Progetto di Reti di Calcolatori 2
-
Implementazione di una rete dorsale

Giorgio Ravera

A.A 2007/2008

Indice

Obiettivo

L'obiettivo del progetto è quello di realizzare una rete dorsale utilizzando apparati CISCO e NOKIA sulla base dei seguenti concetti visti a lezione:

- vlan
- frame relay
- ppp
- algoritmi di routing (IGP e BGP)
- MPLS
- VPN
- firewall

Il progetto è stato realizzato con la collaborazione del Prof. P.P. Baglietto e del Dott. Stefano Lassi.

Capitolo 1

Schema Globale

La rete che si è realizzata è riportata in figura 1.1.

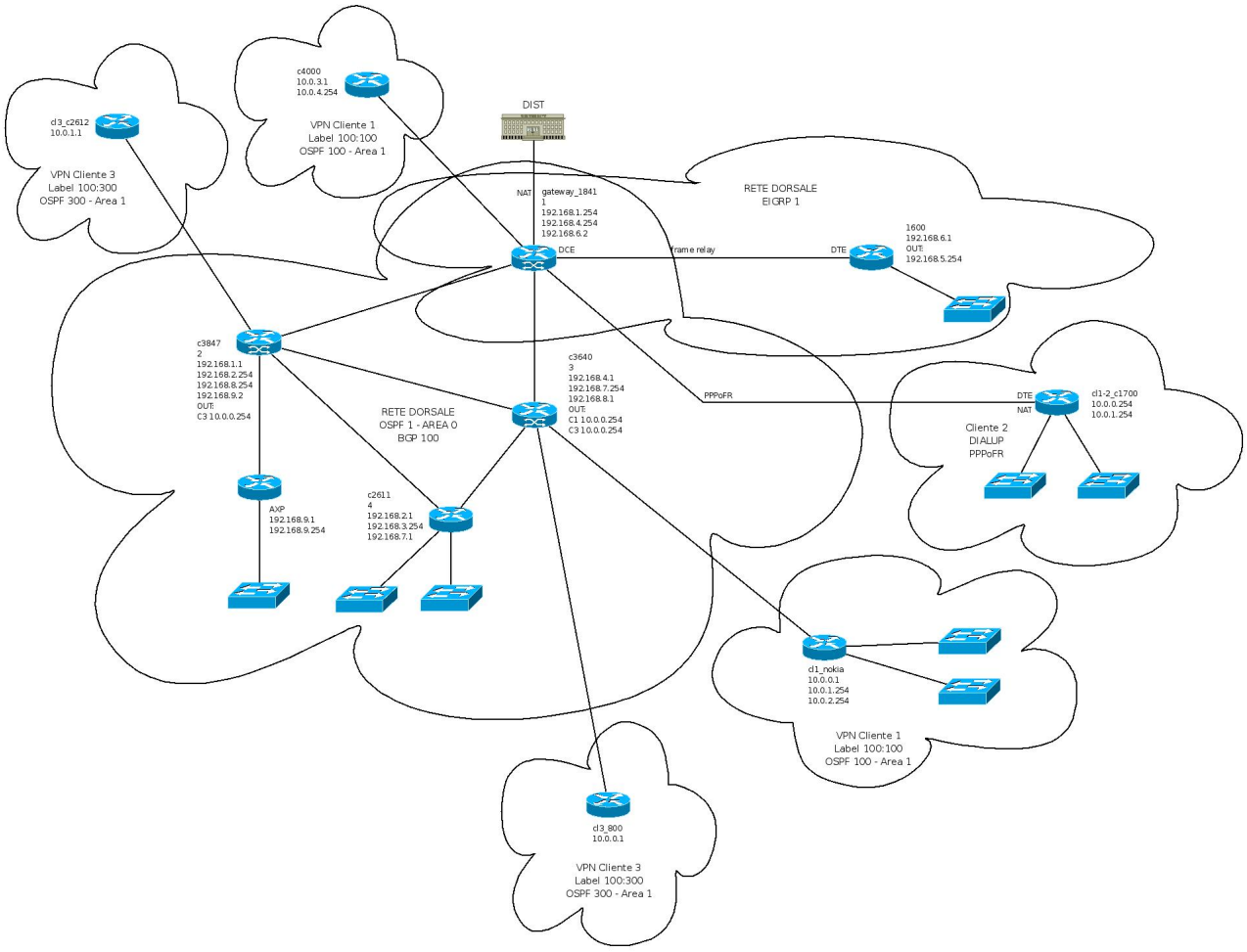

Figura 1.1: Schema della rete

Si possono identificare i seguenti elementi principali:

- Rete Dorsale Principale
- Rete Dorsale Secondaria
- VPN Cliente 1
- Rete Cliente 2
- VPN Cliente 3 (divisa in 2 regioni)
- Link verso Internet

1.1 Rete Dorsale Principale

La rete dorsale principale è composta da 4 router cisco:

- 1841
- 3825
- 3640
- 2611

Il livello fisico è realizzato attraverso **ethernet** sul quale vengono trasportati **pacchetti IP**. Per simulare una vera dorsale, tali router sono stati collegati tra loro attraverso più collegamenti in modo da garantire l'attività qualora uno o più link venissero scollegati. Si è utilizzato, come algoritmo di routing **OSPF** in cui tutte le reti sono dichiarate appartenenti all'area 0.

