

ROUTING AND FORWARDING

L'intestazione del pacchetto contiene le informazioni per l'inoltro dello stesso, esistono anche altri metodi.

Routing: trovare una strada per i pacchetti

Forwarding: far avanzare i pacchetti, implica anche fare routing -> **On the fly routing.**

Switching: trasferimento verso la porta di uscita, scrittura dei byte.

Routing proattivo: i nodi svolgono operazioni di routing, scegliendo la best route, cercando di capire come il pacchetto possa raggiungere la destinazione. È un'operazione fatta a priori, prima che il pacchetto arrivi al nodo, quindi indipendente dal traffico in rete.

Il routing viene gestito da una parte del router chiamato **piano di controllo** mentre il forwarding dal **piano dati**.

Tabelle di routing/forwarding: sono il risultato del routing proattivo o segnalazione. La segnalazione consiste nell'informare gli altri nodi sui possibili percorsi che un pacchetto potrà seguire per andare da quel nodo stesso ad una qualsiasi destinazione. L'operazione viene eseguita prima dell'arrivo del pacchetto.

Classificazione Algoritmi on-the-fly routing/forwarding

1. **Routing by network address :** si utilizzano memoria con prefissi. Nel caso di più match per lo stesso prefisso si utilizza il *longest prefix match*.
2. **Label swapping:** si utilizzano tavole tramite l'etichetta del pacchetto, che indica direttamente la riga della tabella -> indirizzamento diretto.
3. **Source routing:** nel pacchetto stesso c'è scritto il destinatario, non si usano tavole, il routing proattivo viene effettuato dagli altri nodi.

L'etichetta non corrisponde con l'indirizzo, sono due cose diverse.

Classificazione algoritmi Routing Proattivo

Statico: algoritmi non adattativi.

Svantaggi

- errori comuni
- Non si adatta a cambiamenti topologici della rete. In caso di guasto si usano le ruote di backup.

Vantaggi

- amministratore ha totale controllo
- comodo alla periferia della rete

1. **Static routing:** configurazione manuale, ogni nodo sa già dove deve spedire i pacchetti per raggiungere le varie destinazioni.
2. **Selective flooding:** il pacchetto viene inviato a tutti i nodi. Non utilizzato nelle reti ip. **Soggetto al problema della duplicazione dei pacchetti.** Diverse tecniche per la gestione della duplicazione(contatori, riconoscimento, spanning tree).

Dinamico: algoritmi adattativi.

1. **Routing centralizzato:** un unico dispositivo che fa routing proattivo per tutti. Richiede e invia info da e a tutti i nodi perché deve decidere i percorsi per tutti.

Vantaggi

- performance ottimizzate
- facile risoluzione errori

Svantaggi

- collo di bottiglia
- singolo punto di errore
- non adatto a reti dinamiche

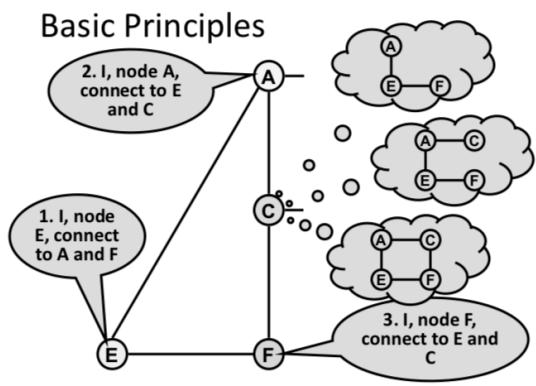
2. **Routing isolato:** ogni dispositivo prende decisioni indipendentemente senza scambio di informazioni. Molto usato nelle reti locali ethernet. Backward learning, non adatto a reti magliate -> spanning tree.

3. **Routing distribuito:** ogni dispositivo prende decisioni indipendentemente ma dopo aver eseguito uno scambio di informazioni con gli altri. È un insieme di routing centralizzato e isolato.

Esempi:

- **Link state**
- **Distance vector**

Link state:



1. Il nodo capisce come è fatta la rete intorno a lui, cioè a chi è collegato tramite OSPF: ogni nodo si presenta ai vicini.
2. Invia a tutti questa info tramite selective flooding
3. Ogni nodo costruisce una mappa della rete
4. Ogni nodo applica Dijkstra, se tutti i nodi hanno la stessa mappa di rete con lo shortest path first le tabelle di routing saranno coerenti

Il vantaggio dell'uso dell'algoritmo di Dijkstra è la sua complessità che è dell'ordine di :

$$L \cdot \log(N)$$

Dove L numero di Link e N numero di nodi.

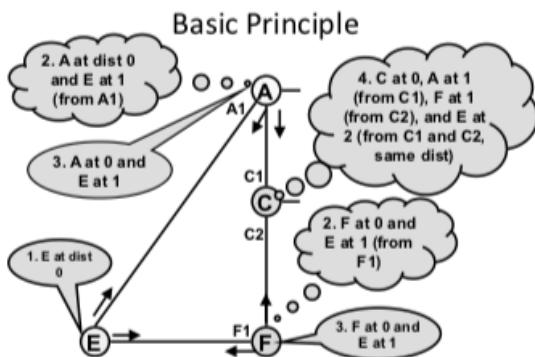
Vantaggi:

- **Converge molto velocemente, velocità con cui si adatta ai cambiamenti topologici della rete**
- **La convergenza di ogni router è indipendente**
- **Meno traffico circolante**
- **Raramente loop di routing e se ci fossero durerebbero molto poco**
- **Semplice rimedio agli errori, ogni nodo ha lo stesso Link State Database.**
- **Ottimo su reti grandi.**

Svantaggi:

- **Alta complessità di realizzazione**
- **Protocolli con configurazioni complesse su reti piccole.**

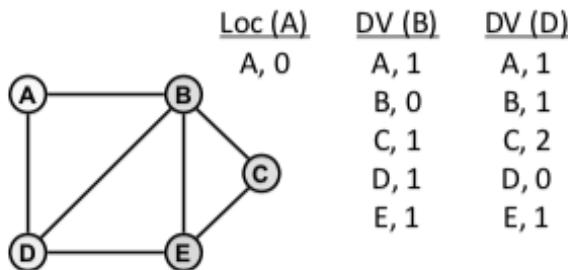
Distance Vector:



1. Ogni nodo manda ai suoi vicini i suoi Link con le rispettive distanze periodicamente. Se distance vector troppo lungo viene mandato a pezzi.
2. Ogni nodo crea la lista di destinazioni

Problema principale DV:

Nell'algoritmo distance vector il problema è che i router non sanno come sia fatta la rete.



Loc (A)	DV (B)	DV (D)	ROUT. TABLE (A)	DV (A)
A, 0	A, 1	A, 1	A, local, 0	A, 0
	B, 0	B, 1	B, A1, 1	B, 1
	C, 1	C, 2	C, A1, 2	C, 2
	D, 1	D, 0	D, A2, 1	D, 1
	E, 1	E, 1	E, A2, 2	E, 2

A -> percorso più veloce restare fermo, locale, costo 0.

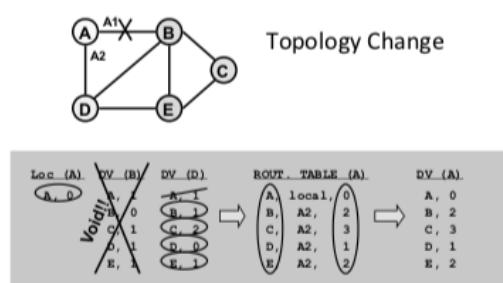
B -> percorso più veloce da B (link diretto), costo 1.

C -> percorso più veloce da B, C dista 1 da B, B dista 1 da A, corso 2.

D -> percorso più veloce da D (link diretto), costo 1.

E -> percorso più veloce da B,D, E dista 1 da B,D che distano 1 da A, costo 2.

In caso di guasto di un Link la lista di destinazioni non è più valida.



In questo caso in nodo A userà solo il DV di D e non anche quello di B per creare il suo DV. Questo perché il link tra A e B si è rotto

Link State e Distant Vector confronto :

LS

- In LS ogni cambiamento si propaga a tutti e poi si ferma.
- Complessità di configurazione.
- Complessità di implementazione
- Non genera quasi mai loop
- LS ottimo su rete grandi.
- troubleshooting facile
- Ogni nodo comunica agli altri i propri collegamenti, in questo modo i nodi riescono a costruirsi una mappa topologica della rete.
- Convergenza veloce

DV

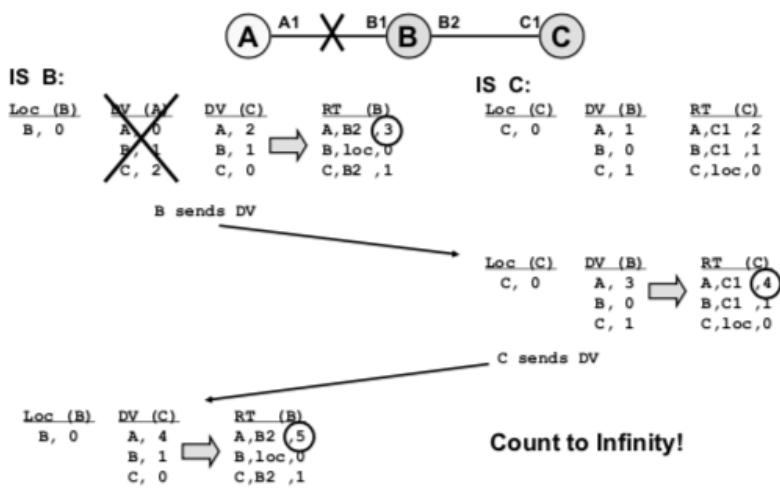
- In DV ogni cambiamento si propaga a tutti ed ogni nodo ripropaga le nuove modifiche ad ogni altro nodo. In questo modo la perdita di un distance vector non pregiudica il funzionamento, verrebbe inviato comunque da un altro nodo.
- DV ottimo su reti piccole.
- Facilità di configurazione.
- Il tempo di convergenza in DV viene determinato da router più lento.
- troubleshooting complesso
- non va bene su reti grandi
- I nodi comunicano agli altri solo le loro distanze (Distance Vector) per questo motivo ogni nodo non sa come sia fatta la rete.

Problemi con Distance Vector:

- **Black holes**
Quando un router non si è ancora calcolato percorso per destinazione e altri router gli stanno mandando pacchetti , il router butta via pacchetti.
Switch non conosce destinazione → flooding
Router non conosce destinazione → scarta pacchetto
- **Count to infinity**
- **Bouncing effect (loop), instabilità**
Ci vuole del tempo prima di ricalcolare il distance vector, se una destinazione viene cancellata, le rimanenti si inviano il pacchetto a vicenda finché non ricalcolano i loro distance vector.

Count to infinity

Count to Infinity



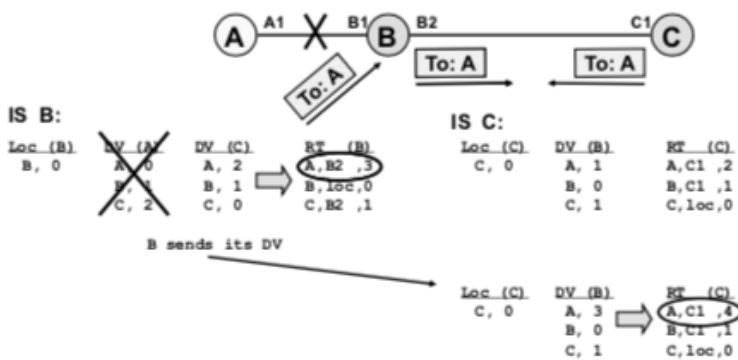
1) Stazione A non si accorgere che non può più utilizzare il DV di A. Utilizza allora il DV C per calcolare la distanza da A e segna quindi A 3.

2) C riceve il DV(B) e vede che la distanza da B ad A è 3 ora. Allora segna la sua distanza da A 4.

3) B riceve il DV(C) e vede la distanza da C ad A è aumentata di 1. Allora aumenta anche la sua.

Bouncing effect

Bouncing Effect



Legato al problema del count to infinity. Le due stazioni B e C continueranno a scambiarsi informazioni su A sbagliate avanti e indietro generano un loop.

Soluzioni ai problemi:

- **Split horizon:** Soluzione a count to infinity.
 - previene loop tra due nodi
 - aumenta la velocità di convergenza
 - distance vector personalizzati, non includi le destinazioni raggiungibili tramite il nodo a cui sto inviando DV.
 - non funziona su topologie magliate (neanche con poisonous reverse)
 - **Path hold down:** quando una destinazione non è più raggiungibile secondo un percorso che prima era agibile, aspetto un certo tempo invece di calcolare subito un nuovo percorso, considerando quelle destinazioni irraggiungibili.
 - Tempo di convergenza alto
 - Il nodo che si accorge della caduta del link non partecipa a nessun loop fino a che non scade il timeOut.
 - **Cost reaction:** se destinazione mi arriva con un costo più alto rispetto al precedente , ignoro quella informazione. Non aspetto che il percorso non sia più valido, mi basta vedere che ci voglia di più per raggiungerlo. Anticipa il Path Hold Down. **Potrebbe però escludere route di cui il costo è aumentato legittimamente. Si è passati quindi a rouotePoisoning.**
 - **Poisonous reverse:** Utilizza *Route Poisoning*, quando una route non è più valida per caduta link o perché costo aumentato quindi lo ignora per punto precedente, invece di non includere la destinazione nel distance vector, la include a costo infinito. Può sostituire o complementare path hold down.
invece di non inviare proprio nessuna informazione, viene inviata comunque ma imponendo distanza infinita.
-

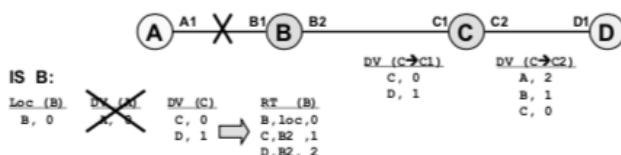
Split Horizon

Split Horizon

"If C reaches destination A through B, it is useless for B trying to reach A through C"

Quando cade nodo A ora B quando guarderà il DV(C) per trovare un collegamento con A non lo troverà perchè C raggiunge A tramite B quindi quando C invia il suo DV(C) a B non include la destinazione A e neanche B.

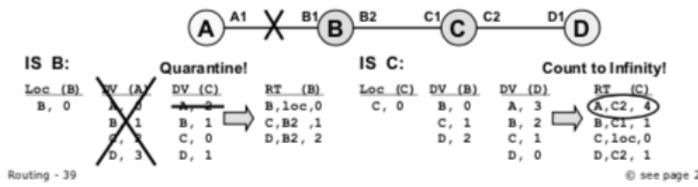
"Dato che C ha ricevuto info su A e B tramite B non invia a B info su A e B.



NELLA REALTÀ QUANDO B GUARDA IL DV(C) E NON TROVA A MA ASPETTA UN CERTO TEMPO (CIOE CHE SCADA LA ROUTE)

Path Hold Down

Path Hold Down
If link L fails, all destinations reachable through link L are considered unreachable for a certain period of time i.e., no routes to them are computed



Quando cade un LINK tutte le destinazioni che sono raggiungibili tramite quel link vengono contate irraggiungibili per un certo periodo di tempo.

Quando cade il LINK tra A e B oltre a non usare il DV(A) si scarta inoltre A dal DV(C). (Con split horizon invece A neanche era presente nel DV(C))

"Se cade un link considero le destinazioni raggiungibili tramite quel link irraggiungibili, non contandole dai distance vector che mi sono arrivati."

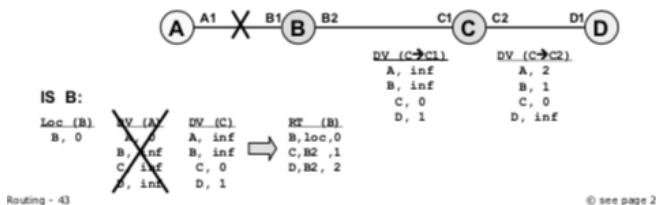
NON EFFICACE DA SOLO, RIMANE PROBLEMA SU ALTRI NODI. SI GENERA COMUNQUE LOOP TRA C E D PERCHE D INVIA A C RIGUARDO DESTINAZIONE A (cosa che non si fa in split horizon)

SE AVESSI TOPOLOGIA MAGLIATA INVECE FUNZIONA ED ELIMINA COUNT TO INFINITY.

Split Horizon - Poisonous reverse

Split Horizon with Poisonous Reverse

- More aggressive
- No theoretical advantages
- Practically, no need to wait for route expiration

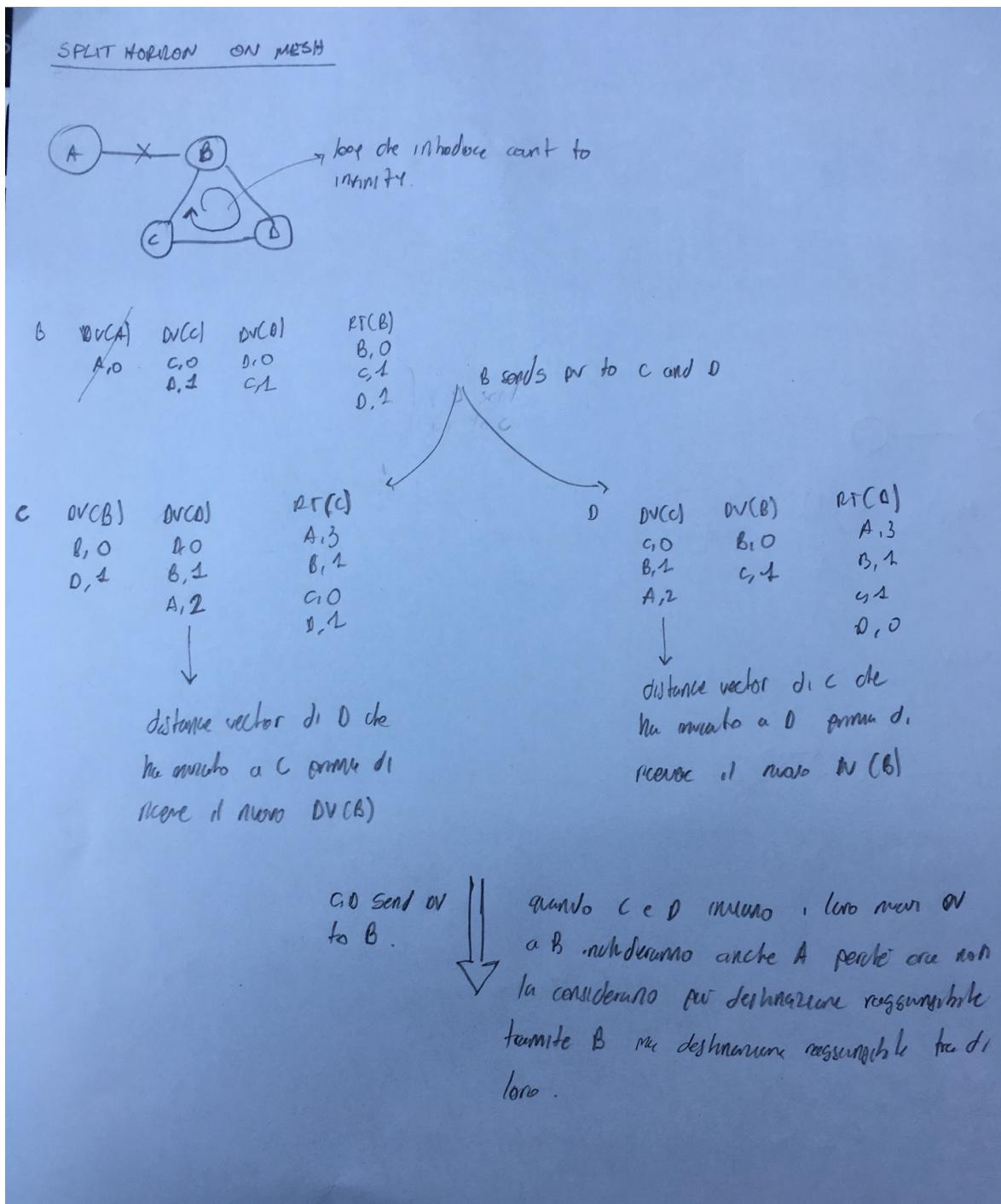


Quando cade Link A nel DV(C) invece di non includere le destinazioni A e B le metto a distanza infinita.

"Come split horizon ma invece di non includere la destinazione la metto a distanza infinita"

**NON DEVO PIU ASPETTARE CHE L'INFORMAZIONE SULLA DESTINAZIONE SBAGLIATA SCADA PERCHE LA COMUNICHERÒ SUBITO A DISTANZA INFINTA
NON FUNZIONA COMUNQUE SU TOPOLOGIA MAGLIATA.**

SPLIT HORIZON ON MESH



INTERNET ROUTING ARCHITECTURE

Il modello OSI non vale per qualsiasi tecnologia.

Protocollo: descrive come le informazioni vengano scambiate.

L'algoritmo dice che un'informazione necessaria è la distanza (Distance Vector) il protocollo definisce la distanza (esempio numero di next hop).

Dominio di routing: insieme di router che utilizzano lo stesso protocollo di routing. Costituiscono una sottorete.

Redistribuzione: un singolo router può appartenere a diversi domini di router, cioè usa più protocolli di routing. Riceve le informazioni tramite due protocolli diversi da due domini diversi per esempio, scambiandole dopo averle tradotte (esempio far combaciare la metrica, uno può usare la distanza in metri mentre un altro il numero di next hop o la banda minima).

Sistema autonomo: è un insieme di sottoreti che sono raggruppate dal punto di vista topologico e organizzativo, cioè gestite dalla stessa organizzazione con politiche uniformi. Esempio la rete di un service provider.

I router che si trovano all'interno del sistema autonomo si chiamano **interior gateway**, quelli che si trovano al bordo **Exterior gateway, border gateway, boundary gateway**.

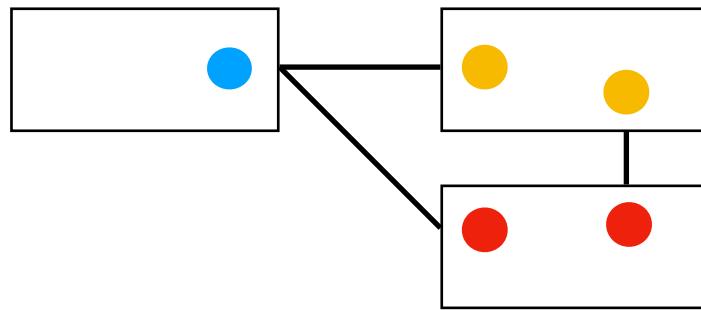
I border router utilizzano un protocollo apposito (*BGP, border gateway protocol*) per scambiarsi informazioni tra un sistema autonomo da un altro.

I dati per essere scambiati (redistribuzione) devono essere aggregati e filtrati (per esempio i border Gateway potrebbero filtrare solo le informazioni necessarie al loro sistema autonomo).

Nello stesso sistema autonomo ci possono essere più domini di routing (esempio exterior Gateway utilizzano un protocollo per parlare con un altro sistema autonomo e un altro protocollo per parlare con gli interiori gateway.)

I sistemi autonomi vengono identificati da un numero di due Byte.

Inter domain routing: routing tra sistemi autonomi basato su protocolli particolari come il **BGP**. Si può decidere quali destinazioni siano raggiungibili tramite operazioni di filtraggio, cioè implementare delle politiche di routing (che non sempre è il percorso più breve, ma il percorso che rispetta le politiche tra sistemi autonomi)



In internet sistemi autonomi gestiti da service provider detti **Tier 1**, tramite collegamenti di **peering** tramite cui scambiano dati. Forniscono servizi al **Tier 2**. I Tier 2 si collegano ai Tier 1 ma il tipo di collegamento che hanno è diverso, è di tipo cliente fornitore, come tra Tier 2 e Tier 3.

I **Tier 3** hanno una copertura locale e sono collegati ai Tieri 2. Si possono avere collegamenti tra stessi Tier 3, detti **private peering**.

Collegare i diversi Tier 1 è costoso perché possono trovarsi lontani tra loro (fibre ottiche molto lunghe e percorsi difficili), si creano quindi dei punti di aggregazione detti **NAP/IXP** che permettono il collegamento tra Tier 1. Si hanno quindi tanti switch che collegano i vari service provider Tier 1 alla rete locale NAP/IXP, definendo diverse sessioni di peering.

ROUTING PROTOCOLS

I protocolli di routing sfruttano gli algoritmi di routing visti precedentemente.

Due tipi di protocollo, **IGP** (interior Gateway protocol) dentro lo stesso sistema autonomo, **EGP** (exterior Gateway protocollo) tra diversi sistemi autonomi.

I costi non sono sempre delle metriche topologiche ma di tipo amministrativo. Nonostante una destinazione sia lontana può avere costo minore rispetto a destinazioni più vicine per motivi di politiche amministrative (prediligere la destinazione lontana).

IGP: diversi protocolli, hanno come output il percorso migliore perchè distribuiscono informazioni topologiche della rete.

DISTANCE VECTOR

LINK STATE

RIP routing information protocol

IGRP interior gateway routing protocol, proprietario CISCO.

E-IGRP enhanced IGRP

OSPF open shortest path first

Integrated IS-IS IS-IS è il protocollo che si usa nell'architettura OSI, basato su link state

EGP: diversi protocolli, il percorso scelto non coincide con il migliore, ma con le politiche amministrative. Si sceglie il percorso preferito non migliore.

BGP border gateway protocol, può essere usato anche come IGP.

IDRP inter domain routing protocol, evoluzione di BGP

STATIC ROUTING

INTERIOR GATEWAY PROTOCOLS

RIP

Anche gli host potevano implementare RIP, partecipando allo scambio di informazioni. Il problema è che se l'host partecipa al protocollo di routing può inviare informazioni di routing sbagliate perché non gestiti dall'amministratore di rete. Oggi un host ha un default Gateway che si occupa dell'inoltro dei pacchetti.

Basato su distance vector, la metrica è il numero di hop, il massimo è 15. La distanza 16 hop vuol dire non raggiungibile (soluzione count to infinity). Distance vector inviati periodicamente ogni 30s, così il router può usarlo appena lo riceve e poi non salvarlo, tanto lo riceverà di nuovo poco dopo. Si riduce così anche il traffico generale.

La differenza con l'algoritmo è che questo viene generato in ambiente astratto (invio distance vector per ogni modifica) mentre il protocollo lo implementa nella realtà (mando distance vector periodicamente, riduco traffico).

Operazioni di time-out, servono per accorgersi di modifiche topologiche della rete, per esempio rottura link. Non c'è altro modo di accorgersi della rottura di un link se non dal fatto di non ricevere distance vector.

IGRP

Proprietario CISCO, funziona quindi con router prodotti da CISCO.

Metriche più articolate, in totale 5 più un contatore di hop (risolve count to infinity):

- ritardo
- banda
- affidabilità link
- carico link
- massima dimensione pacchetti.

La scelta delle metriche dipende o dal tipo di traffico o dalla media pesata.

Multipath routing *se destinazioni raggiungibili tramite due percorsi, li tengo entrambi e li userò a secondo del loro peso. Se uno pesa 50 e uno 3, ogni 50 pacchetti sul percorso costo 3 ne mando 3 sul percorso costo 50. **Sfrutto di più il percorso meno costoso.***

OSPF

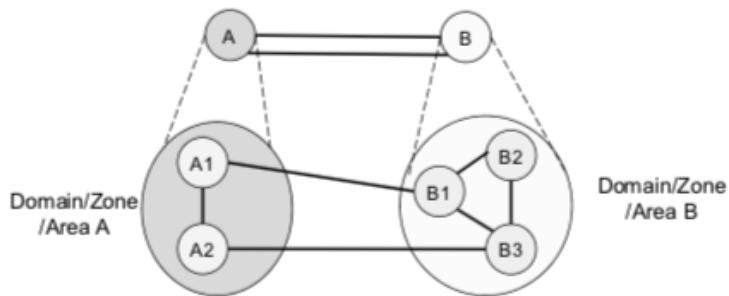
Routing gerarchico: *domini di routing divisi in aree diverse. Si cerca di ridurre la quantità di informazioni che i router devono gestire. I router*

così non si devono creare la mappa di tutto il dominio, ma solo della sua area, ricevendo i link state solo dai router della sua stessa area.

Routing gerarchico stretto

Un router dentro un'area non sa nulla di quello che c'è all'esterno —> massima scalabilità.

Ci sono quindi dei router di “alto livello” che hanno informazioni sulle varie aree.



Quando una destinazione è in un'altra area, lo mando ad un router della stessa area che è collegato ad un'altra area.

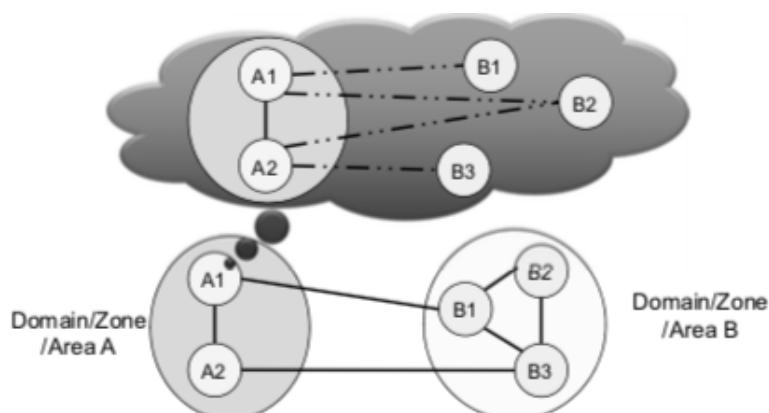
Un numero molto basso di router apprende, oltre alle informazioni sulla propria area, anche informazioni sulla topologia di collegamento tra aree.

È necessario un prefisso che indica in quale area si trovi il router ma questo non è possibile implementarlo su internet perché bisognava farlo prima che venissero assegnati indirizzi. Si usa quindi routing gerarchico lasco.

Scala bene sulla complessità.

Routing gerarchico lasco

Si usa un routing gerarchico lasco, ogni router ha informazioni sulle aree esterne senza sapere l'esatta topologia (si sa qualcosa ma non precisamente).



Questa soluzione è

- meno scalabile
- i router devono gestire una quantità di informazione maggiore
- non richiede indirizzamento gerarchico stretto
- Tutti gli host nello stesso dominio non devono avere lo stesso identifier , si utilizza prefisso.
- Implementabile in IPV4

Backbone area: area centrale a cui si affacciano tutte le altre aree. I router interni all'area si chiamano **internal router**, quelli al bordo si chiamano **area border router** e ricevono linkstate da tutte le aree più la backbone. I link state che riceve sono riassunti e vengono aggregati.

Broadcast Network

N router → N^2 links, complessità Dijsktra dir. proporzionale con numero link. Si usa quindi una struttura a stella attraverso uno pseudo nodo a cui tutti sono collegati in modo da trasformare la topologia a maglia in una a stella.

Solo per routing proattivo.

Integrated IS-IS

Implementa routing gerarchico.

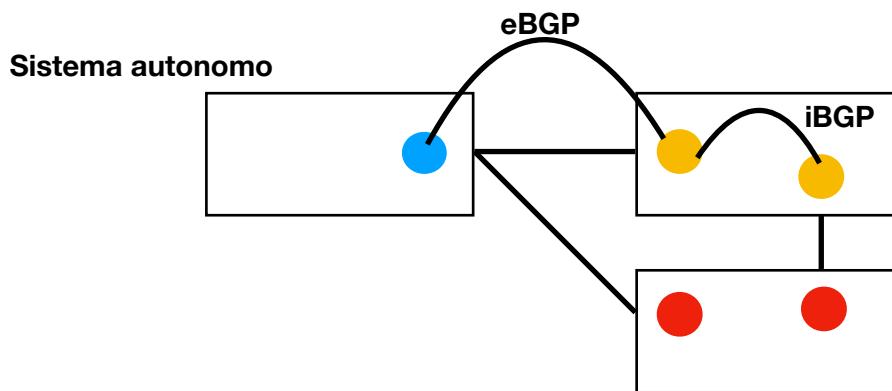
EXTERIOR GATEWAY PROTOCOLS

(e) **BGP (4)**: protocollo di tipo path vector, contiene destinazione più insieme di attributi che descrivono il percorso verso la destinazione (sistemi autonomi da attraversare ecc).

Politica configurabile per il calcolo delle route, ci sono diverse procedure per la scelta di percorsi diversi che dipendono dalle metriche. Con il BGP non posso inviare distance vector periodicamente perché la comunicazione avviene tra sistemi autonomi (troppe destinazioni) ma solo quando ci sono cambiamenti topologici e non viene inviato neanche per intero, ma solo la parte interessata alla modifica topologica. Vengono mandati su TCP

Nei protocolli IGP i router scoprono i vicini automaticamente (link state : OSPF con messaggi di hello, distance vector: ricezione vettore), nei BGP bisogna dirgli di comunicare con un altro configurandoli, definizione delle sessioni di peering.

iBGP è la versione usata come protocollo interno di BGP che serve per far comunicare due border gateway all'interno dello stesso sistema autonomo.



Ogni router di bordo all'interno del sistema autonomo devono essere tutti collegati insieme, peering session tra tutti.

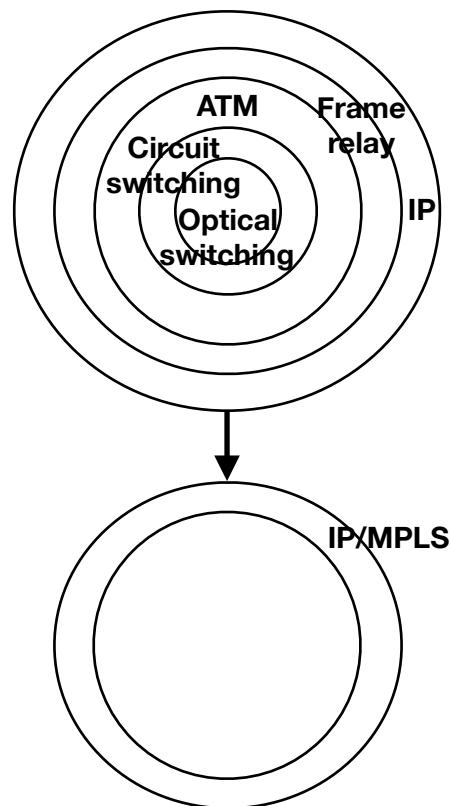
MPLS-multi protocol label switching

Tecnologia che usa routing *on the fly label switching*, viene definita multi protocollo perchè è indipendente dal tipo di protocollo utilizzato. Distribuisce il traffico sulla rete tramite operazioni di traffic engineering.

MPLS è la tecnologia che ci permette di costruire una nuova rete pubblica a larga banda (oggi basata su IP) senza utilizzare una tecnologia per reti pubbliche (ATM, frame relay).

Per rete pubblica si intende una rete su cui vengono venduti dei servizi (internet non era una rete pubblica in questo senso). **Una rete pubblica a più larga banda non potrebbe utilizzare solo rete IP, senza MPLS, perché IP è predisposto alla congestione (pacchetti per stessa destinazione seguono stesso percorso).**

MPLS permette di evitare la struttura stratificata formata da SONET/SDH, ATM, FRAME RELAY, IP.



Inoltre i pacchetti usano label swapping invece che network address. Non si guarda più l'intero pacchetto ma solo l'etichetta. In questo modo i router si velocizzano perché non usano longest prefix matching.

Operazione di **traffic engineering**, distribuzione del traffico, che sarebbe difficile da fare con routing by network address, per questo si usa label swapping.

Anche ATM utilizza etichette, si potrebbe inserire pacchetto IP in cella ATM, ma il routing pro attivo sarebbe diverso perché avrei un routing pro attivo per IP e uno per ATM.

MPLS invece non ha un suo routing proattivo ma utilizza quello dell'IP, massima compatibilità.

Il fatto di utilizzare etichette rende la rete IP **connection oriented (LSP)**, che invece nasce *connection less*, pur mantenendo alcune proprietà delle reti *connection less*. (esiste una modalità MPLS per l'uso in modalità *connection less*).

MPLS non include gli end Systems, ma si occupa solo della parte centrale della rete. Gli **LSR, label switch router**, eseguono label swapping. Ai bordi della rete i **label edge router ingress/egress** tolgoni e mettono le etichette ai pacchetti, togliendo le etichette torna ad essere un pacchetto IP.

LSP, label switched path, è un percorso da seguire per attraversare la rete MPLS, deciso a priori (connection oriented).

Nelle tabelle di “forwarding” di ogni router non c’è l’interfaccia a cui inviare il pacchetto, ma solo etichette di ingresso, next hop, nuova etichetta da utilizzare. Può capitare di avere per pacchetti diversi la stessa etichetta, ma devono essere mandati su interfacce diverse a meno che non vadano verso la stessa destinazione.

MPLS key elements (IMPORTANTE)

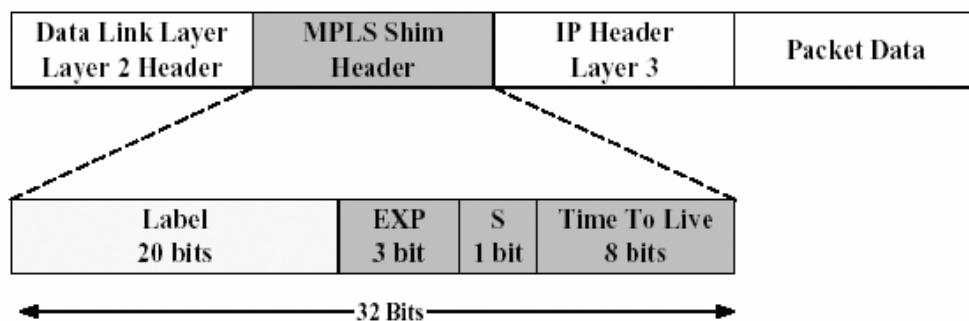
- | | |
|---|-----------------|
| - 1 Header , contiene etichetta. | PIANO DATI |
| - 2 Signaling , protocolli per scambio etichette | PIANO CONTROLLO |
| - 3 Enhanced routing , scelta del percorso soggetta a vincoli. | PIANO CONTROLLO |

1 SHIM HEADER

È messa tra l'intestazione di livello 2 e di livello 3, può crescere in dimensione cioè allontanare le due intestazioni. (shim → cuneo)

La prima parte è l'etichetta, 20 bit, (ci possono essere più etichette, il router guarda quella più esterna e poi eventualmente la toglie, passando ad una più interna)

- 3 bit EXP experimental, non hanno uso specifico, si usano per differenziare classi di servizio.
- 1 bit S bottom of stack, per capire se il modulo dell'header è l'unico o ce ne sono altri prima del pacchetto IP.
- 8 bit TTL time to live, quando l'etichetta viene tolta si può copiare il TTL nell'etichetta successiva oppure direttamente nel pacchetto.



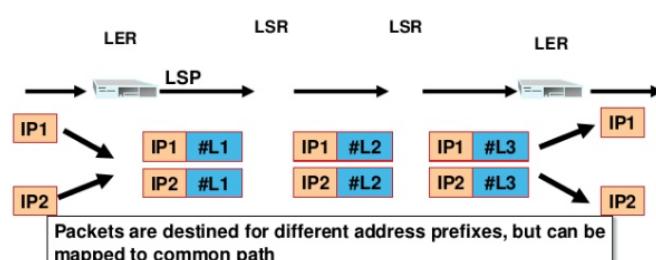
Atm and frame relay

Se faccio fare routing ad uno switch ATM seguendo le politiche MPLS, posso considerare quello switch come MPLS. Tutta la configurazione è solo software, quindi un router ATM può essere configurato come LSR MPLS.

LSP setup- Label switched Path

Stiamo parlando del piano di controllo.

La **FEC**, forwarding equivalence class, è una classe che indica pacchetti che devono essere trattati nello stesso modo da ogni LSR, che seguono lo stesso percorso e ricevono la stessa etichetta.



L'importante è che se assegno ai pacchetti la stessa etichetta vadano a finire nella stessa nuvola di destinazione.

