-	-	0	0
1003	tr	101003	1500	-	-	-	-	-	0	0
1004	fdnet	101004	1500	-	-	-	ieee	-	0	0
1005	trnet	101005	1500	-	-	-	ibm	-	0	0

```
Remote SPAN VLANs
```

---

Crea una VLAN sullo switch 1.

```
S1#conf t
```

```
S1(config)#vlan 10
S1(config-vlan)#name Faculty/Staff
```

```
S1(config-vlan)#vlan 20
S1(config-vlan)#name Students
```

```
S1(config-vlan)#vlan 30
S1(config-vlan)#name Guest(Default)
```

```
S1(config-vlan)#vlan 99
S1(config-vlan)#name Management&Native
```

```
S1(config-vlan)#exit
S1(config)#exit
S1#copy r s
```

Queste VLAN le definisco anche sugli altri 2 switch.

Ora ho 4 nuove VLAN, ma queste non hanno ancora nessuna interfaccia assegnata. Quindi usiamo il comando interface e configuriamo le interfacce.

Partiamo dallo switch 2.

```

S2(config)#interface fastEthernet 0/6
S2(config-if)#switchport mode access
S2(config-if)#switchport access vlan 30
S2(config-if)#interface fastEthernet 0/11
S2(config-if)#switchport mode access
S2(config-if)#switchport access vlan 10
S2(config-if)#interface fastEthernet 0/18
S2(config-if)#switchport mode access
S2(config-if)#switchport access vlan 20

```

Se provo a fare ping, i frame non superano gli switch S2 e S3, come mai?  
Perché manca la porta trunk.

Quindi ora vado su S1.

```

S1(config)#interface FastEthernet 0/1
S1(config-if)#switchport mode trunk
S1(config-if)#switchport trunk native vlan 99
S1(config-if)#interface FastEthernet 0/3
S1(config-if)#switchport mode trunk
S1(config-if)#switchport trunk native vlan 99

```

Se tutto è andato a buon fine, nel terminale comparirà:

```

%CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch discovered on
FastEthernet0/1 (99), with S2 FastEthernet0/1 (1).
%CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch discovered on
FastEthernet0/3 (99), with S3 FastEthernet0/3 (1).

```

Dove segnala che nello switch 1 la VLAN nativa è la 99, mentre in S2 e S3 è la VLAN 1.

Ora verifichiamo che su S2 il trunking è abilitato.

```

S2#show interface fastEthernet 0/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
<output omitted>

```

Vedo che lo switch 1 ha negoziato il trunking con S2 e quindi anche la porta di S2 ora è di tipo trunk.

Ma per essere sicuri che all'accensione la situazione sarà quella che vogliamo, lo configuriamo comunque e con la stessa vlan nativa di S1.

Mettiamo la stessa VLAN nativa perché è meglio.

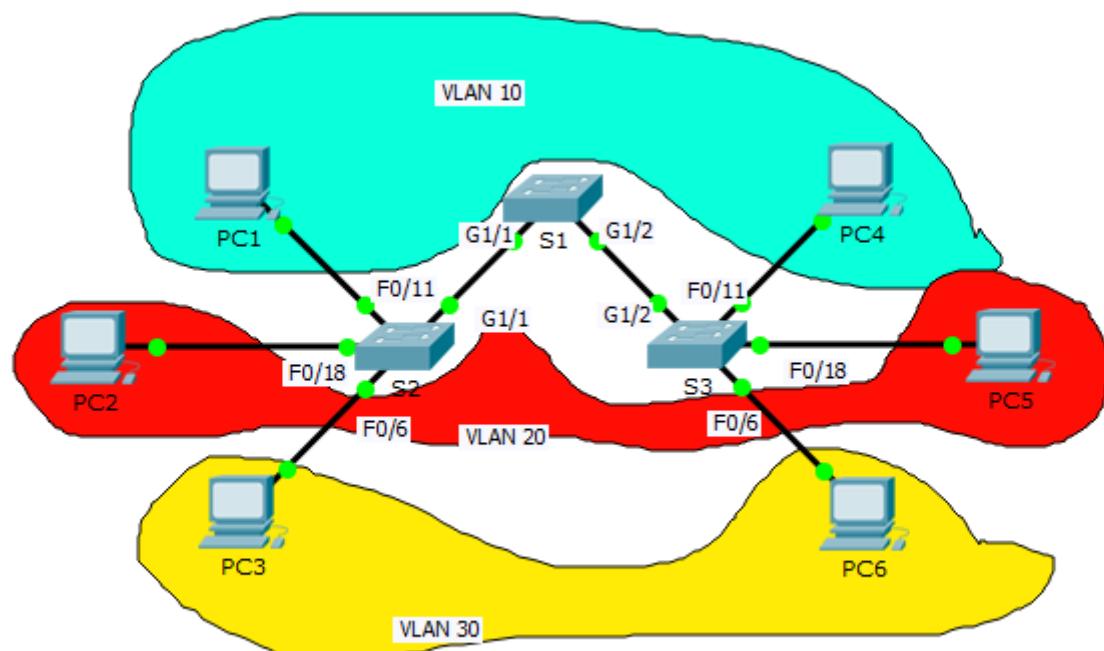
```
S2(config)#interface FastEthernet 0/1
S2(config-if)#switchport mode trunk
S2(config-if)#switchport trunk native vlan 99
```

Stessa cosa su S3.

