Politecnico di Torino
Laurea Magistrale in Ingegneria Informatica

appunti di
Tecnologie e servizi di rete

Autori principali: Lorenzo David, Luca Ghio, Riccardo Zaccone
Docenti: Mario Baldi, Guido Marchetto
Anno accademico: 2013/2014
Versione: 1.1.0.0
Data: 8 maggio 2021

Ringraziamenti

Speciali ringraziamenti vanno a Elia Neishaboori per il suo aiuto nella sezione su H.323, e a Ebrahim Kargarnasrabadi per il suo aiuto nella sezione su SIP.

Speciali ringraziamenti vanno a Giacomo Ratta perché ha concesso di integrare il suo lavoro nel capitolo sulla migrazione a IPv6.

Oltre agli autori precedentemente citati, quest'opera può includere contributi da opere correlate su WikiAppunti e su Wikibooks, perciò grazie anche a tutti gli utenti che hanno apportato contributi agli appunti *Tecnologie e servizi di rete* e al libro *Tecnologie e servizi di rete*.

Informazioni su quest'opera

Quest'opera è pubblicata gratuitamente. Puoi scaricare l'ultima versione del documento PDF, insieme al codice sorgente LaTeX, da qui: http://lucaghio.epizy.com/redirs/2

Quest'opera non è stata controllata in alcun modo dai professori e quindi potrebbe contenere degli errori. Se ne trovi uno, sei invitato a correggerlo direttamente tu stesso realizzando un commit nel repository Git pubblico o modificando gli appunti *Tecnologie e servizi di rete* su WikiAppunti, oppure alternativamente puoi contattare gli autori principali inviando un messaggio di posta elettronica a luca.ghio@studenti.polito.it o a lorenzodavid91@gmail.com.

Licenza

Indice

Capitolo 1

WAN

In senso stretto, una **rete geografica** (WAN, acronimo inglese di "Wide Area Network") è una rete che si estende su un'area ampia, coprendo regioni, Paesi o nel caso di Internet anche tutto il mondo. Più in generale, qualsiasi tecnologia di reti di computer usata per trasmettere dati per lunghe distanze può essere chiamata WAN.

Una tecnologia WAN deve rispettare alcuni requisiti in termini di durata del servizio, bit rate e vincoli di ritardo a seconda dell'applicazione (telemetria, telefonia, trasferimento dati, ecc.) per cui è stata progettata.

ATM rappresenta la convergenza per una grande varietà di tecnologie che in passato i mondi delle telecomunicazioni e dell'informatica hanno introdotto parallelamente per la trasmissione di dati per lunghe distanze:

- nel mondo delle telecomunicazioni, la telefonia è passata da analogica a digitale, poi ISDN e B-ISDN hanno iniziato a trasportare dati insieme alla voce;
- nel mondo dell'informatica, Frame Relay ha soppiantato le linee dedicate analogiche e digitali traendo vantaggio dalla commutazione di circuito, e X.25 spostando la complessità dai nodi centrali a quelli ai margini.

Oggigiorno ATM sta per essere abbandonato in favore di IP grazie alla sua minore complessità e alla sua maggiore semplicità.

1.1 ISDN

Integrated Service Digital Network (ISDN) consente di trasportare dati insieme alla voce: diversi dispositivi digitali possono essere connessi a un bus e possono trasmettere sui canali ISDN disponibili:

- **Basic Rate Access** (BRA) o **Basic Rate Interface** (BRI): offre 2 canali B dati a 64 kbps e 1 canale D di segnalazione a 16 kbps ⇒ totale: 144 kbps (adatto per singoli utenti e piccoli uffici);
- **Primary Rate Access** (PRA) o **Primary Rate Interface** (PRI): offre 30 canali B dati a 64 kbps e un canale D di segnalazione a 16 kbps ⇒ totale: 2 Mbps (adatto per le aziende).

La trasmissione è basata su Time Division Multiplexing (TDM); tutti i canali vanno a un Network Termination ed entrano nella rete su un filo digitale chiamato "local loop". I canali ereditano la logica dagli operatori delle telecomunicazioni: essi rimangono attivi anche quando non vengono scambiati dati.

1.2 PDH

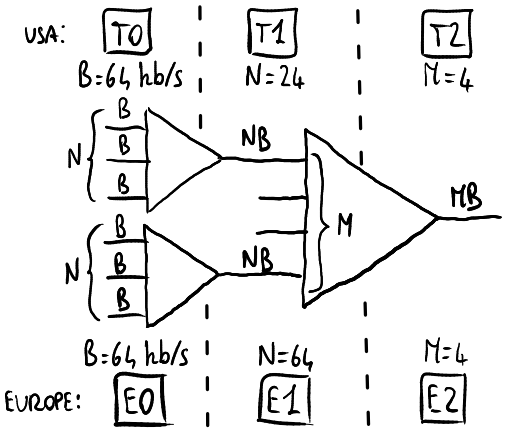


Figura 1.1: Gerarchia PDH.

Plesiochronous Digital Hierarchy (PDH) è un vecchio standard progettato per trasferire canali vocali digitali a 64 Kb/s (PCM) su reti telefoniche digitali basate su TDM. Il sistema è chiamato "plesiocrono" perché è necessaria una stretta sincronizzazione tra trasmettitore e ricevitore, anche se ogni apparato ha il proprio clock.

I flussi di dati sono organizzati in modo gerarchico: i canali vengono aggregati in flussi dal livello più basso a quello più alto (**grooming**), e più è alto il livello gerarchico maggiore è il bit rate. Per esempio, al livello T1 vengono messi 24 canali di livello T0 uno accanto all'altro in un'unica trama: poiché la trama deve durare 125 μs per tutti i livelli, al livello T1 il bit rate sarà 24 volte più alto di quello al livello T0.[1]

1.3 SDH

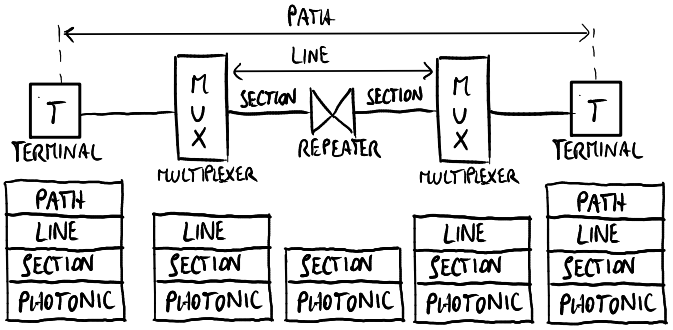


Figura 1.2: Architetture fisica e protocollare di SDH.

Synchronous Digital Hierarchy (SDH), l'equivalente europeo dello standard internazionale SONET, differisce da PDH per le sue velocità più elevate:

- esiste un unico clock per l'intero sistema ⇒ è richiesta una rete di sincronizzazione per una sincronizzazione più stretta;
- è necessario rimpiazzare i cavi di rame con le fibre ottiche;

[1] Non sono considerati i bit di segnalazione.

- il multiplexing dei flussi è più complesso di PDH, perché è progettato per ottimizzare l'elaborazione hardware.

L'architettura protocollare è organizzata come una pila di livelli, e ogni nodo nella architettura fisica della rete li implementa a seconda della sua funzionalità:

- **livello di percorso**: comunicazione end-to-end tra due terminali;
- **livello di linea**: un percorso è suddiviso in linee dai multiplexer;
- **livello di sezione**: una linea è suddivisa in sezioni dai ripetitori (per lunghe distanze);
- **livello fotonico**: il livello più basso per le fibre ottiche.

Ogni trama temporale dura 125 μs e la sua intestazione include informazioni di sincronizzazione usate per combinare e separare canali, e informazioni di OAM (Operation, Administration and Management) usate per rilevare i guasti e recuperare da essi.

SDH e PDH rappresentano il livello di trasporto su cui operano ATM e Frame Relay.

1.4 Frame Relay

Frame Relay è uno standard di livello 2 orientato alla connessione per instaurare dei circuiti permanenti virtuali su reti a commutazione di pacchetto. Ogni circuito permanente è identificato da un **Data Link Connection Identifier** (DLCI).

Lo standard è molto flessibile: infatti non specifica la tecnologia di livello superiore (ATM, X.25...) usata internamente nella rete.

1.4.1 CIR

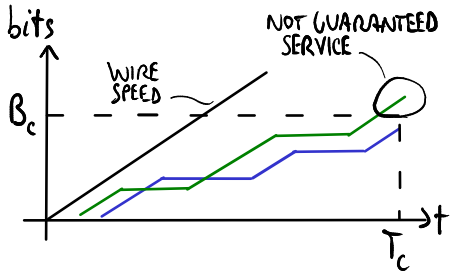


Figura 1.3: Il servizio è garantito per l'utente in blu ma non per quello in verde perché la sua burstiness è troppo alta.

Il massimo bit rate supportato non è sufficiente per descrivere le prestazioni di una rete Frame Relay, perché un utente potrebbe inviare bit consecutivamente al massimo bit rate (velocità del filo) per un lungo periodo di tempo causando una congestione nella rete. Pertanto il fornitore della rete fornisce anche il **Committed Information Rate** (CIR), che è il massimo numero B_C di bit che l'utente può trasmettere entro un certo intervallo di tempo T_C affinché il servizio sia garantito:

$$\text{CIR} = B_C \cdot T_C$$

dove B_C è detto **committed burst size**:

- burstiness bassa: l'utente invia pacchetti raramente $\Rightarrow$ il servizio è sempre garantito;
- burstiness alta: l'utente continua ad inviare pacchetti consecutivamente alla velocità del filo $\Rightarrow$ quando supera il committed burst size il servizio non sarà più garantito.

Il Data Terminal Equipment (DTE) dell'utente può interrompere la trasmissione quando viene raggiunta la massima burstiness.

1.5 ATM

Asynchronous Transfer Mode (ATM) è uno standard orientato alla connessione per instaurare dei circuiti virtuali su reti B-ISDN. Ogni circuito è identificato da un **Virtual Path Identifier** (VPI) e un **Virtual Circuit Identifier** (VCI), e può essere permanente o instaurato dinamicamente tramite messaggi di segnalazione.

Le **celle** ATM sono molto piccole: ogni cella ATM è lunga 53 byte, costituita da un'intestazione lunga 5 byte, contenente gli identificatori della connessione, e da un payload lungo 48 byte ⇒ bassa latenza e bassi ritardi di pacchettizzazione.

Le reti ATM hanno un modello molto complesso, derivato da una mentalità da operatore telefonico per avere il pieno controllo della rete e garantire un'elevata tolleranza ai guasti.

1.5.1 AAL 5

Quando ATM fu progettato, era pensato per essere implementato in modo ubiquo nella rete, anche ai suoi margini nelle schede di rete dei PC dell'utente. Oggigiorno i PC ai margini implementano solo il protocollo IP perché la sua implementazione è meno costosa, e si può trovare ATM solo come livello di trasporto nel cuore della rete nascosto dall'utente.

ATM Adaptation Layer (AAL) di tipo 5 è usato per Segmentation and Reassembly (SAR):

- Segmentation: i pacchetti IP sono suddivisi in celle ATM;
- Reassembly: le celle ATM sono combinate in pacchetti IP.

AAL complica l'interazione tra IP e ATM, perché gli indirizzi IP devono essere tradotti in identificatori di connessione ATM e viceversa ⇒ oggi la tendenza è abbandonare il piano di controllo ATM per adottare il piano di controllo MPLS.

1.6 Reti ottiche

Nelle **reti ottiche** i dati sono trasmessi su onde elettromagnetiche multiplate usando WDM, trasportate tramite fibre ottiche e commutate da sistemi di commutazione ottici basati su specchi.

Il **Wavelength Division Multiplexing** (WDM) consente di mettere più segnali ottici in una sola fibra ottica ⇒ aumenta la capacità di trasmissione delle fibre:

- **Coarse WDM** (CWDM): permette di trasmettere un minor numero di segnali di lunghezze d'onda ben separate l'una dall'altra ⇒ meno costoso perché la demultiplazione è più facile;
- **Dense WDM** (DWDM): permette di trasmettere un maggior numero di segnali di qualsiasi lunghezza d'onda ⇒ più costoso perché la demultiplazione è più complessa.

La **commutazione ottica** è basata su specchi controllati da sistemi micro-elettro-meccanici (MEMS), che riflettono i segnali elettromagnetici da una fibra in ingresso a una fibra in uscita. La commutazione ottica è molto flessibile: sfrutta le proprietà fisiche delle onde elettromagnetiche senza curarsi dei bit ⇒ le reti possono essere aggiornate a velocità più elevate perché i commutatori ottici continuano a funzionare indipendentemente dal bit rate.

Esistono diversi tipi di commutatori ottici:

- **add/drop multiplexer**: è il commutatore ottico più semplice: può essere interposto tra due fibre per inserire (add) segnali provenienti dai trasmettitori nella rete, ed estrarre (drop) segnali dalla rete verso i ricevitori in modo ottico;
- **cross-connect**: può connettere più fibre in ingresso a più fibre in uscita:
  - **fiber cross-connect**: vengono commutate a una fibra in uscita tutte le onde elettromagnetiche provenienti da una fibra in ingresso;

  - **waveband cross-connect**: viene commutato a una fibra in uscita un insieme di onde elettromagnetiche di lunghezze d'onda vicine proveniente da una fibra in ingresso;
  - **wavelength cross-connect**: viene commutato a una fibra in uscita un insieme di onde elettromagnetiche di uguale lunghezza d'onda proveniente da una fibra in ingresso;

- **wavelength switch**: la configurazione è dinamica, cioè i commutatori possono cambiare i circuiti più velocemente dei cross-connect ⇒ il recupero dai guasti è veloce.

Due segnali di uguale lunghezza d'onda possono provenire da due fibre in ingresso diverse ma possono dover essere commutati sulla stessa fibra in uscita ⇒ tramite la **wavelength conversion** un commutatore ottico può cambiare la lunghezza d'onda di un segnale in una non ancora usata nella fibra in uscita, per mantenere separati tutti i segnali.

I commutatori ottici possono essere usati nel backbone della rete per interconnettere i principali punti di accesso, instaurando dei **percorsi ottici** tramite fibra ottica tra le città nel mondo. I commutatori ottici possono instaurare percorsi ottici usando protocolli di segnalazione e di instradamento quali LDP e RSVP. I commutatori ottici sono tolleranti ai guasti: quando un canale si guasta, possono riflettere le onde lungo un altro percorso ottico.

Il WDM può essere distribuito come livello di trasporto su cui può operare qualsiasi protocollo di livello 2 (SONET, Ethernet...) che delimita le trame.

Tuttavia la tecnologia per la commutazione ottica pura è ancora allo stato embrionale: oggi i commutatori WDM sono più costosi di quelli a commutazione di pacchetto, e possono avere poche interfacce perché il sistema a specchi sarebbe molto complesso per molte interfacce. Inoltre la commutazione ottica è orientata alla connessione: quando viene instaurato un circuito, le risorse continuano ad essere allocate anche se il circuito non è attualmente utilizzato ⇒ la commutazione ottica è adatta per il backbone della rete dove il traffico è piuttosto continuo.

Soluzioni più economiche provano a superare i limiti tecnologici rimpiazzando gli specchi con una matrice di commutazione elettrica: ogni segnale ottico è convertito in una sequenza di bit tramite una **conversione optical-to-electrical** (OE) in modo che possa essere commutata più facilmente, quindi è riconvertita in un segnale ottico. Il segnale riconvertito viene rigenerato, potendo viaggiare per una distanza più lunga prima di perdere potenza, ma questa soluzione ha molti svantaggi: i commutatori consumano molta energia rispetto ai commutatori all-optical, e un cambiamento del bit rate richiede l'aggiornamento dei commutatori.

Capitolo 2

MPLS

Multiprotocol Label Switching (MPLS) è la tecnologia di supporto per la nuova rete pubblica (IP) a banda larga. Si può considerare un'architettura di protocolli (o una suite di protocolli) per controllare diversi sotto-protocolli.

MPLS opera ad un livello che generalmente si considera che stia tra le definizioni tradizionali di livello 2 (livello di collegamento dati) e livello 3 (livello di rete).

2.1 Vantaggi

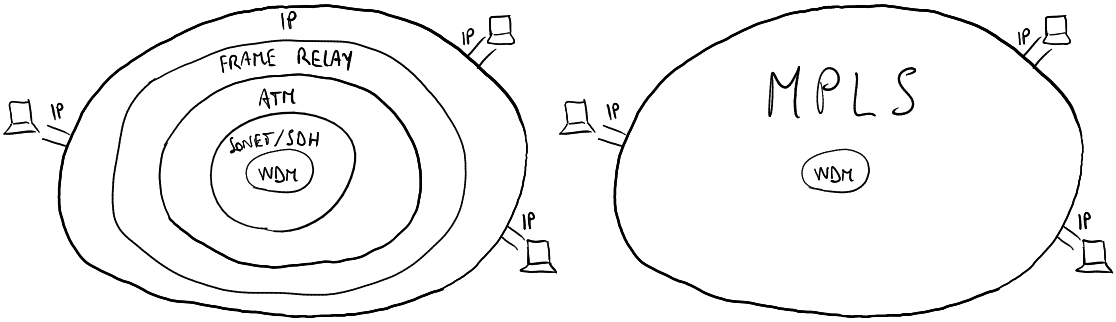


Figura 2.1: L'introduzione di MPLS semplifica la tradizionale "grande cipolla".

Il protocollo IP è stato sviluppato per scopi di ricerca e non è stato progettato per essere venduto come un servizio. È un cosiddetto "protocollo best-effort", cioè non c'è alcuno scopo esplicito nel dare un servizio affidabile garantito (velocità, ritardi...).

Quando IP cominciò a diventare un bene commerciale, l'International Telecommunication Union (ITU) iniziò a sviluppare protocolli (come ATM, frame relay, ecc.) rivolti all'affidabilità e alla stabilità del servizio, pensando che essi avrebbero permeato il mondo delle telecomunicazioni dei computer. Ciononostante gli utenti finali hanno continuato ad utilizzare IP, e di conseguenza i service provider oggigiorno devono avere a che fare con molti protocolli per portare IP agli utenti finali: questa "**grande cipolla**" non ha alcun senso per i service provider a causa degli alti costi per la manutenzione, le apparecchiature e lo sviluppo software per garantire l'interoperabilità.

Cisco Systems fu il primo fornitore ad implementare il tag switching nei router, poi IETF adottò il protocollo e lo denominò MPLS.

MPLS combina le migliori caratteristiche dei protocolli connection-less con le migliori dei protocolli orientati alla connessione, rappresentando la soluzione per il problema della "grande cipolla" per due motivi:

- MPLS fornisce una rete basata su IP con una maggiore affidabilità del servizio e un singolo piano di controllo unificato più isolato dal piano dati:

  - in IP i piani di controllo e dati sono continuamente aggiornati ad ogni cambiamento nella rete;
  - in MPLS l'aggiornamento si verifica solo quando viene creato un nuovo LSP; siccome c'è una separazione tra piano dati e piano di controllo è possibile creare percorsi con vincoli indipendenti;
- MPLS permette di riutilizzare gli apparati ATM tradizionali semplicemente aggiornandone il software.

Caratteristiche principali

- possibilità di traffic engineering: distribuire il carico di traffico nella rete per evitare congestioni;
- indipendenza dai protocolli (multi-protocollo) ⇒ utile nella transizione da IPv4 a IPv6;
- progettato per garantire la qualità del servizio (non ancora supportato);
- piano di controllo unificato: può essere usato per qualsiasi rete oltre a IP (per es. MPλS per le reti ottiche);
- rapido recupero dai guasti: si possono creare due percorsi tra una coppia di nodi, in modo che in caso di guasto nel primo percorso l'LSR può solo notificare il guasto e rapidamente deviare il traffico sul secondo percorso[1] (invece in IP è difficile inserire due percorsi in una tabella di instradamento, e se un canale si guasta i router hanno bisogno di scambiare informazioni di instradamento ed effettuare algoritmi sofisticati per trovare un altro percorso).

2.2 Architettura di rete

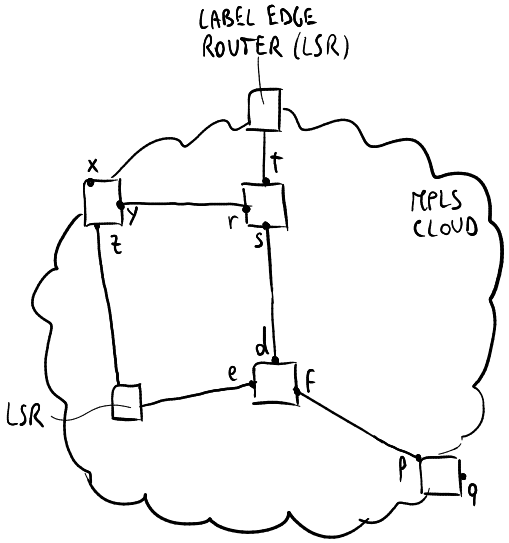


Figura 2.2: Esempio di rete MPLS.

Un **Label Switch Router** (LSR) è l'apparato responsabile per la commutazione delle etichette usate per instradare i pacchetti. Gli LSR sono detti **label edge router** quando posti ai margini della nuvola MPLS. Gli LSR combinano l'intelligenza dei router e la velocità degli switch:

[1] È necessario un overhead per mantenere disponibili due LSP per la stessa FEC.

sono capaci di instradare in modo intelligente come i router, evitando strutture dati e algoritmi complicati come gli switch.

Le nuvole MPLS possono essere distribuite gradualmente: possono crescere e possono essere integrate l'una con l'altra.

2.3 Piano dati

Il **piano dati** è la capacità di commutare pacchetti in base alle loro etichette.

2.3.1 Intestazione MPLS

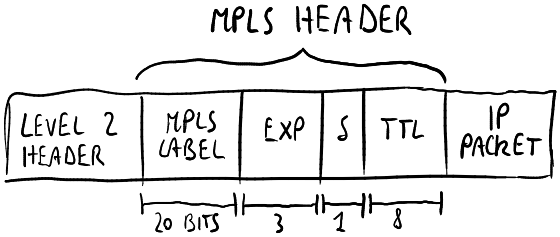


Figura 2.3: Formato di un pacchetto contenente una singola entry nella pila delle etichette.

I pacchetti IP sono prefissati con un'intestazione MPLS contenente una o più entry nella pila delle etichette. Ogni entry nella pila delle etichette contiene quattro campi:

- etichetta: l'instradamento è basato su questo campo invece che sull'indirizzo IP di destinazione;
- classe di traffico (exp): per la priorità della qualità del servizio (QoS) e l'Explicit Congestion Notification (ECN);
- flag bottom of stack (S): se impostato, l'etichetta corrente è l'ultima nella pila;
- Time to Live (TTL).

2.3.2 Commutazione ad etichetta

Un **Label Switched Path** (LSP) è un percorso creato con un protocollo di segnalazione che collega un label edge router sorgente (ingress) ad uno di drain (egress):

- quando l'LSR ingress riceve un pacchetto, vi aggiunge un'etichetta e lo inoltra all'hop successivo dell'LSP creato in precedenza;
- quando l'LSR egress riceve un pacchetto, ne toglie l'etichetta e lo inoltra fuori dalla nuvola MPLS.

Una **Forwarding Equivalence Class** (FEC) è un insieme di pacchetti che possono essere inoltrati nello stesso modo, cioè possono essere associati alle stesse etichette MPLS. Le etichette non sono univoche nell'intera nuvola MPLS, ma cambiano ad ogni hop (**label swapping**). Si tenga conto che garantire l'univocità delle etichette in tutta la rete richiederebbe protocolli troppo complessi ed etichette troppo lunghe.

L'utilizzo delle etichette consente ad MPLS di fornire due tipi di servizi:

- ricerca rapida: il routing di IP, basato sull'algoritmo "longest prefix matching", è sofisticato, difficile da ottimizzare e non abbastanza veloce quando ha a che fare con un'ampia quantità di rotte.
  MPLS fornisce una ricerca più veloce rispetto ad IP perché le decisioni per l'inoltro dei

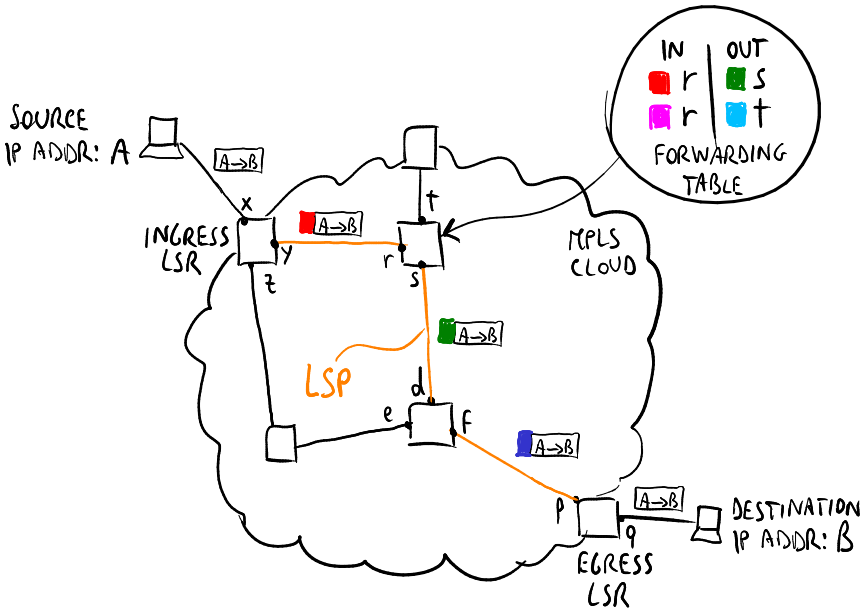


Figura 2.4: Esempio di commutazione ad etichetta MPLS.

pacchetti sono fatte solamente sull'etichetta, posta prima del pacchetto IP, senza la necessità di esaminare il contenuto del pacchetto stesso: ogni etichetta infatti può essere usata come chiave per accedere alla tabella di instradamento come un vettore o una tabella di hash per velocizzare la scoperta delle rotte;

- traffic engineering: IP tende ad aggregare il traffico, ma avere molti pacchetti che attraversano lo stesso percorso non fornisce un servizio efficiente. Ciò non si può evitare facilmente poiché richiederebbe una configurazione di rotte statiche ⇒ costosa e non scalabile.
  MPLS è in grado di controllare il traffico come un protocollo orientato alla connessione: l'instradamento di MPLS coinvolge sia l'etichetta *sorgente* sia quella di *destinazione*, e i router possono assegnare a un nuovo flusso di pacchetti l'etichetta corrispondente al percorso meno carico per evitare la congestione e consentire la distribuzione del traffico. Inoltre un guasto in un percorso dovuto a un nodo non funzionante non influenzerà gli altri percorsi.

Gerarchia e scalabilità

MPLS è molto scalabile: all'interno di una grande nuvola MPLS di dominio 1 è possibile definire in modo gerarchico una nuvola MPLS più piccola di dominio 2 e così via, e si possono memorizzare più entry nella pila delle etichette una accanto all'altra in una struttura dati a pila. Le entry nella pila delle etichette siano aggiunte da quella più interna a quella più esterna mentre il pacchetto entra nelle nuvole di dominio più alto e siano rimosse da quella più esterna a quella più interna mentre il pacchetto esce dalle nuvole di dominio più basso, e gli LSR non ai margini delle nuvole elaborano sempre l'entry nella pila delle etichette più esterna. Questa gerarchia di etichette può corrispondere ad una gerarchia di provider, e il numero delle etichette è limitato solo dalla dimensione della trama Ethernet.

Questa tecnica introduce alcuni vantaggi:

- riduce le dimensioni delle tabelle di instradamento e di inoltro, perché non devono essere globali;
- permette di riutilizzare l'hardware per la commutazione esistente (ATM, frame relay, ecc.): le intestazioni MPLS sono messe direttamente nelle intestazioni di livello 2, in modo che possano essere elaborate dall'hardware esistente che ora elabora il livello 2 semplicemente aggiornandone il software.

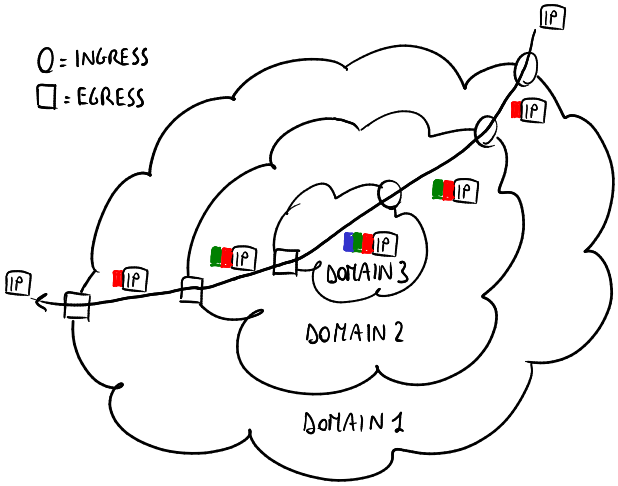


Figura 2.5: Gerarchia di etichette lungo un LSP.

2.4 Piano di controllo

Il **piano di controllo** è la capacità di scegliere le etichette da inserire nei pacchetti.

La creazione di una tabella di inoltro (e in senso lato dell'LSP) per una specifica FEC è effettuata in tre passi:

1. **label binding**: è sempre effettuata dal nodo di downstream, che sceglie un'etichetta per la FEC, e questo può essere effettuato in due modi (non mutualmente esclusivi):
   - non sollecitato: il nodo di downstream può decidere in qualsiasi momento di assegnare le etichette, anche se non c'è traffico nella rete;
   - su richiesta: il nodo di upstream può chiedere al nodo di downstream un'etichetta fissa;
2. **label distribution**: il nodo di downstream comunica l'etichetta scelta al nodo di upstream;
3. **label mapping**: il nodo di upstream crea una nuova entry nella tabella di inoltro associando i pacchetti in entrata, provenienti da una specifica porta con una specifica etichetta, ai pacchetti in uscita, uscenti da una specifica porta con una specifica etichetta.

