MPLS: per trasportare IP su una rete pubblica

IP è un protocollo BEST-EFFORT: non fornisce alcuna garanzia su avvenuta ricezione, ritardo ricezione, etc  
l’operatore vuole vendere una certa infrastruttura.

Immagine che contiene testo, schermata, cerchio, Elementi grafici

Descrizione generata automaticamenteScenario tipico per operatore: cipolla con all’esterno IP e all’internissimo optical switching

Negli anni 90 per l’operatore la rete era a cipolla in quanto aveva sempre trasmesso basandosi su SDH e ATM.

Richiami anni 90:

Commutazione a circuito: operatore crea un circuito tra A e B per farli comunicare  
 🡪 circuito virtuale per riservare delle risorse di banda, quando finisce chiamata elimino circuito  
 🡪 con sonet/sdh

ATM: protocollo per organizzare la rete in un modo efficiente e preciso ma estremamente complesso

È nato IP:   
 🡪questa infrastruttura, hanno aggiunto un layer aggiuntivo per trasportare IP su questa infrastruttura

Commutazione a pacchetto: ogni dispositivo riceve pacchetto e fa una decisione relativa a quel pacchetto 🡪 non c’è concetto di sessione o altro, router prende decisione relativa a quel singolo pacchetto.

Immagine che contiene testo, schermata, Carattere, grafica

Descrizione generata automaticamentefine anni ‘90

MPLS: rimuove il bisogno di avere tutti questi strati della cipolla che servivano per interoperabilità

* Semplifica di molto e permette di trasportare ip su un’infrastruttura esistente sbarazzandosi dei vari protocolli
* Permette di offrire delle garanzie fornendo ottimi servizi
* Miglioramento risorse:
  + COSA SUCCEDEVA PRIMA: Router ip: fa longest prefix match facendo look-up sulla tabella usanso ip destinazione
    - Troppo costoso
    - Invece di fare look-up di una tabella usando ip-destinazione sarebbe più comodo usare un’etichetta 🡪 label 🡪 molto più piccola 🡪 look-up più veloce
  + **ORA** In MPLS si definisce **una label di 20 bit (integer**) 🡪 questo indice e anche l’indice della mia tabella 🡪 look-up velocissimo
    - **Label associabile** a cosa voglio 🡪 posso associarla ad indirizzi IP, al percorso tra un host ad un altro host
      * COME ERA PRIMA: Basandomi solo su ip, la scelta dei percorsi non è ottima in quanto ogni decisione si basa su ip destinazione 🡪 se riesco a scorporare la decisione su come mandare i pacchetti da quello che è l’ip destinazione, posso migliorare indirizzamento e percorso.
      * ORA: posso aggiungere alla rete IP che è connectionless delle caratteristiche connection-oriented:
        + Immagine che contiene testo, schermata, cerchio, diagramma

          Descrizione generata automaticamenteHo una rete basata su MPLS

Ho una nuvoletta MPLS: che crea percorsi tra Ingress LSR e Egress LSR 🡪 LSPtunnel

Nuvoletta ip che circonda MPLS

Label switch router: dispositivo che prende la label e fa switching, la cambia.

Ho bisogno di dispositivi al bordo della rete:

Ingress LSR: aggiungono label per usare su mpls

Egress LSR: rimuovono la label in modo tale da tornare su IP

Immagine che contiene schermata, testo, mappa, diagramma

Descrizione generata automaticamenteImmagine che contiene schermata, testo, diagramma

Descrizione generata automaticamenteImmagine che contiene schermata, mappa, testo, diagramma

Descrizione generata automaticamente

ELEMENTI CHIAVE DI MPS

* Header 🡪 contiene etichetta
* Protocollo di distribuzione per etichette (non è possibile usare i protocolli classici di routing in quanto sono basati su ip destinazione).
* Protocolli di routing vengono usati per capire dove si trovano i router esterni.

Immagine che contiene Elementi grafici, Carattere, Blu elettrico, calligrafia

Descrizione generata automaticamenteSTORIA DI MPLS:  
 VECCHIO CON ATM: blackbox router ip non sanno cosa accade in atm e viceversa

X chiede ad R di creare una strada verso Y

R quindi si collega ad S e chiede di collegarsi a Y

Y collega a D 🡪 chi manda deve già sapere di Y

Immagine che contiene calligrafia, Elementi grafici, cerchio, Carattere

Descrizione generata automaticamenteNUOVO CON MPLS: router IP e MPLS usano lo stesso control plane e si scambiano dei messaggi

Quando X deve mandare il messaggio a D, contatta R che conosce cosa è D

* Chi manda deve solo sapere nextHop

MPLS consente di usare router ATM, mi libero di aTM per tutto quello che riguarda il controlPlane e li uso solo per fare forwarding 🡪 non gli faccio più creare circuiti e cose varie (contorPlane) ma si limita a ricevere l’etichetta e inviare il pacchetto al nextHop corretto.