Sempre di questa rete fa parte il modulo Application eXtendible Platform, collegato come Network Module, al router 3800 che di fatto costituisce un router indipendente. La comunicazione con il reso della rete avviene attraverso un'interfaccia virtuale che collega il modulo con il router ospitante sulla qualle è stata creata una rete punto a punto su ethernet.

1.2 Rete Dorsale Secondaria

La rete è composta da 2 router cisco:

- 1841
- 1601

Il livello fisico è realizzato attraverso un **collegamento seriale** sul quale vengono vengono incapsulati **pacchetti IP** tramite **frame-relay**. Si è scelto di utilizzare l'algoritmo di routing **EIGRP** proprietario di Cisco. L'idea di realizzare questo ramo di rete è nata dal fatto di poter utilizzare link fisici differenti e algoritmi di routing diversi per poterli ridistribuire uno sull'altro.

1.3 VPN Cliente 1

La rete del cliente 2 è divisa in due blocchi collegati tra loro tramite la dorsale. I router utilizzati, Cisco 4000 per la prima parte e Nokia IP 120 per la seconda, comunicano tra loro attraverso la rete dorsale che incapsula i pacchetti ip in **pacchetti MPLS**. L'algoritmo di routing utilizzato è **OSPF** che viene opportunamente ridistribuito all'interno della dorsale stessa tramite il protocollo **BGP** tra i nodi di frontiera.

Logicamente le due reti sono interconnesse direttamente tra loro. La rete MPLS è totalmente trasparente all'utente che si dovrà limitare a configurare in maniera opportuna i due router di frontiera con le specifiche fonrite dal provider per il protocollo di routing interno (OSPF) e impostare gli ip dei rispettivi default gateway (router della dorsale che aggiungono o rimuovono dei label MPLS) dei due segmenti.

1.4 Rete Cliente 2

La rete di un ipotetico Cliente 2 è collegata alla dorsale utilizzando il **protocollo PPP** incapsulato su linea framerelay. Tutti i dati di connessione (default gateway, dns e ip) vengono negoziati tramite il protocollo stesso. Il router del cliente, in questo caso il cisco 1700 gestisce due reti separate che vengono mascherate in uscita verso la dorsale. All'interno della rete del cliente si è utilizzato il protocollo di routing **EIGRP**.

Questa situazione descrive un possibile scenario di utente casalingo che ha più di una rete e vuole connettere più macchine ad internet disponendo di un solo indirizzo ip pubblico. In questo caso, l'ip assegnato apparterrà alla dorsale stessa. L'idea era quella di utilizzare ATM e PPPoA ma non si disponevano di interfacce che lo supportassero.

1.5 VPN Cliente 3

Come la rete del cliente 1, anche quella del cliente 3 è divisa in due blocchi collegati tra loro tramite la dorsale. I router utilizzati, Cisco 2612 per la prima parte e Cisco 877 per la seconda, comunicano tra loro attraverso la rete dorsale che incapsula i pacchetti ip in **pacchetti MPLS**. L'algoritmo di routing utilizzato è **OSPF** che viene opportunamente ridistribuito all'interno della dorsale stessa tramite il protocollo **BGP** tra i nodi di frontiera.

Da notare che gli ip utilizzati all'interno della rete Cliente 1 e Cliente 3 sono gli stessi e quindi, senza MPLS andrebbero in collisione. Il fatto di utilizzare i label (100:100 per la

prima, 100:300 per la seconda). Anche gli algoritmi di routing vengono instradati tramite due istanze di routing virtuale differente (vrf Cliente1 e vrf Cliente3).

Capitolo 2

Vlan

2.1 Descrizione Generale

Per poter interconnettere tra loro i vari apparati si è scelto di utilizzare uno switch Cisco Catalyst 2950, dotato di 12 porte FastEthernet (10-100 Mbps). Per separare i domini di collisione di ogni rete si è scelto di usare il protocollo IEEE 802.1Q. In questo modo si è potuta realizzare una separazione logica delle reti.

In linea di principio, si è creata una vlan per ogni rete presente nello schema. In rari casi si è interconnesso i router direttamente tra loro per risparmiare porte dello switch. Per dare un idea più precisa viene riportato l'output parziale del comando **show vlans**:

```
VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
1    default                           active    Fa0/17, Fa0/18, Fa0/19, Fa0/23
                                                 Gi0/1, Gi0/2
10   RETE_1                            active    Fa0/7, Fa0/8
20   RETE_2                            active    Fa0/9, Fa0/10
30   RETE_3                            active    Fa0/11, Fa0/12
40   RETE_4                            active    Fa0/13, Fa0/14
50   RETE_5                            active    Fa0/15, Fa0/16
70   RETE_7                            active
80   RETE_8                            active
100  Cliente1                          active    Fa0/1, Fa0/2
200  Cliente2                          active    Fa0/3, Fa0/4
300  Cliente3                          active    Fa0/5, Fa0/6
```

Come si può vedere dall'output del comando, le ultime 8 porte (17-24) non sono presenti in lista o appartengono alla vlan 1. Questo accade perché tali porte sono impostate in modalità **trunk**. Nel caso in cui lo stesso router sia connesso a più vlan si è scelto di connetterlo ad una sola porta ethernet dello switch impostandola in questa modalità. Nel router, è necessario dichiarare tante sotto interfacce quante sono le vlan al quale si desidera connetterlo e, per ognuna dichiarare un incapsulazione dot1q.