Le etichette possono essere assegnate in due modi:

- **staticamente**: il gestore della rete crea gli LSP manualmente, come i circuiti virtuali permanenti (PVC) nelle tecnologie orientate alla connessione come ATM ⇒ questa soluzione non scala e limita l'interoperabilità tra i diversi service provider;
- **dinamicamente**: label binding, distribution e mapping sono effettuati automaticamente dagli LSR senza intervento manuale:
  - guidati dai dati: la creazione di un LSP è scatenata dalla ricezione di pacchetti di dati, e ogni LSR sceglie autonomamente le etichette in base al traffico;
  - guidati dal controllo: ad un certo punto l'LSR assegna un'etichetta, anche se non c'è traffico;
  - guidati dalla topologia (o guidati dal protocollo): ogni volta che si scopre una nuova destinazione, viene creato un LSP verso questa destinazione ⇒ no traffic engineering: la rete funziona esattamente come una rete IP;
  - espliciti: la creazione degli LSP, di solito inizializzata dai label edge router o guidata dai dati o tramite configurazione manuale, è effettuata attraverso una segnalazione esplicita.

2.5 Protocolli

2.5.1 Protocolli di label distribution

Tre protocolli, incompatibili tra di loro, possono essere usati dal nodo di downstream per comunicare al nodo di upstream le associazioni delle etichette:

- **Label Distribution Protocol** (LDP): progettato specificatamente per il label distribution;
- **Border Gateway Protocol** (BGP) esteso: il nodo di downstream include nei messaggi di instradamento BGP, usati per annunciare le nuove destinazioni, un nuovo campo che dice al nodo di upstream le etichette scelte (solo per il label binding guidato dal protocollo);
- **Resource Reservation Protocol** (RSVP) esteso: il nodo di downstream include nei messaggi RSVP, usati per notificare i tipi di traffico dei flussi di pacchetti per la qualità del servizio, un nuovo campo che dice al nodo di upstream le etichette scelte (vedere la sezione 7.3 per i dettagli).

2.5.2 Protocolli di instradamento

I protocolli di instradamento tradizionali possono essere migliorati per supportare il **traffic engineering** perché trasportano informazioni sui vincoli di instradamento.

Grazie a protocolli di instradamento come OSPF-TE e IS-IS-TE (basati su OSPF, IS-IS, BGP-4), ogni nodo può raccogliere informazioni sulla topologia della rete per sapere quali nodi sono i suoi nodi di upstream da notificare con le associazioni delle etichette.

Ci sono due possibili strategie di instradamento:

- **hop-by-hop** (così com'è nell'instradamento di IP): protocollo di instradamento distribuito dove ogni LSR decide da solo secondo il criterio del percorso più breve, così può succedere che tutti i router scelgano lo stesso percorso ⇒ rischio di congestione;
- **esplicito** (possibilità di instradamento basato sui vincoli): protocollo di instradamento centralizzato dove gli LSR egress vengono informati per capire quali canali sono attualmente i più carichi e scegliere i canali meno carichi per la creazione dei nuovi LSP in modo che siano disgiunti il più possibile dagli altri percorsi.
  Per supportare l'instradamento esplicito, è necessario estendere i protocolli di base per il label distribution:
  - **Constraint-based Routing LDP** (CR-LDP) è un'estensione a LDP;
  - **RSVP for Traffic Engineering** (RSVP-TE) è un'estensione a RSVP.

Capitolo 3

IPv6

Internet Protocol version 6 (IPv6) è un nuovo protocollo con lo scopo di superare i limiti di IPv4: il principale motivo dell'introduzione di un nuovo protocollo è avere uno **spazio degli indirizzi più grande** rispetto a quello di IPv4.

3.1 Confronto con IPv4

IPv6 espande il protocollo ICMP integrando i seguenti protocolli:

- ARP: chiamato "neighbor discovery" per il processo di configurazione degli indirizzi;
- IGMP: chiamato "Multicast Listener Discovery" per gestire le appartenenze ai gruppi multicast.

Con IPv6 alcuni protocolli devono solo essere aggiornati, principalmente per il fatto che tutti hanno a che fare con indirizzi (questi protocolli non sono indipendenti dal livello 3):

- protocolli DNS;
- protocolli di instradamento: RIP, OSPF, BGP, IDRP;
- protocolli di trasporto: TCP, UDP;
- interfacce socket.

3.1.1 Funzionalità aggiuntive di IPv6

Le funzionalità aggiuntive elencate qui sotto furono originariamente progettate come componenti aggiuntivi per IPv4, poi è stato effettuato il porting per incorporarle in IPv6.

Distribuzione su LAN È più efficiente, grazie a un utilizzo efficiente degli indirizzi multicast e anycast:

- **multicast**: ogni indirizzo multicast identifica un gruppo di stazioni, e il pacchetto viene inoltrato a tutti i nodi nel gruppo;
- **anycast**: ogni indirizzo anycast identifica un gruppo di stazioni, ma il pacchetto viene inoltrato solo al nodo più vicino nel gruppo.

Sicurezza e privacy dei dati Nel protocollo IPv6 sono inclusi meccanismi di sicurezza come IPsec (sezione 3.4.4).

Policy routing È la possibilità di inoltrare i pacchetti utilizzando politiche diverse dall'indirizzo di destinazione (per es. inoltro in base all'indirizzo sorgente).

Plug and play Sono definiti dei protocolli per l'autoconfigurazione:

- **stateless**: solo l'accesso link-local è garantito senza contattare alcun server;
- **stateful**: è possibile avere accesso a Internet usando un server DHCP.

Differenziazione del traffico Non tutti i flussi di dati sono uguali (per es. le chiamate telefoniche richiedono meno ritardi).

Mobilità È la capacità di spostare il dispositivo per reti diverse mantenendo disponibili tutti i servizi (per es. dispositivi mobili che usano GSM/LTE spostandosi intorno a celle diverse).

Nomadicità È la capacità di spostare il dispositivo per reti diverse senza la necessità di garantire i servizi attivi ⇒ meno stretta della mobilità.

Scalabilità migliore con l'instradamento Di norma è necessaria l'aggregazione per semplificare l'instradamento ma richiede uno spreco di indirizzi. L'instradamento IPv6 utilizza quasi le stesse tecniche di IPv4 ma può ridurre le tabelle di instradamento, se gli indirizzi sono dati in un modo efficiente.

3.2 Indirizzamento

3.2.1 Formato degli indirizzi

Ogni indirizzo IPv6 è lungo 128 bit, e il prefisso sostituisce la netmask:

3.2.2 Link

Il concetto di **link** in IPv6 è uguale al concetto di sottorete in IPv4:

- in IPv4 una sottorete è un insieme di host con lo stesso prefisso;
- in IPv6 un link è l'effettiva rete fisica.

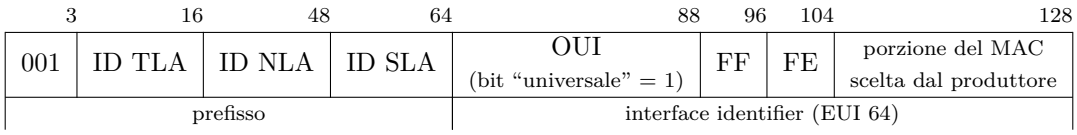
Tutti gli host nella stessa sottorete appartengono allo stesso link e viceversa:

- gli **host on-link** hanno lo stesso prefisso, così possono comunicare direttamente;
- gli **host off-link** hanno prefissi diversi, così possono comunicare attraverso un router.

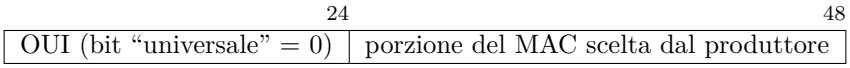
3.2.3 Organizzazione dello spazio di indirizzamento

Indirizzi global unicast

Indirizzi global unicast aggregatable Sono equivalenti agli indirizzi pubblici IPv4, e iniziano con i 3 bit "001":



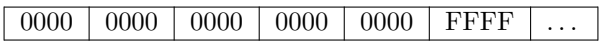
- Prefisso: deve essere uguale a quello assegnato al link a cui è connesso l'host.
  Il criterio per l'assegnazione dei prefissi è basato sulla topologia: essi sono assegnati in base alla gerarchia dei service provider:
  - Top Level Authority (TLA): un service provider grande;
  - Next Level Authority (NLA): un service provider intermedio;
  - Subnet Level Authority (SLA): una azienda.
- Interface identifier: identifica l'interfaccia dell'host.
  Opzionalmente può essere in formato EUI-64: l'interface identifier IPv6 a 64 bit deriva dall'indirizzo MAC a 48 bit dell'host:



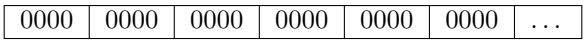
  dove il bit "universale" è il settimo bit nell'OUI e viene cambiato da 0 a 1.

Indirizzi per l'interoperabilità con IPv4 Sono da usare durante la fase di transizione, e iniziano con 80 bit impostati a zero:

- **indirizzi mappati IPv4**: i primi 80 bit sono zeri e i successivi 16 bit sono impostati a uno:

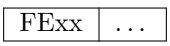


- **indirizzi compatibili con IPv4**: i primi 80 bit sono zeri e i successivi 16 bit sono impostati a zero (per es. l'indirizzo IPv6 "::10.0.0.1" mappa l'indirizzo IPv4 "10.0.0.1"):

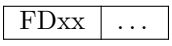


Indirizzi local unicast

Indirizzi link local Si riferiscono agli indirizzi privati "automatici", generati dall'autoconfigurazione, che è il processo in cui una stazione genera automaticamente un indirizzo per connettersi a un link IPv6 (vedere la sezione 3.5.3 per i dettagli):

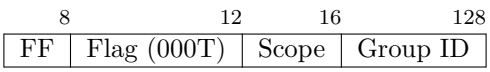


Indirizzi site local Sono equivalenti agli indirizzi privati IPv4:



Indirizzi multicast

Un **indirizzo multicast** identifica un gruppo di stazioni, e ha il formato seguente:



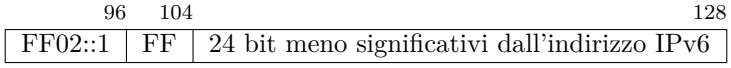
dove i campi sono:

- campo Flag (4 bit): è usato per contrassegnare un gruppo multicast:

  - T = 1: il gruppo multicast è temporaneo (per es. chiamata di conferenza definita dall'utente);
  - T = 0: il gruppo multicast è permanente (per es. indirizzo di tutti gli host della rete, non può essere sovrascritto);
- campo Scope (4 bit): è usato per limitare la diffusione del multicast (meglio del TTL di IPv4):
  1 - node local: il pacchetto non può andare fuori dall'host;
  2 - link local: il pacchetto non può andare fuori dalla rete di livello 2;
  5 - site local: il pacchetto non può andare fuori per es. dalla rete dell'università;
  8 - organization local: il pacchetto non può andare fuori dalla rete dell'organizzazione;
  E - global: il pacchetto può andare ovunque;
- campo Group ID (112 bit): identifica il gruppo multicast, e il pacchetto viene inoltrato a tutti i nodi nel gruppo.

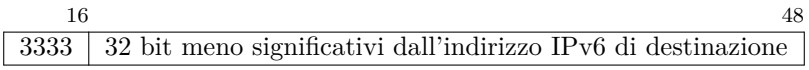
Se un host vuole appartenere a un gruppo multicast, deve chiederlo usando il protocollo ICMP (vedere la sezione 3.5.2); una volta che viene aggiunto al gruppo multicast, riceverà tutti i pacchetti inviati a quel particolare indirizzo multicast. È molto importante notare che gli host che riceveranno un pacchetto multicast non sono definiti dalla sorgente, ma sono "decisi" dalle destinazioni.

Indirizzi multicast solicited node Ogni nodo operativo appartiene per impostazione predefinita a un gruppo multicast solicited node il cui indirizzo deriva dal suo indirizzo IPv6:



Ci può essere più di un host nello stesso gruppo multicast, ma generalmente no perché l'indirizzo multicast è generato dall'indirizzo IPv6.

Mappare IPv6 su Ethernet Ogni pacchetto multicast viene consegnato tramite una trama Ethernet con uno specifico indirizzo MAC derivato da un indirizzo multicast IPv6, in modo che il pacchetto venga elaborato solamente dagli host interessati:



3.2.4 Argomenti avanzati legati agli indirizzi IPv6

Renumbering

Poiché i prefissi degli indirizzi global sono assegnati in base alla gerarchia dei service provider, se una società vuole passare da un service provider a un altro, tutti i link nella rete aziendale dovranno cambiare prefisso. IPv6 è pensato per supportare un renumbering facile sia per gli host sia per i router:

- host: i router gradualmente smettono di annunciare il vecchio prefisso (deprecato) e iniziano ad annunciare quello nuovo (preferito) ⇒ ogni host avrà durante la fase di migrazione due indirizzi con prefissi diversi per la stessa interfaccia;
- router: Router Renumbering è uno standard che consente al router di frontiera di notificare il nuovo prefisso agli altri router interni.

Tuttavia il renumbering ha ancora alcuni problemi irrisolti, relativi a come aggiornare automaticamente ad es. le entry dei DNS, i filtri dei firewall, le politiche aziendali basate sugli indirizzi, ecc.

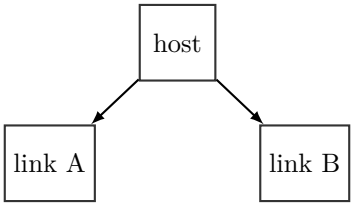
Multi-homing

Una grande società può decidere di acquistare la connettività a Internet da due diversi service provider perché vuole continuare ad essere connessa a Internet anche se uno dei service provider ha qualche problema.

Poiché i prefissi degli indirizzi globali sono assegnati in base alla gerarchia dei service provider, ogni host all'interno della rete aziendale avrà due indirizzi globali con prefissi diversi per la stessa interfaccia ⇒ l'host dovrà selezionare l'indirizzo da utilizzare per ogni pacchetto in uscita. Questo può provocare alcuni problemi di configurazione non banali:

- instradamento basato sull'indirizzo di destinazione: l'host dovrebbe essere capace di selezionare il prefisso giusto per i pacchetti in uscita, altrimenti supponiamo che l'host selezioni il prefisso del provider A ma la destinazione si trovi nella rete del provider B ⇒ il router di frontiera grazie ai meccanismi di instradamento inoltrerà il pacchetto direttamente nella rete del provider B ⇒ il provider B bloccherà quel pacchetto perché l'indirizzo sorgente ha un prefisso diverso;
- registrazione doppia nei DNS: l'host dovrebbe essere registrato nei DNS con due indirizzi diversi per lo stesso alias;
- renumbering automatico: i meccanismi per il renumbering dovrebbero supportare dinamicamente il passaggio da un provider B a un provider C.

Indirizzi scoped

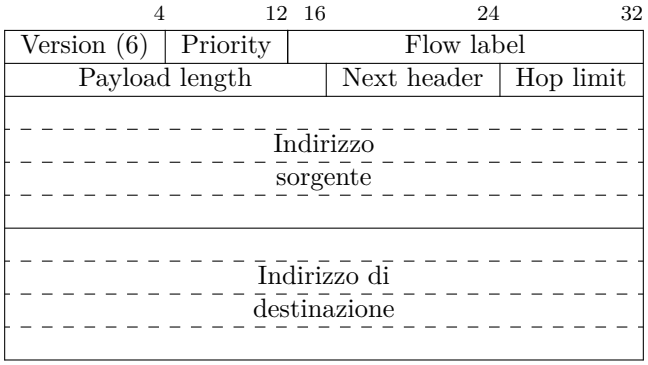


Un host può avere due interfacce (ad es. un'interfaccia Ethernet e una wi-fi) che possono essere connesse a due link diversi allo stesso tempo. Quando l'host vuole mandare il pacchetto a un indirizzo di destinazione link local, non sa se far uscire il pacchetto dall'interfaccia A o dall'interfaccia B, perché entrambi i link hanno lo stesso prefisso; inoltre, poiché ogni indirizzo link local è univoco entro il proprio link, un host nel link A potrebbe avere lo stesso indirizzo link local di un altro host nel link B.

In IPv6 l'host deve specificare nell'indirizzo IPv6 di destinazione un identificativo chiamato **scope** che è usato per identificare l'interfaccia fisica (per es. FE80::0237:00FF:FE02:A7FD%19). I valori degli scope vengono selezionati dal sistema operativo secondo i suoi criteri interni.

3.3 Intestazione IPv6 standard

L'intestazione IPv6 standard ha il seguente formato di dimensione fissa (40 byte):



dove i campi più significativi sono:

- campo Version (4 bit): non è molto usato, perché la discriminazione dei pacchetti è svolta dal livello 2 $\Rightarrow$ ciò permette l'approccio dual-stack (vedere la sezione 4.1.1);
- campo Priority (8 bit): equivalente al campo "Type of Service" di IPv4, permette di distinguere diversi tipi di servizi per la qualità del servizio (vedere la sezione 7.4.1);
- campo Flow label (20 bit): permette di distinguere diversi flussi per la qualità del servizio;
- campo Next header (8 bit): fa riferimento al payload del pacchetto, cioè un'intestazione di livello superiore (per es. TCP/UDP) oppure il primo extension header nella catena (vedere la sezione 3.4);
- campo Hop limit (8 bit): è equivalente al campo "Time To Live" di IPv4;
- campo Source address (128 bit): contiene l'indirizzo IPv6 sorgente del mittente del pacchetto;
- campo Destination address (128 bit): contiene l'indirizzo IPv6 di destinazione del destinatario del pacchetto.

Alcuni campi IPv4 sono stati rimossi:

- campo Checksum: la protezione dagli errori è demandata a livello 2 (frame check sequence);
- campo Fragmentation: la frammentazione è delegata all'extension header "Fragment";
- campo Header length: l'intestazione IPv6 è di dimensione fissa, poiché le funzionalità aggiuntive sono opzionalmente offerte dagli extension header.

3.4 Extension header

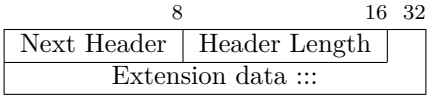
Ci sono sei extension header, aggiunti solo quando necessario ed elaborati nell'ordine seguente:

1. Hop by hop option: include informazioni facoltative che ogni hop deve elaborare (sezione 3.4.1);
2. Routing: consente il **source routing**, cioè la sorgente decide quale rotta deve prendere il pacchetto (sezione 3.4.2);
3. Fragment: gestisce la frammentazione (sezione 3.4.3);
4. Authentication Header (AH): consente di autenticare il mittente (sezione 3.4.4);
5. Encapsulating Security Payload (ESP): consente di criptare il contenuto del pacchetto (sezione 3.4.4);

6. Destination option: include informazioni facoltative che solo la destinazione deve elaborare (sezione 3.4.1).

I router elaborano sempre solo fino all'extension header "Routing".

Tutti gli extension header hanno lo stesso formato generico (la lunghezza dev'essere un multiplo di 64 bit):



dove i campi sono:

- campo Next Header: specifica l'extension header seguente nella catena, o l'intestazione di livello superiore (ad es. TCP/UDP) se questo è l'ultimo extension header;
- campo Header Length: specifica la lunghezza dell'extension header corrente.
  Poiché nel corso del tempo possono venire standardizzati nuovi extension header, i vecchi apparati potrebbero non essere in grado di elaborare gli extension header recenti ⇒ essi possono guardare il campo "Length" per saltare l'extension header sconosciuto.
  Il campo "Header Lengh" potrebbe non essere presente in alcuni extension header (come l'extension header "Fragment") che lo standard IPv6 definisce come aventi lunghezza fissa.

3.4.1 Hop by hop option e Destination option

Gli extension header **Hop by hop option** e **Destination option** possono includere più opzioni aggiuntive:

- Hop by hop option: include opzioni che deve elaborare ogni router attraverso cui passa il pacchetto;
- Destination option: include opzioni che deve elaborare solo la destinazione.

Per esempio, se ci sono due opzioni con valori lunghi 8 bit, l'extension header avrà il seguente formato:

8	16	24	32
Next Header	Header Length	Type1	Length1
Value1	Type2	Length2	Value2

in cui ogni opzione ha sempre i seguenti tre campi:

- campo Length (8 bit): specifica la lunghezza dell'opzione corrente, in modo che i router non in grado di riconoscere l'opzione possano limitarsi a saltarla;
- campo Type (8 bit): identifica l'opzione corrente.
  I primi due bit specificano sempre l'azione da eseguire nel caso in cui l'opzione non venga riconosciuta, mentre il terzo bit specifica se l'opzione può essere modificata al volo:

  00 l'opzione corrente può essere ignorata ed è possibile procedere alla prossima;

  01 il pacchetto deve essere scartato;

  10 il pacchetto deve essere scartato e deve essere generato un ICMPv6 Parameter Problem;

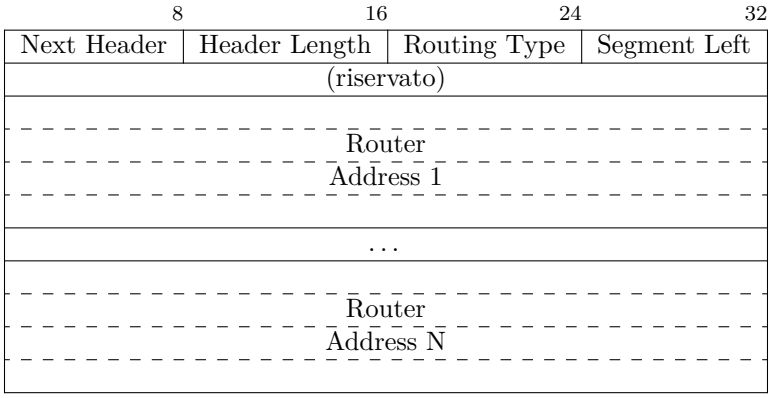
  11 il pacchetto deve essere scartato e deve essere generato un ICMPv6 Parameter Problem, a meno che l'indirizzo di destinazione non sia uno multicast;

  xx0 l'opzione non può essere modificata;

  xx1 l'opzione può essere modificata al volo;
- campo Value (lunghezza variabile): contiene il valore dell'opzione.

3.4.2 Routing

L'extension header **Routing** consente alla sorgente di decidere quale rotta deve prendere il pacchetto (**source routing**), e ha il formato seguente:



dove i campi sono:

- campo Routing Type (8 bit): specifica il tipo di instradamento (attualmente "0" per il classico source routing);
- campo Segment Left (8 bit): specifica il numero di hop rimanenti alla destinazione;
- campi Router Address (128 bit ciascuno): sono la lista degli indirizzi IPv6 dei router attraverso cui deve passare il pacchetto.

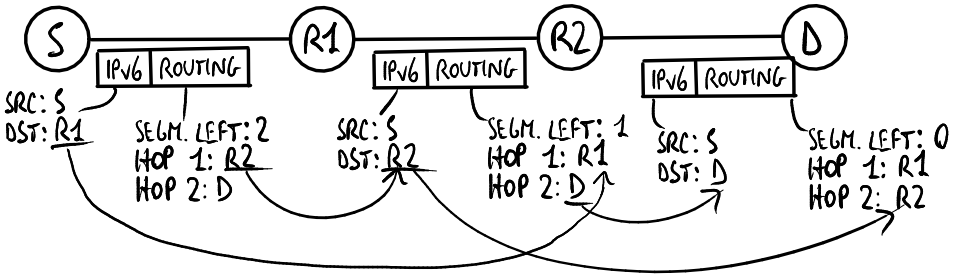


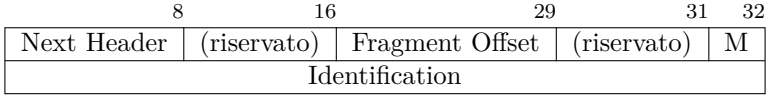
Figura 3.1: Esempio di impiego dell'extension header "Routing".

Nell'esempio in figura 3.1, la sorgente S invia il pacchetto verso la destinazione D, aggiungendo un extension header "Routing" che forza il pacchetto a passare per i router intermedi R1 e R2. Quindi all'inizio il pacchetto ha apparentemente il router R1 come destinazione, mentre la vera destinazione D è specificata come ultimo passo nella lista dei router specificata dall'extension header "Routing". Quando il pacchetto arriva al router R1, questo lo riconosce come apparentemente indirizzato ad esso; infatti, il suo indirizzo compare nel campo "Destination Address" nell'intestazione IPv6. Il router R1 controlla le intestazioni successive e scopre che il pacchetto contiene un extension header "Routing", rendendosi conto che la destinazione finale del pacchetto è un altro host (in particolare il campo "Segment Left" dice che è necessario attraversare due hop prima di arrivare alla destinazione finale). Il router R1 trova l'indirizzo IPv6 del prossimo hop a cui mandare il pacchetto e lo sostituisce con il proprio indirizzo IPv6, quindi manda il pacchetto con la destinazione impostata a R2. Il processo continuerà di hop in hop, finché la destinazione D non riceverà un pacchetto IPv6 il cui extension header "Routing" contiene il campo "Segment Left" impostato a 0, che significa che il pacchetto ha raggiunto la destinazione finale. La destinazione D è in grado di sapere tutti gli hop per i quali è passato il pacchetto perché essi

si trovano tutti scritti nell'extension header "Routing", quindi esso può inoltrare la risposta alla sorgente S specificando la stessa lista (inversa) di hop.

3.4.3 Fragment

L'extension header **Fragment** consente di inviare un pacchetto in parti più piccole chiamate "frammenti", e ha il formato seguente:



dove i campi sono:

- campo Fragment Offset (13 bit): specifica il numero del byte in cui inizia il frammento nella sezione frammentata del pacchetto originale;
- flag More Fragments (M) (1 bit): se è impostato a 0 il pacchetto corrente è l'ultimo frammento;
- campo Identification (32 bit): tutti i frammenti di uno specifico pacchetto hanno lo stesso identificativo.

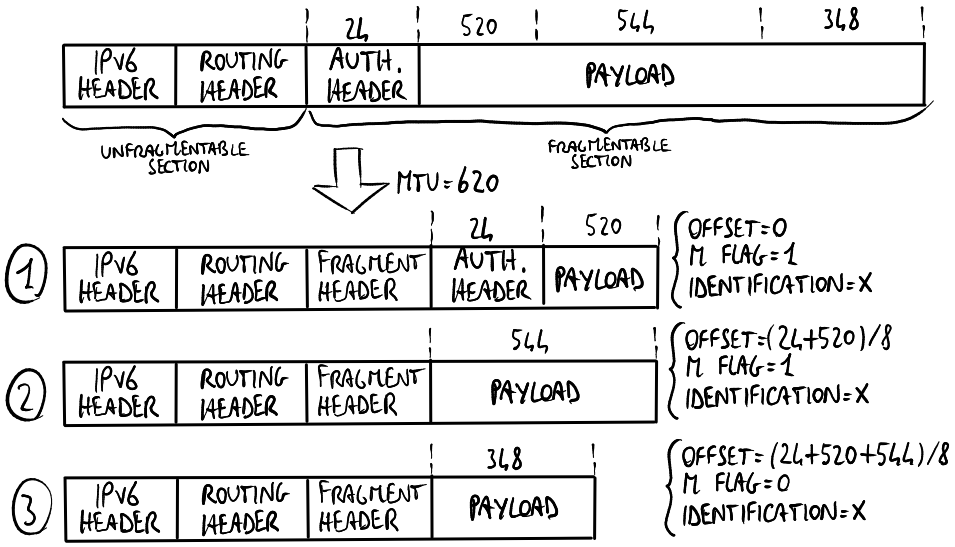


Figura 3.2: Esempio di impiego dell'extension header "Fragment".

Ogni pacchetto include due sezioni:

- una sezione che non può essere frammentata, così viene ripetuta in tutti i frammenti: include l'intestazione IPv6 e tutti gli extension header che precedono l'extension header "Fragment";
- una sezione che può essere frammentata: include tutti gli extension header che seguono l'extension header "Fragment" e il payload del pacchetto.

Al contrario di IPv4, solo al nodo mittente è consentito frammentare i datagram, mentre i router IPv6 non supportano la frammentazione. Inoltre, lo standard IPv6 suggerisce fortemente di utilizzare il Path MTU Discovery al posto della frammentazione per motivi prestazionali (vedere la sezione 3.5.1).

3.4.4 IPsec

Le soluzioni sviluppate per IPv6 derivano dal porting della suite di protocolli IPsec di IPv4. In IPv6 IPsec è una suite di protocolli integrata che definisce due intestazioni:

- Authentication Header (AH): autentica l'intero pacchetto, tranne i campi che cambiano al passaggio da un hop all'altro (per es. limite di hop), garantendo che nessuno abbia modificato il contenuto del pacchetto;
- Encapsulating Security Payload (ESP): autentica e cripta il payload del pacchetto per la privacy dei dati.

SA

IPsec non definisce quali algoritmi devono essere usati per la crittografia e l'autenticazione, ma le due parti devono concordare su quali usare per scambiare informazioni protette con IPsec ⇒ flessibilità: gli algoritmi vengono scelti sulla base delle necessità del momento.