Per decidere che etichetta usare per pacchetti di una certa FEC, bisogna fare:

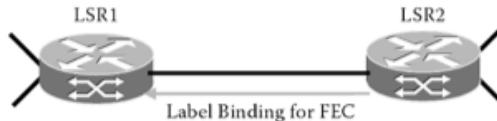
- A Label binding** decidere etichetta
- B Label mapping** associare etichette di entrata con quella di uscita —> tabella di forwarding
- C Label distribution** I nodi devono sapere quale etichetta si usa sul loro link, comunicazione informazioni,

A Label binding

Sempre fatto dall'LSR a valle (quello che riceve il pacchetto), downstream binding.

Il binding può essere:

- **Unsolicited:** quando LSR a valle scopre nuova route fa il binding dell'etichetta da usare a tutti gli altri, sia upstream che downstream.
 1. LSR1 and LSR2 have an LDP adjacency.
 2. LSR2 discovers a “next hop” for a particular FEC.
 3. LSR2 generates a label for the FEC and informs LSR1 about the binding.



- **On demand:** l'etichetta viene scelta da parte del nodo downstream solo dopo richiesta da parte di un altro LSR upstream.

LSR1 e LSR2 hanno pacchetti da inviare a specifica FEC, fanno richiesta di binding. Lo switch ATM inoltra la richiesta alla destinazione LSR4 il quale risponde con le label 7 per i pacchetti da LSR1 e 8 per i pacchetti da LSR2. A questo punto anche LSR3 fa il binding e invia le etichette 5 per pacchetti di LSR1 e 6 per pacchetti di LSR2.

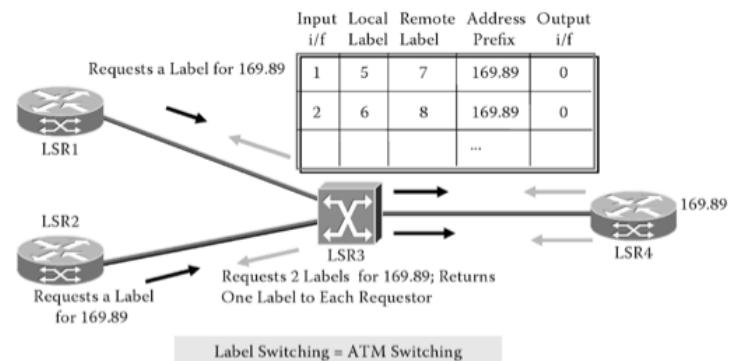


Figura-On Demand

Quindi LSR3 dopo aver scelto le due etichette di INPUT (5,6) fa il mapping con quelle di uscita (7,8). VEDERE MAPPING

Static label binding

È come definire una connessione *connection oriented*.

Non è scalabile, richiederebbe molto lavoro per cambiare tutte le connessioni.
Decisa dall'amministratore del dominio MPLS, massimo controllo.

Dynamic label binding

Scelta delle etichette dinamicamente, può essere iniziato da eventi diversi.

Diverse modalità :

- **Data/Traffic Driven** si decide di fare un binding (*unsolicited o on demand*) quando si ricevono pacchetti per una certa FEC.
- **Control Driven,** faccio binding quando mi arriva messaggio di segnalazione (piano controllo) oppure un messaggio di routing.

La modalità *control driven* si divide a sua volta in due tipologie:

- **Control driven label binding topology based**

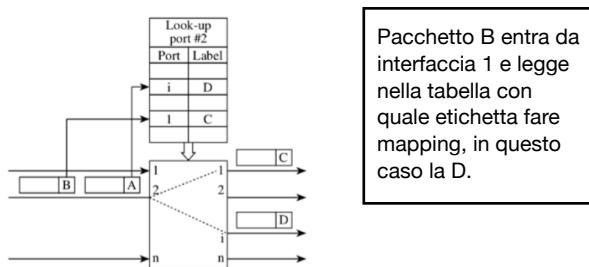
Mi arriva un messaggio di routing che mi fa scoprire destinazione, allora faccio un binding per un FEC di tutti i pacchetti che vogliono andare a quella destinazione. In questo modo ho le caratteristiche di una rete connection less, compreso recupero guasti, nonostante sia connection oriented.

- **Control driven label binding explicit**

Solitamente viene iniziato da un unico router, il LabelEdgeRouter dopo aver fatto binding chiede tramite segnalazione ad altri router di fare binding.
Creo una rete connection oriented.

B Label mapping

Si esegue il mapping tra etichetta di entrata (scelta dall'LSR), etichetta di uscita (scelta dall'LSR downstream) e next hop (basato sul routing). Il next hop viene scelto in base al routing. Queste operazioni creano effettivamente l'LSP.



C Label distribution

Il nodo notifica i vicini del binding delle etichette, o almeno a quelli a monte.

2 LABEL DISTRIBUTION PROTOCOLS

Ci sono tre alternative, incompatibili tra loro.

Per distribuire le etichette posso usare:

- 1) **Routing Protocol:** BGP, quando annuncio destinazione faccio anche distribuzione etichetta insieme. Messaggi BGP facilmente estensibile tramite aggiunta di attributi come etichette. *Lo si usa solo per la modalità di funzionamento topology based.*
- 2) **LDP** (label distribution protocol): nasce per questo specifico motivo, non troppo usato perché evoluzione di un protocollo proprietario CISCO.
CR -LDP constraint-based routing LDP
- 3) **RSVP** (resource reservation protocol) : progettato non per MPLS, ma per prenotare risorse su una rete con servizi a qualità garantita.
Abbastanza inefficiente però è quello maggiormente usato.

3 ROUTING PROTOCOLS

Gli switch MPLS devono scegliere comunque il percorso da far fare ai dati nella rete tramite protocolli di routing. In base al tipo routing (hop-by-hop o explicit) si scelgono i nodi da attraversare cioè l'LSP. Il label binding e mapping e distribution serve solo per permettere la comunicazione tra i nodi che sono già stati scelti dal routing.

Il routing pro attivo è lo stesso in MPLS e in IP tradizionale, quello che cambia è il routing on the Fly , in MPLS tramite etichette e in IP tramite indirizzi.

- 1) OSPF (IGP , link state)
- 2) IS-IS (IGP, link state)
- 3) BGP-4 (EGP, path vector)

Questi protocolli sono utilizzati per trasportare le informazioni sulla topologia della rete.

In MPLS noi vorremmo scambiare altri tipi di dati, constraint data, cioè dati che vincolino le decisioni di routing, per esempio: capacità dei link, utilizzo dei link, dipendenze tra link.

Tutti questi vincoli non servono per decidere il percorso migliore, che è già stato calcolato (dal routing proattivo), ma solo per vincolare le decisioni e per il recovery dagli errori, cioè per operazioni di traffic engineering.

Constraint based routing, è il routing basato su protocolli modificati con vincoli:

- 1) OSPF-TE traffic engineering
- 2) IS-IS-TE traffic engineering
- 3) BGP-4 non modificato, solo topology based.

OSPF-TE e IS-IS TE (routing) vanno usati insieme a LDP e RSVP (distribuzione etichette)

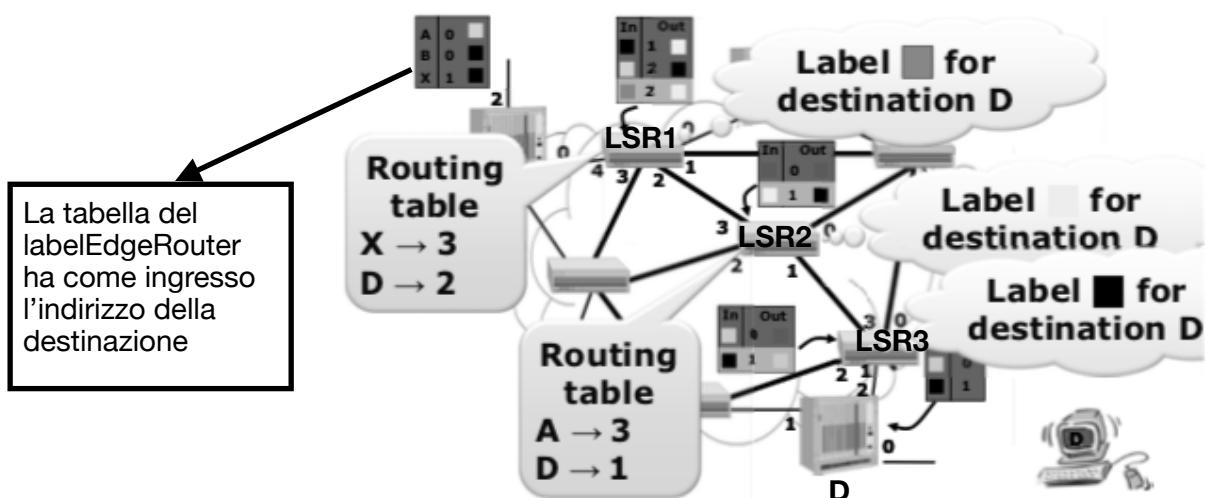
Routing Modes

In MPLS quando router scelgono il percorso facendo il mapping e scegliendo il next hop, lo possono fare:

- 1) HOP BY HOP: modo completamente distribuito, utilizzando le informazioni del routing proattivo (routing table) e non dei vincoli/traffic engineering. Seguono il percorso definito dal protocollo di routing.
- 2) ROUTING EXPPLICITO: ignorano informazioni del routing proattivo e fanno una scelta che viene imposta da un altro router mandandogli un messaggio. Scelta utilizzando i vincoli, metriche e traffic engineering. Solitamente il router che sceglie gli LSR da attraversare è l'ingresso o l'egress label router.

Hop-By-Hop

Un router sceglie la sua etichetta per un FEC, dopo fa il suo binidng, lo distribuirà ai vicini, poi deve fare mapping e deve sapere chi è il next hop. Guarda la sua tabella di routing e vede se il nodo next hop avesse fatto binding precedentemente scegliendo un'etichetta e allora userò quell'etichetta per indicare il next hop. Ricavo quindi l'informazioni sul nextHop dalla tabella di routing che mi sono creato dal routingProattivo.



LSR1 sceglie etichette grigia per destinazione D, e fa lo swapping con quella bianca che gli è stata data dall'LSR2. Questo lo fa perchè leggendo la sua routing table vede che D è raggiungibile tramite la porta 2 la quale è collegata alla porta 3 dell'LSR2.

Stessa coda da lì in poi.

Routing esplicito

Basato su vincoli. Un solo router sceglie il percorso e lo impone a tutti gli altri. Solitamente il router che sceglie il percorso è l'LSR di ingresso ma non sempre è così.

Constraint based routing, non si può operare in modo distribuito, se ogni router decidesse per sé non si avrebbero scelte coerenti, si opera in modo centralizzato, uno solo sceglie per tutti.

Anche i protocolli per il label distribution vanno modificati:

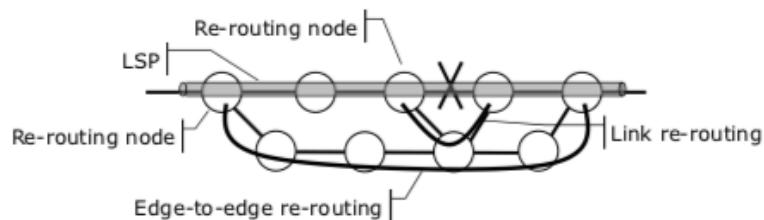
- 1) CR-LDP
- 2) RSVP-TE

Questi vanno usati con OSPF-TE e IS-IS-TE (scelta nodi da attraversare).

In conclusione MPLS offre nuove possibilità di traffic engineering, garanzia della qualità del servizio (non ancora supportato), traffic engineering per classi e fast fault recovery.

Fast Vault recovery

Si usano degli LSP predefiniti per evitare il link rotto. Quando si crea un LSP si crea anche un LSP di backup. Quando si rompe quel link, nella modalità di link re-routing, il nodo vicino si accorge della rottura e lui sa già che c'è un'etichetta scelta per un percorso di backup. Il nodo farà quindi POP dell'etichetta del link rotto e PUSH dell'etichetta del link di backup. L'unico che lavora è quindi il nodo vicino alla rottura, è l'unico che sa di dover cambiare l'etichetta del link rotto con quella del link di backup.



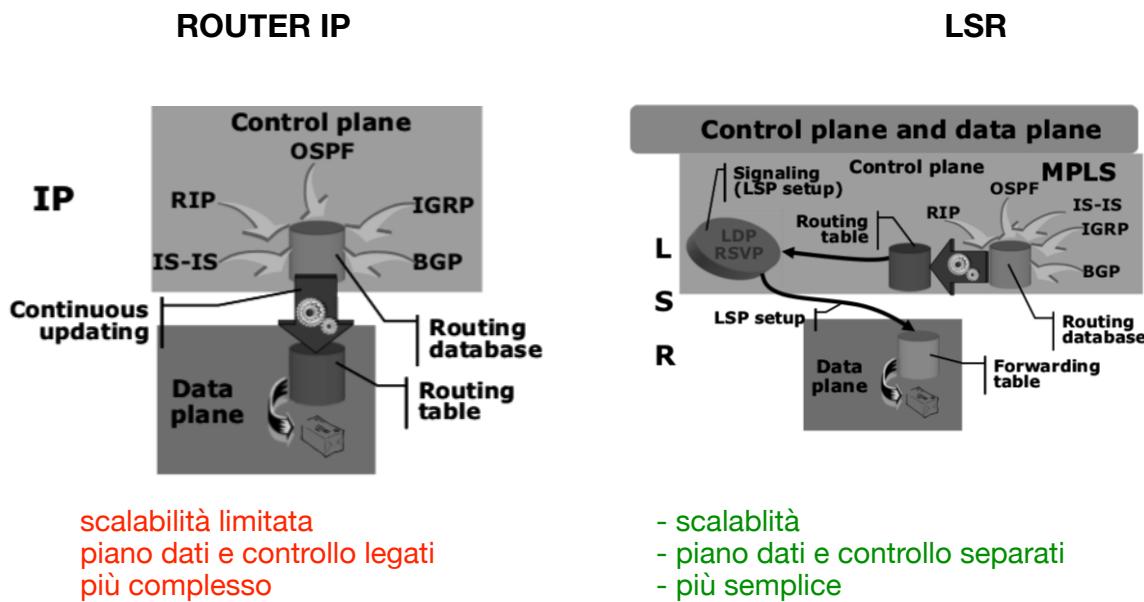
TRAFFIC ENGINEERING - MPLS

Importante perché nel routing IP tradizionale il traffico viene aggregato ai router. Se più router inviano ad una stessa destinazione, i link vicino alla destinazione saranno sovraccarichi mentre gli altri vuoti. —> problema congestione, problema link inutilizzati.

Si cerca quindi di distribuire il traffico. Se più router devono inviare alla destinazione D non invio alla FEC di D, ma invio su più FEC diverse con LSP diverse.

IP) Piano di controllo e piano dati sono legati, le informazioni prodotte dal piano di controllo vengono usate dal piano dati. —> tradizionale

MPLS) Con MPLS piano dati e controllo sono separati. Le informazioni prodotte dal piano controllo (routing table) non servono al piano dati per l'inoltro perché funziona con etichette. Tramite la segnalazione scambio informazioni tra piano dati e piano controllo.



Ci sono delle estensioni di MPLS che gestiscono le operazioni di traffic engineering:

- MPΔS MPLambdaS control plan in reti ottiche
- GMPLS Generalized MPLS control plan in qualsiasi rete, a pacchetto, a circuito o ottica.

CoS

Class of service. Viene data priorità a diverse FEC per operazioni di Per class traffic engineering.

QUALITY OF SERVICE

Rilevante per applicazioni multimediali, diverse rispetto applicazioni tradizionali come: mail, web, file-transfer.

Applicazioni Multimediali caratterizzate

- da flusso continuo di dati, il profilo di traffico generato deve essere uguale al profilo di traffico ricevuto (se mando frame video ogni 30ms devo visualizzare i frame ogni 30 ms).
- Interattività
- richiedono larga banda e solitamente molte comunicazioni di gruppo
- Larga banda di trasmissione
 - alta capacità trasmittiva
 - memoria all'interno dei nodi della rete (buffer)
 - processing power (routing)
 - switching
- comunicazione tanti a tanti

Requisiti imposti

- perdite limitate a parte piccole tolleranze in determinate applicazioni multimediali
- ritardi costanti
- ritardi bassi

I ritardi sono importanti e il problema è che dipendono dal traffico istantaneo che c'è sulla rete in quel momento.

In conclusione dobbiamo limitare il numero di pacchetti che arrivano ai nodi della rete e gestire in modo appropriato i pacchetti che richiedono una particolare Quality of services.

Supporto alla qualità del servizio:

- classificare il traffico (telefonate, email ecc) in base alla sensibilità ai ritardi. Classifico il traffico in base a mittente, destinazioni e porte attraversate.
- Applicare algoritmi di scheduling per capire a quale pacchetto dare precedenza.

- Controllo del traffico in rete
- Qos routing

Strumenti per la qualità del servizio

Ci sono vari strumenti che possono essere adottati

- **classificazione del traffico**

identificare i pacchetti che hanno bisogno di una qualità del servizio garantita.

Capire in quale coda inserire i pacchetti.

Le informazioni necessarie per capire la priorità sono contenute nei TCP/UDP header.

- **Packet scheduling**

Posso implementare una sola coda FIFO (first in first out) oppure più code.

I pacchetti quindi vanno distinti in code diverse, una coda per ogni flusso.

Algoritmi di scheduling:

- Priority queuing
- round Robin (weighted)
- class based queuing
- weighted fair queuing (WFQ) migliore, ad ogni flusso do il servizio come se fosse su circuito dedicato punto-punto a tot bit/s.
- deadline queuing

Questi algoritmi sono troppo complicati per essere usati ad alta velocità, possono essere usati tra due o tre code per esempio, ma questo vorrebbe dire incodare più code in un'unica coda e perdere la gerarchia quindi. Meglio del file transfer ma non è il funzionamento ideale.

- **controllo del traffico (policing & shaping)**

Bisogna garantire che il traffico che entra in rete rispetti un determinato profilo.

Se ciò non succede:

- i pacchetti non conformi vengono ritardati
- vengono scartati
- viene assegnata una priorità più bassa

Si utilizza il CALL ADMISSION CONTROL (CAC) che si occupa della segnalazione di

- descrizione del traffico generato
- descrizione del servizio richiesto

QoS routing

Per QoS routing si intende la ricerca di una route che rispetti le risorse che sono state richieste.

Le decisioni di routing sono basate sulla disponibilità delle risorse e non solo su informazioni topologiche della rete.

Bisogna scambiare dinamicamente informazioni tramite protocolli di routing i quali le distribuiscono real-time.

Solitamente si cerca di prevenire problemi, quindi solitamente la rete dovrebbe essere dimensionata tenendo conto del caso peggiore e della distribuzione del traffico.

INTERNET Q.O.S FRAMEWORKS

INTSERV - Integrated services

Caratteristiche

- per flow resource reservation tramite protocollo RSVP (Resource Reservation Protocol)
- qualità del servizio garantita tramite per flow queuing all'interno dei router

Limiti

- molto complesso
- bassa scalabilità
- non utilizzabile su reti pubbliche, cioè reti di grande scala

DIFFSERV - Differentiated Services

Caratteristiche

- non ci sono garanzie sulla qualità del servizio
- non vengono riservate risorse
- Vengono definiti diversi servizi per differenti classi di traffico, tramite il campo DS (Differentiated services) e code per ogni classe.
- ottima scalabilità

Implementazione tramite operazioni di

- Network engineering
- traffic engineering
- Utilizzo di politiche nei collegamenti con controllo degli accessi

Limiti

- Bassa efficienza, la maggior parte del traffico è best-effort

Nonostante questo viene usato per esempio nella telefonia IP perchè è semplice e offre un'ottima scalabilità.

PRINCIPLES OF MODERN LAN

Standard per reti locali

- IEEE 802:famiglia di standard, contiene ethernet, wifi ecc ...
- EIA/TIA 568: contiene cablaggio strutturato, ci sono regole da seguire per il cablaggio.

In IEEE 802 abbiamo 802.2 (logical link control protocol) LIVELLO 2

Gli altri (802.5,802.3,802.11) utilizzano medium access control sublayer LIVELLO 1,2
802.1 Management , NON è un protocollo di LIVELLO 3

MAC and LLC sublayer

- MAC: definisce l'accesso al mezzo, CSMA/CD protocollo di accesso al mezzo utilizzato in ethernet. In più definisce indirizzamento.
- LLC: offre il protocollo demultiplexing (per riconoscere ipv4 da ipv6), offre connessioni orientate a livello 2 e flow control.

LLC oggi non è necessario, non serve fare demultiplexing. Le funzionalità avanzate come le connessioni orientate e flow control possono essere svolte a livelli più alti, come livello 4 (flow control tramite TCP).

ETHERNET DIX si differenzia rispetto all'802.3 (standard IEEE) perchè non c'è LLC. L'802.11 rimasta fedele allo standard IEEE mantenendo LLC, ma disabilitando le funzionalità a livello 2 come il flow control.

Dispositivi su reti locali:

- LIVELLO 1 **ripetitori - HUB** (repeater multiporta)
- LIVELLO 2 **bridge - SWITCH** (bridge multiporta)
- LIVELLO 3 **router**

Dispositivi necessari per ampliare la rete.

Repeater (HUB=repeater multiporta)

Dispositivo di interconnessione a livello 1 per rigenerare il segnale. Per esempio nel vecchio ethernet su rame il diametro massimo di rete è 2 km, ma il rame non me lo permette per degradazione del segnale. Utilizzo repeater per rigenerazione segnale. Replica solamente il segnale in ingresso , ma rigenerato. Tramite repeater non posso collegare tecnologie diverse (wifi e ethernet) e neanche due reti a velocità diversa. Posso però usarlo per collegare pezzo di rete in fibra e pezzo in rame

perchè hanno stesso MAC.

C'è un unico dominio di collisione quando unisco due reti diverse (rame e fibra). *Gli HUB sono semplici repeater ma multiporta (wifi extender).*

Bridge (Switch=Bridge multiporta)

Evoluzione dell'hub che opera a livello 2. Riceve la trama la elabora e fa il forwarding. **Lavorando a livello 2, posso interconnettere LAN di livello 2 diverse (ethernet, wifi oppure ethernet, ethernet, fast ethernet).** Bridge implementa **store and forward mode**, mi assicuro di ricevere tutta la trama e dopo la ritrasmetto, eventualmente modificandola. La porzione di livello 1 va ricreata, quella di livello 2 viene rigenerata eventualmente riconvertendo indirizzo MAC, LLC. I livelli superiori non vengono modificati.

La metodologia store and forward riesce a sdoppiare il dominio di collisione da quello di broadcast. **Quindi il dominio di collisione viene spezzato ma non quello di broadcast.**

Full duplex, elimina definitivamente il problema delle collisioni (che non veniva risolto del tutto solo dividendo i domini), eliminando anche le collisioni sul link che collegano il client al bridge. Si dividono i canali di invio e ricezione (il cavo ethernet per esempio è costituito da 8 fili) —> se ho canali diversi non posso avere collisioni.

I vantaggi di questa modalità sono 2:

- non mi serve più CSMA/CD (collision detection)
- il throughput raddoppia (almeno teoricamente) utilizzando full duplex

Gli switch sono bridge multiporta.

Hub and spoke topology, topologia a Stella su cui si basano le reti moderne, basate su full duplex, switch e ethernet. (switched ethernet).

Switched ethernet

Basato su transparent bridges, standardizzata in 802.1D (protocollo di management), significa avere bridge plug and play, cioè senza necessità di cambiare la configurazione e senza influenzare le funzionalità della rete. Le performance però possono cambiare. *Ogni porta del bridge ha un indirizzo MAC* ma non viene utilizzato per fare il forwarding, viene usato solo per trame generate/ricevute dallo stesso transparent bridge.

Posso usare meccanismi di forwarding più inteligenti come **smart forwarding**: determinata la porta a cui è collegato dispositivo ed inoltro traffico solo sulla porta corretta. Avrà bisogno di

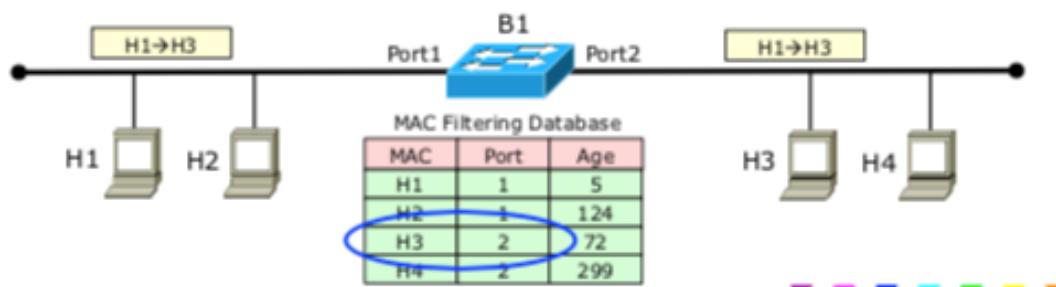
- 1 FILTERING DATABASE, MAC forwarding table.

- 2 BACKWARD LEARNING (per popolare la tabella)
- 3 SPANNING TREE, algoritmo di loop detection e quindi di rottura del loop

1. Filtering database

Tabella basata su tre colonne, la prima identifica le destinazioni, la seconda la porta tramite cui viene raggiunta destinazione e la terza è l'ageing time (tempo di vita entry tabella). A livello locale va bene , **a livello globale no avrei troppe entry per ogni tabella (MAC diversi per ognuno)**. Massimo 2-64k entries dinamiche, massimo 1k entries statiche. *Le entri statiche non hanno ageing time mentre quelle dinamiche si.*

Il flooding viene effettuato spesso solo nel transitorio iniziale perchè devo popolare la tabella, poi tramite 2. **backward Learning** non verrà più utilizzato. Tramite questo algoritmo, tramite indirizzo MAC sorgente posso conoscere nuove informazioni (se H1 invia a H2 pacchetto tramite porta 1, allora sicuramente H1 è raggiungibile tramite porta 1). Durante il backward learning il MAC destinazione non viene considerato.



Per aggiornare tabella:

- refresh ageing time
- refresh port

Se MAC non in tabella, host sicuramente raggiungibile tramite flooding, faccio flooding su tutte le porte a parte quella da cui ho ricevuto indirizzo MAC sconosciuto. —> **quindi se viene inviato pacchetto a un host non presente in tabella quel pacchetto verrà inviato a tutta la rete.**

Se MAC presente in tabella, è probabile che se l'informazione è sbagliata, l'host non sia raggiungibile

Problema mobilità —> host si sposta e non mi accorgo dello spostamento.

Casistiche possibili:

- Se subito dopo spostamento l'host invia frame broadcast non avrò sicuramente problemi perchè tutti riceveranno il frame, switch compresi, che propagheranno tra di loro il frame broadcast —> stesso dominio di broadcast.
- Se viene invece generato pacchetto unicast ci possono essere errori, perchè non tutti gli switch aggioreranno le loro informazioni.
- Se non viene generato nessun frame posso avere problemi di forwarding.

Macchina windows solitamente genera sempre pacchetti broadcast —> posso spostarle senza avere problemi.

Unix server e virtual host generano poco broadcast —> il loro spostamento può causare problemi.

L'ageing time è una soluzione ragionevole in caso di non invio di segnale broadcast perchè sicuramente il tempo sarà scaduto e le entries sbagliate saranno cancellate —> solo in casi relativamente semplici.

I filtering databases sono soggetti a possibili attacchi:

- MAC flooding: *si generano trame con mac sorgenti casuali* che cambiano velocemente —> **saturo filtering database che manderà in flooding tutto il traffico in eccesso.**
Si limita quindi il numero di MAC che può essere appreso su ogni porta.
- Packet storms: *generare frame verso destinazioni random e non esistenti.* Gli switch continueranno a fare flooding perchè non sanno dove sono quei dispositivi. **Si cerca di generare il maggiore traffico possibile —> occupazione di risorse con traffico inutile.**

3. Spanning tree

Disabilita virtualmente un link che genera il LOOP.

Problemi:

- loop di pacchetti
- backward learning non può più operare

Nel caso di loop di pacchetti, che in IP non creano problemi perchè pacchetti hanno time to live.

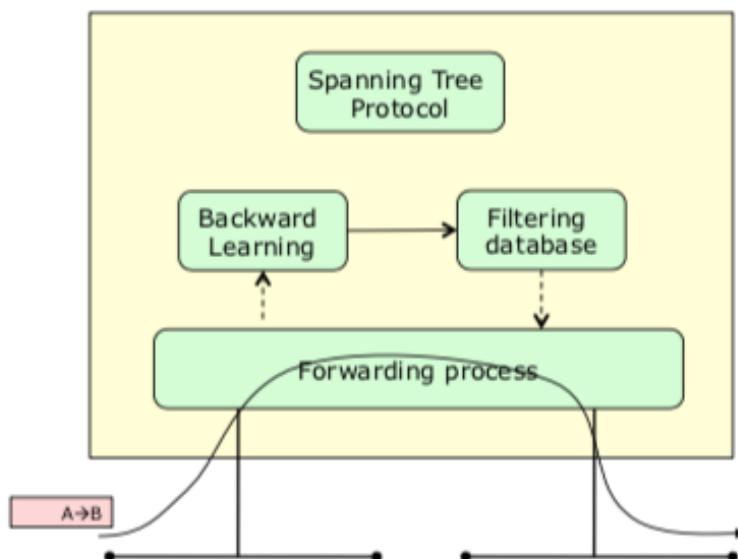
In ethernet non c'è TTL e i loop creano problemi. I pacchetti multicast e broadcast molto probabilmente creeranno loop. Frame verso nodo non presente

nella rete può creare loop perchè tutti gli switch continueranno a fare flooding. Solitamente raramente succede questo se non nel caso di un attacco alla rete. Standardizzato 802.1D

Architettura Bridge/Switch

Forwarding processo a livello fisico tramite shared bus (economica) oppure switching matrix (più costosa, pacchetti inviati contemporaneamente su più uscite). Database filtering, spanning tree, backward learning a livello software.

Per migliorare le prestazioni anche Database filtering, backward learning sono stati implementati in hardware tramite TCAM. Spanning tree restato in SW, sta nel piano di controllo, non ha bisogno di prestazioni elevate.



Router

Dispositivo di livello 3 (IP), spezza dominio di collisione e di broadcast (cosa che switch non fa). Il loro utilizzo è dovuto al fatto di limitare il traffico broadcast.

Il router nonostante sia di livello 3 può essere utilizzato all'interno di una rete locale, ma in determinati casi. Il livello 2 a livello locale è plug and play +LVL2, dispositivi di livello 2 non richiedono configurazioni per farli funzionare. Se invece metto router in una rete devo configurarlo altrimenti non funziona. -LVL3

Il livello 2 inoltre supporta la mobilità +LVL2, posso spostare apparecchio di livello in un'altra rete e funzionerebbe comunque senza configurarlo (MAC univoci).

Per il router non funziona così perchè sto cambiando rete IP e quindi il mio vecchio indirizzo IP non funzionerà sulla nuova rete, dovrò cambiarlo. **-LVL3**

EFFICIENZA Più lan insieme a differenza di una singola lan migliorano l'efficienza, una sola lan avrebbe troppo broadcast. **+LVL3**

SICUREZZA Switch più vulnerabile per privacy utenti e ad attacchi. **+LVL3**

GESTIONE se ho molti switch la rete è difficile da gestire a livello 2, per esempio per spanning tree. **+LVL3**

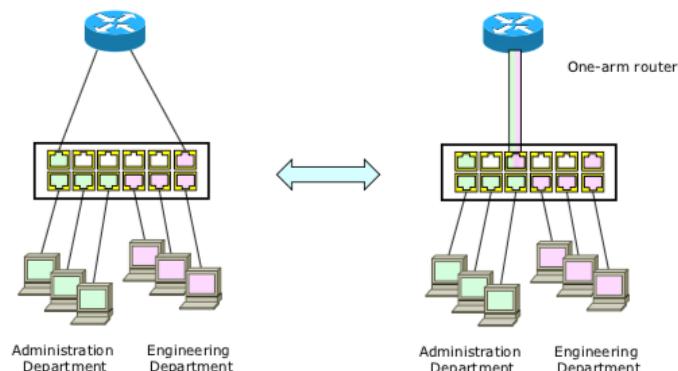
Bisogna capire quale tecnologia utilizzare, se rimanere a livello 2 o usare livello 3 o entrambi.

Per gestire più LAN, creo reti di switch differenti —> non è la soluzione migliore, spesso la divisione tra reti di livello 2 e 3 non viene pensata geograficamente (edificio 1,2,3) ma per unità aziendali (amministrazione, segreteria ecc). Costi elevati, per ogni LAN ho n switch , per M LAN o NM switch, molti dei quali non saranno totalmente utilizzati.

Una soluzione migliore sono le VLAN (virtual LAN), utilizzo unica rete di livello 2 ma separata virtualmente, unico switch su cui decido che alcune porte siano destinate ad una certa LAN e altre ad un'altra LAN. L'architettura dello switch ora cambia perchè avrà per ogni partizione virtuale di SWITCH N backward learning e N filtering database. A livello fisico tutte le filtering database sono sulla stessa TCAM ma virtualmente separate.

Le VLAN separano quindi il dominio di broadcast, come fare per farle comunicare?
Per farle comunicare ho bisogno di un router, tratto le VLAN come fossero LAN ethernet separate. Posso destinare altre due porte dello switch, una per ogni VLAN alle porte del router.

ONE ARM ROUTER, invece di usare più porte dello switch, utilizzo una sola porta che sarà collegata però a più VLAN. —> sarà il default Gateway delle due VLAN,



quindi dovrò avere due indirizzi IP per ogni subnet. Preferibile questa soluzione perchè l'altra con router normali è poco flessibile. Messaggi Broadcast non possono passare da una VLAN all'altra perchè domini separati da router.

Come fare per associare un frame ad una VLAN?

- Possiamo marcare le porte dello switch, ogni computer invia trame non marcate, è lo switch che colora poi le trame sulla base della porta di ingresso. Quindi se pacchetto che entra da porta rosa è destinato a porta verde non verrà inoltrato perchè sono VLAN diverse. Questo metodo ha validità solo LOCALMENTE, con un solo switch. **ACCESS**
- Se ho più switch, sul link che li collega ho più trame di diverso colore e in più la porta collegata al link dovrebbe avere più colori → lo switch che riceve non potrà più basarsi sul colore della porta, ma deve basarsi su quello delle trame. Si risolve con TAGGING. **TRUNK**

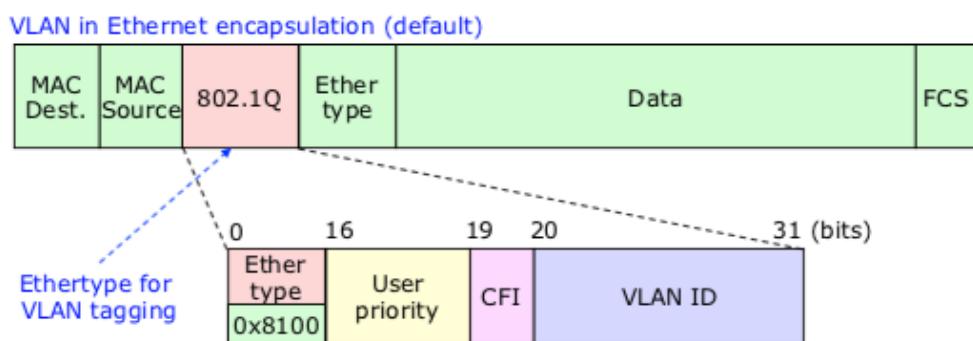
Quindi mentre in modalità ACCESS i vari colori delle porte mi servono solo per distinguere diversi percorsi all'interno dello stesso SWITCH (quindi le trame che poi escono dallo switch saranno di nuovo bianche, LOCALE), con la modalità TRUNK estendo il colore anche oltre lo SWITCH. Tramite TRUNK l'host locale è collegato allo switch tramite access, il quale colora il pacchetto. Lo switch a sua volta è collegato in modalità trunk ad un altro switch in modo da mantenere il colore.

Ethernet non mette a disposizione un campo per identificare le trame, devo modificare la trama aggiungendo un campo per gestire TAGGING. Viene definito un ETHER-TYPE specifico, che dice al dispositivo che è un ether type specifico che avverte che i prossimi bit saranno destinati al VLAN_ID.

Tagging standardizzato IEEE 802.1Q (.1, destinato al piano controllo non al piano dati)

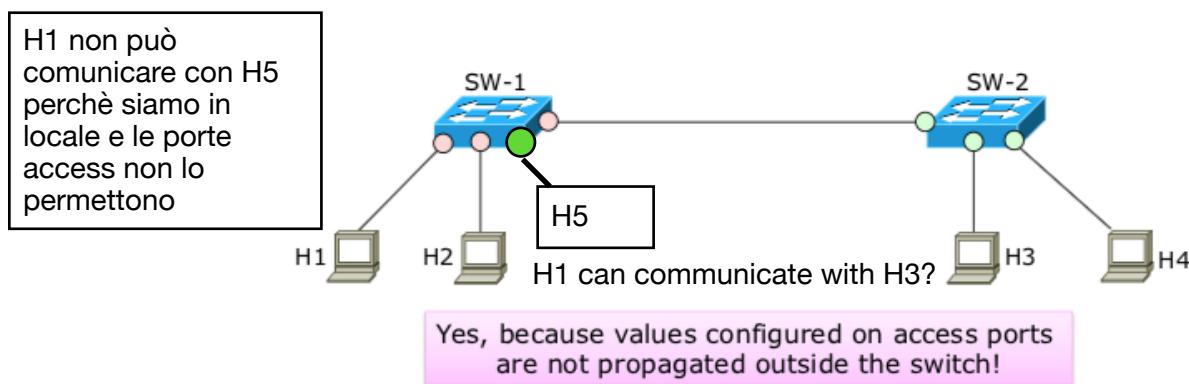
- VLAN_ID mi dice qual'è la vlan a cui appartiene la trama, 32 BIT, 1-4094.
- CFI serve per token ring, non si usa più.
- USER PRIORITY definisce le priorità su ethernet a livello IP c'era il type of service).

HEADER ETHERNET



- Se voglio colorare le porte, soluzione 1 (non tagging) configuro le porte in **modalità access**. *In questo caso le trame non sono colorate ma solo le porte.*

Il fatto di colorare le porte funziona solo localmente, tramite porte di colori diversi posso riconoscere VLAN diverse ma all'interno dello stesso switch. Switch diversi con porte con colori diversi posso comunque comunicare perché i vincoli non vengono propagati al di fuori dello switch, cioè i pacchetti sul Link di collegamento tra le due VLAN le trame sono bianche (perchè non ho configurato quelle porte in modalità TRUNK).



- Se voglio colorare le trame devo usare **modalità trunk**, configuro le porte in modalità trunk dicendo tutti i colori che vorrò utilizzare con quella porta in modo che sia in grado di inoltrare tutte le trame di quei colori.

Posso avere anche porte ibride che gestiscano sia trame bianche e trame colorate.

Come assegno gli host alle VLAN?

- **1 Port based VLAN**
- **2 Transparent assignment**
- **3 Per user assignment**
- **4 Cooperative Assignment**

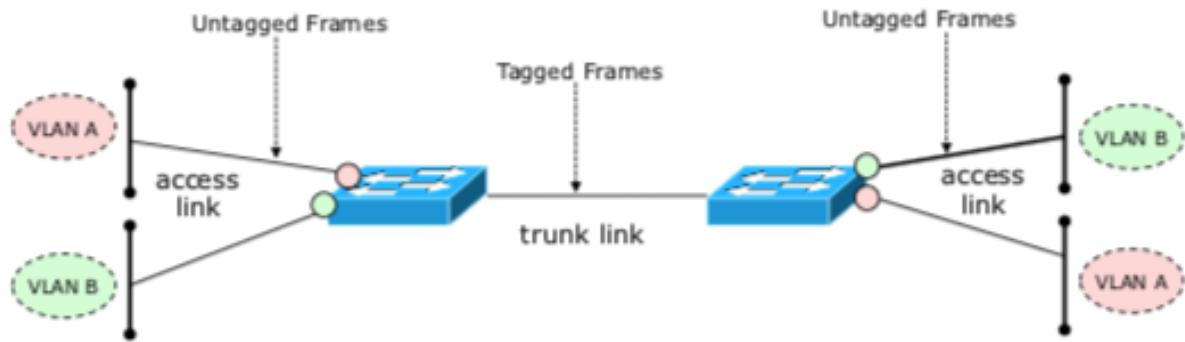
1 Port based VLAN :

Gli host mandano trame bianche e poi vengono associati alle VLAN sulla base della configurazione statica dello switch. (ACCESS). Devo decidere a priori se ad una certa porta di un certo switch collegherò un host che appartiene alla VLAN rossa , verde o gialla.