Vedo che Avviene di nuovo la negoziazione con S1 con la stessa VLAN nativa.

```
%SPAN TREE-2-UNBLOCK_CONSIST_PORT: Unblocking FastEthernet0/3 on
VLAN0099. Port consistency restored.
```

### Esercizio 9.3 - Troubleshooting



Device	Interface	IPv4 Address	Subnet Mask	Switch Port	VLAN
PC1	NIC	172.17.10.21	255.255.255.0	S2 F0/11	10
PC2	NIC	172.17.20.22	255.255.255.0	S2 F0/18	20
PC3	NIC	172.17.30.23	255.255.255.0	S2 F0/6	30
PC4	NIC	172.17.10.24	255.255.255.0	S3 F0/11	10
PC5	NIC	172.17.20.25	255.255.255.0	S3 F0/18	20
PC6	NIC	172.17.30.26	255.255.255.0	S3 F0/6	30

1. Testa la connettività tra i PC dentro le VLAN.
2. Investiga sui problemi di connettività.
3. Risolvili.

Per prima cosa, controlliamo le cose banali, per escluderle subito.

### Controlla gli Indirizzi IP dei PC

*L'indirizzo IP di PC6 è errato.*

### Controlla che le Porte siano effettivamente quelle giuste

*Nota che PC5 è collegato alla Fa0/17 e non alla Fa0/18.*

Una volta che le cose “banali” sono state sistamate passiamo a fare i primi test.

### Prova il Test di Connattività

Provando il test di connattività (mostrando solo i pacchetti ARP e ICMP) vedo che i pacchetti ARP non arrivano a destinazione.

In particolare, fermiamoci sul ping da PC1 a PC4 (VLAN 10).

Il pacchetto ARP arriva allo switch S2 ma non viene inviato allo switch S1, ma solo alla porta Fa0/6, dove risiede PC3.

### Controlla se le VLAN sono state definite correttamente.

Uso il comando `show vlan brief` su S1, S2 e S3 per vedere se le VLAN 10, 20, 30 e 99 sono configurate.

Le VLAN sono state definite ma le porte non sono state associate correttamente, alcune sono state scambiate, le sistemiamo e poi passiamo avanti.

In particolare in S2:

```
S2(config)#interface fastEthernet 0/6
S2(config-if)#switchport mode access
S2(config-if)#switchport access vlan 30
S2(config-if)#interface fastEthernet 0/11
S2(config-if)#switchport mode access
S2(config-if)#switchport access vlan 10
S2(config-if)#interface fastEthernet 0/18
S2(config-if)#switchport mode access
S2(config-if)#switchport access vlan 20
```

### Controlla che le porte sono state definite correttamente (trunk o access)

Vado sullo switch S2.

Uso il comando `S2#show interfaces Gig0/1 switchport`

Vedo che la porta che collega S2 a S1 è marcata come Access e non come trunk, stessa cosa in S1 e S3.

Quindi marco la porta `Gig0/1` di S2 come trunk.

```
S2(config)#interface Gig0/1
S2(config-if)#switchport mode trunk
S2(config-if)#switchport trunk native vlan 99
```

Stessa cosa per `Gig0/1` e `Gig0/2` in S1 e per `Gig0/2` in S3.

Ora il test di connettività intra-VLAN funziona correttamente.

I problemi relativi alle VLAN erano principalmente 2:

1. Interfacce assegnate non correttamente alle VLAN negli Switch S2 e S3.
2. Interfacce trunk di S1, S2 e S3 non configurate correttamente.
  - *Erano configurate come access, appartenenti alla VLAN di default, quindi ovviamente i test di ping della VLAN 10 non potevano attraversare questa porta.*

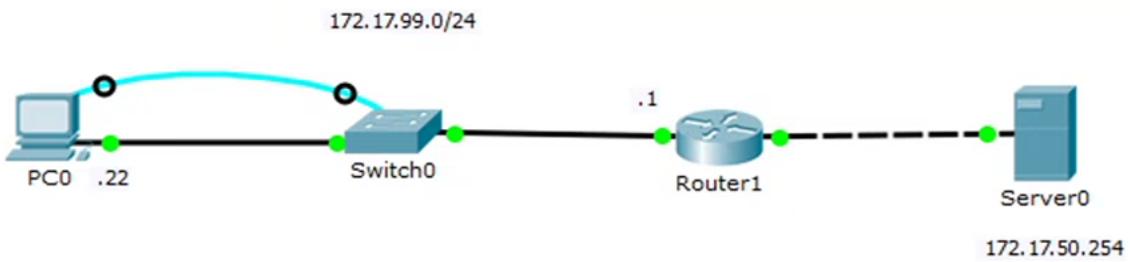
Nei router IP, se un router non conosce dove inoltrare un pacchetto, lo scarta, mentre negli switch si fa il flooding, lo invia in tutte le porte eccetto quello da cui ha ricevuto quel pacchetto, si può fare flooding perché la topologia con gli switch è un albero, ciò che necessariamente non è con i router.

