UNIVERSITÀ DEGLI STUDI DI PAVIA
FACOLTÀ DI INGEGNERIA
Corso di Laurea in Ingegneria Informatica
Sede di Mantova

Connettività IPv4-IPv6: costruzione di un nodo traduttore di protocolli

Relatore:
Prof. GIUSEPPE FEDERICO ROSSI

Tesi di Laurea di:
EMANUELE GOLDONI
Matricola: 291575

Anno Accademico 2004-2005

Indice

Introduzione

Una rete di telecomunicazioni è definibile come un insieme di nodi, canali trasmissivi e procedure mediante le quali due o più dispositivi possono scambiarsi delle informazioni; in particolare le entità comunicano utilizzando un protocollo, ovvero un insieme di regole che definiscono il formato, la sintassi e la semantica dei messaggi scambiati. La rete oggi più estesa e conosciuta è sicuramente Internet, con quasi 400 milioni di nodi connessi, ed il protocollo su cui si basa è IP, Internet Protocol.

Esistono diverse versioni di IP, aventi in comune le caratteristiche di base pur restando comunque protocolli distinti. In particolare la versione 4 di Internet Protocol è quella attualmente utilizzata su Internet e nella maggior parte delle reti locali mentre Internet Protocol versione 6, inizialmente sviluppato con il nome di IPng (IP next generation), è stato formalizzato negli anni '90 e si sta affiancando progressivamente ad IPv4 con l'obiettivo di sostituirlo. Sicuramente IPv6 supera numerose limitazioni del suo predecessore e molti suoi aspetti sono stati progettati proprio alla luce delle problematiche emerse negli ultimi 20 anni di intenso utilizzo di IPv4; purtroppo però IPv4 e IPv6 non sono perfettamente compatibili e, pertanto, programmi e sistemi progettati per uno standard non possono comunicare direttamente con quelli pensati per l'altro. Inoltre la diffusione capillare e radicata dei sistemi basati su IPv4 non consente al giorno d'oggi di realizzare una transizione rapida dal vecchio al nuovo protocollo, rendendo di conseguenza necessaria l'adozione di meccanismi e strategie in grado di permettere alle applicazioni di continuare a funzionare correttamente durante tutta la lunga fase di 'aggiornamento' della rete mondiale.

In particolare in questo elaborato verrà approfondito NAT-PT, un meccanismo che traduce i dati in transito da una rete IPv6 ad una rete IPv4 o viceversa, e ne verranno presentate le implementazioni software esistenti nonché i problemi che può comportare l'utilizzo di questa particolare strategia di 'migrazione' da IPv4 a IPv6.

Capitolo 1

Architetture di comunicazione

Internet è senza dubbio un sistema estremamente complicato e caratterizzato da un'elevata eterogeneità di componenti che dialogano tra loro: applicazioni e protocolli, terminali e router connessi dai più disparati mezzi trasmissivi fisici, dalla fibra ottica alle onde radio passando per il doppino di rame. Data l'enorme complessità, è evidente come un approccio di tipo "monolitico", con la costruzione di un unico componente in grado di fornire tutte le funzionalità necessarie, non possa essere affatto efficiente in termine di semplicità di manutenzione, efficienza e versatilità.

La strategia adottata per l'implementazione delle funzionalità necessarie in una rete a commutazione di pacchetto è stata quindi quella della suddivisione dell'architettura di comunicazione in più strati, o livelli. Quello che si vuole ottenere è una struttura stratificata che costruisce un servizio comunicativo sofisticato operando 'per gradi', dove ciascuno degli N livelli che la compongono è incaricato di svolgere un ben preciso insieme di controlli, per implementare funzionalità via via sempre più complesse risalendo lungo la pila.

L'invio di un messaggio originato dall'ultimo livello comporta la 'discesa' lungo lo stack architetturale e, ad ogni attraversamento di un livello, l'aggiunta (incapsulamento) da parte del livello stesso di informazioni di controllo in testa o in coda al messaggio; la nuova struttura dati che si viene così a creare costituisce il pacchetto che verrà poi trasmesso sui canali. In maniera del tutto analoga, la ricezione di un pacchetto comporta la risalita dello stack con l'eliminazione dei relativi incapsulamenti e la consegna infine del messaggio contenuto al destinatario.

Questa brillante soluzione si basa su due semplici principi alla base dell'ingegneria e della progettazione in generale: scomporre un problema complesso in più sotto-problemi semplici e separare l'implementazione dall'interfaccia, ovvero distinguere il cosa deve essere fatto dal come.

Il modello a strati ha spinto negli anni 70 tutti i principali costruttori a formalizzare delle architetture di comunicazione proprietarie; tra queste vanno sicuramente ricordate SNA (System Network Architecture) di IBM, DNA (Digital Network Architecture) di Digital, AppleTalk di Apple, Novell Netware e il meno fortunato Xerox Network Service. Parallelamente allo sviluppo di architetture di comunicazione proprietarie, si sono avuti importanti progetti per la definizione di architetture non proprietarie, definite invece da organismi internazionali non legati a specifici costruttori: OSI, DoD e TCP/IP. Il modello OSI, Open Systems Inteconnection, è un'architettura comunicativa formalizzata dall'ISO (International Standard Organization) che prevede 7 distinti livelli (Physical, Data Link Control, Network, Transport, Session, Presentation e Application) ma che non ha mai avuto successo; tuttavia ancora oggi questa architettura viene presa come modello di riferimento nello studio delle altre architetture.

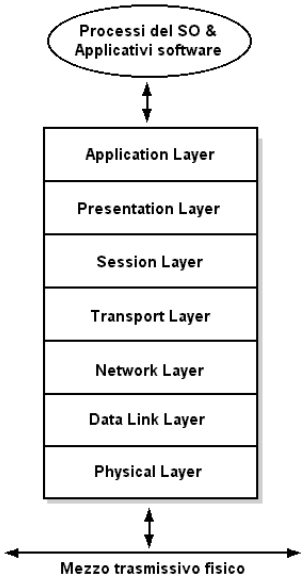

Figura 1.1 Il modello ISO/OSI

Lo stack protocollare DoD del Dipartimento della Difesa americano, sviluppato negli anni 70 per la rete militare finanziata da DARPA (Defence Advanced Research Projects Agency), è invece un modello a 4 livelli basato su un approccio pragmatico e molto più vicino alle reali esigenze di networking rispetto al modello OSI (non è un caso che proprio DoD sia stato grande fonte di ispirazione per i progettisti di altri protocolli).

L'architettura di comunicazione che però si è progressivamente affermata e infine imposta come standard de facto è però quella nota come Suite TCP/IP. Il suo successo è sicuramente legato alla sua natura non proprietaria, che svincola l'infrastruttura di rete da specifiche tecnologie proprietarie, e alla sua essenzialità, a volte persino esasperata ma tale da rendere questo protocollo estremamente flessibile ed è in grado di interconnettere reti basate su tecnologie eterogenee. Inoltre occorre considerare come TCP/IP sia riuscito, a differenza di molti altre suite, a guadagnare la fiducia dell'ambiente accademico, venendo adottato nel 1983 dal progetto ARPAnet (la rete del DARPA che collegava tra loro tutti i maggiori centri di ricerca nazionali) in sostituzione del vetusto NCP e implementato quindi nello Unix di Berkley (BSD4.3), distribuito in tutto il mondo.

Capitolo 2

La suite TCP/IPv4

Le specifiche dei protocolli della suite TCP/IPv4, le cui idee alla base furono per la prima volta pubblicate in un articolo nel 1974 (V. Cerf e R. Kahn, "A protocol for packet network interconnection"), partono dal 3° livello architetturale rispetto al modello di riferimento OSI. In particolare il protocollo IP, che si inserisce a livello Network, consente la costruzione di una rete a commutazione di pacchetto che in linea di principio può utilizzare qualunque DLC mentre a livello transport operano sia TCP che UDP, fornendo tipologie di servizi 'duali'; il primo infatti è affidabile ma sofisticato, il secondo è invece più snello ma inaffidabile. Esistono quindi moltissimi altri protocolli di livello applicativo che si appoggiano ai servizi forniti dai sottostanti TCP e UDP e che permettono di interconnettere diversi processi applicativi (Ftp, Http, Telnet, ssh…).

2.1 IPv4

Internet Protocol v4 (RFC791) appartiene alla famiglia dei protocolli di tipo connectionless non confermati e svolge solamente le funzioni di frammentazione e protocol multiplexing delle PDU provenienti dai livelli superiori, non introducendo invece meccanismi per il controllo di flusso, la consegna in sequenza o il controllo degli errori. Potrebbe sembrare paradossale il fatto che l'intera rete mondiale sia basata su un protocollo così semplice e inaffidabile; in realtà è proprio questa assenza di complessità che permette ad IP di gestire un traffico di una rete così vasta, delegando eventuali controlli o funzionalità aggiuntive ai livelli superiori. La struttura della PDU-DATI (datagram) del protocollo IPv4 è composta da 14 diversi campi, di cui 12 aventi lunghezza fissa, secondo lo schema sotto riportato.

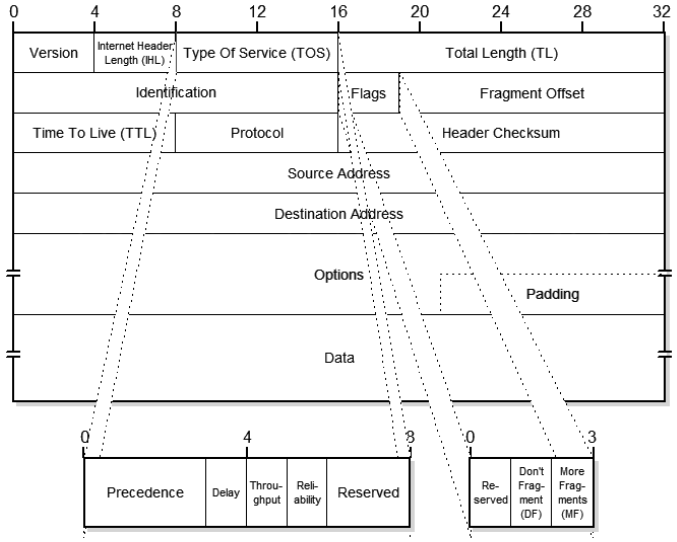

Figura 2.1 Formato del datagram IPv4

IPv4 introduce un proprio sistema di indirizzamento a priori, indipendente dall'indirizzamento di livello DLC, prevedendo indirizzi di lunghezza fissa pari a 32 bit espressi con notazione 'dotted decimal' (si esprime cioè il valore decimale di ogni byte usando i punti come separatori, es. 193.206.71.152) di tre diversi tipi: unicast, multicast e broadcast.

Gli indirizzi unicast IPv4 non sono di tipo flat bensì sono strutturati su almeno due campi: Network address e Host address. Questi indirizzi vengono associati alle singole interfacce dei nodi di rete (e non al nodo) e devono rispettare due regole principali: interfacce adiacenti, ovvero sullo stesso link, devono avere stesso lo stesso Network address e diversi valori del campo Host address, mentre interfacce non adiacenti devono avere diversi Network address. Un indirizzo di multicast invece ha una struttura flat, cioè è a campo unico, e identifica un gruppo di nodi: un pacchetto inviato verso un indirizzo IPv4 multicast viene cioè ricevuto da tutti i nodi che in quel momento appartengono a quel gruppo di multicast. Un messaggio inviato verso un indirizzo di tipo broadcast deve essere recapitato a tutti i nodi IPv4 adiacenti (ovviamente non tutta Internet!). L'Internet Protocol v4 si appoggia poi a due protocolli di 'servizio' rispettivamente per effettuare la risoluzione degli indirizzi e per la segnalazione di errori e messaggi di controllo tra macchine TCP/IP.

La trasmissione di datagram tra nodi IPv4 appartenenti allo stesso link presenta un problema ricorrente nel caso in cui il DLC sottostante utilizzi un sistema di indirizzamento: è infatti necessario determinare a quale indirizzo di livello DLC inviare il datagram, 'risolvere' cioè l'indirizzo di livello 2 conoscendo l'indirizzo di livello 3 corrispondente. Le tecniche di risoluzione dell'indirizzo sono varie e differiscono in base al tipo di link utilizzato; in particolare si distinguono le tre fondamentali tecniche di risoluzione statica (esiste una tabella definita a priori di corrispondenze tra indirizzi IPv4 e DLC), dinamica (automatismi ricavano all'occorrenza la corrispondenza indirizzo IPv4 – DLC) e algoritmica (l'indirizzo di livello 2 è ricavato a partire dall'indirizzo di livello 3).