Configurazione non ottimale, devo configurare a mano tutte le porte e soprattutto non gestisce in modo opportuno la mobilità.

In realtà questa è la più usata, nonostante non gestisca mobilità.

Le differenti VLAN vengono collegate tramite TRUNK LINK tra switch.



2 Transparent assignment:

Cerco di offrire servizio di mobilità, non configuro a mano porte switch ma faccio in modo che lo switch impari quale colore è necessario assegnare alla porta a runtime sulla base di qualche informazione del dispositivo connesso (esempio indirizzo mac). Difficile da gestire, usata poco.

3 Per user assignment:

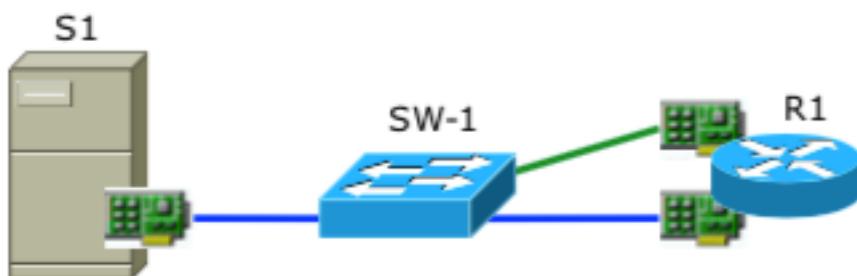
Utilizzo username e password per utente, simile al port based Assignment ma qui il colore si basa sull'UserId. Standardizzato 802.1X, protocollo di autenticazione. Buona soluzione ma solitamente se ne fa a meno perché facilmente soggetto ad errore.

4 Cooperative Assignment:

Sono io host che dico alla rete di che VLAN faccio parte, è l'*HOST che va a colorare il traffico*.

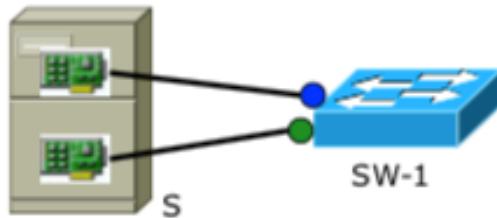
Modalità anarchica della gestione, non piacevole per amministratore di rete dato che è il client a gestire. Risolve però problema mobilità, scelgo io in che VLAN sono indipendentemente dal posto in cui mi trovi.

Single Vlan per NIC:

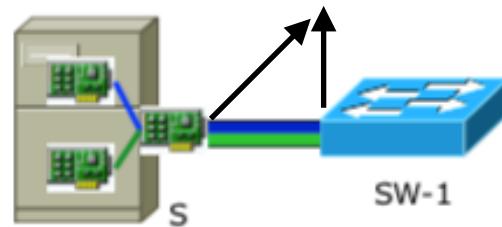


Host con singola scheda di rete S1, manda pacchetti per la sua VLAN (blu) taggando il proprio traffico configurando la propria interfaccia.

Multiple Vlan per NIC:



Entrambe in trunk



Senza Vlan nell'host:

- Due interfacce virtuali di rete, ognuna collegata ad una Vlan

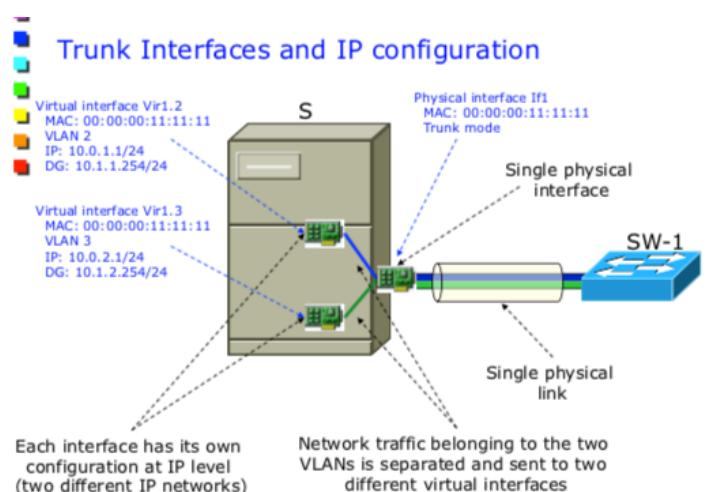
Con Vlan nell'host:

- due interfacce virtuali di rete, ognuna fa parte di una Vlan, associate all'interfaccia di rete fisica. Scheda di rete fisica in modalità TRUNK. Condividono tutte lo stesso MAC ma su LAN diverse quindi mantengo univocità.

Gestione dell'interfaccia fisica in modalità trunk

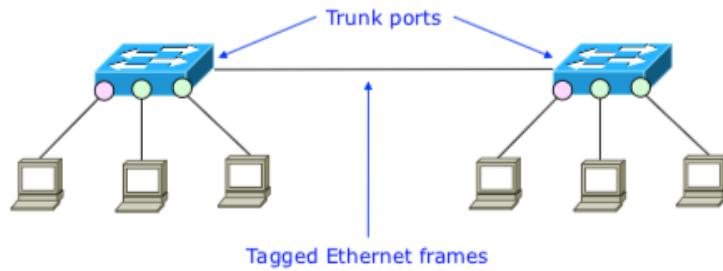
Vado a configurare l'interfaccia fisica in modalità trunk tramite comando, dopo a questa scheda vengono associate delle interfacce virtuali, in questo modo per il dispositivo è come avere due interfacce separate con l'indirizzi IP configurati nel modo opportuno (2 VLAN, 2 indirizzi IP su subnet diversi) restando compatibile con il funzionamento originario dell'IP.

Le due interfacce virtuali usano lo stesso indirizzo MAC della porta fisica però alla fine vado ad utilizzare su due interfacce diverse lo stesso indirizzo MAC , perdo univocità però non è un problema perché tanto siamo su due LAN diverse.



Assegnazione VLAN a porte TRUNK

Solitamente ci sono molte porte access e poi qualche porta trunk che collegano i vari switch. **Se devo configurare la VLAN sul link tra gli switch devo mettere le porte in trunk,** ma su quali colori?



Esiste un protocollo, **GVRP**, che capisce quali sono le VLAN configurate su rete locale e in automatico associa ai link trunk solo le VLAN necessarie.

Per semplicità si utilizza però una configurazione manuale, cioè configuro io tutte le VLAN manualmente.

VLAN e spanning tree

Come prima soluzione posso usare un'unica istanza di spanning tree, anche se ho più VLAN e assumendo che sui Link trunk abbia associato tutte le VLAN (evito problema di associazione VLAN a link trunk).

Soluzione due posso creare uno spanning tree per ogni VLAN.

Cisco non permette di avere un singolo albero, per forza più spanning tree per ogni vlan.

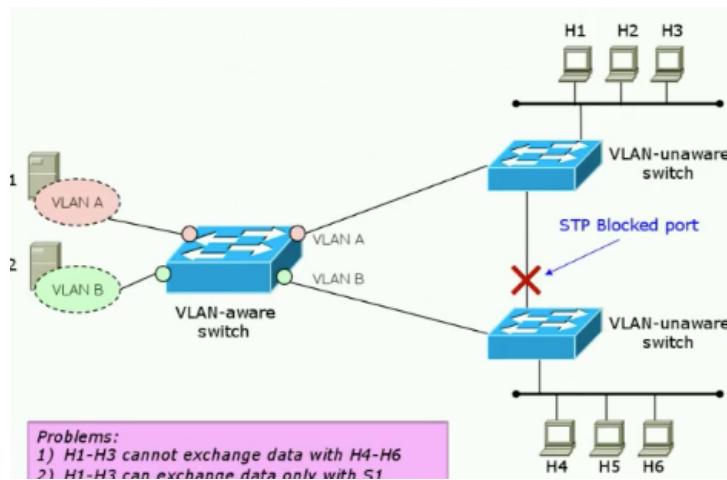
VLAN e Broadcast

Le VLAN sono solo virtualmente reti separate, i link vengono condivisi. **Se avessi problemi di broadcast storm su una VLAN, sarebbero influenzate anche le altre perché condividono i link.** Se voglio avere totale divisione ho bisogno di una divisione con router che spezza dominio di broadcast.

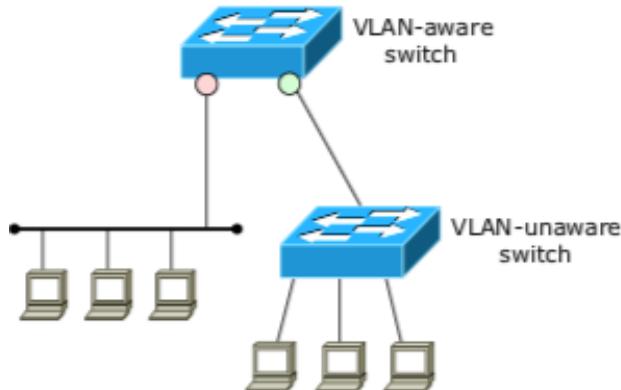
Mescolare VLAN aware / unaware switch

È frequente che in una rete locale qualcuno aggiunga un pezzo di rete, quindi posso avere più tipi di switch aware (accettano trame colorate)/unaware (solo trame bianche)

Potrebbe essere un problema collegare i vari tipi di switch perchè se lo spanning tree disabilitasse il link di collegamento unaware non potrei più far comunicare le due VLAN.



Come soluzione basta avere le VLAN unaware agli estremi della rete senza avere maglie.



Creazione VLAN cisco

Devo creare VLAN database, con tutte le VLAN che utilizzerò su quello switch. Devo entrare in privildge mode tramite ENABLE.

Si scrive VLAN DATABASE e poi si elencano con una particolare sintassi le VLAN.

Per configurare le porte digitò CONFIGURE TERMINAL, si sceglie poi un'interfaccia tramite INTERFACE.

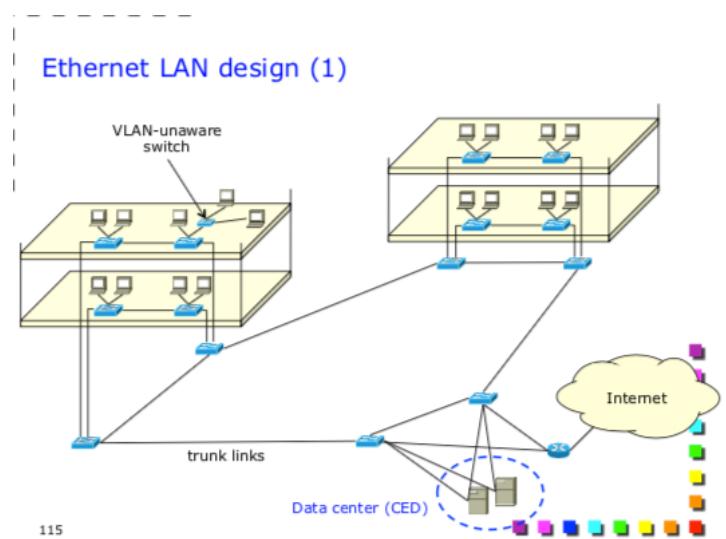
Tramite SWITCHPORT assegno una porta in una certa modalità (access/trunk) ad una VLAN.

Ehternet LAN design

Si usano solitamente più switch per edificio (porzione di rete). Se ho più piani avrò più switch per piano creando maglie con tutti gli altri switch dell'edificio. Si crea ridondanza a livello di piano e a livello di edificio.

Il data center solitamente è collegato a più swtich per garantire funzionamento in caso di rottura link.

Abbiamo poi un router, default Gateway che permette il collegamento a internet.



VOICE OVER IP

marchetto-lunedì 29 ottobre 2018

La rete circuito (telefonia a circuito) funziona in modo digitale a 64kb/s, codifica PCM64 (8000 campioni da 8 bit l'uno al secondo = 64kb/s). Canale full duplex, ognuno a 64 kb/s.

Non si applica compressione ulteriore al segnale vocale. Non c'è multiplazione statistica, banda non utilizzata non può essere utilizzata da altri. Non viene fatta soppressione delle pause, tanto ormai le risorse sono state allocate sul mio circuito. Viene sempre eseguita segnalazione per la creazione del circuito. La comunicazione non avviene quasi mai superiore a 64 kb/s e se avviene è un multiplo di 64.

Trasferimento da telefonia a circuito a rete a pacchetto IP

- maggiore compressione, risorse non usate da me usate da altri
- inviare traffico a rate maggiore
- soppressione delle pause, riduco traffico, tramite multiplazione statistica la rete può essere usata da qualcun altro.
- segnalazione diversa, è end-to-end , coinvolge solo i due nodi e non tutta la rete
- la qualità della comunicazione, a differenza di quella a circuito (telefonia) che è sempre garantita, deve essere garantita.

Per i provider di servizi il VOIP è importante perchè permette di ridurre i costi per le infrastrutture. Utilizzano VOIP non sugli end Systems ma per trasferire la telefonia normale su rete a pacchetto TOIP (telephone over IP), quindi prendo campioni derivanti dalla tecnologia a circuito e li impacco su rete IP.

Operazioni svolte

- campionamento : digitalizzo segnale analogico
- codifica : processo i campioni ottenuti. Più la codifica è complessa più tempo richiede, introduce quindi del ritardo di codifica. Solitamente non si cerca tramite codifica di aumentare l'efficienza della trasmissione perchè ci possono essere utenti che ancora usano tecnologie tradizionali (Fax), si usa quindi solitamente PCM64.

Uno dei problemi legati alla telefonia è l'eco, la voce uscente dallo speaker rientra nel microfono. Se ho un ritardo basso non ho la possibilità di accorgermene, la mia voce e l'eco quasi si sovrappongono. Se invece ritardo elevato mi accorgo dell'eco. Nella telefonia a circuito questo problema è trascurabile perchè ritardi bassi e garantiti. Sulla rete a pacchetto è molto più frequente. Sia in hardware che in software questo problema è risolvibile sopprimendo l'eco.

Impacchettamento

Per inviare flusso a 64 kb/s se immetto un solo campione per pacchetto viene fuori un flusso da 3.7 Mb/s, perchè per ogni livello (applicazione, rete, fisico) devo aggiungere l'header appropriato per ogni protocollo.

Posso quindi impacchettare più di un campione per pacchetto, quindi devo aspettare che arrivino anche gli altri campioni, aumenta l'efficienza ma aumenta anche il ritardo. Solitamente si aspetta al massimo dai 20 ai 40 ms.

Trammissione

Il problema principale è la congestione, quando ho più flussi che condividono lo stesso canale di uscita. Capita che nel buffer di uscita i pacchetti di diversi flussi vengano mischiati, e non riescano ad entrare uno dietro l'altro

AAAAA buffer
|ABABABBA|
BBBBB

Si creano quindi due buffer, in modo da dividere i pacchetti, dando precedenza ad un certo tipo di pacchetto. Questa soluzione è critica quando la mole di dati da trasmettere è in maggioranza di tipo non vocale. Inoltre quando c'è troppo traffico vocale il traffico dati deve aspettare troppo tempo —> problema STARVATION. Questa soluzione viene usata su reti proprietarie (aziende ecc).

Un altro problema legato alla trasmissione è la dimensione del pacchetto. Avendo adottato la soluzione precedente con due buffer (code), se mentre trasmetto un pacchetto a bassa priorità ne arriva uno ad altra priorità non posso smettere di trasmettere quello a bassa priorità altrimenti lo taglierei. In realtà ora questo non è un problema perchè il tempo per trasmettere un singolo pacchetto ormai è trascurabile. Come soluzione posso aumentare la banda oppure evitare di usare applicazioni bandwidth intensive durante chiamate vocali.

Marcamento

Si scrive il campo type of service dell'header IP. Questi byte vengono scritti non dall'utente.

Può essere effettuato dai router ma il problema è riconoscere i pacchetti dato che il traffico è sempre più criptografato.

Chi fa il marking quindi è il service provider tramite il router che ci viene fornito.

DeJitter

Operazione che viene svolta dal destinatario, il jitter è la variazione del ritardo. Prima di tutto devo ordinare i campioni e poi metterli alla distanza giusta (uno ogni 125 ms per PCM64). Questi moduli prendono il nome di **replayBuffer**.

Parametri sessione vocale

- ritardo la somma di tutti i ritardi introdotti dalle varie fasi tra sorgente e destinatario devono stare sotto una certa soglia convenzionale (ritardo end-to-end 0-150ms).
- banda non serve avere più banda possibile, applicazioni anaelastiche.
- perdite trasmissioni multimediali tolleranti alle perdite (5%).

RTP

Real time protocol, aggiunge servizi ad UDP. UDP non offre riordino, potrei fare in modo che il riordino sia gestito dall'applicazione multimediale, ma questo non è efficiente, per questo viene introdotto RTP.

RTP anche utile per comunicazioni multicast su rete che supporta solamente comunicazioni unicast.

RTCP è il protocollo di controllo RTP.

Il campo PT del pacchetto RTP contiene il coDec che è stato utilizzato nel payload, informazioni utile all'applicazione multimediale.

Sequence Number e timestamp utilizzati per riordino.

RTP Mixer

Componente per realizzare multicast grazie a RTP.

- Se rete sottostante non supporta multicast → utilizzo più sessioni unicast, si ha quindi più flussi per ogni utente. trasmissione n-1, ricezione n-1
- Se rete sottostante multicast → unico flusso per singolo utente. ricezione n-1, trasmissione 1
- Se rete sottostante unicast e uso RTP → RTP mixer ricevo 1, invio 1

Con RTP MIXER ogni utente invia flusso unicast a RTP MIXER, il quale decodifica e poi li somma, quindi in uscita ho la somma di ogni voce di ogni utente. Il flusso risultato non occupa più spazio, sempre PCM64 se in entrata ho PCM64.

Mixed Network

Si utilizza backbone IP, coesistendo con rete telefonica tradizionale.

Interfacciamento tramite Gateway (TOIP). Anche tra rete IP e backbone IP c'è un Gateway , usato come accesso al servizio. Per esempio può occupare di localizzare il destinatario sulla rete IP oppure della tariffazione (funzioni diverse rispetto al Gateway TOIP).

Protocolli di segnalazione

- H.323 **standardizzato da ITU, il protocollo è troppo complicato.**
- SIP **standardizzato da IETF, protocollo testuale HTTP-like. Utile per mantenere servizio all'interno della propria realtà, evitare outsourcing(Skype for business).**

SIP

mercoledì 31 ottobre 2018

Session initiation protocol, protocollo di livello applicazione (su internet). Basato su interazione client-server, utente chiede servizio a sipServer poi la chiamata in realtà è peer-2-peer.

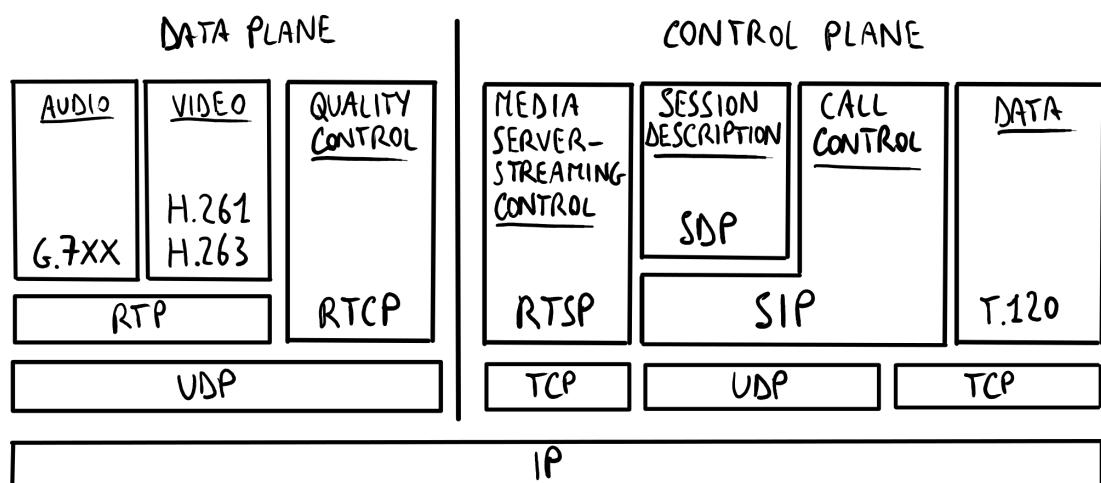
Protocollo testuale come HTTP.

SIP prevede la possibilità di usare tre modalità, TCP, TLS (versione sicura TCP con ssl), UDP.

- UDP soluzione più semplice.
- TCP può avere senso per superare firewall e nat.
- TLS per criptografare messaggi di segnalazione ma perdo le funzionalità della testualità del protocollo.

I servizi supportati da SIP vengono intesi come piano di controllo, non piano dati. Sono tutte le azioni coinvolte nel controllo del servizio, alla segnalazione.

Il SIP PROTOCOL STACK comprende sia piano dati che controllo. Il piano dati è delimitato da RTCP.



SDP – session description protocol, descrive la sessione
Dentro il pacchetto SIP c'è anche un messaggio SDP.

m= → significa media, tipo di comunicazione

m= audio 5004 RTP/AVP 0 3 → comunicazione audio, 0 o 3 identifica il coDec, 0 PCM64, 3 GSM

Se ho più linee con m=.... Vuol dire che sto avendo una comunicazione multimediale, non solo audio per esempio ma anche video.

SIP gestisce la segnalazione, definendo i messaggi che devono essere scambiati, senza dire però il formato con cui scrivere questi messaggi, uso cioè protocolli già esistenti.

Componenti SIP

-Location server	serve per gestire la localizzazione degli utenti
-register server	permette ad utenti di registrarsi al servizio, utile anche per la localizzazione tramite indexing.
-authentication server	gestisce autenticazione utenti per servizio
-MCU	opzionale, serve per fare RTP mixing quando voglio offrire servizio multicasting su rete IP.
-Media server	server per segreteria telefonica
-Redirect server	gestisce chiamate
-media proxy	fa da relay per nat traversal.
-proxy server	legato alla segnalazione, serve come aiuto per accesso al servizio. Il proxy pensa ad interagire con tutti gli altri server almeno non devo farlo io. Funzionalità interna al SIP server, quindi viene sempre eseguita quando viene contattato SIP server.

Nel mondo SIP i client si chiamano USER AGENT SIP.

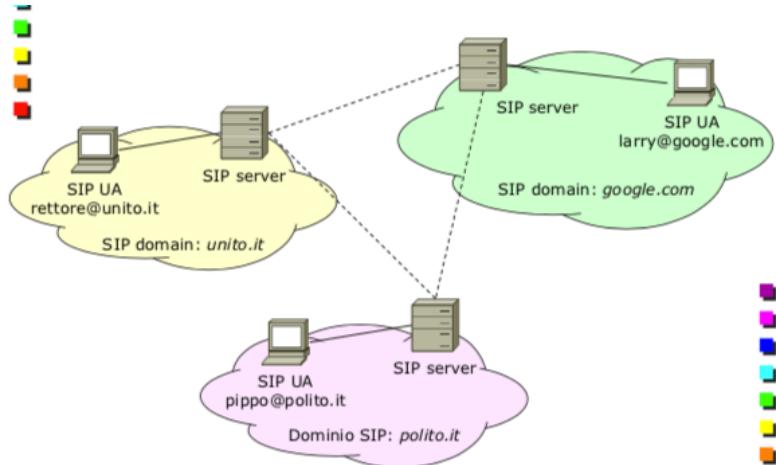
Spesso il SIP SERVER viene chiamato SIP PROXY.

SIP è basato su domini come la posta elettronica, l'unica differenza è che la mail non è real time (le mail che arrivano sul server decido io quando andarle a vedere) mentre SIP deve essere real time, i messaggi non devono essere salvati su un server ma arrivare direttamente all'utente finale. (non posso salvare una chiamata su un server, il telefono destinatario deve squillare subito).

Viene usata una architettura distribuita, l'indirizzo IP locale di ogni utente non ha impatti sulla struttura logica della rete.

Vantaggi:

- scalabilità
- facile gestione
- Privacy



Abbiamo bisogno però di interconnettere i vari domini.

Configurazione DNS

Sul DNS devo configurare queste entry:

- SRV
- NATPTR
- A/AAAA

NATPTR Come chiave fornisco il nome del dominio e come risposta avrò tutti i servizi con annessi protocolli utilizzati in quel dominio.

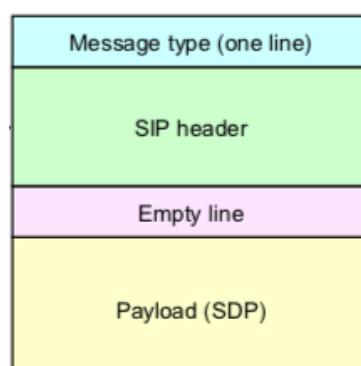
SRV Come chiave do il nome del servizio/protocollo scelto e come risposta ottengo l'alias/nome del server che implementa quelle funzionalità.

A/AAAA Come chiave do il nome del server e come risposta ottengo il suo indirizzo IP. (A per IPV4 , AAAA per IPV6)

In realtà il DNS sa che dopo la prima query NAPTR gli arriveranno le altre due, quindi solitamente risponde a tutto in un'unica risposta.

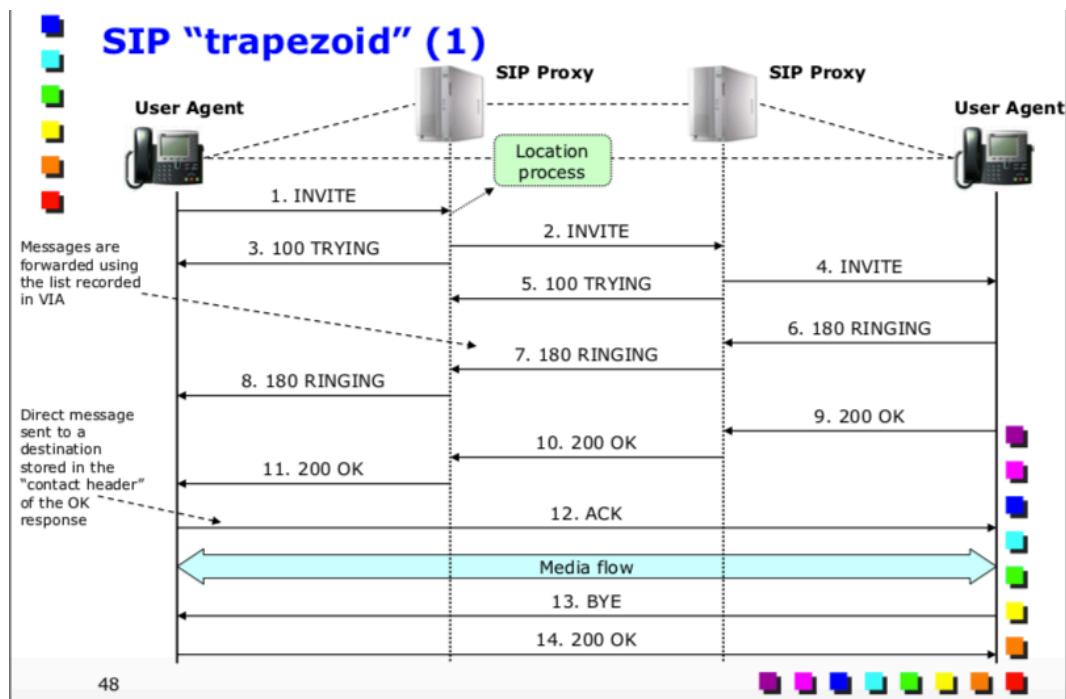
Messaggio SIP

riga dedicato per il messaggio, riga per header, riga bianca e poi un payload.



Ci sono diversi tipi di messaggi

- Register messaggio che user agent deve inviare al SIPSERVER per registrarsi.
- Invite invita utente al servizio (chiamata, conferenza, videochiamata). Nel payload bisogna includere la descrizione della sessione SDP.
- Ack a livello applicazione, serve per notificare gli utente dell'avvenuta registrazione o invio messaggi.
- Bye termina la chiamata
- Cancel termina chiamate pendenti
- Options serve per cambiare parametri durante la chiamata
- Notify
- Message serve per gestire messaggi testuali (chat)



I primi messaggi vengono sempre inviati tramite server Proxy perchè l'invio è più sicuro. Inoltre se l'alto user Agent fosse dietro un NAT tramite un messaggio diretto UserAgent-UserAgent non riuscirei ad oltrepassare il NAT.

Prima di mandare il primo Invite devo fare la registrazione al mio Sip Server tramite REGISTRAZIONE SIP.

Esempi di header:

- From indica chi sta effettuando la chiamata
- To chi riceve la chiamata
- Contact indica indirizzo IP utenti
- Via indica tutti i sistemi intermedi attraversi per far segnalazione
- Record Routing** *indica che i messaggi SIP dovranno sempre passare dal proxy.*
Utile per nat traversal e tariffazione perchè il bye deve passare dal proxy per terminare tariffazione.
- Call-ID ID univoco per INVITE o registrazione utente.
- Subject
- Content type
- Content length
- Content encoding

Codici di risposta

I codici provisional non indicano successo ma indicano che sta avvenendo qualcosa.

Ricerca di un indirizzo SIP

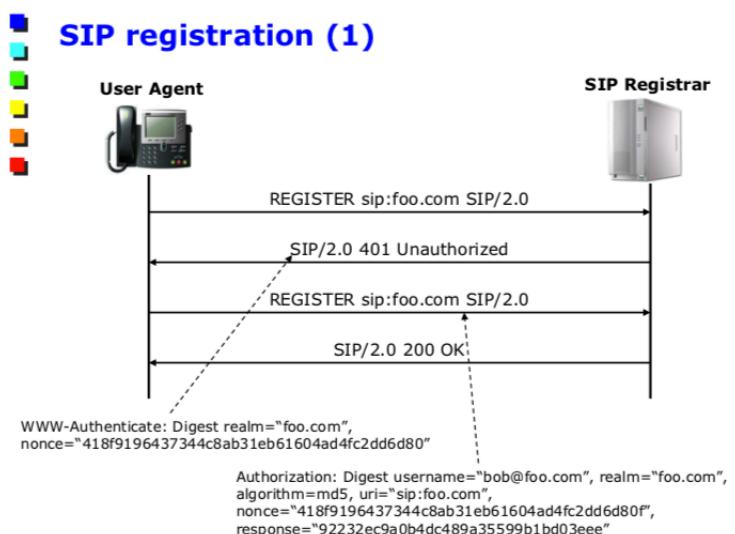
La stessa procedura è usata sia per l'INVITE per cercare un User Agent che per il REGISTER usato per cercare un registration server.

Come visto prima posso fare query:

- NAPTR → mi restituisce la lista dei protocolli usati dal SIP SERVER
- SRV → mi restituisce quale porta usare e quale priorità è associata al sip server
- A/AAAA → mi restituisce l'indirizzo SIP

Registrazione SIP

Prima di inviare Register invio richiesta al DNS per scoprire indirizzo del mio SipServer in base al nome del mio dominio. La Query al DNS si compone di 3 querys viste prima (NAPTR,SRV,A/AAAA).



Solitamente il server SIP può contenere al suo interno un SIP REGISTRAR, non sono sempre due server separati. Quindi tutti i messaggi di registrazioni posso anche inviarli al sip server.

Invio register, non metto username e password, se non sono previsti va a buon fine, se sono previsti, il REGISTER manda messaggio di errore indietro con scritto NON AUTORIZZATO includendo nel messaggio una stringa che l'utente dovrà usare per fare hashing con la sua pw e rinviarla. Dopo averla rinviata, il REGISTER applica de-hashing e vede se la password combacia. Se tutto va a buon fine viene inviato all'utente il messaggio 200 OK.

Le procedure DNS posso essere usate anche dallo user agent per trovare il REGISTER e non solo per la comunicazione tra due SIP-proxy.

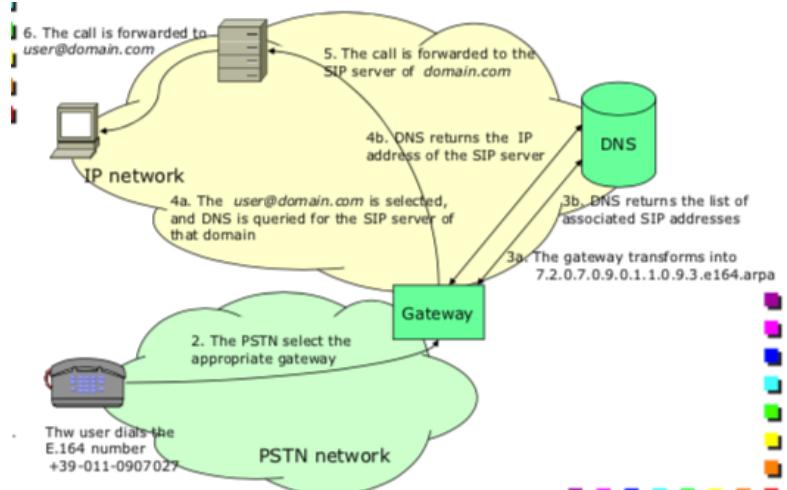
Possibili scenari di comunicazione

PSTN-IP

Si vuole permettere di utilizzare lo standard E.164, il quale permetteva l'utilizzo di numeri di telefono.

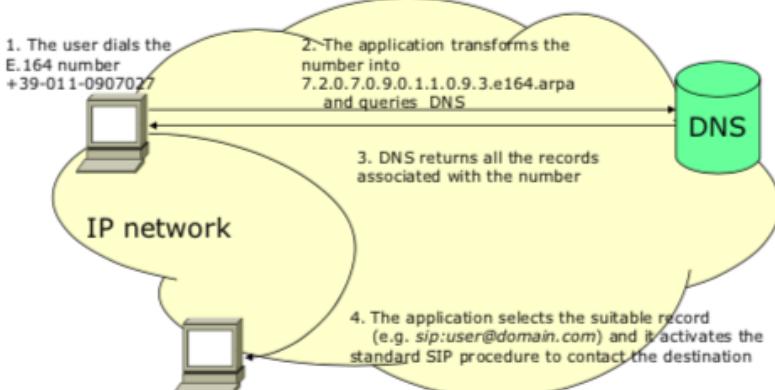
Bisogna quindi permettere la traduzione di un numero di telefono ad un dominio per poi far richiesta al DNS con il nome del dominio.

La traduzione viene effettuata dal gateway della rete PSTN.



IP-IP

Come prima ma la traduzione viene effettuata da un soft-phone o SIP proxy.



IPV6

lunedì 5 novembre 2018

Si passa ad IPV6 per avere uno spazio di indirizzamento più ampio dovuto alla maggior lunghezza degli indirizzi.

Tutte le caratteristiche di IPV6 sono presenti in IPV4 (a parte uso più efficiente delle LAN).

L'implementazione del protocollo (IPV4 o IPV6) è nel sistema operativo, quindi cambiando protocollo bisogna aggiornare il software host. Sono le applicazioni che usano gli indirizzi, cambiando gli indirizzi devo cambiare tutte le applicazioni che utilizzano la rete.

Indirizzi IP usati in modo gerarchico, tutti gli host della stessa rete fisica hanno lo stesso prefisso → spreco indirizzi, perchè il prefisso assegnato indica un tot di indirizzi molti dei quali non verranno usati.

1) Il problema della scarsità degli indirizzi in IPV4 era già stato affrontato con soluzioni tampone quali NAT con divisione degli indirizzi pubblici e privati.

Soluzioni IPV4 alla scarsità degli indirizzi:

- netmask
- indirizzi privati, NAT, ALG (application laser Gateway)

2) Oltre alla scarsità con IPV4 c'è anche problema della scalabilità, dimensione delle tabelle di routing cresciuta in modo esponenziale.

Soluzioni:

- Posso aggregare più routes in una sola.
- CIDR

Quindi il passaggio ad IPV6 è dovuto anche al problema della scalabilità.

Indirizzi

Vengono indicati con 8 cifre esadecimali separati da : in gruppi di 2 bytes (4 cifre).
Per semplificare se ogni gruppo di 2 bytes è preceduto da 0 posso non scriverli o se l'intero gruppo è fatto di 0 posso non scriverlo.

ABCD:7:EFDA:AOOC → 7 è preceduto da 3 zeri

ABCD::EFDA:AOOC → il gruppo dopo ABCD è costituito da soli 0

Indirizzamento multicast

Parte dello spazio di indirizzamento viene usato per gli indirizzi multicast (come per IPV4 ma in IPV4 prima sono stati assegnati gli indirizzi per gli host e poi quelli multicast, qui il contrario). *Gli indirizzi che iniziano con 1111:1111 sono multicast, cioè in esadecimale inizia con FF.*

FF00::/8 → devo sempre scrivere come minimo un blocco di 4 cifre, metto poi :: per indicare che gli altri sono 0. Lo / indica il numero di bit a cui sto facendo riferimento (quelli non a 0).

Indirizzamento host

I principi di routing di IPV4 sono gli stessi per IPV6.

In IPV6 in base al tipo di indirizzo della destinazione so quanto è lungo il prefisso della destinazione. Guardo il numero dopo lo /. (su IPV4 utilizzavo la subnet mask)

Ogni indirizzo ha n bit di prefisso e 128-n di identfier.

Prefisso n bit	Interface Identifier (128 -n)
128 TOTALI	

Indirizzi locali

- link local $1111\ 1110\ 1000 \rightarrow \text{FE80::}/64$
prefiso FE80 poi 96 zeri
 - site local $1111\ 1110\ 1100 \rightarrow \text{FEC0..}/10$
 $\text{FE}[\text{C-F}]$
 $/10 \rightarrow \text{ci sono più prefissi}$

Ogni host quindi ha un indirizzo link local per cominciare con stazione su stesso link e uno site local per comunicare anche con stazioni a diversi link.

Indirizzi privati

- Unique local addresses $1111\ 1100 \rightarrow FC00::/7$
 - privati (sottoinsieme di Unique) $1111\ 1101 \rightarrow FD00::/8$

[Negli indirizzi privati i primi 8 fissi](#), i 40 generati random, e scelgo come voglio solo gli ultimi 16.

Fisso	Random	subnetID	InterfaceID
FD			
8	40	16	64
ORGANIZZAZIONE		INTERFAZCE	

Indirizzi Global Unicast

Identificano in modo univoco su tutta la rete. Servono per interpretare indirizzi IPv4.

- IPv4 interoperability addresses 0::/80
80 bit iniziali a 0.
- IPv4 mapped addresses 0:0:0:0:FFFF::/96
80 a 0, 16 ad 1 poi indirizzo IPV4.
- IPv4 compatible 0:0:0:0:0:A00:1 (estesa)
::A00:1 (forma compatta)
::10.0.0.1 (forma speciale)
80 a 0, 16 a 0 e poi indirizzo IPV4.

Indirizzi IPV6 - aggregable global unicast

Usati per indicare stazioni IPV6.