Lo switch è un computer che ha bisogno di un SO per funzionare, il SO a bordo degli switch è lo stesso di quello dei router, quindi loS, **il set di comandi ha un'intersezione non nulla, ma non coincidente con il set di comandi di un router**, ovvero i modi di configurazione sono esattamente quelli visti per i router.

Negli switch c'è un'immagine del SO, e al bootstrap viene cercata prima nella memoria persistente, oppure può cercarla in rete ecc.

La differenza macroscopica con un router è il numero di porte che vedo, sul router il numero di porte è dell'ordine delle unità, in uno switch il numero di porte è dell'ordine di qualche decina.

Adesso andiamo su packet tracer, abbiamo questo schema:



Lo switch lavora a livello 2, non ha senso assegnare indirizzi IP alle porte, le porte Ethernet di un Router sono di livello 3, invece le porte dello switch sono di livello 2. Lo switch, se arriva un frame, non controlla se l'indirizzo MAC di quel frame coincide con il MAC della porta in cui arriva, non ha nemmeno senso assegnare un IP a quella porta e nemmeno un IP diverso per ciascuna porta dato che non devo fare routing da una porta all'altra.

Nell'esempio PC0 è collegato allo switch attraverso un link seriale, per passare nel modo privilegiato si utilizza il comando **enable**.

Con il seguente comando si visualizza la tabella MAC:

```

Sw1#show mac-address-table
      Mac Address Table
-----
Vlan      Mac Address          Type      Ports
----      -----
1        0003.e4ea.0b02    DYNAMIC   Fa0/5

```

L'indirizzo MAC nella tabella o è quello del router oppure è quello dell'host, se controlliamo nel router, notiamo che quell'indirizzo è l'indirizzo MAC della fastEthernet0/1 del router, in questo momento, lo switch non conosce l'indirizzo MAC dell'host, lo conosce non appena l'host invia un frame.

Se voglio configurare un'interfaccia, posso passare alla modalità di configurazione e ad esempio cambiare la comunicazione da half-duplex a full-duplex, si può configurare la velocità di trasmissione delle interfacce:

```

Sw1(config)#int fa0/5
Sw1(config-if)#
  cdp           Global CDP configuration subcommands
  channel-group Etherchannel/port bundling configuration
  channel-protocol Select the channel protocol (LACP, PAgP)
  description   Interface specific description
  duplex        Configure duplex operation.
  exit          Exit from interface configuration mode
  ip            Interface Internet Protocol config commands
  mdix          Set Media Dependent Interface with Crossover
  mls           mls interface commands
  no            Negate a command or set its defaults
  shutdown      Shutdown the selected interface
  spanning-tree Spanning Tree Subsystem
  speed         Configure speed operation.
  storm-control storm configuration
  switchport    Set switching mode characteristics
  tx-ring-limit Configure PA level transmit ring limit
Sw1(config-if)#duplex ?
  auto  Enable AUTO duplex configuration
  full  Force full duplex operation
  half  Force half-duplex operation
Sw1(config-if)#speed ?
  10    Force 10 Mbps operation
  100   Force 100 Mbps operation
  auto  Enable AUTO speed configuration
Sw1(config-if)#mdix ?
  auto  Enable automatic MDI crossover detection on this interface
Sw1(config-if)#mdix

```

Mdix è il meccanismo che configura su quale coppia di collegamenti trasmettere e su quale coppia di collegamenti ricevere, se in modalità automatica la configurazione avviene automaticamente.

Se vogliamo vedere lo stato della porta 5, andiamo a scrivere:

```
--  
Sw1#show interfaces fa0/5 ?  
switchport Show interface switchport information  
<cr>  
Sw1#show interfaces fa0/5  
FastEthernet0/5 is up, line protocol is up (connected)  
Hardware is Lance, address is 00d0.ffcd.4805 (bia 00d0.ffcd.4805)  
BW 100000 Kbit[ DLY 1000 usec,  
    reliability 255/255, txload 1/255, rxload 1/255  
Encapsulation ARPA, loopback not set  
Keepalive set (10 sec)  
Full-duplex, 100Mb/s  
input flow-control is off, output flow-control is off  
ARP type: ARPA, ARP Timeout 04:00:00  
Last input 00:00:08, output 00:00:05, output hang never  
Last clearing of "show interface" counters never  
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0  
Queueing strategy: fifo  
Output queue :0/40 (size/max)  
5 minute input rate 0 bits/sec, 0 packets/sec  
5 minute output rate 0 bits/sec, 0 packets/sec  
    956 packets input, 193351 bytes, 0 no buffer  
    Received 956 broadcasts, 0 runts, 0 giants, 0 throttles  
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort  
    0 watchdog, 0 multicast, 0 pause input  
    0 input packets with dribble condition detected  
    2357 packets output, 263570 bytes, 0 underruns  
--More--
```