Una **Security Association** (SA) si può definire come l'insieme degli accordi tra le due parti A e B riguardo le chiavi e gli algoritmi privati da usare per l'autenticazione e la crittografia ESP e per l'autenticazione AH. Ogni SA è identificata da un tag di identificazione chiamato **Security Parameter Index** (SPI), incluso nelle intestazioni AH e ESP, ed è un canale logico unidirezionale: A e B devono aprire una SA per concordare sulle chiavi e sugli algoritmi per i messaggi che vanno da A a B, e devono aprire un'altra SA per concordare su di essi per i messaggi che vanno da B a A. Spesso per ogni porta TCP viene aperta una SA.

IKE

A e B come fanno a mettersi d'accordo su chiavi segrete evitando che degli estranei le sappiano? Ci sono tre principali strategie:

- configurazione statica: le chiavi sono configurate manualmente in A e B ⇒ la negoziazione delle chiavi non è richiesta affatto;
- metodo di Diffie-Hellman: permette di concordare su una chiave senza scambiarla ⇒ nessuno può scoprire le chiavi segrete sniffando il traffico tra A e B;
- protocollo **Internet Key Exchange** (IKE): utilizza i certificati digitali e la crittografia asimmetrica per inviare le chiavi segrete in un modo sicuro.

Il protocollo IKE specifica che si deve stabilire una SA IKE da A a B per concordare sulle chiavi segrete per la SA figlia da A a B, e viceversa un'altra per la SA figlia da B ad A. La SA IKE da A a B consiste delle seguenti operazioni basate sulla **crittografia asimmetrica**:[1]

1. B chiede ad A una chiave segreta da usare per la SA figlia da A a B;
2. A chiede ad una autorità di certificazione fidata il certificato digitale di B, per sapere se B è veramente chi dice di essere;
3. l'autorità di certificazione fornisce ad A il **certificato digitale** di B, criptato usando la chiave privata dell'autorità di certificazione, contenente la firma di B, cioè l'associazione tra B e una chiave pubblica;
4. A decripta il certificato digitale usando la chiave pubblica dell'autorità di certificazione ed apprende la chiave pubblica associata a B;
5. A manda a B la chiave segreta per la SA figlia, criptando il messaggio usando la chiave pubblica associata a B in modo che possa essere decriptato solo conoscendo la chiave privata di B;

[1] Per semplicità si suppone che è necessaria una sola chiave segreta per la SA.

6. B riceve il messaggio da A, lo decripta usando la sua chiave privata ed apprende la chiave segreta decisa da A per la SA figlia;

7. si può aprire la SA figlia da A a B che usa la chiave segreta concordata.

Degli estranei potrebbero guardare il traffico scambiato tra A e B e indovinare le chiavi segrete dopo un po' di tempo, effettuando degli attacchi brute-force o analizzando alcune informazioni statistiche dedotte. **Internet Security Association Key Management Protocol** (ISAKMP) è un sotto-protocollo di IKE per rinegoziare periodicamente le chiavi segrete in un modo sicuro, in modo che gli estranei non abbiano tempo per indovinarle.

AH

L'**Authentication Header** (AH) garantisce integrità connectionless e autenticazione dell'origine dei dati per i pacchetti IP: autentica l'intero pacchetto, tranne i campi che cambiano al passaggio da un hop all'altro (per es. campo "Hop limit"), garantendo che nessuno abbia modificato il contenuto del pacchetto.

AH ha dei problemi con i NAT, perché autentica anche gli indirizzi e le porte.

Concetto chiave: nessuno può modificare il pacchetto, tutti possono leggerlo.

L'Authentication Header ha il seguente formato:

8	16	32
Next Header	Payload Length	(riservato)
SPI		
Sequence Number		
Authentication Data :::		

dove i campi sono:

- campo Next Header (8 bit): specifica il prossimo protocollo incapsulato;
- campo Payload Length (8 bit): specifica la dimensione dell'Authentication Header in parole da 32 bit − 2 (può essere azzerato);
- campo Security Parameters Index (SPI) (32 bit): identifica la Security Association per questo datagram (se azzerato, non esiste una Security Association; i valori nell'intervallo da 1 a 255 sono riservati);
- campo Sequence Number (32 bit): contiene un valore contatore monotonicamente crescente;
- campo Message Digest (lunghezza variabile): riassume il contenuto del pacchetto utilizzando una chiave segreta: tutti coloro che vogliono modificare il contenuto del pacchetto devono sapere la chiave per ricalcolare il message digest (simile al campo per il rilevamento degli errori).

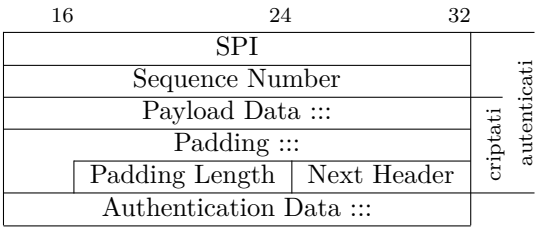
ESP

L'intestazione **Encapsulating Security Payload** (ESP) fornisce autenticità dell'origine, integrità e tutela della riservatezza per i pacchetti IP: autentica e cripta il payload del pacchetto per la privacy dei dati.

Sebbene ESP sia in grado di autenticare, non effettua la stessa funzionalità di AH: ESP non autentica l'intero pacchetto IPv6.

Concetto chiave: nessuno può leggere il pacchetto, perciò nessuno può modificarlo.

L'intestazione ESP è sempre l'ultima nella catena delle intestazioni e ha il seguente formato:

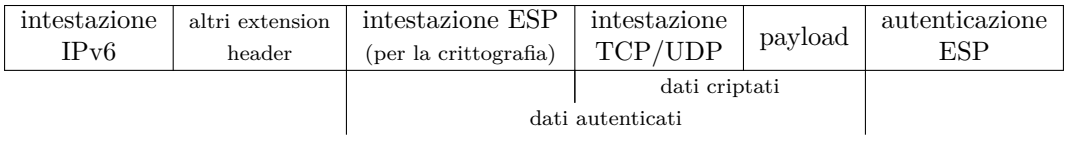


dove i campi sono:

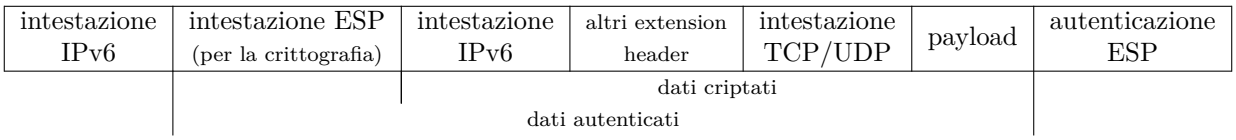
- campo Security Parameters Index (SPI) (32 bit): identifica la Security Association per questo datagram;
- campo Sequence Number (32 bit senza segno): contiene un valore contatore monotonicamente crescente.
  Il campo "Sequence number" è obbligatorio per il trasmettitore ed è sempre presente anche se il ricevitore non sceglie di attivare il servizio anti-replay per una specifica SA, ma l'elaborazione di questo campo è a discrezione del ricevitore;
- campo Payload Data (lunghezza variabile): contiene i dati descritti dal campo "Next Header";
- campo Padding (lunghezza variabile da 0 a 255 bit): può essere richiesto del padding, a prescindere dai requisiti degli algoritmi di crittografia, per assicurare che il testo cifrato risultante termini su un boundary di 4 byte;
- campo Padding Length (8 bit): specifica la dimensione del campo "Padding" (in byte);
- campo Next Header (8 bit): un numero di protocollo IPv4/IPv6 che descrive il formato del campo "Payload Data";
- campo Authentication Data (lunghezza variabile): contiene un Integrity Check Value (ICV) calcolato sul pacchetto ESP meno il campo "Authentication Data".
  La lunghezza del campo "Authentication Data" è specificata dalla funzione di autenticazione selezionata. Il campo "Authentication Data" è facoltativo: viene incluso solo se il servizio di autenticazione è stato selezionato per la SA in questione. La specifica dell'algoritmo di autenticazione deve specificare la lunghezza dell'ICV e le regole di confronto e le fasi di elaborazione per la convalida. Si noti che il campo "Authentication Data" non è criptato.

Sono possibili due modi d'uso per ESP (opzionalmente in combinazione con AH):

- **transport mode**: ESP non cripta l'intestazione IPv6 ⇒ chiunque nel mezzo è in grado di vedere gli indirizzi IP sorgente e di destinazione nell'intestazione IPv6:



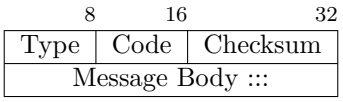
- **tunnel mode**: è possibile incapsulare un pacchetto IPv6 in un altro pacchetto IPv6 avente ESP (e opzionalmente AH) per proteggere anche l'intestazione IPv6 (contenente gli indirizzi IP sorgente e di destinazione):



3.5 ICMPv6

Internet Control Message Protocol version 6 (ICMPv6) è parte integrante dello standard IPv6, e a sua volta integra espandendole le funzionalità dei protocolli ARP e IGMP.

Tutti i messaggi ICMPv6 vengono posti subito dopo gli extension header nel pacchetto, e hanno lo stesso formato generico:



dove il campo "Type" identifica il tipo di messaggio ICMPv6:

- messaggi per la diagnostica: come in ICMPv4, consentono di segnalare errori o problemi nella rete:

  1 = Destination Unreachable

  2 = Packet Too Big (sezione 3.5.1)

  3 = Time Exceeded

  4 = Parameter Problem

- messaggi utilizzati dal comando `ping`:

  128 = Echo Request

  129 = Echo Reply

- messaggi Multicast Listener Discovery: espandono le funzionalità di IGMP (sezione 3.5.2):

  130 = Multicast Listener Query

  131 = Multicast Listener Report

  132 = Multicast Listener Done

- messaggi Neighbor Discovery: espandono le funzionalità di ARP (sezione 3.5.3):

  133 = Router Solicitation

  134 = Router Advertisement

  135 = Neighbor Solicitation

  136 = Neighbor Advertisement

  137 = Redirect

3.5.1 Packet Too Big

Quando un router riceve un pacchetto che ha una dimensione troppo grande, effettua una tecnica chiamata **Path MTU Discovery**: scarta il pacchetto e invia in risposta un messaggio ICMPv6 di tipo **Packet Too Big** al fine di notificare al mittente la dimensione della Maximum Transmission Unit (MTU) consentita e di forzarlo a rimandare il pacchetto stesso (e i pacchetti successivi) con una dimensione che non superi la MTU specificata dal router. Questa tecnica ha l'obiettivo di evitare il più possibile la frammentazione.

3.5.2 Multicast Listener Discovery

Multicast Listener Discovery è la componente in ICMPv6 che espande le funzionalità del protocollo IGMP di IPv4 per gestire le appartenenze ai gruppi multicast:

- **Multicast Listener Query**:
  - General Query: il router chiede agli host se sono interessati ad appartenere a un qualche gruppo multicast;
  - Multicast Address Specific Query: il router chiede agli host se sono interessati ad appartenere a un particolare gruppo multicast;
- **Multicast Listener Report**: l'host notifica al router che vuole appartenere a un particolare gruppo multicast per ricevere tutti i pacchetti multicast indirizzati all'indirizzo multicast corrispondente al gruppo multicast specificato;
- **Multicast Listener Done**: l'host notifica al router che vuole smettere di ricevere i pacchetti multicast di un particolare gruppo multicast.

3.5.3 Neighbor Discovery

Neighbor Discovery è la componente in ICMPv6 che espande le funzionalità del protocollo ARP di IPv4:

- **Neighbor Solicitation**: l'host invia un pacchetto multicast avente, come indirizzo IPv6 di destinazione, l'indirizzo multicast solicited node corrispondente all'indirizzo IPv6 di cui vuole apprendere l'indirizzo MAC;
- **Neighbor Advertisement**: l'host avente l'indirizzo IPv6 specificato invia in risposta il suo indirizzo MAC;
- **Router Solicitation**: l'host invia un pacchetto multicast per sollecitare il router a inviare in risposta un messaggio "Router Advertisement" contenente l'interface identifier associato al link;
- **Router Advertisement**: il router annuncia la propria presenza nel link segnalando l'interface identifier associato al link.

I messaggi ICMPv6 "Neighbor Discovery" si usano per autoconfigurare gli indirizzi IPv6 di un host che si connette a un link: prima l'host deve ottenere un indirizzo link local per poter contattare gli altri host nel link, quindi deve ottenere un indirizzo global per poter uscire dal link e accedere a Internet con un indirizzo globalmente univoco.

Processo di autoconfigurazione dell'indirizzo link local

L'indirizzo link local viene autoconfigurato utilizzando i messaggi ICMPv6 "Neighbor Solicitation" e "Neighbor Advertisement":

1. l'host genera da sé un indirizzo IPv6 candidato ad essere il suo indirizzo link local:
   - prefisso: è sempre "FE80::";
   - interface identifier: può essere generato basato sull'indirizzo MAC (formato EUI-64) oppure in modo casuale per motivi di privacy (tracciabilità);
2. l'host invia in multicast un messaggio "Neighbor Solicitation" a tutti gli host nel link, specificando come indirizzo IPv6 di destinazione il suo indirizzo auto-generato e chiedendo se esiste nel link un host il cui indirizzo link local è uguale all'indirizzo IPv6 specificato (**Duplicated Address Detection**);

3. se esiste già nel link un host avente l'indirizzo link local del mittente, esso invia in risposta un messaggio "Neighbor Advertisement" al mittente, che dovrà generare in modo casuale un altro indirizzo candidato e inviare in multicast un altro messaggio "Neighbor Solicitation";

4. se nessuno risponde, l'indirizzo è univoco nel link e l'host è in grado di contattare ogni altro host entro lo stesso link utilizzando il suo indirizzo link local, ma non è ancora in grado di accedere a Internet perché ha bisogno di un indirizzo global.

Processo di autoconfigurazione dell'indirizzo global

L'indirizzo global viene autoconfigurato utilizzando i messaggi ICMPv6 "Router Solicitation", "Router Advertisement", "Neighbor Solicitation" e "Neighbor Advertisement":

1. l'host invia in multicast un messaggio "Router Solicitation" per sollecitare il router a inviare in risposta un messaggio "Router Advertisement" contenente l'interface identifier associato al link;[2]

2. il router invia in risposta un messaggio "Router Advertisement" contenente i due flag "Managed Address Configuration" (M) e "Other configuration" (O):

   - M = 1: l'host deve contattare il server DHCP per il prefisso del link e gli altri parametri di configurazione della rete (come l'indirizzo del DNS), senza curarsi dei messaggi "Router Advertisement" dal router (**configurazione stateful**);
   - M = 0: l'host deve guardare il flag "O":
     - O = 1: l'host può prendere il prefisso del link dal messaggio "Router Advertisement", ma deve comunque contattare il server DHCP per gli altri parametri di configurazione della rete (come l'indirizzo del DNS);
     - O = 0: l'host può prendere il prefisso del link dal messaggio "Router Advertisement", e nessun'altra informazione di configurazione è disponibile dal server DHCP (**configurazione stateless**) ⇒ gli altri parametri di configurazione della rete (come l'indirizzo del DNS) dovranno essere configurati a mano sull'host, oppure l'host può ottenere l'indirizzo del server DNS tramite IPv4 (vedi la sezione 4.1.3);

3. l'host genera da sé un indirizzo IPv6 candidato ad essere il suo indirizzo global:

   - prefisso: è uguale al prefisso del link, preso dal messaggio "Router Advertisement" oppure contattando il server DHCP;
   - interface identifier: può essere generato basato sull'indirizzo MAC (formato EUI-64) oppure in modo casuale per motivi di privacy (tracciabilità);

4. l'host invia in multicast un messaggio "Neighbor Solicitation" a tutti gli host nel link, specificando come indirizzo IPv6 di destinazione il suo indirizzo auto-generato e chiedendo se esiste nel link un host il cui indirizzo global è uguale all'indirizzo IPv6 specificato (**Duplicated Address Detection**);

5. se esiste già nel link un host avente l'indirizzo global del mittente, esso invia in risposta un messaggio "Neighbor Advertisement" al mittente, che dovrà generare in modo casuale un altro indirizzo candidato e inviare in multicast un altro messaggio "Neighbor Solicitation";

6. se nessuno risponde, l'indirizzo è globalmente univoco e l'host è in grado di accedere a Internet utilizzando il suo indirizzo global.

Un'altra implementazione proposta da Microsoft consiste nella possibilità per l'host di contattare il server DNS senza conoscerne l'indirizzo: l'host manda i pacchetti a un indirizzo anycast fisso, e la rete si prende cura di consegnare il pacchetto al server DNS. Tuttavia questa implementazione non è in realtà usata:

[2] Questo passo non è obbligatorio se il router è configurato per inviare periodicamente in multicast messaggi "Router advertisement".

- le implementazioni per la gestione degli indirizzi anycast sono rare;
- questa soluzione non è supportata dal sistema operativo GNU/Linux.

L'autoconfigurazione è basata sull'indirizzo MAC, così se la scheda di rete si rompe e richiede di essere sostituita l'host dovrà cambiare il proprio indirizzo, ma le cache (ad es. la cache DNS) non sono in grado di aggiornarsi immediatamente ⇒ è sempre possibile la configurazione statica, specialmente per le macchine fisse (ad es. i server di siti Web pubblici) che devono evitare di cambiare i loro indirizzi al fine di continuare ad essere raggiungibili in un modo il più continuo possibile.

Capitolo 4

Migrazione a IPv6

4.1 Introduzione

Durante la fase di transizione, gli host devono cominciare gradualmente a poter raggiungere le destinazioni IPv6 continuando a essere in grado di raggiungere le destinazioni IPv4. La migrazione di tutti gli apparati di rete è una condizione necessaria ma non sufficiente: l'utente deve farli lavorare insieme creando un nuovo piano di indirizzamento per l'intera rete.

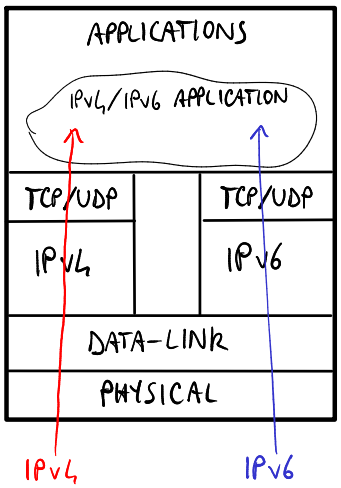
4.1.1 Migrazione degli host

Migrazione delle applicazioni

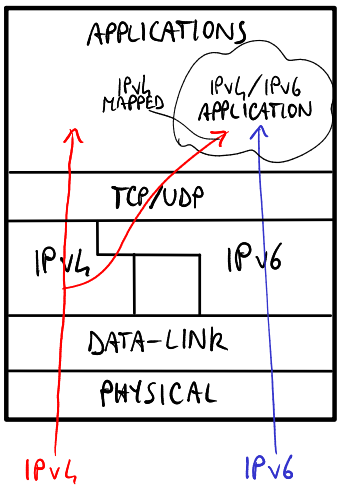
L'introduzione del supporto a IPv6 nelle applicazioni comporta la necessità di modificare il codice sorgente:

- server: il processo in esecuzione su un server deve aprire due thread, uno in ascolto sul socket IPv4 e l'altro in ascolto sul socket IPv6, al fine di poter servire richieste sia IPv4 sia IPv6;
- client: le applicazioni come i browser Web devono essere in grado di stampare in output e ricevere in input gli indirizzi nel nuovo formato.

Migrazione dei sistemi operativi



(a) Dual stack senza dual layer.



(b) Dual stack con dual layer.

Le applicazioni giacciono per lo più sulle librerie del sistema operativo, che possono introdurre il supporto a IPv6 adottando l'approccio **dual stack**:

- senza dual layer: il sistema operativo elabora indipendentemente gli indirizzi IPv4 e IPv6 ⇒ il software deve essere in grado di gestire indirizzi sia IPv4 sia IPv6;
- con dual layer: il sistema operativo è in grado di convertire un indirizzo IPv4 in un indirizzo IPv6 IPv4-mapped ⇒ il software può limitarsi a supportare gli indirizzi IPv6 senza curarsi degli indirizzi IPv4.

La variante con dual layer è la più utilizzata perché sposta la complessità al core del sistema operativo.

4.1.2 Migrazione degli apparati di rete

Migrazione degli switch

Anche se in teoria gli switch non dovrebbero essere influenzati assolutamente dai cambiamenti a livello 3 perché essi lavorano fino al livello 2, ci potrebbero essere alcuni problemi con le funzioni aggiuntive: per esempio l'**IGMP snooping**, una funzionalità utilizzata per filtrare i pacchetti multicast in arrivo, ha bisogno di guardare all'interno del pacchetto ⇒ poiché cambiano il formato e i campi del pacchetto lo switch non riesce a riconoscere i pacchetti IPv6 multicast e li scarta.

Migrazione dei router[1]

Oggi i router sono per lo più pronti per IPv6, anche se le prestazioni in IPv6 sono ancora peggiori rispetto a quelle in IPv4 a causa della mancanza di esperienza e della più bassa domanda di traffico.

Tipicamente i router che supportano IPv6 adottano l'approccio dual stack di tipo "navi nella notte": IPv4 e IPv6 sono supportati da due pile indipendenti per il livello trasporto ⇒ questo richiede la completa duplicazione di tutti i componenti: protocolli di instradamento, tabelle di instradamento, liste di accesso, ecc.

Tabelle di instradamento L'instradamento in IPv6 è effettuato nello stesso modo di IPv4 ma richiede due tabelle di instradamento distinte, una per le rotte IPv4 e l'altra per le rotte IPv6. Le tabelle di instradamento IPv6 possono memorizzare diversi tipi di entry, tra cui:

- entry indirette (codici O/S): specificano gli indirizzi, tipicamente link local, delle interfacce dei router next hop ai quali inviare i pacchetti indirizzati verso link remoti;
- entry dirette: specificano le interfacce del router stesso attraverso le quali inviare i pacchetti indirizzati verso i link locali:
  - reti connesse (codice C): specificano i prefissi dei link locali;
  - indirizzi di interfaccia (codice L): specificano gli interface identifier nei link locali.

Protocolli di instradamento I protocolli di instradamento che supportano IPv6 possono adottare due approcci:

- instradamento integrato (ad es. BGP): il protocollo consente di scambiare informazioni di instradamento sia IPv4 sia IPv6 allo stesso tempo ⇒ gli indirizzi IPv4 e IPv6 appartenenti alla stessa destinazione possono essere trasportati tramite un singolo messaggio ⇒ efficienza maggiore;

[1] Si rimanda al capitolo *Instradamento IPv6* negli appunti di "Protocolli e architetture di routing".

- navi nella notte (ad es. RIP, OSPF): il protocollo consente di scambiare informazioni di instradamento solo IPv6 ⇒ data una destinazione, deve essere scambiato un messaggio per il suo indirizzo IPv4 e un altro messaggio per il suo indirizzo IPv6, e i messaggi sono completamente indipendenti tra loro ⇒ flessibilità maggiore: si possono usare due protocolli diversi, uno per le informazioni di instradamento IPv4 e un altro per le informazioni di instradamento IPv6.

4.1.3 Migrazione dei DNS

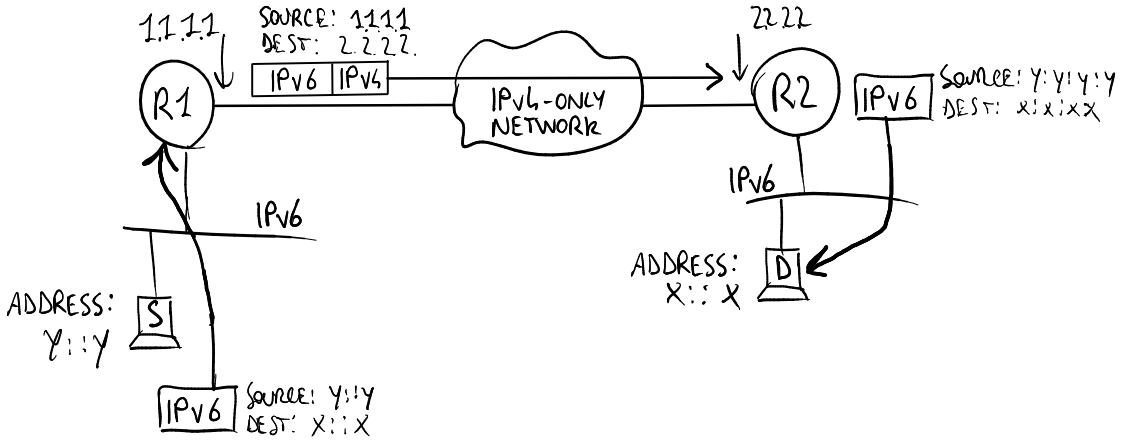
I DNS che supportano IPv6 possono mappare due indirizzi IP allo stesso alias: un indirizzo IPv4 e uno IPv6 ⇒ una destinazione pubblica può essere raggiungibile tramite IPv4 oppure tramite IPv6.

I DNS che supportano IPv6 possono restituire gli indirizzi IPv6 non solo tramite IPv6, ma anche tramite IPv4: i messaggi DNS appartengono infatti al livello applicazione, così non importa il livello trasporto utilizzato per inoltrare le query e le risposte DNS. Le query DNSv6 vengono effettuate con il comando seguente: `set q=aaaa`.

Una società può decidere di offrire l'accesso al proprio sito Web pubblico anche tramite IPv6. Tuttavia, attualmente la maggior parte del traffico è tramite IPv4, così generalmente il servizio per il traffico IPv4 è più affidabile in termini di prestazioni e tolleranza ai guasti rispetto a quello per il traffico IPv6. Perciò la società, soprattutto se basa il business sul suo sito Web, non vuole che l'utente che si connette tramite IPv6 decida di passare a un altro sito Web concorrente a causa di problemi prestazionali. Una soluzione possibile è effettuare alcuni accertamenti preliminari per testare le prestazioni della connettività tra l'utente e il server della società, e implementare un meccanismo aggiuntivo nei DNS: essi dovrebbero essere capaci di guardare l'indirizzo sorgente della query DNS, e restituire solamente l'indirizzo IPv4 se non è stato effettuato alcun accertamento per la connettività, oppure entrambi gli indirizzi IPv4 e IPv6 se le prestazioni sono sufficientemente buone.

4.2 Soluzioni di tunneling

La rete non sarà compatibile con IPv6 dal giorno zero ⇒ il traffico IPv6 potrebbe dover attraversare delle porzioni di rete solo IPv4. Le soluzioni di tunneling orientate alla rete permettono la connettività tra le reti IPv6 anche se esse sono connesse attraverso un'infrastruttura solo IPv4, e consistono nell'incapsulamento del pacchetto IPv6 all'interno di un'intestazione IPv4 solo per il trasporto lungo il tunnel:



La dimensione del pacchetto nel tunnel, compresa l'intestazione IPv4 lunga 20 byte, non deve superare la dimensione massima dei pacchetti IPv4 ⇒ sono possibili due soluzioni:

- frammentazione: i router dovrebbero frammentare il pacchetto IPv4 prima di mandarlo nel tunnel ⇒ la frammentazione è deprecata per motivi prestazionali;
- pacchetti IPv6 più piccoli: gli host dovrebbero generare dei pacchetti IPv6 con una dimensione della MTU più piccola per tenere conto della dimensione supplementare dovuta all'inserimento dell'intestazione IPv4 ⇒ i router possono specificare la dimensione della MTU consentita attraverso i messaggi ICMPv6 "Router Advertisement".

4.2.1 Soluzioni di tunneling orientate agli host

Le soluzioni di tunneling orientate agli host sono più plug-and-play per gli host, ma non sono soluzioni professionali e non risolvono il problema della scarsità di indirizzi IPv4 perché ogni host ha ancora bisogno di avere un indirizzo IPv4. Le soluzioni di questo tipo sono:

- uso degli indirizzi IPv6 IPv4-compatible, con terminazione del tunnel sull'host o sul router;
- 6over4;
- ISATAP.

Uso degli indirizzi IPv6 IPv4-compatible

Si basano sul fatto che gli host dual-stack, quando è necessario contattare una destinazione IPv4, inviino pacchetti IPv6 ad un indirizzo IPv6 IPv4-compatible, cioè costruito con i primi 96 bit alti a zero e con i restanti 32 coincidenti con quelli dell'indirizzo IPv4 di destinazione. Tale pacchetto IPv6 viene poi incapsulato in un pacchetto IPv4, la cui destinazione è diversa a seconda che si voglia terminare il tunnel sull'host di destinazione o su un router dual-stack, in particolare:

- terminazione end-to-end: la pseudo-interfaccia sull'host dual-stack effettua l'incapsulamento in un pacchetto IPv4 destinato all'host che si vuole contattare;
- terminazione sul router dual-stack: la pseudo-interfaccia sull'host manda i pacchetti destinati ad un host all'indirizzo IPv4 del router dual-stack, quindi:
  1. si genera un indirizzo IPv6 IPv4-compatible per la destinazione, come prima;
  2. si incapsula il pacchetto IPv6 in uno IPv4 destinato al router dual-stack;
  3. il router dual-stack decapsula il pacchetto e invia all'host di destinazione.

6over4

L'idea è quella di emulare, attraverso IPv4, una rete locale che ha supporto al multicast. In pratica, come per connettere due host IPv6 attraverso la rete Ethernet sottostante si usa la neighbor discovery, appoggiandosi al fatto che Ethernet ha dei meccanismi per il broadcast, in questa soluzione si ragiona come se fosse IPv4 il protocollo di livello inferiore e si modifica la neighbor discovery per trovare indirizzi IPv4 al posto di indirizzi MAC. Questo discorso può essere generalizzato al caso in cui si vogliono connettere non dei singoli host, ma nuvole di reti IPv6 attraverso router dual-stack che comunicano in una rete IPv4. In questo caso, oltre alla neighbor discovery, si può utilizzare una versione modificata della router discovery, in modo da inviare una router solicitation per scoprire gli indirizzi IPv4 dei router connessi alla rete IPv4 dell'host che consentono di raggiungere varie reti IPv6; infatti dalla router advertisement l'host può avere informazioni sulle reti IPv6 che si possono raggiungere da quel router.