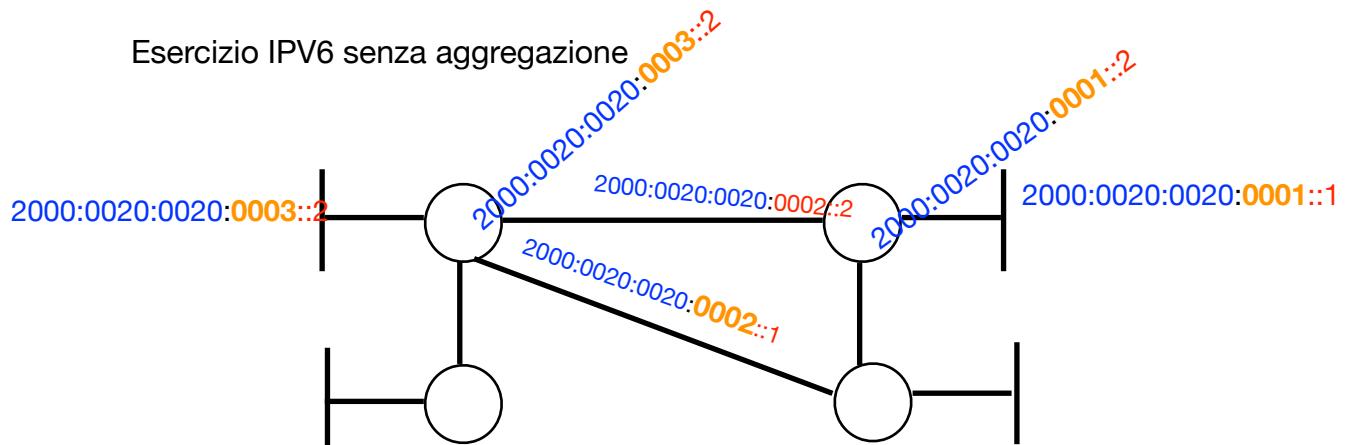
Iniziano con [2-3].

Per aggredarli dobbiamo fare un'assegnazione basata sulla topologia.

Per ottenere più flessibilità;

3 bit	Global routing prefix	Subnet ID	Interface ID
Indicano global unicast	Indirizzi assegnati al cliente	Sotto-indirizzi scelti dal cliente	<ul style="list-style-type: none"> • Contiene anche indirizzo MAC per assicurarsi che ind. cliente univoco. • INTERFACE-ID
GLOBAL UNICAST S.P. + POLI	<u>48 bit organizzazione</u>	UFFICI	
ORGANIZZAZIONE	16 BIT	LAB	64 BIT

Esercizio IPV6 senza aggregazione



Ogni singolo collegamento è una sottorete.
Le sottoreti hanno diverso SUBNET_ID (1, 2, 3).

Ci sono indirizzi speciali che non possono essere usati:

- loopback
- multicast a tutti i nodi FF02::1
- multicast a tutti i router FF02::2
- unspecified Address

IPV6 cerca di mantenere la caratteristica di PLUG&PLAY, avendo due modalità di autoconfigurazione:

- stateless non c'è bisogno di un sever, comunicazione solo tra computer della stessa rete.
- statefull basato su uso di un server DHCP (stesso di IPV4)

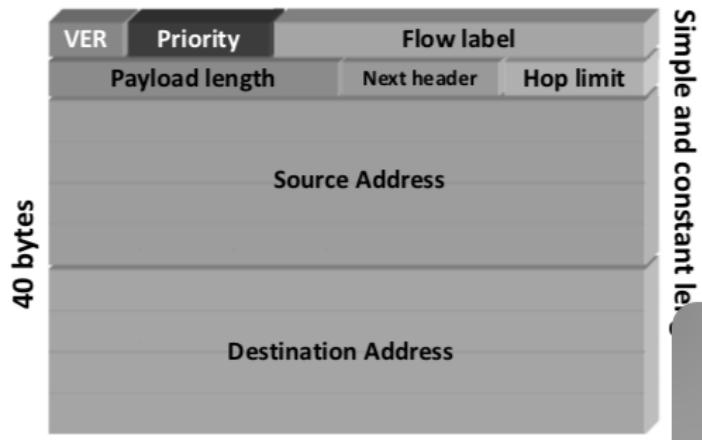
MODIFIED PROTOCOLS

Cambiano i protocolli:

- IP
- ICMP cambiano formato e funzionalità
- ARP viene integrato in ICMP, serve per trovare MAC conoscendo IP destinazione
- IGMP protocollo per multicast, integrato in ICMP
- DNS non cambiano le funzionalità ma dobbiamo estendere i messaggi,
- RIP e OSPF cambia il modo in cui identifichiamo la destinazione ma il routing sarà sempre fatto nello stesso modo, non cambia la logica del protocollo ma il formato e le strutture dati.
- BGP e IDRP
- TCP e UDP
- SOCKET INTERFACE interfaccia usata per scrivere programmi di rete.
Cambia perchè per aprire connessione devo specificare l'indirizzo che è cambiato.

PACKET HEADER FORMAT

La prima cosa che cambia è che in IPV6 ci sono molti meno campi, ma questo non vuol dire che sia più piccola, anzi gli indirizzi IP sono diventati 4 volte più grandi con IPV6.



Campi tolti rispetto a Header IPV4:

- | | |
|-----------------|---|
| CHECKSUM | viene tolto perchè superfluo |
| FRAGMENT OFFSET | Viene tolta anche la gestione della frammentazione perchè in IPV6 si cerca di evitare frammentazione definendo una dimensione massima dei pacchetti utilizzabili lungo un link. |

Campi Header IPV6:

- **FLOW LABEL** è un campo di 20 bit che identifica ogni flusso con un identificativo univoco. È univoco solo per quel particolare host, non globalmente.
- **VER** indica versiones IPV4 o IPV6
- **PRIORITY** è il vecchio type of service. Associa pacchetto ad una classe ma non dice come trattare quella classe, lo deciderà il router.
- **PAYLOAD LENGTH** lunghezza contenuto, non comprende anche la lunghezza dell'header perchè tanto è fissa.
- **HOP LIMIT** come vecchio timeToLive.
- **NEXT HEADER** intestazione modulare, il prossimo header potrebbe essere anche un altro header IPV6. Non è detto che ci sia un protocollo di livello superiore (come era in IPV4).
EXTENSION HEADER.

Gli extension Header non sono obbligatori, vengono aggiunti solo quando servono.

EXTENSION HEADER:

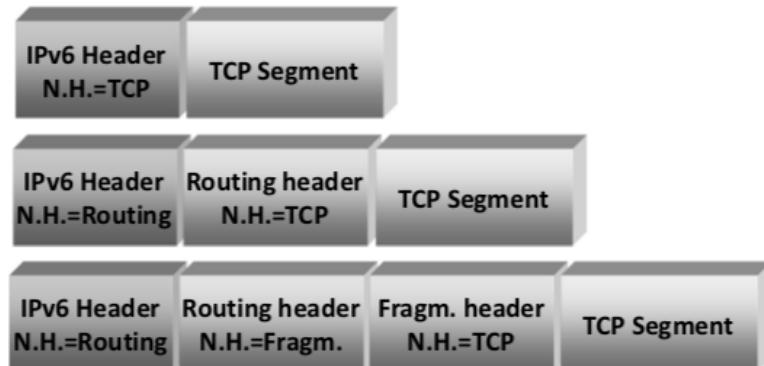
Esistono diversi tipi di Extension Header:

- HopByHop Option viene elaborato da tutti i router.
- **A) RoutingHeader** serve per source routing, elaborato tutti i router.
- Fragment serve per frammentazione, elaborato tutti i router
- Authentication serve per mettere in sicurezza il pacchetto, elaborato solo da end host, destinazione.
- Encrypted security elaborate end router payload
- Destination option elaborate end router



I primi byte del next header mi dicono se il prossimo header è un header esteso o no.

Gli header estesi devono essere allineati a 64 bit o multipli. Aggiungo bit di padding per arrivare a 64 o multipli, in modo che il prossimo estensione header cominci ad un multiplo di 64.



Formato TLV

Formato che viene usato negli HopByHop e Destination Options extension header.

type-length-value

Type	Length	Value
Identifica opzione	lunghezza valore effettivo opzione	contenuto effettivo del opzione

Se un router vede arrivare un pacchetto con un opzione che non conosce, prende la lunghezza e salta quei bit in modo da passare alla prossima opzione.

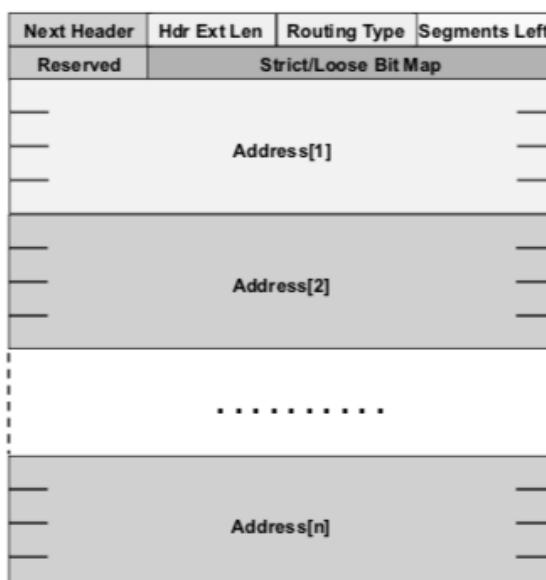
Se la conosce la elabora.

lunedì 12 novembre 2018

A) Routing Header

Usato da un host IPV6 per avere la lista dei nodi da attraversi per arrivare alla destinazioni voluta.

Inizia con next header, length , routing type (indica tipo di routing header), segment laft (ogni router rappresenta segment address[n], segment laft indica da quanti router devo ancora passare.), reserved (non utilizzati, per allineamento), bitMap (ci dice se l'hop è strict(router successivi sono adiacenti) o loose (tra due hop elencati ci potrebbero essere router intermedi). Non si mette come destinazione la destinazione reale ma il next hop. La destinazione effettiva viene indicata nei segmentAddress. Il next hop riconosce se il pacchetto è per lui o no guardando segmentLaft. Ogni volta che nodo legge pacchetto scrive il suo ID al posto segmentAddress successivo.



S	R1	R2	D
TO R1	TO R2	TO D	
SegmentAddress:	SegmentAddress:	SegmentAddress:	
R2	R1	R2	
D	D	D	
SegmetLaft=2	SegmentLaft=1	SegmentLaft=0	
		Arriverà a D con SegmentLaft=0 così capirà che è per lui.	

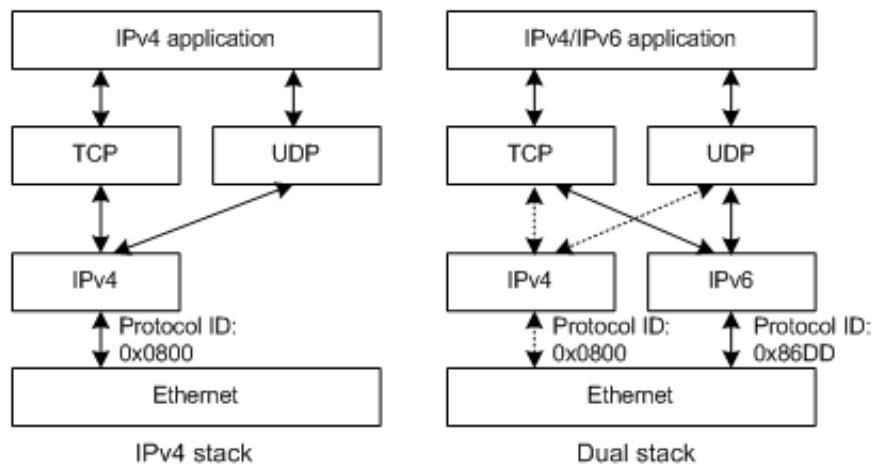
INTERFACE WITH LOWER LEVEL - DS LITE

Pacchetto IP imbustato in trama di livello 2, se trama ethernet si usa un valore diverso rispetto a quello di IPV4 in EtherType.

Approccio **dualStack**, a livello 3 abbiamo due protocolli diversi IPV4 e IPV6, e da lì nascono due stack protocollari diversi.

Si utilizza questo approccio in modo da non modificare il funzionamento di IPV4, non modifco il software IPV4 per utilizzare anche IPV6.

Per scoprire indirizzo MAC destinatario uso ICMP (IPV6) con neighbor discovery. (in IPV4 si usava ARP)



IP MULTICASTING IPV4

Si usa un indirizzo specifico per il multicasting. I router non sanno quanti saranno i membri del gruppo che riceveranno il multicast.

Non è ben accetto perchè rende difficili meccanismi di Policing effettuati sul bordo della rete, viene solitamente reso attivo solo su porzioni di rete ben controllate.

Il gruppo multicast viene identificato con indirizzi che iniziano con 1110, indirizzi di classe D.

Classi A B C venivano assegnati alle stazioni, classe D solo ai gruppi.

224.0.0.0-239.255.255.255

Per inoltrare i pacchetti su rete locale delego la consegna di gruppo a livello 2.

Le reti ethernet prevedono un meccanismo di comunicazione di gruppo a livello 2, dove la trama ethernet viene mandata ad un indirizzo di gruppo e uno switch lo fa arrivare a tutte le schede di rete che hanno aderito a quel gruppo.

Gli indirizzi MAC di gruppo hanno il primo byte trasmesso dispari. Per prima cosa viene trasmessa indirizzo destinazione e di ogni byte si comincia a trasmettere dal LSB (LittleEndian) (00000001 sarà trasmesso come 10000000, dispari)

DX 00000001	SX	Ethertype
----------------	----	-----------

Primo campo trasmesso

01-00-5E-0 Prefisso per indirizzi multicast IPV4.

gli ultimi 23 bit hanno lo stesso valore dei 23 bit dell'indirizzo IP che sto incapsulando.

(Per IPV6 si usa 33:33)

Ogni host deve configurare la propria scheda per ricevere pacchetti inviati con un certo indirizzo MACmulticast.

IGMP protocollo utilizzato per mandare messaggio ai vari router che si trovano sulla sua rete per avvisare a quale gruppo multicast sono interessato. In IPV6 si è passato a ICMP.

Si effettuano quindi operazioni di routingMulticast tramite utilizzo di MulticastRoutingProtocol.

Protocolli di routing multicast

- varianti protocolli esistenti unicast OSPF (5 tipi diversi di LinkStateAdvertisement LSA, uno di questi serve per dire che sono collegato ad un certo gruppo multicast)
- DVMRP distanceVectorMulticastRoutingProtocol
- PIM protocolIndependentMulticast

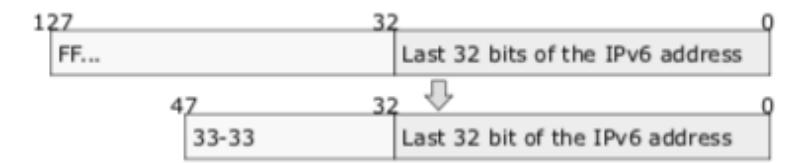
IP MULTICASTING IPV6

mercoledì 14 novembre 2018

Imbustamento in trama ethernet di indirizzi IPV6 Multicast

Cambia l'indirizzo MAC che si costruisce.

Primi due Byte dispari ad identificare che è multicast e nella restante parte abbiamo 32 bit meno significativi dell'indirizzo IPV6 MULTICAST.



Esempio:

Indirizzo IPV6 multicast FFOC::89:**AABB:CCDD**
Multicast **33:33:AA:BB:CC:DD**

Tutti i dispositivi che terminano quindi con **AABB:CCDD** riceveranno il pacchetto inviato in multicast.

Neighbour discovery

Meccanismo che si basa su multicast, equivalente ARP in IPV4.

I vicini possono essere host e router sulla stessa rete locale.

Se voglio scoprire informazioni di un vicino che ha un certo indirizzo IP (quindi devo conoscere l'indirizzo di IP di chi richiedo informazioni) mando una richiesta neighbor solicitation ad un gruppo multicast IP il cui indirizzo lo ricavo dall'indirizzo IP dell'host da cui voglio scoprire informazioni.

Indirizzo multicast usato si chiama **SOLICITED NODE MULTICAST ADDRESS** (SNMA), ogni host si iscrive ad un SNMA che è specifico per il suo indirizzo.

L'indirizzo SNMA non è uguale a un qualsiasi indirizzo Multicast IPV6 perché ha validità solo all'interno della mia rete locale , si usa per neighbor discovery. Gli indirizzi Multicast IPV6 invece hanno validità ovunque.

SNMA

FF02::1:FF?????????

104 bit fissi 24 bit meno significativi dell'indirizzo IP dell'host che voglio scoprire.

Esempio) Trovare MAC di 2001::ABCD:EF98

Trovo indirizzo SNMA →

FF02::1:FF CD:EF98

fissi presi da indirizzo

Ottengo così un indirizzo multicast a cui inviare richiesta per scoprire info.

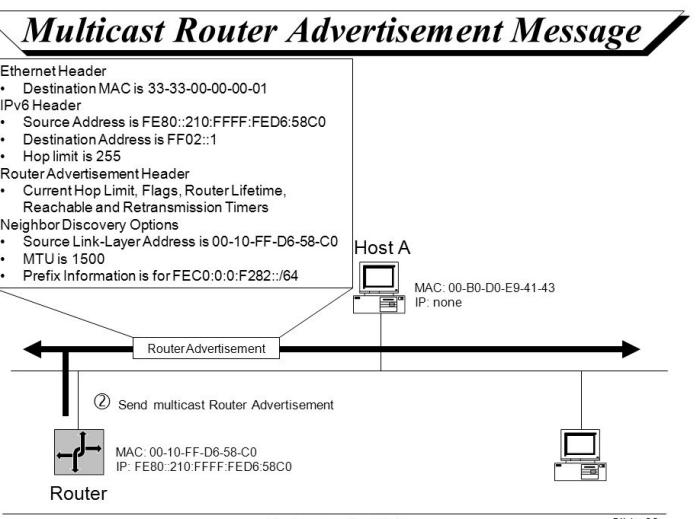
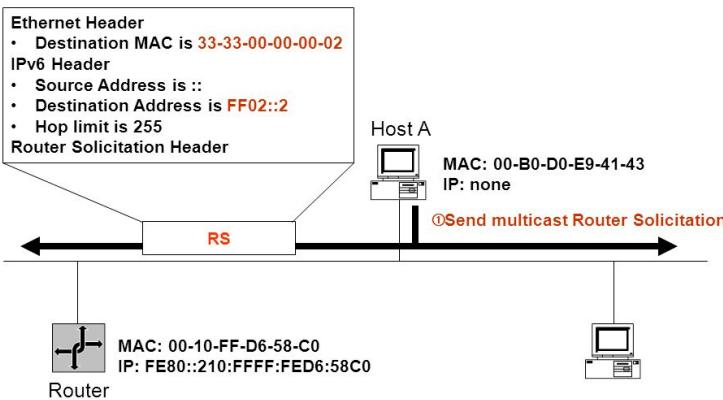
Incapsulo in **trama ethernet** → 33:33:FF:CD:EF:98

*I MAC POSSONO ESSERE RICONOSCIUTI SOLO TRA HOST SU STESSA RETE LOCALE, NON VIENE INOLTRATO DAI ROUTER. Esiste anche un messaggio di **RouterSolicitation** e **RouterAdvertisement** per scoprire quali router ci sono sulla rete.*

Il messaggio di risposta è il neighborAdvertisement mandato all'indirizzo di quello che ha fatto la richiesta. Quindi richiesta in multicast risposta in unicast.

Router Sollicitation e Advertisment

Multicast RS Message



ICMPv6

Utile per:

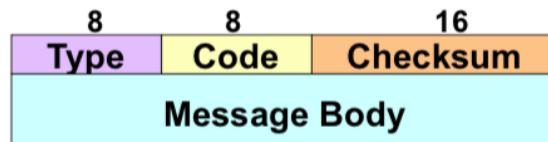
- Diagnostica ping tramite echo request e echo reply. Uso ping invece che per esempio richiesta HTTP su server web perchè magari se la connessione TCP di livello 4 non funziona mi darebbe che l'host è in down. Oppure anche il firewall potrebbe dare problemi.
- Neighbor Discovery Tramite protocollo avverte router che voglio ricevere pacchetti di un certo gruppo.
- Gestione gruppi Multicast
- Notificare problemi notificare agli host gli errori che sono avvenuti nella rete, esempio scarto di pacchetto.

I messaggi ICMP non vengono inviati frequentemente perché rivelano identità dei router, informazioni utili per possibili attacchi alla rete. Inoltre non è detto che l'host che riceve messaggio ICMP lo elabori.

Formato messaggio

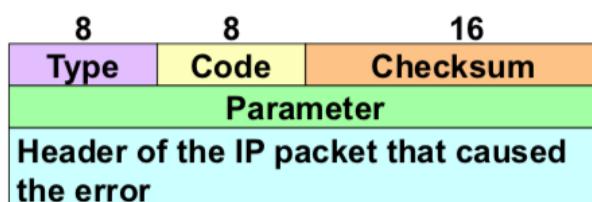
Messaggio inserito dentro pacchetto IP. Mai più lungo di 576 byte così la probabilità di frammentazione è molto bassa.

Type	indica che tipo di messaggio è, sollicitation, discovery, advertisement
Code	sorta di sottotipo, diversi formati per ogni tipo.
CheckSum	
Body	diverso a seconda del tipo



Messaggi Errore

Parameter	
Header IP	Header del pacchetto che ha creato errore



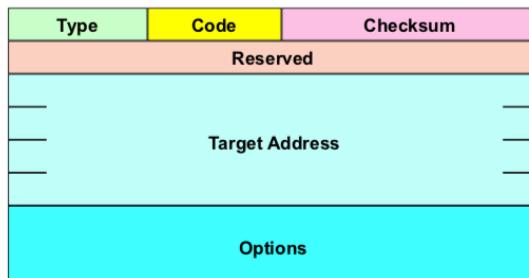
Echo message

Sequence	Number utile per frammentazione
Data	irrilevanti, servono solo per fare pacchetto più grande o più piccolo



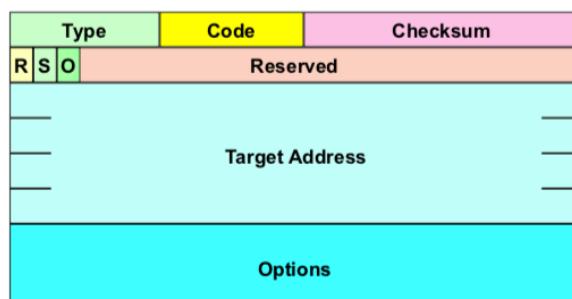
Neighbor solicitation

Reserved	32 bit, per allineamento a 64
TargetAddress	indirizzo del vicino di cui sto cercando di scoprire informazioni
Options	chiedere informazioni particolari



Neighbor advertisement

RSQ	S indica se è una risposta ad una solicitation
Reserved	
TargetAddress	nella risposta si ripete indirizzo che si cercava di risolvere
Options	



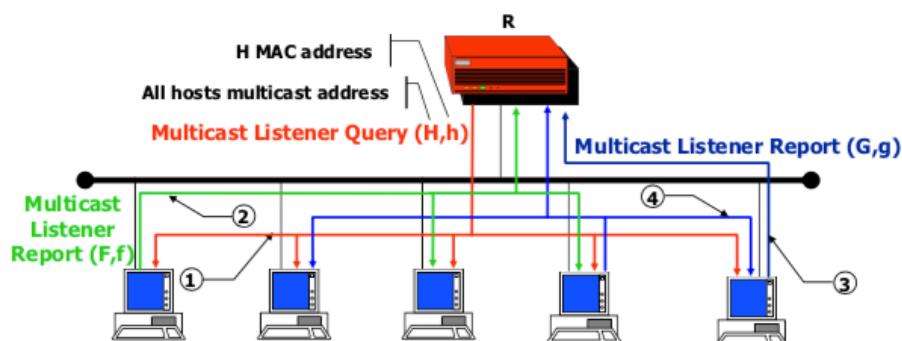
Host member discovery

Come un host indica che è interessato a messaggi di un certo gruppo multicast.

- Multicast listener query

Il router periodicamente deve scoprire chi c'è su tutte le reti locali a cui lui è collegato e chi è interessato a quali gruppi. Manda **MULTICAST LISTENER QUERY** a indirizzo multicast di tutti gli host in ogni rete locale. Questi messaggi vanno a tutti gli host delle reti locali a cui router collegato. Gli host che sono interessati ad un certo gruppo rispondono con **MULTICAST LISTENER REPORT** indicando a quale gruppo multicast sono interessati. Non lo dicono direttamente al router inviandogli al pacchetto senno si creerebbe congestione al router se lo fanno tutti. Invia quindi il messaggio in multicast all'indirizzo del gruppo multicast in questione. Tutti i membri del gruppo multicast vedono il messaggio e sanno che qualcuno ha già risposto. Arriverà anche al router, il quale è iscritto a tutti i gruppi, ma ne riceverà solo uno. In realtà si aspetta che almeno altri due/tre host rispondano ancora.

Come si vede dalla foto tutti gli host interessati al gruppo Multicast Rosso inviano un solo Report al router e non 5 diversi.



Meccanismo di SOFT STATE, la conoscenza di ogni router dopo un po scade.

Boot macchina

Solitamente quando una macchina esegue il boot, dopo essersi generata indirizzo link local

- 1) Dopo essersi creato indirizzo verifica univocità tramite Neighbor sollicitation
- 2) Manda group membership report al proprio SNMA
- 3) Manda routerAdvertisement per ottenere indirizzo DefaultGateway
- 4) Dal router ottiene prefisso per costruirsi indirizzo global
- 5) Ripete DAD per indirizzo global.

IPV6 DEVICE CONFIGURATION

lunedì 19 novembre 2018

Queste informazioni servono per ogni interfaccia che hanno. **Alcune indispensabili.**

- prefisso
- **1 identificativo interfaccia**
- default gateway da usare per raggiungere le destinazioni che non sono onLink.

Fino a qui per inoltro pacchetti IP. Da qui in poi funzionalità aggiuntive, non obbligato

- server DNS
- Hostname
- Domain Name
- MTU dimensione massima pacchetti

Se host ha MTU più grande e invia pacchetti tramite MTU più piccole, dovranno frammentare il pacchetto. Si cerca di evitare la frammentazione, ma l'host deve sapere qual'è la MTU su tutta la rete in moto che il pacchetto non venga frammentato.

Come passare questi parametri:

- **Configurazione Manuale**
- **1 Statefull configurartion**, fa uso di server DHCP.
- **2 Stateless Configuration**, non ho bisogno di server, l'host si autogenera un interfaceID. Per il prefisso può
 - generarlo lui (solo link local)
 - chiederlo al router (anche global)
- **Hybrid**, ibrido tra stateful e stateless, configrazione Stateless ma uso DHCP per ottenere server DNS, hostname, Domain name ecc.

2 Stateless - generazione InterfaceID

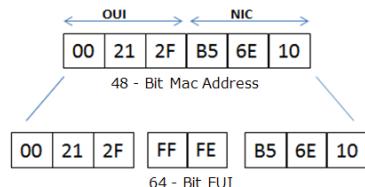
Con configrazione Stateless si può generare in due modi:

- **A** utilizzando proprio indirizzo MAC su 64 bit (invece che su 48)
- **B** Privacy aware, ogni volta che host genera interfaceID ne genera uno diverso in modo da perdere la rintracciabilità.

A Generazione interfacelD tramite MAC

Se interfaccia non ha indirizzo MAC su 64 devo costruirlo da quello di 48.

Prendo i 3 byte più significativi di quello vecchio (24 bit), in mezzo metto **FFFF** (16 bit), aggiungo ultimi tre byte di quello vecchio (48 bit).



OUI cccccc0gccccccccc	Manufacturer selected		
OUI cccccc1gccccccccc	0xFF	0xFE	Manufacturer selected

cccccc1g viene trasmesso come g1cccccc

Dell'indirizzo ottenuto cambio anche OUI cambiando il valore del 7mo bit, cioè il secondo bit del primo byte che viene trasmesso. Il bit 8 ha il significato di identificare se indirizzo multicast o se è associato a una singola scheda. Il bit 7 dice se questo indirizzo è universale (globale) o locale.

Svantaggio : **il mio host diventa tracciabile, ogni volta che genera indirizzo avrà sempre lo stesso InterfacelD. Cambia prefisso ma l'interfacelD rimane sempre lo stesso.**

Posso generare InterfacelD Random rendendo più difficile il lavoro di tracciabilità, anche se più difficile posso comunque essere tracciato.

Infine si è passati al PrivacyAware.

B Generazione InterfacelD tramite Privacy Aware

Quando host genera **InterfacelD**, se si era già precedentemente calcolato un interfacelD e se dotato di memoria, lo salva e ci aggiunge l'intefacelD che lui genererebbe con l'interfacelD generato tramite MAC (metodo precedente). Se non ha memoria (sensori) invece che salvare l'interfacelD vecchio, lo genera random.

Previous interfaceID 64 bit	InterfaceID_MAC generated or Random 64 bit
--------------------------------	---

Applico MD5

64 bit	RISUL	TATO	64 bit
--------	-------	------	--------

AND (prendo solo 64 bit alti)
11111101 
INTERFACE_ID (64 bit)

MD5 è un algoritmo crittografato che prende metà della somma dei due interfaceID e ne genera uno unico da cui è molto difficile risalire ai due parametri.

Algoritmo

Prende interfaceID ci appende InterfaceID_MAC, applica MD5, prende 64 bit più significativi e li mescola → Setta 7mo bit a 0 tramite AND perché non è detto che quell'interfaceID sia univoco dopo tutte le operazioni di manipolazione che ho fatto. Settandolo a 0 indica che ha validità locale (vedere interfaceID generato da MAC).

Address usage

Quando host ha tanti indirizzi quali usa?

Se gli arriva pacchetto deve usare lo stesso indirizzo a cui è stato inviato il pacchetto.

In altri casi ci sono due modi:

- Default
- Configurabile dall'utente o dall'applicazione (tramite priorità scritta nel codice)

Address prefix

Dove prende prefisso host?

- manualmente
- DHCPv6
- Generato automaticamente (solo LinkLocal)
- Ottenuto da un router tramite messaggi ICMP di routerAdvertisement/Solicitation. (anche global)

Non posso usare neighbour discovery per router perché i router hanno informazioni particolari come indirizzo MAC, prefissi utilizzabili sui link , MTU. L'host richiede un routerAdvertisement quando non ha una configurazione, ma non sapendo quali sono i router sulla rete invia ad indirizzo MULTICAST di tutti i router (FF02::2).

RouterSolicitation

Type	Code	Checksum
Reserved		
Options		

Sent to the all-routers multicast address
(FF01::2)

RouterAdvertisement

Type (134)	Code (0)	Checksum
Cur Hop Limit	M	O Reserved
Router Lifetime		
Reachable Time		
Retrans Timer		
Options		

ICMP-REDIRECT

ICMP REDIRECT si usa quando router deve dire ad un host che è meglio che usi un altro router come gateway per mandare pacchetti alla destinazione. Server per identificare il miglior router sullo stesso Link, sulla stessa rete locale.

Type	Code	Checksum
Reserved		
Target Address		
Destination Address		
Options		

DAD-Duplicate address detection

DAD, se voglio assicurarmi che il mio indirizzo IPV6 sia univoco uso neighbour discovery e vedo chi ha lo stesso indirizzo tramite neighbour solicitation al SNMA. Se dopo 1 secondo nessuno mi ha risposto allora il mio indirizzo è univoco, devo quindi iscrivermi al SNMA e mandare ICMP multicast listener report.

2 Stateless - prefisso senza router

Si genera in modo autonomo indirizzo link local. Verifica univocità indirizzo tramite DAD e si iscrive al SNMA e poi manda ICMP muticastListenerReport.

2 Stateless -prefisso con router

Manda routerSolicitation per scoprire prefisso, aspetta la risposta del routerAdvertisement e si costruisce indirizzo tramite prefisso inviato dal router. Verifica univocità indirizzo tramite DAD e si iscrive al SNMA e poi manda ICMP muticastListenerReport.

Stateless configuration Renumbering

Stateless permette anche di cambiare indirizzo host. Host ascolta sempre routerAdvertisement e se router gli invia informazione diversa (prefisso) loro si ricalcolano indirizzo. Lo stato dell'indirizzo in uso può essere

- Preferred In utilizzo
- Deprecated potrei ancora usarlo ma non dovrei.

1-Statefull configuration DHCP

Per ricevere una configurazione uso 4 messaggi (solicit, advertise, request, reply). Il DHCP nonostante venga fatto girare su un router non è una funzione di un router ma di un server DHCP.

IL ROUTER NON FA DHCP.

Release	rilascia configurazione attuale
Reconfigure	nuova configurazione

Scope

Quando host/router ha più interfacce in IPV4 ha un prefisso diverso su ogni interfaccia, guarda indirizzo destinazione, vede prefisso e vede se combacia con una delle sua interfacce.

In IPV6 anche ma su qualsiasi interfaccia ho un prefisso Link Local uguale per tutte.

Come faccio a capire quale interfaccia usare tramite prefixMatching?

Basterebbe che si tenesse traccia dell'interfaccia che viene usata per inviare pacchetto ma si creerebbe dipende tra livelli diversi.

Si opta quindi per un'altra soluzione, negli host gli indirizzi IPV6 invece che essere memorizzati su 16 byte (128 bit) vengono salvati su 17 byte.

Il 17esimo byte è lo scope, il numero dell'interfaccia di cui si voleva tenere traccia. Quando il pacchetto passa da livello al livello 3, si appende al pacchetto IP lo scope. Quando il livello 3 deve inviare pacchetto legge lo scope per capire su quale interfaccia inoltrarlo.

Un indirizzo con scope lo si conosce perchè è composto da un indirizzo IPV6 a cui viene aggiunto %[InterfaceNumber].

Esempio FE80::0237:00FF:FE02:a7FD%19.

ROUTING IPV6

Il routing si compone di più fasi:

- OnTheFlyRouting usa la routing table
- ProactiveRouting crea la routing table

Solitamente il routing IPV6 è disabilitato all'interno del router.

Per abilitare IPV6 su router Cisco:

```
Router#configure terminal  
Router(config)#ipv6 unicast-routing
```

La routing table (routing on the Fly) è basata su longest prefix matching (come in IPV4) ed è separata da quella IPv4. Avrò quindi due tabelle di routing, una per IPV4 e una per IPV6.

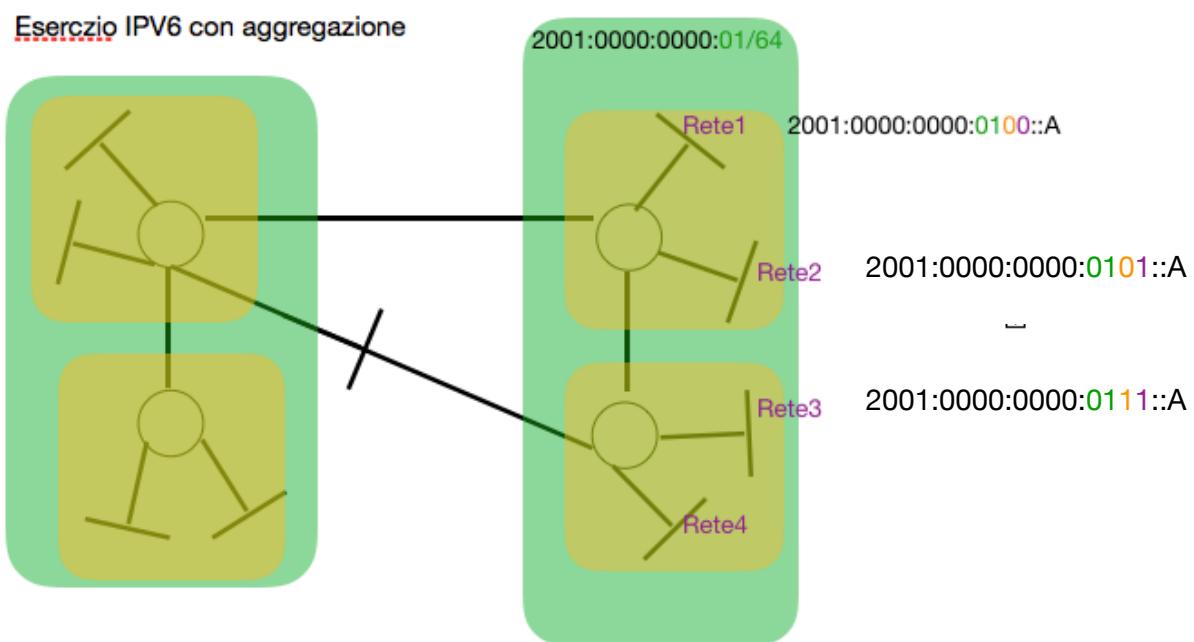
Protocolli di routing

Possono essere classificati in

- integrated routing Quando abbiamo rete che è fatta di nodi IPV6, o anche mista IPV4/IPV6, possiamo usare un singolo protocollo per scoprire topologia e poi identificare le destinazioni con gli appositi indirizzi IPV4 o IPV6 o entrambi.
 - Evito di costruire due grafi due volte, uno per IPV4 e uno per IPV6.
 - Si ha una nuova versione del protocollo, nuova implementazione può causare bachi/ malfunzionamenti.
 - IPV4 e IPV6 possono avere topologie diverse, il grafo che ottengo che le descrive entrambe non è dettagliato.
- ships in the night Protocollo solo per IPV6, i router si costruiscono una mappa della rete per IPV6 e una per IPV4.
 - Si possono usare diversi protocolli di routing
 - Migrazione più semplice
 - Trouble shooting più semplice
 - I meccanismi li eseguo due volte, una per IPV4 e una per IPV6.

Protocolli:

- Static ship
- RIPng ship
- EIGRP ship
- OSPV-v3 ship/integrated
- IS-IS integrated
- MP-BGP ship/integrated



Voglio aggregare sia come livello giallo che come livello verde. Aggregando come livello giallo ottengo il vantaggio di raggiungere per esempio entrambe la porzione di rete contenente rete1 e rete2 con un solo indirizzo.

Aggregare significa che oltre al prefisso di rete, creo un sottoprefisso che in comune a più reti.

2001:0000:0000:**0100** global routing prefix, accesso intera rete
64 bit 2 fisso identifica global routing
 identifica le posizioni di rete verde
 uso per distinguere le reti all'interno della porzione
 gialla
 uso per rete locale

Ho ancora 64 bit per distinguere i vari apparati di rete in ogni sottorete.

TRANSITION TO IPV6

Transizione da IPV4 a IPV6, come viene gestito il passaggio dalla situazione attuale a quella futura, come possiamo fare coesistere i due protocolli nella stessa rete. Due aspetti diversi:

- permettere in una rete con stazioni IPV4 e IPV6, alle stazioni IPV4 di comunicare tra di loro nonostante siamo in una rete mista e viceversa.
- permettere a stazione IPV4 di comunicare con stazioni IPV6 e viceversa.

I protocolli sono diversi ma il livello trasporto e superiori sono uguali.

La transizione deve essere

- **incrementale**
- **se voglio passare ad IPV6 non devo perdere l'uso delle applicazioni che fanno uso di IPV4.**
- **non ci devono essere errori**

Devo farli coesistere all'interno dello stesso host come due protocolli diversi, due stack protocolli diversi → DUAL STACK.

Il problema è che per mantenere IPV4 ogni stazione deve avere anche indirizzo IPV4 ma non ho più indirizzi IPV4. Vorrà dire che dovrò anche andare a mettere stazioni che saranno solo IPV6. Quelle che avranno IPV4 e IPV6 saranno quelle già esistenti in IPV4.

Possibili soluzioni

- Address Mapping
- Tunneling
- Translation Mechanism

Isolated IPV6 network

La transazione è iniziata con delle isole IPV6. Cioè in una rete IPV4 inizio a creare rete IPV6 o singole stazioni IPV6.

Due isole IPV6 possono comunicare in un mare IPV4 tramite un tunnel. Prendo pacchetto IPV6 e lo metto dentro pacchetto IPV4. Dentro le due isole IPV6 devo avere dispositivi dualStack in grado di togliere e mettere l'involucro IPV4.

Andando avanti con il tempo queste isole IPV6 cresceranno sempre di più fino ad arrivare ad avere la situazione in cui host IPV6 vorranno comunicare con host IPV4.

Devo avere quindi avere una traduzione del pacchetto. Man mano che le isole crescono ancora arriveremo al punto di avere due isole IPV6 che risulteranno collegate (non c'è più bisogno di tunnel IPV4, c'è connettività diretta IPV6). Alla fine di questo procedimento avrò il capovolgimento della situazione, la rete sarà diventata IPV6 (le isole crescendo sono diventate il mare) e i dispositivi IPV4 saranno diventati le isole. (tunneling opposto, tolgo e metto intestazioni IPV6).