IPv4 dispone di ARP (Address Resolution Protocol), un protocollo assistente che svolge la funzione di costruzione dinamica delle associazione tra indirizzi IPv4 e DLC. ARP è incluso nell'implementazione di IPv4 e avrebbe poco senso se non utilizzato in combinazione con l'Internet Protocol; tuttavia i due sono concettualmente due protocolli completamenti separati e le richieste ARP non sono incapsulate in datagram IPv4, bensì sono messaggi separati incapsulati direttamente nelle frame DLC.

ICMP (Internet Control Message Protocol) è invece un protocollo di servizio, pensato ad hoc per IPv4, utilizzato per effettuare "error reporting" verso la sorgente dati. A differenza di quanto accade per ARP, i messaggi ICMP vengono incapsulati in IP e a ciascun messaggio è associato un primo codice che ne indica la tipologia e un sotto-codice che specifica il particolare errore

riscontrato. Alcuni semplici comandi (es. ping, traceroute) utilizzati per verificare il funzionamento della rete utilizzano proprio ICMP, inviando particolari pacchetti e verificando quali messaggi ICMP vengono rispediti in risposta.

2.2 TCP/UDP per IPv4

La suite TCP/IP definisce due protocolli di livello Transport sotto molti punti di vista 'duali': TCP (Transmission Control Protocol) e UDP (User Datagram Protocol). Ciò permette di fornire agli applicativi modalità comunicative su cui appoggiarsi radicalmente diverse, in modo tale che questi possano utilizzare quello migliore in funzione del tipo di servizio che devono offrire all'utente finale. La rete di router intermedi, aventi la sola funzione di inoltrare il traffico dal nodo mittente verso il nodo destinatario, è costituita da dispositivi di interconnessione di livello 3 e pertanto, per definizione, non possiede livelli Transport o superiori; questi invece saranno presenti sui nodi finali, sui quali risiedono anche le applicazioni sorgenti e destinatarie del traffico. Sarà quindi compito del nodo destinario effettuare la multiplazione dei diversi flussi comunicativi dei livelli superiori. A questo proposito è opportuno ricordare che, poiché non esistono in TCP/IP indirizzi di livello Transport, un layer TCP o UDP viene identificato solamente tramite l'indirizzo del sottostante livello IP sul nodo e il tipo di protocollo stesso, specificato dal valore del campo Protocol presente nella header IP. Lo stessa problema si ripresenta per il livello applicativo; in questo caso ad ogni processo applicativo non viene associato un indirizzo bensì un numero di porta univoco, sul quale il processo stesso sarà in ascolto per catturare i flussi comunicativi a lui diretti.

UDP (RFC768) è un protocollo non confermato estremamente 'sottile', che fornisce cioè ai livelli sovrastanti servizi simili a quanto fornito dal IP (introducendo quindi un overhead computazionale decisamente contenuto): UDP opera infatti in modalità connectionless e non fornisce garanzia circa la consegna del messaggio, ovvero la comunicazione non è affidabile. Quello che viene fornito ai protocolli di livello applicativo è però un'interfaccia di programmazione applicativa, ovvero le meglio note Socket. Non esistendo in UDP il concetto di sessione comunicativa, quando l'applicazione mittente consegna a UDP un messaggio, questo provvede immediatamente a generare un datagram e ogni datagram generato nel tempo sarà indipendente dagli altri (protocollo stateless). Un protocollo di questo tipo risulta molto conveniente in situazioni in cui le prestazioni hanno la precedenza sulla qualità: si pensi ad esempio ad uno streaming audio o video, dove un risultato fluido ma con qualche imperfezione è preferibile rispetto ad un immagine o audio perfetti ma non fruibili perché troppo lenti.

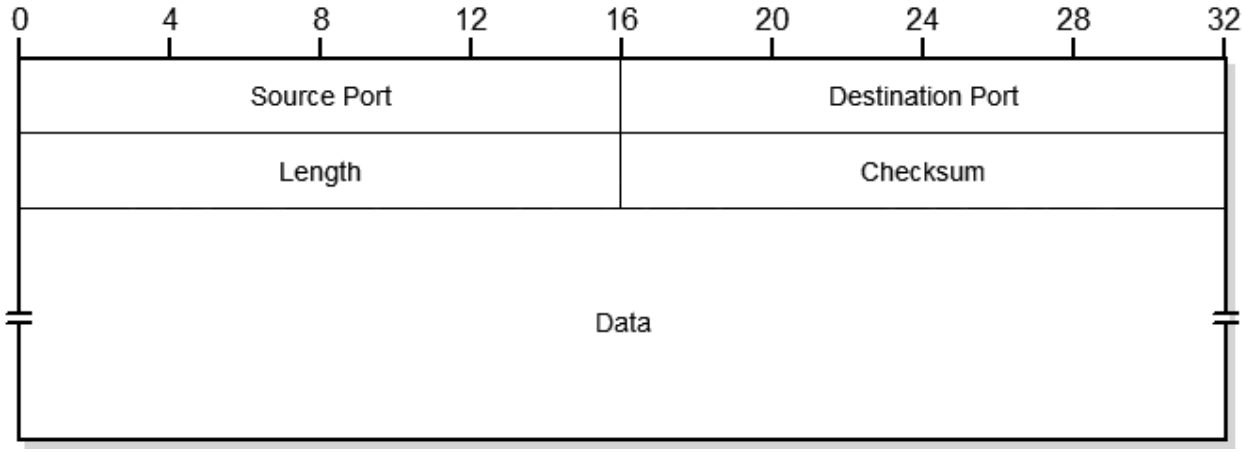

Figura 2.2 Formato del Datagram UDP

TCP (standardizzato per la prima volta con RFC761 e perfezionato quindi con RFC793) si appoggia su servizi di livello 3 ritenuti inaffidabili (come lo IPv4) ed ovvia quindi a queste mancanze implementando tutte le funzionalità necessarie per rendere la comunicazione affidabile. TCP è quindi, a differenza di UDP, un protocollo connection-oriented (stateful): esiste cioè il concetto di sessione comunicativa e il trasferimento di dati applicativi, la cui consegna in sequenza è garantita, può avvenire solo a patto che sia stata instaurata una connection TCP tra le due entità comunicanti e che questa sia attiva (non sia cioè ancora stata abbattuta). Inoltre è un protocollo confermato che adotta uno schema di controllo di flusso a finestra di messaggi ibrido (Selective-Repeat in condizioni normali, Go-Back-n in caso di errori) prevedendo PDU-ACK e time-out di ritrasmissione. A differenza di UDP, TCP non trasferisce singoli messaggi applicativi bensì sequenze di byte (byte stream service); il processo applicativo passa cioè al TCP uno o più messaggi che il TCP tratta come un'unica sequenza di byte, suddividendola quindi a propria discrezione in blocchi (aventi comunque una dimensione massima, detta Maximum Segment Size) e costruendo poi le PDU (Segment TCP) che verranno passate all'IP. Negli anni numerosi RFC sono stati inoltre rilasciati al fine di proporre nuovi e più efficienti algoritmi di funzionamento, specialmente in situazioni di congestione della rete o di perdita di dati trasmessi. Si può quindi intuire come il TCP sia un protocollo complesso ma affidabile, adatto in tutte quelle situazioni in cui le prestazioni non sono fondamentali ma lo è l'integrità dei dati ricevuti (es. trasferimento di file, e-mail…).

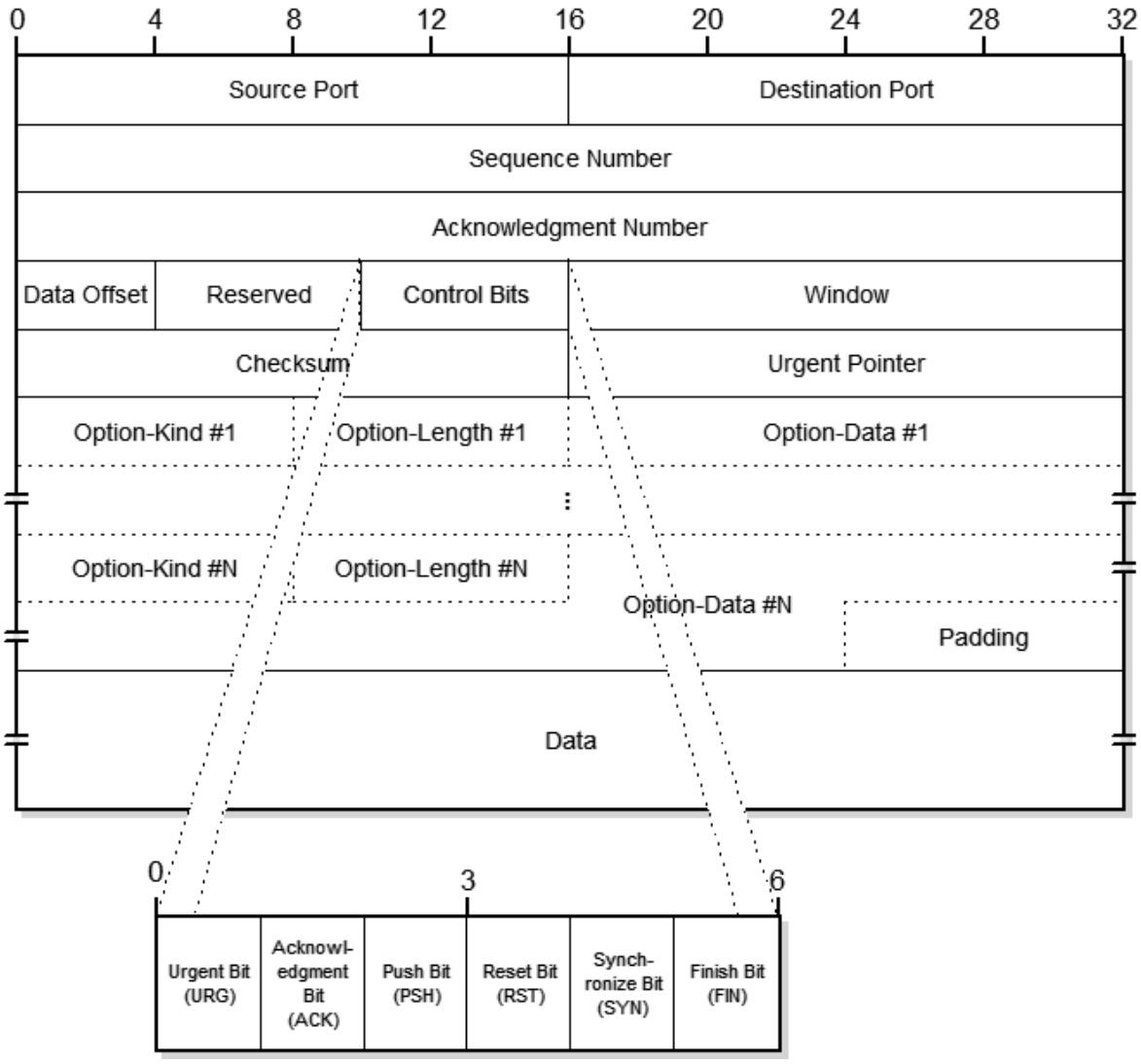

Figura 2.3 Formato del Segment TCP

2.3 Le limitazioni di questo protocollo

Seppur pensato oltre 20 anni fa, la lungimiranza dei progettisti ha permesso a questa suite di protocolli di crescere insieme ad Internet stesso e ancora oggi grazie al TCP/IPv4 milioni di dispositivi e programmi sempre nuovi ogni giorno possono comunicare tra loro da un capo all'altro del mondo. Tuttavia nel corso degli anni alcuni problemi sono venuti a galla e diverse estensioni sono state introdotte per permettere ad Internet di continuare a funzionare a dovere nonostante la crescita esponenziale dei nodi e, di conseguenza, del traffico. Oggi la tecnologia però si sta incamminando lungo nuove strade, impensabili solo pochi anni fa, e taluni limiti intrinseci di questa versione dell'Internet Protocol rappresentano oggi un forte ostacolo allo sviluppo tecnologico.