Da notare che sono state dichiarate due vlan (70 e 80) pur non avendo alcuna porta in access. Questa operazione è necessaria perché il traffico di tali porte passa ugualmente nelle interfacce dichiarate in trunk. Se non fossero state definite, i pacchetti ad esse associate non verrebbero instradati nelle porte in trunk.
Ecco, per completezza, un estratto del comando **show interfaces trunk**:

```
Port        Mode         Encapsulation  Status        Native vlan
Fa0/20      on           802.1q         trunking      1
Fa0/21      on           802.1q         trunking      1
Fa0/22      on           802.1q         trunking      1
Fa0/24      on           802.1q         trunking      1

Port        Vlans allowed on trunk
Fa0/20      1-4094
Fa0/21      1-4094
Fa0/22      1-4094
Fa0/24      1-4094

Port        Vlans allowed and active in management domain
Fa0/20      1,10,20,30,40,50,70,80,100,200,300
Fa0/21      1,10,20,30,40,50,70,80,100,200,300
Fa0/22      1,10,20,30,40,50,70,80,100,200,300
Fa0/24      1,10,20,30,40,50,70,80,100,200,300
```

In questo output non compaiono le porte 17 18 19 e 23 che nella schermata precedente erano state rilevate come appartententi alla vlan 1. Questo avviene perchè a tali porte non è connesso nulla.
Ecco un estratto del comando **show interfaces**:

```
Vlan10 is up, line protocol is up
  Hardware is EtherSVI, address is 001e.1330.2f41 (bia 001e.1330.2f41)
  Internet address is 192.168.1.253/24
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
.....
Vlan20 is up, line protocol is up
  Hardware is EtherSVI, address is 001e.1330.2f42 (bia 001e.1330.2f42)
  Internet address is 192.168.2.253/24
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
.....
FastEthernet0/1 is up, line protocol is up (connected)
  Hardware is Fast Ethernet, address is 001e.1330.2f03 (bia 001e.1330.2f03)
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
```

```
     reliability 255/255, txload 1/255, rxload 1/255
.....
FastEthernet0/2 is down, line protocol is down (notconnect)
  Hardware is Fast Ethernet, address is 001e.1330.2f04 (bia 001e.1330.2f04)
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
.....
FastEthernet0/19 is down, line protocol is down (notconnect)
  Hardware is Fast Ethernet, address is 001e.1330.2f15 (bia 001e.1330.2f15)
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
.....
FastEthernet0/20 is up, line protocol is up (connected)
  Hardware is Fast Ethernet, address is 001e.1330.2f16 (bia 001e.1330.2f16)
  MTU 1500 bytes, BW 100000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
```

E' interessante notare che questo switch di livello 3 può possedere più di un indirizzo ip, uno per vlan. Lo si può vedere dall'interfaccia virtuale vlan10 e vlan20 corrispondenti alle vlan ominime.

2.2 Configurazione Access

Impostare una porta in modalità access su una vlan è molto semplice:

```
switch_mpls# configure terminal
switch_mpls(config)#interface FastEthernet 0/1
switch_mpls(config-if)#switchport mo
switch_mpls(config-if)#switchport mode acc
switch_mpls(config-if)#switchport mode access
switch_mpls(config-if)#switchport access vlan 100
switch_mpls(config-if)#end
switch_mpls#write mem
Building configuration...
[OK]
switch_mpls#
```

Una tale procedura genera, nel file di configurazione, le seguenti righe:

```
interface FastEthernet0/1
 switchport access vlan 100
 switchport mode access
```

Come si può vedere l'output dei comandi riflette esattamente ciò che verrà inserito nel file di configurazione. Pertanto, nel proseguimento del testo, verranno solamente mostrati i file di configurazione.

2.3 Configurazione Trunk

Una porta, per essere configurata in modalità trunk, necessita la specifica del protocollo di incapsulamento da utilizzare. In questo caso si è scelto IEEE 802.1Q e quindi il protocollo, in ambiente cisco, prende il nome di **dot1q**.
Questo è un esempio di configurazione di un interfaccia in trunk:

```
interface FastEthernet0/17
 switchport trunk encapsulation dot1q
 switchport mode trunk
```

2.4 Spanning Tree

2.5 Cisco Discovery Protocol

CDP è un protocollo che consente di identificare gli apparati cisco direttamente connessi. Per avere un idea della sua utilità viene presentato un output del comando **show cdp neighbors**