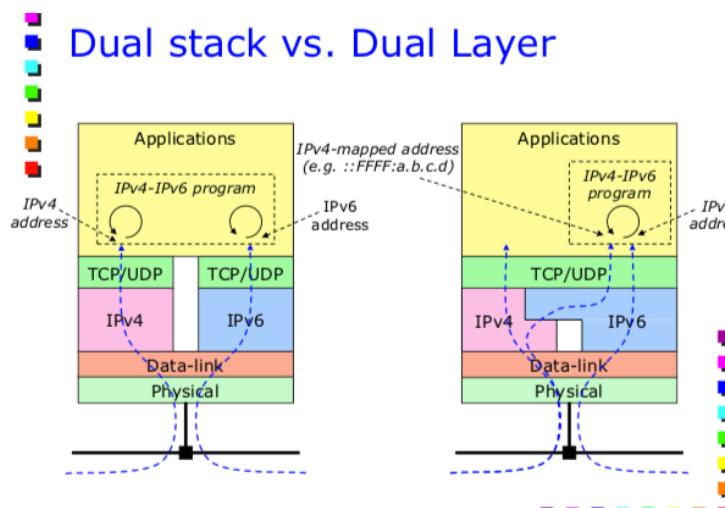
TRANSITION FROM IPV4 TO IPV6

Fondamentale approccio DualStack. Essere dual stack vuol dire che avrò protocolli di routing diversi, routing table diverse e access List (sono dei filtri) diverse.

Uno dei limiti (non proprio limite) dell'approccio dualStack è che chi decide se usare IPV4 o IPV6 sono le applicazioni. L'amministratore di rete non è contento di questo perchè non è lui che ha il controllo.

C'è quindi anche un altro approccio , chiamato DualLayer, non abbiamo più stack duplicato fino all'applicazione.

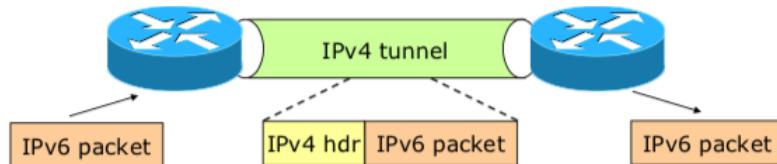
A livello 2 sono sempre protocolli diversi, ma dal livello di traporto ne abbiamo uno unico.



Perdiamo il vantaggio del dualStack, dobbiamo modificare il livello trasporto IPV4 creando nuovo livello trasporto. Le applicazioni così non decideranno più se usare IPV4 o IPV6 ma lo farà il livello trasporto. Con questa configurazione posso anche creare un tunnel o tradurre pacchetti.

Traversing a IPV4 only network

Stazioni IPV6 dualStack che vogliono comunicare attraverso rete IPV4. Usano tunneling, prendo pacchetto e lo mettono dentro pacchetto IPV4.



Il problema ora è che mentre prima in MPLS quando avevo stazioni ATM, le stazioni ATM utilizzavano stesso protocollo di routing dell'IP ora i protocolli di routing sono diversi, uno per IPV4 e uno per IPV6. Una soluzione come quelle di MPLS che crea piano di controllo unico non è più utilizzabile. Non c'è una buona soluzione, ce ne sono diverse ma non ottime.

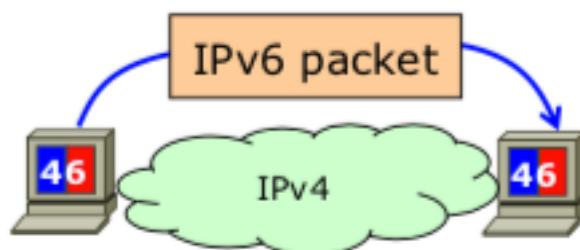
Tunneling

Diverse soluzioni

- Protocollo GRE (generic routing encapsulation), non è specifico per IPV6 e IPV4 posso usarlo anche con altri protocolli. Serve quindi per mettere "qualcosa" dentro un pacchetto IPV4 (non per forza IPV6).
- IPV6 in IPV4 nel campo protocollo di IPV4 metto 41 in esadecimale, che mi dice che quello che è contenuto nel pacchetto IPV4 è un pacchetto IPV6.

SOLUTIONS - Host-centered solutions

Soluzioni basate sugli host dualStack, è l'host stesso che manda pacchetto IPV6 a host IPV6 tramite rete IPV4. Le soluzioni basate su rete sono migliori ma le vedremo dopo.



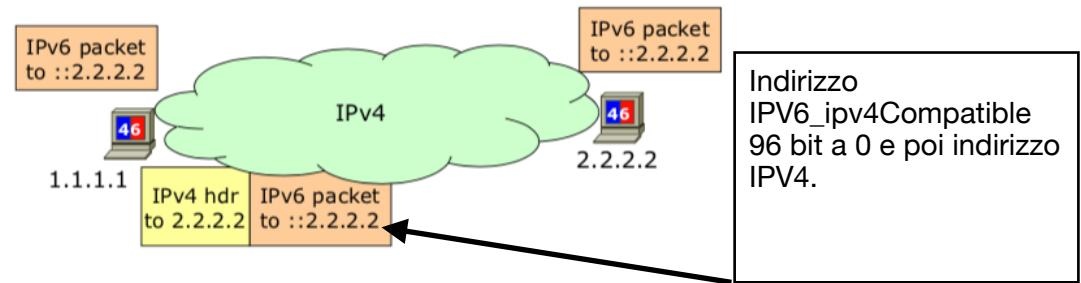
1 IPV4 compatible/Aumatic Tunneling

*L'host capisce automaticamente qual'è l'endPoint che deve fare tunnel. Se un'applicazione vuole usare IPV6 deve usare un **indirizzo IPV4 compatibile**. (96 bit a 0 e poi 32 bit indirizzo ipv4) Non è una soluzione che posso usare in situazione di scarsità di indirizzi.*

La stazione che invia pacchetto sa che quando invio pacchetto ad un certo indirizzo IPV4 compatibile devo mettere quel pacchetto in un tunnel.

Nella tabella di routing IPV6 si ha una route statica per pacchetti per `::/96` che dice che dovranno uscire per una pseudo-interface. È un interfaccia che non esiste veramente ma che è implementata in software che fa il tunneling. Quando la stazione vede che pacchetto inviato a indirizzo IPV4compatibile invece di passarlo al software dell'ethernet lo manda alla pseudointerface.

End-to-End tunnel, tunnel da host a host.



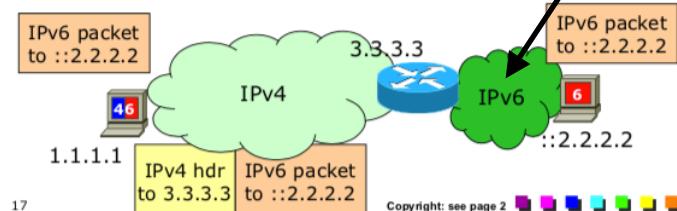
IPV4 header

source: indirizzo IPv4 del mittente
dest: indirizzo IPv4 del destinatario

Pacchetto IPV6

source: indirizzo IPV6 del mittente
dest: indirizzo IPV6 del destinatario

Posso anche non avere stazione IPv6 direttamente collegata alla rete IPv4 (end-to-end tunnel) ma passare tramite una rete IPv6. Nella mia stazione che invia pacchetto devo inserire route statica che mi dica che tutto ciò che va a `::/96` deve passare per la pseduo Interface ma anche che tutto ciò che va `2.0.0.0/104` deve andare a 3.3.3.3 tramite pseudo Interface.



IPV4 header
source: indirizzo IPV4 del mittente
dest: indirizzo IPV4 del router

Pacchetto IPV6
source: indirizzo IPV6 del mittente
dest: indirizzo IPV6 del destinatario

2 6OVER4

Ogni indirizzo IPV6 di ogni macchina (dual stack) contiene il proprio indirizzo IPV4 all'interno dell'interfacceID. Funzionerebbe anche non indirizzi non link local.

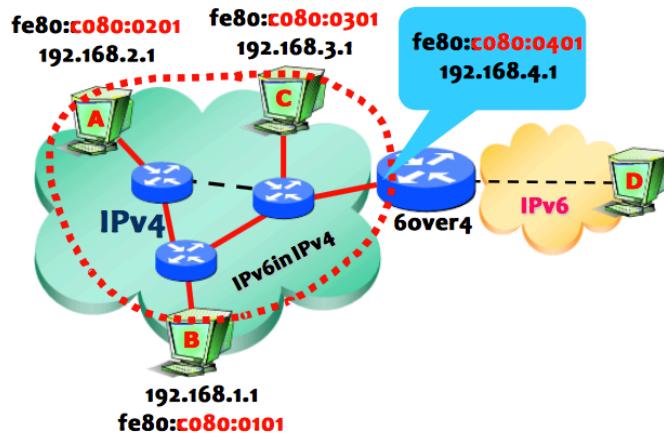
16

80

32



6over4 Example



IPV4 multicast address
src. Indirizzo IPV4 host
Dst. Indirizzo IPV4 host

Pacchetto IPV6 multicast perchè uso SNMA
Src. Indirizzo IPV6 host sorgente FE80::ipv4
Dst. Indirizzo IPV6 host destinazione FE80::ipv4

6OVER4 Multicast

Il fatto di voler far comunicare IPV6 e IPV4 è la stessa cosa che facciamo tra IPV4 e ethernet. Quindi dobbiamo fare neighbor Discovery però tramite multicast IPV4, che non è molto usato ma comunque c'è. Usiamo quindi la rete IPV4 come se fosse la rete Ethernet. IPV6 sta sopra IPV4 come IPV4 sta sopra ethernet.

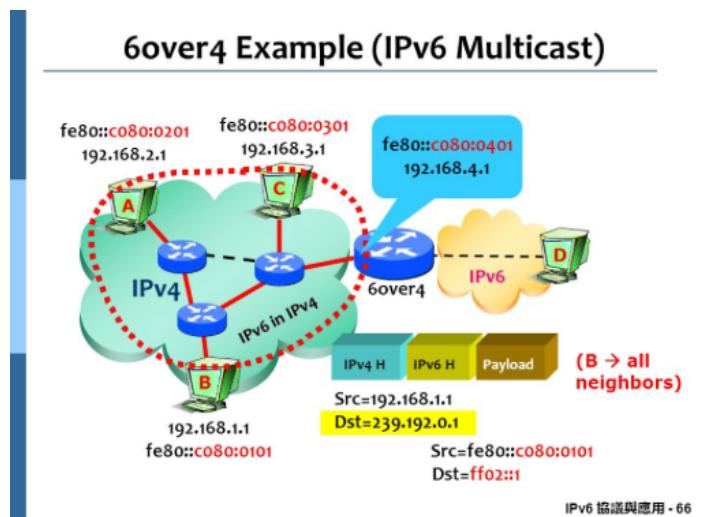
Quando A vuole mandare pacchetto a B devo fare neighbor Discovery tramite SNMA. Indirizzi multicast IPV6 mappati su indirizzi multicast IPV4.

ff02.192.x.y dove x,y sono i due ultimi bytes dell'indirizzo IPV6 SNMA

SI USA PER TUNNELING DI PACCHETTI MULTICAST IPV6

Esempio

fe80::0101:0101	link local IPV6 dell'host che voglio raggiungere
ff02::1:ff01:0101	SNMA costruito dall'indirizzo destinazione IPV6
239.192.1.1	6over4 multicast , imbustato in IPV4multicast (239.192.x.y sarebbe come 33:33 prima)



Quindi io ho un indirizzo IPV4, questo indirizzo lo uso per formare il mio indirizzo IPV6. Dal mio indirizzo IPV6 mi costruisco il mio SNMA e gli ultimi due byte li uso per formare indirizzo IPV4 6over4 multicast.

IPV4 multicast address

Src. Indirizzo IPV4 host
Dst. Indirizzo IPV4 multicast 239.192.x.y

Pacchetto IPV6 multicast perchè uso SNMA

Src. Indirizzo IPV6 host sorgente
Dst. Indirizzo IPV6 SNMA host destinazione

Questo è utile quando per esempio una macchina vuole scoprire i router IPV6 sulla sua rete. Al posto dell'indirizzo dest SNMA metterà FF02::1.

3 ISATAP

Si chiama IntraSite perchè sto creando Tunnel direttamente nella nuvola IPV4 di un service provider in cui sono immersi i miei host DualStack.

Intra-site, non hai multicast IPV4 quando la rete è di un service provider che non supporta multicast.

Non posso usare neighbour Discovery perchè non ho multicast IPV4, quindi non posso ricavarmi indirizzo IPV4 dei miei vicini. (prima avrei ricavato indirizzo MAC, ma ora al posto dell'ethernet ho IPV4).

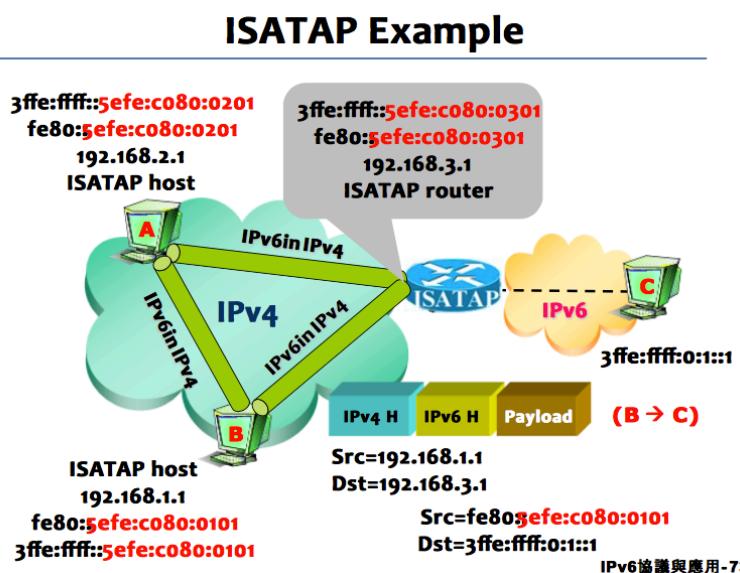
Allora lo includo nell'indirizzo IPV6. Facendo così l'interfaceID contiene l'indirizzo IPV4.

00005efe prefisso, mi dice che sto usando ISATAP

Esempio

fe80::5efe:0101:0101 for 1.1.1.1 indirizzo IPV6 costruito dal proprio ind IPV4.

64 Bits	32 Bits	32 Bits
link local or global IPv6 unicast prefix	0000:5EFE	IPv4 address of the ISATAP link



Se B vuole comunicare con C

Header IPV4	Pacchetto IPV6
source: indirizzo IPV4 hostISATAP dest: indirizzo IPV4 del routerISATAP	source: indirizzo IPV6 hostISATAP dest: indirizzo IPV6

Se B vuole comunicare con stazione su stessa rete IPV4 si usa Automatic Tunneling.

Quindi se conosco indirizzo IPV6 della destinazione questo indirizzo IPV6 conterrà il prefisso che mi dice che è ISATAP più l'indirizzo IPV4 della destinazione.

Gli ultimi 4 byte dell'interfaceID della destinazione sono l'indirizzo IP destinazione. Questo va bene per stazioni OnLink.

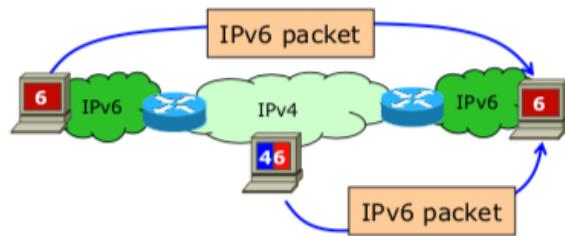
Dato che non ho multicast IPV4 non posso scoprire tutti i router sulla rete. Devo quindi avere una Potential router List.

Una stazione che usa ISATAP si crea interfaceID a partire da suo indirizzo IPV4 e poi si crea lista di router tramite:

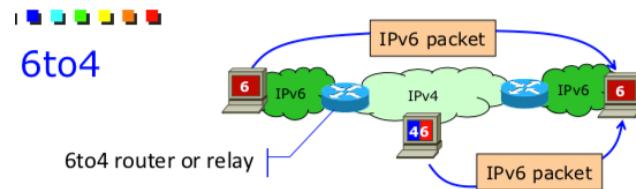
- DHCP quando stazione fa configurazione IPV4 tramite DHCP riceve la PRL. Soluzione proprietaria.
- DNS La stazione che sa di usare ISATAP fa una richiesta di risoluzione per ISATAP.NomeDominio. Ritorna lista di possibili router IPV6 che stazione ISATAP può usare. La stazione a questo punto manda richiesta di routerSolicitation e router gli manda routerAdvertisement con per esempio prefissi da usare su quel link. A questo punto l'hostISATAP incapsula il pacchetto in questo modo e lo manda al router e il router quando riceverà pacchetto toglie l'header IPV4 e inoltra pacchetto IPV6.

SOLUTIONS - NetWork Centered solutions

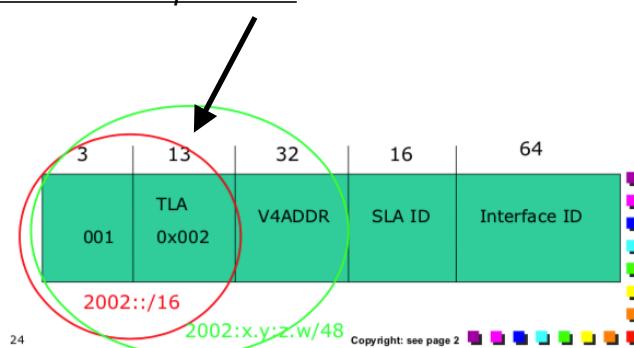
Due nuvole IPV6 con host sia dualStack che solo IPV6 devo comunicare attraverso backBone IPV4. Un dispositivo si occuperà di fare ciò che serve per far passare IPV6 su IPV4.



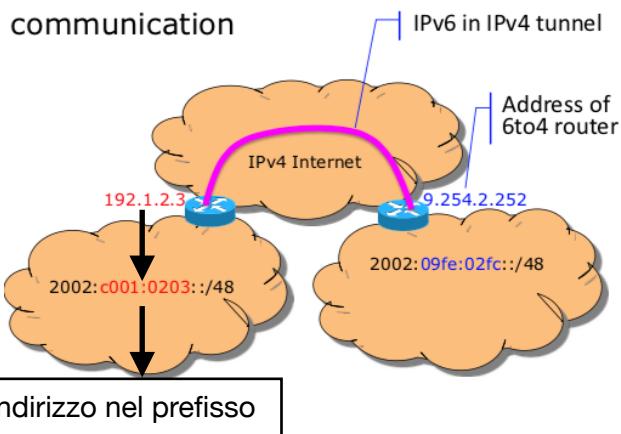
1 6TO4



Si fa un mapping dell'indirizzo IPV4 del 6to4 router nell'indirizzo IPV6, si include indirizzo IPV4 in indirizzo IPV6. Molto simile ad 6over4 ma 6over4 è usato dall'endHost (indirizzo IPV4 nell'interfaceID) mentre ora abbiamo sottorete diverse, quindi tanti host che fanno capo allo stesso terminatore di tunnel, quindi non metto indirizzo nell'interFacId ma nel prefisso.

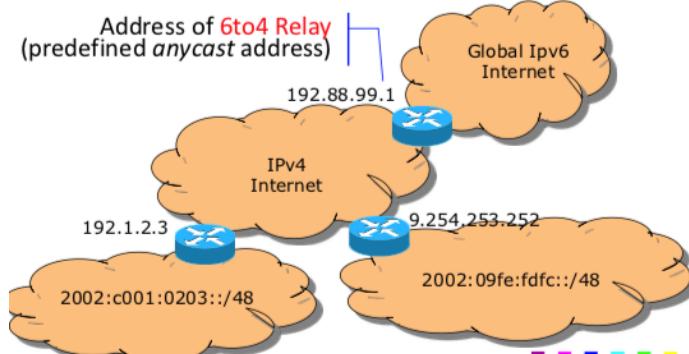


2002::/16 (prefisso) + x.y.z.w/48 (IPV4) + ::/64 (sottorete IPV6)



Header IPV4	Pacchetto IPV6
Source: indirizzo IPV4 6to4 router collegato a rete IPV6 mittente dest: indirizzo IPV4 del 6to4 router collegato a rete destinatario	source: indirizzo IPV6 6to4 del mittente dest: indirizzo IPV6 6to4 destinatario

Quando pacchetto inviato da utente rosso vuole raggiungere utente blu, l'utente rosso lo invia al suo router 6to4. Il router prende l'indirizzo IPV6 destinazione e guarda il nextHop nella sua tabella di Routing. Trovato il nextHop si ricava il suo indirizzo IP dal prefisso e inoltra pacchetto. Il router 6to4 dell'utente blu toglie intestazione IPV4 e manda pacchetto a utente blu.



6to4Relay sono i 6to4Router che fanno da defaultGateway, hanno un indirizzo unicast IPV4 predefinito. Ci possono essere anche più relay collegati ma tutti devono usare stesso indirizzo IP —> indirizzo Anycast, non va ad un host particolare ma ad uno di un insieme di host. Avremmo quindi diversi host che annunceranno la stessa destinazione.

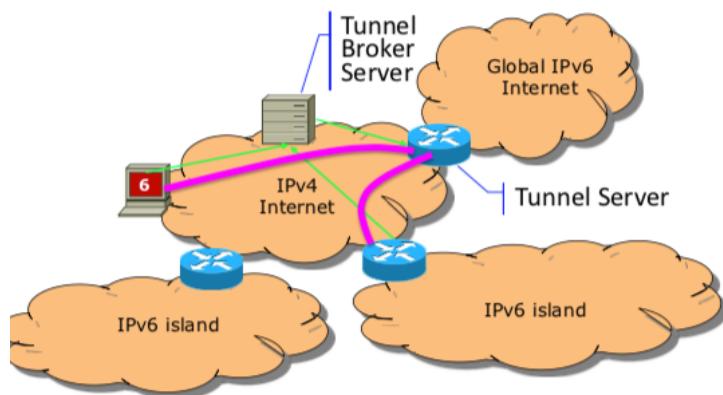
Nel backBone IPV4 non posso usare il NAT.

2 Teredo

L'imbustamento dei pacchetti IPV6 è fatto dentro UDP e non direttamente dentro IPV4, così posso usare le porte e oltrepassare il NAT.

3 TunnelBroker

Si usa un sever, tunnelBrokerServer. Quando A manda pacchetto a B il pacchetto viaggia fino al RouterDiBordo, il router deve sapere indirizzo IPV4 del router di Bordo di B. Questo lo chiede al tunnelBrokerServer che ha la lista di tutti i router nel backBone IPV4.



Se il tunnelBrokerServer non trova la destinazione invia il pacchetto verso tunnelServer che è collegato a internet.

TunnelSetupProtocol serve per stabilire I tunnel.

TunnelInformationControl Serve per registrarsi al tunnelBroker.

Il tunnelBrokerServer è un single point of failure.

SOLUTIONS - Scalable, Carrier-grade Solution

Abbiamo host nativi IPV6 oppure IPV4 che voglio scambiare pacchetti attraverso rete IPV4 o IPV6.



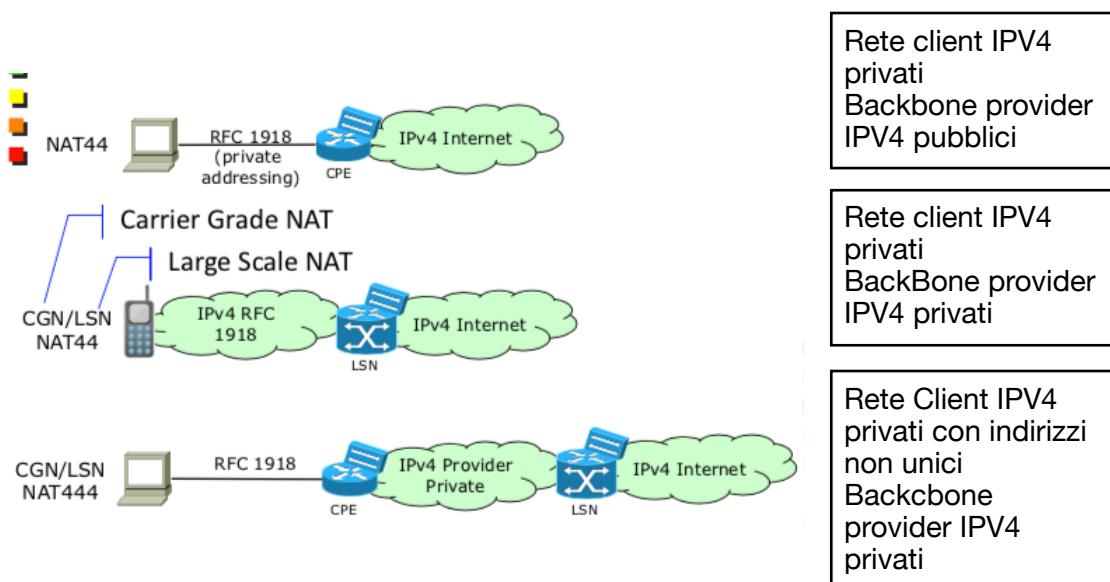
Soluzioni che agiscono in un contesto diverso, c'è un serviceProvider che da servizio a tanti clienti eventualmente residenziali. In questo caso qui il serviceProvider si trova in una situazione che influenza la soluzione.

Il service provider avrà a che fare

- stazioni IPV4 con indirizzi privati
- le stazioni saranno dei client, inizieranno la comunicazione mentre prima avevamo isole Peer, potevamo avere sia client che server.

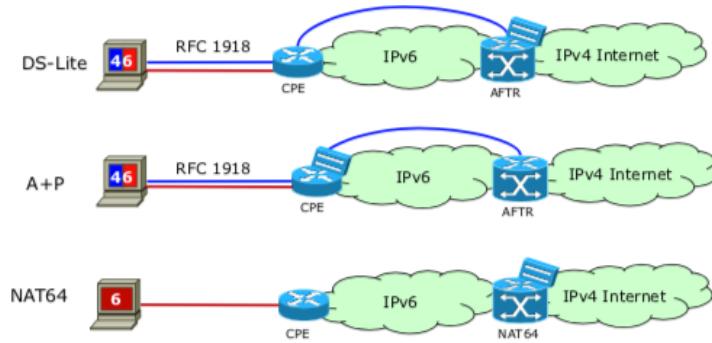
NAT (IPV4)

I serviceProvider usano pesantemente indirizzamento privato. Possiamo avere indirizzamento privato solo su rete locale dell'utente finale e poi abbiamo un NAT (CPE, customer Equipment, modem) che traduce l'indirizzo in pubblico e viceversa e poi pacchetto viaggia su internet IPV4. (qui ci saranno indirizzi pubblici, nel backbone).



In realtà molti serviceProvider usano indirizzi privati anche nella loro rete. Dovremo avere un dispositivo CarrierGradeNat/LargeScaleNat LSN che quindi dovrà gestire un numero molto maggiore di indirizzi rispetto al NAT nella prima configurazione. In questa configurazione ho un'unica grande rete locale per provider e client.

Oppure serviceProvider ha rete IPV4 con indirizzi privati, clienti con indirizzi privati ma non sono unici, cioè client in reti locali diverse collegate al serviceProvider posso riutilizzare stessi indirizzi privati. Quindi avremo prima un CPE che traduce privato in privato poi un LSN che traduce privato in pubblico. Con questa configurazione posso avere più reti locali private per i vari client tutte mappate sullo stesso indirizzante privato.



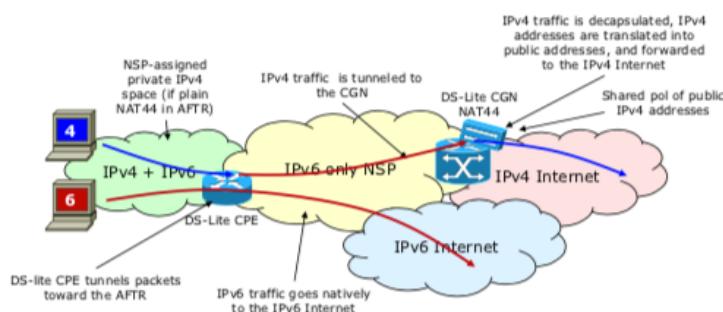
Si cerca di trasformare il backBone del serviceProvider in IPV6.

L'AFTR permette ad utenti IPV4 di comunicare con altri utenti IPV4 tramite un backbone di un serviceProvider IPV6.

Diversi approcci:

DS-LITE

utenti solo IPV4 o solo IPV6 su rete IPV4+IPV6, IPV6 only backbone del serviceProvider. Un solo AFTR che estraе pacchetto IPV4 da pacchetto IPV6 incapsulato dal CPE.



L'AFTR ha anche il NAT che fa traduzione indirizzo privato-pubblico. Tra rete locale e BackBone non devono esserci indirizzi uguali, devo dare indirizzi diversi ai clienti dato che non ho NAT su CPE. Ogni CPE nella sua rete locale deve usare indirizzi diversi. Si chiama così perchè il dualStack lo ho solo agli estremi della rete non nel BackBone.

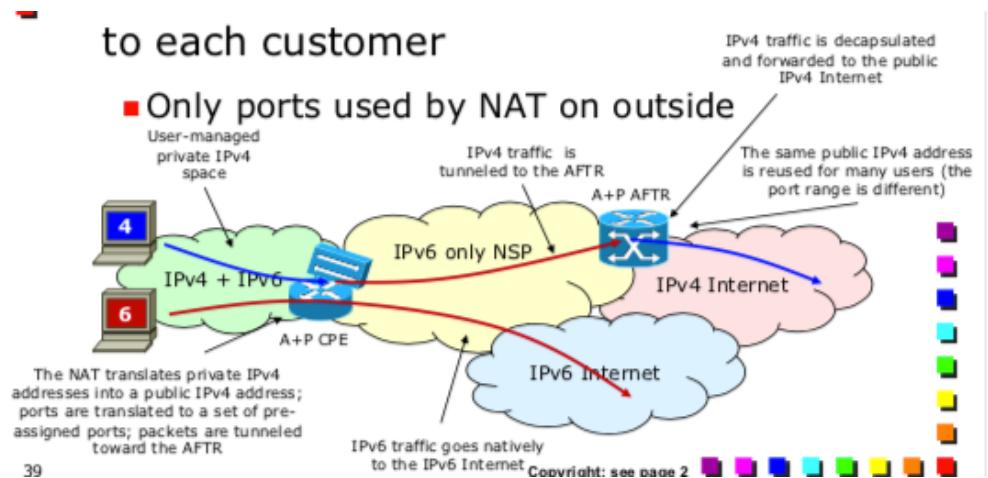
Per un pacchetto che viaggia dalla nuvola IPV4+IPV6 verso internet.

HEADER IPV6 Src. Indirizzo IPV6 CPE Dst. Indirizzo IPV6 AFTR	HEADER IPV4 Src. Indirizzo IPV4 privato Dst. Indirizzo IPV4 pubblico
--	--

In questo caso il NAT non è sotto il controllo del cliente.UN HOST IPV4 NON PUÒ CONTATTARE HOST IPV6.

A+P

Si sposta NAT da AFTR a CPE. Ora nel CPE il NAT non ha a che fare con tanti indirizzi privati, come prima.



Si fa un NAT anche delle porte. Si assegna al CPE un RANGE di porte che può usare. I CPE possono rinegoziare range di porte assegnate allargandolo o stringendolo. Se faccio range di porte larghi posso supportare meno CPE e viceversa. Ogni NAT sui CPE richiede indirizzo IPV4 pubblico, si usano indirizzi duplicati e i CPE si distinguono grazie alle porte quindi in ogni rete locale di ogni CPE gli indirizzi che useremo saranno gli stessi e li distinguiamo in base alle porte assegnate a quel CPE.

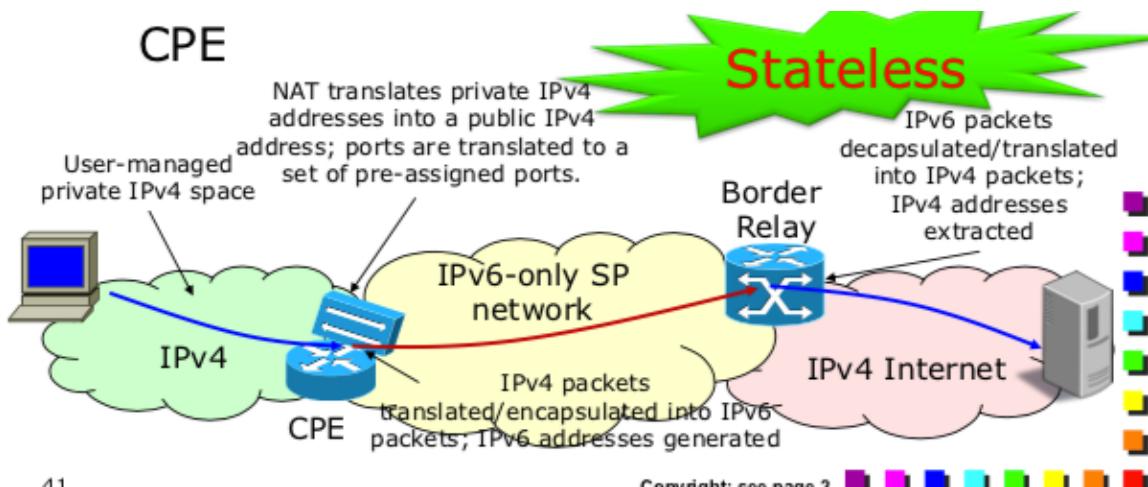
In questo caso il cliente ha il controllo del NAT.

Per un pacchetto che transita da nuvola IPV4 + IPV6 a internet IPV4:

HEADER IPV6 Src. indirizzo IPV6 CPE Dst Indirizzo IPV6 AFTR	HEADER IPV4 Src. Indirizzo IPV4 pubblico CPE Dst. Indirizzo IPV4 pubblico internet + PORTE
---	--

MAP

Fa cosa del tutto analoga a prima, fa sì che CPE abbiano il NAT al loro interno, più CPE condividono indirizzo IPV4 pubblico e più CPE usano SET di porte diverse. Soluzioni STATELESS, la tabella non dipende da quanti CPE ho. L'AFTR, che ora si chiama BorderRelay, si costruisce automaticamente l'indirizzo IPV6 dell'endPoint del tunnel (CPE) a partire dall'indirizzo IPV4 e dalla porta del pacchetto che ritorna dal server.



Ci sono due varianti

- MAP-E tunneling (encapsulation)
- MAP-T translation, IPV4 tradotto in IPV6 e poi borderRelay li traduce di nuove in IPV4 e viceversa.

La differenza tra queste due varianti riguarda l'indirizzo del border relay. Se faccio tunneling l'indirizzo del server sta nell'intestazione IPV4. Se faccio traduzione non c'è intestazione IPV4 perché viene tradotta ma devo comunque ricordarmi dell'indirizzo IPV4. La traduzione di cui si sta parlando non è quella del NAT ma è quella che serve al posto di fare Tunneling.

Port set

Nella soluzione MAP si introduce il concetto di PortSet (diverso da RANGE). Devo mantenere traccia delle porte che sono state assegnate al CPE così almeno riesco a distinguere i CPE in base alle porte che gli sono state assegnate.

Invece che assegnare un range di porte (contigue) assegno un set di porte (non contigue) questo per load balancing e molti altri motivi.

numero di porta (16bit).

Il set viene identificato per ogni CPE dando un numero fisso di bit (kBits) che stanno in mezzo, e quelli all'inizio alla fine posso cambiarli. Più k è grande più posso condividere quell'indirizzo IP con tanti CPE però ogni CPE avrà meno porte da usare. 16 bit totali.

A bits	K bits PSID (portSet identifier) parametro di configurazione che posso variare, determina la sparsità del set	M bits
--------	--	--------

A>0 per evitare porte statiche, quelle che si usano normalmente nei server.

CPE IPV6 Address

Serve per costruire indirizzo IPV6 del CPE partendo dal suo IPV4.

n bits o bits s bits 128-n-o-s bits

Rule IPV6 prefix Fornito al momento della configurazione del CPE	EA bits Contengono un certo numero di bit dell'indirizzo pubblico e il PSID	SubnetID Rispecchiano la divisione degli indirizzi IPV6, stesso nome	InterfaceID 128 bit
---	--	---	------------------------

Mapping Rule

Configurazione del CPE, ho bisogno di

- Rule IPV6 prefix
- Rule IPV4 prefix indirizzo IPV4 pubblico che CPE usa per fare traduzione
- EA bits Length
- offset PSID sarebbe il valore di A (vedere portSet)

Esempio costruzione:

2001:1:1100::/40 prefisso 40 bit

195.2.2.20/24 prefisso IPV4

EABit length 16

PSID offset 4

PSID 0x33

Indirizzo IPV4 CPE → 195.2.2.4

Indirizzo IPV6 CPE

40 bit	16 bit-EA	8 bit	64 bit
2001:1:11 Identifica univocamente CPE	04:33	0 0	0000:C302:0204:0033 0000+ind IPV4 CPE+PSID

2001:0001:1104:3300:0000:C302:0204:0033

Questo è l'indirizzo mittente che userà CPE quando fa tunnel. O viceversa è l'indirizzo destinazione che usa il borderRelay per fare tunnel verso CPE.

Border Relay (MAP)

Più BorderRelay con stesso indirizzo IPV6 uso anycast. Gli indirizzi dei borderRelay devono essere conosciuti dai CPE.

Outside IPV4 destination - Address Translation (MAP-T)

Uso IPV4-embedded IPV6 address per non perdere indirizzo IPV4 quando faccio la traduzione. Serve per costruire IPV6 destinazione partendo dall'id IPV4 della dest.

64 bit	8 bit	32 bit	24 bit
--------	-------	--------	--------

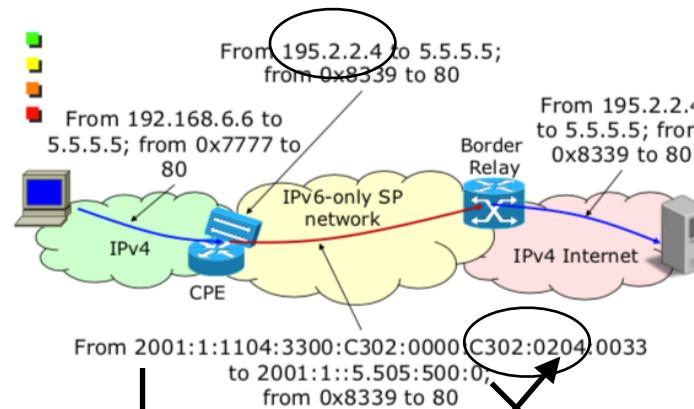
Prefisso del borderRelay 64 bit	0 0 zeri 8 bits	IPV4 address Indirizzo del server che voglio raggiungere 32 bit	0....0 zeri 24 bit
---	---------------------------	--	------------------------------

Esempio

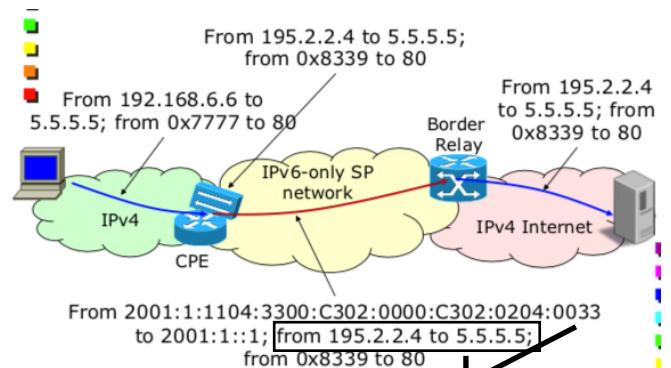
2001:1::/64 prefisso borderRelay
5.5.5.5 outside Destination

2001:1::0005:0505:0500:0000

MAP-T (translate)



MAP-E (encapsulation)



Indirizzo destinazione tradotto, mantenendo IPV4

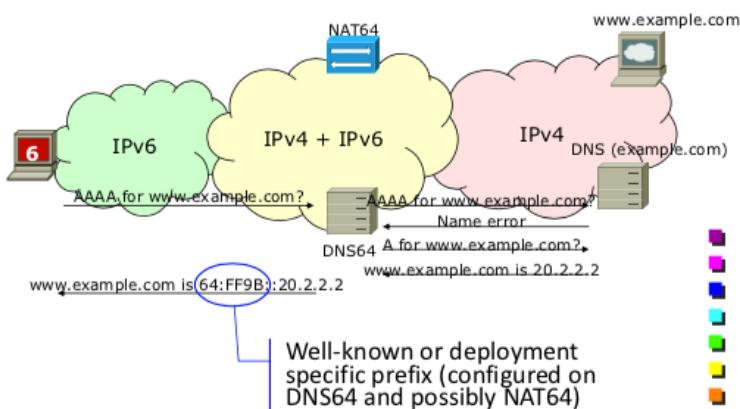
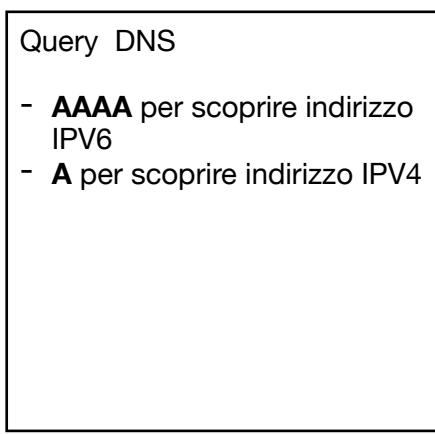
Indirizzo del CPE ipv4 pubblico (195.2.2.4) tradotto in IPV6 mantenendo IPV4.