2.3.1 Limitazione dello spazio di indirizzamento

I 32 bit che compongono un indirizzo IPv4 consentono teoricamente un massimo di 4.294.967.296 indirizzi univoci; tuttavia molti di questi sono riservati per specifiche applicazioni (reti locali, comunicazioni multicast, indirizzi di loopback o per usi futuri) e non utilizzabili direttamente per connettere host ad Internet. Inoltre occorre ricordare come durante la fase iniziale di sviluppo di Internet ad alcune istituzioni o enti coinvolti nello sviluppo vennero allocati spazi di indirizzamento di dimensioni spropositate: per fare un esempio, il Massachusset Institute of Technology (MIT) dispone di un intero blocco di indirizzi '/8', che si traduce in oltre 16 milioni di indirizzi per una sola Università! Per ridurre la crescente domanda di indirizzi sono state introdotte nel tempo diverse misure tra cui CIDR, DHCP e NAT, nonché un controllo più capillare attraverso i Regional Internet Registries e la riassegnazione di grandi blocchi di indirizzi allocati agli albori di Internet ma non più utilizzati oggigiorno. Nonostante tutto si è però consapevoli che tutte queste misure sono soltanto dei palliativi e non possono certo risolvere il problema: la diffusione di host sempre connessi ad Internet attraverso la linea ADSL o via cavo e la proliferazione di dispositivi wireless mobili quali PC portatili, Tablet PC, palmari e cellulari hanno in questi ultimi anni dato un impulso notevole alla richiesta di indirizzi IP, specialmente in paesi asiatici emergenti e densamente popolati quali Cina e India. Secondo alcune stime tutti gli indirizzi disponibili verranno assegnati entro il 2013 circa (http://bgp.potaroo.net/ipv4/) ma è presumibile che già alcuni anni prima i costi subiranno un'impennata, diventando proibitivi per la maggior parte delle utenti.

2.3.2 Routing inefficiente

La rapida espansione di Internet a messo ha dura prova i router della dorsale; l'aumento delle dimensioni delle routing table (ad oggi circa 160000 regole) ha infatti comportato non pochi problemi in termini di prestazioni e di aggiornamento delle entry. La causa di questo è sicuramente da ricercare nella struttura degli indirizzo IPv4, originariamente pensato con una struttura gerarchica basata su classi poche di indirizzi, decisamente poco efficiente in termini di routing. Sicuramente l'aggregazione di più regole consentita da CIDR ha risolto in parte questo problema, ma gli errori compiuti in passato in fase di allocazione degli spazi e l'aggiunta di nuovi prefissi per una stessa organizzazione o ISP hanno introdotto e continuano ad introdurre una forte frammentazione ed un conseguente aumento delle regole.

2.3.3 Scarsa sicurezza

A lungo considerata un compito esclusivo dei livelli applicativi, la sicurezza è invece stata una delle carenze più avvertite del protocollo IPv4. L'RFC2401, Security Architecture for the Internet

Protocol, ha cercato di porre rimedio a questa situazione; IPsec rimane tuttavia una soluzione di ripiego e non una caratteristica nativa del protocollo sfruttabile a pieno (non è utilizzabile ad esempio in presenza di dispositivi NAT).

2.3.4 Mancanza di autoconfigurazione

La configurazione di nodi IPv4 non è mai stata semplice, dal momento che non sono stati previsti meccanismi nativi di autoconfigurazione; la massiccia diffusione di dispositivi mobili e la dimensione sempre crescente delle reti realizzate ha portato allo scoperto con prepotenza questo problema.

2.3.5 Mancanza di Flow Label

In IPv4 non è prevista la possibilità di gestire il routing dei flussi comunicativi. IPv4 è infatti un protocollo connectionless best-effort: non è cioè possibile utilizzare etichette per identificare particolari flussi comunicativi, garantendo ad esempio una Qualità del Servizio o un trattamento omogeneo per tutti i pacchetti che vi appartengono.

2.3.6 Scarso supporto per la mobilità

Come già detto poco sopra, la massiccia diffusione di dispositivi mobili ha messo in evidenza alcune gravi carenze di IPv4 quali la limitazione dello spazio di indirizzamento e la mancanza di veri meccanismi di autoconfigurazione. Tuttavia esistono altri problemi specifici del mobile computing, quali l'inoltrare i dati verso una postazione che continua a cambiare posizione; si pensi ad esempio al caso in cui un utente è collegato con il proprio portatile o palmare ad una internetwork utilizzando un link wireless e si sposta continuamente, cambiando quindi punto di connessione, mentre è in corso una sessione di lavoro (*nomadic computing*). Da qui la necessità di creare meccanismi *ad hoc* per "seguire" i dispositivi, ridirezionando di volta in volta il traffico verso il nuovo punto di connessione alla rete.

Capitolo 3

La suite TCP/IPv6

I limiti e i problemi di IPv4 non hanno purtroppo tardato ad emergere e le soluzioni proposte si sono rivelate funzionanti ma, sicuramente, non definitive. Per questo motivo la comunità di Internet già nei primi anni 90 ha deciso di realizzare un nuovo protocollo da zero piuttosto che continuare ad aggiungere "pezze" ad IPv4; diverse proposte avanzate da IETF e da molti altri gruppi di lavoro hanno portato quindi alla definizione di più protocolli sperimentali (IP Encaps, P Internet Protocol, Simple Internet Protocol…), che hanno contribuito in diversa misura alla formalizzazione definitiva della nuova versione del Protocollo di Internet, IPv6 (RFC2460). IPv6 fornisce estese funzionalità di rete: uno spazio di indirizzamento pressoché infinito, con conseguente eliminazione di soluzioni NAT-like e ripristino delle connessioni end-to-end per ogni host connesso ad Internet, migliori funzionalità di autoconfigurazione, supporto nativo del multicasting, estensibilità del protocollo stesso e routing più efficiente.

3.1 IPv6

Come più volte ricordato, IPv6 costituisce una evoluzione di IPv4 e ne mantiene pertanto le caratteristiche fondamentali. Anche IPv6 rimane quindi un protocollo di livello 3 connectionless e che garantisce un servizio di comunicazione inaffidabile, privo di ogni meccanismo di controllo di errore o di flusso e di servizio di consegna in sequenza. Il sistema di indirizzamento introdotto da questo protocollo è forse la novità più evidente a livello macroscopico: in IPv6 la lunghezza degli indirizzi è stata infatti fissata a 128 bit. Ogni indirizzo IPv6 è sempre associato ad un'interfaccia di un nodo e non al nodo stesso (come in IPv4) e ad una singola interfaccia è possibile associare più indirizzi. Per quel che riguarda la rappresentazione, l'indirizzo viene suddiviso in 8 porzioni da 16 bit e ciascuna di esse è rappresentata in esadecimale e separata dalle altre con il simbolo ":". Sempre in analogia con il suo predecessore, IPv6 fa pesantemente ricorso al concetto di "prefisso di indirizzo" e utilizza la notazione CIDR introdotta con IPv4 per indicare la lunghezza del prefisso, ovvero il numero di bit più significativi che costituiscono l'indirizzo IPv6 stesso (es. 3ffe:1ce1::/64 per indicare un prefisso lungo 64 bit).

Gli indirizzi IPv6 possono essere di tre diverse tipologie: unicast, multicast e anycast. Gli indirizzi unicast identificano una singola interfaccia e i pacchetti inviati ad un indirizzo di questo tipo sono consegnati all'interfaccia associata all'indirizzo stesso. In particolare gli indirizzi unicast si dividono a loro volta in indirizzi global, utilizzati nelle comunicazioni punto-a-punto, link-local, utilizzati per la trasmissione di un pacchetto lungo uno stesso segmento di rete (ovvero i router non inoltreranno pacchetti con indirizzi link-local unicast), e site-local, usati per limitare ad una

intranet l'area di validità dei pacchetti (ovvero i router di confine, che connettono la rete interna con quella esterna, non inoltreranno all'esterno della rete i pacchetti con indirizzi di tipo site-local unicast). Un pacchetto inviato invece ad un indirizzo multicast viene ricevuto da tutte le interfacce associate ad esso. Infine, un indirizzo anycast identifica infine una serie di interfacce solitamente presenti su nodi differenti e il pacchetto inviato ad un indirizzo anycast viene consegnato all'interfaccia associata a questo indirizzo che si trova più vicino al mittente; a differenza del caso multicast, quando un nodo riceve il messaggio la trasmissione è terminata e non è previsto che anche gli altri nodi lo ricevano.

Il datagram di IPv6 è stato semplificato rispetto ad IPv4, al fine di ottimizzare i tempi di forwarding del router, e le funzionalità di base sono state separate da quelle "opzionali". Il campo Header_Checksum è stato eliminato mentre sono stati inseriti campi per poter discriminare classi di traffico (attraverso il campo Traffic_Class) e per gestire in maniera differenziata i flussi comunicativi (identificati tramite il campo Flow_Label); inoltre non sono stati inseriti campi per la gestione della frammentazione, operazione che ora può essere fatta esclusivamente dal mittente, e per eventuali opzioni aggiuntive.

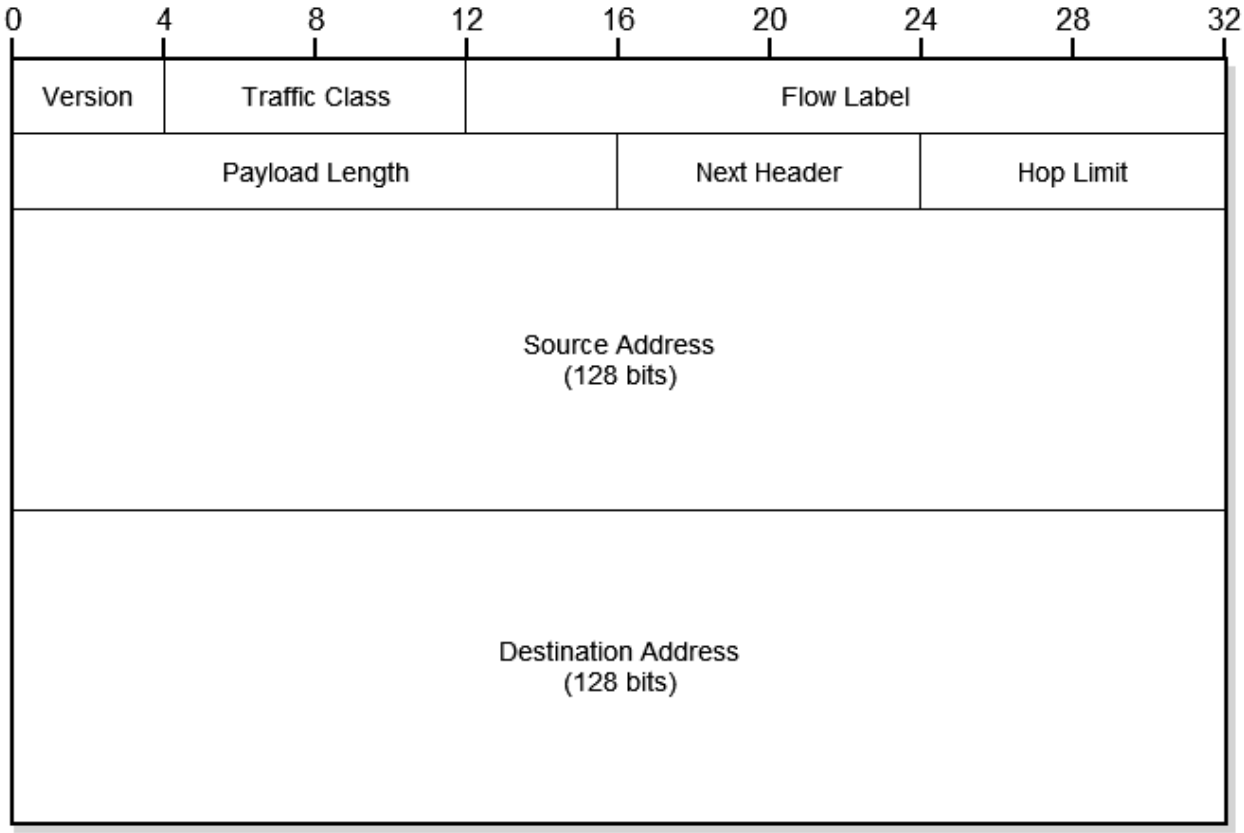

Figura 3.1 Formato della Basic Header IPv6

Per specificare tutte le altre opzioni viene utilizzato invece il meccanismo delle Extension Header, un sistema innovativo introdotto da IPv6 che permette di estendere di volta in volta la header di

base IPv6 in funzione delle esigenze. Nella sostanza l'header di base IPv6 utilizza il campo Next_Header per puntare all'header successiva (ed indicarne il tipo) e così via ogni Extension header utilizza un campo analogo, creando così una serie di header concatenate in sequenza e dove l'ultimo "frammento" di datagram puntato è l'Upper Layer Protocol. Ogni nodo deve elaborare le varie header nel medesimo ordine nel quale esse compaiono nel datagram IPv6; quelli intermedi solitamente devono elaborare la Basic Header e la Hop-by-hop Option Header (se presente). Tutte le altre header aggiuntive devono invece essere analizzate solo sul nodo (o sui nodi, se l'indirizzo è di tipo multicast) individuato dall'indirizzo indicato come destinatario del datagram nella Basic Header. L'ordine di costruzione delle header è stato pensato appositamente alla luce di questo fatto, suggerendo di far comparire nei primi posti le header da elaborare anche sui router intermedi e quindi quelle di interesse solamente per il destinatario finale. La raccomandazione di IETF prevede quindi che la Basic header sia, nell'ordine, seguita prima dalle testate aggiuntive Hop-by-Hop Options, Destinations Option e Routing Header e quindi dalle Extension headers Fragment, Autentication ed Encapsulating Security Payload; la PDU di livello superiore, come già detto prima, occupa infine sempre l'ultima posizione nel datagram IPv6.