Inoltre con MPLS posso anche usare gli Switch ottici (gli switch ottici dovevano mettersi d’accordo sulle frequenze), sfritto il loro protocollo di signaling per mettermi d’accordo sul colore delle etichette.

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamenteMPLS HEADER

Tra il livello 2 e il livello 3 ci aggiungo l’header MPLS

Exp 🡪Class of Service simile a quality, mi permette di dividere il traffico in classi in modo tale da poter configurare delle gestioni specifiche

S: bottom of stack 🡪 ogni pacchetto può avere al suo interno più etichette 🡪push di etichette in ingresso, pop di etichette per uscire. L’ultimo (primo inserito) avrà valore bottomOfStack=1 🡪 in modo tale che se è l’ultimo, tolgo tutto l’header MPLS 🡪 e leggo ip destinazione.

La label MPLS la vado a mettere dentro il campo VCI/VPI (ATM) in modo tale da poter usare gli switch ATM per fare forwarding.

CONTROL PLANE:

FEC: Forwaring Equivalence Class

Tutti i pacchetti appartenenti alla stessa FEC vengono trattati allo stesso modo dai router 🡪 fanno lo stesso percorso

* Dunque le etichette vengono stabilite tramite FEC
  + Se voglio posso fare FEC in base ad IP destinazione
  + Se voglio posso fare FEC su IP destinazione + IP sorgente
  + Se voglio posso fare FEC su IPdestinazione+ switch da cui ho ricevuto
  + Se voglio posso fare FEC su IPdestinazione+ porta destinazione

Nota: se ho una coppia sorgente-destinazione, posso fare una FEC solo per loro

SCELTA COLORE ETICHETTA

* LABEL BINDING: LSR decide etichetta che deve essere associata ad una determinata FEC
  + DOWNSTREAM BINDING: etichetta scelta dal router a valle (del link)
    - *Es: Y(router MPLS) comunica che tutto il traffico per D(destinazione) deve essere rosso e quindi se X deve mandare qualcosa per D lo coloro di rosso*
    - Pacchetti dello stesso FEC ricevono stessa etichetta
    - Nodo a monte del link viene informato
  + Due tipi di modalità
    - Unsolicited: Y scopre una nuova destinazione e quindi fa il binding dell’etichetta
    - On-demand: es: quando Y chiede di mandare pacchetto a D
* LABEL MAPPING
  + Associazione tra:
    - Etichetta in input 🡪 decisa da me (colui che ha ricevuto in quel momento)
    - Etichetta in output 🡪 decisa dal nextHop
    - NextHop 🡪 deciso dai protocolli di routing
* Label DISTRIBUTION:
  + Immagine che contiene schermata, Elementi grafici, arte

    Descrizione generata automaticamenteNotifica dell’etichetta scelta per un determinato FEC
  + Un LSR che ha deciso l’etichetta, la comunica ai vicini (etichetta univoca su quel link)
  + Etichetta univoca sul link, io posso dire ad X che per raggiungere D deve mandarmi un pacchetto rosso, mentre a W posso dirgli che lo voglio nero

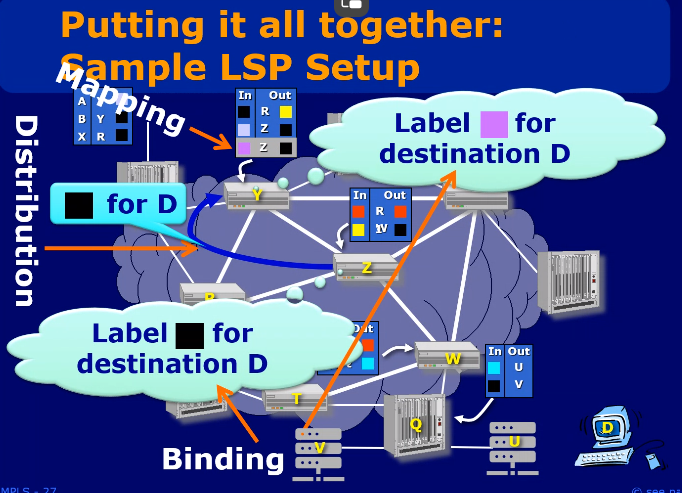


Immagine che contiene testo, schermata, Carattere, Blu elettrico

Descrizione generata automaticamenteMAPPING :