Facciamo un ping tra PC0 e router1, viene creata una entry nella tabella MAC, non vediamo l'età però se non si fa più niente, la entry scompare dopo un po', se facciamo un ping tra PC0 ed il server, abbiamo un frame ARP e poi un pacchetto ICMP contenuto in un frame Ethernet, sono due frame che trasportano cose diverse, ma hanno lo stesso sorgente MAC, se vediamo la tabella MAC, vediamo che è comparsa una riga per la Fa0/18:

Vlan	Mac Address	Type	Ports
---	-----	-----	-----
1	0003.e4ea.0b02	DYNAMIC	Fa0/5
1	00d0.baed.lacb	DYNAMIC	Fa0/18

Assegniamo allo switch un indirizzo IP, questa operazione va pensata come se lo switch fosse un host collegato alla stessa LAN, lo switch ha le porte su cui fa switching a livello 2, ha internamente un'interfaccia virtuale SWI, né ha una per ciascuna VLAN configurata nello switch, lo switch ha come predefinita la VLAN 1, quindi per assegnare l'indirizzo, facciamo:

```

Sw1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Sw1(config)#interface vlan1
Sw1(config-if)#ip address 172.17.99.2 255.255.255.0
Sw1(config-if)#no sh
Sw1(config-if)#no shutdown

```

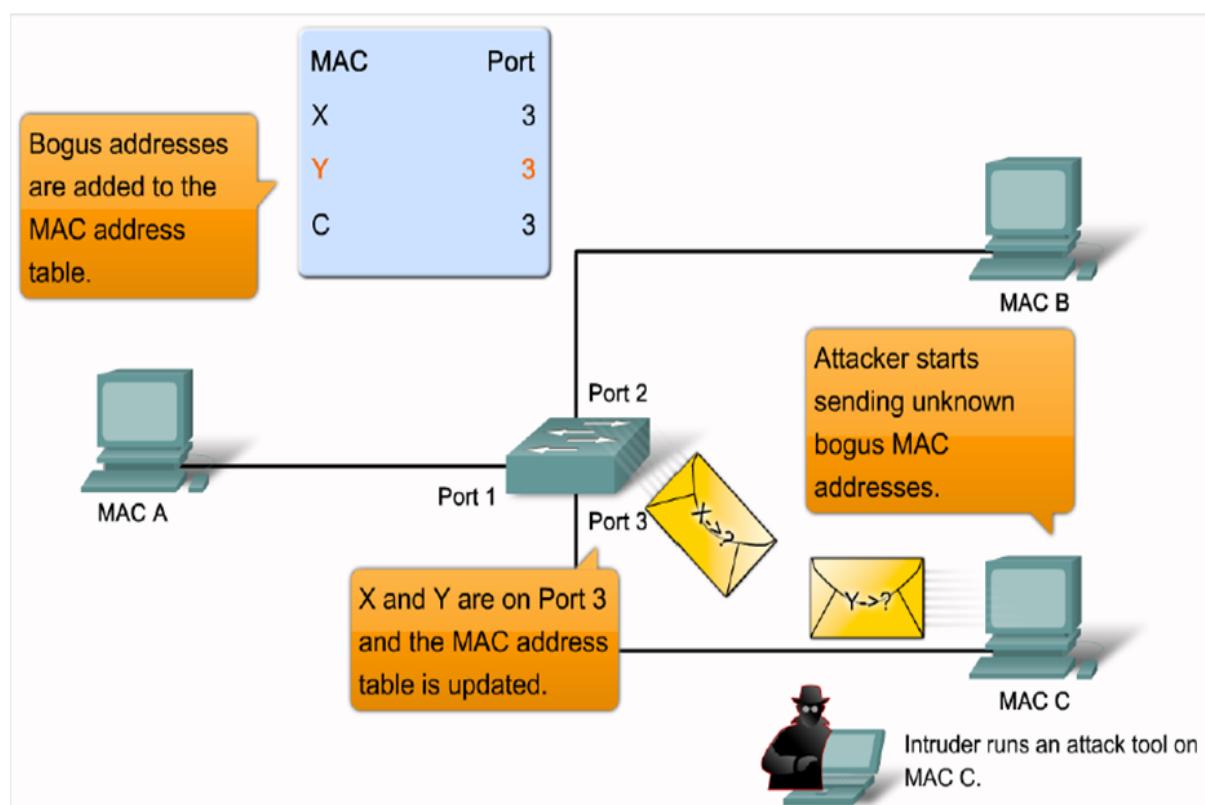
Gli si assegna anche il gateway, e adesso è possibile fare ping verso lo switch:

```

S1#configure terminal
S1(config)#interface vlan 1
S1(config-if)#ip address 172.17.99.2 255.255.255.0
S1(config-if)#no shutdown
S1(config-if)#exit
S1#ip default-gateway 172.17.99.1
S1#end

```

Se si prova ad utilizzare Telnet, non funziona all'inizio perché sullo switch non è configurata una password di accesso dell'interfaccia a riga di comando, in questa situazione, la connessione da remoto è disabilitata.