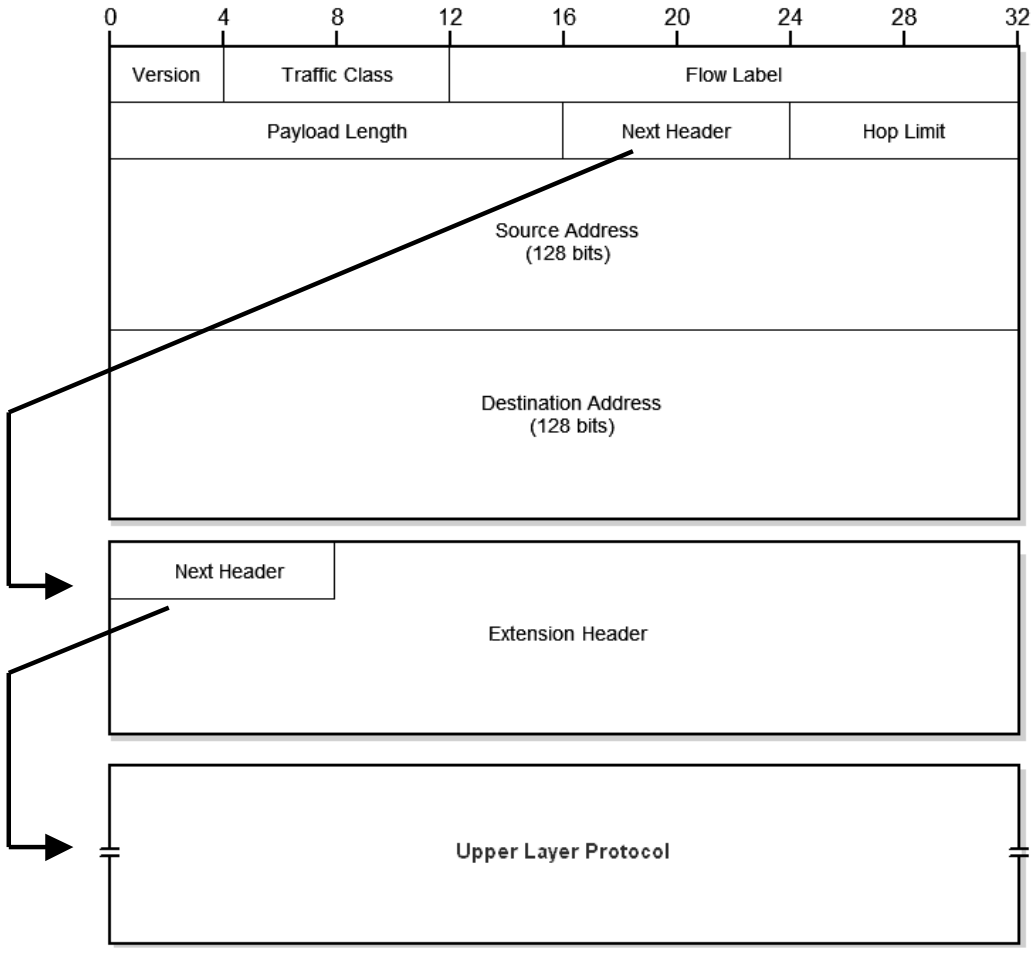

Figura 3.2 Schema di funzionamento delle Extension headers IPv6

Analogamente al caso IPv4, IPv6 viene assistito nelle sue funzioni dal protocollo di servizio ICMP (Internet Control Message Protocol). Tuttavia, a differenza del predecessore, ICMPv6 (RFC2463) non svolge solamente le funzioni basilari di ICMP bensì integra anche tutte le funzioni proprie di ARP e numerose altre introdotte appositamente per supportare nuove caratteristiche di IPv6. Tra le funzioni di questo protocollo, strettamente necessario per il funzionamento di IPv6 stesso, possiamo indicare le basilari funzioni di error reporting e diagnostica di reti, risoluzione degli indirizzi IPv6 in indirizzi di livello DLC, individuazione del router corretto attraverso cui uscire dalla rete, verifica degli indirizzi IPv6 assegnati (es. in caso di indirizzi duplicati), autoconfigurazione degli indirizzi IPv6 nonché calcolo della Path-MTU (ovvero la dimensione massima supportata dai nodi intermedi lungo l'intero tragitto dal mittente al destinatario). In analogia con ICMPv4, ICMPv6 viene trasportato tramite imbustamento in datagram IP e identifica i vari tipi di messaggi con un codice che ne identifica la tipologia ed uno che specifica il particolare messaggio.

3.2 TCP/UDP per IPv6

Le modifiche al protocollo IP, operativo al terzo livello dello stack architetturale, si riflettono sui livelli superiori dell'architettura per due diversi motivi. In primo luogo l'introduzione di nuovi campi (es. Flow_Label) e la nuova dimensione di altri (es. gli indirizzi) richiede modifiche sui livelli superiori che, attraverso nuove funzioni primitive, dovranno comunicare all'IPv6 i diversi valori per il datagram che verrà generato.

Inoltre alcune operazioni effettuate ai livelli superiori, teoricamente non lecite, utilizzano informazioni provenienti dal livello network. Con IPv6 ad esempio l'integrità dei dati trasmessi è garantita dai protocolli di livello superiore opportunamente modificati: il campo Checksum già presente in TCP e ICMP è stato reso obbligatorio anche per UDP ed è stata definita un'unica modalità di calcolo del checksum stesso per tutti e tre questi protocolli. L'algoritmo per la determinazione del checksum ricalca quello utilizzato per TCP con IPv4 (complemento a 1 della somma in complemento a 1 su 16 bit della struttura costruita concatenando Pseudo-Header, ULP-Header e ULP-Data), con una Pseudo-Header creata ad hoc e contenente proprio gli indirizzi di livello 3.

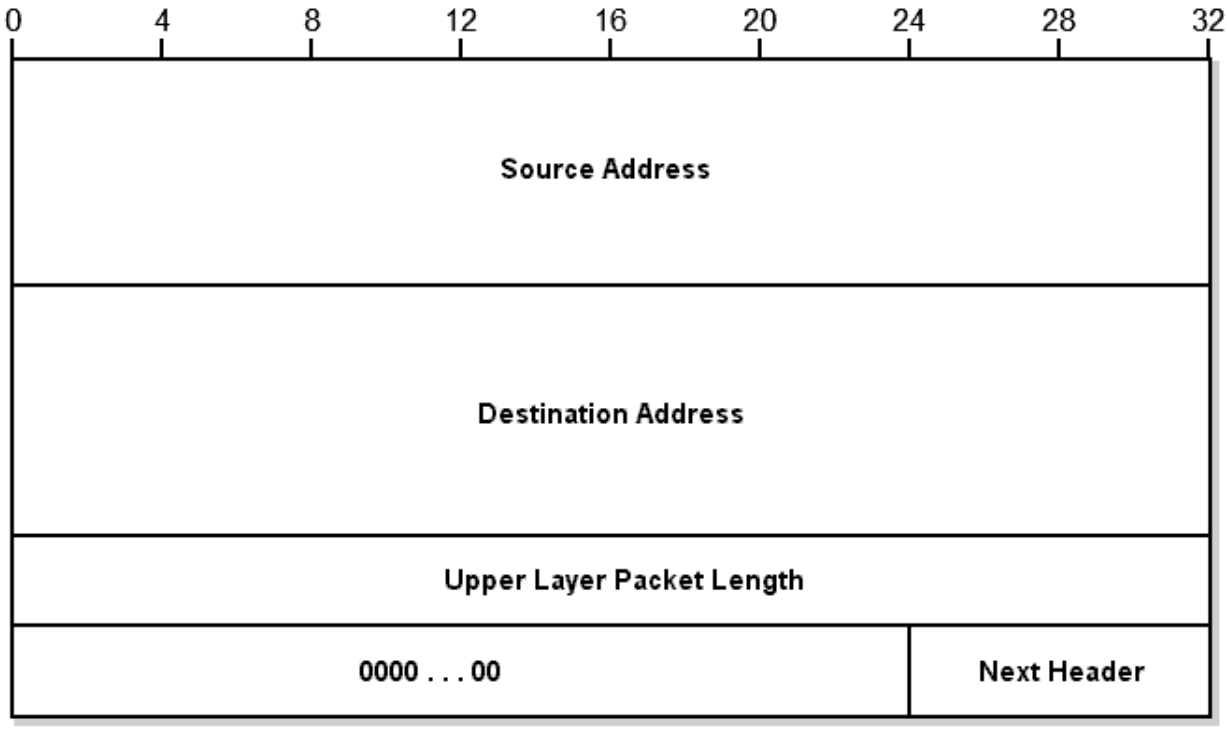

Figura 3.3 Pseudo-header utilizzata per da TCP, UDP e ICMP V6 per il calcolo del checksum

La lunghezza del campo Total_Lenght di IPv4 pari a 16 bit impone sugli ULP una lunghezza dei messaggi inferiore a 65535 byte; sia UDP che TCP per IPv4 sono stati pertanto definiti coerentemente con questo limite (campo UDP_Length e campo MSS in MSS_Option lunghi 16 bit). Nel passaggio da IPv4 a IPv6 non vi sono normalmente problemi con le dimensioni dei messaggi provenienti dagli ULP poiché anche il campo Payload_length di IPv6 è lungo 16 bit. Un'opzione specifica del nuovo protocollo, Jumbo Payload Option (RFC2675), consente tuttavia agli ULP di costruire ed inviare messaggi, detti jumbogram, lunghi fino a 4 Gb: in questo caso è necessario invece modificare opportunamente il funzionamento degli stessi protocolli UDP e TCP. Si è quindi convenzionalmente stabilito che, per richiedere un trasporto tramite jumbogram IPv6, i valori di UDP_Lenght e MSS di UDP e TCP devono rispettivamente essere impostati a 0 e 65535; la lunghezza del messaggio viene dedotta per differenza dalla lunghezza del datagram IPv6.

3.3 Le migliorie rispetto al predecessore

Facendo tesoro anche degli errori di progettazione di IPv4 e dei suoi limiti, i progettisti di IPv6 sono riusciti a realizzare un nuovo protocollo estremamente completo, versatile ed in grado di soddisfare le più disparate esigenze, introducendo molte novità rispetto alla versione precedente.

3.3.1 Indirizzi a 128 bit

IPv6 risolve il problema della limitazione dello spazio di indirizzamento di IPv4 aumentando la lunghezza fissa degli indirizzi da 32 a 128 bit: un tale spazio è in grado di assegnare teoricamente un massimo di $3{,}4*10^{38}$ indirizzi. Una tale lunghezza ha consentito di ottimizzare le operazioni di routing e forwarding ma ha reso indispensabile l'introduzione di meccanismi di autoconfigurazione degli host.

