

Lezione 9: Livello 3 - Quagga e Routing dinamico con OSPF

Claudio Ardagna, Patrizio Tufarolo – Università degli Studi di Milano

Insegnamento di Laboratorio di Reti di Calcolatori



Introduzione - 1

- ▶ Nell'ultima lezione abbiamo approfondito il routing statico, andando a compilare le tabelle di routing degli host manualmente
- ▶ In ambienti di produzione, tuttavia, una pratica simile è fortemente sconsigliata
 - ▶ Poca dinamicità
 - ▶ Rischio di causare malfunzionamenti sulla rete
 - ▶ Difficoltà di gestione di molte tabelle di routing in topologie di rete complesse



Introduzione - 2

- ▶ Occorre trovare degli algoritmi e dei metodi di comunicazione per gestire in automatico
 - ▶ Tabelle di routing distribuite su più router
 - ▶ Situazioni di malfunzionamento dei link
 - ▶ Situazioni di sovraccarico dei link e auto bilanciamento della rete
- ▶ Passiamo quindi dal routing statico al routing dinamico
- ▶ Abbiamo, in questo caso, due tipi di protocolli:
 - ▶ Protocolli Distance Vector (come RIP) – basati sull'algoritmo di Bellman-Ford, e che usano la distanza inter-nodo per stabilire una metrica
 - ▶ Protocolli Link-State (come IS-IS o OSPF) – basati su una conoscenza **globale** della topologia di rete. Tutti i costi dei collegamenti sono noti ai router, l'algoritmo generalmente utilizzato è l'algoritmo di Dijkstra.



Terminologia - 1

- ▶ ISO/OSI (Open System Interconnection)
 - ▶ Standard de iure che organizza l'architettura di una rete di calcolatori in una struttura composta da 7 livelli (stack di rete)
- ▶ IPv4
 - ▶ Internetworking protocol, protocollo per interconnettere reti di tipologie eterogenee, versione 4
- ▶ Indirizzo IP
 - ▶ Identificatore numerico univoco a 4 byte (32 bit) suddiviso in *net id o prefisso* e un *host id*
- ▶ Subnet (Sottorete)
 - ▶ **Partizionamento** dello spazio di indirizzamento in sottoreti, realizzato tramite l'applicazione di una **subnet mask**

Terminologia - 2

- ▶ Router (instradatore)
 - ▶ Dispositivi per l'interconnessione di reti eterogenee (o sottoreti differenti), che stabilisce un percorso logico di comunicazione, instradando i pacchetti
- ▶ Hop
 - ▶ Router che un pacchetto deve attraversare per arrivare a destinazione (salti)
- ▶ Tabella di routing
 - ▶ Database memorizzato in un router, che contiene le rotte di destinazione di una rete. Permette di memorizzare informazioni sulla topologia di rete.

Terminologia – 3

- ▶ RIP (Routing Information Protocol)
 - ▶ Protocollo daemon-routed implementato in molti sistemi UNIX, di tipo Distance Vector (calcola la metrica in base al numero di HOP)
- ▶ IS-IS (Intermediate System to Intermediate System)
 - ▶ Protocollo standardizzato RFC di tipo link-state (RFC 1142)
 - ▶ Standard de-facto per il routing sulle backbone degli ISP
 - ▶ Come OSPF usa l'algoritmo di Dijkstra per il calcolo del percorso ottimo
- ▶ OSPF (Open Shortest Path First)
 - ▶ Protocollo di tipo Link-State (RFC2328 per IPv4, RFC5340 per IPV6)
 - ▶ Standard de-facto per il routing all'interno di reti Enterprise

RIP – Ripasso dalla teoria

- ▶ Sviluppato nel 1988 come parte di ARPANET
- ▶ Usa l'algoritmo di Bellman Ford
- ▶ Lavora a livello applicativo, ed è basato su UDP (porta 520).
 - ▶ Anche se è effettivamente un protocollo che opera a livello applicativo, è comunemente trattato come protocollo L3
- ▶ Tre versioni
 - ▶ RIPv1 (RFC 1058)
 - ▶ usa il routing classful (gli aggiornamenti alla tabella di routing non contengono la subnet mask, rendendo impossibile il VLSM)
 - ▶ non implementa autenticazione
 - ▶ RIPv2 (RFC 2453)
 - ▶ Implementato il trasporto delle informazioni relative alla subnet mask, supportando il CIDR
 - ▶ Implementa autenticazione basata su testo in chiaro o hashing MD5
 - ▶ RIP-ng (RFC 2080)
 - ▶ Supporta IPv6, ma lavora sulla porta 521 per non creare conflitto con RIP per IPv4