Nella figura sopra, gli switch hanno un funzionamento di forwarding di eccezione (**Fail Open** sugli switch Cisco), lo switch ha risorse limitate, in particolare la memoria è limitata. **Uno switch quanti indirizzi MAC deve memorizzare?** Quelle delle interfacce della sua stessa LAN, quindi è legato al numero di host contemporaneamente connessi ad una LAN, se tutti si comportano correttamente, mi aspetto che questa tabella non venga saturata mai, ogni

volta che ricevo un frame con indirizzo che non è presente nella tabella, quell'indirizzo non viene memorizzato. Nella figura supponiamo ci sia un cattivo che si collega a quello switch, esso genera frame ad alta velocità cambiando ogni volta l'indirizzo MAC sorgente, l'effetto è saturare la tabella MAC dello switch, dato che memorizza ogni volta un nuovo indirizzo, allora ad un certo punto la tabella è piena e mentre è piena, continuano ad arrivare nuovi pacchetti, affinché lo switch funzioni si introduce il **fail open**, lo switch praticamente diventa un Hub, qualunque frame che riceve in una porta, lo trasmette a tutte le altre. L'utente maligno potrebbe essere interessato al fatto che la rete si comporti in modo fail open perché in questa modalità, tutto viene mandato su tutte le porte, quindi l'utente riceve tutto il traffico che viene scambiato su quella rete.

**Come si impedisce di saturare la tabella MAC dello switch?** Una soluzione potrebbe essere verificare il MAC degli host della rete e configurare gli switch in modo tale che su ogni porta si possano avere frame che provengono da un solo host, è un caso estremo. C'è una soluzione intermedia, io non so l'indirizzo MAC, il primo che vedo però rimane l'unico che potrà comunicare su quella porta.

### Port security

L'idea principale è fare un controllo sugli indirizzi MAC sorgente che ricevo su ciascuna porta e impostare un limite sul numero di indirizzi validi che accetto, il limite può essere uno, o più di uno, ma comunque un numero limitato. L'altro aspetto è come divido questi indirizzi, o lo faccio **staticamente**, quindi a mano, oppure vengono appresi **dinamicamente**, il limite massimo è uno, l'unico valido è il primo che passa, gli altri sono non validi. Un terzo modo è quello **sticky**, ovvero è un mix tra statico e dinamico, se uso la modalità statica, utilizzo il comando di configurazione, quindi il comando fa parte della configurazione dello switch, ogni volta che lo switch parte esegue quel comando e varrà quell'indirizzo MAC, invece con la modalità dinamica, prendo il primo indirizzo MAC valido, però non c'è nessun comando di configurazione, quindi se spengo lo switch e lo collego ad un altro host, adesso ho collegato il nuovo indirizzo MAC.

Sticky è un mix tra le due, l'apprendimento è dinamico, però se io ho configurato questa modalità, una volta appreso l'indirizzo MAC, questo viene memorizzato nel file di configurazione running, se copio running-config in startup-config è come se avessi forzato la configurazione statica, questa variabilità serve per semplificare il lavoro per chi deve gestire la rete.

Configuriamo la porta 18, utilizzeremo spesso il comando **switchport**, che configura le modalità di switching di una determinata porta.

```
Sw1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Sw1(config)#interface fa0/18
Sw1(config-if)#switchport mode access
Sw1(config-if)#switchport port-security
Sw1(config-if)#switchport port-security maximum 5
Sw1(config-if)#switchport port-security mac-address sticky
Sw1(config-if)#switchport port-security violation ?
    protect    Security violation protect mode
    restrict   Security violation restrict mode
    shutdown   Security violation shutdown mode
Sw1(config-if)#switchport port-security violation restrict
Sw1(config-if)#end
```

Con **maximum** si indica il numero massimo di indirizzi MAC diversi che ammetto su questa porta;

con **mac-address** si indica quale o quali indirizzi MAC sono ammessi.

Aggiungiamo un PC, (PC1) se alla porta 18, connetto PC1 invece che PC0, si attiva la port-security nell'interfaccia Fa0/18

```
Sw1(config)#int fa0/18
Sw1(config-if)#switchport port-security
^
% Invalid input detected at '^' marker.

Sw1(config-if)#switchport port-security
```

Andiamo a vedere lo stato di sicurezza della porta 18:

```
Sw1#show port-security int fa0/18
Port Security           : Enabled
Port Status              : Secure-up
Violation Mode          : Shutdown
Aging Time               : 0 mins
Aging Type               : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses    : 1
Total MAC Addresses       : 0
Configured MAC Addresses : 0
Sticky MAC Addresses      : 0
Last Source Address:Vlan : 0000.0000.0000:0
Security Violation Count : 0
```

---

```
Sw1#
```

Adesso configuriamo in modo statico l'indirizzo MAC ammesso, quello di PC1:

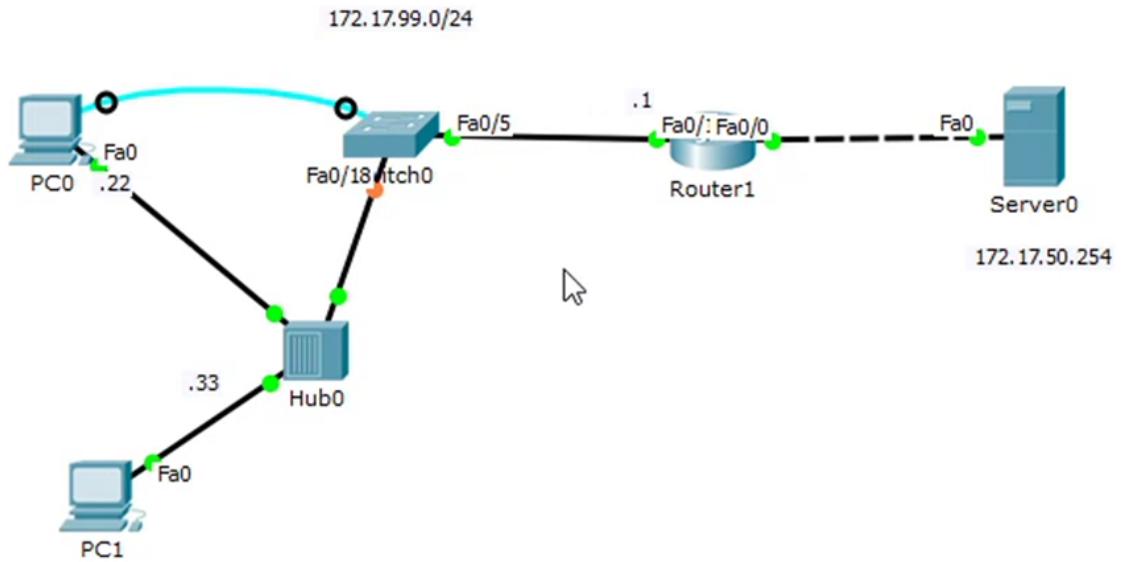
```
Sw1(config-if)#switchport port-security mac
Sw1(config-if)#switchport port-security mac-address 0001.423B.820C
Sw1(config-if)#exit
Sw1(config)#exit
```

Se collegiamo PC0 alla porta 18, nella MAC table abbiamo l'indirizzo di PC1 configurato staticamente, se facciamo ping su PC0, vediamo che non riesce a trasmettere pacchetti.

Le porte dello switch e del PC0 sono diventate rosse, questo perché essendoci stata violazione, lo switch di default spegne la porta, questa cosa si può cambiare indicando di non spegnerla ma di inviare un messaggio ad un certo sistema.

Andiamo sull'interfaccia 18 ed eliminiamo l'indirizzo MAC address, la port security è attiva, quindi il limite ad un indirizzo c'è sempre, non è specificato però l'indirizzo MAC, se facciamo ping su PC0, questa volta ci riesce perché lo switch deve ancora apprendere qual è l'indirizzo MAC.

Adesso prendiamo un Hub, collegiamo la rete in questo modo:



Andiamo sullo switch, si abilita la port security in Fa0/18, PC1 utilizza la rete per primo, lo switch su quella porta l'ha visto, ha memorizzato l'indirizzo MAC, siccome al massimo 1 è il numero di indirizzi MAC diversi che possono essere associati su una porta, allora se PC0 utilizza la rete, si ha una violazione in quanto c'è già un indirizzo.

Supponiamo che io sappia chi è quello che può utilizzare la rete, tra PC0 e PC1, se si utilizza il primo metodo (statico) se deve passare prima PC1, si guarda il MAC di PC1 e si configura, svantaggio è che devo controllare tutti gli indirizzi MAC degli autorizzati, nel caso dinamico non ho il controllo, nel caso sticky rendo il meglio dei due.

Configuriamo in modalità sticky con tale comando:

```
Sw1(config-if)#switchport port-security mac-address
sticky
Sw1(config-if)#exit
Sw1(config)#exit
```

Se guardiamo la port security dello switch vediamo che sticky è diventato 1, e da solo ha rilevato il MAC di PC1 (Noi in questo metodo non abbiamo mai inserito l'indirizzo). Lo ha salvato in automatico, se quindi salvo in startup e riavvio, quell'indirizzo MAC è scritto lì, se PC0 prova a fare ping, la porta va in violazione.

Il problema della sicurezza su switch può riguardare anche il DHCP, potrebbe esserci un server DHCP falso che possa dare ad un host un indirizzo IP valido, ma un gateway falso, ovvero il proprio, per indirizzare il traffico verso di esso (**DHCP spoofing**), come ci si protegge da questo meccanismo? Su ciascuna porta dello switch di accesso, si va a dire quali sono quelle su cui pacchetti di risposta a richieste DHCP sono ammessi.