3.3.2 Routing e forwarding efficienti

La dimensione dello spazio di indirizzamento introdotta con IPv6 consente di creare a più livelli gerarchie di indirizzi, riducendo il carico di lavoro sui router grazie alla possibilità di utilizzare routing table più compatte e algoritmi più semplici. Inoltre i router intermedi possono provvedere ad inoltrare il pacchetto verso l'interfaccia destinataria prima ancora di aver ricevuto l'intero pacchetto (cut throught): la funzione di frammentazione è infatti consentita solamente sul nodo mittente, la verifica di integrità non è più prevista a livello IP e il formato innovativo della header (e l'ordine delle extension header suggerito da IETF) permette ai router stessi di ottenere tutte le informazioni necessarie all'inoltro solamente analizzando la parte iniziale del datagram.

3.3.3 Integrazione con IPSec

Sebbene compatibile anche con IPv4, IPSec (RFC2401) si è dovuto scontrare con la diffusione dei dispositivi di NAT, che ne hanno reso impossibile una diffusione su larga scala. IPv6 è stato invece progettato in modo tale da integrare nativamente i meccanismi di sicurezza previsti da IPSec. La definizione di uno standard di questo genere a livello IP permette a tutte le applicazioni, anche quelle non sicure per definizione, di usufruire di un canale comunicativo affidabile anche dal punto della sicurezza. IPSec garantisce infatti l'integrità, la provenienza e la confidenzialità dei dati definendo un header per l'autenticazione (AH) ed una funzionalità di incapsulamento sicuro del payload (ESP), insieme a metodologie per la gestione delle chiavi e particolari algoritmi di cifratura o autenticazione.

3.3.4 Autoconfigurazione

La crescente complessità e dimensione delle reti TCP/IP, nonché la lunghezza stessa degli indirizzi IPv6 ha spinto i progettisti ad implementare nel nuovo protocollo efficienti e soddisfacenti meccanismi di autoconfigurazione degli host per quel che riguarda indirizzi IP associati alle interfacce, router da utilizzare per la connessione ad Internet, etc… A questo scopo IPv6 dispone di diverse metodologie di autoconfigurazione, alcune delle quali si appoggiano ad altre macchine presenti nella rete (router o server DHCP) e altri 'autonomi'. Un primo meccanismo prevede che

all'avvio ogni host assegni a ciascuna interfaccia un indirizzo di tipo link-local ricavato dall'indirizzo di livello DLC (ovviamente l'algoritmo utilizzato varia in base al tipo di link sottostante). L'indirizzo link-local generato verrà poi utilizzato per comunicare all'interno della rete e ottenere quindi le informazioni necessarie per la configurazione di indirizzi global o site-local; in questo caso è possibile che venga utilizzato il protocollo ICMPv6 (configurazione stateless) oppure DHCPv6 (configurazione stateful, su DHCPv6 vedi RFC3315).

3.3.5 Identificazione dei flussi comunicativi

Il campo Flow_Label, che consente di identificare i pacchetti appartenenti allo stesso flusso comunicativo, può essere utilizzato in combinazione con l'Extension header Hop-by-Hop per garantire percorsi predefiniti. Inoltre la pratica di etichettare i flussi comunicativi può ridurre notevolmente i tempi di elaborazione dei dati sui router intermedi: quando il primo datagram di un flusso comunicativo viene consegnato ad un router, questo può memorizzare temporaneamente il valore del campo Flow_Label e quindi inoltrare tutti gli altri pacchetti ricevuti in seguito ma appartenenti allo stesso stream direttamente lungo lo stesso cammino, senza dover ogni volta consultare la tabella di routing.

3.3.6 Supporto alla mobilità

IPv6 è stato progettato con un occhio di riguardo alla mobilità (RF3775) e, non a caso, l'architettura di comunicazione 3G è stata pensata in funzione di questo protocollo. Con IPv6 ogni nodo mobile (cellulari, palmari etc.) è identificato da un indirizzo (Home_Address), utilizzato dagli altri nodi per comunicare con esso, ma nel momento in cui il dispositivo si sposta, collegandosi ad una rete esterna, ottiene temporaneamente un nuovo indirizzo, detto Care-of Address. Viene quindi utilizzato un 'agente' incaricato di tener traccia degli spostamenti del nodo (e quindi dei relativi indirizzi Care-of assegnati), intercettando i pacchetti destinati al nodo mobile e inoltrandoli verso l'indirizzo Care-of in quel momento assegnato al nodo.

Capitolo 4

La transizione da IPv4 a IPv6

Pensato non tanto come un protocollo rivoluzionario quanto piuttosto come un'evoluzione del suo predecessore, IPv6 introduce tuttavia un numero tale di migliorie e di novità da rendere questo protocollo sostanzialmente diverso da IPv4. Proprio per facilitare il processo di migrazione verso la nuova versione e la convivenza con l'onnipresente IPv4, IETF ha previsto e formalizzato diversi meccanismi di transizione. Tuttavia, poiché alcuni componenti dell'architettura di comunicazione TCP/IP violano il principio di indipendenza dei livelli, l'introduzione di IPv6 comporta comunque modifiche anche ai livelli superiori.

4.1 Le principali differenze tra i due protocolli

Pensato non tanto come un protocollo rivoluzionario quanto piuttosto come un'evoluzione del suo predecessore, IPv6 introduce tuttavia un numero tale di migliorie e di novità da rendere questo protocollo sostanzialmente diverso da IPv4. Non essendo quindi garantita la retrocompatibilità con IPv4, è evidente come la transizione al nuovo protocollo non sarà trasparente ai livelli sovrastanti IP. La nuova tipologia di indirizzi introdotti richiederà ad esempio modifiche in tutti quegli applicativi che trattano strutture dati contenenti indirizzi IP e, allo stesso modo, le diverse interfacce di programmazione (API) di rete dovranno essere estese al fine di supportare sia IPv4 che IPv6, permettendo peraltro all'applicazione di selezionare di volta in volta il protocollo desiderato. Sostanziali modifiche dovranno essere introdotte anche su tutti i nodi intermedi al fine di supportare l'innovativa struttura dell'header IP e gestire i nuovi campi introdotti.

4.1.1 Nuovo formato degli indirizzi

Indirizzi di lunghezza pari a 32 bit e notazione "dotted-decimal" con blocchi di 8 bit per IPv4 contro i 128 bit di un indirizzo IPv6, suddiviso in porzioni da 16 bit rappresentate in formato esadecimale; il nuovo spazio di indirizzamento introdotto da IPv6 è sicuramente una delle principali differenze con IPv4, nonché fonte di problemi di incompatibilità. Per facilitare il processo di transizione dal vecchio al nuovo protocollo sono state tuttavia introdotte tre tipologie di indirizzi unicast IPv6, pensate per poter contenere al loro interno indirizzi IPv4. In particolare gli indirizzi IPv4-Compatible IPv6, impiegati in alcune situazioni di protocol tunneling, sono formati da 96 bit posti a 0 seguiti dall'indirizzo IPv4 unicast da cui derivano (es. ::193.206.71.151). Gli indirizzi IPv4-Mapped IPv6 sono invece formati da 80 bit a 0 seguiti a una porzione di 16 bit a 1 e infine dall'indirizzo IPv4 da cui derivano (es. ::FFFF:193.206.71.151). Gli

indirizzi IPv4-Translated IPv6 sono invece formati da 64 bit a 1, seguiti da 16 bit a posti 1 e quindi altri 16 a 0 e quindi dai bit dell'indirizzo IPv4 da cui derivano (es. ::FFFF:0:193.206.71.151). Queste ultime due tipologie di indirizzi vengono impiegate in quelle situazioni in cui è necessario far colloquiare nodi IPv4-only con nodi IPv6-only.

4.1.2 Campo Header_Checksum

Nel protocollo IPv6 è stato eliminato il campo Header_Checksum presente in IPv4, contenente un checksum di controllo dell'integrità del messaggio. Questo accorgimento riduce notevolmente il carico computazionale sui router intermedi e la stessa velocità di forwarding, dal momento che non è più richiesto a questi dispositivi di attendere la ricezione dell'intera PDU prima di provvedere all'inoltro della stessa verso il destinatario per verificarne l'integrità o persino per ricalcolarne il valore (come accade oggi con IPv4 in presenza di frammentazione sui nodi intermedi o con dispositivi intermedi quali NAT, Proxy etc..).

4.1.3 Campo Flow_Label

Per flow si intende un flusso, una sequenza distinguibile di pacchetti inviati da una particolare sorgente verso uno specifico destinatario. In IPv4 una particolare sessione comunicativa viene univocamente identificata da una quintupla contenente l'indirizzo sorgente e destinatario, le porte sorgente e destinataria e il protocollo di livello transport. Tuttavia alcuni di questi campi possono non essere disponibili poiché cifrati o presenti in frammenti diversi dello stesso datagram IP; inoltre una classificazione basata solamente sul layer IP rende più agevole l'introduzione di protocolli di livello 4 alternativi. Per questo motivo in IPv6 la caratterizzazione dei flussi comunicativi è stata resa più efficiente mediante l'introduzione del campo Flow_Label (20 bit) generato casualmente dal mittente che, insieme agli indirizzi del mittente e del destinatario, permette di 'etichettare' una particolare sequenza di dati (detta anche microflow). La soluzione adottata da IPv6 rende sicuramente più efficiente le operazioni, poiché tutti i campi interessati si trovano esclusivamente all'interno del datagram IP e, soprattutto, in posizioni fisse. Per quel che riguarda tuttavia l'utilizzo del campo Flow_Label, non esiste un impiego unico e precodificato e la questione è ancora in corso di discussione. Si può ad esempio pensare che i router intermedi discriminino i diversi flussi comunicativi per trattare con le stesse modalità (stessa qualità del servizio) tutti i datagram appartenenti al medesimo flusso; una soluzione alternativa prevede l'utilizzo del campo per ospitare una label utilizzata per un forwarding sulle reti di tipo label-swapping (ATM, MPLS). In ogni caso questo campo non presenta particolari problemi in termini di retro-compatibilità con IPv4 (ad esempio in caso di traduzione 4-6): gli host o i router che non

supportano questa soluzione è sufficiente che impostino a zero il valore nei pacchetti originati e che lo ignorino in quelli ricevuti.

4.1.4 Traffic_Class

Il campo Traffic Class (8 bit) di IPv6 permette ai nodi mittenti e intermedi di identificare e impostare diverse classi o priorità dei pacchetti, fornendo quindi al traffico un servizio differenziato in base alla tipologia di servizio richiesto. Il campo corrispondente in IPv4 è Type_of_Service ed i primi 6 bit di entrambi sono stati utilizzati dal gruppo di lavoro "Differentiated Services" dell'IETF come DS_Field (RFC3260). I meccanismi del modello DiffServ sono decisamente complessi e non verranno approfonditi in questa sede; basti sapere che impostando opportunamente la classe di traffico è possibile ottenere diverse tipologie di comportamento, dallo standard best-effort forwarding (RFC1812) al forwading che fornisce garanzie sul recapito (Assured Forwarding) o sul tempo di forwarding (Expedited Forwarding) dei pacchetti.

4.1.5 ARP/ICMP e ICMPv6

Le funzionalità di risoluzione degli indirizzi e di error o information reporting fornite in IPv4 da distinti protocolli di supporto sono state in IPv6 integrate nell'unico protocollo ICMPv6. Inoltre, come già in precedenza spiegato, ICMPv6 è stato ulteriormente espanso al fine di implementare servizi necessari al nuovo protocollo di Internet. A quest'ultima categoria appartengono i servizi di Neighbor Discovery (RFC2461), Address Autoconfiguration (RFC2462), Duplicate Address Detection (RFC2462) e Path MTU Discovery; data la loro complessità non verranno ulteriormente approfonditi ma è facile comprendere come questi meccanismi, essendo specifici di IPv6 e non trovando quindi un corrispondente in IPv4, rendano difficile l'interoperabilità tra i due protocolli.

4.1.6 IP_Option e Extention Headers

In IPv4 IP_Options è uno strumento attraverso cui è possibile aggiungere nuovi campi alla header di base; si tratta comunque di uno strumento piuttosto limitato, dal momento che lo spazio massimo disponibile è di soli 40 byte. IPv6 invece supera il concetto di IP Options, implementando un meccanismo estremamente modulare e flessibile che prevede che un header di base concatenata ad una serie di header successive che specificano di volta in volta opzioni aggiuntive. Una scelta di questo tipo presenta indubbiamente due principali vantaggi: la dimensione dell'header principale non ha più dimensione variabile (come accadeva in IPv4) bensì fissa e l'aggiunta di nuove funzionalità o la modifica di alcune di esse non ha alcuna ricaduta sull'header base di IPv6 né sulle altre Extension header.