DYNAMIC LABEL BINDING:

* Data/Traffici driven
  + Scatenati dai pacchetti dato
* Control driven
  + Scatenati dai messaggi di controllo
    - Signaling
    - Routing 🡪 presenza nuovo link, cambiamenti topologia
  + Modalità:
    - Topology based: nuovo messaggio dai protocolli di rouuting 🡪 faccio binding
      * unsolicited
    - Explicit: richiesta esplicita di fare binding
      * On demand
      * Solitamente inviati da edge router
      * Meccanismo molto più simile ai protocolli connection-oriented:
        + X riceve un pacchetto e crea un percorso dove riservo delle etichette per quel flusso (solo etichette, non banda o altro)

LABEL DITRIBUTION PROTOCOL

* Routing protocol: BGP
  + Basato su topologia
* Label distribution protocol (LDP)
  + Disegnato specificatamente da Cisco per questo motivo
* Resource reservation protocol (RSVP)
  + Protocollo standard
  + Molto usato per quality of service

PRotoccollo di routing usati per decidere LSP (mapping etichette e quindi anche routing)

Protocolli esistenti:

* OSPF
* IS-IS
* BGP-4

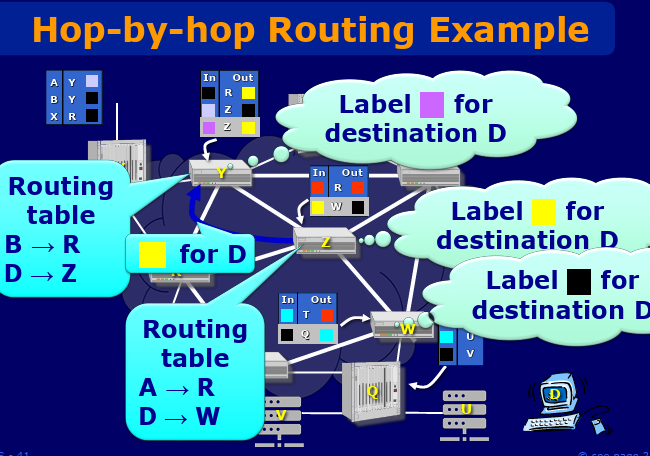
Possono trasportare informazioni ulteriori come quelle sulla capacità del link, informazioni sull’uso del link, informazioni riguardo le dipendenze tra i link (es: uso per fault recovering (*se ad esempio ho un guasto su un router alimentato da un certo generatore e il generatore mi si rompe, non posso usare come rotta di backup una in cui c’è un router alimentato dallon stesso geenratore*))

Protocolli potenziati per fare Traffic Enginering in maniera più efficiente:

* OSPF-TE
* IS-IS-TE

Immagine che contiene testo, schermata, diagramma, mappa

Descrizione generata automaticamenteROUTING MODES🡪 slide 36 37 38



Usando la **modalità HOP-By-Hop** diventa complesso fare traffic enginering perché si hanno le limitazioni presenti anche in IP

**EXPLICIT** ROUTING 🡪 mi permette di visualizzare anche i constraint 🡪 c’è un singolo switch che decide il percorso 🡪 modalità molto simile al source & routing

Si riesce a fare Constraint Based Routing:

* C’è bisogno di uno che faccia le scelte 🡪**routing esplicito** con un nodo che decide per tutti
  + *Nota: non è possibile usare DJIKSTRA in quanto non gestisce i costraint*
  + Difficile mantenere informazioni sui constraint sincronizzata
* Label distribution protocols deve essere modificato per fare Constraint based routing
  + CR-LDP 🡪ConstraintBased Routing LDP
  + RSVP-TE
* Mi garantisce nuove possibilità
  + Traffic engingering
  + Garanzie per QualityOfService (si dovrebbero riservare le risorse ma ha un costo troppo elevato
  + Permette traffic engenering per classi
  + Fast fault recovery in meno di 50ms

TRAFFIC ENGINEERING:

Immagine che contiene schermata, testo, grafica, diagramma

Descrizione generata automaticamenteL’ideale sarebbe usare in maniera abbastanza bilanciata i miei canali, in IP quando si fa routing, ogni router sceglie il percorso con shortest path, quindi tutto il traffico verso un nodo da parte di un router, fa sempre quel percorso 🡪 si rischia congestione su alcune linee e altre linee inutilizzate

Una soluzione era quella di ridirezionare il traffico quando vedevo congestione andando a cambiare le tabelle di routing 🡪 di conseguenza rimbalzava la situazione di congestione da un percorso all’altro 🡪 continua instabilità 🡪 non risolvo una minchia.