Il problema di questa soluzione è l'uso dell'IPv4 multicast, che di solito è disabilitato nelle reti che coinvolgono provider diversi. Questa soluzione è utilizzabile quando si ha una rete tutta sotto il proprio controllo: per questo motivo essa non è utilizzabile per migrare la rete globale da IPv4 a IPv6.

Neighbor discovery di 6over4 Su proposta dell'RFC, gli indirizzi IPv6 sono mappati sugli indirizzi IPv4: in pratica l'indirizzo IPv4 viene usato come interface identifier dell'indirizzo IPv6 della destinazione. Ciò renderebbe inutile il meccanismo illustrato finora, perché l'host potrebbe effettuare il tunneling direttamente, senza bisogno della neighbor discorvery per conoscere l'indirizzo IPv4. Ciò ovviamente non è valido quando l'indirizzo IPv6 non è costruito a partire da quello IPv4, quindi per contattare un router è comunque necessario un meccanismo più generale. Supponendo, quindi, di conoscere soltanto un indirizzo IPv6, si invia la neighbor discovery al solicited node multicast address (ad esempio se l'indirizzo IPv6 è `fe80::101:101` allora si manda a `ff02::1:ff01:101`) su una rete IPv4 6over4 multicast all'inidirizzo `239.192.x.y`, costruito con gli ultimi 16 bit dell'indirizzo IPv6 (quindi nell'esempio precedente sarà `239.192.1.1`).

ISATAP

L'idea è simile a quella del 6over4, cioè usare la rete IPv4 come link fisico per raggiungere le destinazioni IPv6, ma si vuole superare la limitazione di richiedere il supporto al multicast. In assenza dei meccanismi di neighbor discovery, l'indirizzo IPv4 delle destinazioni che usano ISATAP viene incorporato nell'indirizzo IPv6, più precisamente nell'interface identifier, che ha la forma `0000:5efe:x:y`, in cui `x` e `y` sono i 32 bit dell'indirizzo IPv4. Come si intuisce, questa soluzione non affronta il problema per cui si è introdotto IPv6, cioè la scarsità di indirizzi IPv4. Questa soluzione è però più utile nello scenario di un link IPv4 che connette non host, ma router cha hanno al confine delle nuvole IPv6. In questo caso un host all'interno della rete IPv4 che voglia comunicare in IPv6 con un host appartenente ad una nuvola dovrà essere dotato di una Potential Router List (PRL). I problemi che a questo punto si pongono sono:

- Come viene acquisita la PRL?

  Ci sono due diverse soluzioni: la prima, che è proprietaria, si basa sull'uso del DHCP; la seconda, che è standard, si basa sull'uso del DNS. Nella seconda si usa una query DNS per un nome particolare dal formato `isatap.dominio.it`, che fornirà la PRL dei router IPv6 collegati alla rete IPv4 del dominio specificato nella query.

- A quale router devono essere inviati i pacchetti per la destinazione IPv6?

  Si usa una router discovery unicast ad ognuno dei router della PRL, in modo da farsi rispondere con una router advertisement. Si ricordi, infatti, che nella router advertisement i router possono anche annunciare la lista delle reti IPv6 che si possono raggiungere tramite essi (vedi il flag `L=0` nella *Prefix Information Option* della *ICMP Router Advertisement*).

4.2.2 Soluzioni di tunneling orientate alla rete

Tipicamente le soluzioni di tunneling orientate alla rete richiedono la configurazione manuale, e l'incapsulamento può essere basato su IPv6 in IPv4 (protocol type = 41), GRE (vedere la sezione 5.2.2), IPsec, ecc.

6to4

Il più grande passo in avanti rispetto alle soluzioni precedenti viene dalla considerazione che nel nuovo scenario c'è una intera rete IPv6 che ha bisogno di un indirizzo IPv4 per uscire dalla nuvola IPv6, non più un singolo host. Il mapping tra i due indirizzi viene quindi effettuato nel prefisso IPv6, non nell'interface identifier: si assegna a tutte le reti IPv6 un prefisso speciale che includa l'indirizzo IPv4 assegnato all'interfaccia del router dual-stack che si affaccia sulla nuvola. Il prefisso `2002::/16` identifica delle stazioni IPv6 che stanno usando 6to4: nei successivi 32 bit si pone l'indirizzo IPv4 e altri 16 rimangono disponibili per rappresentare più subnet diverse, mentre l'interface identifier si ottiene come negli altri casi di uso di IPv6. In questa soluzione esiste anche un router che ha un ruolo particolare, il **6to4 Relay**, che deve essere il default gateway dei router 6to4, in modo da inoltrare alla IPv6 globale i pacchetti che non hanno il formato dei 6to4 appena visto. Questo router ha indirizzo `192.88.99.1`, che è un indirizzo anycast: è stato

usato da chi ha pensato il 6to4 perchè si è pensato allo scenario in cui nella stessa rete ci siano più 6to4 Relay, da cui nascerebbe il problema di dover usare indirizzi diversi. In questo modo invece, poichè l'indirizzo anycast è processato in maniera diversa dai protocolli di routing, si può usare lo stesso indirizzo e fornire anche del *load balancing*.

Esempio pratico Ipotizziamo che ci siano due nuvole IPv6 collegate ad una nuvola IPv4, e che le interfacce dei router dual-stack abbiano, per le interfacce collegate alle nuvole IPv6, gli indirizzi `192.1.2.3` per la rete A e `9.254.2.252` per la rete B. Supponiamo inoltre che un host a appartenente alla rete A voglia inviare un pacchetto all'host b appartenente alla rete B. Dalla configurazione delineata e da quanto detto si ricava che gli host presenti nella rete A avranno un indirizzo del tipo `2002:c001:02:03/48` e quelli presenti nella rete B `2002:09fe:02fc::/48`. Il pacchetto IPv6 da a a b sarà incapsulato in un pacchetto IPv4 che ha come indirizzo di destinazione `9.254.2.252`, ricavato dal prefisso dell'indirizzo IPv6 di destinazione: quando il pacchetto arriverà a quel router, esso sarà decapsulato e inoltrato secondo il piano di indirizzamento IPv6 della nuvola contenente la rete B.

Teredo

È molto simile al 6to4, tranne per il fatto che l'incapsulamento è effettuato dentro un segmento UDP contenuto in una pacchetto IPv4, al posto di essere semplicemente incapsulato in IPv4. Questo si fa per superare un limite del 6to4, cioè il passaggio attraverso i NAT: poichè nel 6to4 non è presente un segmento di livello 2 dentro il pacchetto IPv4 incapsulante, il NAT non può funzionare.

Tunnel broker

Il problema della soluzione 6to4 è che non è sufficientemente generica: si è vincolati all'uso degli indirizzi `2002::/16` e non si possono usare i comuni global unicast. Nella soluzione con tunnel broker, visto che non è più possibile dedurre dal prefisso IPv6 quale sia l'endpoint a cui mandare il pacchetto, si utilizza un server che, dato un generico indirizzo IPv6, fornisce l'indirizzo del tunnel endpoint da contattare. I router che implementano il tunnel broker sono chiamati **tunnel server**, mentre i server che forniscono il mapping si chiamano **tunnel broker server**. I tunnel sono realizzati come in 6to4, quindi IPv6 dentro IPv4: se ci fosse il problema di attraversare un NAT si potrebbe anche pensare di usare l'approccio di Teredo, incapsulando dentro UDP, quindi dentro IPv4.

Il tunnel broker server deve essere configurato: viene usato il **Tunnel Information Control** (TIC) per inoltrare le informazioni sulle reti raggiungibili da un determinato tunnel server, dal tunnel server che si sta configurando al tunnel broker server. Il **Tunnel Setup Protocol** (TSP) viene invece usato per chiedere le informazioni al tunnel broker server. Anche in questo caso si può avere un default gateway per la rete IPv6 globale. Ricapitolando, un router con questa configurazione, quando arriva un pacchetto, può:

- inoltrarlo direttamente se fa match con una entry nella tabella di instradamento (situazione classica);
- chiedere al tunnel broker server per vedere se si tratta di un indirizzo per cui serve il tunneling;
- mandarlo su un default gateway per la IPv6 globale se la risposta del tunnel broker server è negativa.

Problemi

- È una soluzione centralizzata e per questo il tunnel broker server è un *single point of failure*.
- Complica il piano di controllo.

- Se questo server serve ad interconnettere reti diverse, anche appartenenti a provider diversi, nasce il problema della responsabilità della sua gestione.

Vantaggi

- È più flessibile del 6to4, perchè consente l'uso di tutti gli indirizzi global unicast.

4.3 Portare il supporto a IPv6 ai margini della rete

4.3.1 Soluzioni basate su NAT

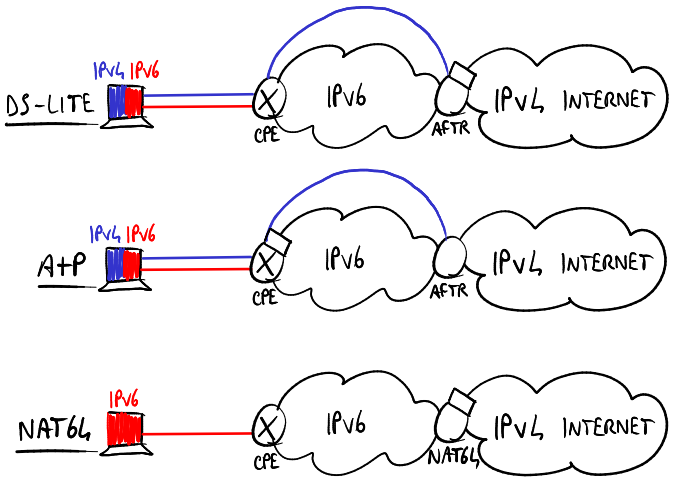


Figura 4.2: Principali soluzioni basate su NAT.

L'obiettivo è migrare grandi reti di provider, in modo che le nuvole IPv4 e/o IPv6 al margine della rete possano usare il backbone IPv6 per interoperare. Lo scenario comune è un utente che vuole connettersi a una destinazione IPv4 attraverso la rete IPv6 del provider.

Tutte le opzioni disponibili fanno uso del NAT. L'utilizzo del NAT è un po' controcorrente dato che IPv6 aveva tra gli obiettivi quello di evitare l'utilizzo dei NAT nelle reti a causa dei numerosi problemi portati dai NAT (modifica dei pacchetti in transito, problemi su reti peer-to-peer, ecc.). Tuttavia il fatto che queste soluzioni sono basate su NAT presenta una serie di vantaggi: i NAT sono largamente diffusi nelle reti, se ne conoscono problemi e limitazioni, si conoscono le applicazioni che possono avere dei problemi nel passare attraverso i NAT; così in generale il vantaggio è la grande esperienza accumulata finora.

Principali componenti

Nelle soluzioni basate su NAT ci sono tre principali componenti:

- **Customer-Premises Equipment** (CPE): è il router all'edge del cliente subito prima della rete del provider;
- **Address Family Transition Router** (AFTR): è un **IPv6 Tunnel Concentrator**, cioè l'apparato alla fine di un tunnel IPv6;
- **NAT44/NAT64**: è un NAT per la traduzione degli indirizzi IPv4/IPv6 in indirizzi IPv4.

Principali soluzioni basate su NAT

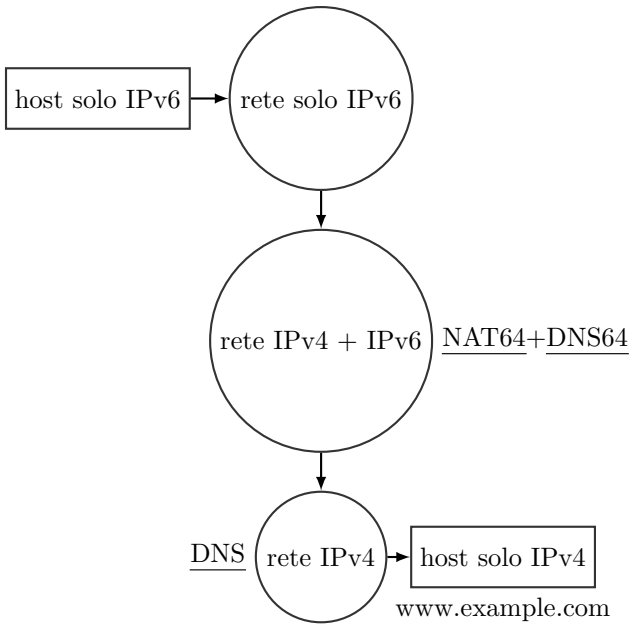
- NAT64 (sezione 4.3.2);
- Dual-Stack Lite (DS-Lite): NAT44 + 4-over-6 tunnel (sezione 4.3.3);
- Dual-Stack Lite Address+Port (DS-Lite A+P): DS-Lite con intervalli di porte preconfigurati (sezione 4.3.4);
- NAT444: CGN + CPE NAT44, ovvero quando un utente domestico, che riceve il servizio dalla compagnia telefonica, inserisce un NAT nella propria rete domestica; ogni pacchetto in uscita dalla rete domestica è sottoposto a due address translation;
- Carrier Grade NAT (CGN): NAT44 su larga scala, ovvero NAT utilizzato dalle compagnie telefoniche per mappare le centinaia di migliaia di indirizzi privati (degli utenti) nei limitati indirizzi pubblici a disposizione.

Per la migrazione di grosse reti orientate a dispositivi mobili si sta scegliendo la soluzione con NAT64.

Per migrare verso IPv6 mantenendo la compatibilità IPv4 ai margini della rete alcuni operatori telefonici stanno pianificando migrazioni massive a DS-Lite poiché è una soluzione abbondantemente sperimentata, ed esistono numerosi dispositivi compatibili già in commercio.

La soluzione A+P non è ancora presa seriamente in considerazione a causa della poca esperienza.

4.3.2 NAT64



1. L'utente solo IPv6 digita `www.example.com` nel browser, ed essendo IPv6 invia una richiesta AAAA al DNS64 del provider. Si supponga che `www.example.com` abbia l'indirizzo IPv4 "20.2.2.2".
2. Il DNS64, in caso non abbia la risoluzione del nome, deve inviare la query ad un DNS superiore, presumibilmente nella rete IPv4.

a. Nel caso migliore DNS64 invia la query AAAA al DNS superiore e ottiene una risposta di tipo AAAA (quindi IPv6), che ritrasmette così com'è all'host (è assolutamente lecito inviare in un pacchetto IPv4 una query DNS che richiede la risoluzione di un nome in un indirizzo IPv6).

b. Nel caso peggiore il DNS superiore non ha supporto IPv4, quindi risponde con un "Name error"; il DNS64 invia nuovamente la query ma questa volta di tipo A, a seguito della quale riceverà una risposta corretta. Questa risposta verrà convertita in AAAA e ritrasmessa all'host. Nella risposta trasmessa all'host, gli ultimi 32 bit sono uguali a quelli inviati dal DNS superiore nel record di tipo A, mentre gli altri 96 bit completano l'indirizzo IPv6; quindi l'indirizzo finale sarà "64:FF9B::20.2.2.2".

3. L'host è ora pronto a instaurare una connessione TCP con `www.example.com`.

4. Entra in gioco il NAT64: converte il pacchetto IPv6 proveniente dall'host in IPv4, e fa l'operazione inversa per i pacchetti provenienti da 20.2.2.2.

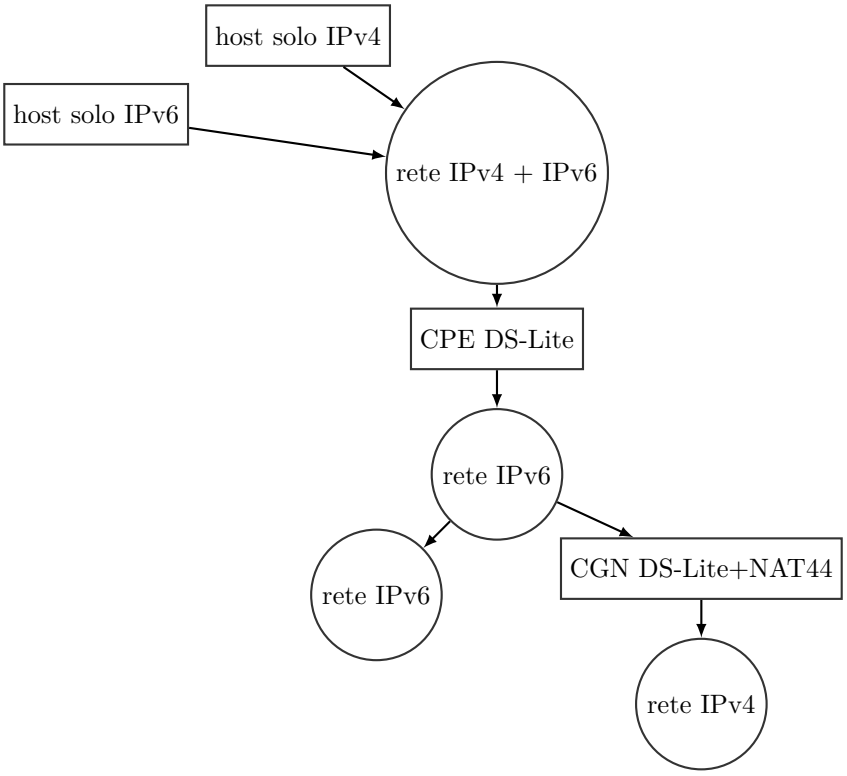
Considerazioni

- In un scenario del genere non c'è tunneling: l'intestazione IPv6 viene solo sostituita con una IPv4 e viceversa.
- L'host solo IPv6 non è consapevole del fatto che l'indirizzo di destinazione è relativo a un indirizzo IPv4.
- Il NAT64 non solo è in grado di tradurre indirizzi IPv6 in indirizzi IPv4, ma in una certa maniera fa credere alla rete che ci sono a disposizione 2^{32} indirizzi IPv6 dato che ogni pacchetto dall'host al NAT64 avrà come indirizzo di destinazione "64:FF9B::20.2.2.2", con il prefisso "64:FF9B/96".
- La rete del provider, quella in cui si trovano il NAT64 e il DNS64, è IPv6 nativa, quindi un host nella rete del provider può contattare direttamente un altro host dotato di supporto IPv6 senza coinvolgere in alcun modo il NAT64.
- "64:FF9B/96" è lo spazio di indirizzamento standardizzato appositamente per questa tecnica di traduzione, assegnato al NAT64, ma l'amministratore di rete potrebbe decidere di cambiarlo a seconda delle necessità. Si noti che l'amministratore di rete deve configurare l'instradamento affinché ogni pacchetto avente tale prefisso vada al NAT64, e deve configurare il NAT64 affinché traduca ogni pacchetto IPv6 avente tale prefisso in IPv4 e lo inoltri nella nuvola IPv4.

Svantaggi

- La presenza del NAT introduce un problema tipico: l'host dietro il NAT non può essere facilmente raggiunto dall'esterno.
- Succede spesso che quando un DNS non ha la risoluzione di un indirizzo non risponde affatto, invece di inviare un "Name error"; il risultato è un allungamento dei tempi a causa dell'attesa del timeout del DNS64, che alla scadenza del timeout invia una query di tipo A.
- Questa soluzione non funziona se l'utente vuole digitare direttamente l'indirizzo IPv4: l'utente deve sempre specificare il nome della destinazione.

4.3.3 DS-Lite



La soluzione **Dual-Stack Lite** (DS-Lite) consiste nella semplificazione dei CPE spostando le funzionalità di NAT e DHCP ai margini della rete del provider, quindi in un apparato che funge da AFTR e da CGN-NAT44.

1. Il server DHCP del provider assegna un indirizzo IPv6, univoco all'interno della rete del provider, ad ogni host di ogni CPE.

2. Quando l'utente deve spedire pacchetti IPv4 è necessaria un'operazione di tunneling, in modo da incapsulare pacchetti IPv4 in pacchetti IPv6 dato che la rete del provider è solo IPv6. Allora, quando un CPE riceve un pacchetto IPv4 lo deve far passare in un tunnel in un pacchetto IPv6 per poterlo spedire all'AFTR oltre il quale c'è la nuvola IPv4; quindi lo scenario è costituito da tante operazioni di tunneling sulla rete IPv6 del provider tra l'AFTR e uno dei numerosi CPE degli utenti. In particolare, il pacchetto tra CPE e AFTR avrà come indirizzo sorgente quello dell'AFTR, e come indirizzo di destinazione quello della destinazione nella rete IPv4.

3. L'AFTR, dopo aver eliminato l'intestazione IPv6, lo invia al NAT44 che sostituisce l'indirizzo sorgente IPv4 (privato) con l'indirizzo IPv4 a cui il NAT riceverà i pacchetti associati a questo flusso.

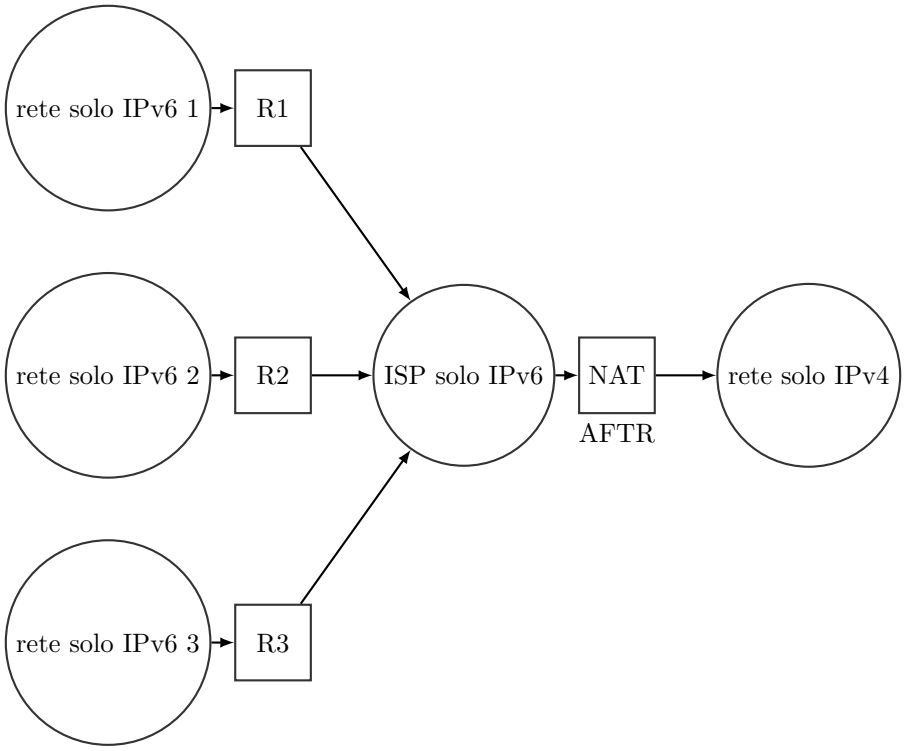
Il vantaggio maggiore di DS-Lite è quello di ridurre notevolmente il numero di indirizzi pubblici del provider.

Possono esserci indirizzi IPv4 duplicati nella rete del provider? No, perché il NAT44 traduce direttamente gli indirizzi IPv4 degli host negli indirizzi IPv4 pubblici disponibili. Se ci fossero indirizzi IPv4 privati duplicati il NAT avrebbe problemi di ambiguità.

Svantaggi

- Un host IPv4 non può contattare una destinazione IPv6 ⇒ le destinazioni IPv6 sono raggiungibili solo da host IPv6. Invece, un host IPv6 può spedire e ricevere pacchetti da nodi IPv6 senza passare attraverso l'AFTR del provider.
- Alcuni tipi di applicazioni non sono in grado di funzionare in una situazione del genere: il fatto che il NAT non sia gestibile dall'utente, dato che non si trova più sul CPE, rende impossibile effettuare operazioni comuni come l'apertura/chiusura delle porte necessarie a specifiche applicazioni.

4.3.4 DS-Lite A+P



La soluzione **Dual-Stack Lite Address+Port** (DS-Lite A+P) consiste nell'avere ancora una rete del provider solo IPv6, ma il NAT viene spostato sul CPE in modo che l'utente possa configurarlo in base alle proprie esigenze.

Come in DS-Lite, un pacchetto IPv4 in uscita dal CPE viene ancora fatto passare in un tunnel poiché la rete del provider è solo IPv6.

Il fatto che il NAT su ogni CPE richieda un indirizzo IPv4 pubblico viene risolto permettendo la duplicazione di indirizzi IPv4 pubblici, e i CPE vengono distinti in base alla porta. Infatti ogni CPE utilizza un determinato intervallo di porte, e l'AFTR, che conosce l'intervallo di porte utilizzato da ogni CPE, riesce a distinguere i flussi da e verso uno specifico CPE nonostante ci siano diversi CPE aventi lo stesso indirizzo IPv4 pubblico.

Questa soluzione è simile a quella con DS-Lite, ma lo spazio di indirizzi IPv4 privati è più sotto il controllo dell'utente finale, perché dato che il NAT è sul CPE dell'utente l'utente può configurarlo, anche se con alcune limitazioni: non può aprire e usare porte che non sono nel proprio intervallo. Questo metodo permette di risparmiare indirizzi IPv4 (ma comunque meno rispetto a DS-Lite).

Questa soluzione in Italia è sostanzialmente illegale perché, siccome il numero di porta non viene memorizzato, in caso di attacco non si riuscirebbe a risalire all'attaccante.

4.3.5 MAP

L'introduzione di questa soluzione deriva dalla considerazione che in entrambe le soluzioni basate su NAT viste finora c'è uno stato che deve essere mantenuto per ogni connessione, e che gestire quindi lo stato di molte connessioni è oneroso per l'AFTR.

MAP-E

Nella soluzione **MAP-E** si vuole quindi ottenere lo stesso risultato delle soluzioni precedenti in maniera *stateless*, codificando le informazioni che prima venivano memorizzate nelle tabelle dell'AFTR direttamente nell'indirizzo IPv6 sorgente usato dal CPE. Ad ogni CPE viene assegnato un **Port Set IDentifier (PSID)** univoco e un indirizzo IPv4 pubblico, perchè si vuol fare NAT sul CPE come nella soluzione A+P. L'indirizzo destinazione del tunnel è configurato staticamente sul CPE.

Il Border Relay non è comunque totalmente stateless, ma deve conservare l'informazione sul legame tra l'indirizzo IPv4 usato dal CPE e il suo PSID, in modo da poter ricostruire al contrario l'indirizzo IPv6 del CPE. Questa soluzione è più conveniente del NAT tradizionale perchè sul Border Relay ci sarà una sola entry per una intera rete IPv4, contenente poche informazioni su EA bit length, PSID offset, IPv4 e IPv6 prefix.

MAP-T

In quanto visto finora l'indirizzo IPv4 destinazione è contenuto nel pacchetto IPv4 incapsulato nel tunnel. Nella soluzione **MAP-T** si vuole invece evitare di incapsulare i pacchetti IPv4 dentro IPv6 tramite un meccanismo di traduzione degli header. Si riprende l'idea del MAP-E di incorporare nell'indirizzo IPv6 informazioni aggiuntive, per cui l'indirizzo IPv6 destinazione del pacchetto viene costruito nel seguente modo: nell'interface identifier i primi 8 bit alti vengono posti a zero, poi si accodano i 32 bit dell'indirizzo IPv4 destinazione e infine i restanti 24 bit sono lasciati a zero. Questi indirizzi vengono chiamati **IPv4-embedded IPv6**: ad esempio per una destinazione IPv4 `5.5.5.5` si usa l'indirizzo IPv6 `2001:1::5:505:500:0`. Questo non è certamente l'indirizzo IPv6 del Border Relay, ma viene usato nel seguente modo:

- il Border Relay annuncia sulla propria rete, tramite i protocolli di routing, che il prefisso `2001:1::/64` è raggiungibile attraverso di lui;
- questi indirizzi sono però fittizi e speciali, quindi è possibile distinguerli dai normali IPv6 che devono essere inoltrati secondo il normale routing;
- quando il Border Relay riconosce un IPv4-embedded IPv6 sa che deve fare traduzione, quindi toglie l'header IPv6, costruisce l'IPv4 sorgente dall'IPv6 sorgente e l'IPv4 destinazione dall'IPv6 destinazione. L'informazione sulla porta di livello 4 non cambia ed è comunque contenuta nell'IPv6 sorgente.

Questa soluzione è molto usata perchè è scalabile e richiede di memorizzare poche informazioni.

4.4 Trasportare il traffico IPv6 nella rete centrale

L'obiettivo principale è avere nella rete globale traffico IPv6 senza stravolgere la rete esistente da più di 20 anni che attualmente funziona bene. Non sarebbe possibile sostenere costi di migrazione umani e tecnologici a livello mondiale per cambiare radicalmente la rete IPv4 esistente per renderla IPv6.

4.4.1 6PE

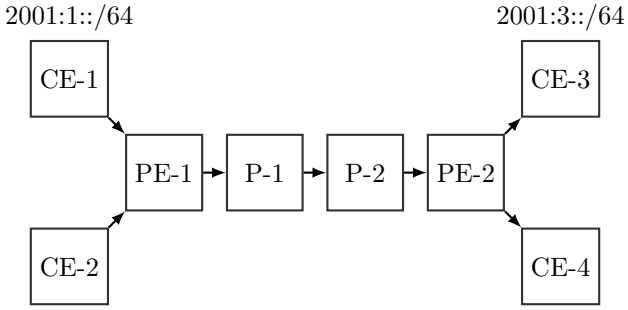
L'obiettivo della soluzione **6 Provider Edge** (6PE) è connettere delle nuvole IPv6 tra loro attraverso un backbone MPLS. 6PE richiede che la rete dell'operatore funzioni con MPLS. In

questo scenario il margine del provider è rappresentato dai primi router che incontrano i CPE degli utenti.