RIP – Problemi e soluzioni

- ▶ Il funzionamento di RIP si basa sullo stato istantaneo della topologia di rete
- ▶ Il costo di un cammino è calcolato di volta in volta dai router, che conoscono il costo per i possibili hop successivi
- ▶ Nel caso si interrompa un link, si presenta il problema del count-to-infinity
 - ▶ Soluzione parziale: sono necessarie delle tecniche come split-horizon e poison-reverse per evitare di ripubblicare le rotte attraverso le stesse interfacce da cui sono state ricevute
 - ▶ Soluzione reale: instradamento di tipo link-state!

OSPF – Ripasso dalla teoria

- ▶ Protocollo di instradamento di tipo Link-state
- ▶ È il protocollo più usato in reti aziendali molto grandi, o di autonomous system
- ▶ Caratteristiche
 - ▶ Molto più veloce di RIP
 - ▶ Minimo traffico di protocollo
 - ▶ Supporta il VLSM
 - ▶ Ogni router ha le stesse informazioni sullo stato delle connessioni, e calcola autonomamente le informazioni da inserire nella tabella di routing, applicando l'algoritmo di Shortest Path Tree.

RIP e OSPF su Linux: Quagga –1

- ▶ RIP e OSPF sono implementati su Linux tramite il software Quagga (<http://www.nongnu.org/quagga/>)
- ▶ Quagga è un fork di GNU Zebra e consiste di un demone core e di più demoni satellite che implementano vari protocolli di routing. Questi sono
 - ▶ Zebra – demone che si interfaccia con il kernel linux, e fornisce la possibilità di configurare le interfacce di rete e le rotte statiche
 - ▶ RIPd, RIPNGd – demoni che implementano RIP (IPv4) e RIPng (IPv6)
 - ▶ OSPFd, OSPF6d – demoni che implementano il protocollo OSPF sia per IPv4 che per IPv6
 - ▶ BGPd – demone che implementa il Border Gateway Protocol, altro protocollo di routing dinamico per la propagazione di rotte inter-Autonomous-System
 - ▶ ISISd – demone che implementa il protocollo IS-IS
- ▶ Quagga utilizza un terminale CISCO-Like

RIP e OSPF su Linux: Quagga – 2

- ▶ I file di configurazione di Quagga sono in `/etc/quagga`.
- ▶ Esiste un file di configurazione generale:
 - ▶ `/etc/quagga/daemons`
che permette di abilitare o disabilitare i vari demoni
- ▶ Ogni demone ha poi un suo file di configurazione chiamato come il demone stesso, seguito da `.conf`
 - ▶ Noi non useremo direttamente questi file di configurazione, ma ci baseremo sulla riga di comando
- ▶ Quagga è installato per default su IMUNES, e i demoni da utilizzare sono personalizzabili facendo doppio click su un host

Gestione del terminale di Quagga

- ▶ Il comando per aprire il terminale di Quagga è
 - ▶ `vttysh`
- ▶ Ci ritroveremo quindi in un ambiente CISCO-Like.
- ▶ I router Cisco sono gestibili con una Command Line Interface leggermente diversa da quella di Linux
- ▶ Bisogna tenere in mente il concetto di macchina a stati
- ▶ La CLI può trovarsi in vari stati, a seconda dello stato (ambiente) in cui ci si trova sono disponibili diversi comandi. L'ambiente è sempre indicato nel prompt. Alcuni comandi permettono di spostarci in vari ambienti
- ▶ La CLI consente di configurare gli indirizzi di rete, come abbiamo fatto utilizzando la suite IP nelle lezioni precedenti, e di amministrare il nostro host come se fosse un router CISCO

Gestione del terminale di Quagga – 2

- ▶ Dopo aver lanciato vtysh ci troviamo in questa situazione

```
hello, this is Quagga (version 0.99.23.1).  
Copyright 1996-2005 Kunihiro Ishiguro, et al.  
  
router1# █
```

- ▶ Questo è l'ambiente di default. È indicato il nostro hostname, e ci viene data la possibilità di dare dei comandi
- ▶ Scrivendo «?» ci viene fornito un riassunto dei comandi disponibili
- ▶ Con il comando «list» ci viene fornito un riassunto dei comandi più utili e comunemente usati
- ▶ Per uscire da un ambiente, si usa il comando «exit» o la combinazione di tasti <CTRL+D>