Nota: era fattibile con SDN ma era molto costoso

MPLS molto più intelligente 🡪 usa routing table solo per scegliere LSP 🡪 non intacca forwarding table

Immagine che contiene schermata, Policromia

Descrizione generata automaticamente

Cosa succede in MPLS:

* Creo vari LSP
* Se cambia il carico potrebbe variare la routing table ma non la forwarding table, quindi continuo a mandare come stavo mandando e cambio in determinati casi: (vedi prima)

QoS rimandata a data da destinarsi

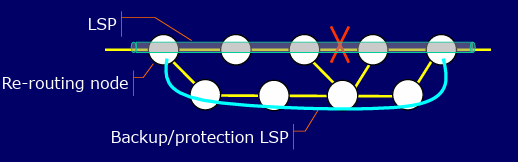
Immagine che contiene schermata, linea, diagramma, cerchio

Descrizione generata automaticamenteFAST FAULT RECOVERY:

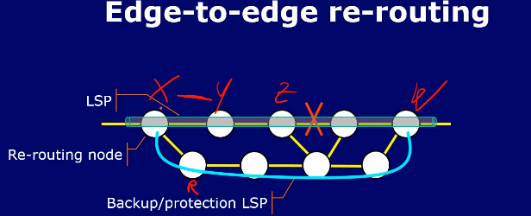
* LINK RE-ROUTING:
  + LSP creato a priori
  + LSP nuovo in azzurrino che fa protezione sul link dove c’è la X in modo tale da garantire il collegamento tra i due nodi anche in caso di guasti, nel caso in cui ci sia un guasto, MPLS se ne accorge e fa un’operazione di RE-routing andando ad utilizzare LSP azzurro

Immagine che contiene testo, schermata, diagramma, Carattere

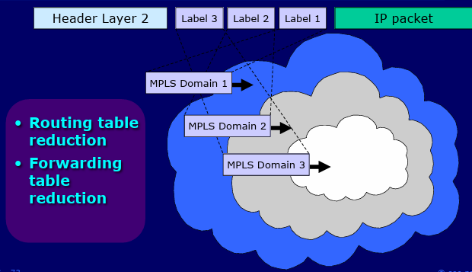
Descrizione generata automaticamenteX-Y rotto  
Y aveva chiesto ad X etichetta rossa  
Z aveva chiesto ad X etichetta verde  
X deve mandare ad Y  
c’è un guasto  
capisce che non può mandarlo direttamente ad Y  
è a conoscenza della presenza di Z grazie ad una routing tabel dove sa che il percorso azzurrino porta ad Y  
X quindi pusha etichetta rossa e poi pusha etichetta verde   
Z quindi fa la pop e toglie così l’etichetta verde  
Pacchetto arriva ad Y con la sola etichetta rossa



Altro meccanismo più efficiente: Edge to Edge Re roouting: protezione di un intero percorso e non più di un solo link 🡪 protezione di un intero LSP



Z notifica Y, Y notifica X, x quindi si rende conto che quel percorso non va bene e quindi deve mandarlo ad R e non più ad Y 🡪 label esterna scelta da R, Label più interna quella scelta da X.

Dunque posso avere più label per ogni pacchetto:

* Usata sia per Recovery che nel caso in cui ho più domini MPLS
* Favorisce gerarchia e scalabilità

Se attivo il PHP: **Penultimate Hop Popping** 🡪 non è l’ultimo router ha togliere la label MPLS ma il penultimo in quanto gli ultimi devono fare già un bel lavoro di preprocessing dato che ricevono MPLS e devono trovare dove mandarlo in IP. La forwarding table di Y(ultimo) e basata su IP/MPLS di livello inferiore. Nel protocollo c’è il campo PHP che specifica se si vuole fare PHP o meno.

**Come attivare PHP:**

**IMPLICIT PHP**: Volendo posso rendere automatico questo procedimento dove il penultimo toglie la label MPLS (che altrimenti non è possibile dato che non sa di essere il penultimo) 🡪 **l’egress LSR mette come valore di label 3**, in questo modo se un router riceve un pacchetto e deve inviarlo a label3, capisce che deve fare PHP Popping

Si può anche fare **Explicit PHP** 🡪 **ultimo dice a penultimo di mettere label 0** 🡪 a questo punto il penultimo non rimuove l’header mpls 🡪 in questo modo **non vengono rimosse alcune informazioni presenti in mpls (es: QoS, Classe)** che rimuovendo la label sarebbero rimosse e quindi arriverebbero senza questa cose all’ultimo.