Indirizzo IPV6 del CPE tradotto come prima per metterlo in headerIPV6.

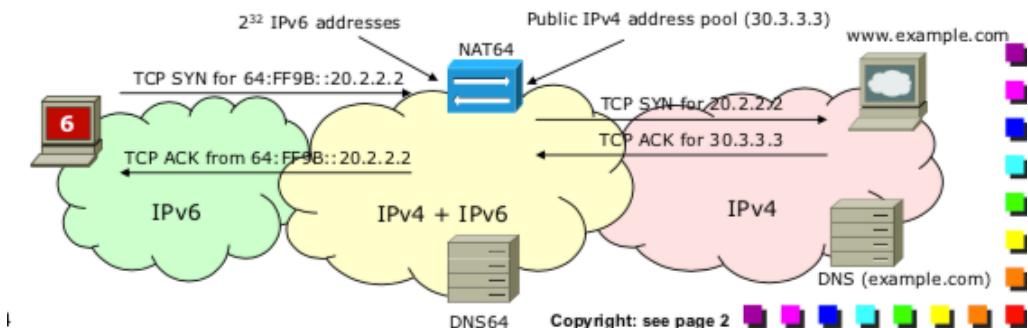
Il pacchetto IPV4 viene incapsulato in uno IPV6, ho tutte le informazioni degli indirizzi IPV4

Nat64+DNS64

Gli endPoint sono IPV6. Il backBone (IPV6 + IPV4) e anche gli utenti sono IPV6. Quando A vuole comunicare con B , B è IPV4 ma A è IPV6. Uso di nuovo indirizzi IPV4 mapped. A quindi si deve costruire questo indirizzo però ora A è una qualsiasi stazione IPV6 e non c'è più il CPE. Allora A per collegarsi a IPV6 avrà ricevuto un nome di B e farà una risoluzione dei nomi. Ci sarà un DNS 64 che partecipa all'operazione, quando A fa la richiesta al DNS (richiesta AAAA e poi A) questa risoluzione arriva al DNS64 il quale costruisce l'indirizzo IPV4 mapped per A (il quale essendo solo IPV6 non poteva farlo, mentre prima il CPE poteva farlo).

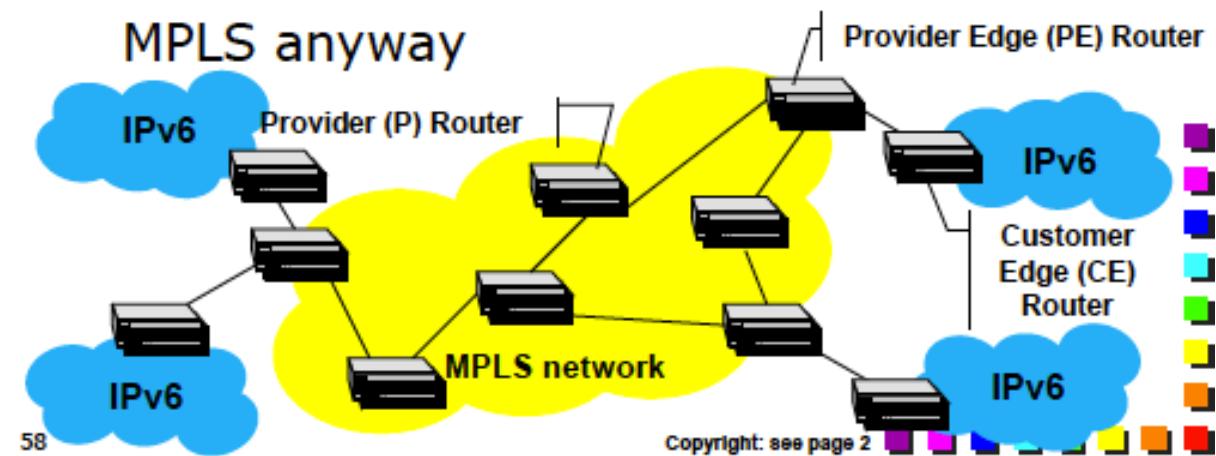


A a questo punto invia pacchetto all'indirizzo ricevuto, arriva al NAT64, il quale conosce il prefisso usato per il mapping, lo traduce in IPV4 e lo manda a B. Come mittente del pacchetto ci sarà l'indirizzo pubblico che usa il NAT64.



Soluzione MPLS

Backbone IPV4 MPLS (*MPLS IPV4 perchè uso il piano di controllo IPV4 per la creazione degli LSP, il piano dati è indifferente*) e clienti in zone IPV6. MPLS è multi protocol , una volta messe le etichette gestisce sia pacchetti IPV4 che IPV6.



Il problema è la creazione dell'LSP tra due utenti.

Si può creare in modo proattivo degli LSP tra tutti i router che sono al bordo della rete (Ingress/Egress LSR) e in questo modo loro possono scambiarsi informazioni di routing. → NON SCALABILE, NON VA BENE.

Come detto prima il piano di controllo usa solo indirizzi IPV4, è un problema. Viene risolto con 6PE.

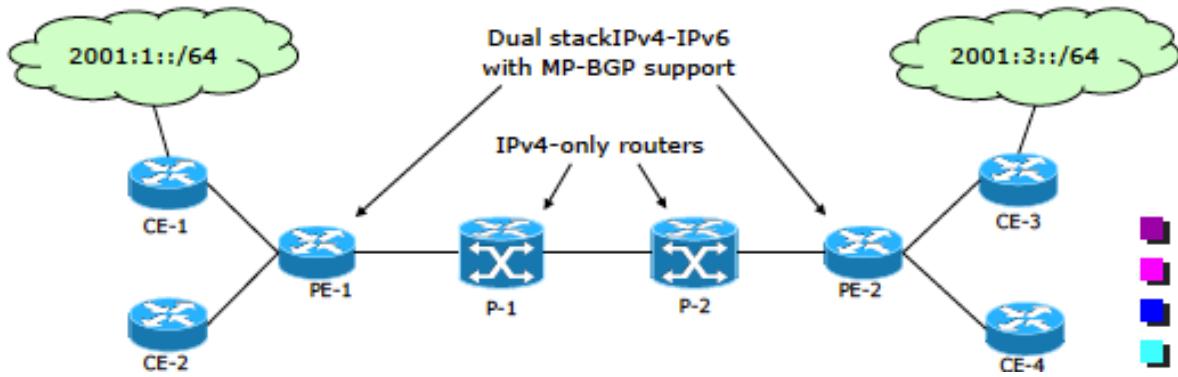
6PE

In molti contesti gli apparati al bordo della rete vengono chiamati PE (providerEdge, indica che rete MPLS è di un serviceProvider, cioè da un lato conosce IPV4-backBone e dall'altro conosce IPV6-client).

Il router che sta nella nuvola IPV6 client invece viene chiamato CE (customerEdge)

Solo i dispositivi di EDGE devono capire IPV6 all'interno della rete MPLS-backBone.

Si cerca di fare soluzione scalabile, i router vogliono scambiarsi informazioni su stazioni IPV6 in modo da aprire LSP verso queste destinazioni ma senza cambiare tutti i router della rete (la maggior parte li teniamo IPV4 a parte gli EDGE).

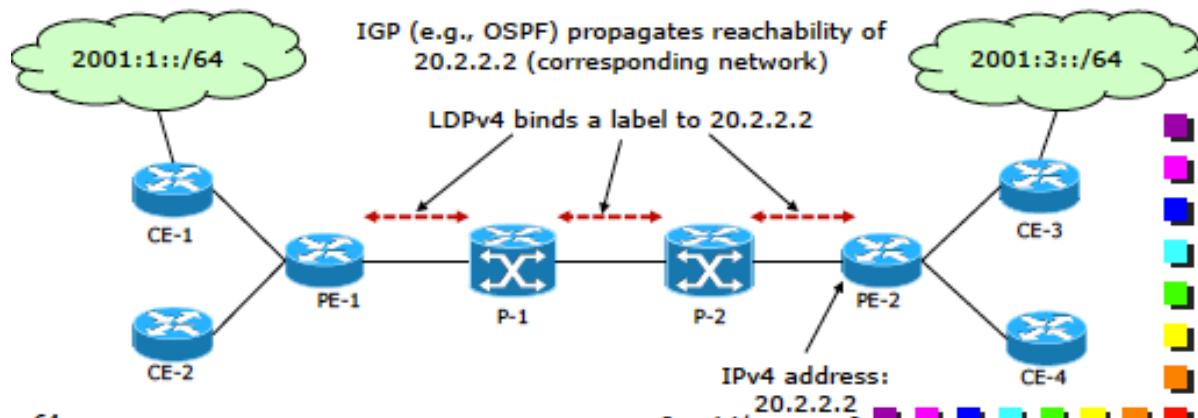


Nel backBone i router capiscono solo IPV4 in modo che riescano a scambiarsi informazioni tramite IPV4 e creare degli LSP tra i router del backbone.

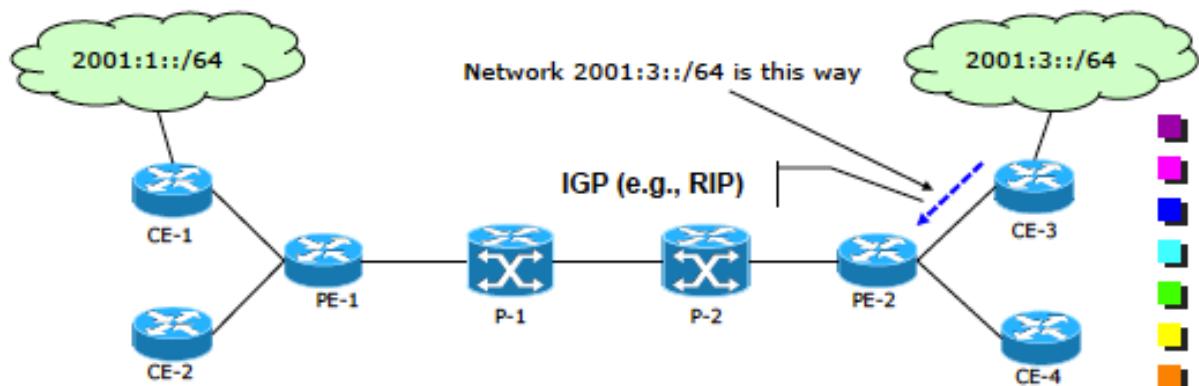
I PE creeranno degli LSP verso gli altri PE. I PE usano BGP (protocollo usato da router di bordo degli autonoMoSystem, I-BGP perchè stanno all'interno della stessa rete MPLS).

Svantaggio configurazione complicata.

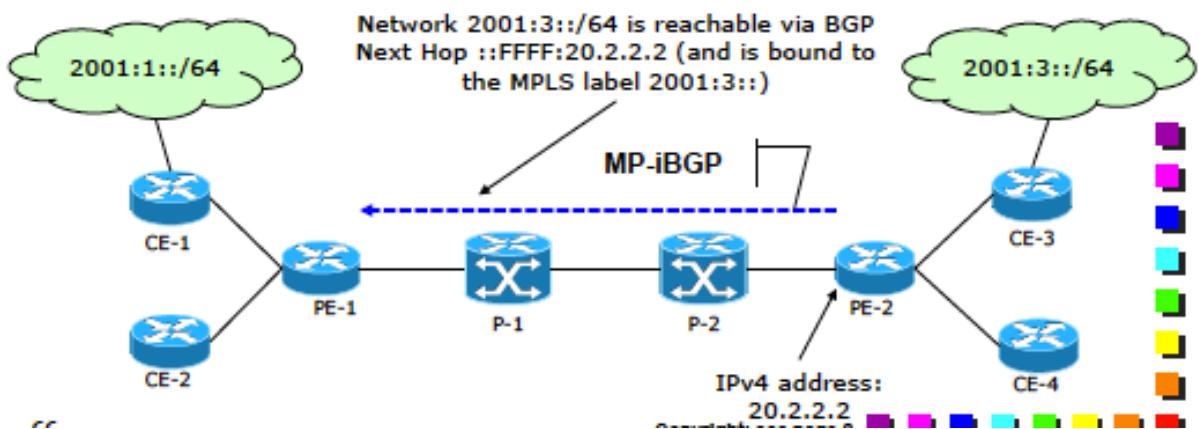
Tramite routing interno i PE e i P si scambiano informazioni riguardanti prefissi IPV4.



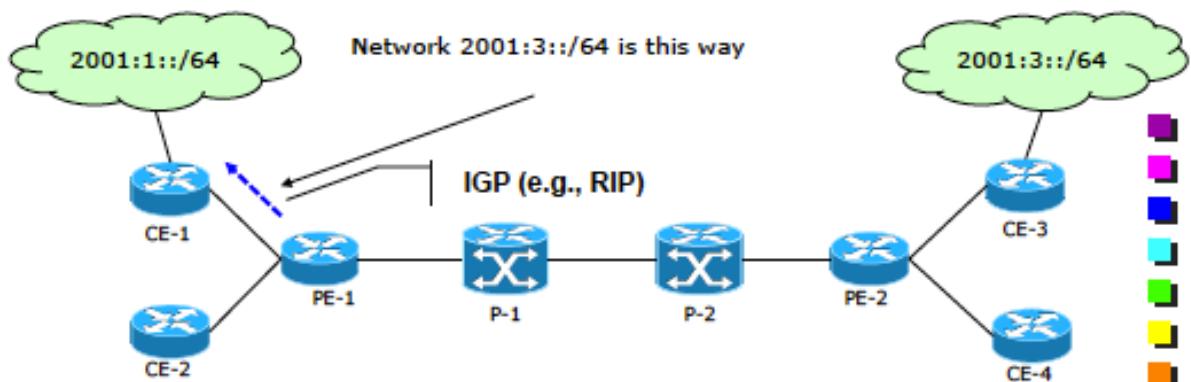
Anche i PE e i CE si scambiano informazioni di routing tramite IPV6 (con protocollo di routing IPV6 per esempio OSPFv6) riguardo le nuvole IPV6 tramite BGP,RIP,OSPF.



Dobbiamo ora distribuire le informazioni riguardanti le RouteIPv6 che i PE hanno appena appreso.



Ora le informazioni riguardanti le reti IPV6 dopo aver passato il backbonePV4 posso essere distribuite.

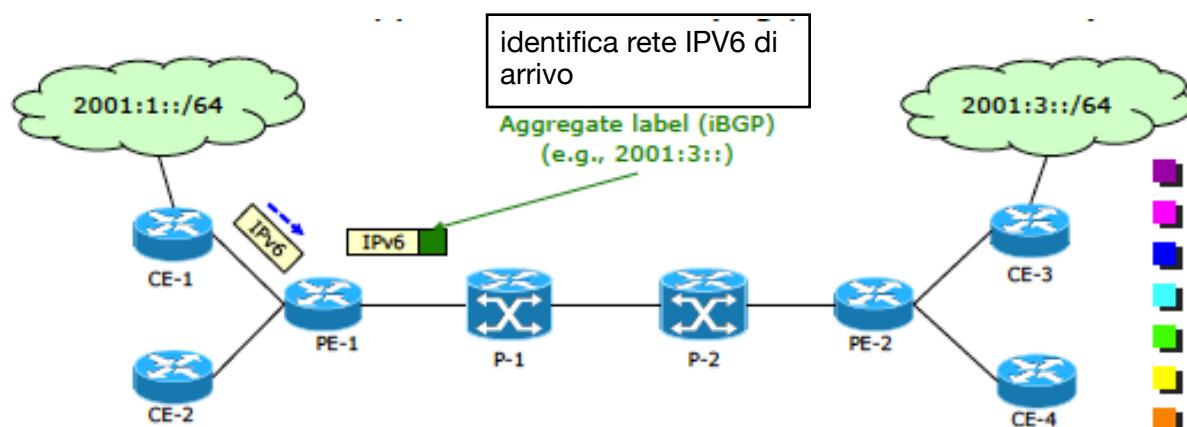
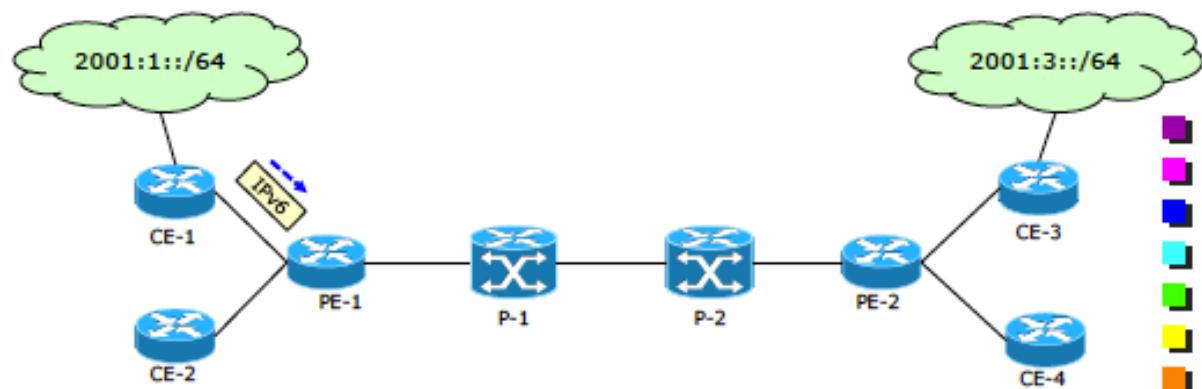


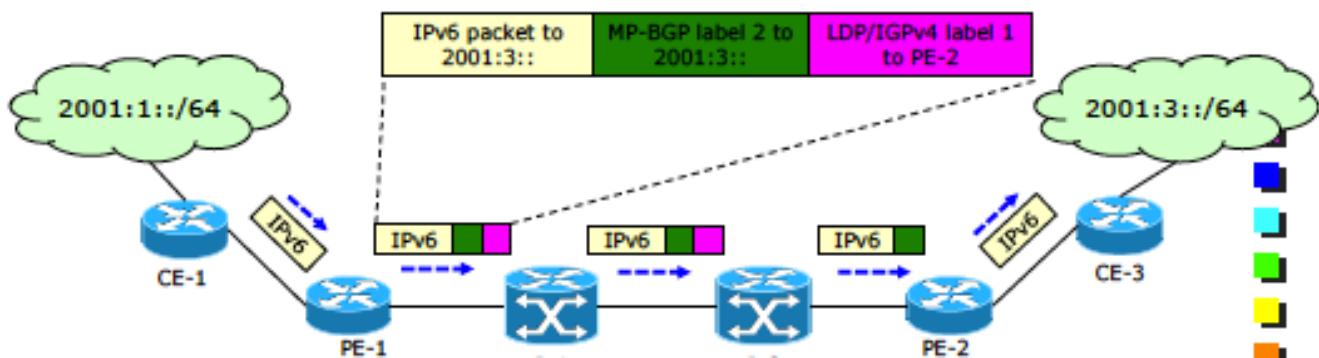
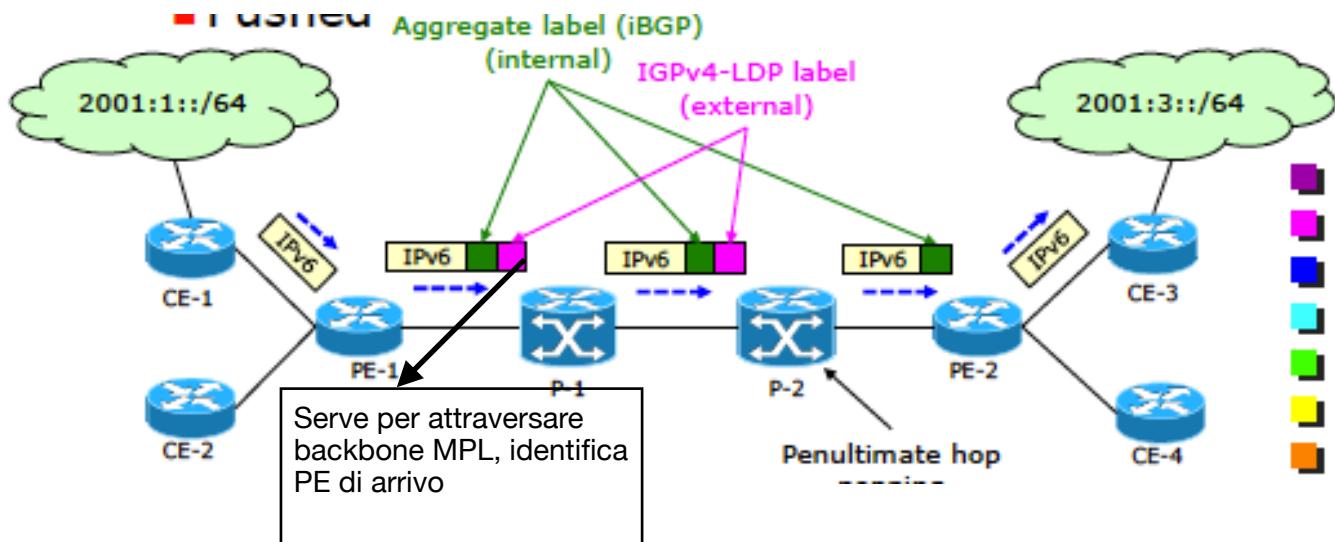
BGP può essere facilmente esteso, oltre ad annunciare destinazioni con indirizzo IPV4 posso annunciare destinazioni con altri indirizzi, dividendoli per famiglie. Quando annuncia la destinazione viene annunciato anche nextHop (strano perché BGP usa distance vector in cui si ha solo destinazione-distanza e non il nextHop). Ma questo è BGP esteso, aggiungo una metrica, il nextHop). Come nextHop scrive il proprio indirizzo IPV6 che è IPV4 mapped.

In MPLS la distribuzione delle etichette può essere fatta anche tramite BGP. Quando PE annuncia destinazione distribuisce anche un etichetta. Il PE si sta creando un Bindind tra etichetta e destinazione, vuole vedersi arrivare tutti i pacchetti per la destinazione che ha annunciato con quell'etichetta. Anche l'altro PE quindi può fare mapping di quell'etichetta.

Per far passare il pacchetto attraverso i router IPV4 del backBone viene aggiunta etichetta che hanno scelto i router interni scambiandosi informazioni. Quindi ogni pacchetto avrà due etichette, quella più interna scelta tra PE (e indica informazioni scambiate con il proprio CE) e quella più esterna scelta dai router interni della rete MPLS IPV4.

Esterna	Interna	Pacchetto
Per attraversare rete IPV4 MPLS, identifica PE di arrivo	Per identificare rete IPV6 di arrivo.	





Il vantaggio di usare questa soluzione è che solo i PE dovranno essere dualStack. Per la rete MPLS il fatto di avere pacchetto IPV4 o IPV6 non cambia niente perché una volta che il pacchetto ha un'etichetta la rete sa come gestirlo indipendentemente dal tipo pacchetto.

PHP penultimate hop popping, l'etichetta la toglie il penultimo nodo dell'LSP. Si fa così perchè all'ultimo nodo l'etichetta non serve in realtà perchè nel pacchetto c'è scritto indirizzo destinazione per inoltrare pacchetto verso nextHop. Viene fatto solitamente con l'etichetta esterna solo (quella viola nelle slide).

VPN

Realizzare connettività su un'infrastruttura condivisa come se fosse in realtà privata.

Uso rete condivisa come se in realtà fosse dedicata solo a me, quindi posso applicare proprio indirizzamento (utilizzo di indirizzi privati) per esempio. Si cerca quindi di inviare pacchetti con indirizzi privati su una rete che non è privata.

Per problemi legati alla sicurezza si usa la crittografia.

Si usano le VPN per evitare di dover fare infrastruttura privata.

Se azienda ha vari siti remoti solitamente si creava rete privata invece ora usiamo la rete internet/ rete di service provider per connettere le varie sedi remote.

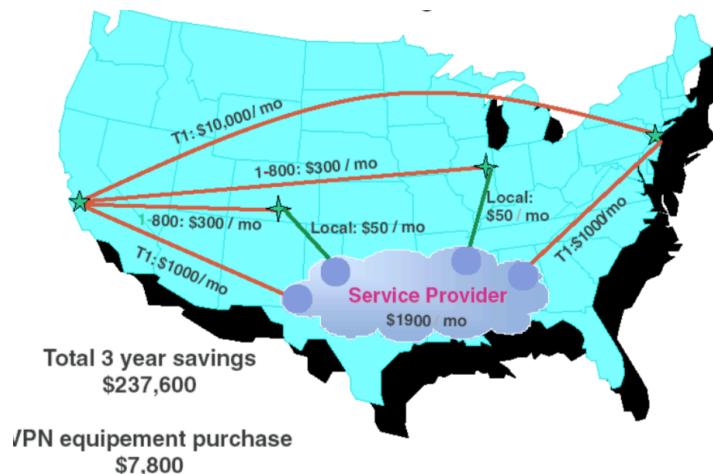
Elementi principale VPN

Tunnel	metto pacchetto IP dentro un pacchetto che viene trasmesso su infrastruttura condivisa. Il contenuto del pacchetto esterno può essere cifrato/autenticato per ragioni di sicurezza dato che sull'infrastruttura condivisa si trovano tutti.
VPN Gateway	dispositivi di terminazione del tunnel.

Motivazioni

- I costi vengono ridotti perchè i collegamenti dedicati sono costosi sono fatti con tecnologia più costosa. **Se ho azienda con punti dislocati a lunga distanza per effettuare collegamento a lunga distanza tra le varie sedi pagherei troppo. Se passo invece dalla rete di un service provider risparmio.**
Devo collegare il mio router a quello del service provider, basso costo, collegamenti a bassa distanza (locali). In più devo pagare la rete del service provider ma i costi sono comunque inferiori perchè sfruttando la commutazione di pacchetto e la multiplazione statistica il service provider sfrutta il fatto che i canali saranno occupati solo dal traffico effettivo presente sulla rete a differenza della commutazione di circuito in cui venivano allocate risorse a priori. Riesci a dividere i costi tra i vari clienti.

Avremo quindi le varie sedi aziendali collegate ad una rete condivisa (internet o rete di un service provider).



Oltre a limitare i costi le VPN offrono anche maggior sicurezza

- servizi limitati ad utenti esterni tramite firewall

Tutte le funzionalità della rete aziendale saranno disponibili come se fossimo direttamente collegati alla rete aziendale.

Diverse soluzioni VPN

Caratteristiche principali delle soluzioni

Deployment model

- Overlay infrastruttura condivisa non sa che l'azienda usa VPN. la rete condivisa sta sotto la VPN, la usiamo solo per trasportare pacchetti. Come in 6over4 IPV6 sta su IPV4.
- Peer infrastrutture condivisa sa che l'azienda usa VPN e la aiuta per esempio scegliendo instradamento migliore.

	overlay	peer
access	L2TP, PPTP	
site-to-site	IPsec, GRE	MPLS

Provisioning model

- Customer l'azienda implementa la VPN installando i propri dispositivi.
- Provider l'azienda si collega alla rete del service provider, e questo fornisce un servizio di VPN. Installa e gestisce i VPN gateway.

Protocol model

- Protocol layer 2 VPN implementata con protocolli di livello 2.
- Protocol layer 3 VPN implementata con protocolli di livello 3.
- Protocol layer 4 VPN implementata con protocolli di livello 4.

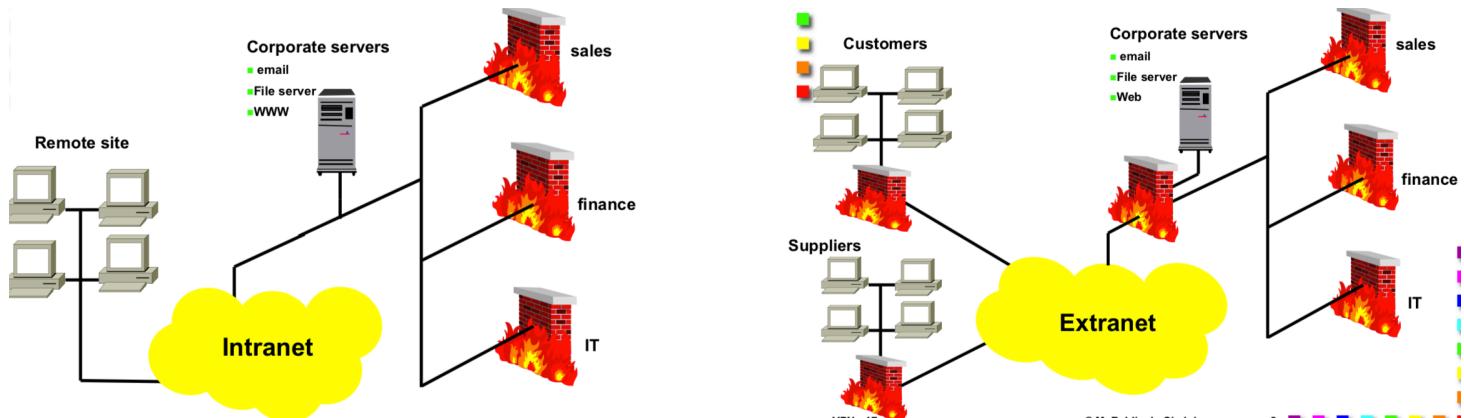
Tipi di VPN

- VPN di accesso un singolo host remoto che deve comunicare con azienda tramite struttura condivisa.
PPTP, L2TP
- VPN site-to-site più siti aziendali, cioè più reti remote che devono comunicare tramite struttura condivisa
IPsec, GRE, MPLS

Usi di VPN

- | | |
|-------------------|--|
| Scenario Intranet | rete privata aziendale (non vuol dire che usi per forza indirizzi privati) in tecnologia IP. |
| Scenario Extranet | rete a cui solo alcuni possono accedere ma non fanno parte di una sola organizzazione. Esempio scenario tra fornitori e clienti che condividono accesso ai server. |

Il FireWall fa il forwarding solo di alcuni pacchetti, è una specie di router con delle regole basate sul contenuto delle intestazioni.



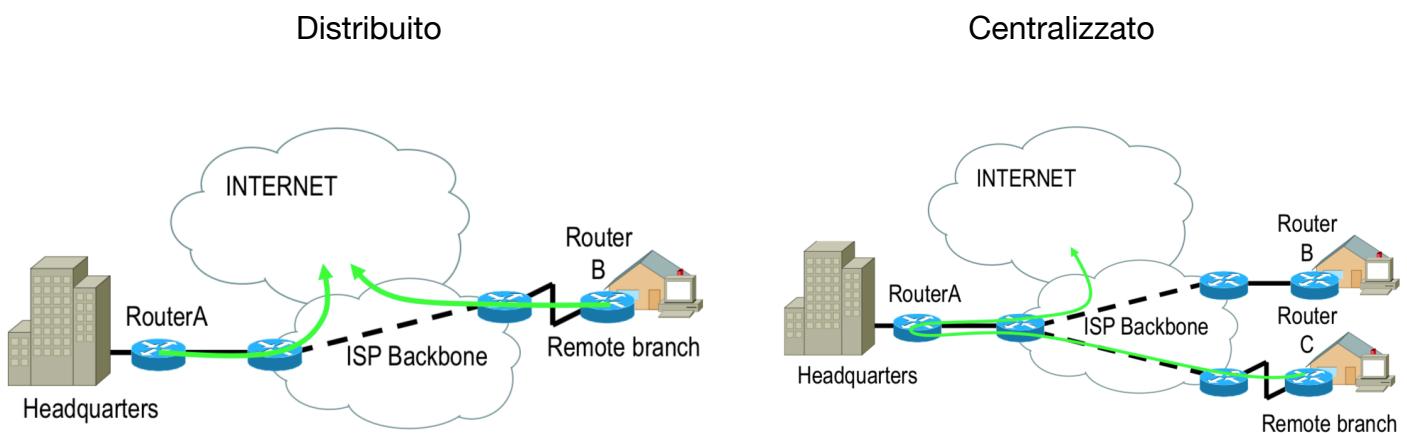
Nello scenario extranet abbiamo bisogno di un fireWall all'ingresso della VPN perchè non tutti devono poter accedere.

Distributed/centralized internet access

Abbiamo siti aziendali collegati attraverso infrastruttura condivisa. Se ho del traffico che deve andare ad un certo server in Internet lo faccio andare direttamente dalla

sede remota (accesso ad internet distribuito) o lo faccio andare prima alla sede principale e poi da lì a internet? (accesso ad internet centralizzato).

Centralizzato	Usando accesso centralizzato usiamo un solo firewall per la sede centrale e le sedi dislocate. La VPN in questo caso è usata sia per trasportare traffico aziendale sia traffico internet.
Distribuito	Usando accesso distribuito devo mettere firewall in sede centrale e in ogni sede dislocata. La VPN in questo caso è usata solo per trasportare traffico aziendale dato che l'accesso ad internet è consentito direttamente ad ogni sede dislocata.



Deployment model

Overlay	<p>la rete pubblica non partecipa alla realizzazione della VPN. Non sa dove siano le destinazioni VPN, permette solamente il collegamento dei vari VPN gateways.</p> <p>Ogni VPN gateway deve essere in collegamento con tutti gli altri, si avrà quindi una maglia di tunnel molto densa.</p> <p>Il routing viene effettuato dai VPN gateways.</p>
Peer	<p>Ogni VPN gateway comunica con un router della rete pubblica scambiandosi informazioni di routing.</p>

Provisional Model

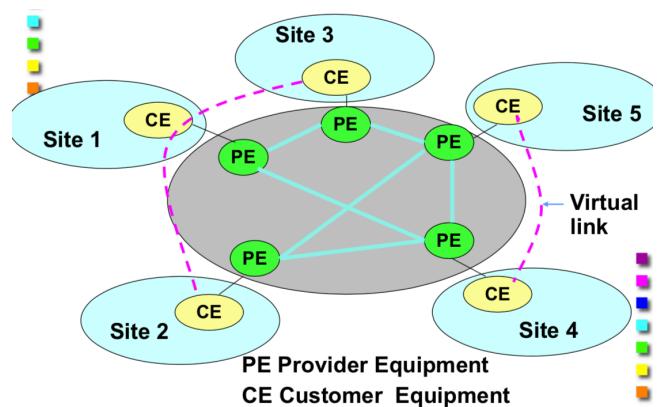
Customer provisioned	<p>l'azienda implementa la VPN e installa i propri dispositivi. Il service provider pubblico non è a conoscenza che il traffico generato dal cliente sia generato da una VPN. Tutte le funzionalità della VPN vengono implementate</p>
----------------------	--

all'interno dei dispositivi aziendali. Il CE sarà il terminatore di tunnel.

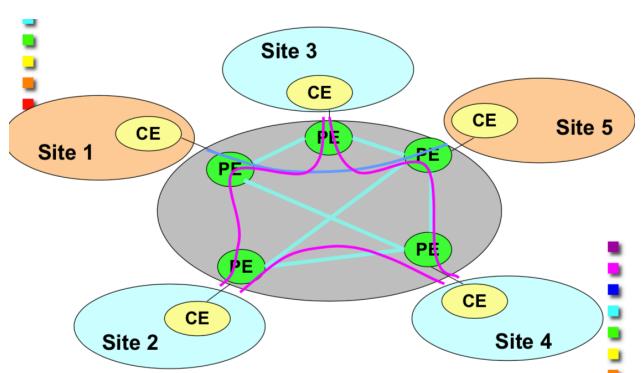
Provider provisioned

è il provider ad implementare la VPN. Possiede e gestisce i dispositivi che implementano funzionalità VPN. Il traffico che appartiene a VPN diverse viene separato dai dispositivi del service Provider. Il PE sarà il terminatore di tunnel.

Customer provisioned



Provider Provisioned



Componenti principali

Tunneling diversi protocolli di tunneling

Cifratura per cifratura pacchetti

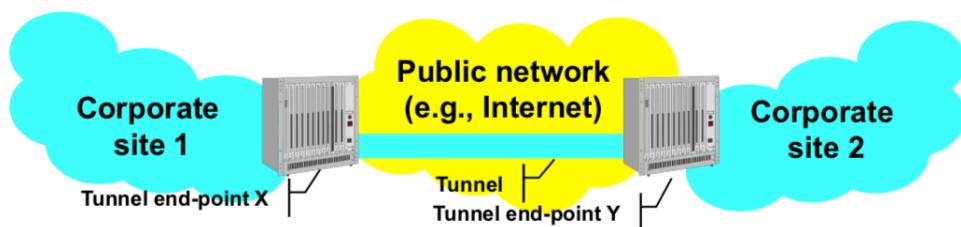
Integrità evitare che qualcuno modifichi i dati

Autenticazione autenticare endPoint VPN.

Fanno tutti parte di meccanismi di sicurezza

Tunneling

Pacchetto tra reti private che viene imbustato dal VPN gateWay è trasportato su un infrastruttura condivisa.



Topologia dei Tunnel

Mesh	magliatura di tunnel. Richiede di creare tanti tunnel tra ognuno dei terminali senza passare dalla sede centrale, problematico perchè i tunnel vengono creati manualmente. Il routing però è ottimizzato.
Hub and Spoke	Si collegano i vari VPN gateway tutti verso la sede principale. Problema congestione del traffico verso sede principale, maggior carico di lavoro nella sede centrale. Si riduce però il numero di tunnel. Routing non ottimale.

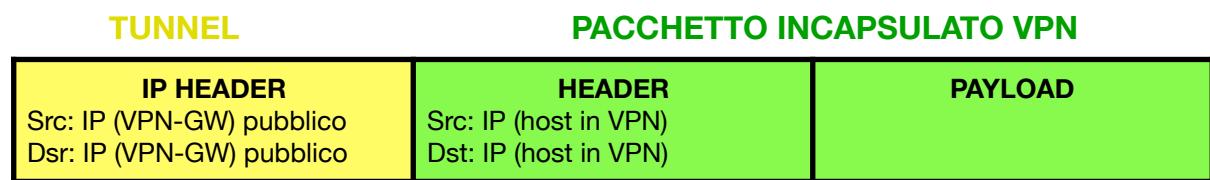
Layer N VPN

Prima abbiamo definito le VPN in base al livello diverso di protocollo usato.

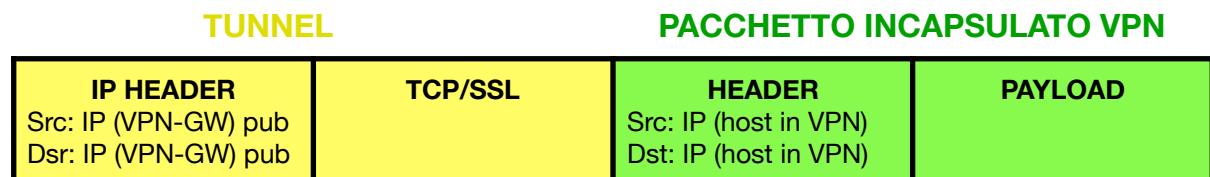
Soluzione di livello 2	mi permette di trasferire tramite ethernet, emulo il funzionamento delle LAN.
Soluzione di livello 3	nel trasferire le trame ethernet le mette dentro pacchetti IP.