4.1.7 Frammentazione e Path MTU Discovery

I diversi campi previsti da IPv4 per consentire la frammentazione dei datagram sono stati eliminati nella header di base IPv6 ed il procedimento di frammentazione è stato semplificato, negando la possibilità ai router intermedi di frammentare e consentendo solamente al host mittente di utilizzare l'apposita Extension header (Fragment Header) durante la generazione del pacchetto. Se si considera inoltre che il percorso che deve seguire un datagram per giungere a destinazione solitamente è costituito da una sequenza di link aventi diversi MTU (*Maximum Trasmission Unit*, in questo caso la dimensione massima del datagram IP consentita dal DLC) appare evidente come sia importante per il destinatario impostare dimensioni dei datagram accettabili da tutti i nodi intermedi. Per risolvere questo problema il destinatario può utilizzare la MTU minima prevista per tutti i link che trasportano IPv6 (1280 byte), sottoutilizzando però quasi sicuramente i canali, oppure decidere di 'scoprire' dinamicamente la MTU massima del percorso e trasmettere quindi datagram aventi dimensioni ottimizzata. Questa seconda soluzione è resa possibile in IPv6 dal meccanismo di Path-MTU Discovery, previsto anche in IPv4 ma quasi mai utilizzato, che permette al nodo mittente di scoprire per "trial-and-error" la MTU del percorso (PMTU). Inizialmente la stazione mittente assume come PMTU la MTU del link a cui è direttamente connessa: se la MTU del primo link è la più piccola del percorso allora essa coincide con la PMTU e pertanto il datagram verrà consegnato a destinazione. Se, al contrario, il pacchetto incontra un link con una MTU inferiore a quella impostata il router interessato scarta il pacchetto ed invia un messaggio di errore ICMPv6 del tipo 'Packet Too Big' al mittente, indicando inoltre il valore di MTU che ha causato il problema. Compito del mittente è quindi generare un nuovo pacchetto avente MTU pari alla nuova MTU minima scoperta lungo il percorso ogni volta che viene ritornato un messaggio "Packet Too Big".

4.2 I meccanismi di transizione

Come già accennato in precedenza, IPv4 si è ormai radicato a tal punto da rendere impossibile un passaggio rapido ad IPv6; quello che invece sta avendo luogo (e accadrà ancora per diversi anni) è una graduale transizione di Internet, dove però ancora per lungo tempo IPv4 e IPv6 convivranno. A questo proposito il gruppo di lavoro ngtrans della Internet Engineering Task Force ha proposto un gran numero di strategie di migrazione, essenzialmente riconducibili a tre principali tipologie: Dual stack, Translation e Tunneling.

4.2.1 Dual Stack

La prima componente fondamentale per la migrazione è la tecnica dual stack. Come suggerisce il nome stesso, i nodi di rete implementano due distinti stack di rete che operano in parallelo,

permettendo alle applicazioni IPv4 e IPv6 di utilizzare lo stack corrispondente. La discriminazione dei diversi flussi comunicativi in ingresso viene fatta sulla base del valore presente nei primi 4 bit del datagram IP (0100 per IPv4, 0110 per IPv6) mentre, per la spedizione, ci si basa sul formato dell'indirizzo di destinazione. Analogamente, la scelta dello stack appropriato per le risposte DNS ricevute viene fatta sulla base del tipo di record contenuto.
Dual stack è una soluzione estremamente semplice e che non richiede alcuna modifica ai livelli applicativi ma che consente solamente ai due protocolli di convivere all'interno della stessa rete, senza però che questi possano interoperare tra loro (host IPv4 possono cioè comunicare solo con altri host IPv4 e lo stesso dicasi per IPv6).

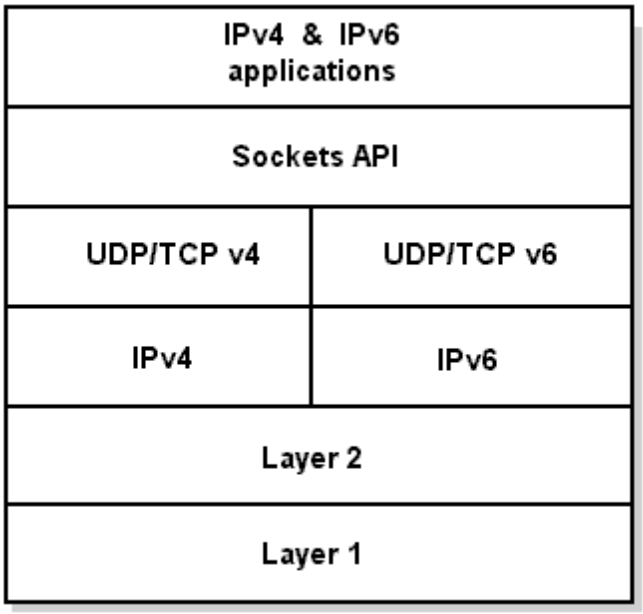

Figura 4.1 Schema del meccanismo di transizione Dual Stack

4.2.2 Translation

Per consentire l'interoperabilità tra i diversi protocolli è necessario ricorrere ad un meccanismo di translation, ovvero di 'conversione' di un protocollo in un altro attraverso la trasformazione dell'header ed, eventualmente, del payload. La traduzione, che può avvenire a diversi livello nello stack di rete (network, transport e persino applicativo), molto spesso ha il difetto di introdurre perdite di informazioni. In tutti quei casi in cui non esiste una corrispondenza tra i campi dei due protocolli i valori possono infatti essere persi o impostati a valori standard senza particolare significato; la traduzione di un datagram IPv6 in IPv4 comporterà sicuramente la perdita, ad esempio, del valore di Flow_Label.
In base al funzionamento, i meccanismi di traduzione possono essere distinti in stateless e stateful. Un traduttore stateless è in grado di effettuare ogni singola conversione indipendentemente dalle

altre; un traduttore stateful invece necessita di mantenere in memoria informazioni sulle traduzioni effettuate in precedenza (es. corrispondenze tra le due tipologie di indirizzi).

4.2.2.1 SIIT

Essenziale per il processo di transizione è sicuramente la conversione dei pacchetti IP e ICMP; a questo proposito è stato ideato SIIT. L'algoritmo SIIT, acronimo di Stateless IP/ICMP Translation (RFC2765), indica le modalità di traduzione bidirezionale delle testate IPv4 e IPv6 e dei messaggi ICMPv4 e ICMPv6. Questo meccanismo, su cui si basano diversi traduttori, ignora molte delle Extension header di IPv6 così come diverse opzioni di IPv4; tuttavia è stato progettato appositamente in maniera da mantenere inalterato nel processo di traduzione il checksum di UDP e TCP calcolato utilizzando le pseudo-header.

4.2.2.2 BIA/BIS

Alcune tipologie di traduttori operano direttamente sugli end-system, inserendosi sotto forma di modulo aggiuntivo all'interno dello stack TCP/IP. IETF ha proposto due diversi meccanismi, entrambi pensati per consentire ad applicazioni IPv4 di operare all'interno di reti IPv6.

La soluzione BIS, Bump-in-the-Stack (RFC2767), prevede un modulo traduttore inserito tra il 4 livello in grado di identificare i pacchetti di livello applicativo che attraversano lo stack TCP/IPv4 e di conseguenza tradurli prima di inoltrarli via IPv6.

Anche BIA, Bump-in-the-API (RFC3338), permette alle applicazioni pensate per IPv4 di comunicare con host IPv6 ma, trovandosi ad un livello superiore rispetto a BIS, è in grado di intercettare direttamente le chiamate alle Socket API. Questo permette a BIA di evitare la traduzione di pacchetti IP e non è nemmeno necessaria la modifica del nucleo del sistema operativo.

Sia BIS che BIA utilizzano per il loro funzionamento due componenti comuni: un name resolver ed un address mapper. Il primo ha il compito di effettuare richieste DNS per decidere se il destinatario del datagram IP supporta solamente IPv6 (ovvero se la traduzione è effettivamente necessaria o meno); il secondo si incarica invece di allocare temporaneamente un indirizzo IPv4 utilizzato dall'applicazione e associato all'interfaccia IPv6 destinataria. Entrambi i meccanismi presentano tuttavia il grosso limite di non poter gestire (e quindi tradurre) eventuali indirizzi inseriti nei protocolli di livello applicativo.

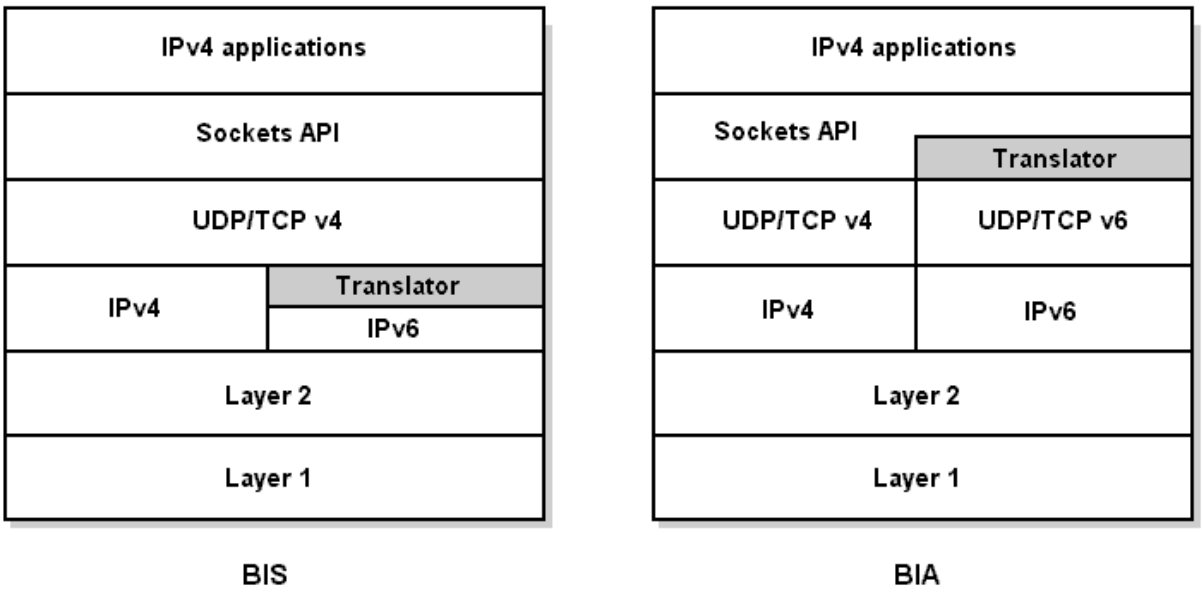

Figura 4.2 Schema dei meccanismi di transizione BIA e BIS

4.2.2.3 NAT-PT

Network Address Translation – Protocol Translation (RFC2766) è un traduttore IPv4/IPv6 stateful che basa il proprio funzionamento sull'algoritmo descritto da SIIT. Esistono differenti modalità di funzionamento di questo traduttore ma, in linea di massima, è possibile pensare a NAT-PT come ad un traduttore in grado di permettere ad uno o più nodi IPv6 di comunicare con host IPv4 in maniera analoga a quanto avviene in presenza di un Proxy, allocando temporaneamente per ciascuno di essi indirizzi IPv4 e tenendo traccia delle associazioni. Utilizzando inoltre appositi ALG (Application Level Gateway) NAT-PT è in grado di effettuare la traduzione dei protocolli di livello applicativo (FTP, DNS etc..). Tuttavia, poiché questo meccanismo deve tenere traccia delle associazioni effettuate tra gli indirizzi (stateful), è necessario che tutti i datagram di una sessione vengano inoltrati attraverso il medesimo dispositivo NAT-PT.