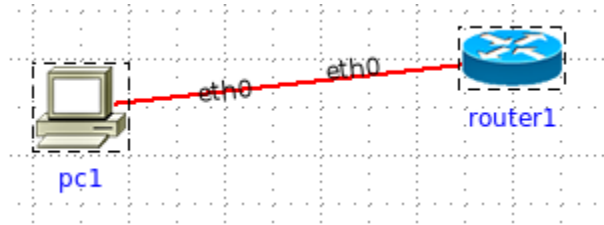
Gestione del terminale di Quagga – 3

- ▶ Postponendo «?» a qualunque comando ci viene fornito un elenco di opzioni possibili per completarlo
- ▶ Il comando «configure» permette di entrare negli ambienti di configurazione, con «configure terminal» entriamo, ad esempio, nell'ambiente di configurazione del terminale
- ▶ Nell'ambiente di configurazione del terminale possiamo configurare tutti gli aspetti del nostro router, come le interfacce di rete o i router
- ▶ Da qualunque ambiente, tramite il comando «do» si possono eseguire comandi dell'ambiente di default
- ▶ In ambiente di configurazione, antepoendo «no» a un comando, si può invertirne l'effetto disabilitando il comportamento

Esercizio di utilizzo di Quagga - IMUNES

- ▶ Dopo aver disabilitato l'autoassegnamento degli indirizzi IP, creare una topologia di rete su Quagga composta da un PC connesso a un router
- ▶ Assegnare al PC un indirizzo IP utilizzando i comandi visti nelle scorse lezioni (ip addr)
- ▶ Assegnare al Router un indirizzo IP appartenente alla stessa sottorete del primo, utilizzando Quagga tramite vtysh
- ▶ Eseguire un ping da PC verso Router, e viceversa

Esercizio di utilizzo di Quagga - IMUNES



Comandi pc1

```
▶ ip addr add \
  10.0.0.2/24 dev \
  eth0
```

Comandi router1

```
router1# conf t
router1(config)# interface eth0
router1(config-if)# ip address
10.0.0.1/24
router1(config-if)# exit
router1(config)# exit
router1# ping 10.0.0.2
```