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

Device ID        Local Intrfce     Holdtme    Capability  Platform  Port ID
cl1-2_c1700      Fas 0/3           164           R S      1721      Fas 0
gateway_1800     Fas 0/21          127         R S I      1841      Fas 0/1
cl3_c800         Fas 0/5           161         R S I      877       Fas 0
c3640            Fas 0/22          126         R S I      3640      Eth 0/0
c1600            Fas 0/15          162           R        1601      Eth 0
c2611            Fas 0/24          153           R        2611      Eth 0/0
c3825            Fas 0/20          179         R S I      3825      Gig 0/0
```

Appare chiaro come sono collegati i componenti tra loro e su quale porte vengono visti. Questo strumento può essere di grande aiuto. Per completezza si consiglia di visionare il file di configurazione dello switch nel paragrafo 6.1.1.

Capitolo 3

Frame Relay

3.1 Descrizione Generale

l Frame Relay o (Frame-mode Bearer Service) è una tecnica di trasmissione a commutazione di pacchetto su interfacce seriali, introdotta dalla ITU-T a partire dalla raccomandazione I.122 del 1988. Può essere pensato come una linea virtuale affittata tra 2 punti tra i quali si possono inviare frame di lunghezza variabile. Rispetto a X.25 fornisce un servizio minimale; infatti, non fornisce informazioni sull'avvenuta ricezione e non fornisce controllo di flusso.

Permette di inviare dati con banda a richiesta: lutente può richiedere banda più alta secondo il bisogno. Comunque si possono specificare banda minima garantita oltre che banda massima. Implementa solo lo strato fisico e quello di data-link (rispettivamente i livelli 1 e 2 del modello OSI), quindi può essere usato, per esempio, come trasporto per IP su internet. Viene usato per connettere diverse LAN su WAN, dal momento che permette di trasmettere frame di dimensione massima 8192 byte, e può mandare pacchetti senza problemi di frammentazione. Simula quindi una LAN estesa su WAN.

Nel progetto sono state realizzate due connessioni seriali:

- Rete Dorsale Secondaria
- Rete Cliente 2

In entrambe le reti si è scelto di incapsulare i pacchetti (IP nel primo caso e PPP nel secondo) in frame-relay. Per realizzare fisicamente un collegamento framerelay si possono scegliere due tecniche di realizzazione dei circuiti:

- Permanent Virtual Circuit: PVC
- Switched Virtual Circuit: SWC

Per semplicità si è scelto di creare il collegamento definendo a priori il circuito e basandoci su una mappaura IP-circuito dinamica e quindi automatica.

Frame relay usa due tipologie di interfacce: DTE (Data Terminal Equipment), utilizzata lato cliente, e DCE (Data Circuit-Terminating Equipment) usata lato provider. La

differenza deriva dal fatto che lato provider arrivano più circuiti virtuali (uno per utente) e serve uno switch che li commuti correttamente.

3.2 DTE

Lato utente la configurazione è molto semplice e viene riassunta dal seguente file di configurazione:

```
interface Serial0
 no ip address
 encapsulation frame-relay
 frame-relay lmi-type cisco
 ip address 192.168.6.1 255.255.255.252
 frame-relay interface-dlci 60
```

Come si può vedere il circuito virtuale permanente è il numero 60. Tale numero è identificato dal **DLCI** (Data Link Connection Identify). Sempre dal file di configurazione si può dedurre che esistono diverse implementazioni di **LMI** (Local Management Interface), uno standard di segnalamento tra router in ambiente frame-relay che fornisce informazioni su indirizzi globali, ip, circuiti e stati delle connessioni, i più comuni sono:

- cisco
- asci
- q933a

In caso di assenza di tale parametro, viene attivata una procedura di autosense. Da notare che non vengono indicate alcune mappature ip circuito virtuale. Questo indica che si utilizza un Dynamic Map. Per renderlo statico bastava aggiungere la seguente istruzione:

```
frame-relay map ip 192.168.6.2 60
```

che indicava che l'ip 192.168.6.2 si trovava all'altro capo del circuito virtuale 60.

In questa configurazione si può associare un solo circuito virtuale per linea seriale. E' possibile assegnare più di un circuito alla stessa linea tramite la definizionedi ulteriori interfacce virtuali. I parametri relativi alla linea resteranno nell'area relativa all'interfaccia, quelli specifici del canale (ip ad esempio) andranno in quella virtuale.

```
interface Serial0
 no ip address
 encapsulation frame-relay
 frame-relay lmi-type cisco
!
interface Serial0.60 point-to-point
```

```
 ip address 192.168.6.1 255.255.255.252
 frame-relay interface-dlci 60
!
```

Da notare che il nome dell'interfaccia virtuale è il numero del canale DLCI non devono essere necessariamente correlati. Lo si fa per semplicità e per mantenere una logica nella configurazione.

3.3 DCE

La configurazione dell'interfaccia DCE è un po' più complicata perché richiede parametri aggiuntivi. Segue un estratto del file di configurazione del router 1841:

```
frame-relay switching
!
...
interface Serial0/0/0
 bandwidth 2000
 no ip address
 encapsulation frame-relay
 clock rate 2000000
 frame-relay lmi-type cisco
 frame-relay intf-type dce
!
interface Serial0/0/0.60 point-to-point
 ip address 192.168.6.2 255.255.255.252
 ip nat inside
 ip virtual-reassembly
 frame-relay interface-dlci 60
!
```

Come si può osservare è necessario, in questo caso, specificare sia la banda che il clockrate. Il DTE si calcola questi parametri dal DCE al quale è connesso e il DCE stesso li deve imporre. E' anche necessario indicare la tipologia di interfaccia (frame-relay intf-type dce) perché di default una seriale in frame-relay è imostata in modalità DTE.

Da notare frame-relay switching. Questa impostazione consente di fare switching tra due o più canali virtuali tramite interfacce dce. Quando si ha un interfaccia DCE è necessario impostare questa opzione.
Per comprendere meglio si osservino gli output sottostanti:

```
LATO DTE

c1600#show interfaces serial 0
Serial0 is up, line protocol is up
```

```
  Hardware is QUICC Serial
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation FRAME-RELAY, loopback not set
.....

c1600#show interfaces serial 0.60
Serial0.60 is up, line protocol is up
  Hardware is QUICC Serial
  Internet address is 192.168.6.1/30
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation FRAME-RELAY
  Last clearing of "show interface" counters never

c1600#show frame-relay pvc

PVC Statistics for interface Serial0 (Frame Relay DTE)

               Active     Inactive      Deleted       Static
  Local           1            0            0            0
  Switched        0            0            0            0
  Unused          0            0            0            0