4.2.2.4 TRT

E' possibile anche pensare di utilizzare un router posto a cavallo di due distinti segmenti di rete, ciascuna delle quali basata su una versione differente del protocollo IP. Il Transport Relay Translator (RFC3142) converte sessioni TCP/UDPv6 in sessioni TCP/UDPv4. La traduzione ha quindi luogo solo a livello 4; anche in questo caso, operando su sessioni comunicative, è necessario che tutto il traffico relativo alla stesso flusso attraversi lo stesso router (stateful). La comunicazione, iniziata dal lato IPv6, attraversa il router traduttore; quest'ultimo provvederà a terminare la sessione IPv6 e inizierà quindi una nuova sessione comunicativa IPv4 verso il

destinatario, il cui indirizzo viene ricavato direttamente dall'indirizzo IPv6 (al prefisso di 64 bit deve seguire infatti l'indirizzo IPv4 dell'host destinatario).

4.2.3 Tunneling

Il tunneling come meccanismo di transizione è utilizzato per interconnettere tra loro host che altrimenti non potrebbero comunicare poiché 'separati' da reti incompatibili. I datagram IP sono quindi trasportati all'interno di altri datagram IP, in modo che il protocollo incapsulato veda quello incapsulante come un DLC, un 'link virtuale'. In base alle esigenze si possono utilizzare differenti combinazioni di tunneling IP (IPv4 in IPv6 o viceversa ma anche IPv4 in IPv4 o IPv6 in IPv6) e vari tipologie di invio sul link virtuale ottenuto.

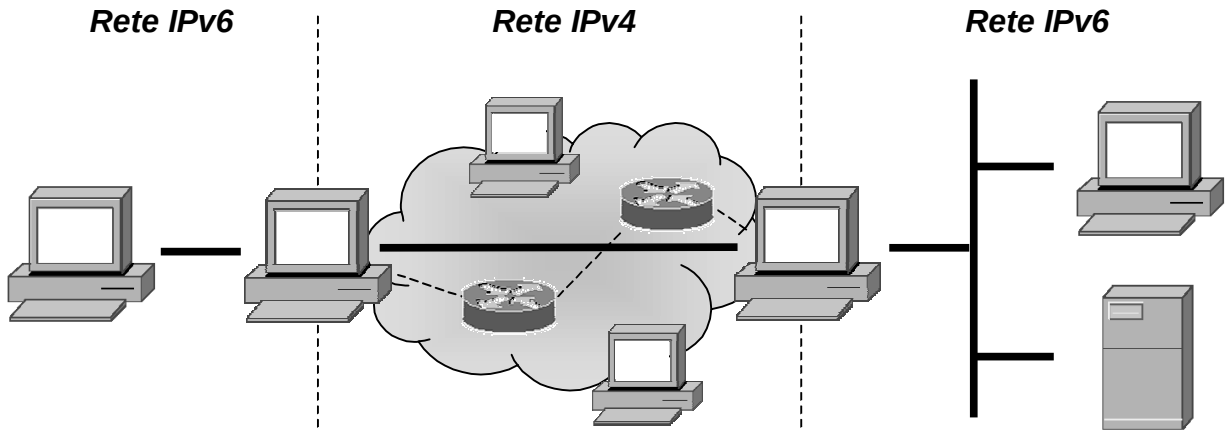

Figura 4.3 Esempio di tunneling IPv6-in-IPv4 per interconnettere due 'isole' IPv6

4.2.3.1 6over4

Il primo dei meccanismi di tunneling utilizzabili è 6over4 (RFC2529), altresì noto come 'virtual Ethernet' poiché in grado di mettere in comunicazione tra loro via tunneling più host IPv6 dual stack 'isolati' connessi sulla stessa rete IPv4. Con 6over4 i nodi IPv6 risultano connessi ad un'unica Ethernet virtuale, basata su IPv4 e identificata da un indirizzo IPv4 multicast. IPv6 si appoggia quindi su IPv4 per svolgere tutte le sue funzioni tipiche (autoconfigurazione degli indirizzi link-local, neighbor discovery…), utilizzando in sostanza gli indirizzi del protocollo V4 sottostante in sostituzione dei MAC_Address.

4.2.3.2 Configured IP-in-IP

Configured IP-in-IP è invece la forma di tunneling forse più conosciuta e già utilizzata con IPv4 per la costruzione di VPN (Reti Private Virtuali). Utilizzando sempre un incapsulamento di datagram IPv6 in IPv4 viene configurata una connessione punto-a-punto virtuale IPv6 tra i due end-point del tunnel, dotati di doppio stack. A questo proposito è opportuno ricordare che sono

stati definiti alcuni meccanismi automatizzati (Tunnel Broker) per l'attivazione di tunnel IPv6; fondamentale in questo ambito è stato il lavoro di ricerca svolto dal TelecomItalia Lab di Torino, che ha portato alla formalizzazione di questo meccanismo nell'RFC3053.

4.2.3.3 6to4

6to4 Automatic Tunneling infine è un meccanismo che permette a reti IPv6 di comunicare tra loro attraverso una dorsale IPv4 senza la necessità di alcuna configurazione esplicita. Ciascuna delle reti interessate viene infatti indirizzata attraverso speciali prefissi globali IPv6 appositamente riservati, identificati da un prefisso di 48 bit ottenuto concatenando '2002' con i 32 bit dell'indirizzo IPv4 del router di collegamento alla dorsale (es. 2002:193.206.71.156::/48).

4.3 IPv6 e gli ULP

Un'architettura a strati, come visto in precedenza, costruisce un servizio comunicativo operando 'per gradi' e prevede che vengano rispettati alcuni principi comunicativi perché questa possa essere ritenuta effettivamente tale. In particolare il principio di indipendenza dei livelli prevede che un qualsiasi livello possa essere sostituibile, in maniera del tutto trasparente, con un qualsiasi altro livello in grado di fornire lo stesso servizio al livello sovrastante e richiedere lo stesso servizio a quello sottostante. Non tutti i protocolli rispettano questo principio: l'approccio seguito dal TCP/IP prevede l'utilizzo di dati del 3° livello per alcune operazioni (nello specifico la generazione delle pseudo-header) effettuate dai protocolli TCP e UDP e lo stesso dicasi per alcuni protocolli di livello applicativo. E' pertanto evidente come si siano dovute introdurre modifiche su questi ULP, per renderli compatibili con il nuovo protocollo IP, ed è altrettanto semplice intuire come si debba tenere conto anche degli ULP applicativi durante le fasi di traduzione di un datagram IPv4 in IPv6, pena rendere inutilizzabili gli stessi dati trasmessi.

4.3.1 Il protocollo FTP

Il protocollo FTP, acronimo di File Transfer Protocol (RFC959), è una delle applicazioni client-server Internet più utilizzate per lo scambio di file tra host connessi in rete. FTP è un servizio esclusivamente basato su TCP e si distingue dagli altri protocolli per il suo 'insolito' comportamento, che prevede l'utilizzo di due connessioni TCP anziché una sola. Ogni volta che ci si connette ad un server FTP viene instaurata una connessione, detta connessione di controllo, basata sul protocollo Telnet; questa rimane attiva per tutta la durata della comunicazione e viene utilizzata dal client per l'invio di messaggi di richiesta ai quali il server risponderà. Per il trasferimento effettivo dei singoli file o directory verrà invece instaurata di volta in volta una nuova connessione, chiamata connessione dati; a differenza della Control connection, le diverse

Data connection vengono abbattute immediatamente una volta terminato il trasferimento del file. Le porte utilizzate da client e server per l'instaurazione delle connessioni e la scelta dell'entità a cui spetta il compito di richiedere l'apertura di nuove connessioni dati dipendono dalla modalità di funzionamento, che può essere 'attiva' o 'passiva'.

4.3.1.1 Modalità attiva

In active mode il server viene invitato a collegarsi ad una porta specificata arbitrariamente dal client. Inizialmente il client FTP si connette alla porta 21 del server FTP da una porta N non privilegiata (cioè superiore a 1024). Una volta stabilita la connessione di controllo, il client si mette quindi in ascolto sulla porta N+1 e la comunica al server con il comando PORT; a questo punto il server originerà ogni nuova connessione per il trasferimento dati dalla porta 20 e si connetterà alla porta N+1, indicata proprio dal client.

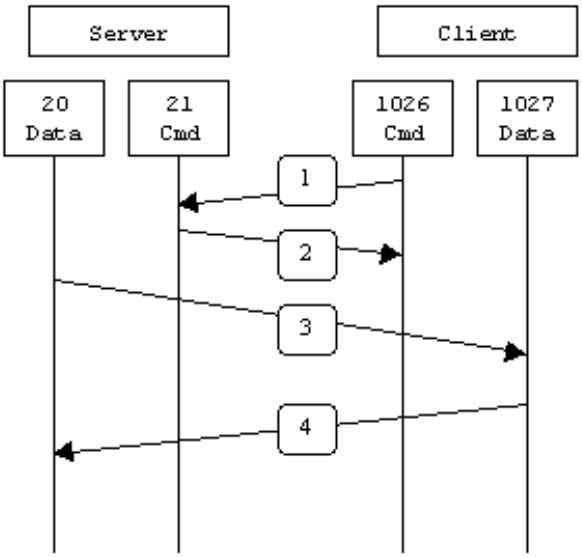

Figura 4.4 Schema di funzionamento FTP in modalità attiva

4.3.1.2 Modalità passiva

In modalità attiva è il server ad originare le connessioni per il trasferimento dati; questa soluzione può presentare diversi problemi nel caso in cui il client risieda dietro ad un firewall o un NAT, sistemi che molto spesso bloccano i tentativi di connessione dall'esterno. Per ovviare a questo problema, consentendo ai client di originare tutte le connessioni e non solamente quelle di controllo, è stata introdotta una modalità alternativa detta passiva. Questo modo di funzionamento, peraltro ufficialmente indicato come preferibile rispetto alla modalità attiva (RFC1579), prevede che il client FTP apra inizialmente una porta N non privilegiata e la porta N+1; la prima viene utilizzata per contattare il server, in ascolto sulla porta 20 e la seconda per originare le connessioni

dati. Per sapere tuttavia quale porta del server contattare, il client deve inviare il comando PASV; a questa richiesta il server risponde indicando in numero di porta P (con P maggiore di 1024) sulla quale è in ascolto. A questo punto il client originerà tutte le connessioni di trasferimento dei file dalla porta N+1 verso la porta P del server.

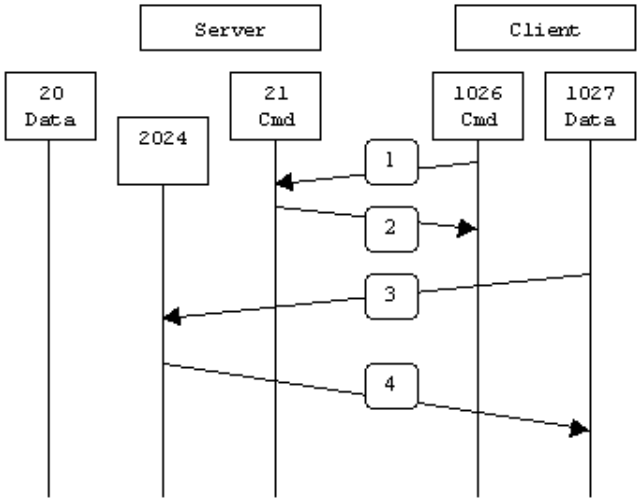

Figura 4.5 Schema di funzionamento FTP in modalità passiva

4.3.1.3 FTP ed IPv6

Il fatto che FTP utilizzi le porte per discriminare i flussi comunicativi, pur violando il principio di indipendenza dei livelli, non rappresenta un problema per l'introduzione di IPv6: anche questo protocollo utilizza le porte in maniera del tutto analoga a IPv4. Tuttavia il formato dei comandi PORT e PASV è stato appositamente pensato in funzione di IPv4, dal momento che vengono utilizzati come parametri nelle richieste o nelle risposte anche indirizzi IP. A tal proposito sono stati proposti (RFC2428) due comandi alternativi, in grado di supportare sia IPv4 che IPv6 o eventuali futuri protocolli di rete.