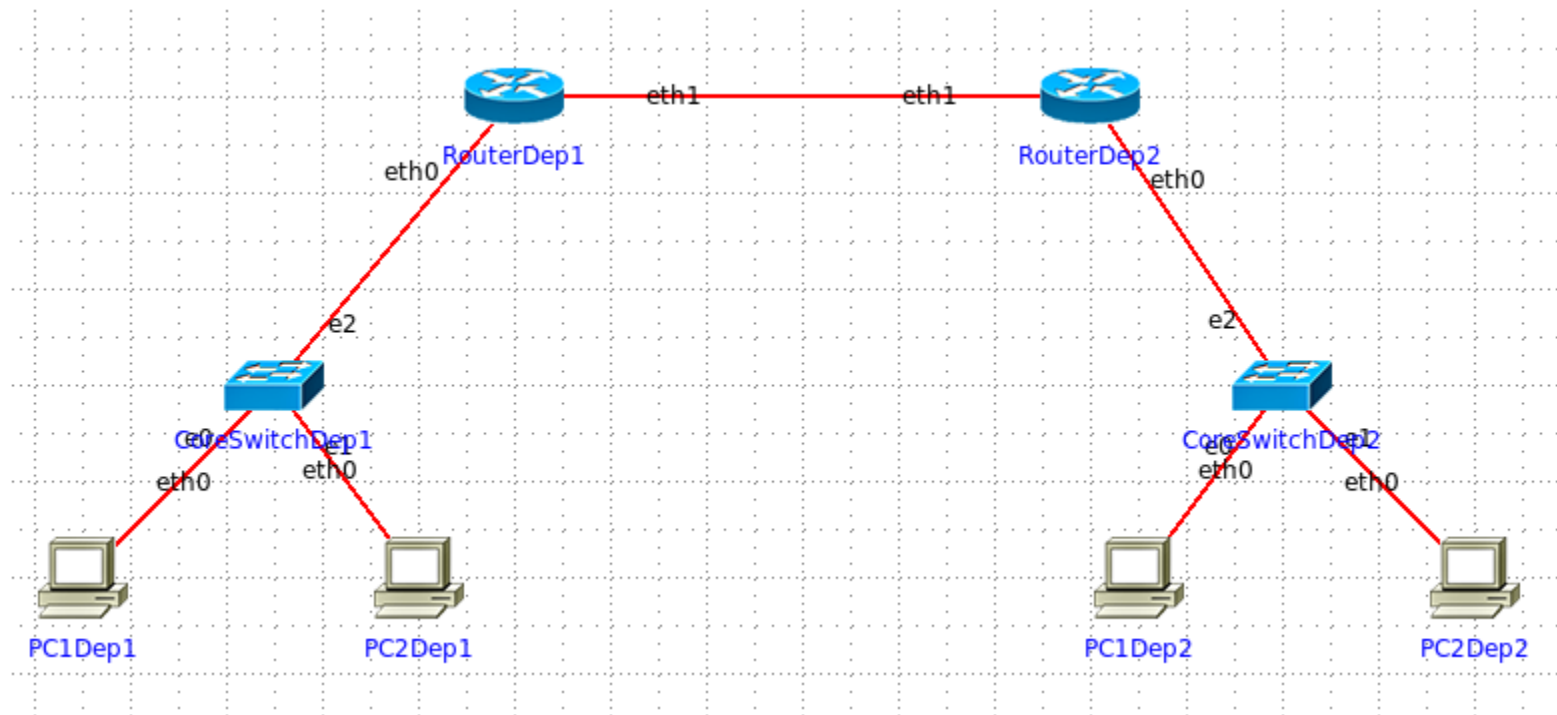

OSPF su Quagga e Cisco

- ▶ OSPF è implementato su Quagga tramite il comando «router» (ambiente di configurazione), che permette di configurare il routing sia statico che dinamico, in tutti i protocolli supportati
- ▶ Per far funzionare OSPF è necessario configurare questo router, dichiarando (tramite il comando «network») le reti che il router conosce, associate a un'area OSPF (numero arbitrario a 32 bit). Useremo l'area di backbone (area 0).
- ▶ OSPF farà il resto

Esercizio routing dinamico su IMUNES

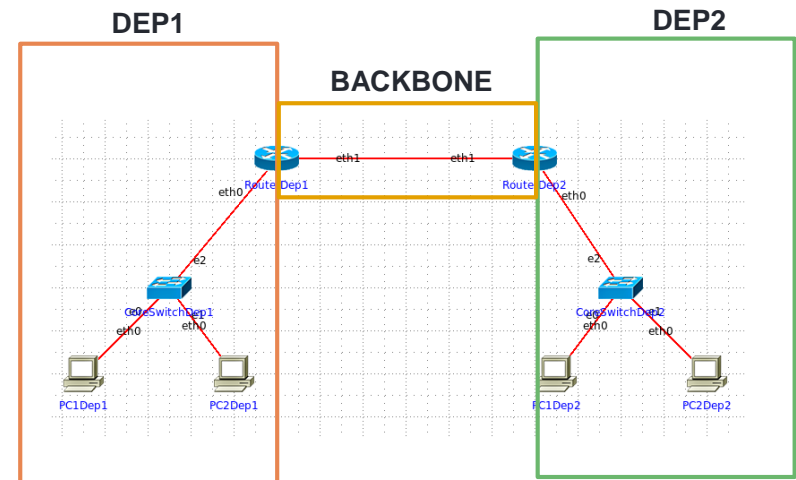
- ▶ Utilizzare la topologia realizzata nella scorsa lezione (esercizio 2 – routing statico)
- ▶ Prima di eseguire l'esperimento fare doppio click sui router per disabilitare tutte le flag per il routing dinamico
- ▶ Dopo aver lanciato l'esperimento (con assegnamento automatico degli indirizzi IP disabilitato), assegnare gli indirizzi IP degli host con i comandi classici, abilitare i demoni Zebra e OSPFd modificando `/etc/quagga/daemons` e riavviare il demone quagga (`service quagga restart`)
- ▶ Assegnare gli indirizzi IP dei router utilizzando Quagga tramite «vtysh»
- ▶ Assegnare ai pc una rotta di default verso il proprio router
- ▶ Dopo aver verificato il funzionamento della rete, abilitare OSPF sui router, dichiarando le reti ad essi conosciute tramite il comando «network» permettendo a PC1 di comunicare con PC2 (verificare con ping e traceroute)
 - ▶ Per ogni rete bisogna dichiarare un'area OSPF (numero a 32 bit), noi utilizzeremo l'area 0 (backbone)

Esercizio routing dinamico su IMUNES



Esercizio 1 routing dinamico su IMUNES – Soluzione - 1

- ▶ 3 Reti
 - ▶ Rete CoreSwitchDep1 composta da
 - ▶ RouterDep1
 - ▶ Pc1Dep1
 - ▶ Pc2Dep1[IP NELLA SOTTORETE 192.168.0.0/24]
 - ▶ Rete CoreSwitchDep2 composta da
 - ▶ RouterDep2
 - ▶ Pc1Dep2
 - ▶ Pc2Dep2[IP NELLA SOTTORETE 192.168.1.0/24]
 - ▶ Rete di backbone composta da
 - ▶ RouterDep1
 - ▶ RouterDep2[IP NELLA SOTTORETE [192.168.2.0/24]