DLCI = 60, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0.60

  input pkts 266           output pkts 254          in bytes 22029
  out bytes 23311          dropped pkts 0           in pkts dropped 0
  out pkts dropped 0                 out bytes dropped 0
  in FECN pkts 0           in BECN pkts 0           out FECN pkts 0
  out BECN pkts 0          in DE pkts 0             out DE pkts 0
  out bcast pkts 134       out bcast bytes 12880
  5 minute input rate 1000 bits/sec, 3 packets/sec
  5 minute output rate 1000 bits/sec, 2 packets/sec
  pvc create time 00:12:33, last time pvc status changed 00:04:43

c1600#show frame-relay lmi interface serial 0

LMI Statistics for interface Serial0 (Frame Relay DTE) LMI TYPE = CISCO
  Invalid Unnumbered info 0 Invalid Prot Disc 0
  Invalid dummy Call Ref 0 Invalid Msg Type 0
  Invalid Status Message 0 Invalid Lock Shift 0
  Invalid Information ID 0 Invalid Report IE Len 0
```

```
  Invalid Report Request 0 Invalid Keep IE Len 0
  Num Status Enq. Sent 85 Num Status msgs Rcvd 66
  Num Update Status Rcvd 0 Num Status Timeouts 19
  Last Full Status Req 00:00:13 Last Full Status Rcvd 00:00:13

c1600#show frame-relay map
Serial0.60 (up): point-to-point dlci, dlci 60(0x3C,0xCC0), broadcast
          status defined, active

LATO DCE
gateway_1800#show interfaces serial 0/0/0
Serial0/0/0 is up, line protocol is up
  Hardware is GT96K Serial
  MTU 1500 bytes, BW 2000 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation FRAME-RELAY, loopback not set
.....

gateway_1800#show interfaces serial 0/0/0.60
Serial0/0/0.60 is up, line protocol is up
  Hardware is GT96K Serial
  Internet address is 192.168.6.2/30
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation FRAME-RELAY
  Last clearing of "show interface" counters never

gateway_1800#show frame-relay pvc

PVC Statistics for interface Serial0/0/0 (Frame Relay DCE)

              Active     Inactive      Deleted       Static
  Local          1            0            0            0
  Switched       0            0            0            0
  Unused         0            0            0            0

DLCI = 60, DLCI USAGE = LOCAL, PVC STATUS = ACTIVE, INTERFACE = Serial0/0/0.60

  input pkts 21979         output pkts 21899        in bytes 1816000
  out bytes 1844280        dropped pkts 0           in pkts dropped 0
  out pkts dropped 0                 out bytes dropped 0
  in FECN pkts 0           in BECN pkts 0           out FECN pkts 0
  out BECN pkts 0          in DE pkts 0             out DE pkts 0
```

```
  out bcast pkts 20699     out bcast bytes 1778215
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  pvc create time 1d00h, last time pvc status changed 00:10:39
.....

gateway_1800# show frame-relay map
Serial0/0/0.60 (up): point-to-point dlci, dlci 60(0x3C,0xCC0), broadcast
          status defined, active
.....
```

3.4 SVC

Per impostare SVC si sarebbe dovuto inserire una serie di comandi simili alla seguente:

```
interface serial 0
 ip address 192.168.6.2 255.255.255.252
 encapsulation frame-relay
 map-group GR_NAME
 frame-relay lmi-type q933a
 frame-relay svc
!

map-list GR_NAME source-addr E164 123456 dest-addr E164 654321
 ip 192.168.1.254 class fast
 ip 192.168.2.1 class slow
!
map-class slow-class
  frame-relay cir in 128000
  frame-relay cir out 56000
!
map-class frame-relay fast
 frame-relay cir in 1000000
 frame-relay cir out 128000
!
```

Fondamentale è il comando map-list che da indicazione su sorgente e destinazione dei pacchetti per stabilire il circuito. Sempre attraverso le map-list si possono definire classi di servizio differenti.

3.5 PPP over Frame Relay

Il **PPP** (Pont-to-Point Protocol) è un protocollo

Le rete del Cliente 2 è stata creata per simulare una connessione dialin da parte di un generico cliente. L'idea originaria era quella di realizzare una connessione PPPoA o PPPoE su ATM nel primo caso, su ATM Lan Emulation nel secndo. Purtroppo, non dispondenti di interfacce seriali o SDH compatibili con tale tecnologia si è dovuto ripiegare su seriale con frame-relay.

L'idea è quella di utilizzare un collegamento seriale sul quale transitano pacchetti IP incapsulati prima in PPP e poi in frame-relay. Anche in questo caso sarà necessario definire l'interfaccia seriale come in precedenza (un capo DTE, l'altro DCE) e predisporre tante interfacce virtuali quanti sono i canali. Nel caso di PPP è necessario utilizzare un ulteriore interfaccia virtuale, chiamata **Virtual-Template** che serve per definire i parametri di atutenticazione e ciò che dovrà essere negoziato (ip, gateway, dns e altro ancora).
Vediamo, per prima cosa, i parametri di configurazione delle due interfacce seriali:

```
LATO DCE - Provider

interface Serial0/1/0
 bandwidth 2000
 no ip address
 encapsulation frame-relay
 no snmp trap link-status
 clock rate 2000000
 frame-relay lmi-type ansi
 frame-relay intf-type dce
!
interface Serial0/1/0.100 point-to-point
 frame-relay interface-dlci 100 ppp Virtual-Template1
!