Idea

- Mantenere invariato il core della rete (senza escludere la possibilità di future modifiche).
- Aggiungere il supporto a IPv6 al margine della rete del provider.
- Distribuire informazioni di routing IPv6 in MPLS/BGP, come nelle VPN (vedere la sezione 5.4.2).

Requisiti



Il requisito fondamentale è avere una rete centrale MPLS.

Nella figura:

- la rete centrale MPLS è quella costituita da PE-1, P-1, P-2, PE-2;
- i due apparati laterali, PE-1 e PE-2, sono parzialmente immersi nella rete MPLS;
- si può pensare ai canali tra CE e PE come dei canali che forniscono collegamento ADSL all'utente domestico.

6PE è pensato per prendere una rete centrale pienamente funzionante, in grado di trasportare pacchetti IPv4 con MPLS, e aggiungere il supporto IPv6 solo agli edge router del provider (PE). Infatti, una volta che un pacchetto viene incapsulato in un pacchetto MPLS, gli apparati intermedi non si interesseranno più al tipo di pacchetto contenuto, ma saranno solo interessati alla etichetta che permette loro di distinguere l'LSP nel quale instradarlo.

Infatti, sui PE è richiesto un ulteriore aggiornamento al fine di aggiungere il supporto **MG-BGP**, protocollo che permette di trasportare e comunicare sia rotte IPv4 sia rotte IPv6.

Quindi il grosso vantaggio di questa soluzione consiste nel richiedere l'aggiornamento solo dei PE e non di tutti i router intermedi: in fin dei conti, un'operazione che il provider può gestire senza elevati costi.

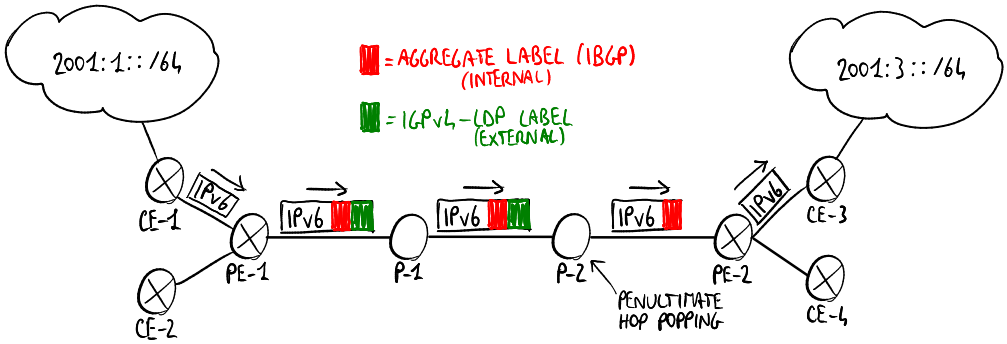
Come vengono annunciate le reti IPv6

1. CE-3 annuncia che può raggiungere la rete IPv6 "2001:3::/64".
2. Questa informazione viene ricevuta anche da PE-2.
3. PE-2 invia questa informazione a tutti i PE nella rete, dicendo che può raggiungere "2001:3::/64" attraverso il next hop "FFFF:20.2.2.2", nonostante la sua interfaccia sia IPv4 (questo perché se è data una rotta IPv6 deve essere dato un next hop IPv6).
4. PE-1 riceve questa informazione.

5. PE-1 invia l'informazione ricevuta a tutti i router ad esso collegati, quindi anche ai CE domestici, dicendo che può raggiungere la rete "2001:3::/64".

6. Se non esiste ancora un percorso MPLS tra PE-1 e l'indirizzo "20.2.2.2", vengono usati i classici meccanismi MPLS (quindi il protocollo di segnalazione LDPv4) per instaurare questo percorso.

Come viene instradato il traffico IPv6



Per instradare un pacchetto IPv6 vengono utilizzate due etichette:

etichetta esterna LDP/IGPv4 verso PE-2	etichetta interna MP-BGP verso CE-3	pacchetto IPv6 verso la destinazione IPv6

- etichetta MP-BGP (interna): identifica il CE di destinazione a cui il PE di destinazione deve inviare il pacchetto;
- etichetta LDP/IGPv4 (esterna): identifica l'LSP tra i due PE nella rete MPLS.

Si supponga che un host nella rete "2001:1::/64" voglia inviare un pacchetto a un host nella rete "2001:3::/64":

1. il pacchetto arriva a CE-1;
2. CE-1 sa che la rete "2001:3::1/64" esiste e invia il pacchetto verso PE-1;
3. PE-1 mette due etichette davanti al pacchetto: l'etichetta interna e l'etichetta esterna;
4. PE-1 invia il pacchetto a P-1, che lo invia a P-2;
5. P-2, che è il penultimo hop, rimuove l'etichetta esterna dal pacchetto (**penultimate label popping**) e lo invia a PE-2;
6. PE-2 rimuove l'etichetta interna e invia il pacchetto a CE-3;
7. CE-3 inoltra il pacchetto alla destinazione nella rete "2001:3::/64".

Considerazioni

- I router PE devono essere **dual stack** e devono supportare MP-BGP, mentre i router intermedi non hanno bisogno di alcuna modifica.
- Questa soluzione fornisce un servizio IPv6 nativo ai clienti senza cambiare la rete centrale MPLS IPv4 (richiede costi e rischi operativi minimi).
- Questa soluzione scala fino a quando ci sono poche nuvole IPv6 da distribuire.

4.5 Problematiche di sicurezza

Si ha poca esperienza con i problemi di sicurezza perché IPv6 non è ancora molto usato ⇒ IPv6 potrebbe ancora avere delle falle di sicurezza non scoperte che potrebbero essere sfruttate da malintenzionati. Inoltre, durante la fase di migrazione gli host hanno bisogno di aprire due porte in parallelo, una per IPv4 e un'altra per IPv6 ⇒ devono essere protette due porte da attacchi dall'esterno.

Attacchi DDoS con SYN flooding L'interfaccia di un host può avere più indirizzi IPv6 ⇒ può generare più richieste SYN TCP, ognuna con un diverso indirizzo sorgente, a un server al fine di saturarne la memoria facendo aprire ad esso diverse connessioni TCP non chiuse.

Messaggi Router Advertisement falsi Un host potrebbe iniziare a mandare dei messaggi "Router Advertisement" annunciando dei prefissi falsi ⇒ gli altri host nel link inizieranno a inviare pacchetti con dei prefissi sbagliati negli indirizzi sorgente.

Capitolo 5

VPN

Una società che vuole creare una rete privata aziendale per i suoi terminali remoti (utenti singoli, data-center, filiali) può adottare due diversi approcci:

- la società può costruire la propria infrastruttura dedicata (linea dedicata, connessione ad accesso remoto);
- la società può adottare una soluzione VPN.

Una **Virtual Private Network** (VPN) permette a una società di distribuire la connettività a più utenti su una infrastruttura condivisa pubblica (Internet o la rete di un Internet Service Provider) forzando le proprie politiche (come la sicurezza, la qualità del servizio, l'indirizzamento privato) come se fosse la propria rete privata.

I vantaggi di una soluzione VPN sono:

- economicità: non è più necessario costruire costose connessioni fisiche, ma la soluzione VPN sfrutta un'infrastruttura pre-esistente in modo che il costo sia condiviso;
- selettività: grazie a un firewall possono accedere solo gli utenti che hanno i diritti ⇒ maggiore sicurezza;
- flessibilità: gli utenti ammessi possono essere facilmente aggiunti, e gli utenti possono facilmente spostarsi.

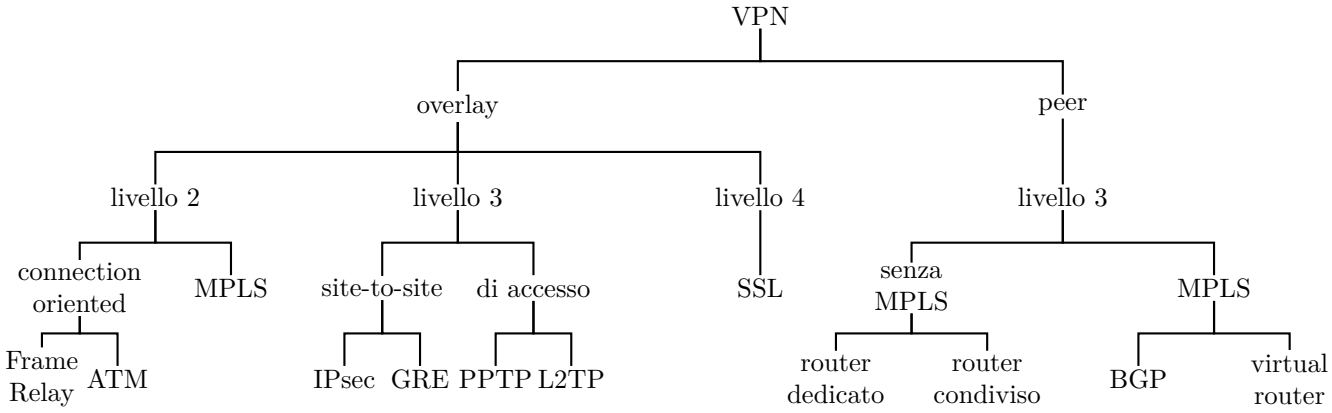
Una VPN è costituita da alcuni componenti principali:

- i dati vengono consegnati attraverso dei **tunnel**, così essi sono separati da altri dati in giro per la infrastruttura condivisa, usando protocolli come GRE, L2TP, MPLS, PPTP e IPsec;
- i dati vengono **crittografati** per una maggiore sicurezza, usando protocolli come IPsec e MPPE;
- viene verificata l'**integrità** dei dati, così essi non possono essere manomessi, grazie al checksum TCP o ad AH in IPsec;
- prima di instaurare un tunnel, viene verificata l'identità di chi sta dall'altra parte del tunnel attraverso dei meccanismi di **autenticazione** (per esempio usando i certificati digitali).

5.1 Classificazione

Le soluzioni VPN si possono classificare in base a:

- distribuzione: di accesso o site-to-site (intranet/extranet) (sezione 5.1.1);
- accesso a Internet: centralizzato o distribuito (sezione 5.1.2);



- modello: overlay o peer (sezione 5.1.3);
- provision: customer o provider (sezione 5.1.4);
- livello: livello 2, livello 3 o livello 4 (sezione 5.1.5);
- topologia virtuale: hub and spoke o mesh (sezione 5.1.6).

	overlay	**peer**
di accesso	L2TP, PPTP	
site-to-site	IPsec, GRE	MPLS

5.1.1 Scenari di distribuzione

Le VPN possono essere distribuite in due scenari principali:

- **VPN di accesso** (anche chiamata "VPN remota" o "dial in virtuale"): virtualizza la connessione ad accesso remoto e connette un singolo utente a una rete aziendale attraverso ISDN, PSTN, modem via cavo, LAN senza fili usando protocolli come PPTP e L2TP (vedere la sezione 5.3);
- **VPN site-to-site**: virtualizza la linea dedicata e connette più reti remote tra loro usando protocolli come IPsec, GRE e MPLS (vedere la sezione 5.4).
  Le VPN site-to-site possono essere distribuite in due modi:
  - **VPN intranet**: tutte le reti interconnesse appartengono alla stessa società;
  - **VPN extranet**: le reti interconnesse appartengono a più società.

Le funzionalità di una VPN devono rispettare alcuni requisiti:

- sicurezza: un firewall può limitare l'accesso alle risorse della rete;
- crittografia dei dati: per proteggere le informazioni trasmesse su un'infrastruttura condivisa;
- affidabilità: si deve garantire che il traffico mission critical arrivi a destinazione;
- scalabilità: la soluzione deve funzionare per reti sia piccole sia grandi;
- distribuzione incrementale: la soluzione continua a funzionare mentre la rete cresce;
- requisiti aggiuntivi per le VPN di accesso:
  - autenticazione forte: per verificare le identità degli utenti mobili (ad es. un computer portatile personale potrebbe essere rubato);

  - gestione centralizzata degli utenti e dei loro account;

- requisiti aggiuntivi per le VPN extranet site-to-site:
  - accesso selezionato: ogni società può dare ad altre società l'accesso ad alcuni servizi ma impedire l'accesso ad altri servizi della sua rete privata;
  - gestione delle collisioni di indirizzi: un indirizzo privato può appartenere a due diverse reti private ⇒ è necessario un NAT;
  - utilizzo di una soluzione aperta basata su uno standard: società diverse devono poter condividere la stessa soluzione VPN;
  - controllo del traffico: ogni società ha bisogno di controllare la quantità di traffico in entrata proveniente dalle reti di altre società e di eliminare i colli di bottiglia ai punti di accesso della rete.

5.1.2 Accesso a Internet

Un terminale remoto connesso a una VPN può accedere all'Internet pubblico in due modi:

- **accesso a Internet centralizzato**: tutto il traffico da e verso Internet passa sempre attraverso il gateway VPN della sede centrale;
- **accesso a Internet distribuito**: il traffico da e verso Internet non coinvolge la VPN, che è distribuita solo per il traffico aziendale.

Accesso a Internet centralizzato

Vantaggi

- gestione delle politiche centralizzata: una società può imporre le proprie politiche per l'accesso ad Internet (ad es. proibendo ai dipendenti di accedere a certi siti Web mentre stanno lavorando) una volta per tutti i terminali remoti connessi;
- vantaggio di sicurezza: il firewall aziendale può proteggere gli host dai pacchetti malevoli provenienti da Internet.

Svantaggi

- velocità minore ai terminali remoti: i pacchetti da e verso Internet devono viaggiare per un maggior numero di hop perché devono sempre passare attraverso il gateway VPN della sede centrale invece di dirigersi direttamente verso la destinazione;
- maggiore larghezza di banda richiesta alla sede centrale;
- connessione obbligata: la VPN deve essere sempre attiva, cioè l'utente non deve poter disabilitare temporaneamente la VPN e navigare su Internet senza la protezione del firewall aziendale, altrimenti se viene infettato quando ritorna alla VPN infetterà la rete aziendale perché il traffico che proviene da lui non è protetto dal firewall.

Accesso a Internet distribuito

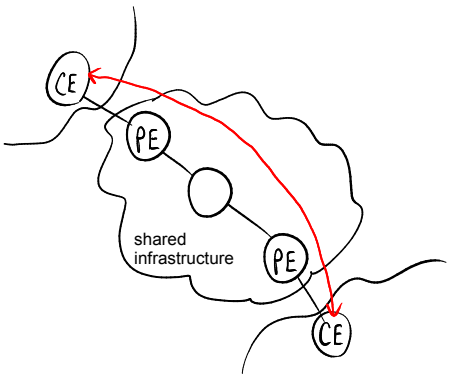
Vantaggi

- velocità maggiore ai terminali remoti: i pacchetti si dirigono sempre verso la destinazione su Internet;
- connessione volontaria: l'utente può disabilitare la VPN in qualsiasi momento senza rischi aggiuntivi per la sicurezza.

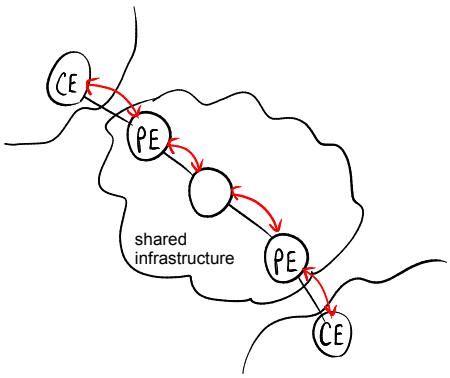
Svantaggi

- gestione delle politiche distribuita: la società deve configurare più router ai terminali remoti per imporre le proprie politiche;
- minaccia per la sicurezza: manca la protezione del firewall aziendale.

5.1.3 Modelli



(a) Modello overlay.

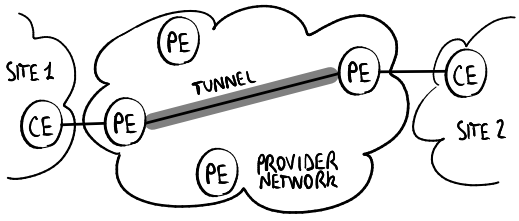


(b) Modello peer.

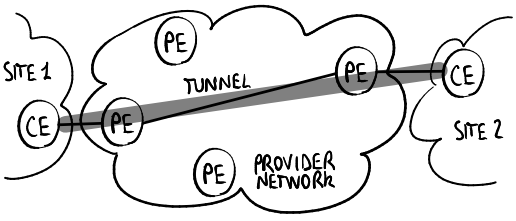
Le VPN si possono suddividere in due modelli a seconda del ruolo dell'infrastruttura condivisa:

- **modello overlay**: l'infrastruttura non è consapevole della soluzione VPN e offre solo il servizio di connettività IP: trasporta solo i pacchetti, senza sapere che sono pacchetti VPN, tra i gateway VPN che non interagiscono con l'infrastruttura ⇒ adatto per la privacy dei dati: il gestore dell'infrastruttura non può guardare dati privati né può approfittare di informazioni sull'instradamento;
- **modello peer**: i gateway all'interno dell'infrastruttura partecipano alla creazione della VPN e interagiscono tra loro ⇒ instradamento migliore, più scalabilità.
  Le VPN con modello peer non basate su MPLS possono essere:
  - **a router dedicato**: il gestore dell'infrastruttura dedica alcuni router completamente a un cliente, altri completamente a un altro cliente, e così via;
  - **a router condiviso**: ogni router nell'infrastruttura è condiviso tra più clienti ⇒ costo minore.

5.1.4 Provision



(a) Provider provision.



(b) Customer provision.

Le VPN possono essere customer o provider provisioned:

- **customer provision**: l'utente crea e gestisce la VPN da solo, e i tunnel vengono instaurati tra Customer Edge (CE);

- **provider provision**: la VPN è fornita e gestita interamente dal fornitore della connettività Internet, e i tunnel vengono instaurati tra Provider Edge (PE).

Le VPN customer provisioned non possono essere con modello peer perché il fornitore non può essere consapevole di una VPN che il cliente ha creato da sé.

5.1.5 Livelli

La connettività VPN può essere a livelli diversi:

- livello 2: nella VPN vengono scambiate trame Ethernet:
  - **Virtual Private LAN Service** (VPLS): virtualizza una LAN: i terminali sono connessi come se fossero nella stessa LAN ⇒ i pacchetti broadcast sono consentiti;
  - **Virtual Private Wire Service** (VPWS): virtualizza una linea dedicata (su una rete a commutazione di pacchetto);
  - **IP-only LAN-like Service** (IPLS): virtualizza una rete IP, ma sono consentiti solo i pacchetti IP (ICMP e ARP);
- livello 3: nella VPN vengono scambiati pacchetti IP;
- livello 4: nella VPN vengono instaurate connessioni TCP (eventualmente con SSL per la sicurezza).

5.1.6 Topologie virtuali

Le VPN si possono suddividere in due categorie in base alla topologia virtuale:

- **hub and spoke**: ogni coppia di terminali può comunicare l'uno con l'altro solo passando per la sede centrale ⇒ può avvenire un collo di bottiglia alla sede centrale a causa del traffico più alto;
- **mesh**: ogni coppia di terminali può comunicare l'uno con l'altro direttamente senza passare per la sede centrale ⇒ l'instradamento è migliore, ma il numero di tunnel è più alto.

5.2 Protocolli

5.2.1 PPP

Point-to-Point Protocol (PPP) è un protocollo di livello 2 usato nelle connessioni punto punto (ad accesso remoto, ISDN) per incapsulare qualsiasi protocollo di livello superiore. È un'estensione di HDLC.

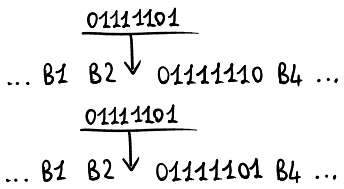
Un pacchetto PPP ha il seguente formato:

1 byte	1 byte	1 byte	1 o 2 byte		2 o 4 byte	1 byte
01111110 (flag)	Address	Control	Protocol	Data	CRC	01111110 (flag)

dove i campi più significativi sono:

- flag iniziale e finale: sono necessari per la delimitazione dei dati;
- campi Address e Control: sono ereditati da HDLC ma sono completamente privi di significato nelle connessioni punto punto; sono stati mantenuti per rendere semplice l'aggiornamento degli algoritmi di elaborazione negli apparati HDLC;
- campo Protocol: specifica il protocollo di livello superiore del pacchetto incapsulato.

"01111110" è la sequenza che delimita la trama, ma per inviare quella sequenza come dati sono necessarie delle regole di stuffing. In PPP lo stuffing è al livello dei byte: la sequenza di escape "01111101" è inserita sia quando nei dati compare un byte uguale al byte di delimitazione, sia quando compare un byte uguale al byte di escape stesso:



Link Control Protocol (LCP) ha il compito di aprire e chiudere le connessioni PPP, negoziando alcune opzioni (come la massima lunghezza delle trame, il protocollo di autenticazione).

5.2.2 GRE

Un pacchetto IP non può incapsulare direttamente un protocollo di livello 3 o inferiore, perché il campo "Protocol" nell'intestazione IPv4 può specificare solo protocolli di livello superiore (come TCP, UDP, ICMP).

Generic Routing Encapsulation (GRE) è un protocollo per incapsulare qualsiasi protocollo (compresi IP e altri protocolli di livello inferiore) in IP: l'intestazione GRE è inserita tra l'intestazione IP incapsulante e il pacchetto incapsulato, e il campo "Protocol" nell'intestazione IP incapsulante è impostato a 47.

L'intestazione GRE ha il formato seguente:

<table>
<tr><th colspan="5"></th><th>5</th><th colspan="2">8</th><th colspan="2">13</th><th colspan="2">16</th><th>32</th></tr>
<tr><td>C</td><td>R</td><td>K</td><td>S</td><td>s</td><td colspan="2">Recur</td><td colspan="2">Flags</td><td colspan="2">Version (0)</td><td>Protocol type</td></tr>
<tr><td colspan="11">Checksum (facoltativo)</td><td>Offset (facoltativo)</td></tr>
<tr><td colspan="12">Key (facoltativo)</td></tr>
<tr><td colspan="12">Sequence number (facoltativo)</td></tr>
<tr><td colspan="12">Routing (facoltativo) :::</td></tr>
</table>

dove i campi più significativi sono:

- flag C, R, K, S (1 bit ciascuno): indicano la presenza/assenza dei campi facoltativi;
- flag strict source routing (s) (1 bit): se impostato a 1, il pacchetto viene scartato se, quando la lista di source routing (campo "Routing") finisce, la destinazione non è ancora stata raggiunta (simile a TTL);
- campo Recur (3 bit): specifica il massimo numero di incapsulamenti aggiuntivi consentito (attualmente non supportato);
- campo Version (3 bit): specifica la versione del protocollo GRE (0 in questo caso);
- campo Protocol type (16 bit): specifica il protocollo del pacchetto incapsulato;
- campo Routing: specifica una sequenza di indirizzi IP corrispondenti ai router intermedi attraverso cui deve passare il pacchetto (**source routing**).
  Il campo "Routing" è a sua volta costituito da alcuni campi (oltre alla lista degli indirizzi IP), tra cui:
  - campo Address family: specifica il tipo degli indirizzi nella lista (IP in questo caso);
  - campo SRE Offset: è un puntatore all'elemento della lista contenente l'indirizzo IP del next hop corrente, aggiornato ad ogni hop di source routing;
  - campo SRE Length: specifica la lunghezza della lista (in byte).

Enhanced GRE

Il formato dell'intestazione **Enhanced GRE** introduce il nuovo campo "Acknowledgment number":

				5		8		13		16	32
C	R	K	S	s	Recur		A	Flags	Version (1)		Protocol type
Key (lunghezza payload)											Key (call ID)
Sequence number (facoltativo)											
Acknowledgment number (facoltativo)											

dove i campi più significativi sono:

- campo Key (32 bit):
  - lunghezza payload (16 bit): specifica la lunghezza del pacchetto esclusa l'intestazione GRE (in byte);
  - call ID (16 bit): specifica l'ID di sessione del pacchetto;
- campo Acknowledgment number (32 bit): il mittente mette nel pacchetto il numero di sequenza dell'ultimo pacchetto che ha ricevuto dalla destinazione (ACK cumulativo).
  Il nuovo campo "Acknowledgment number", in combinazione con il campo "Sequence number", consente alcuni meccanismi aggiuntivi:
  - **controllo di flusso**: le finestre scorrevoli evitano di inondare la destinazione, e si spostano quando vengono ricevuti i pacchetti ACK;
  - **rilevamento dei pacchetti out-of-order**: Enhanced GRE è stato progettato per l'incapsulamento di PPP, e PPP è un protocollo per connessioni punto punto dove non è previsto che i pacchetti arrivino out-of-order ⇒ i pacchetti out-of-order vengono sempre scartati;
  - **rilevamento dei pacchetti persi**: quando un timeout scade, cioè nessun ACK è stato ricevuto, il pacchetto viene rilevato come perso, ma i pacchetti persi rilevati non vengono ritrasmessi;
  - **controllo di congestione**: quando vengono rilevati troppi timeout, la trasmissione dei pacchetti viene rallentata per non congestionare la rete.

5.2.3 L2TP

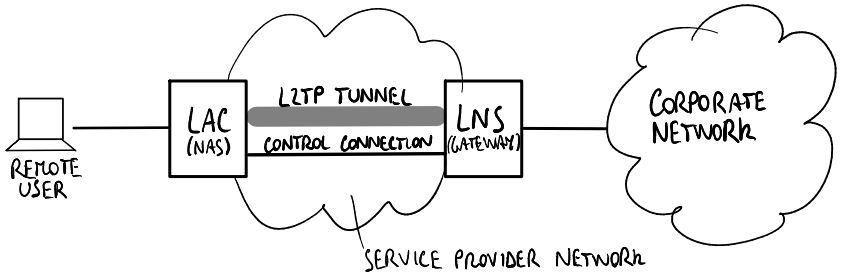


Figura 5.3: Scenario di riferimento originale per L2TP: modo di distribuzione provider provisioned.

Layer 2 Tunneling Protocol (L2TP) è un protocollo per far passare in un tunnel IP qualsiasi protocollo di livello 2 (PPP in questa trattazione). L2TP fu originariamente progettato per VPN di accesso provider provisioned e fu standardizzato da IETF; in seguito fu esteso alle VPN

di accesso customer provisioned semplicemente implementando le funzionalità del LAC nella macchina dell'utente.

Nello scenario di riferimento originale per L2TP, un utente remoto vuole inviare un pacchetto attraverso una connessione punto punto (PPP) ad un server interno all'interno della rete aziendale. Sono necessari un tunnel L2TP verso l'L2TP Network Server (LNS) di destinazione e una sessione L2TP all'interno del tunnel per emulare la connessione PPP sulla rete del service provider.

Quando la trama PPP arriva all'L2TP Access Concentrator (LAC), se non è ancora stato stabilito un tunnel L2TP verso l'LNS, prima di aprire una sessione L2TP il LAC deve stabilire un tunnel verso l'LNS: l'LNS si autentica al LAC usando un meccanismo basato su challenge simile al **Challenge Handshake Authentication Protocol** (CHAP), e viene creata una nuova connessione di controllo.

Ogni sessione L2TP usa due canali all'interno del tunnel:

- **canale dati**: per trasportare i messaggi di dato, cioè le trame PPP;
- **canale di controllo**: per scambiare i messaggi di controllo, che servono per gestire la comunicazione (come verificare che i pacchetti arrivino, ritrasmettere i pacchetti persi, controllare il giusto ordine dei pacchetti).
  Al contrario dei messaggi di dato, i messaggi di controllo vengono scambiati in modo affidabile: i messaggi di controllo persi vengono sempre ritrasmessi.

Possono coesistere più sessioni dentro lo stesso tunnel, che condividono la stessa connessione di controllo, per distinguere più flussi di trame PPP di più utenti remoti: ogni tunnel è identificato da un tunnel ID, ogni sessione è identificata da un session ID.

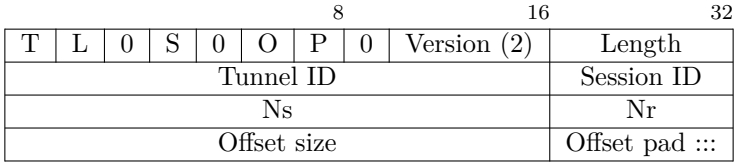
Sicurezza Oltre all'autenticazione assicurata durante la fase di stabilimento del tunnel, L2TP non fornisce di per sé alcun meccanismo di sicurezza: infatti non ha senso usare meccanismi come la crittografia per i pacchetti L2TP che viaggiano lungo il tunnel, perché il LAC del service provider può comunque guardare le trame di livello 2 $\Rightarrow$ qualsiasi meccanismo di sicurezza va implementato end-to-end, cioè tra l'utente e la destinazione finale nella rete aziendale.
Opzionalmente è possibile usare IPsec attraverso il tunnel: esso fornisce una sicurezza forte ma è complicato da implementare.

Un pacchetto all'interno di un tunnel L2TP include diverse intestazioni incapsulate:

intestazione MAC	intestazione IP	intestazione UDP	intestazione L2TP	intestazione PPP	payload PPP

Intestazione PPP Identifica la connessione punto punto tra l'utente remoto e il server interno nella rete aziendale.