Esercizio 1 routing dinamico su IMUNES – Soluzione - 2

Comandi PC1Dep1

- ▶ `ip addr add \`
`192.168.0.2/24 dev eth0`
- ▶ `Ip route add 0.0.0.0/0 \`
`via 192.168.0.1/24 dev \`
`eth0`

Comandi PC2Dep1

- ▶ `ip addr add \`
`192.168.0.3/24 dev eth0`
- ▶ `ip route add 0.0.0.0/0 \`
`via 192.168.0.1/24 dev \`
`eth0`

Comandi PC1Dep2 e PC2Dep2

- ▶ `ip addr add \`
`192.168.1.2/24 dev eth0`
- ▶ `ip route add 0.0.0.0/0 \`
`via 192.168.1.1/24 dev \`
`eth0`

Comandi PC1Dep2 e PC2Dep2

- ▶ `ip addr add \`
`192.168.1.3/24 dev eth0`
- ▶ `ip route add 0.0.0.0/0 \`
`via 192.168.1.1/24 dev \`
`eth0`



Esercizio 1 routing dinamico su IMUNES – Soluzione - 2

Comandi RouterDep1

- ▶ `sed -i "s/zebra=no/zebra=yes/g" /etc/quagga/daemons`
- ▶ `sed -i "s/ospfd=no/ospfd=yes/g" /etc/quagga/daemons`
- ▶ `service quagga restart`
- ▶ `vttysh`
`RouterDep1# configure terminal`
`RouterDep1(configure)# interface eth0`
`RouterDep1(configure-if)# ip address 192.168.0.1/24`
`RouterDep1(configure-if)# exit`
`RouterDep1(configure)# interface eth1`
`RouterDep1(configure-if)# ip address 192.168.2.1/24`
`RouterDep1(configure-if)# exit`
`RouterDep1(configure)# router ospf`
`RouterDep1(configure-router)# network 192.168.0.0/24 area 0`
`RouterDep1(configure-router)# network 192.168.2.0/24 area 0`
`RouterDep1(configure-router)# exit`
`RouterDep1(configure)# exit`
`RouterDep1# exit`

Esercizio 1 routing dinamico su IMUNES – Soluzione - 3

Comandi RouterDep2

- ▶ `sed -i "s/zebra=no/zebra=yes/g" /etc/quagga/daemons`
- ▶ `sed -i "s/ospfd=no/ospfd=yes/g" /etc/quagga/daemons`
- ▶ `service quagga restart`
- ▶ `vttysh`
RouterDep2# `configure terminal`
RouterDep2(configure)# `interface eth0`
RouterDep2(configure-if)# `ip address 192.168.1.1/24`
RouterDep2(configure-if)# `exit`
RouterDep2(configure)# `interface eth1`
RouterDep2(configure-if)# `ip address 192.168.2.2/24`
RouterDep2(configure-if)# `exit`
RouterDep2(configure)# `router ospf`
RouterDep2(configure-router)# `network 192.168.1.0/24 area 0`
RouterDep2(configure-router)# `network 192.168.2.0/24 area 0`
RouterDep2(configure-router)# `exit`
RouterDep2(configure)# `exit`
RouterDep2# `exit`

Esercizio 2 routing dinamico su IMUNES

- ▶ Disegnare una topologia di rete composta da cinque router, interconnessi in modo circolare.
- ▶ Collegare a due router non adiacenti due computer
- ▶ Prima di eseguire l'esperimento fare doppio click sui router per disabilitare tutte le flag per il routing dinamico
- ▶ Dopo aver lanciato l'esperimento (con assegnamento automatico degli indirizzi IP disabilitato), assegnare gli indirizzi IP degli host con i comandi classici, abilitare il demone OSPF e riavviare quagga.
- ▶ Assegnare gli indirizzi IP dei router utilizzando Quagga e configurare OSPF
- ▶ Assegnare ai pc una rotta di default verso il proprio router
- ▶ Osservare il comportamento con traceroute
- ▶ Spegnerne un'interfaccia del router intermedio nel percorso mostrato da traceroute, interrompendo il percorso, attendere mezzo minuto e tentare nuovamente un traceroute

Esercizio 2 routing dinamico su IMUNES