LATO DTE - Client
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

Come si può notare, la configurazione delle due interfacce è analoga alla precedente. L'unica differenza risiede nell'interfaccia seriale virtuale. é necessario specificare che il canale DLCI (100 nel nostro caso) sia dedicato al traffico di pacchetti PPP e i parametri di tale protocollo sono definiti in un Virtual-Template (1 nel nostro caso). Vediamo ora la configurazione delle due interfacce Virtual-Template:

```
LATO DCE - Provider

username framerelay password 7 030752180500
!
...
!
interface Virtual-Template1
```

```
 ip unnumbered FastEthernet0/1.10
 ip nat inside
 ip virtual-reassembly
 peer default ip address dhcp
 ppp authentication chap pap
 ppp chap hostname gateway
 ppp chap password 7 045802150C2E
 ppp pap sent-username gateway password 7 121A0C041104
!
AAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

La keyword ip *unnumbered* indica che l'indirizzo IP dell'interfaccia sarà quello definito su quella in argomento al comando stesso (in questo caso FastEthernet0/1.10 ha come ip 192.168.1.254). In questo modo si realizza un bridge tra le due interfacce.

Per quanto riguarda l'autenticazione si è scelto di utilizzare, sia **PAP** () che **CHAP** (). Si imposta questa configurazione utilizzando la stringa *ppp authentication chap pap*. Siccome l'autenticazione deve essere fatta solo dal provider, lato client è necessario aggiungere la chiave callin che indica che non è necessario autenticare l'altro apparato.

Nel caso in cui non si usino sistemi di autenticazioni complessi come Radius, l'autenticazione viene fatta sulla base della coppia hostname - password (quest'ultima di default è *cisco*). Pertanto dovranno essere creati i rispettivi utenti su entrambe le macchine. Si possono bypassare tali imposizioni attraverso i comandi *ppp chap password* e *ppp pap sent-username*. Come sempre, attivando il servizio **password-encryption**, nel file di configurazione non verrà mostrata la password in chiaro ma solamente il suo hashcode.

Infine si noti la gestione del peer: lato provider è necessario impostare la chiave *peer default ip address dhcp* che indica al router che dovrà assegnare l'indirizzo al peer tramite dhcp. Il server dhcp che verrà usato sarà quello assegnato alla rete su cui si trova l'interfaccia seriale. Nel nostro caso, siccome è dichiarata unnumbered su FastEthernet 0/1.10 si userà il dhcp della rete 192.168.1.0 che è il router Cisco 3847. Lato client sarà necessario impostare, il fatto che tramite PPP dovranno essere negoziati:

- indirizzo IP - COMANDO
- indirizzo del default gateway - COMANDO
- indirizzo dei server DNS - COMANDO

Capitolo 4

Algoritmi di routing

Quando si realizza una rete di dimensioni molto grandi, composta da diverse sottoreti, è necessario, per ogni router, impostare le tabelle di routing e calcolare un percorso ottimo (o presento tale) verso il default gateway. Se i router sono tanti e di diverse tipologie, la loro configurazione può essere complicata e lunga. In più una minima variazione sulla topologia implicherebbe la riconfigurazione di molti apparati.

Una soluzione per ovviare a simili problemi è rappresentata dall'utilizzo di un appropriato algoritmo di routing. Esistono due tipologie di algoritmi differenti:

- **IGP**: Interior Gateway Protocoll: usato per distribuire le tabelle di routing tra più reti direttamente connesse
- **BGP**: Border Gateway Protocoll: usato nei router di frontiera per interconnettere più reti tra loro

4.1 IGP

I protocolli IGP hanno il compito di distribuire le tabelle di routing agli apparati direttamente connessi e si occupano di trovare il percorso ottimo per collegare tra loro le reti. Sono molto utili in caso di collegamenti multipi tra gli apparati per ricalcolare i percorsi quando un link dovesse interrompersi.
I protocolli IGP più diffusi sono classificati in due categorie differenti in base al loro funzionamento:

- **metrico** : calcola i pesi sulla base degli hop rilevati. I più diffusi sono:
  - **RIP** 1 e 2 (Routing Information Protocoll): il più famoso e facile da configurre ma presenta problemi di gestione e falle di sicurezza.
  - **IGRP** (Interior Gateway Routing Protoll): protocollo proprietario di Cisco sviluppato per migliorare RIP.
  - **EIGRP** (Enhanced Interior Gateway Routing Protoll): evoluzione di IGRP. Consente procedure di autenticazione.

- **link state**: calcola i pesi sulla base dello stato del link. I più diffusi sono:
  - **OSPF** (Open Short Path First): in assoluta è il protocollo più diffuso e performante. Consente procedure di autenticazione e può ragruppare più reti in una struttura gerarchica.
  - **IS-IS**: protocollo poco usato e difficile da configurare

Nel progetto sono stati implementati due algoritmi IGP: EIGRP ed OSPF. Il primo nella Dorsale Secondaria, il secondo nella Primaria con la redistribuzione di uno sull'altro e viceversa.

4.1.1 EIGRP

La configurazione di EIGRP è molto semplice se non si richiedono parametri particolari. E' necessario dichiarare un numero identificativo del processo di routing e inserire per esso le specifiche. E' possibile configurare diverse istanze di EIGRP, come di qualsiasi altro protocollo, impostando opportunamente i parametri.
Si consideri la seguente configurazione:

```
router eigrp 1
 network 192.168.5.0
 network 192.168.6.0 0.0.0.3
 no auto-summary