Intestazione L2TP Identifica il tunnel L2TP:



dove i campi più significativi sono:

- flag T (1 bit): se impostato a 0 il pacchetto è un messaggio di dato, se impostato a 1 è un messaggio di controllo;
- campo Tunnel ID (16 bit): identifica il tunnel L2TP;

- campo Session ID (16 bit): identifica la sessione L2TP, cioè il canale dati nel tunnel;
- campo Ns (16 bit): contiene il numero di sequenza del messaggio di dato/controllo;
- campo Nr (16 bit): contiene il numero di sequenza del prossimo messaggio di controllo da ricevere per una connessione di controllo affidabile.

Intestazione UDP Perché viene usato un protocollo di livello 4 come UDP per spostare delle trame di livello 2? Ciò si può spiegare tenendo conto dei firewall in una rete generale: se un pacchetto non contiene un incapsulamento di livello 4, è più facile che venga scartata dai firewall. Un'altra possibile motivazione è che al livello 4 si può accedere tramite i socket, mentre il sistema operativo è responsabile del livello 3. Siccome L2TP vuole essere una soluzione contro PPTP (proposto dai fornitori di sistemi operativi), i progettisti di L2TP hanno voluto renderlo accessibile alle applicazioni e non essere legati al volere dei fornitori di sistemi operativi.
È anche possibile usare un protocollo di livello 4 diverso (come ATM, Frame Relay).

Intestazione IP Contiene gli indirizzi IP delle estremità del tunnel.
Nello scenario di riferimento originale, gli indirizzi IP corrispondono al LAC e all'LNS.

5.2.4 PPTP

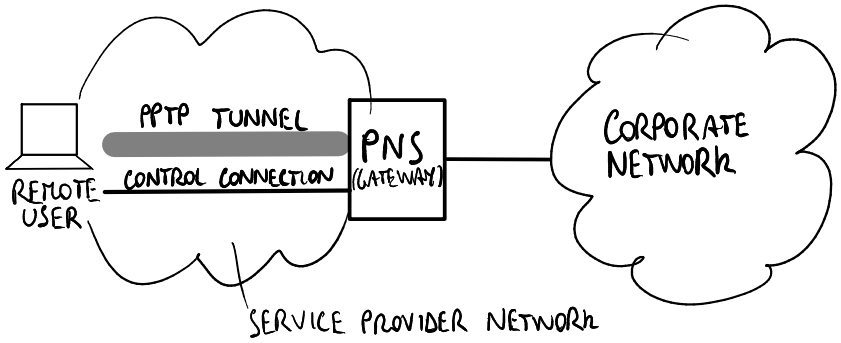


Figura 5.4: Scenario di riferimento originale per PPTP: modo di distribuzione customer provisioned.

Point-to-Point Tunneling Protocol (PPTP) è un protocollo per far passare in un tunnel IP il protocollo PPP. PPTP fu originariamente sviluppato per le VPN di accesso customer provisioned e fu sviluppato dai più importanti fornitori di sistemi operativi; in seguito fu esteso alle VPN di accesso provider provisioned con l'introduzione di un PPTP Access Concentrator (PAC) con funzionalità simili a quelle del LAC.

Lo scenario di riferimento originale per PPTP si differenzia da quello per L2TP per il fatto che il tunnel PPTP viene stabilito tra l'utente remoto stesso e il PPTP Network Server (PNS).

A differenza di L2TP, PPTP fornisce dei deboli meccanismi di sicurezza (facoltativi): lo standard copre l'uso di specifici protocolli di sicurezza proprietari Microsoft come MPPE per la crittografia e MS CHAP per l'autenticazione, così non c'è alcuna negoziazione di protocolli.

Canale dati

Per il canale dati passano le trame PPP incapsulate:

intestazione IP	intestazione GRE	intestazione PPP	payload PPP

PPTP utilizza il protocollo Enhanced GRE per incapsulare le trame PPP in IP. Il payload PPP può essere opzionalmente criptato con MPPE.

Canale di controllo

Per il canale di controllo passano i messaggi di controllo PPTP:

intestazione IP	intestazione TCP	messaggio di controllo PPTP

I messaggi di controllo sono necessari per l'instaurazione, la gestione e l'abbattimento della sessione dati del tunnel, e vengono inviati alla well-known port TCP 1723 del PNS.

5.2.5 IPsec

Le funzionalità della suite di protocolli **Internet Protocol Security** (IPsec) in IPv4 sono molto simili a quelle in IPv6, perciò si rimanda alla sezione 3.4.4 per i dettagli.

La differenza principale è che in IPv6 IPsec è un extension header incluso nello standard, mentre in IPv4 è un nuovo protocollo aggiuntivo incapsulato in IP (per AH il campo "Protocol" è impostato a 51, per ESP è impostato a 50):

Tabella 5.1: Authentication Header (AH)

intestazione IPv4	intestazione AH	intestazione TCP/UDP	payload
dati autenticati			

Tabella 5.2: Encapsulating Security Payload (ESP)

intestazione IPv4	intestazione ESP (per la crittografia)	intestazione TCP/UDP	payload	autenticazione ESP
		dati criptati		
	dati autenticati			

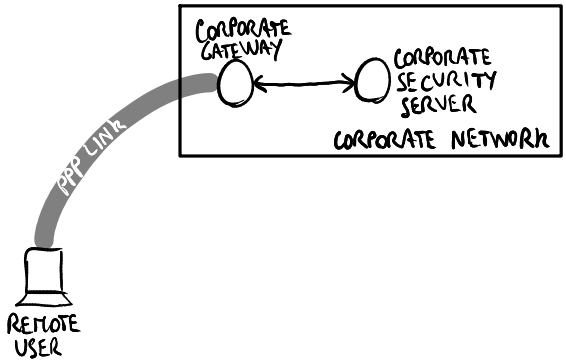
5.2.6 SSL

Secure Sockets Layer (SSL), oggi chiamato **Transport Layer Security** (TLS), è un protocollo crittografico che è progettato per fornire la sicurezza di comunicazione tra un client e un server:

1. il client apre una connessione TCP sulla porta 443 con il server, e invia un challenge per l'autenticazione del server;
2. il server invia al client un messaggio Hello contenente il suo certificato e la risposta al challenge criptato con la chiave privata del server;
3. il client verifica il certificato guardando in una lista di certificati data da una autorità di certificazione fidata, quindi decripta la risposta al challenge utilizzando la chiave pubblica del server;
4. il client decide una chiave privata per la comunicazione sicura e la invia al server criptandola tramite la chiave pubblica del server ⇒ da questo punto in avanti tutti i messaggi di record verranno criptati utilizzando questa chiave privata (che va periodicamente rinegoziata);
5. spesso il server chiede all'utente di digitare nome utente e password su una pagina Web per l'autenticazione del client (a livello applicazione).

5.3 VPN di accesso

5.3.1 Scenario con connessione ad accesso remoto



Fondamentalmente un utente remoto, tipicamente un dipendente di una società, vuole contattare un server passando per la rete aziendale. Può stabilire una connessione punto punto ad accesso remoto con il gateway aziendale che usa PPP per incapsulare i pacchetti IP:

- tramite **Link Control Protocol** (LCP) può negoziare i parametri di livello 2 (come il protocollo di autenticazione);
- tramite **Network Control Protocol** (NCP) può negoziare i parametri di livello 3 (come l'indirizzo IP privato nella rete aziendale, il DNS).

Prima di accettare la connessione ad accesso remoto, il gateway aziendale verifica l'utente contattando il security server aziendale tramite il protocollo Remote Authentication Dial In User Service (RADIUS). Il security server aziendale è un **server AAA** centralizzato:

- Authentication: identificare l'utente (per es. tramite nome utente e password);
- Authorization: verificare a quali servizi può accedere l'utente e quali sono limitati all'utente;
- Accounting: tenere traccia dell'attività dell'utente (per es. per addebito).

Le VPN di accesso sono in grado di virtualizzare le connessioni ad accesso remoto tra un utente remoto e la rete aziendale sull'infrastruttura condivisa del service provider per ridurre i costi. Continuerà ad essere usato il protocollo PPP incapsulato nei tunnel della VPN, evitando di cambiare troppo il gateway aziendale.

5.3.2 Customer provision

Nelle VPN di accesso customer provisioned, il tunnel viene stabilito tra l'utente remoto e il gateway aziendale:

1. l'utente richiede di stabilire una connessione ad accesso remoto PPP con il NAS del service provider, chiedendo di negoziare i parametri di configurazione per la connessione tramite LCP e NCP;
2. il NAS verifica l'identità dell'utente attraverso il security server del provider usando il protocollo RADIUS;
3. il NAS fornisce all'utente i parametri di configurazione per la connessione ad accesso remoto PPP, in particolare l'indirizzo IP pubblico;
4. l'utente richiede di aprire un tunnel VPN con il gateway aziendale, inviando una trama PPP contenente un pacchetto IP il cui indirizzo IP di destinazione è l'indirizzo IP pubblico del gateway aziendale;

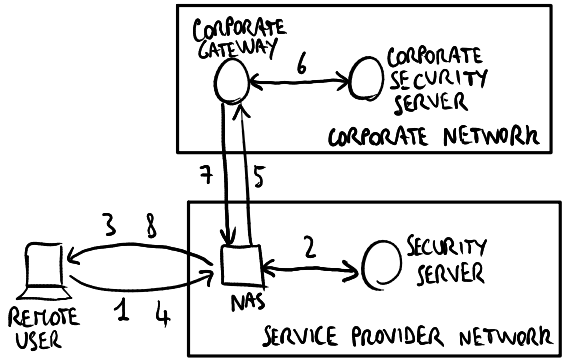


Figura 5.5: VPN di accesso distribuita come customer provision.

5. il NAS inoltra la richiesta sulla rete del service provider al gateway aziendale;
6. il gateway aziendale verifica l'identità dell'utente attraverso il security server aziendale usando il protocollo RADIUS;
7. il gateway aziendale fornisce all'utente i parametri di configurazione per il tunnel VPN, in particolare l'indirizzo IP privato nella rete aziendale;
8. il NAS inoltra la risposta del gateway aziendale all'utente remoto.

Una volta instaurata la VPN, l'utente ha due indirizzi IP: uno pubblico per il NAS del service provider e uno aziendale per il gateway aziendale. Le trame PPP che l'utente può inviare attraverso il tunnel hanno il seguente formato:

intestazione PPP	intestazione IP	intestazione PPP	intestazione IP	payload IP
	src: indirizzo IP pubblico dell'utente dst: indirizzo IP pubblico del gateway aziendale		src: indirizzo IP aziendale dell'utente dst: indirizzo IP aziendale/pubblico della destinazione finale	

Vantaggio L'utente è indipendente dal service provider: quest'ultimo infatti fornisce all'utente solo la connessione a Internet al gateway aziendale.

Svantaggio L'utente può disabilitare temporaneamente la connessione VPN e connettersi ad un server esterno su Internet ⇒ se prende un malware, quando ritorna alla VPN infetterà la rete aziendale.

5.3.3 Provider provision

Nelle VPN di accesso provider provisioned, il tunnel viene stabilito tra il NAS del service provider e il gateway aziendale:

1. l'utente richiede di stabilire una connessione ad accesso remoto PPP con il gateway aziendale, chiedendo di negoziare i parametri di configurazione per la connessione tramite LCP e NCP;
2. il NAS verifica l'identità dell'utente attraverso il security server del provider usando il protocollo RADIUS, in particolare identificando l'utente come un utente della VPN;
3. il NAS richiede di aprire un tunnel VPN, correlato all'utente, con il gateway aziendale, inviando un pacchetto IP il cui indirizzo IP di destinazione è l'indirizzo IP pubblico del gateway aziendale;

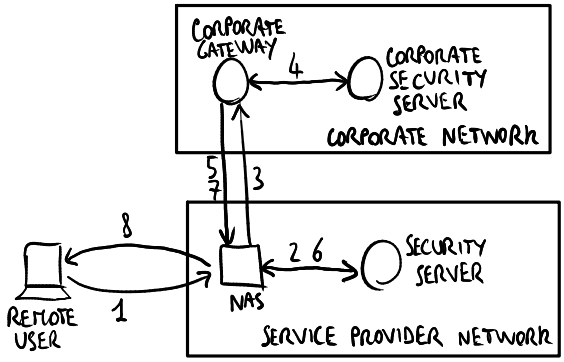


Figura 5.6: VPN di accesso distribuita come provider provision.

4. il gateway aziendale verifica l'identità dell'utente attraverso il security server aziendale usando il protocollo RADIUS;
5. il gateway aziendale fornisce al NAS i parametri di configurazione per il tunnel VPN;
6. il NAS registra opzionalmente l'accettazione e/o il traffico per far pagare il servizio alla società;
7. il gateway aziendale fornisce all'utente i parametri di configurazione per la connessione PPP (l'accesso remoto è virtualizzato), in particolare l'indirizzo IP privato nella rete aziendale;
8. il NAS inoltra la risposta del gateway aziendale all'utente remoto.

Una volta instaurata la VPN, l'utente ha solo l'indirizzo IP aziendale, non consapevole del tunnel tra il NAS del service provider e il gateway aziendale. I pacchetti IP che il NAS può inviare attraverso il tunnel hanno il seguente formato:

intestazione IP		intestazione PPP	intestazione IP		payload IP
src:	indirizzo IP pubblico del NAS del service provider		src:	indirizzo IP aziendale dell'utente	
dst:	indirizzo IP pubblico del gateway aziendale		dst:	indirizzo IP aziendale/pubblico della destinazione finale	

Vantaggio L'utente non può accedere ad Internet senza passare per il gateway aziendale (accesso centralizzato) ⇒ maggiore sicurezza.

Svantaggio L'utente non è indipendente dal service provider: infatti se cambia service provider la connessione VPN non funzionerà più.

5.4 VPN site-to-site

5.4.1 VPN basate su IPsec

Nelle VPN basate su IPsec, IPsec viene usato in tunnel mode tra i gateway VPN: il pacchetto IP tra i due host viene incapsulato in un pacchetto IP, che ha l'intestazione ESP (e opzionalmente l'intestazione AH), tra i due gateway VPN, in modo che anche gli indirizzi IP dei due host possano essere criptati da ESP.

La intranet privata può essere protetta da:

- un firewall: filtra il traffico secondo le politiche aziendali per esempio riguardanti gli indirizzi IP ammessi, e può essere messo in diverse posizioni in relazione al gateway VPN:

  - prima del gateway VPN: il gateway VPN è protetto, ma il firewall non può filtrare il traffico VPN perché è criptato;
  - dopo il gateway VPN: il firewall può filtrare il traffico VPN decriptato, ma il gateway VPN non è protetto;
  - in parallelo al gateway VPN: i pacchetti passano per il firewall sia prima sia dopo il gateway VPN, così il gateway VPN è protetto e il traffico VPN è filtrato, ma il carico di lavoro per il firewall è maggiore;
  - integrato nel gateway VPN: entrambe le funzionalità di gateway VPN e di firewall sono integrate in un unico apparato ⇒ massima flessibilità;

- un Intrusion Detection System (IDS): osserva il traffico provando a rilevare se ci sono degli attacchi in corso, e vengono messe due sonde IDS in parallelo con il gateway VPN:
  - la sonda prima del gateway VPN osserva il traffico che arriva dal tunnel e protegge dagli attacchi provenienti dall'infrastruttura condivisa;
  - la sonda dopo il gateway VPN osserva il traffico della VPN e protegge dagli attacchi provenienti dalla rete aziendale (ad es. malware installato sui PC dei dipendenti, attacchi da un'altra società se la VPN site-to-site è extranet).

La presenza di NAT potrebbe provocare dei problemi con IPsec:

- AH autentica l'intero pacchetto, quindi comprende anche l'intestazione IP ⇒ i NAT hanno bisogno di cambiare gli indirizzi IP, quindi l'autenticazione non funzionerà più;
- ESP cripta il payload, quindi i NAT posizionati lungo il tunnel non saranno in grado di vedere gli indirizzi IP e le porte TCP/UDP criptati per gestire siti VPN diversi.

5.4.2 VPN basate su MPLS

Livello 2: PWE3

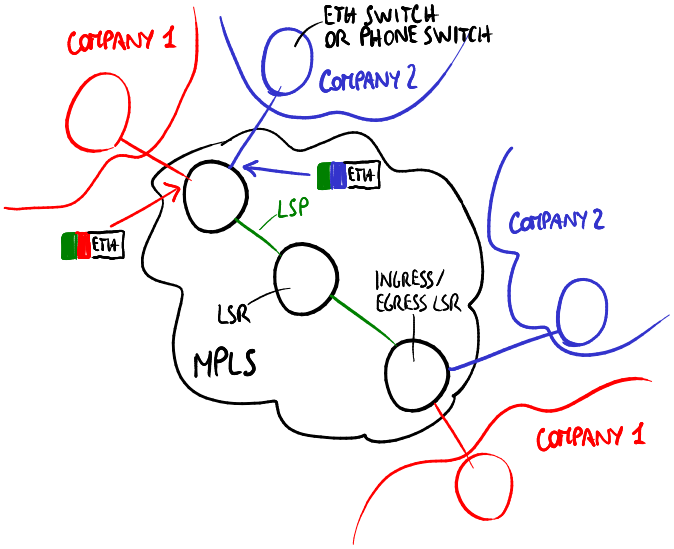


Figura 5.7: Esempio di VPN site-to-site di livello 2 basata su MPLS.

Lo standard **Pseudo Wire Emulation End-to-End** (PWE3) permette di emulare fili su una rete MPLS per scambiare trame Ethernet tra terminali di livello 2 quali switch Ethernet e

centralini telefonici.[1] Questo tipo di connessione richiede alcuni requisiti in termini di ritardi costanti delle trame Ethernet come se esse viaggiassero su un filo fisico.

Il traffico è trasportato attraverso un LSP, quindi deve poter andare ad uno tra più siti collegati a interfacce diverse dell'LSR ingress/egress:

- etichetta esterna: identifica l'LSP tra due LSR ingress/egress, ed è seguita da una tra più etichette interne;
- etichetta interna: identifica il tunnel VPN per una società, e l'LSR ingress/egress la usa per inviare la trama fuori da una delle sue interfacce verso il sito della società.

Le VPN di livello 2 basate su MPLS non sfruttano tutti i vantaggi di MPLS, perché i protocolli di instradamento per il traffic engineering lavorano bene con IP ⇒ di solito gli LSP vengono configurati manualmente.

Livello 3: BGP

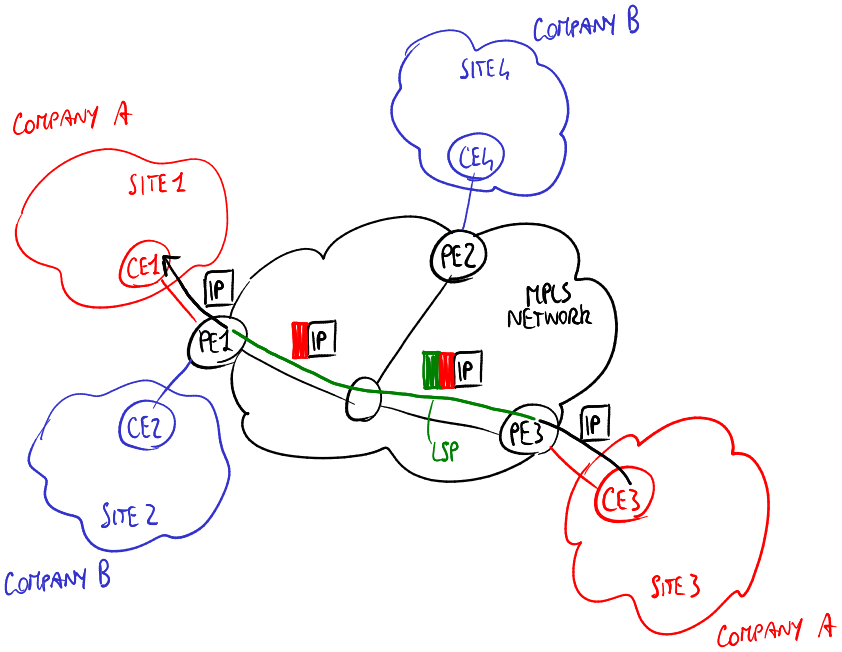


Figura 5.8: Esempio di VPN site-to-site di livello 3 MPLS/BGP.

Border Gateway Protocol (BGP) è stato esteso a **internal BGP** e **external BGP** per supportare la distribuzione delle VPN su MPLS, per esempio per il sito 1 della società A:

1. CE1 della società A annuncia a PE1 le destinazioni nel sito 1, cioè dice a PE1 tutti gli indirizzi IP che possono essere raggiunti nel sito 1, attraverso l'**internal BGP**;
2. PE1 assegna una etichetta ad ogni indirizzo IP da CE1 e registra il mapping in una tabella **VPN Routing and Forwarding** (VRF) specifica della porta verso CE1;
3. PE1 annuncia le etichette scelte agli altri PE attraverso l'**external BGP** inviando pacchetti contenenti ciascuno un indirizzo IP più un **route distinguisher** che identifica la famiglia di indirizzi, che effettivamente è la porta verso il sito della società A, utile nel caso ci siano due indirizzi privati identici in due diverse reti aziendali.
   Questa fase deve essere effettuata manualmente: l'amministratore del sistema deve aprire una **peering session** tra PE1 e ciascuno degli altri PE nella rete MPLS;

[1] È anche possibile usare i router come terminali, ma ha pochissimo senso.

4. gli altri PE possono elaborare i messaggi di advertisement o ignorarli:
    - ogni altro PE a cui è collegato uno dei siti della società A (nella figura PE3) registra nella sua tabella VRF le informazioni annunciate da PE1, cioè l'etichetta scelta per ogni destinazione IP nel sito 1, in altre parole gli indirizzi IP associati alla famiglia di indirizzi del sito 1, più l'indirizzo IP di PE1;
    - ogni altro PE a cui non è collegato alcun sito della società A (nella figura PE2) ignora semplicemente le informazioni annunciate da PE1;

Una volta che le destinazioni IP sono state annunciate tra i PEs, si può iniziare ad inviare dati della VPN lungo gli LSP MPLS, per esempio da CE3 della società A a CE1 della società A:

1. PE3, cioè l'LSR ingress, effettua l'operazione di push di due etichette nella pila delle etichette:
    - etichetta interna: quella precedentemente annunciata da PE1;
    - etichetta esterna: quella decisa dagli LSR attraverso i protocolli MPLS per label binding, distribution e mapping per l'LSP da PE3 a PE1;
2. gli LSR intermedi non si preoccupano dell'etichetta interna, ma operano solo sull'etichetta esterna lungo l'LSP;
3. l'ultimo LSR subito prima di PE1 rimuove l'etichetta esterna (**penultimate label popping**) e invia il pacchetto a PE1;
4. PE1, che è l'LSR ingress, cerca nelle sue tabelle VRF l'etichetta interna e trova la porta dove far uscire il pacchetto, quindi rimuove anche l'etichetta interna e invia il pacchetto a CE1.

Vantaggio Questa soluzione è molto scalabile:

- ogni LSR intermedio deve avere a che fare con tanti LSP quanti sono i PE, not con tanti LSP quante sono le destinazioni IP;
- ogni PE deve avere a che fare con tante tabelle VRF quanti sono i siti ad esso collegati.

Svantaggio Le peering session tra i PE sono da configurare manualmente.

Questa soluzione non fornisce alcuna crittografia perché è provider provisioned e la società deve fidarsi del service provider.

Livello 3: virtual router

Nelle VPN basate su MPLS con **virtual router**, ogni PE esegue un'istanza di (virtual) router per ogni società che è collegata ad esso, in modo che ogni istanza debba avere a che fare solo con una rete aziendale ⇒ il protocollo è più semplice perché deve avere a che fare solo con una VPN alla volta, ma la scalabilità è minore perché le istanze di router devono essere configurate manualmente.

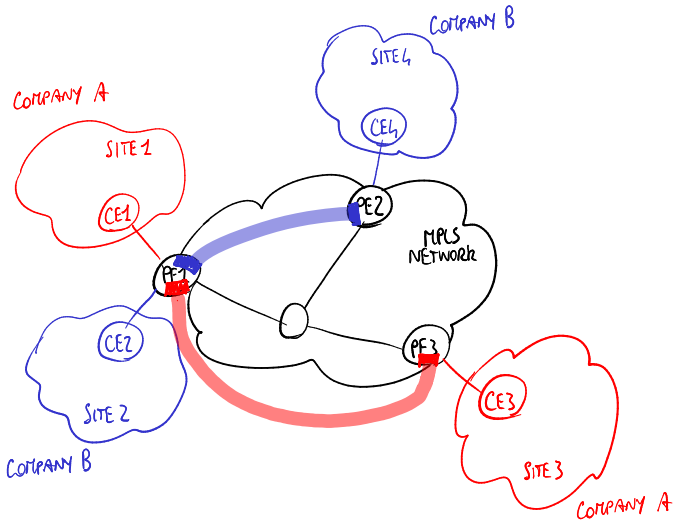


Figura 5.9: Esempio di VPN site-to-site di livello 3 MPLS/virtual router.

5.5 (Pseudo)VPN SSL

SSL può essere usato per creare delle VPN site-to-site e di accesso, ma è principalmente utilizzato nelle **(pseudo)VPN SSL** per garantire un accesso sicuro ai servizi (come servizio Web, posta elettronica) tramite il tunneling basato su TCP/UDP. In questo caso non si passa più dalla creazione di un tunnel, dall'assegnazione di un IP aziendale per accedere al servizio: si usa unicamente SSL per avere sia l'autenticazione che la sicurezza, per questo si parla di (pseudo)VPN SSL. Il vantaggio principale delle (pseudo)VPN SSL è che hanno una buona probabilità di funzionare in qualsiasi scenario di rete, senza alcun problema nell'attraversare i NAT, i firewall e i router, e i servizi SSL possono anche essere acceduti da browser Web attraverso HTTPS.

5.5.1 Confronto con soluzioni alternative

Confronto con IPsec

Vantaggi SSL è più conveniente di IPsec in termini di:

- utilizzo: IPsec ha troppe opzioni da configurare e da amministrare, mentre SSL richiede solo di compilare l'applicazione insieme a una libreria che implementa SSL;
- sicurezza: lavora a livello applicazione ⇒ un attacco a SSL coinvolge solo l'applicazione e non l'intero sistema operativo;
- maturità: è usato da molti anni con molte versioni ⇒ pochi bug nel codice;
- compatibilità con il NAT traversal:
  - nessuna autenticazione dell'intestazione IP perché l'SSL è sopra il livello trasporto;
  - nessuna crittografia delle porte come con ESP di IPsec.

Svantaggio SSL è critico con gli attacchi DoS: i pacchetti hanno sempre bisogno di essere elaborati fino al livello trasporto, mentre in IPsec essi vengono scartati già a livello rete. Questo significa che, se l'attaccante riesce a sfruttare un bug a livello 3 o 4, con IPsec si avrebbe comunque protezione, mentre con SSL no.

Confronto con PPTP

SSL supera alcuni problemi di PPTP:

- interoperabilità scarsa con piattaforme non Microsoft;
- alcuni amministratori di rete potrebbero configurare i router per bloccare GRE, su cui si basa PPTP, a causa del fatto che a loro non piace la funzionalità di source routing.

5.5.2 Flavor di (pseudo)VPN SSL

Web proxying Il server Web non supporta HTTPS ⇒ un "VPN gateway",[2] al margine della rete aziendale, scarica le pagine Web dal server Web tramite HTTP, e le consegna all'utente, al di fuori della rete aziendale, tramite HTTPS.

Port forwarding Il client esegue un'applicazione che supporta un protocollo applicativo (come FTP, SMTP, POP3) ⇒ un port forwarder installato sul client converte i pacchetti, inviati a una specifica porta, dal protocollo applicativo in pacchetti HTTPS inviandoli da un'altra porta, e viceversa.

Application translation Il server Web è un server applicativo (ad es. server di posta) che supporta un protocollo applicativo (come FTP, SMTP, POP3) ⇒ il "gateway VPN" converte HTTPS nel protocollo applicativo e viceversa tramite un meccanismo di port forwarding implementato nel "gateway VPN".

Protocolli SSL'ed Alcuni protocolli applicativi possono integrare nativamente SSL in se stessi (ad es. SMTP-over-SSL, POP-over-SSL) ⇒ il client e il server Web possono comunicare direttamente in modo sicuro senza la necessità della traduzione da parte di un "gateway VPN".

Application proxying Il client utilizza un protocollo SSL'ed, ma il server Web non lo supporta ⇒ è ancora necessario un "gateway VPN" con meccanismo di port forwarding.

Network extension Sia il client sia il server Web non supportano SSL ⇒ sono necessari due "gateway VPN", uno dal lato dell'utente e l'altro dal lato del server Web, così il tunnel SSL viene creato tra i due "gateway VPN".[3]

[2] Questa è una definizione lasca di gateway VPN: in verità sarebbe più accurato definirlo come un proxy.
[3] Questa non è una (pseudo)VPN SSL.

Capitolo 6

VoIP

Voice over IP (VoIP) è l'insieme delle tecnologie per trasportare le chiamate vocali, insieme ai dati multimediali, su reti IP.

6.1 Commutazione di circuito versus commutazione di pacchetto

6.1.1 Rete telefonica a commutazione di circuito

Nella rete telefonica tradizionale a **commutazione di circuito** (POTS), la voce è trasportata tramite l'allocazione di circuiti statici in cui la voce è campionata a un bit rate di 64 Kbps (secondo il teorema del campionamento). Utilizzando una tale rete ci sono alcune limitazioni:

- nessuna compressione: non avrebbe senso risparmiare dei bit poiché sono staticamente allocati 64 bit al secondo per ogni telefonata;
- per supportare la comunicazione multimediale o multicanale va allocato un numero intero di circuiti;
- nessuna soppressione dei silenzi: i campioni vocali vengono trasmessi anche durante le pause e i circuiti rimangono allocati;
- nessuna multiplazione statistica: non è possibile condividere dinamicamente della larghezza di banda tra più chiamate a seconda delle loro esigenze;
- è richiesta la procedura di segnalazione (tono di chiamata, tono di occupato, tono di inattività, ecc.) per l'allocazione dei circuiti.