### Design Gerarchico di una LAN Ethernet

[Immagine slide 2]

- **Switch di Accesso**
  - Switch a cui sono connessi i dispositivi finali, come PC o altro.
- **Switch di Distribuzione**
  - Switch a cui non sono connessi altri dispositivi ma solo altri switch, il loro compito è quello di inoltro dei frame da una porzione di frame all'altra.
  - I link che collegano gli switch di distribuzione a quelli di accesso devono avere una capacità molto alta per evitare colli di bottiglia.
  - Spesso sono collocati insieme agli switch di accesso.
- **Switch di Core**
  - Ha il compito di collegare gli switch di distribuzione tra loro.
  - Gli switch core sono tutti connessi tra loro.
  - Ha il compito di collegare anche gli apparati di rete come i router.

Nelle reti piccole, il livello distribution e il livello core sono uniti in uno solo.

Ho sempre almeno due livelli, in particolare: il livello di accesso c'è sempre, a prescindere dalla dimensione della rete.

#### *Punto debole del Sistema*

Il punto debole di questa topologia è la mancanza di ridondanza, il guasto di un qualsiasi switch in questa rete porta all'isolamento di una specifica porzione di rete.

#### *Perché accade questa cosa?*

Essendo topologicamente un albero, c'è un unico percorso all'interno di quella rete attraverso il quale 2 foglie possono comunicare, nel momento in cui elimino un qualsiasi nodo o arco, un insieme di percorsi smettono di esistere.

#### *Tolleranza ai Guasti della rete*

Questo svantaggio non è accettabile del tutto, io vorrei che la mia rete possa continuare ad operare in termini di connettività in presenza di uno o più guasti.

#### **Uno dei metodi per aumentare la tolleranza ai guasti è la ridondanza.**

Ossia faccio in modo che per collegare due foglie ci siano sempre almeno 2 percorsi disponibili.

[Immagine slide 3]

La figura che vediamo rappresenta una rete con gli stessi switch (nodi) di prima, ma ho più link (archi).

Nello specifico:

- Ciascuno switch di accesso è più collegato a due switch di distribuzione diversi.
- Ciascuno switch di distribuzione è più collegato a due switch di core diversi.
- Gli Host continuano ad essere collegati ad un unico switch di accesso.

Quindi ho (circa) raddoppiato il numero di link nella mia rete, ma ora se si guastasse un link o uno switch qualsiasi, la mia rete continuerebbe a mantenere la stessa connettività di prima.

In questo caso specifico: se si guastano due link o due switch o un link e uno switch contemporaneamente non è garantita la connettività di prima.

### **Problemi della Ridondanza**

La ridondanza non porta solo benefici, ma a volte può portare anche a degli svantaggi.

La ridondanza porta alla creazione di percorsi chiusi e quindi portando la topologia a non essere più un albero.

### **Problemi della Ridondanza - Duplicate Unicast Frames**

Supponiamo che PC1 voglia inviare un frame a PC4.

Il frame in questo esempio è di tipo UNICAST (diretto ad uno specifico Host) quindi nel campo address del frame è specificato il MAC di PC4.

1. PC1 confeziona il frame.
  - a. *PC1 mette il MAC di destinazione di PC4 nell'intestazione del frame.*
2. PC1 invia il frame a S2.
3. S2 riceve il frame e consulta la tabella MAC per decidere dove mandarlo.
  - a. *Supponiamo che nella tabella di S2 l'indirizzo MAC di PC4 non c'è.*
4. S2 invia il frame su tutte le sue porte.
  - a. *Tranne quella di PC1.*
5. PC2 riceve il frame e lo scarta.
6. PC3 riceve il frame e lo scarta.
7. S1 riceve il frame e consulta la tabella MAC per decidere dove mandarlo.
  - a. *S1 lo inoltra a PC4.*
8. S3 riceve il frame e consulta la tabella MAC per decidere dove mandarlo.
  - a. *Il fatto che S3 conosca PC4 non è rilevante.*
9. S3 invia il frame sulla porta verso S1.
10. S1 riceve il frame e consulta la tabella MAC per decidere dove mandarlo.
  - a. *S1 lo inoltra a PC4.*

#### PC4 riceve due volte lo stesso frame.

Questo però non viola la regola del protocollo IP.

Il protocollo IP è di tipo *best effort*, quindi quando trasmetto il pacchetto:

- Non ho garanzie che arrivi.
- Non ho garanzie che arrivi integro.
- Non ho garanzie che ne arrivi una copia sola.
- Non ho garanzie che l'ordine della sequenza venga rispettato.

Però questa cosa vorrei comunque che non accadesse.

### **Problemi della Ridondanza - Broadcast Storm**

Supponiamo che PC1 voglia inviare un frame broadcast a tutta la rete.

Un esempio di frame broadcast è il pacchetto ARP, quindi un tipo di pacchetto che viene inviato regolarmente su tutte le reti.

1. PC1 confeziona il frame.
  - *PC1 mette il MAC di broadcast nell'intestazione del frame.*
2. PC1 invia il frame a S2.
3. S2 riceve il frame e lo inoltra su tutte le sue porte.
  - *Tranne quella di PC1.*
4. PC2 riceve il frame.
5. PC3 riceve il frame.
  