Tunnel layer 3 VPN

Un pacchetto è trasportato attraverso una rete ip pubblica all'interno di un pacchetto IP. Il pacchetto che deve essere imbustato può essere ethernet o IP (si fa tunnel IP-in-IP).



Tunnel layer 4 VPN

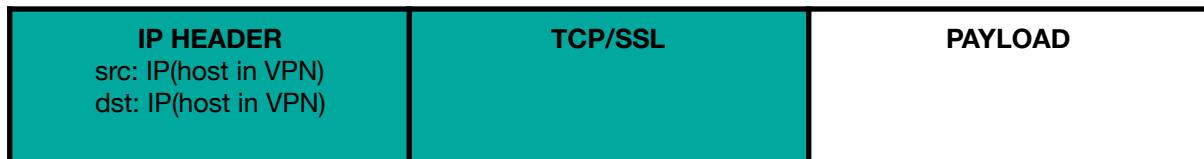


Il tunnel viene realizzato grazie ad una connessione TCP tra VPN-GateWay.

Per ottenere sicurezza uso SSL/TLS.

Il tunnel potrebbe essere anche creato non tra VPN gateWays ma direttamente tra host, in questo caso.

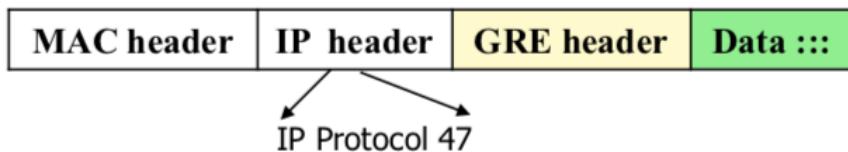
TUNNEL



GRE

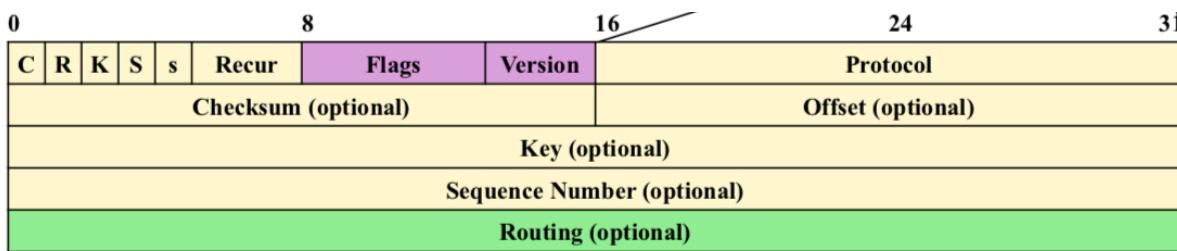
Generic routing encapsulation. Protocollo usato per fare incapsulamento e routing dentro IP. Si mette pacchetto qualsiasi dentro pacchetto IP.

Nel campo protocollo dell'intestazione IP scrivo 47 che mi identifica l'intestazione GRE.



Nell'header GRE c'è un campo di due byte che mi permette di descrivere che pacchetto sto incapsulando dentro IP. (protocol)

Sequence number ethernet di suo garantisce l'ordine dei pacchetti. Se però usando GRE incapsulo ethernet dentro IP devo sempre garantire ordine dei pacchetti tramite sequenceNumber.



Con il GRE posso specificare sequenza di tunnelEndPoint attraverso cui il pacchetto GRE deve passare.

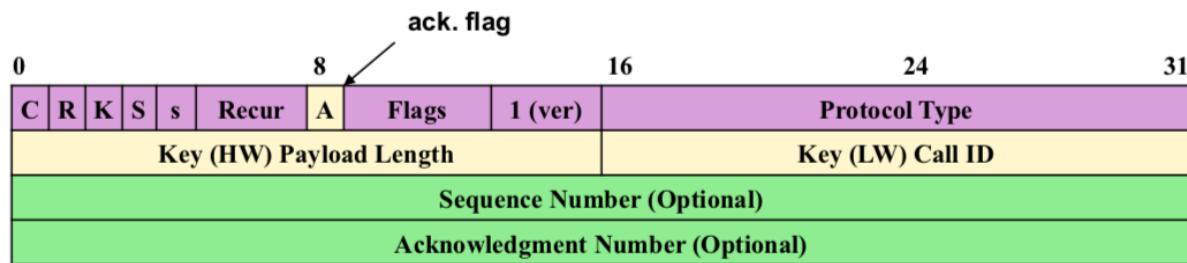
C R K S flag che indicano la presenza o assenza di campi opzionali
s stricto source routing flag, se la destinazione non è stata raggiunta

dopo aver finito la lista di nextHop da attraversare il pacchetto viene scartato.

Protocol	ID del protocollo usato nel payLoad
Routing	sequenza di nextHop da attraversare per source routing. Contiene <ul style="list-style-type: none"> - SRE offset offset del corrente indirizzi IP - SRE length lunghezza totale degli indirizzi presenti nella lista

Esiste una nuove versioni di GRE che aggiunge un campo

AckFlag serve per controllo di flusso, mi assicuro di non mandare troppi pacchetto rispetto a quelli che il ricevitore può ricevere.



Sicurezza

Tre obiettivi

- autenticare l'endPoint della comunicazione.
Voglio essere sicuro di comunicare con il mio endPoint VpnGateway.
- 1. AUTENTICAZIONE**
- consolidare integrità dei dati, i dati che invio non devono essere modificati lungo il percorso.
- 2. INTEGRITÀ**
- confidenzialità, se qualcuno su infrastruttura condivisa vede i pacchetti che invio voglio fare in modo che non capisca cosa ci sia scritto. Criptografia, si usa degli algoritmi pubblici che usano delle chiavi segrete.
- 3. CONFIDENZIALITÀ**

La criptografia posso usarla per

- firmare i miei pacchetti, se qualcuno cambia i dati lungo il percorso la verifica della firma fallisce e riconosco la modifica. FIRMA
- Cittografarli per non essere letti.

Algoritmi simmetrici o a chiave condivisa

La stessa chiave viene usata sia per la firma/cifratura che per la decifratura/verifica.

Problema della condivisione della chiave nel caso che i due comunicanti non siano stati gestiti in precedenza e ad entrambi è stata assegnata la stessa chiave.

Difficile quindi la condivisione della chiave tra i due comunicanti perché la chiave deve rimanere segreta.

Diff-helmann due entità riescono a mettersi d'accordo su una chiave scambiandosi informazioni che se anche venissero lette da qualcun altro questo non riuscirebbe a indovinare la chiave.

Algoritmi antrisimmetrici

La chiave usata per cifratura/firma è diversa da quella usata per decifrare/verificare.

Se voglio mandare messaggio a qualcun altro uso una certa chiave. Quando il messaggio arriva il ricevente per decifrarlo usa una chiave diversa rispetto a quella usata dal mittente.

Le due chiavi sono diverse ma comunque legate, perché generate dallo stesso algoritmo.

Quindi ho una chiave pubblica che mi serve per inviare pacchetti che tutti sanno ed una chiave privata, che so solo io per decifrare.

Certificati

Documenti che servono per verificare la proprietà di una chiave pubblica.

Il documento deve contenere

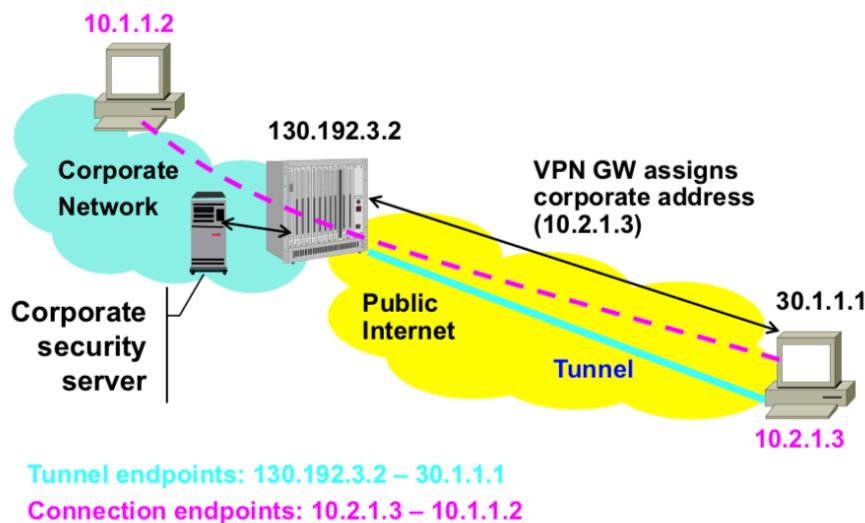
- “Nome” del proprietario
- la chiave pubblica
- firma di una certification Authority.

Se devo verificare una chiave devo avere la chiaveRoot della certificationAuthority. Posso ottenerla presentandomi di persona ma in realtà è pre caricata nei nostri web browser.

(Le chiavi private per mantenerle sicure si mettono nelle smartCard, sono dei processori che si occupano della cifratura della chiave).

VPN DI ACCESSO

Customer Provisioned



L'utente aziendale ha nel suo laptop il clientVPN che fa da tunnelEndPoint.
VPN tra dispositivo aziendale e VPN gateWay dell'azienda.

VPN gateway deve verificare autenticità dell'host che vuole comunicare con lui.
Deve verificare che quell'host abbia una chiave (chiave condivisa o segreta con cui i due si sono messi d'accordo oppure token ecc). L'host per inviare la password al VPN gateway può usare challenge handshake, il VPN gateway manda una sfida (sequenza casuale di byte) e l'host deve cifrare la sfida usando la propria chiave.

Si ha una sessione di negoziazione in cui si scelgono diversi parametri come gli algoritmi da usare per la cifratura ecc. Viene eseguito dal clientVPN, modulo software all'interno dell'host remoto.

Prima fase della sessione è l'autenticazione.

Usando gli indirizzi che il serviceProvider ha dato al VPN gateway e all'host i due dispositivi possono comunicare.

Le chiavi sono salvate all'interno del corporate security server che contiene tutte le password degli utenti aziendali.

VPN gateway usa protocollo particolare, **RADIUS**, per verificare le credenziali di un utente. Se utente autorizzato ad accedere alla VPN allora il server da l'ok al VPN gateWay.

Inoltre il corporateGateway deve fornire un indirizzo aziendale al client perchè all'interno della azienda devo dividere le varie unità (accounting, management, HR ecc) e l'host che vuole accedere deve ottenere un indirizzo specifico per il suo ramo aziendale.

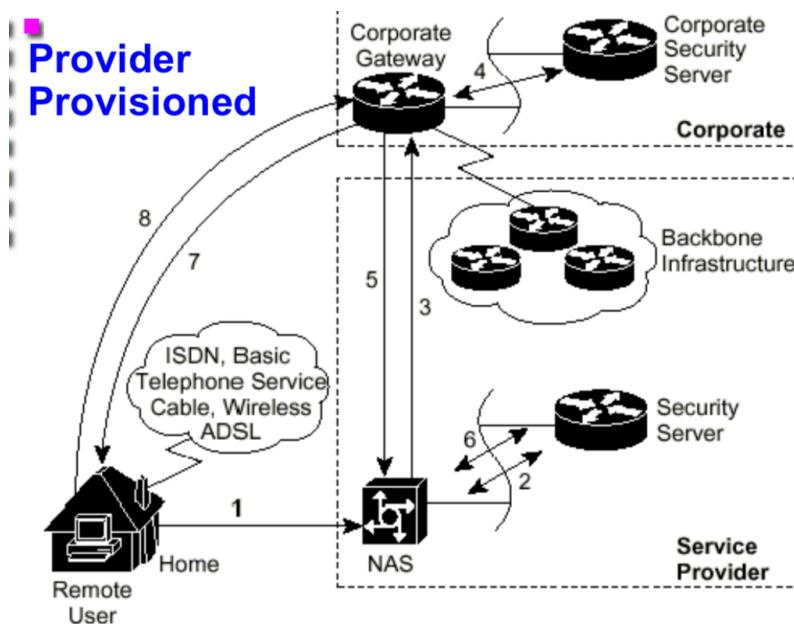
Si ha quindi un protocollo per l'assegnazione dinamica di un indirizzo aziendale all'host che vuole accedere alla VPN.

Il security server viene chiamato anche server AAA

- Authentication perchè autentica l'utente che vuole accedere
 - Authorization capiamo cosa quell'utente è autorizzato a fare, gli diamo dei parametri tipici del suo ruolo: indirizzi, algoritmi per cifratura ecc.
 - Accounting Il server tiene traccia che a quella particolare ora l'utente si è collegato alla VPN utilizzando uno specifico indirizzo.

Gli utenti aziendali che ricevono pacchetti da fuori non sanno che arriva da un host fuori dalla rete aziendale.

Provider Provisioned



Il servizio VPN viene offerto dal service provider.

NAS network access server. Router che deve anche autenticare l'utente e assegnare indirizzo IP ad utente che si collega.

- LCP link control protocol. Protocollo per scambio di parametri che mi serve per far funzionare collegamento sul link tra casa e nas.
- IPCP parametro (indirizzo) che mi serve per usare IP su link tra casa e nas.

Soluzione VPN

L'utente deve verificarsi al serviceProvider tramite NAS e protocollo di negoziazione. Il serviceProvider deve verificare se l'utente è autorizzato a collegarsi. Chiede al securityServer se utente abilitato al collegamento, il security server avverte che l'utente è aziendale e invia al NAS l'indirizzo del VPN gateway. Ora il NAS invia richiesta di autenticazione al VPN gateWay tramite la creazione di un tunnel L2TP o PPTP e il VPN gateWay lo passa al server aziendale, corporate security server (le password degli utenti aziendali vengono sempre gestite dal server aziendale). Il VPN GATEWAY poi passerà la risposta al NAS che la invierà al security server per registrare l'autenticazione o il traffico. A questo punto si può creare una negoziazione PPP tra l'utente e il VPN gateWay.

// NAS serve solo per inoltrare pacchetti al VPN gateWay,

Il mio Modem e il mio PC avranno indirizzi IP aziendali non del serviceProvider.

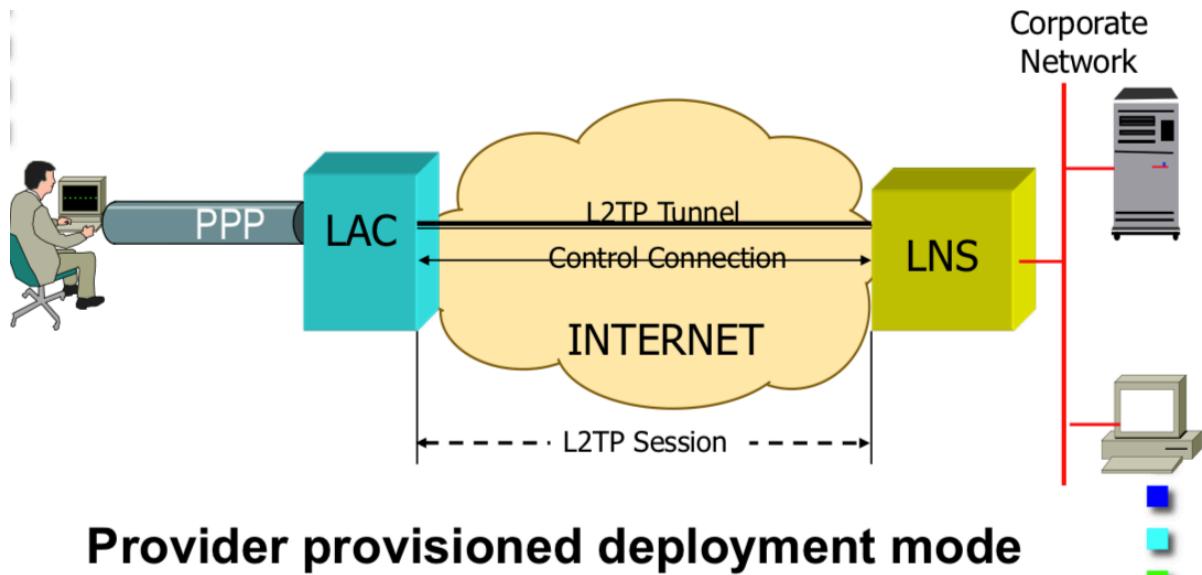
Differenze Customer/Provider Provisioned

CUSTOMER PROVISIONED	PROVIDER PROVISIONED
L'host remoto devono iniziare il Tunnel HOST -> VPN GATEWAY	L'host remoto non si deve preoccupare del tunnel perchè se ne occupa il serviceProvider
L'host remoto ha due indirizzi: uno interno alla VPN aziendale e un altro ottenuto dal serviceProvider	L'host remoto ha un solo indirizzo che è quello della VPN aziendale.
L'host remoto termina il tunnel VPN-GATEWAY->HOST	NAS termina VPN tunnel
La comunicazione può avvenire attraverso una qualsiasi connessione internet.	Richiede accesso ad uno specifico ServiceProvider.

Protocolli

- L2TP level 2 tunneling protocol.
- PPTP point-to-point tunneling protocol, funziona solo su link in cui si usa PPP (tra client e NAS).

L2TP-Provider Provisioned



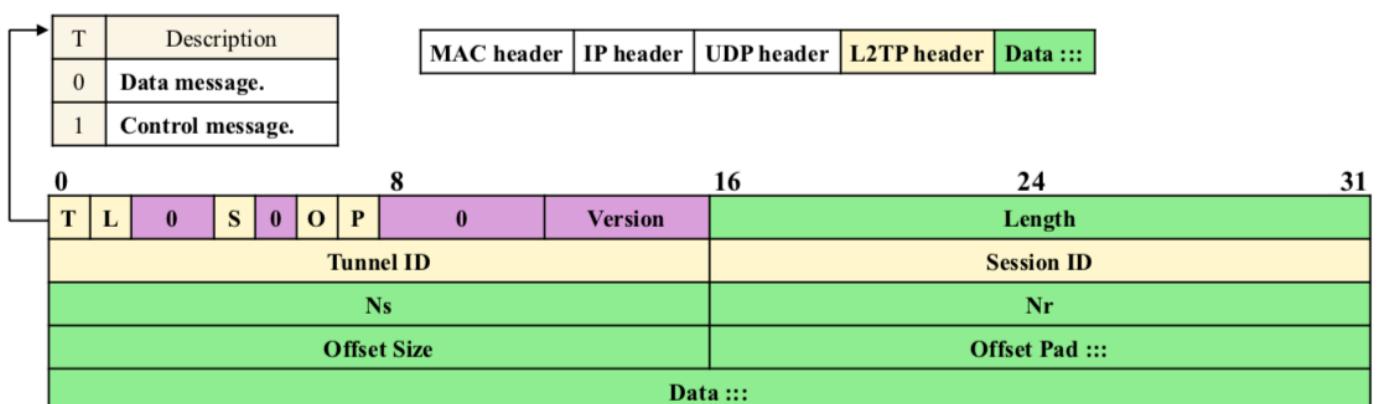
Ci sono due funzioni

- | | |
|-----|--|
| LAC | L2TP Access Concentrator.
Funzione che sta dentro il NAS nello scenario provider Provisioned. |
| LNS | L2TP Network Server , riceve tutti i tunnel de vari LAC.
sta nel VPN gateway. |

C'è un connessione di controllo che serve per negoziare i tunnel.

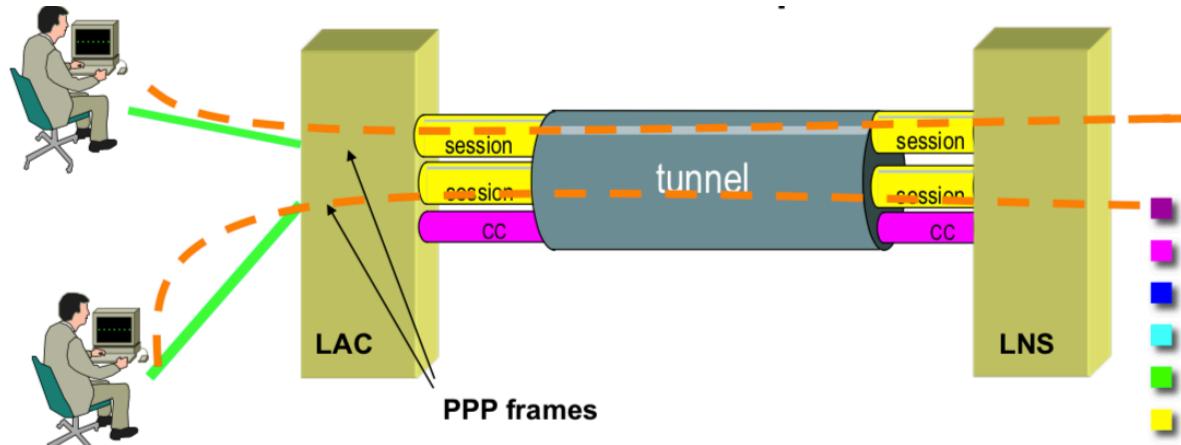
(Se fossimo in uno scenario customer Provisioned la funzione del LAC sarebbe interna all'host remoto.)

L'header L2TP imbustato in UDP. Il primo bit dell'intestazione mi dice se è un messaggio dati o un messaggio controllo.



All'interno di un tunnel ci possono essere diverse sessioni

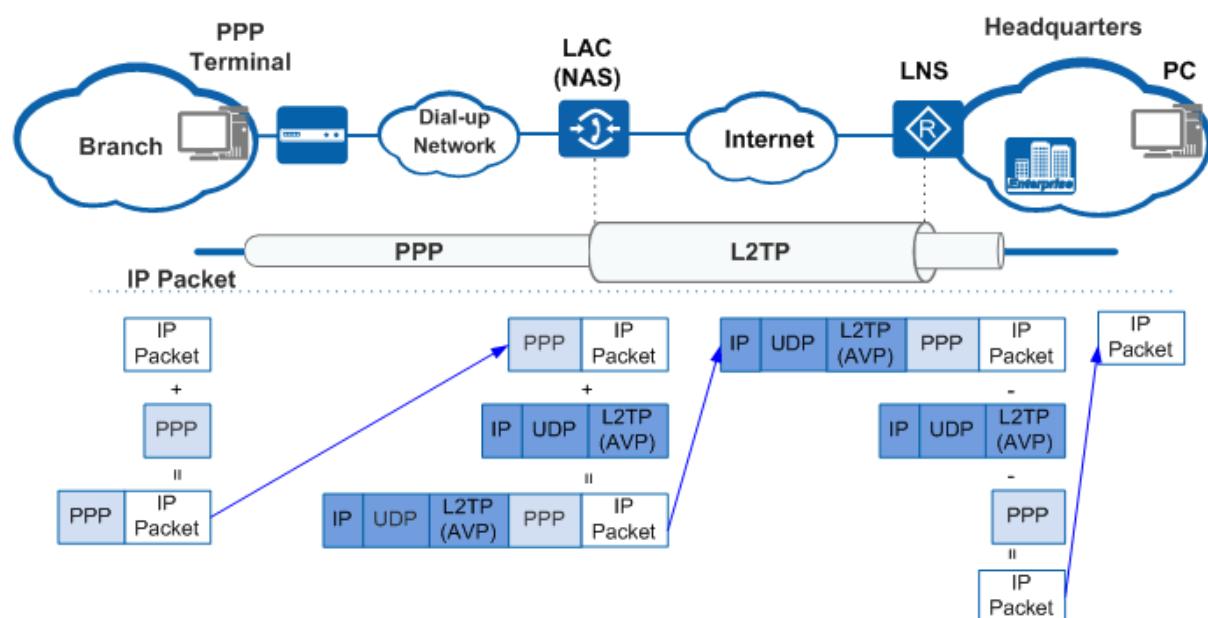
- tunnellID
- sessionID



Sequenza operazioni

- 1 si stabilisce una connessione di controllo tra LAC e NAS prima che venga effettuata una richiesta di connessione
- 2 si stabilisce una o più sessioni prima che i frame PPP vengano inviati tramite tunnel

La trasmissione dati non prevede ritrasmissione e ack. Solamente i pacchetti di controllo usano ack.



L2TP offre un servizio sicuro solo sul controllo, non sui pacchetti dati.

Quando il LAC stabilisce il tunnel con LNS si deve autenticare perchè LNS deve essere sicuro che il dispositivo che sta cercando di collegarsi (LAC) possa farlo. Si usano protocolli per autenticazione

- challenges CHAP

- Non c'è controllo autenticazione durante comunicazione:

Nonostante il LAC si autentichi la comunicazione non è sicura, non sono sicuro comunque che chi mi sta parlando sia davvero lui perchè l'autenticazione avviene solo in fase di creazione del tunnel, un altro utente potrebbero iniettare nel tunnel suoi pacchetti.

- Non c'è sicurezza a livello pacchetto:

Inoltre la cifratura l'autenticazione e l'integrità dei dati devono essere forniti da un meccanismo di trasporto come IPSEC.

- Non c'è autenticazione end-to-end (tra end host della comunicazione)

Anche l'autenticazione end-to-end deve essere fornita da IPSEC.

LNS non sa se i pacchetti che gli stanno arrivando siano davvero quelli del LAC. È normale sia così perchè la soluzione L2TP è stata ideata per essere implementata sulla rete del serviceProvider il quale con altre soluzioni la renderà sicura.

I'IPSEC è una soluzione a livello IP sicura a differenza di L2TP . IPSEC offre cifratura, autenticazione e integrità dei dati durante il trasporto.

Per rendere sicuro L2TP posso usarlo con IPSEC.

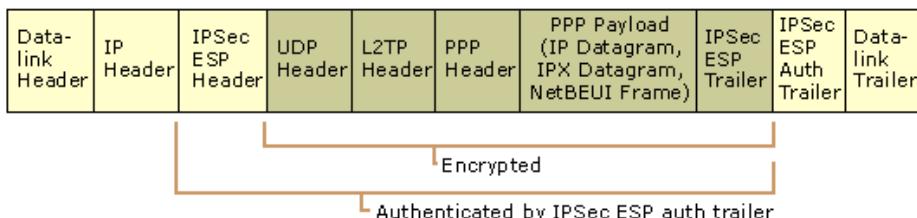
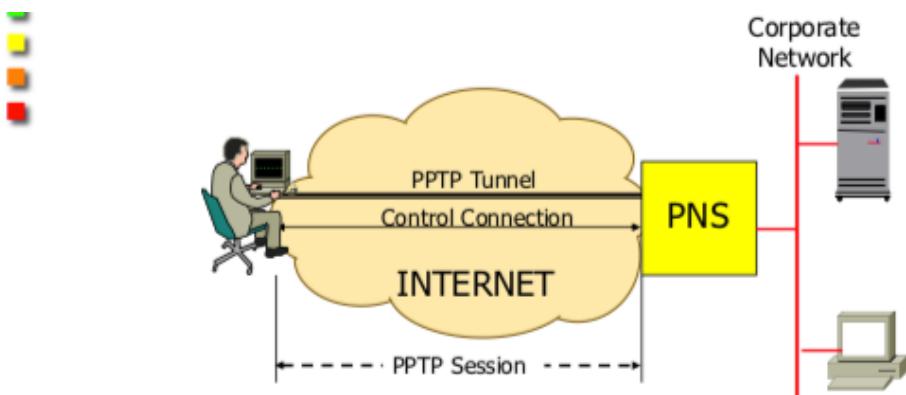


Figure 9.12 L2TP Packet Encapsulation

PPTP- Customer Provisioned



Customer provisioned deployment mode

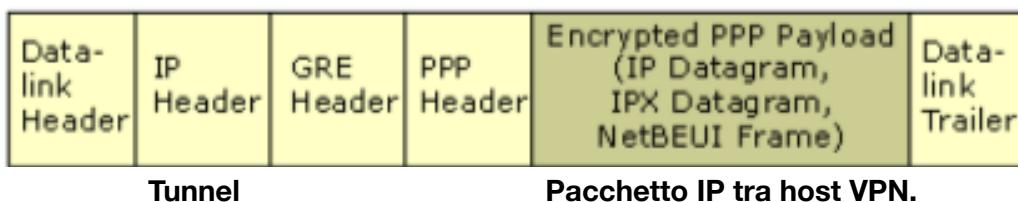
Customer provisioned. Il tunnel avrà una singola sessione perchè è tra un utente e il gateway.

C'è la necessità di autenticazione dell'utente, usando protocollo MS CHAP (autenticazione). Si applica anche la cifratura tramite MPPE (cifratura).

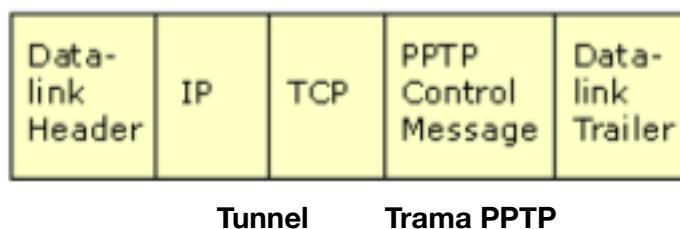
PNS **PPTP Network Server, VPN gateway nella modalità customerProvisioned**, riceve i tunnel dei vari host.

(Se usassimo PPTP con una VPN provider provisiooned il PNS si chiamerebbe PAC, **PPTP Access Concentrator**)

Inviando pacchetti dati si usa GRE per imbustare trame PPP in pacchetti IP.



Inviando pacchetti controllo si una encapsulamento trame PPTP in TCP. La parte di controllo avviene su connessione TCP tramite porta 1723.



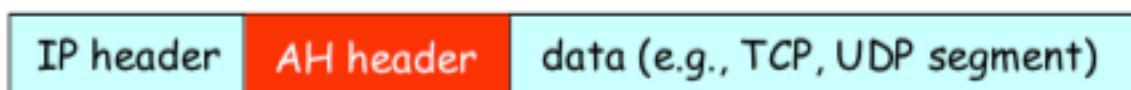
VPN SITE-TO-SITE

Vogliamo interconnettere diverse reti remote aziendali tramite infrastruttura condivisa.

AH-Authentication Header Protocol

Garantisce autenticazione della sorgente e integrità dei dati. Non garantisce confidenzialità.

L'header AH viene inserito tra l'Header IP e il payload.



AH header contiene

- SessionID
 - Come verificare la firma algoritmo di cifratura, chiave
 - Firma
 - Next header

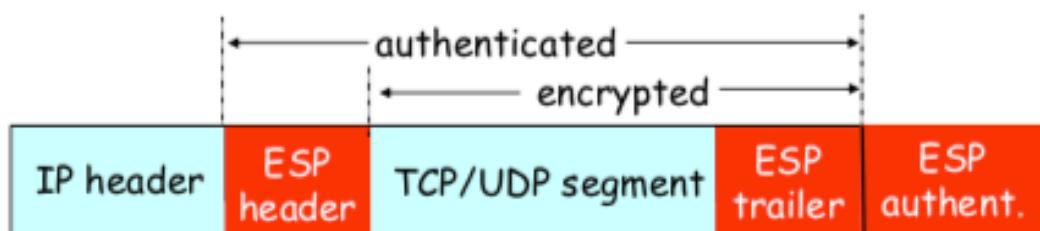
Quei due parametri fanno parte del campo SPI (Security parametri Index).

L'header permette di autenticare sia l'intestazione che il payload.

In AH header c'è un'indicazione (campo SPI) che è identificativo per uno scambio di pacchetti tra due host. L'SPI mi dice in che modo sto facendo l'autenticazione, che algoritmo sto usando, che chiave sto usando.

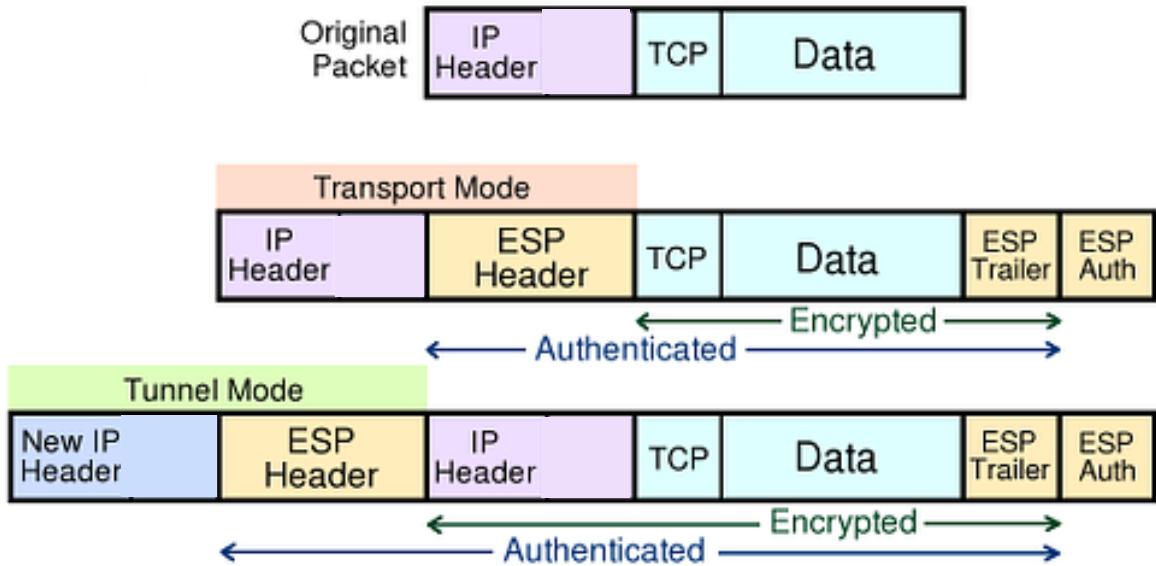
ESP-Encapsulating security PayLoad

Garantisce confidenzialità dei dati, autenticazione e integrità.



Data e ESP trailer vengono cifrati. **Viene autenticato solo payload.**

Se qualcuno cambia qualcosa nell'intestazione IP non me ne accorgo.



AH

e

ESP non posso usarlo se uso NAT, perchè AH cifra l'header mentre ESP cifra anche le porte.

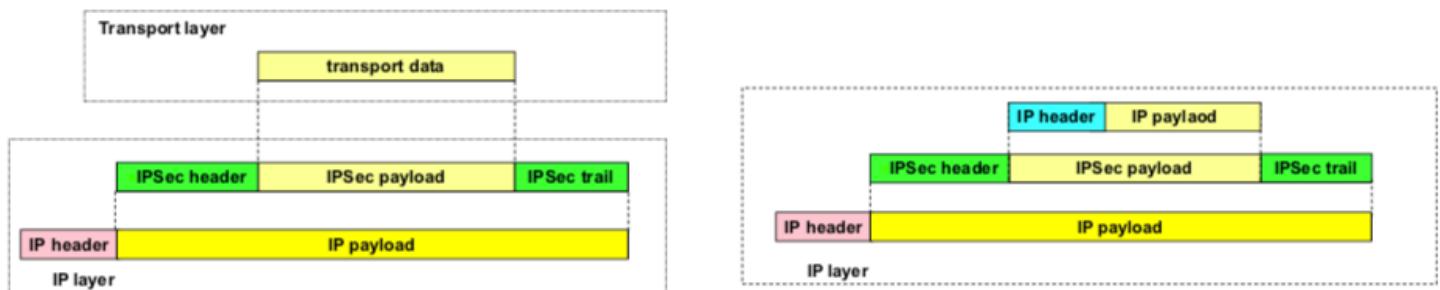
IPSEC

Il tunnel viene creato tra VPN gateway.

Protocollo che serve per scambiare pacchetti IP in modo sicuro cioè autenticare utente e destinatario, autenticare dati e cifrarli.

Vengono inseriti header aggiuntivi. Nel caso IPV4 gli header sono del protocollo IPSEC nel caso di IPV6 sono extension Header, cioè header IPV6.

IPSEC viene usato nelle VPN site-to-site tra i due gateway, imbustano il pacchetto IP dentro un pacchetto IP sicuro con intestazione AH o ESP.



IPSEC ha due modalità di funzionamento :

- Transport Mode Viene protetto solo Payload e non l'header IP.
- Tunnel Mode prendo pacchetti IP e li metto dentro pacchetti IP sicuri

e poi faccio tunnel senza usare più GRE perchè l'imbustamento è fatto da IPSEC stesso. Viene protetto sia l'headerIP che il payload.

Transport Mode

Tunnel Mode

ESP-Tunnel mode

L'ESP cifra solo il payload ma in questo caso come payload è considerato l'intero pacchetto IP che sto incapsulando dentro NEW IP HEADER (quello del tunnel)

SA - security Association

Gli SA sono degli insiemi di parametri che decidono le due stazioni che vogliono comunicare prima di iniziare a scambiarsi pacchetti IPSEC.

Devono decidere quali algoritmi e quali chiavi useranno per la cifratura mettendosi d'accordo.

Quando A vuole mandare pacchetto a B mi sono messo d'accordo che metto intestazione ESP (campo Protocol) e poi degli algoritmi e delle chiavi per autenticare e cifrare (Authentic e Encrypt). In questo modo quando B riceve pacchetto sa quali algoritmi deve usare per autenticare e decifrare.

IKE-Internet key exchange

Tramite protocolli IKE posso rinegoziare le chiavi di cifratura o più in generale gestiscono la negoziazione delle SA. Prima dello scambio gli host si scambiano messaggi previsti da protocolli IKE.

ISAKAMP protocollo usato per mettersi d'accordo sulle chiavi che useranno per creare SA per poi scambiare dati. **In questo modo se una sessione dura da un po di tempo può capitare che si veda rinegoziare le chiavi. MOLTO SICURO.**



- 1 vengono negoziati i parametri IKE
- 2 vengono scambiate le chiavi pubbliche
- 3 vengono scambiati i certificati e verificati
- 4 vengono scambiati dati firmati per autenticazione

SSL

- IPSEC complicato da usare.
- SSL meccanismo per rendere la comunicazione sicura a livello dei socket. Socket interfaccia che serve ad applicazione per scambiare informazioni sulla rete. Esempio le richieste HTTP sono imbustate dentro TCP. Quando browser vuole mandare richiesta HTTP facendo chiamata al socket il contenuto di quella chiamata venga trasmessa in modo sicuro sulle connessione TCP.
SSL prende la mia richiesta HTTP e la protegge mettendo una firma e cifrandola.

IP	TCP	Message Digest	HTTP
----	-----	----------------	------

Tunnel con SSL tra VPN gateway

IP tunnel	TCP	Message Digest	IP interno
-----------	-----	----------------	------------

Cifratura e autenticazione avvengono solo nel “payload” TCP, header IP e TCP non vengono toccati.

SSL si può usare per il tunnel tra VPN gateway di VPN site-to-site. Essendo un tunnel di livello 4 (TCP) si ha il un problema di fondo. A livello 4 abbiamo tutte quelle funzionalità come flow control e ritrasmissione che dovrebbero essere effettuato tra dispositivi end-to-end, cioè tra i due host che vogliono comunicare. Invece usando SSL tra i due VPN gateway queste operazioni vengono effettuare due volte , sia tra i due VPN gateway sia tra i due end host.

QUESTO PROVOCA PESSIME PEFORMANCE.

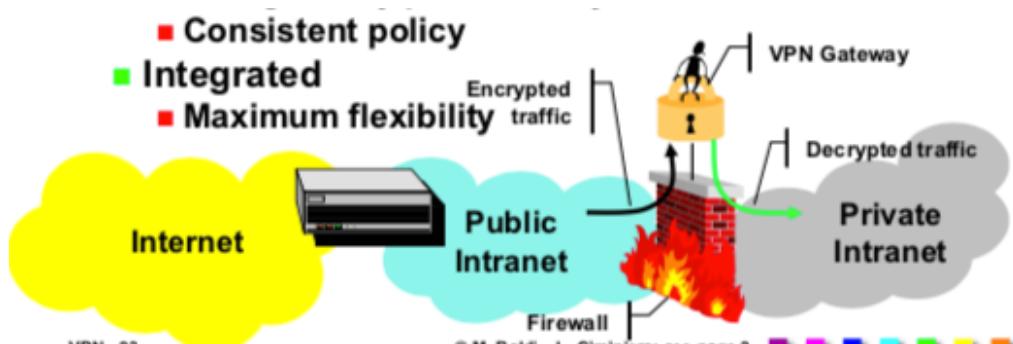
VPN GATEWAY POSITIONING

VPN gateway dovrebbe stare vicino al punto di accesso a internet. Deve essere sulla parte pubblica della rete aziendale, all'esterno del firewall.