6.1.2 Rete dati a commutazione di pacchetto

In una rete dati a **commutazione di pacchetto** (IP), la voce viene trasportata dinamicamente tramite i pacchetti, e ciò permette delle nuove funzionalità:

- migliore compressione per un numero di pacchetti più ridotto;
- comunicazione ad alta qualità: il bit rate non è più limitato a 64 Kbps;
- soppressione dei silenzi: non viene trasmesso alcun pacchetto durante le pause;
- multiplazione statistica: l'allocazione della larghezza di banda è flessibile;
- la procedura di segnalazione non alloca più delle risorse statiche;

- nomadicità: l'utente può essere raggiungibile attraverso lo stesso numero di telefono o lo stesso account quando si sposta.

Tuttavia viene introdotto un nuovo problema: le risorse non si possono veramente riservare in una rete a commutazione di pacchetto ⇒ è molto difficile garantire la qualità del servizio per le chiamate vocali perché i pacchetti potrebbero arrivare con dei ritardi oppure potrebbero andare persi:

- ritardi: sono stati definiti da ITU alcuni valori di riferimento per i ritardi end-to-end:
  - 0–150 ms: questo è accettabile per l'orecchio umano;
  - 150–400 ms: questo è accettabile solo per le chiamate inter-continentali;
  - > 400 ms: questo non è accettabile e nuoce alla conversazione;
- perdite: l'orecchio umano può tollerare senza problemi al più il 5% di pacchetti mancanti.

TCP o UDP? I pacchetti UDP e TCP arrivano (teoricamente) al ricevitore nello stesso tempo; l'unica differenza è che il TCP deve attendere i pacchetti di acknowledge ⇒ l'UDP sarebbe la scelta più naturale. In realtà Skype spesso utilizza il TCP perché è più semplice passare attraverso NAT e firewall anche se qualche volta potrebbero verificarsi dei piccoli silenzi dovuti ai meccanismi delle finestre a scorrimento.

6.2 Migrazione dalla commutazione di circuito alla commutazione di pacchetto

La tradizionale rete a commutazione di circuito (POTS) può essere migrata ad una rete a commutazione di pacchetto basata su IP in modo graduale:

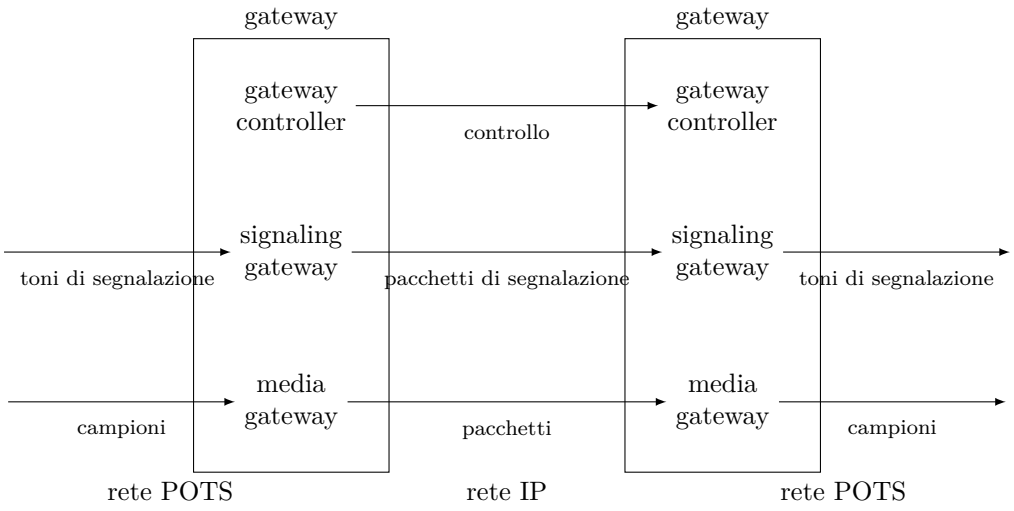
1. rete basata su **Telephone over IP** (ToIP): i terminali ai margini della rete funzionano ancora in un modo a commutazione di circuito, ma il backbone della rete è basato su IP ed effettua internamente la pacchettizzazione ⇒ poiché l'utilizzo di VoIP è nascosto all'utente i servizi multimediali aggiuntivi non sono disponibili all'utente finale.
   I nuovi operatori di telecomunicazioni possono creare le loro reti telefoniche come reti basate su ToIP ⇒ gli operatori di telecomunicazioni possono risparmiare denaro costruendo e mantenendo una singola infrastruttura integrata;
2. rete mista: alcuni terminali sono VoIP, altri sono ancora tradizionali;
3. rete IP: tutti i terminali sono VoIP, ma i servizi di rete intelligente (per es. numeri verdi) sono ancora tradizionali, perché funzionano così bene che gli operatori di rete sono restii ad ammodernarli;
4. rete solo IP: tutti i terminali sono VoIP e tutti i servizi di rete intelligente lavorano su IP.

6.2.1 Gateway

Il **gateway** è un apparato che consente di connettere una rete POTS con una rete IP. È composto da tre componenti:

- **media gateway**: è in grado di convertire i campioni vocali dalla rete POTS in pacchetti di dati alla rete IP e viceversa;
- **signaling gateway**: è in grado di convertire i toni di segnalazione dalla rete POTS in pacchetti di segnalazione alla rete IP e viceversa;[1]

[1] La distinzione tra media e signaling gateway spesso non è chiara: infatti i toni di segnalazione sono dei normali campioni audio, e i pacchetti di segnalazione sono dei normali pacchetti di dati.



- **gateway controller**: è responsabile della supervisione e del monitoraggio dell'intero gateway, controllando la qualità del traffico, effettuando l'autorizzazione, effettuando l'autenticazione (per addebito), localizzando le destinazioni, e così via.
  Il gateway controller da solo è anche utile nelle reti solo IP.

6.3 Fasi per la creazione di flussi VoIP

6.3.1 Al lato trasmettitore

Il trasmettitore deve effettuare i seguenti passi:

1. campionamento
2. codifica
3. pacchettizzazione
4. accodamento
5. trasmissione

Campionamento

Il campionamento consente di convertire la voce da un segnale analogico in campioni digitali. Il campionamento è caratterizzato dalla sensibilità (bit), dalla frequenza di campionamento (hertz = 1/s) e dal bit rate teorico (bit/s).

Codifica

Le tecniche di codifica consentono di ridurre il bit rate, ma potrebbero introdurre dei ritardi aggiuntivi dovuti agli algoritmi di codifica.

Le principali tecniche di codifica sono:

- **codifica differenziale**: ogni campione viene codificato in base alle differenze rispetto al campione precedente e al campione successivo;
- **codifica pesata**: durante una videochiamata la figura dell'interlocutore deve venire codificata a un bit rate maggiore rispetto all'ambiente circostante;

- **codifica con perdita**: alcune informazioni audio e video vengono rimosse in modo irreversibile (possibilmente la perdita di qualità non deve essere percepita dai sensi umani).

La complessità è una questione importante quando gli algoritmi di codifica devono essere eseguiti su terminali mobili (come sistemi embedded o dispositivi a bassa potenza non molto performanti). Inoltre alcuni servizi non supportano i dati codificati con una compressione con perdita: per esempio, un fax non supporta la perdita di qualità. Per questi motivi gli operatori di telecomunicazioni preferiscono comunque utilizzare il codec PCM64 con un bit rate costante di 64 Kbps: richiede meno energia per l'elaborazione ed è supportato anche dai fax e dagli altri servizi che utilizzano la rete telefonica.

La voce di chi parla, dopo essere uscita dall'altoparlante del ricevitore, potrebbe tornare indietro attraverso il microfono del ricevitore arrivando all'altoparlante del trasmettitore dopo un ritardo chiamato **round trip delay** che se significativo potrebbe disturbare chi parla ⇒ la **cancellazione dell'eco** è pensata per evitare che chi parla senta la propria voce.

Pacchettizzazione

Il ritardo di pacchettizzazione dipende dal numero di campioni inserito in ogni pacchetto, che è un compromesso tra ritardo ed efficienza:

- ritardo: se vengono messi troppi campioni in un singolo pacchetto, il pacchetto deve attendere l'ultimo campione prima di essere inviato ⇒ se vengono pacchettizzati insieme troppi campioni, il primo campione arriverà con un ritardo importante;
- efficienza: ogni pacchetto IP ha un overhead nella dimensione dovuto alle intestazioni ⇒ se vengono pacchettizzati insieme troppo pochi campioni, il bit rate aumenterà in modo significativo a causa dell'overhead delle intestazioni.

Possono essere previste delle tecniche di correzione degli errori basate sulla **ridondanza**: ogni pacchetto trasporta anche il campione precedente insieme al nuovo campione, così se il pacchetto precedente va perso sarà comunque possibile recuperarne il campione.

Accodamento

Quando il traffico in ingresso supera la capacità del canale in uscita, il router deve immagazzinare i pacchetti in attesa della trasmissione (buffering) ⇒ ciò aumenta il ritardo e il jitter. La gestione delle code a priorità affronta queste problematiche (vedere il capitolo 7 per i dettagli sulla qualità del servizio).

Trasmissione

Al fine di ridurre i ritardi di trasmissione ci sono alcune soluzioni possibili:

- aumentare la larghezza di banda, ma i provider ADSL di solito sono più interessati ad aumentare solo la larghezza di banda in downstream;
- utilizzare il PPP interleaving, cioè suddividere una trama grande in diverse trame PPP più piccole, ma i provider non sempre implementano il PPP interleaving;
- evitare l'utilizzo di altre applicazioni (ad es. trasferimento dati) durante le chiamate vocali.

6.3.2 Al lato ricevitore

Il ricevitore deve effettuare i seguenti passi:

1. **de-jitter**: i moduli per il de-jitter devono riprodurre i pacchetti alla stessa velocità impiegata per generarli;

2. **riordinamento**: poiché è a commutazione di pacchetto la rete potrebbe consegnare dei pacchetti out-of-order ⇒ è necessario un modulo per il riordinamento;

3. **decodifica**: gli algoritmi per la decodifica devono implementare alcune tecniche:

   - i pacchetti mancanti vanno gestiti utilizzando delle tecniche di predizione, inserendo del rumore bianco o riproducendo i campioni dell'ultimo pacchetto ricevuto;
   - **soppressione dei silenzi**: il ricevitore introduce del rumore bianco durante le pause nella conversazione, perché i perfetti silenzi vengono percepiti dall'utente come malfunzionamenti della chiamata. È importante riuscire a interrompere immediatamente il rumore bianco non appena l'interlocutore riprende a parlare.

6.4 RTP

Real-time Transport Protocol (RTP) è utilizzato per trasportare i flussi VoIP su UDP.

6.4.1 Funzionalità

Trasmissione in multicast nativa RTP permette la trasmissione in multicast anche su una rete che non supporta il multicast.
In realtà IP supporta multicast, ma il suo utilizzo richiede che il provider della rete configuri gli apparati di rete per creare un gruppo multicast per ogni flusso VoIP ⇒ RTP permette di inviare dati in multicast a livello applicazione in modo plug-and-play senza l'invervento del provider della rete.

Solo le funzionalità essenziali RTP non specifica le funzionalità che si suppongono essere gestite dai livelli sottostanti, come la frammentazione dei pacchetti e il rilevamento degli errori di trasmissione (checksum).

Indipendenza dai formati di dati RTP include solo il campo "Payload Type" per specificare il tipo di contenuto del pacchetto e il codec utilizzato, ma non specifica come codificare i dati e quali codec utilizzare (queste informazioni sono specificate separatamente dai documenti "Audio Video Profiles").
È impossibile associare un codice a ogni codec nel mondo ⇒ il trasmettitore e il ricevitore devono concordare sui codici da utilizzare per identificare i codec durante l'instaurazione della sessione, e tali codici sono validi solo nella sessione.

Trasporto di dati in tempo reale Sono permessi pacchetti mancanti ⇒ i campi "Sequence Number" e "Timestamp" sono combinati per far ripartire la riproduzione audio/video all'istante di tempo giusto in caso di perdita dei pacchetti.

Differenziazione dei flussi Una sessione multimediale ha bisogno di aprire una sessione RTP, quindi una connessione UDP, per ogni flusso multimediale (audio, video, lavagna, ecc.).

RTP Control Protocol (RTCP) Effettua il monitoraggio e il controllo della connessione: la destinazione raccoglie alcune statistiche (informazioni su perdite, ritardi, ecc.) e le manda periodicamente alla sorgente in modo che quest'ultima possa ridurre o aumentare la qualità del flusso multimediale al fine di far funzionare il più possibile il servizio secondo le capacità correnti della rete. Per esempio, il ricevitore può capire che un certo codec ha un bit rate troppo elevato che non è supportato dalla rete, e perciò può passare a un codec avente un bit rate più basso.

Porte non standard RTP non definisce delle porte standard ⇒ i pacchetti RTP sono difficili da rilevare per i firewall e per la qualità del servizio. Tuttavia alcune implementazioni utilizzano degli intervalli di porte statici, per evitare di aprire troppe porte sui firewall e per semplificare la marcatura per la qualità del servizio.

6.4.2 Trasmissione in multicast

Soluzioni tradizionali

(a) Lato TX: $N-1$ flussi. *(b) Lato RX: $N-1$ flussi.*

Figura 6.1: Host unicast.

(a) Lato TX: 1 flusso. *(b) Lato RX: $N-1$ flussi.*

Figura 6.2: Host multicast.

Le soluzioni tradizionali senza il mixer RTP richiedono sempre delle capacità di banda elevate per tutti gli host.

Mixer RTP

(a) Lato TX: 1 flusso. *(b) Lato RX: 1 flusso.*

Figura 6.3: Host unicast con mixer.

Il **mixer RTP** è un dispositivo in grado di manipolare i flussi RTP per le trasmissioni in multicast: per esempio, in una videoconferenza il mixer per ogni host prende i flussi provenienti dagli altri host e li mescola in un singolo flusso verso quell'host.

Ogni host trasmette e riceve un singolo flusso ⇒ il mixer è utile per risparmiare banda: anche un host con una bassa larghezza di banda può unirsi alla videoconferenza. Il mixer dovrebbe essere l'host avente la capacità di banda più elevata, in modo da poter ricevere tutti i flussi dagli altri host e trasmettere tutti i flussi agli altri host.

6.4.3 Intestazione RTP

L'intestazione RTP ha il seguente formato:

2	3	4	8	9	16	32
V	P	X	CC	M	Payload Type	Sequence Number
Timestamp						
Synchronization source identifier (SSRC)						
Contributing source identifier (CSRC) :::						

dove i campi più significativi sono:

- campo CSRC Count (CC) (4 bit): specifica il numero di identificativi nel campo "CSRC";
- flag Marker (M) (1 bit): è usato per marcare il pacchetto come ad alta priorità o a bassa priorità per la qualità del servizio;
- campo Payload Type (PT) (7 bit): specifica il tipo di payload del pacchetto; generalmente contiene il codice corrispondente al codec utilizzato;
- campo Synchronization source identifier (SSRC) (32 bit): identifica il mixer RTP (il mixer M nell'esempio sottostante);
- campo Contributing source identifier (CSRC) (lunghezza variabile): identifica le sorgenti multiple che contribuiscono a un flusso multicast (le sorgenti S_1, S_2, S_3 nell'esempio sottostante).

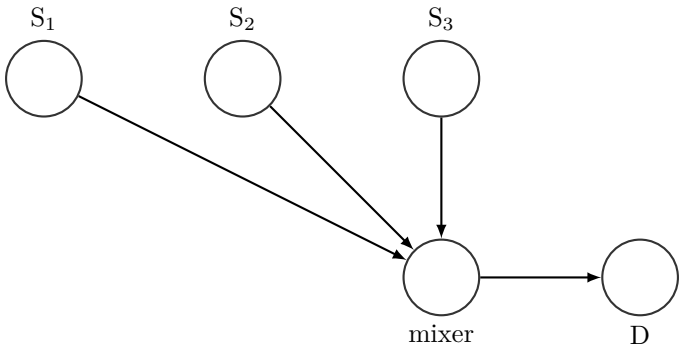


Figura 6.4: Le voci dalle sorgenti S_1, S_2, S_3 vengono mescolate in un flusso verso la destinazione D.

6.5 H.323

H.323 è una suite di protocolli di segnalazione di livello applicazione standardizzata da ITU. È uno standard molto complesso perché eredita le logiche dagli operatori di telefonia.

6.5.1 Componenti di una rete H.323

H.323 fu sviluppato originariamente per consentire la comunicazione (audio, video, lavagna condivisa...) tra host connessi ad una LAN aziendale[2] e dispositivi remoti connessi alla tradizionale rete a commutazione di circuito (PSTN):

- **gatekeeper**: implementa il gateway controller, responsabile dell'autenticazione e della localizzazione degli utenti, di tenere traccia degli utenti registrati, ecc.;[3]

[2]Sarebbe meglio parlare più in generale di reti aziendali, perché H.323 non fornisce in realtà alcuna assunzione sulla tipologia della rete sottostante.

[3]Il gatekeeper non è obbligatorio: un cliente può contattare direttamente una destinazione se ne conosce l'indirizzo.

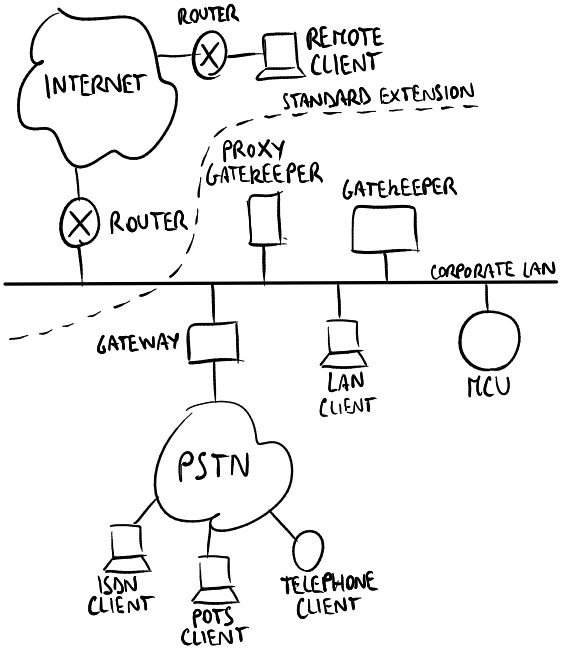


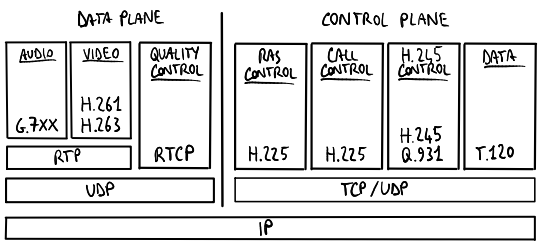
Figura 6.5: Esempio di una rete H.323.

- **proxy gatekeeper**: il client contatta il gatekeeper indirettamente attraverso il proxy gatekeeper ⇒ questo riduce gli sforzi per i dispositivi client a bassa potenza, ma non è obbligatorio;
- **Multipoint Control Unit** (MCU): implementa il mixer RTP;
- **gateway**: implementa il signaling gateway e il media gateway, traducendo i canali dati, i canali di controllo e le procedure di segnalazione tra la LAN e la PSTN, ed è visto come terminale H.323 nella LAN e come terminale telefonico nella PSTN.

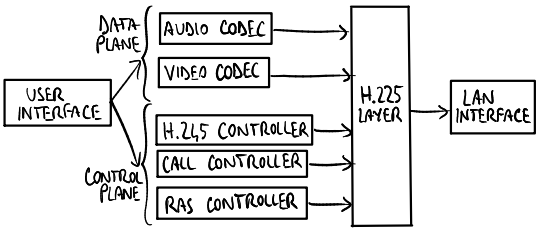
Successivamente lo standard H.323 è stato esteso su una rete geografica (WAN), permettendo la comunicazione anche con utenti remoti tramite Internet.

La **zona** di un gatekeeper è costituita dall'insieme dei terminali che esso gestisce. Una zona può coinvolgere livelli di rete diversi, come più LAN separate da router.

6.5.2 Architettura protocollare di H.323



(a) Pila protocollare di H.323.



(b) Diagramma a blocchi di un terminale H.323.

La pila protocollare di H.323 è piuttosto complessa perché è composta da diversi protocolli:

- piano dati: consiste dei protocolli RTP e RTCP che stanno su UDP (vedere la sezione 6.4);

- piano di controllo: consiste dei protocolli che stanno su TCP/UDP per la segnalazione:
  - **controllore RAS**: permette a un terminale di scambiare messaggi di controllo con il gatekeeper:
    - *messaggi di Registration*: il terminale chiede al gatekeeper di unirsi ad una zona;
    - *messaggi di Admission*: il terminale chiede al gatekeeper di contattare un altro terminale;
    - *messaggi di Status*: il terminale dice al gatekeeper se è attivo;
    - *messaggi di banda*: il controller RAS notifica il gatekeeper sui cambiamenti nella banda, anche mentre è in corso la chiamata, cosicché il gatekeeper potrà negare nuove chiamate se il canale è sovraccarico;
  - **controllore chiamate**: permette a un terminale di scambiare messaggi di controllo direttamente con un altro terminale;
  - **controllore H.245**: permette a una coppia di terminali di concordarsi su parametri come i codec;
  - **dati**: permette a un terminale di inviare messaggi di controllo per la condivisione della scrivania o altri flussi di dati multimediali.

Alla fine il livello **H.225** mette insieme tutti i messaggi: permette di creare una sorta di **tunnel virtuale affidabile** per l'invio di messaggi H.323 sulla inaffidabile rete IP emulando l'affidabilità di un circuito.

6.5.3 Indirizzamento

Ogni terminale è identificato da una coppia (indirizzo IP, porta TCP/UDP), così può essere contattato direttamente attraverso la coppia indirizzo/porta senza la necessità di un gatekeeper.

Se c'è un gatekeeper, le coppie indirizzo/porta possono essere mappate in **alias** più facili da ricordare per gli utenti (per esempio name@domain.com, numero di telefono E-164, nickname). Poiché sono associati ad account utente, gli alias permettono la nomadicità: un utente continuerà ad essere raggiungibile anche se si sposta cambiando indirizzo IP.

6.5.4 Fasi principali di una chiamata H.323

Una chiamata H.323 avviene in sei fasi principali:

1. registrazione: il terminale chiamante cerca un gatekeeper entro la sua zona e apre un canale RAS usando il controllo RAS;
2. instaurazione della chiamata: il terminale chiamante instaura il canale verso il terminale chiamato usando il controllo di chiamata;
3. negoziazione: vengono negoziati parametri come la larghezza di banda e i codec usando il controllo H.245;
4. trasferimento dati: la voce è trasportata da RTP;
5. chiusura: viene chiuso il canale dati usando il controllo H.245;
6. abbattimento: viene chiuso il canale RAS usando il controllo RAS.

Il gatekeeper può giocare due ruoli:

- gatekeeper routed call: la chiamata passa sempre attraverso il gatekeeper ⇒ questo può essere utile per il NAT traversal: il gatekeeper agisce come un server relay;
- gatekeeper direct endpoint: la chiamata va direttamente al punto di arrivo, ma prima i client chiamante e chiamato devono effettuare la fase di Admission con il gatekeeper a scopi di addebito e gestione della banda.[4]

[4] La fase di Admission non è obbligatoria se il chiamante conosce l'indirizzo IP del chiamato.

6.5.5 Principali problemi e critiche

- lo standard H.323 non fornisce alcuna assistenza per le **tolleranze ai guasti** perché prevede solo un singolo gatekeeper ⇒ i fornitori hanno sviluppato le proprie personalizzazioni, che sono incompatibili tra di loro, per fornire questa funzionalità;
- lo standard H.323 non fornisce alcun supporto per la **comunicazione tra zone diverse** ⇒ un'azienda non può "unire" la sua zona con quella di un'altra azienda;
- i messaggi sono codificati usando il formato **ASN.1**: non è testuale, perciò il debug è molto difficile ed è necessario avere a che fare con dettagli di basso livello delle macchine (per es. little-endian);
- la pila protocollare è costituita da **molti protocolli**, uno per ogni funzionalità.

6.6 SIP

Session Initiation Protocol (SIP) è un protocollo di segnalazione di livello applicazione standardizzato da IETF tramite un RFC. Oggi SIP sta crescendo molto più velocemente di H.323, principalmente grazie al suo approccio di seguire la filosofia di Internet ("keep it simple"): per esempio, usa un approccio basato sul **testo** (come HTTP), così la codifica è facile da capire. L'interazione è **client-server**.

6.6.1 Funzionalità

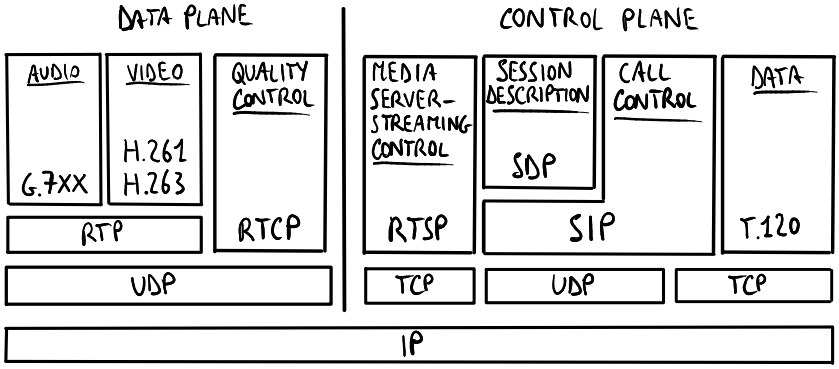


Figura 6.7: Pila protocollare di SIP.

La pila protocollare di SIP è più semplice di quella di H.323 perché SIP è un livello comune nel piano di controllo. SIP copre solo la segnalazione: affida gli aspetti non correlati alla segnalazione, come la gestione della larghezza di banda, ad altri protocolli già esistenti, riducendo la complessità della sua progettazione:

- **RTP/RTCP**: viene usato per trasmettere e controllare un flusso multimediale (vedere la sezione 6.4);
- **SDP**: viene usato per notificare delle informazioni di controllo sui flussi multimediali (vedere la sezione 6.6.4);
- **RTSP** (Real Time Streaming Protocol): è un protocollo simile a RTP utilizzato per gestire sia i flussi in tempo reale sia altri tipi di risorse (per es. l'avanzamento rapido di un messaggio vocale registrato per una segreteria telefonica);

- **RSVP**: viene usato per riservare risorse sulle reti IP, tentando[5] di creare una sorta di rete a commutazione di circuito su una a commutazione di pacchetto (vedere la sezione 7.3).

SIP può operare su uno di tre possibili livelli di trasporto:

- **UDP**: non deve essere tenuta attiva una connessione TCP ⇒ adatto per dispositivi a bassa potenza;
- **TCP**: garantisce una maggiore affidabilità ed è utile per il NAT traversal e per attraversare i firewall;
- **TLS** (TCP con SSL): i messaggi sono criptati a scopo di sicurezza, ma viene perso il vantaggio dei messaggi testuali.

SIP fornisce alcuni servizi principali alle chiamate vocali:

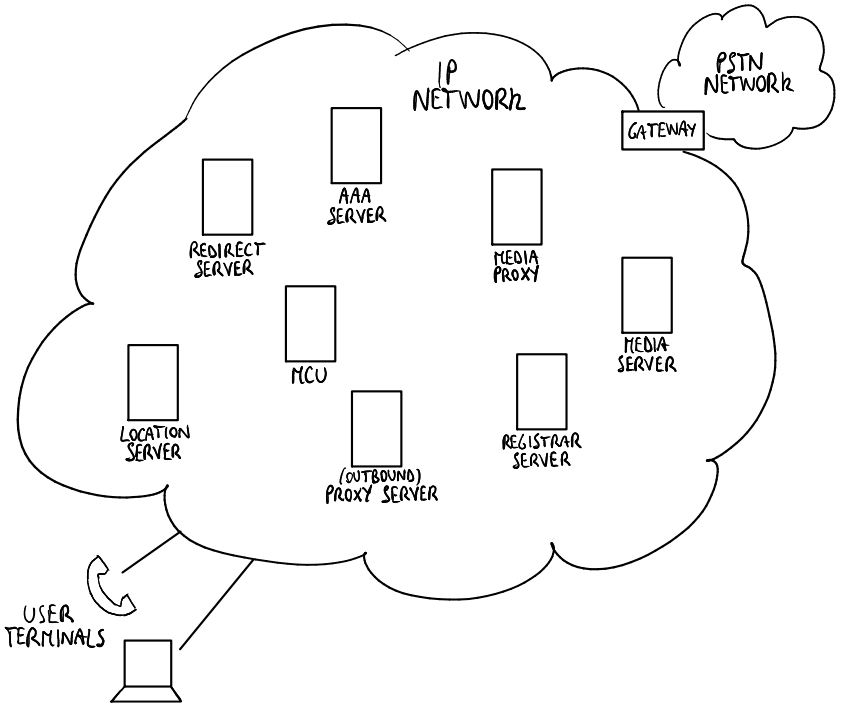
- localizzazione dell'utente: definisce il terminale di destinazione da contattare per la chiamata;
- capacità dell'utente: definisce i mezzi (audio, video...) e i parametri (codec) da usare;
- disponibilità dell'utente: definisce se il chiamato vuole accettare la chiamata;
- instaurazione della chiamata: stabilisce una connessione con tutti i suoi parametri;
- gestione della chiamata.