REDISTRIBUTE SUBNETS
!
```

Da questa configurazione si può capure che l'istanza 1 di EIGRP è attiva sulle interfacce che appartengono alle reti 192.168.5.0/24 e 192.168.6.0/30. Questo vuol dire che su quelle interfacce verranno instradati i pacchetti EIGRP. Qualsiasi apparato che sia in una di quelle due reti sul quale sia attivo un processo EIGRP 1 saprà che il router in questione è connesso a 192.168.5.0/24 e 192.168.6.0/30 e gli comunicherà quali reti sono incluse nel proprio processo di routing.

Se il router in questione, però, fosse connesso ad altre reti, ad esempio 192.168.0.0/24, con questa configurazione, non verrebbe comunicata e su quella rete l'apparato non invierebbe pacchetti EIGRP.

Capitolo 5

MultiProtocol BGP con MPLS

Capitolo 6

File di Configurazione

Per rendere più completa la trattazione del lavoro svolto, vengono allegati i file di configurazione dei vari router utilizzati.

6.1 Switch

6.1.1 Cisco 2950

```
!
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
service password-encryption
!
hostname switch_mpls
!
enable secret 5 $1$i8L2$Fq9Z3LCnYp./xbiIF.dzN1
!
username admin password 7 14141B180F0B
no aaa new-model
system mtu routing 1500
ip subnet-zero
ip routing
ip name-server 130.251.1.4
!
!
!
!
no file verify auto
```

```
!
spanning-tree mode pvst
spanning-tree extend system-id
no spanning-tree vlan 547
!
vlan internal allocation policy ascending
!
interface FastEthernet0/1
 switchport access vlan 100
 switchport mode access
 spanning-tree portfast
!
interface FastEthernet0/2
 switchport access vlan 100
 switchport mode access
 spanning-tree portfast
!
interface FastEthernet0/3
 switchport access vlan 200
 switchport mode access
 spanning-tree portfast
!
interface FastEthernet0/4
 switchport access vlan 200
 switchport mode access
 spanning-tree portfast
!
interface FastEthernet0/5
 switchport access vlan 300
 switchport mode access
 spanning-tree portfast
!
interface FastEthernet0/6
 switchport access vlan 300
 switchport mode access
 spanning-tree portfast
!
interface FastEthernet0/7
 switchport access vlan 10
 switchport mode access
 spanning-tree portfast
!
interface FastEthernet0/8
```

```
 switchport access vlan 10
 switchport mode access
 spanning-tree portfast
!
interface FastEthernet0/9
 switchport access vlan 20
 switchport mode access
 spanning-tree portfast
!
interface FastEthernet0/10
 switchport access vlan 20
 switchport mode access
 spanning-tree portfast
!
interface FastEthernet0/11
 switchport access vlan 30
 switchport mode access
 spanning-tree portfast
!
interface FastEthernet0/12
 switchport access vlan 30
 switchport mode access
 spanning-tree portfast
!
interface FastEthernet0/13
 switchport access vlan 40
 switchport mode access
 spanning-tree portfast
!
interface FastEthernet0/14
 switchport access vlan 40
 switchport mode access
 spanning-tree portfast
!
interface FastEthernet0/15
 switchport access vlan 50
 switchport mode access
 spanning-tree portfast
!
interface FastEthernet0/16
 switchport access vlan 50
 switchport mode access
 spanning-tree portfast
```

```
!
interface FastEthernet0/17
 switchport trunk encapsulation dot1q
 switchport mode trunk
!
interface FastEthernet0/18
 switchport trunk encapsulation dot1q
 switchport mode trunk
!
interface FastEthernet0/19
 switchport trunk encapsulation dot1q
 switchport mode trunk
!
interface FastEthernet0/20
 switchport trunk encapsulation dot1q
 switchport mode trunk
!
interface FastEthernet0/21
 switchport trunk encapsulation dot1q
 switchport mode trunk
!
interface FastEthernet0/22
 switchport trunk encapsulation dot1q
 switchport mode trunk
!
interface FastEthernet0/23
 switchport trunk encapsulation dot1q
 switchport mode trunk
!
interface FastEthernet0/24
 switchport trunk encapsulation dot1q
 switchport mode trunk
!
interface GigabitEthernet0/1
 description UPLINK VERSO ALTRO CATALYST
 switchport trunk encapsulation dot1q
 switchport mode trunk
!
interface GigabitEthernet0/2
!
interface Vlan1
 no ip address
!
```

```
interface Vlan10
 ip address 192.168.1.253 255.255.255.0
!
interface Vlan20
 ip address 192.168.2.253 255.255.255.0
!
interface Vlan30
 ip address 192.168.3.253 255.255.255.0
!
interface Vlan40
 ip address 192.168.4.253 255.255.255.0
!
interface Vlan70
 no ip address
!
interface Vlan80
 no ip address
!
ip default-gateway 192.168.1.254
ip classless
ip route 0.0.0.0 0.0.0.0 192.168.1.254
ip http server
!
!
control-plane
!
!
line con 0
line vty 0 4
 password 7 094F471A1A0A
 login local
line vty 5 15
 login
!
end
```

6.2 Dorsale Principale

6.2.1 Cisco 1841

```
!
version 12.3
```

```
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
hostname gateway_1841
!
boot-start-marker
boot-end-marker
!
enable secret 5 $1$lLSw$dl1fvu7mr4WOv0sb7XPEd/
!
no aaa new-model
!
resource policy
!
clock timezone Rome 2
mmi polling-interval 60
no mmi auto-configure
no mmi pvc
mmi snmp-timeout 180
ip subnet-zero
ip cef
!
!
no ip dhcp use vrf connected
!
!
no ip ips deny-action ips-interface
!
frame-relay switching
!
!
!
!
username admin password 7 094F471A1A0A
username framerelay password 7 030752180500
!
!
!
!
!
!
interface Loopback0
```