6. S1 riceve il frame da S2 e lo inoltra su tutte le sue porte.
  - S1 invia il broadcast anche a S3.
7. S1 riceve il frame da S3 e lo inoltra su tutte le sue porte.
  - S1 invia il broadcast anche a S2.
  
8. S3 riceve il frame da S2 e lo inoltra su tutte le sue porte.
  - S3 invia il broadcast anche a S1.
9. S3 riceve il frame da S1 e lo inoltra su tutte le sue porte.
  - S3 invia il broadcast anche a S2.

Noto che questo frame broadcast continuerà a circolare tra gli switch, portando a saturare la capacità della rete.

Inoltre gli host PC1, PC2 e PC3 continueranno a ricevere questo frame broadcast.

Ammenochè i frame non abbiano un TTL (ossia un numero massimo di HOP).

### **Protocollo Spanning Tree - STP**

Progettato da una **donna**.

Nel momento in cui viene inviato un frame broadcast, ogni dispositivo lo riceverà una sola volta.

A quei tempi (anni 70) gli switch erano molto costosi e una configurazione di questo tipo era una cosa lontana.

Al massimo si avevano meno di cinque switch.

Questo protocollo nacque per le dimostrazioni di come funzionavano questi switch, perché nelle dimostrazioni avveniva un broadcast storm.

Quale è la filosofia del protocollo?

Lo switch abilita e disabilita le sue porte in modo che da un punto di vista logico la topologia risulti un albero.

Notare che questo protocollo è “*figlio del suo tempo*”, ossia dà per scontato che il sistema sia estremamente distribuito, perché durante la guerra fredda la parola “*centralizzata*” era fuori dal mondo.

Perché con una soluzione estremamente centralizzata (come una rete di tipo SDN) si potrebbe risolvere questo problema in modo molto più semplice.

Come funziona STP?

Faccio il power cycle su packet tracer.

Gli switch fanno il bootstrap.

Le porte degli switch sono tutte bloccate (arancioni).

Dopo un pò di tempo, lo spanning tree viene individuato.

Alcune porte degli switch vengono portate allo stato di forwarding (verdi)  
“*Sono accese e se arriva un frame esso viene inoltrato regolarmente*”.

Mentre le altre le lascia bloccate (arancioni).

“*Sono accese ma se arriva un frame esso viene rifiutato*”.

Uno switch può ricevere copie duplicate (dovute ai percorsi chiusi), ma ho la garanzia che quelle copie verranno ricevute su porte bloccate e quindi verranno scartate a prescindere.

*Questo protocollo rende la rete non ridondante?*

No, nel momento in cui cambia la topologia (si rompe un link o uno switch), STP ricalcola dinamicamente l'albero e ne crea uno nuovo.

Quando un link cade.

Il protocollo STP (che gira su uno switch specifico) dopo un pò di tempo (durante il quale è possibile che una parte della rete risulti isolata) riconosce che il collegamento è morto e mi aspetto che una porta che precedentemente era bloccata diventi disponibile.

Quindi STP cosa fa in soldoni: “*STP rompe la ridondanza temporaneamente per presentare ai frame una topologia ad albero, però in presenza di guasti STP ricalcola l'albero e continua a mantenere la rete connessa*”.

Quindi STP permette di giovare dei benefici della ridondanza (ossia la tolleranza ai guasti) ma di evitare i problemi dovuti ai percorsi chiusi.

### Come funziona STP nello specifico

Noi non vediamo la parte relativa al cambio di topologia, vedremo solo la parte iniziale, ossia come crea l'albero.

L'idea è “*Selezionare un sottoinsieme di porte, sulle quali viene disabilitato il forwarding dei frame in entrambe le direzioni*”.

L'albero viene costruito in 3 fasi, le quali possono essere eseguite in modo concorrente (ossia possono essere mischiate tra di loro):

1. **Root Election**
  - Tra tutti gli switch (*bridge*) viene eletto una Radice (*Root*).
2. **Root Port Selection**
  - Gli altri switch (*Non Root*) scelgono tra le proprie porte la porta con cui raggiungere la Root (*Root Port*).
3. **Designated Port Selection**
  - Gli altri switch decidono autonomamente cosa fare con le altre porte.

Gli switch si scambiano informazioni con frame ethernet che contengono un pacchetto BPDU (Bridge Protocol Data Unit).

All'inizio le porte degli switch sono tutte bloccate, ma attraverso queste porte i pacchetti BPDU possono passare, contrariamente a tutto il resto.

Nei pacchetti BPDU si specificano molte cose tra cui:

- 1. Flags [ 1 Byte ]**
- 2. Root ID [ 8 Byte ]**
  - Identificatore dello Switch che è stato eletto Root.
- 3. Bridge ID [ 8 Byte ]**
  - Identificatore dello Switch che ha trasmesso questo BPDU.
- 4. Root Path Cost [ 4 Byte ]**
  - Ha un numero.
  - Indica il costo del percorso verso il Root Switch partendo dallo Switch che ha trasmesso questo BPDU.
- 5. Port ID [ 2 Byte ]**
  - Identificatore della porta (da cui è stato inviato questo BPDU) dello switch (che ha inviato questa BPDU).