La segnalazione SIP può essere usata per diversi servizi aggiuntivi oltre alle chiamate vocali: e-presence (lo stato dell'utente: disponibile, occupato, ecc.), messaggistica istantanea, condivisione di una lavagna, trasferimento di file, giochi interattivi e così via. SIP supporta la **nomadicità**: a ogni utente è associato un account, così continuerà ad essere raggiungibile anche se si sposta cambiando indirizzo IP.

6.6.2 Componenti di una rete SIP

- **Terminale**: ogni host deve essere sia client sia server (server per essere raggiungibile).
- **Registrar server**: è responsabile di tenere traccia delle mappature tra gli host e gli indirizzi IP.
  Implementa il gatekeeper: un host deve registrarsi per entrare in una rete SIP.
- **Proxy server**: gestisce lo scambio di messaggi tra gli host e gli altri server.
  Un host potrebbe decidere di parlare solo con il proxy server, delegando ad esso tutte le operazioni richieste per le chiamate SIP.
- **Redirect server**: è utilizzato per reindirizzare le chiamate in arrivo (per es. un utente vuole essere raggiungibile sul suo cellulare di lavoro solo durante le ore lavorative).
- **Media server**: è utilizzato per archiviare dei contenuti a valore aggiunto (ad es. segreteria telefonica).
- **Media proxy**: può essere utilizzato come server relay per attraversare i firewall.
- **Location server**: è utilizzato per localizzare gli utenti.
  Quando un host vuole fare una telefonata chiede al location server di trovare l'indirizzo dell'utente di destinazione.

[5] RSVP si limita a tentare di fare così, perché è impossibile garantire un servizio a commutazione di circuito su una rete a commutazione di pacchetto.



- **AAA server** (Authentication, Authorization, Accounting): il registrar server scambia messaggi con l'AAA server per verificare gli utenti (per es. se l'utente è autorizzato a entrare nella rete).
- **Gateway**: connette la rete IP alla rete PSTN, traducendo i pacchetti SIP in campioni e viceversa.
- **Multipoint Control Unit** (MCU): implementa il mixer RTP, con le stesse funzionalità come in H.323.

In molti casi una singola macchina, chiamata **server SIP** (o SIP proxy), implementa le funzionalità di registrar server, proxy server, redirect server, media proxy. Inoltre, il location server di solito è localizzato nel server DNS, e l'AAA server di solito è localizzato nel server AAA aziendale.

6.6.3 Accounting e domini

Ogni utente ha un account SIP, così continuerà ad essere raggiungibile anche se si sposta cambiando indirizzo IP (**nomadicità**). Gli indirizzi degli account sono nella forma `nome_utente@dominio.com`; anche i terminali telefonici possono avere degli indirizzi SIP nella forma `numero_di_telefono@gateway`.

Una rete SIP ha una architettura distribuita: ogni server SIP è responsabile di un **dominio SIP** (l'equivalente della zona H.323), e tutti gli host che fanno riferimento allo stesso server SIP appartengono allo stesso dominio SIP e hanno lo stesso nome di dominio negli indirizzi di account. A differenza di H.323, un utente può contattare un utente appartenente a un altro dominio SIP: il suo server SIP sarà responsabile di contattare il server SIP dell'altro utente.

Si supponga che un utente americano appartenente al dominio Verizon si sposti in Italia e si colleghi alla rete di Telecom Italia. Per continuare ad essere raggiungibile ha bisogno di contattare il server SIP di Verizon per registrarsi, ma sta usando l'infrastruttura di rete di Telecom Italia $\Rightarrow$ ha bisogno di passare attraverso il server SIP di Telecom Italia SIP server, che è il suo **outbound proxy server**, come un servizio di tipo roaming, e in questo modo Telecom Italia può tenere traccia delle chiamate dell'utente a scopi di addebito.

Interconnessione di domini

Per interconnettere i domini, è necessario che tutti i registrar server possano essere trovati, poiché essi memorizzano le mappature tra gli alias degli account e gli indirizzi IP ⇒ sono necessari due record aggiuntivi nei server DNS per localizzare i registrar server:

- **record NAPTR**: definisce quale protocollo di trasporto può essere usato per il dominio specificato, specificando l'alias da utilizzare per la query SRV;
- **record SRV**: specifica l'alias, da utilizzare per la query A/AAAA, del registrar server e la porta per il protocollo di trasporto specificato;
- **record A/AAAA**: specifica l'indirizzo IPv4/IPv6 dell'alias del registrar server specificato.

La tabella dei record DNS può contenere più di un record SRV/NAPTR:

- più record NAPTR: sono disponibili più registrar server per il protocollo di trasporto specificato, e il campo "Preference" specifica la preferenza d'ordine;
- più record SRV: sono disponibili più protocolli di trasporto per il dominio specificato, e il campo "Priority" specifica la preferenza d'ordine (in ordine: TSL/TCP, TCP, UDP);

oppure può non contenere alcun record SRV/NAPTR:

- nessun record NAPTR: l'host si limita a provare delle query SRV (spesso UDP) e userà il protocollo di trasporto corrispondente alla prima reply SRV;
- nessun record SRV: l'indirizzo del server registrar deve essere configurato staticamente sull'host, e l'host userà la porta standard 5060.

Standard ENUM Come digitare l'indirizzo di un account su un telefono tradizionale per contattare un utente SIP? Ogni account SIP è associato per impostazione predefinita a un numero di telefono chiamato **indirizzo E.164**:

1. l'utente digita il numero di telefono sul suo telefono tradizionale;
2. il gateway tra la rete POTS e la rete SIP converte il numero di telefono in un alias con il dominio fisso `e164.arpa` e interroga il DNS chiedendo se esistono dei record NAPTR:
   (a) se vengono trovati dei record NAPTR dal server DNS, il numero di telefono è associato a un account SIP e la chiamata è inoltrata al proxy SIP di destinazione;
   (b) se non viene trovato alcun record NAPTR dal server DNS, il numero di telefono corrisponde a un utente nella rete POTS.

6.6.4 Messaggi SIP

Ogni messaggio SIP ha il seguente formato testuale:

1. tipo di messaggio (una riga): specifica il tipo di messaggio;
2. intestazione SIP: contiene informazioni sul flusso multimediale;
3. riga vuota (comportamento come HTTP);
4. messaggio SDP (payload): contiene informazioni di controllo sul flusso multimediale.

Principali tipi di messaggio

Un messaggio SIP può essere di diversi tipi, tra cui:

- messaggio `REGISTER`: è usato per registrarsi a un dominio, e può essere inviato in multicast a tutti i registrar server;
- messaggio `INVITE`: è usato per instaurare una telefonata;
- messaggio `ACK`: è l'ultimo messaggio SIP subito prima dell'inizio del flusso RTP;[6]
- messaggio `BYE`: è usato per chiudere una telefonata;
- messaggio `CANCEL`: è usato per annullare una richiesta in sospeso per l'instaurazione di una chiamata;
- messaggi `SUBSCRIBE`, `NOTIFY`, `MESSAGE`: sono usati per la e-presence e la messaggistica istantanea;
- messaggi con codice: includono:

  1xx codici *Provisional*: si riferiscono a operazioni in corso (ad es. `100 Trying`, `180 Ringing`);

  2xx codici *Success*: sono dei codici di successo (ad es. `200 OK`);

  4xx codici *Client Error*: sono dei codici di errore (ad es. `401 Unauthorized`).

Principali campi nell'intestazione SIP

L'intestazione SIP può contenere diversi campi, tra cui:

- campo `From`: contiene l'indirizzo SIP del terminare che vorrebbe iniziare la chiamata;
- campo `To`: contiene l'indirizzo SIP del terminale che il terminale chiamante vorrebbe contattare;
- campo `Contact`: è usato dal server SIP per specificare l'indirizzo del terminale chiamato, che può essere utilizzato dal terminale chiamante per contattare direttamente il terminale chiamato;
- campo `Via`: è usato per tenere traccia di tutti i server SIP attraverso cui deve passare il messaggio (ad es. outbound proxy server);
- campo `Record Routing`: specifica se tutti i messaggi SIP devono passare per il proxy, utile per il NAT traversal;
- campo `Subject`: contiene l'oggetto della connessione SIP;
- campi `Content-Type`, `Content-Length`, `Content-Encoding`: specificano informazioni su tipo (in un formato di tipo MIME, ad es. SDP), lunghezza (in byte) e codifica del payload.

[6] Questo messaggio non va confuso con i pacchetti ACK di TCP: esso lavora a livello applicazione, quindi anche su UDP.

SDP

SDP (Session Description Protocol) è un protocollo basato su testo per descrivere le sessioni multimediali: numero di flussi multimediali, tipo di media (audio, video, ecc.), codec, protocollo di trasporto (ad es. RTP/UDP/IP), larghezza di banda, indirizzi e porte, tempi di inizio/fine di ogni flusso, identificazione della sorgente.

SDP viene incluso nel payload di un pacchetto SIP per notificare le informazioni di controllo sul flusso multimediale (per es. il messaggio SIP che trasporta un messaggio di invito a una chiamata telefonica ha anche bisogno di notificare quale codec utilizzare). Siccome SDP è stato progettato un po' di tempo fa, ha alcune funzionalità (come i tempi di inizio/fine di ogni flusso) che sono inutili per SIP, ma SDP è stato semplicemente adottato da SIP senza alcun cambiamento per riutilizzare il software esistente.

Formato dei messaggi SDP Ogni messaggio SDP è composto da una sezione di sessione e una o più sessioni media (una per ogni flusso multimediale):

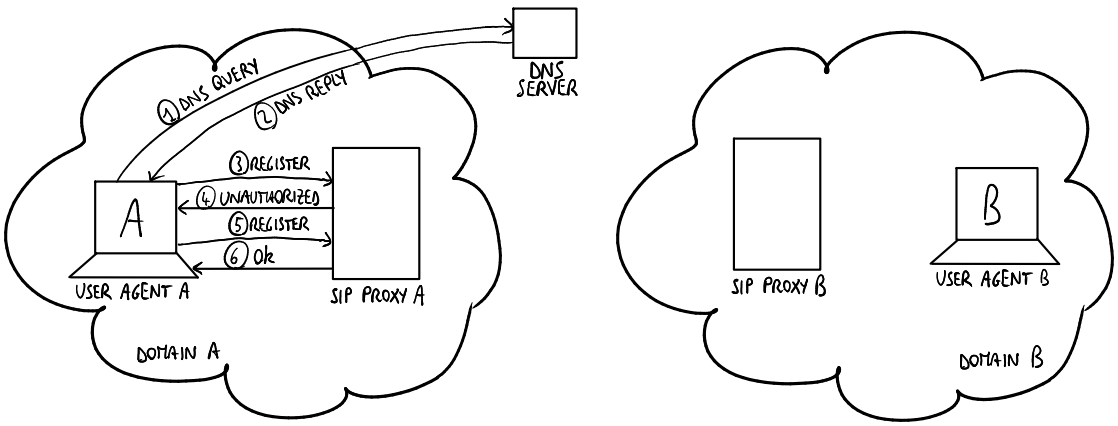
- **sezione di sessione**: iniziante con una riga `v=`, include i parametri per tutti i flussi multimediali nella sessione corrente;
- **sezione media**: iniziante con una riga `m=`, include i parametri per il flusso multimediale corrente.

6.6.5 Fasi di una chiamata SIP

Una chiamata SIP avviene in 4 fasi:

1. registrazione: il terminale chiamante si registra a un dominio;
2. invito: il terminale chiamante chiede di instaurare una chiamata;
3. trasferimento dati: la voce è trasportata da RTP;
4. abbattimento: la chiamata viene chiusa.

Fase di registrazione



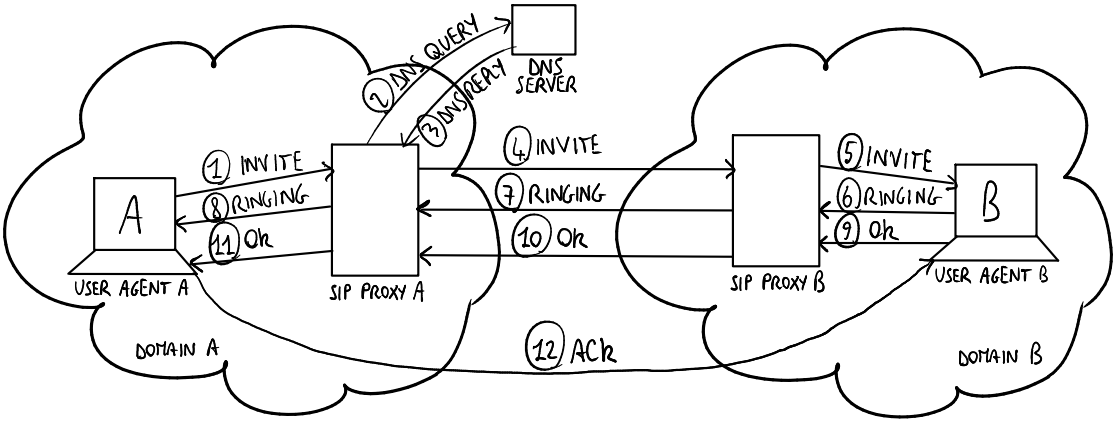
L'user agent A vuole registrarsi al dominio A contattandone il proxy SIP:[7]

1. 2. query e reply DNS (NAPTR, SRV, A/AAA): A chiede al server DNS l'indirizzo IP del proxy SIP;

[7] Qui il registrar server si suppone implementato nel proxy SIP.

3. messaggio `REGISTER`: A chiede al proxy SIP di essere registrato, senza inserire la sua password qui;
4. messaggio `401 Unauthorized`: il proxy SIP chiede l'autenticazione inserendo un **challenge**, che viene cambiato a ogni registrazione;
5. messaggio `REGISTER`: A calcola una funzione hash in base al challenge e alla password e invia la stringa risultante al proxy SIP;
6. messaggio `200 OK`: il registrar server verifica la risposta al challenge e garantisce l'accesso all'utente.

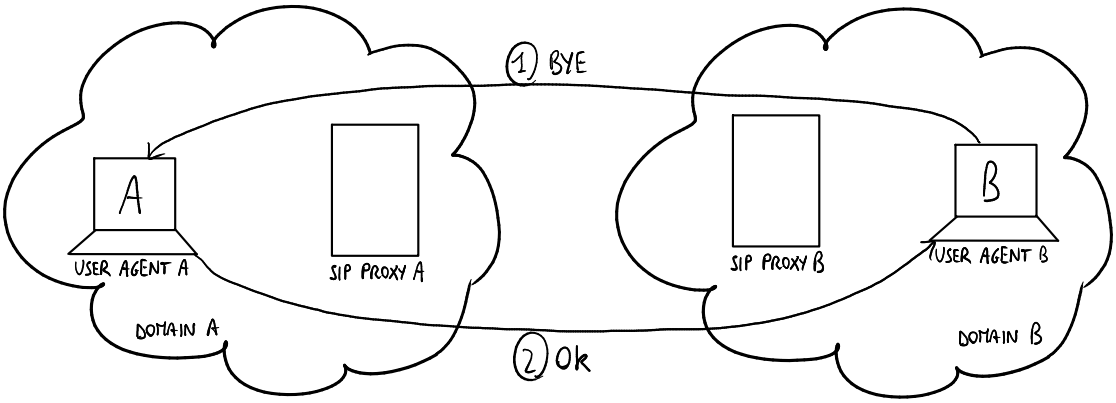
Fase di invito



L'user agent A vuole instaurare una chiamata con l'user agent B attraverso il proxy SIP di B:

1. A chiede al suo proxy SIP di contattare B inviando ad esso un messaggio `INVITE`;
2. 3. il proxy SIP di A effettua le query DNS per trovare l'indirizzo IP del proxy SIP di B (NAPTR, SRV, A/AAA);
4. il proxy SIP di A invia un messaggio `INVITE` al proxy SIP di B.
5. il proxy SIP di B invia un messaggio `INVITE` a B;
6. 7. 8. B fa squillare il telefono di A inviando, attraverso i proxy SIP, un messaggio `RINGING` ad A;
9. 10. 11. B accetta la chiamata inviando, attraverso i proxy SIP, un messaggio `OK` ad A;
12. A, attraverso i proxy SIP oppure direttamente a seconda del valore del campo `Record Routing`, notifica B che ha ricevuto il messaggio `OK`.

Fase di abbattimento



Alla fine della chiamata, dopo aver chiuso il flusso RTP:

1. messaggio `BYE`: B notifica A che vuole chiudere la chiamata;
2. messaggio `OK`: A notifica B che ha ricevuto il messaggio `BYE`.

Capitolo 7

Qualità del servizio

La **qualità del servizio** è l'insieme delle tecnologie per tentare[1] di garantire specifici requisiti sul ritardo e sul jitter[2] dei pacchetti per le applicazioni multimediali di rete che generano del traffico anelastico.

Approcci principali

Sono stati proposti tre approcci per la qualità del servizio:

- servizi integrati (IntServ): richiede cambiamenti fondamentali all'infrastruttura di rete in modo che l'applicazione possa riservare banda end-to-end ⇒ nuovo software complesso negli host e nei router (vedere la sezione 7.3);
- servizi differenziati (DiffServ): richiede meno cambiamenti all'infrastruttura di rete (vedere la sezione 7.4);
- laissez-faire: chi se ne importa dei ritardi e della qualità del servizio, la rete non sarà mai congestionata ⇒ tutta la complessità a livello applicazione.

7.1 Principi

1. Marcatura dei pacchetti necessaria per il router per **distinguere** tra classi diverse; e nuova politica del router per trattare i pacchetti di conseguenza.
2. Fornire protezione (**isolamento**) per una classe dalle altre.
3. Mentre si fornisce isolamento, si desidera utilizzare le risorse in modo il più **efficiente** possibile.
4. Il flusso dichiara le proprie necessità con una call admission, quindi la rete può bloccare la chiamata (ad es. segnale occupato) se non è in grado di soddisfare le necessità.

7.2 Meccanismi

7.2.1 Meccanismi di scheduling dei pacchetti

L'obiettivo dei meccanismi di scheduling è gestire le priorità dei pacchetti in arrivo.

[1] La qualità del servizio si limita a tentare di fare così, perché è impossibile garantire un servizio a commutazione di circuito su una rete a commutazione di pacchetto.

[2] Il **jitter** è la variabilità dei ritardi dei pacchetti nello stesso flusso di pacchetti.

Scheduling FIFO È facile da implementare ed efficiente solo se c'è una politica di scarto sofisticata:

- drop della coda: scarta sempre il pacchetto in arrivo;
- casuale: scarta un pacchetto a caso nella coda;
- priorità: scarta il pacchetto con la classe a più bassa priorità.

Scheduling a priorità Per ogni classe è disponibile un buffer, e viene servito sempre il pacchetto con la classe a più alta priorità.
Non garantisce l'isolamento e può introdurre starvation: i pacchetti nei buffer a bassa priorità non vengono mai serviti perché continuano ad arrivare pacchetti ad alta priorità. Inoltre, se temporaneamente la coda ad alta priorità è vuota permettendo di iniziare la trasmissione di un pacchetto nella coda a bassa priorità, ma arriva un pacchetto ad alta priorità subito dopo l'inizio della trasmissione, quest'ultimo dovrà aspettare che la trasmissione finisca, soprattutto se il pacchetto è lungo ⇒ vengono introdotti dei ritardi di trasmissione.

Scheduling round robin Effettua ciclicamente la scansione delle code delle classi, servendone una per ogni classe (se disponibile).
Garantisce l'isolamento ed è equa, ma non garantisce la priorità.

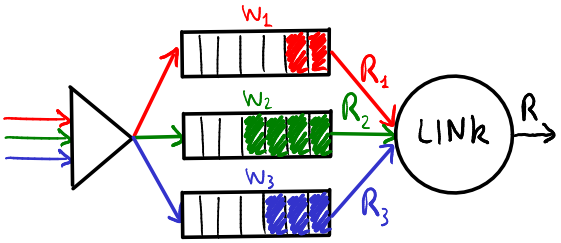


Figura 7.1: Esempio di weighted fair queuing.

Weighted fair queuing Generalizza il round robin combinandolo con lo scheduling a priorità. Ogni classe riceve una quantità pesata del servizio in ogni ciclo, e la larghezza di banda R_i della classe i avente peso w_i è data dalla formula seguente (le code vuote hanno peso nullo):

$$R_i = \frac{w_i}{\sum_j w_j} R_{tot}$$

Tuttavia questa soluzione non è molto scalabile perché la formula, che coinvolge operazioni in virgola mobile, deve essere calcolata per ogni singolo pacchetto.

7.2.2 Meccanismi di policing

L'obiettivo dei meccanismi di policing è limitare il traffico in modo che non superi i parametri dichiarati, come:

- frequenza media (a lungo termine): quanti pacchetti si possono inviare per unità di tempo;
- frequenza di picco: misurata in pacchetti al minuto;
- burst size (massimo): massimo numero di pacchetti inviati consecutivamente (senza inattività intervenienti).

Il **secchiello a gettoni** è la tecnica usata per limitare l'ingresso ad uno specifico burst size e ad una specifica frequenza media:

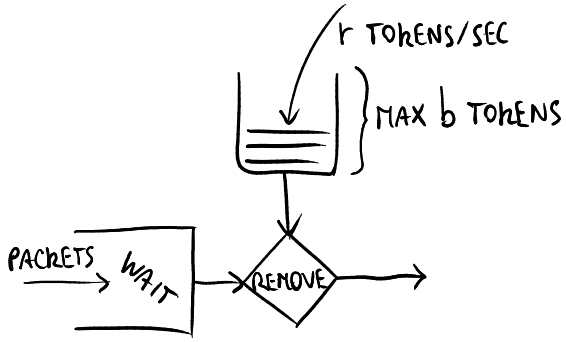


Figura 7.2: Secchiello a gettoni.

- un secchiello può contenere b gettoni;
- i gettoni vengono generati alla frequenza r gettoni/s, a meno che il secchiello non sia pieno;
- su un intervallo di lunghezza t, il numero di pacchetti ammesso è minore o uguale a $(rt+b)$.

7.3 IntServ

Riservamento delle risorse Fondamentalmente un host chiede un servizio che richiede alcune risorse (**messaggio di path**): se la rete è in grado di fornire questo servizio servirà l'utente, altrimenti non lo servirà (**messaggio di reservation**).
Il riservamento delle risorse è una funzionalità che non è nativa in IP.

Call admission La sessione in arrivo utilizza il protocollo di segnalazione **Resource Reservation Protocol** (RSVP) per dichiarare:

- **R-spec**: specifica la qualità del servizio richiesta;
- **T-spec**: definisce le caratteristiche del traffico.

Il ricevitore, non il mittente, specifica il riservamento delle risorse.
Questa è senza dubbio una soluzione non scalabile. Ha ancora dei problemi importanti, e attualmente non c'è alcuna ragione per implementare meglio IntServ.

7.4 DiffServ

Differentiated Services (DiffServ) è un'architettura proposta da IETF per la qualità del servizio: sposta la complessità (secchielli e buffer) dal core della rete agli edge router (o host) $\Rightarrow$ più scalabilità.

7.4.1 Architettura

L'architettura DiffServ è costituita da due componenti principali che effettuano al gestione del traffico per flusso:

- edge router: contrassegnano i pacchetti come **in-profile** (ad alta priorità) o **out-profile** (a bassa priorità);
- core router: effettuano il buffering e lo scheduling in base alla marcatura eseguita ai margini, dando preferenza ai pacchetti *in-profile*.

7.4.2 Marcatura

La marcatura viene eseguita dagli edge router nel campo **Differentiated Service Code Point** (DSCP), che sta nei 6 bit più significativi del campo "Type of Service" nell'intestazione IPv4 e del campo "Priority" nell'intestazione IPv6.

Sarebbe meglio lasciare che la sorgente, a livello applicazione, effettui la marcatura perché solo la sorgente conosce esattamente il tipo di traffico (traffico voce o traffico dati), ma quasi tutti gli utenti dichiarerebbero tutti i pacchetti come ad alta priorità perché non sarebbero onesti ⇒ la marcatura va eseguita dai gateway che sono sotto il controllo del provider. Tuttavia alcuni studi hanno rilevato che i router sono in grado di riconoscere correttamente al massimo il 20-30% del traffico, a causa per esempio del traffico criptato ⇒ si può semplificare la distinzione per i router connettendo il PC a una porta e il telefono a un'altra porta, così il router può marcare il traffico in base alla porta di ingresso.

7.4.3 PHB

Sono stati sviluppati alcuni **Per Hop Behaviour** (PHB):

- **expedited forwarding**: il tasso di partenza dei pacchetti di una classe eguaglia o supera un tasso specificato;
- **assured forwarding**: quattro classi di traffico: ognuna garantisce una minima quantità di larghezza di banda e tre partizioni di preferenze di drop.

I PHB specificano i servizi da offrire, non come implementarli.

Capitolo 8

Cenni di sicurezza e crittografia

8.1 Obiettivi di base e applicazioni della crittografia

Nel contesto delle reti, gli obiettivi di base dei meccanismi di sicurezza sono:

- **autenticazione degli endpoint**;
- **integrità dei dati**: si vuole essere sicuri che i dati non siano stati modificati nel percorso dalla sorgente alla destinazione;
- **confidenzialità**: si vuole garantire che i dati non siano letti da nessuno che non sia la destinazione prefissata.

Si ottengono questi obiettivi attraverso l'uso di meccanismi di crittografia. Questa viene usata in due diversi contesti, cioè per due azioni diverse:

- **criptazione**: consiste nel cifrare i dati scambiati, cioè cambiare il contenuto dei pacchetti in maniera che solo chi è stato autorizzato possa ricostruire il contenuto originale;
- **firma**: serve per garantire l'integrità dei dati e l'autenticazione del mittente, e si effettua aggiungendo in fondo al messaggio una piccola sequenza di byte, che dipende dai dati stessi e da alcune informazioni che ha il mittente:
  - gli endpoint possono accorgersi se i dati sono stati modificati, ricalcolando questa seguenza di byte;
  - il funzionamento è simile a quello di un codice di rilevamento d'errore, ma diversamente da esso la sequenza di byte apposta è basata su una **chiave** segreta.

8.2 Tipi di chiavi

Si distinguono due tipi di chiavi:

- **chiave simmetrica** (o **condivisa**): una stessa chiave, che è una sequenza di byte, viene usata sia per criptare/firmare, sia per decriptare/autenticare i dati. La chiave deve essere tenuta segreta tra le stazioni che comunicano, e questo rappresenta una difficoltà perché la comunicazione messa in atto per negoziare la chiave dovrebbe essere essa stessa sicura;
- **chiave asimmetrica**: due chiavi diverse vengono usate per criptare/firmare e decriptare/autenticare i dati. Una delle due, quella detta *chiave pubblica*, serve per decriptare i pacchetti che l'host sorgente ha criptato tramite la propria *chiave privata*, e può essere condivisa senza preoccupazione; l'altra deve essere tenuta segreta. Le due chiavi sono tali che ciò che viene criptato con una delle due può essere decriptato solo con l'altra.

  - Se, ad esempio, si vuole inviare un file in maniera sicura, è possibile applicare un algoritmo di crittografia usando la chiave privata e diffondere quella pubblica: in questo modo chi vuole comunicare in modo sicuro con quell'host può usare quella chiave per cifrare il messaggio, poiché solo quell'host avrà la chiave privata e potrà decodificarlo.
  - Per verificare l'identità del mittente, il meccanismo è analogo: se si vuol fare in modo che degli utenti possano verificare che un messaggio proviene da un certo mittente, basterà che usino la chiave pubblica del mittente per decifrare i messaggi da esso inviati.

8.2.1 Vantaggi e svantaggi dei tipi di chiave

- Le chiavi asimmetriche sono meno robuste e richiedono più risorse computazionali per l'algoritmo che le impiega rispetto alle chiavi simmetriche.
- In molti casi, si usano le chiavi asimmetriche per comunicare in maniera sicura e concordare una chiave simmetrica:
  - un host invia la chiave pubblica che intende usare per quella comunicazione unidirezionale con l'host destinazione;
  - l'host destinazione sceglie una chiave simmetrica e la inoltra criptandola con la chiave pubblica prima ricevuta;
  - l'host sorgente decripta il messaggio utilizzando la chiave privata, e da quel momento userà la chiave simmetrica contenuta.

Punto chiave: quando qualcuno riceve la chiave pubblica relativa ad una entità, esso deve essere sicuro dell'identità dell'entità, cioè che essa sia veramente chi dice di essere: a questo scopi si usano i **certificati**.

8.3 Certificati

Sono dei documenti che consentono di verificare l'appartenenza di una chiave pubblica ad una entità. Un certificato digitale contiene:

- informazioni sulla chiave;
- informazioni sull'identità dei proprietari;
- la firma digitale di un'entità che ha verificato i contenuti del certificato.

La firma non è altro che una sequenza di byte, una sorta di *digest*, cifrato con la chiave privata della certification authority. Verificare la firma significa controllare che il certificato sia stato validato dalla certification autority, quindi basta usare il suo certificato, che conterrà la chiave pubblica, per decifrare la firma. Il certificato dell'entità certificante potrebbe essere già noto all'host, ad esempio perchè è già presente nel sistema operativo o nel web browser, o potrebbe dover essere scaricato e a sua volta verificato nello stesso modo. Il certificato della **Root CA** deve essere ottenuto inevitabilmente in maniera affidabile.