```
 ip address 192.168.0.1 255.255.255.255
!
interface FastEthernet0/0
 ip address dhcp
 ip nat outside
 ip virtual-reassembly
 duplex auto
 speed auto
!
interface FastEthernet0/0.10
!
interface FastEthernet0/1
 no ip address
 duplex auto
 speed auto
!
interface FastEthernet0/1.10
 encapsulation dot1Q 10
 ip address 192.168.1.254 255.255.255.0
 ip nat inside
 ip virtual-reassembly
 no snmp trap link-status
 mpls ip
!
interface FastEthernet0/1.40
 encapsulation dot1Q 40
 ip address 192.168.4.254 255.255.255.0
 ip nat inside
 ip virtual-reassembly
 no snmp trap link-status
 mpls ip
!
interface Serial0/0/0
 bandwidth 2000
 no ip address
 encapsulation frame-relay
 clock rate 2000000
 frame-relay lmi-type cisco
 frame-relay intf-type dce
!
interface Serial0/0/0.60 point-to-point
 ip address 192.168.6.2 255.255.255.252
 ip nat inside
```

```
 ip virtual-reassembly
 frame-relay interface-dlci 60
!
interface Serial0/1/0
 bandwidth 2000
 no ip address
 encapsulation frame-relay
 no snmp trap link-status
 clock rate 2000000
 frame-relay lmi-type ansi
 frame-relay intf-type dce
!
interface Serial0/1/0.100 point-to-point
 frame-relay interface-dlci 100 ppp Virtual-Template1
!
interface Virtual-Template1
 ip unnumbered FastEthernet0/1.10
 ip nat inside
 ip virtual-reassembly
 peer default ip address dhcp
 ppp authentication chap pap
 ppp chap hostname gateway
 ppp chap password 7 045802150C2E
 ppp pap sent-username gateway password 7 121A0C041104
!
router eigrp 1
 redistribute ospf 1 metric 1000 100 250 1 1500
 network 192.168.6.0 0.0.0.3
 no auto-summary
!
router ospf 1
 log-adjacency-changes
 redistribute eigrp 1 subnets
 network 192.168.0.1 0.0.0.0 area 0
 network 192.168.1.0 0.0.0.255 area 0
 network 192.168.4.0 0.0.0.255 area 0
!
router bgp 100
 no synchronization
 bgp log-neighbor-changes
 neighbor 192.168.0.2 remote-as 100
 neighbor 192.168.0.2 update-source Loopback0
 neighbor 192.168.0.3 remote-as 100
```

```
 neighbor 192.168.0.3 update-source Loopback0
 neighbor 192.168.0.4 remote-as 100
 neighbor 192.168.0.4 update-source Loopback0
 no auto-summary
 !
 address-family vpnv4
 neighbor 192.168.0.2 activate
 neighbor 192.168.0.2 send-community both
 neighbor 192.168.0.3 activate
 neighbor 192.168.0.3 send-community both
 neighbor 192.168.0.4 activate
 neighbor 192.168.0.4 send-community both
 exit-address-family
!
ip classless
!
!
ip http server
no ip http secure-server
ip nat inside source list 1 interface FastEthernet0/0 overload
!
access-list 1 permit 192.168.0.0 0.0.255.255
access-list 1 permit 10.0.0.0 0.0.255.255
!
!
!
!
control-plane
!
!
!
!
banner login ^] Cisco 1841 ^]
banner prompt-timeout ^] login timeout
!
line con 0
 login local
line aux 0
line vty 0 4
 login local
!
scheduler allocate 20000 1000
ntp clock-period 17179261
```

```
ntp server 193.204.114.232
end
```

6.2.2 Cisco 3847

6.2.3 Cisco 3640

6.2.4 Cisco 2611

6.2.5 Cisco AXP

6.3 Dorsale Secondaria

6.3.1 Cisco 1601

```
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
hostname c1600
!
boot-start-marker
boot-end-marker
!
enable secret 5 $1$G2Ok$F.s13nMN8k3wdv/Lg46Th.
!
clock timezone Rome 2
no aaa new-model
ip subnet-zero
ip name-server 130.251.1.4
!
username admin password 7 02050D480809
!
!
!
!
interface Ethernet0
 ip address 192.168.5.254 255.255.255.0
!
interface Serial0
 no ip address
```

```
 encapsulation frame-relay
 frame-relay lmi-type cisco
!
interface Serial0.60 point-to-point
 ip address 192.168.6.1 255.255.255.252
 frame-relay interface-dlci 60
!
interface BRI0
 no ip address
 encapsulation hdlc
 shutdown
!
router eigrp 1
 network 192.168.5.0
 network 192.168.6.0 0.0.0.3
 no auto-summary
!
ip default-gateway 192.168.6.2
ip classless
ip route 0.0.0.0 0.0.0.0 192.168.6.2
ip http server
!
!
line con 0
 login local
line vty 0 4
 login local
!
end
```

6.4 VPN Cliente 1

6.4.1 Cisco 4000

6.4.2 Nokia IP 120

6.5 Rete Cliente 2

6.5.1 Cisco 1711

6.6 VPN Cliente 3

6.6.1 Cisco 800

6.6.2 Cisco 2612