Ma così sarebbe esposto agli attacchi informatici.

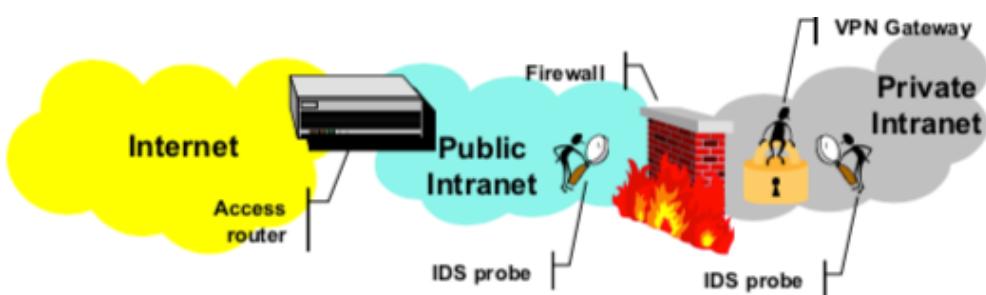
Se lo metto a valle del fireWall invece non ho controllo sul traffico che esce dal VPN gateWay quindi potrei propagare virus all'interno della mia rete privata aziendale.

Metto quindi VPN gateWay in parallelo al fireWall.



Intrusion detection System

cerca di identificare gli attacchi mentre fireWall cerca solo di bloccarli.



Si usano diversi IDS, uno esterno al fireWall e uno all'interno del VPN gateWay.

PEER VPN and MPLS-bases Solutions

PWE3 - MPLS based layer 2 VPN

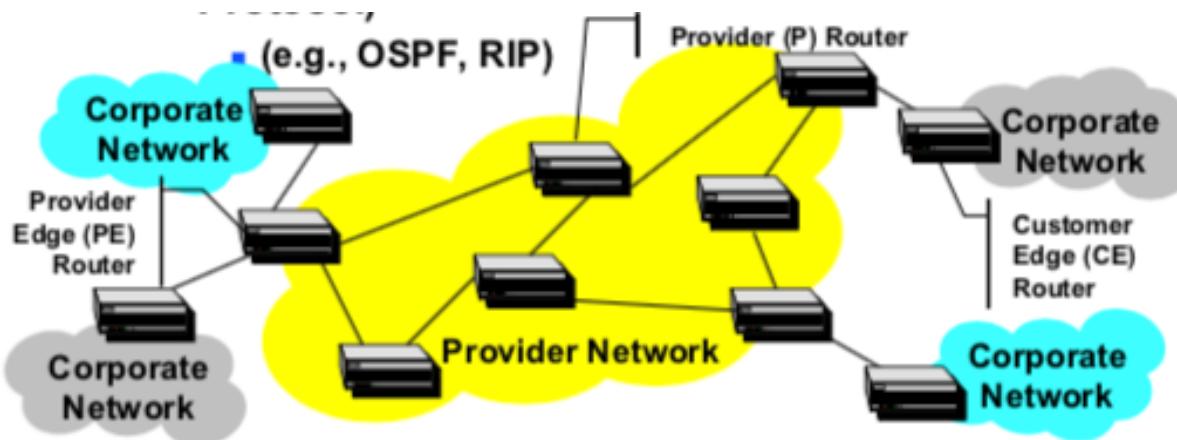
Pseudo wire Emulation end to end.

Stessa cosa della soluzione MPLS vista su backBone IPV6.

Il traffico viene inviato tramite LSP tra CE. Si utilizzano due etichette

- esterna per routing all'interno della rete
- interna per fare multiplexing di utenti e servizi allo stesso accessPoint.

Componenti VPN-MPLS based



CE

- annuncia le sue destinazioni
- riceve informazioni sulle router per raggiungere altre VPN

P

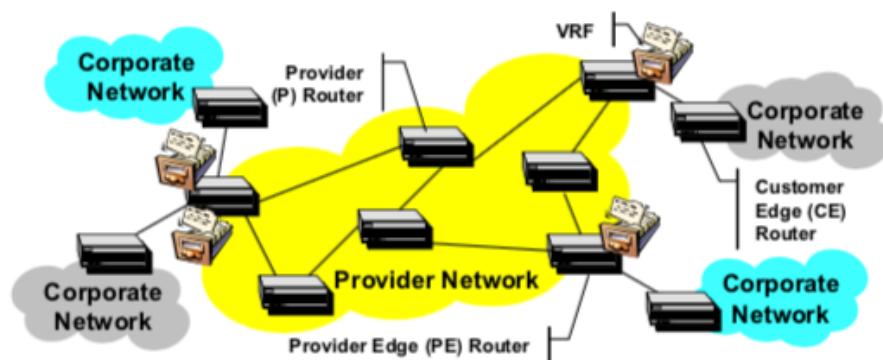
Sono i router che stanno all'interno del backBone.

- hanno le routes per raggiungere tutti i PE

PE

- instaurano LSP tra altri PE
- mantengono le route solo per le VPN connesse a loro

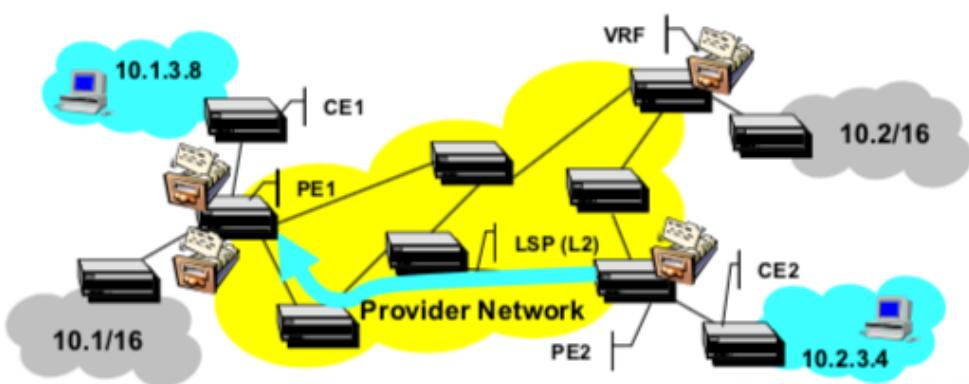
Componenti VPN-MPLS/BGP based



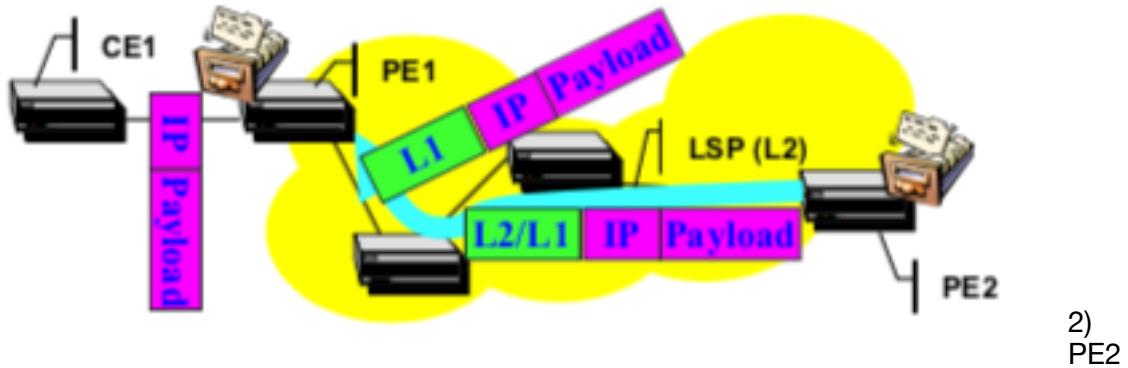
VRF VPN routing&forwarding table. Viene associata ad una o più porte. Mantiene le informazioni di forwarding per traffico ricevuto sulla porta a cui è associata.

Esempio di routing di un pacchetto:

Pacchetto da 10.2.3.4 a 10.1.3.8



- 1) PE2 guarda nella VRF di 10.2.3.4 per vedere una corrispondenza con 10.1.3.8 e vede che il next hop è PE1. Viene scelta l'etichetta L1 per la destinazione 10.1.3.8.



guarda nella main Table e vede che il nextHop è P1, viene scelta etichetta L2 (per creare LSP tra PE2-PE1). Il fatto di aver scelto P1 è definito dal punto prima.

- 3) Tutti i P router (quelli interni al backBone) inoltreranno il pacchetto usando l'etichetta esterna L2.
- 4) L'ultimo Hop prima di PE1 toglierà l'etichetta L2 (PHP).
- 5) PE1 a questo punto riceverà un pacchetto con etichetta L1 (quella interna che identifica la VPN) la toglie e inoltra pacchetto.

Vantaggi

- **indirizzi univoci solo all'interno della VPN**
- **CE router non scambiano informazioni tra di loro**
- **Customer non gestiscono il backBone**
- **Provider non hanno un backbone virtuale per ogni customer**
- **QoS supportato tramite bit sperimentali nell'MPLS header**

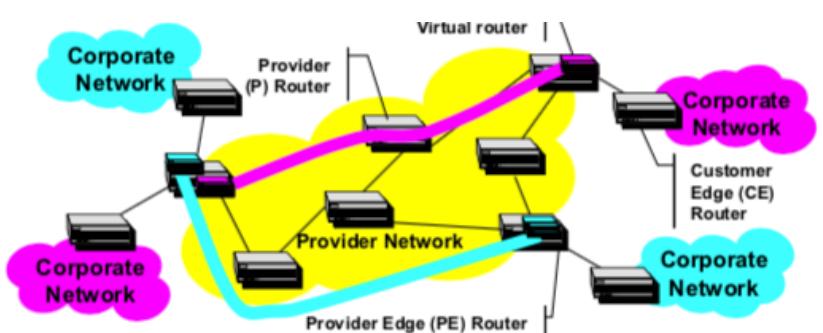
Si usano indirizzi VPN-IPV4 concatenando un routeDistinguisher all mio indirizzo IP. In questo modo come detto prima gli indirizzi devono essere univoci solo all'interno della VPN.

RouteDistinguisher	Ip Address
Identifica il colore delle reti ai bordi della rete	

Utilizzando MP-BGP, multi protocol BGP, si estende il supporto a indirizzi di famiglie diverse.

VPN-MPLS/Virtual Router based

I PE eseguono un'istanza di router virtuale per ogni VPN. Tutti i virtual router della stessa VPN comunicano grazie ad LSP.

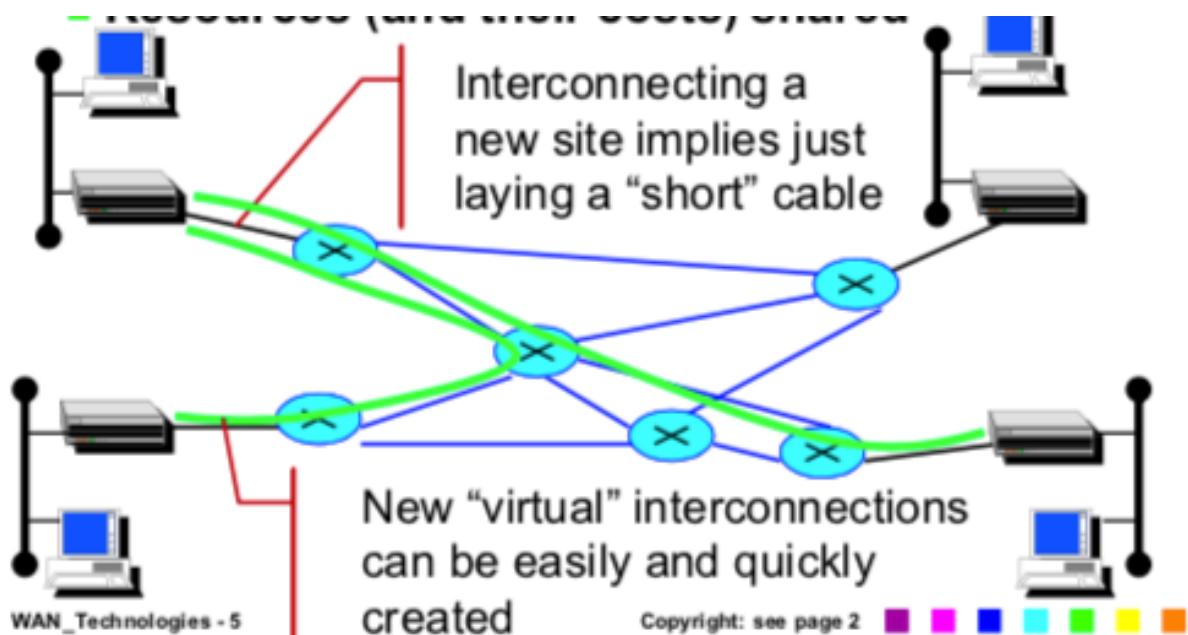


WAN TECNOLOGIES

Perchè non collego diverse reti direttamente con cavo? Perchè è molto costoso, richiede del tempo ed è molto statica.

Meglio creare infrastruttura centrale condivisa , in questo modo

- i cavi vengono posati una sola volta
- possibilità di configurazione per le connessioni
- risorse e costi condivisi



Prima di costruire un'infrastruttura nuova è sempre meglio usare quelle già esistenti.

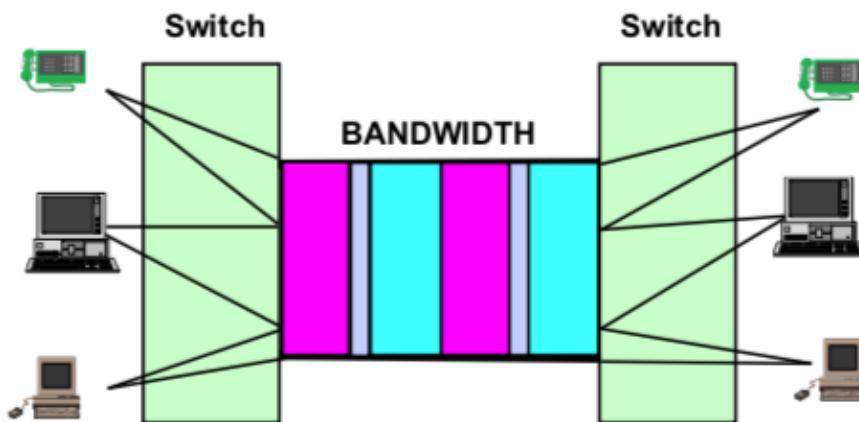
Tecnologie più diffuse

- **Commutazione di circuito** (time division multiplexing)
 - ISDN
 - PDH
 - SONET/SDH
- **Commutazione di pacchetto** (statistica multiplexing)
 - FRAME RELAY
 - ATM
 - TECNOLOGIA OTTICA

La più recente di tutte

TECNOLOGIE A COMMUTAZIONE DI CIRCUITO

TDM time division multiplexing, si suddivide la banda in slot temporali fissi in cui viene assegnata la banda a diverse comunicazioni per un certo intervallo di tempo .



Svantaggi

- **Se un router non manda pacchetti in rete nel suo slot temporale non si fa niente —> spreco di risorse.**
- **Poco flessibile**
- **Costosa perché la banda la paghi sia che la usi o che non la usi e anche perchè gli switch lavorano in base al tempo. Con link ad alta velocità gli intervalli di tempo sono piccoli e quindi switch devono avere precisione elevatissima. Devo sincronizzare tutti i nodi della rete.**

Vantaggi

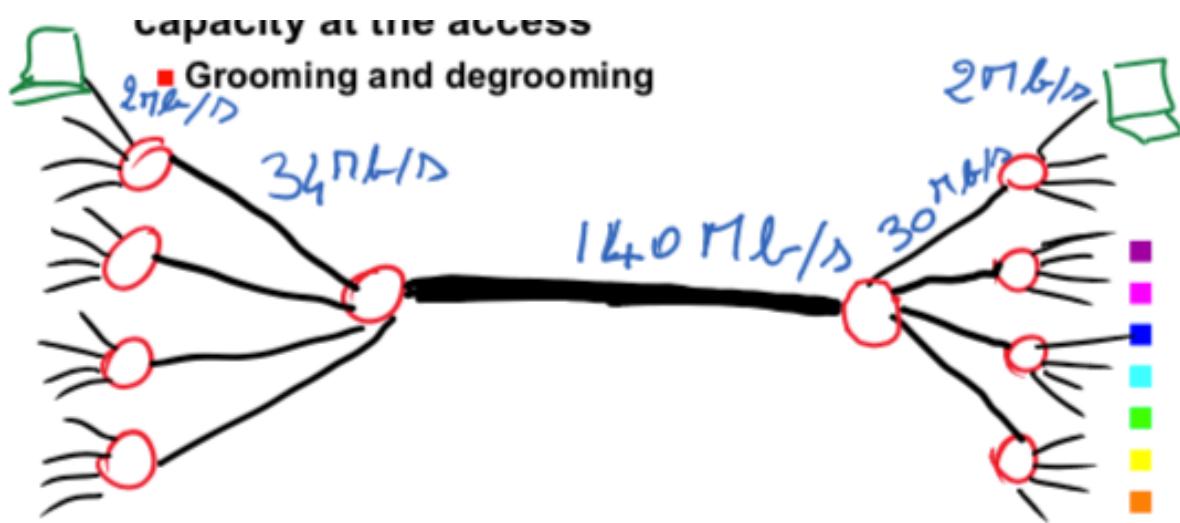
- **controlla molto bene variazioni di ritardo ma ci interessano poco alla fine. Se collegiamo router non ci interessano perchè i router mettono pacchetti nei buffer.**
- **Performance calcolata in modo deterministico**

Gerarchie di canali. Si definiscono velocità standard sui link tra due switch e sui canali.

Ci sono standard che definiscono come unire più canali a velocità bassa ad uno a più alta velocità. E anche il contrario, cioè come separarli. (multiplexing/demultiplexing).

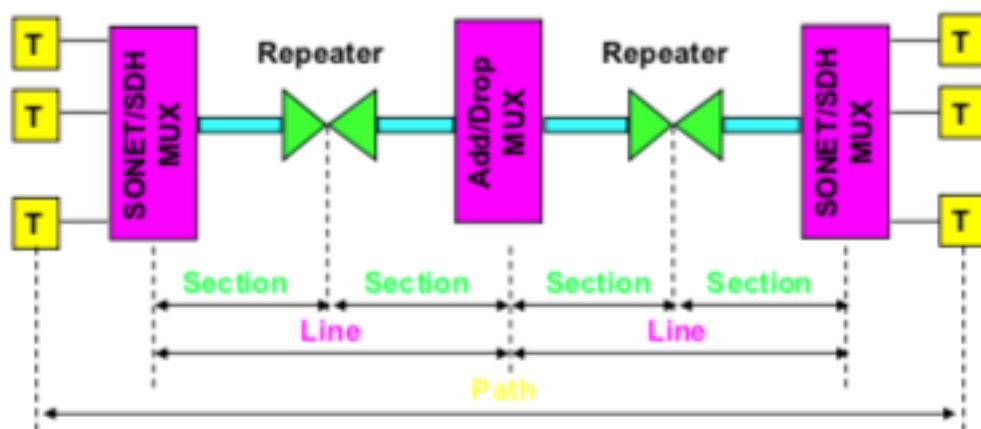
PHD nodi sincronizzati ma non in maniera ottimale. Non posso sfruttare quindi la massima velocità.

La gerarchia viene definita per fare grooming e degrooming dei canali, in modo tale da raggruppare canali più lenti in un unico canale più veloce. In questo modo posso avere backBone più veloci che si diramano alle estremità in link più lenti.



SDH/SONET

Si usa tecnologia ottica (fibra) per trasmettere, maggiore velocità di trasmissione.



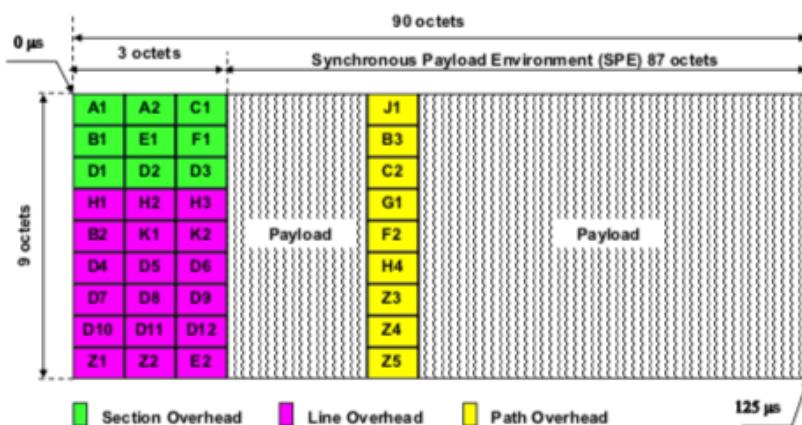
Section	Fibra ottica tra ripetitori
Linea	Sequenza di sezioni tra due dispositivi che operano a livello di linea.
Path	connessione end-to-end

Caratteristiche del protocollo

- diversi livelli, architettura del protocollo molto complicata
- monitoraggio delle funzionalità
- Fast valuto recovery

Un frame SDH/SONET è simile ad un pacchetto ma qui abbiamo un'intestazione e poi byte che appartengono a comunicazioni diverse se usiamo la commutazione a circuito

Gli switch riorganizzano i bit in dei frame per riconoscere a quale comunicazione appartengono. Il frame è lungo 810 byte. Il frame viene sempre trasmesso nella stessa quantità di tempo utilizzando la commutazione a circuito.



(Mentre nella comunicazione a pacchetto byte della stessa comunicazione sono nello stesso pacchetto con stessa intestazione.)

CONFRONTO	
COMMUTAZIONE A PACCHETTO	COMMUTAZIONE A CIRCUITO
Frame contengono byte di una singola comunicazione	Frames contengono bytes di più comunicazioni diverse
La destinazione è inserita nell'header	La destinazione viene identificata dalla posizione
Frame inviati asincronicamente ogni volta che ci sono dati da trasmettere e un link è libero.	Frames trasmessi sincronicamente
Multiplexing statistico	

Vantaggi

- **Tempo di trasmissione costante a 125 us**
- **più bytes per ogni frame a più alta velocità**
- **possibilità di sfruttare la gerarchia**

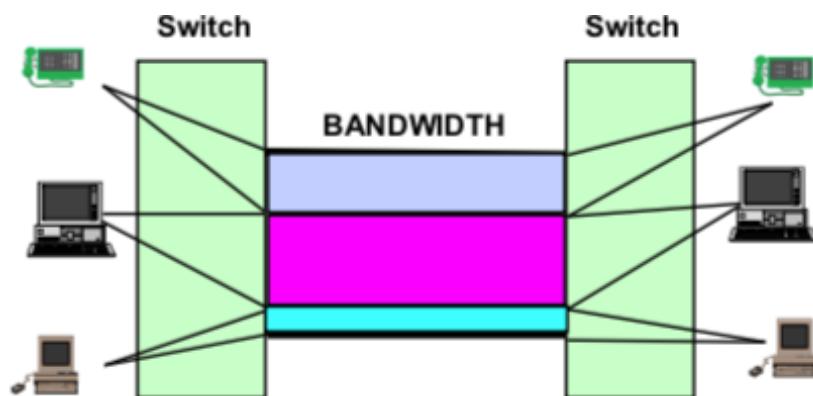
Svantaggio

- **commutatori SONET/SDH costosissimi**

Queste tecnologia viene usata perchè è usata semplicemente come tecnologia trasmissiva.

TECNOLOGIE A COMMUTAZIONE DI PACCHETTO

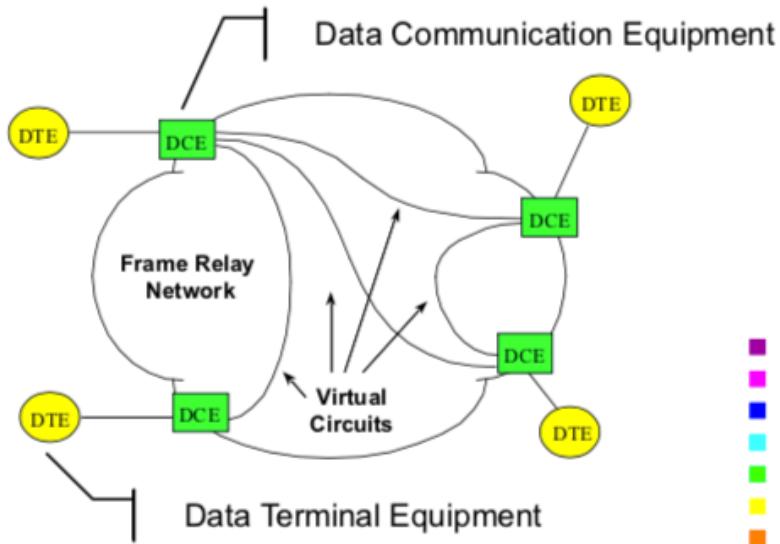
Commutazione di pacchetto, viene usata multiplazione statistica. Se una sorgente non invia traffico la banda non viene sprecata. L'infrastruttura può essere condivisa da più comunicazioni condividendo i costi. La performance però non è più deterministica.



Quindi se un utente non occupa la banda può essere usata da un altro.

La tecnologia a commutazione di pacchetto può essere implementata sopra quella a commutazione di circuito.

Frame relay



Tecnologia a pacchetto per reti pubbliche dove si creano dei circuiti virtuali tra gli apparati di accesso. Nel nostro contesto i DTE sono dei router. Invece che mettere fibra tra i due router metto fibra tra Router (DTE) e DCE e tra i vari DCE creo circuiti virtuali. Il pacchetto IP inviato dal ruoter verrà imbustato in una trama frame Relay di livello 2 che contiene l'identificativo del circuito virtuale a cui appartiene.

Il link tra DTE e DCE è un semplice link PDH, l'interfaccia del router (DTE) per interconnetterlo al DCE non è costosa.

CIR committed information rate. Il service provider si impegna a trasportare una certa quantità di traffico tra i due punti del circuito virtuale ad un certo bit rate. Devo considerare anche la committed burst size, cioè qual'è la massima quantità di traffico che un terminale può inviare nella rete alla velocità del suo link prima che la rete li scarti.

Vantaggi

- **Correzione di errori con ritrasmissione ma sono agli estremi della rete, non tra nodi intermedi**
- **Latenza minima**
- **Possibilità di specificare la banda da dedicare ad un cliente**

Svantaggi

- **Problemi per trasmissione voce/video**

ATM - Asynchronous Transfer Mode (ATM)

Caratteristiche

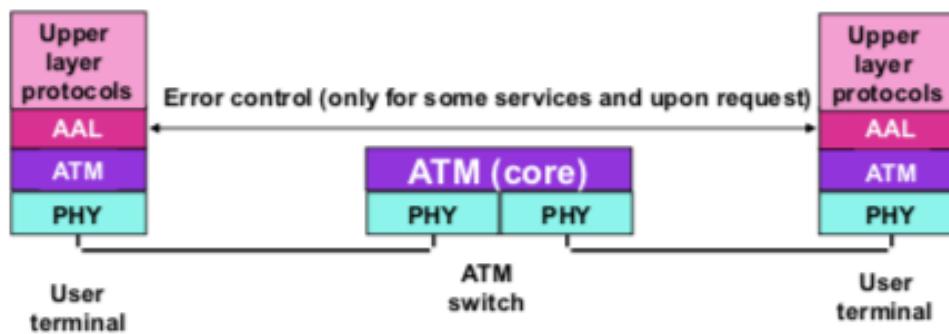
- Cella di 48 byte (non di più per evitare ritardi nella comunicazione, cioè cerco di non accumulare troppi campioni vocali in un'unica cella) + 5 byte intestazione = 53 byte. Quindi cella piccola e di lunghezza fissa.
- link molto veloci con basso bit-rate di errore (per l'anno in cui è stato progettato)
- bassa latenza
- Segnalazione e flow Control sofisticati (piano controllo complesso)
- Allocazione dinamica della banda
- Adatto per applicazioni che sono sensibili ai ritardi o alla perdita dati

Caratteristiche tecniche

- Celle trasmesse back-to-back con la possibilità di inserimento di celle vuote
- Ogni cella contiene VCI e VPI (che insieme costituiscono un'etichetta)
- Correzione errore come in frame Relay (Approccio Core-Edge)
- Flow control sofisticato, migliore rispetto a sliding windows.

Approccio Core-Edge

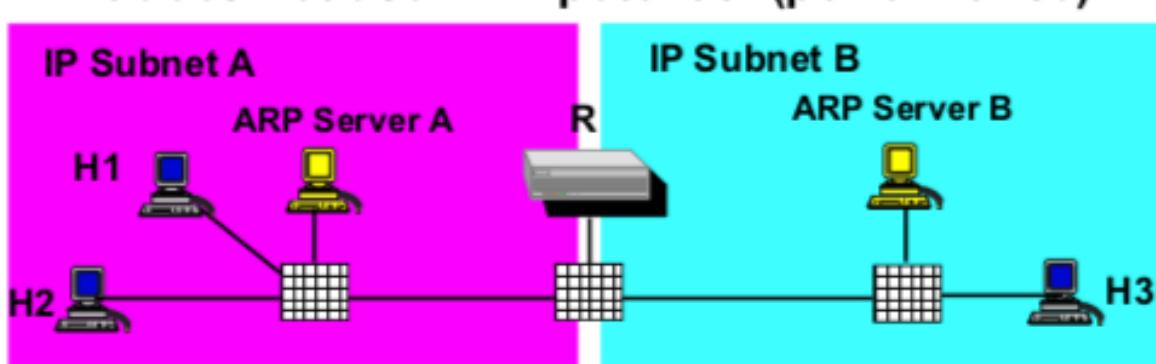
Viene usato anche in frame Relay. I nodi intermedi eseguono solo funzionalità di base come switching e multiplexing mentre i nodi al bordo della rete hanno funzionalità aggiuntive. Si semplifica il piano dati all'interno della rete ma si rende più complesso il piano di controllo per tutte le funzionalità aggiuntive ai bordi della rete.



Principale problema

Come fa un router IP a sapere l'indirizzo di un altro router IP dalla parte opposta della rete ATM , se il backBone è ATM? I commutatori ATM non posso aiutarlo perché non gestiscono indirizzi IP. Sono state proposte allora diverse soluzioni come quelle elencate qui sotto ma alla fine l'unica soluzione migliore è stata quella di usare MPLS. In questo modo router e ATM/MPLS switches scambiano informazioni di routing.

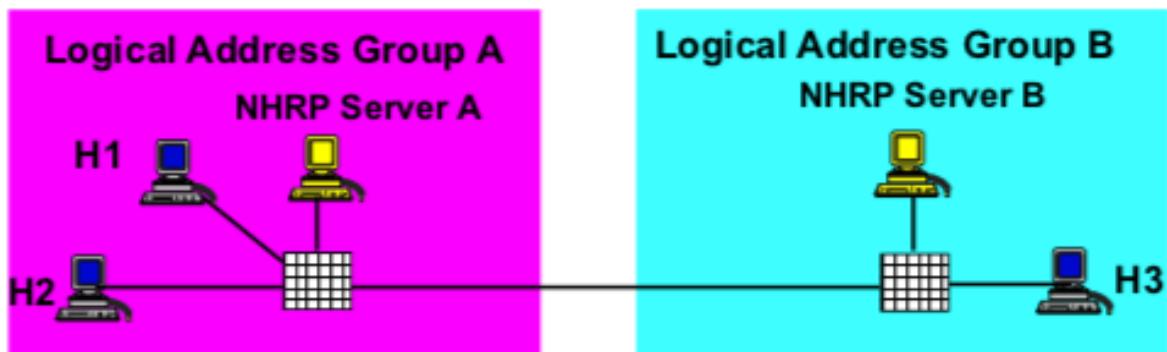
IP over ATM - Classico



Comunicazione diretta all'interno della stessa subnet. Si usano router per connettere più Subnet. Si emula ARP per l'address resolution all'interno della stessa subnet in modo da trovare l'indirizzo ATM della destinazione o del router.

IP over ATM - soluzione avanzata

Si utilizza Next Hop Resolution Protocol. La sorgente trova l'indirizzo ATM nel best hop per una destinazione. La destinazione deve essere sulla rete ATM altrimenti se è fuori dalla rete ATM si utilizza il Router.



Soluzione complicata.La soluzione migliore è far parlare la stessa lingua ai router e agli switch ATM → soluzione MPLS. Router e ATM/MPLS switch si scambiano informazioni di Routing e le destinazioni vengono identificate con indirizzo IP.

RETI OTTICHE

WDM	Wave division multiplexing. avere la possibilità su una fibra ottica di trasmettere più segnali usando delle lunghezze d'onda diverse.
CWDM	poche frequenze, più economico, meno complicato
DWDM	molte frequenze, più costoso, più complicato

Con DWDM non viene aumentata la capacità della fibra ottica, che dipende dalle proprietà fisiche dei materiali, ma aumenta la capacità di sfruttarla al meglio.

Succesivamente è stato introdotto l'add/drop multiplexing.

Infine la sua finale funzionalità è stata quella utilizzata per le così dette (waveLength Switched networks). Viene introdotto l'Optical switching-WaveLength switching.

Optical Switching

Ha il vantaggio di essere relativamente semplice ed avere un basso costo di ogni bit trasmesso.

Switching Optical core

Sfrutta le caratteristiche fisiche del materiale per deflettere la luce dalla fibra in entrata a quella in uscita. Sfrutta la frequenza del segnale per fare la commutazione, non devo guardare intestazione pacchetti ecc.

Vantaggi commutazione ottica

- **bassi costo dei materiali (ho uno specchio non un circuito elettronico complesso).**
basso costo dei processi (una volta che la tecnologia sarà matura)
- **indipendente da segnale e bitRate , si ha un semplice specchio che deflette l'onda elettromagnetica a seconda della lunghezza d'onda indipendentemente dal fatto che la trasmissione avvenga a 100 Gb/s o ad 1 Tb/s.**
- Multi standard**
- **Basso consumo di energia**

Svantaggi

- **alto costo per la produzione (perchè la tecnologia non è ancora matura)**
- **alta attenuazione del segnale senza rigenerazione**

Switching Electrical core

Convertono un segnale ottico in uno elettrico. Vengono perse tutte le caratteristiche vantaggiose di quelli ottici, quindi

- Compattezza (più la trazione è veloce più veloce dovrà lavorare lo switch elettronico)
- alto consumo di energia
- alto costo

Nonostante questo, per ora i core elettrici sono meno costosi di quelli ottici e anche meno complessi.

In realtà oggi si fanno commutatori ottici che hanno un core elettronico. Il segnale in ingresso viene demultiplo e poi convertito in una sequenza di bit. Questo perché i commutatori completamente ottici sono più costosi oggi come oggi. In ogni caso la commutazione dello switch elettronico è molto più semplice di quella di uno switch SONET, commuta semplicemente bit da una porta ad un'altra.

Piano di controllo

Le principali funzionalità del piano di controllo sono

- scoperta delle risorse topologia, access Point e nodi della rete, utilizzo delle risorse.
- Gestione delle connessioni e segnalazione Lighpath setup, lighpath take down, lighpath modification
- routing distribuito
- Mesh/ring network protection and recovery
- stabilire classi di protezione

Protocolli utilizzati (Piano di controllo)

per scoperta risorse (Routing)

- OSPF
- IS-IS
- BGP

per segnalazione (lightPath setup)

- RSVP
- LDP

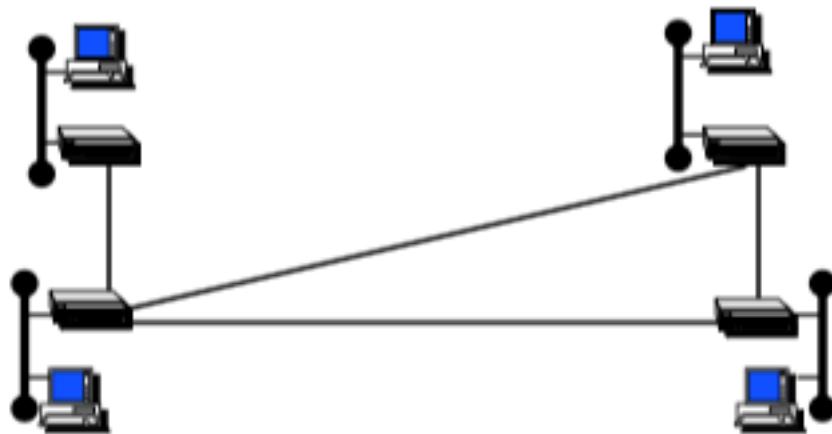
IP OVER

Diverse possibilità di interconnessione tra router.

Non si usa un semplice collegamento tramite cavo perchè:

- **costoso**
- **richiede molto tempo**
- **Poco flessibile, statica**

Fibra ottica



I router sono connessi tramite fibra, il segnale ottico trasmesso da un router è ricevuto direttamente da un altro.

La trasmissione a livello 2 è sincrona.

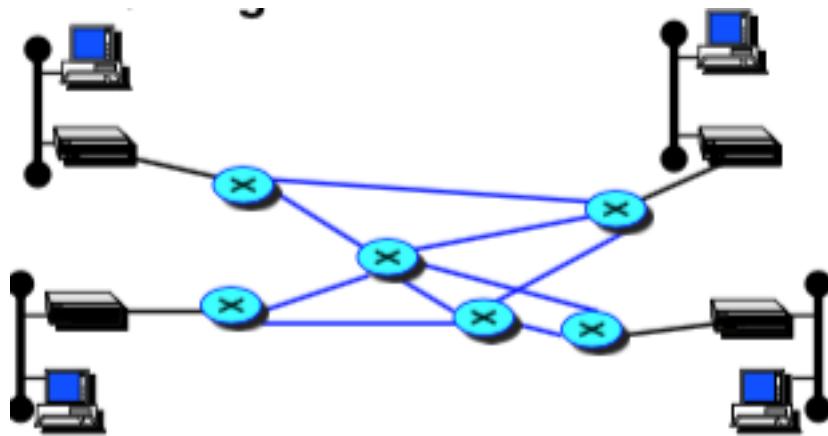
Vantaggi

- **No sovraccarico**

Svantaggi

- **Connessioni fisiche, statico. Non c'è possibilità di riconfigurazione**
- **una interfaccia per ogni connessione**
- **gestione complicata dell'infrastruttura**

DWDM (dense wavelength division multiplexing)



Diverse trasmissioni con differenti lunghezze d'onda (colori) trasmesse sulla stessa fibra. In questo modo si sfrutta nel miglior modo la capacità della fibra.
Avrà bisogno di dispositivi adatti per fare lo swtich di più segnali ottici (wavelength routing).

A livello 2 si usano trame PPP.

Vantaggi

- Alta capacità trmissiva
- Connessioni virtuali con optical routing

Svantaggi

- Complessa gestione dell'infrastruttura
- Multiplexing statico

SONET/SDH

Vantaggi

- Architettura OAM standard
- Connessione virtuali

Svantaggi

- Un'interfaccia per connessione
- multiplexing statico
- bassa flessibilità di configurazione

ATM (Asynchronous Transfer Mode)

Caratterizzato da commutazione di pacchetto. Circuiti virtuali semi permanenti. Non c'è bisogno di un frame di livello 2.

Vantaggi

- **connessioni virtuali**
- **Più connessioni per interfaccia**
- **massima flessibilità di configurazione**
- **supporto per altri dispositivi (frame relay)**
- **multiplexing flessibile**

Svantaggi

- **sovraffollamento**

Frame Relay

Commutazione di pacchetto. Connessioni virtuali semi permanenti. Non c'è bisogno di un frame di livello 2.

Vantaggi

- **Connessioni virtuali**
- **Più connessioni per interfaccia**

Svantaggi

- **Qualità del servizio non garantita**

MPLS

Necessitò di introdurre un nuovo livello per lo scambio di pacchetti in modo da far fronte alla mancanza di capacità di IP di fare traffic engineering (congestione di pacchetti per stessa destinazione)

Commutazione di pacchetto, connessioni virtuali dinamiche.

Vantaggi

- **Connessioni virtuali dinamiche (LSP con label switching)**
- **Più connessioni per interfaccia**
- **Un solo piano di controllo (no IP over)**
- **Qualità del servizio non garantita**