Il comando PORT infatti prevede come argomento sei numeri decimali lunghi 8 bit, separati da virgole, che indicano il socket di connessione da utilizzare: nello specifico i primi 4 numeri corrispondo agli ottetti del indirizzo IPv4 di destinazione mentre la porta da utilizzare è specifica negli altri due. Poiché il numero massimo di porte disponibili in un sistema è 65535, gli 8 bit più significativi del numero di porta vengono indicati con il quinto valore mentre gli 8 bit meno significativi si trovano in ultima posizione (es. 'PORT 193,206,71,151,4,1' indica la porta 1025 dell'indirizzo 193.206.71.151). Il comando EPRT sostituisce il PORT, consentendo di specificare un indirizzo di livello 3 più esteso da utilizzare per la connessione, unitamente al tipo di protocollo utilizzato e agli indirizzi di livello transport (le porte nel caso di TCP). Nello specifico il comando EPRT deve essere seguito da un numero indicante il protocollo di livello transport utilizzato (1 per IPv4, 2 per IPv6), l'indirizzo ed infine la porta; ogni valore deve essere inoltre separato dagli altri

utilizzando un apposito carattere ASCII come separatore (è raccomandato l'utilizzo di '|'). A questo proposito è stato introdotto anche il nuovo codice di risposta 522, utilizzabile dal server per indicare che il protocollo indicato nel comando EPRT non è supportato.
Il comando PASV invece non prevede l'invio da parte del client di ulteriori parametri; la risposta del server a questa richiesta deve tuttavia contenere un messaggio di testo seguito da 6 decimali separati da virgole, indicanti l'indirizzo e la porta su cui il server stesso si è posto in ascolto (es. "Sto entrando in modalità passiva 193,206,71,151,4,1"). Per supportare IPv6 PASV è stato sostituito dal comando EPSV, che prevede che il server risponda al client indicando semplicemente la porta TCP a cui connettersi tra parentesi e preceduta da due campi lasciati vuoti (es. "Sto entrando in modalità passiva (|||1025|)"). L'indirizzo IP ed il tipo di protocollo non devono invece essere specificati poiché si presuppone che l'indirizzo ed il protocollo da utilizzare per la Data connection siano gli stesso utilizzati dal client per la connessione di controllo. Il client può tuttavia richiedere l'utilizzo di uno specifico protocollo di livello network inviando il comando EPSV seguito dal numero del protocollo; se non è supportato dal server, quest'ultimo dovrà rispondere ritornando un mesaggio di errore 522 (Extended Port Failure, uknown network protocol). Infine il comando EPSV può essere seguito dall'argomento ALL per indicare ai dispositivi NAT intermedi che i comandi EPRT, PORT e PASV non verranno più usati durante la connessione.

4.3.2 Il protocollo DNS

DNS (RFC1035) è un complesso sistema che svolge un compito fondamentale per il funzionamento di Internet, effettuando la traduzione dei nomi simbolici degli host in indirizzi IP. Nell'utilizzo quotidiano infatti ci si riferisce difficilmente alle macchine connesse in rete con il loro indirizzo di rete, preferendo utilizzare nomi mnemonici. Proprio a questo scopo è stato introdotto oltre 20 anni fa un database distribuito, implementato in una gerarchia di server, che permette di risalire all'indirizzo associato ad un hostname. Per interrogare questo complesso database è disponibile un protocollo di livello applicativo client-server che si appoggia su UDP per effettuare la ricerca delle corrispondenze. E' pertanto evidente come, pur trattandosi di un protocollo sito in cima alla pila, DNS per sua natura trasporti informazioni di livello 3 (cioè gli indirizzi IP).

4.3.2.1 DNS ed IPv6

Il protocollo DNS ha una struttura piuttosto complessa ma è decisamente flessibile e facilmente espandibile. Nella sostanza l'introduzione di IPv6 non ha reso necessaria alcuna modifica alla struttura di base né sono stati introdotti problemi di incompatibilità o conflitti con IPv4. Infatti

nelle Query DNS e nelle relative Reply è stata introdotta una nuova 'sezione', un nuovo resource record (RR) contenente l'indirizzo IPv6 richiesto (RFC3596). Questa soluzione permette ad esempio di effettuare un'unica richiesta e ottenere sia l'indirizzo IPv4 che quello IPv6 in RR distinti; sarà poi compito del host richiedente utilizzare solo le informazioni di interesse. Ogni RR contenuto in una risposta DNS è identificato da uno codice Type che ne identifica la tipologia: così come il tipo A (codice decimale 1) identifica indirizzi IPv4, la tipologia AAAA (codice 28) è stata introdotta per gli indirizzi IPv6. Durante una prima fase di sperimentazione era stata introdotta anche la tipologia A6, ora però non più utilizzata poiché deprecata da IETF in favore di AAAA.
Con IPv4 è possibile inoltre effettuare delle 'query inverse' per risalire al hostname a partire all'indirizzo ad esso assegnato utilizzando un particolare dominio, IN-ADDR.ARPA, strutturato indicativamente come la gerarchia di indirizzamento di Internet. Per poter effettuare questo tipo di richieste è sufficiente effettuare una query DNS indicando come nome da risolvere il prefisso IN-ADDR.ARPA, preceduto dai quattro ottetti dell'indirizzo IPv4 in ordine inverso. Questa operazione è consentita anche in IPv6 utilizzando l'apposito dominio IP6.ARPA, preceduto dalle 32 cifre esadecimali che compongono l'indirizzo IPv6, riportate invertendo l'ordine e separandole con dei punti.

4.3.3 Il protocollo SIP

Numerosi protocolli di comunicazione utilizzati in Internet necessitano di un protocollo di sessione per negoziare lo scambio di dati tra gli end-point. Inoltre, con l'evoluzione del mobile computing e delle reti wireless, l'utilizzo di sessioni è necessario per gestire gli spostamenti degli utenti e l'eventuale aggiunta o rimozione di dispositivi durante la comunicazione. A questo proposito IETF ha realizzato Session Initiation Protocol (SIP, RFC3261), un protocollo di livello applicativo semplice e flessibile studiato per consentire a due user agent di ritrovarsi e accordarsi sui parametri da utilizzare durante la sessione comunicativa, come ad esempio una telefonata via Internet. Con SIP è possibile stabile nuove sessioni, invitare nuovi partecipanti ad unirsi ad essa e gestire anche gli spostamenti degli utenti. In realtà questo protocollo non costituisce un sistema di comunicazione completo ma è in grado di cooperare con altri protocolli per fornire agli utenti servizi completi e sicuri; tipicamente viene utilizzato in combinata con Real-Time Transport Protocol (RFC3550) per garantire trasferimenti in tempo reale con una determinata QoS o con Session Description Protocol (RFC3266) per descrivere sessioni multimediali o ancora con Real-Time Streaming Protocol (RFC2326) per controllare la diffusione di stream audio/video.
Poiché al suo interno SIP trasporta gli indirizzi degli user agent, necessari per inizializzare la sessione comunicativa, è lecito supporre che vi possano essere dei problemi modificando il protocollo di 3° livello. In realtà i progettisti di SIP hanno sviluppato questo protocollo rendendolo

compatibile sia con IPv4 che con IPv6: indirizzi IPv6 possono essere infatti tranquillamente utilizzati all'interno dell'header racchiudendoli tra parentesi quadre. Per quanto riguarda invece gli altri protocolli sopra citati utilizzati insieme a SIP, è necessario di volta in volta analizzare il formato dei messaggi per verificare la compatibilità con il nuovo protocollo. Infine è bene tenere a mente che anche SIP, come gli altri ULP contenenti indirizzi IP all'interno dei messaggi inviati, è inutilizzabile in presenza di tutti quei meccanismi di transizione che operano la traduzione dei protocolli solamente a livello TCP/IP (SIIT, TRT, NAT-PT senza ALG…).

4.3.4 Gli altri protocolli applicativi

La maggior parte dei protocolli applicativi fortunatamente non contiene informazioni di livello IP all'interno dei propri messaggi e non sono pertanto richiesti particolari accorgimenti o modifiche al protocollo stesso. Http e Pop, per citare solo i due più noti, utilizzano infatti i nomi simbolici per consentire agli host di comunicare tra loro, rendendo il funzionamento del protocollo slegato dai protocolli sottostanti.

Capitolo 5

SIIT

SIIT, Stateless IP/ICMP Translation (RFC2765), è un algoritmo che definisce le operazioni necessarie per realizzare la traduzioni di pacchetti IP e ICMP dalla versione 4 alla 6 e viceversa in modalità stateless; è cioè possibile effettuare la traduzione di ogni singolo pacchetto in maniera indipendente dagli altri. Formalizzato dal gruppo di lavoro ngtrans di IETF, questo algoritmo solitamente non opera autonomamente ma viene utilizzato insieme ad altre componenti per realizzare una traduzione completa dei flussi comunicativi tra host IPv4-only e host IPv6 che non hanno un indirizzo IPv4 permanente assegnato. Seppur fondamentale ai fini della traduzione, SIIT non definisce infatti alcuna modalità per l'assegnazione temporanea degli indirizzi o le modalità di routing dei pacchetti.

5.1 Il funzionamento

Un traduttore SIIT può essere utilizzato all'interno di reti di dimensioni limitate contenenti host IPv4-only e host IPv6-only. L'impiego di questo meccanismo è invece impensabile all'interno di reti di grandi dimensioni, poiché ogni host IPv6 dovrebbe acquisire un indirizzo IPv4 assegnato da un dispositivo SIIT "distante" e verrebbe propagato un numero di regole di routing eccessivo per i diversi indirizzi IPv4-mapped.

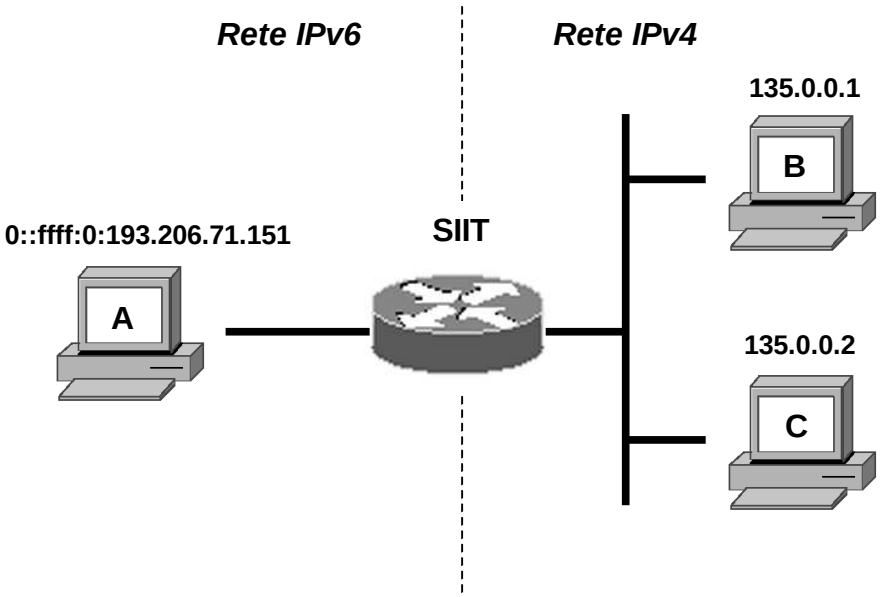

Figura 5.1 Traduttore SIIT utilizzato per interconnettere una rete IPv4 con una IPv6

Come avevamo già visto in precedenza, per facilitare il processo di transizione da IPv4 ad IPv6 sono state definite tre particolari strutture di indirizzi IPv6, costruiti concatenando un apposito

prefisso di 96 bit ed l'indirizzo IPv4 da 'trasformare' in indirizzo IPv6. In particolare gli indirizzi di tipo IPv4-compatible (es. ::a.b.c.d.) sono utilizzati per fornire un supporto automatico ai meccanismi di tunneling, quelli IPv4-mapped (es. 0::ffff:a.b.c.d) per indirizzare un pacchetto ad un nodo IPv4-only e quelli IPv4-translatable, ovvero del tipo 0::ffff:0:a.b.c.d, per riferirsi ad un nodo IPv6. Queste ultime due tipologie di indirizzi sono utilizzate da SIIT per la trasmissione di pacchetti da e per la rete V4 all'interno della rete IPv6: in particolare alle interfacce degli host IPv6-only, poiché non è possibile assegnare direttamente un indirizzo IPv4 da utilizzare per comunicare con la rete esterna, viene assegnato un indirizzo IPv4-translated. Inoltre, trattandosi sempre di host IPv6-only, i pacchetti destinati alla rete V4 verranno inviati all'indirizzo del host destinatario 'trasformato' in IPv4-mapped attraverso il nodo traduttore. Occorre inoltre notare come la struttura di queste due classi di indirizzi IPv6 sia stata appositamente studiata in modo tale da mantenere inalterato nel passaggio da IPv4 ad IPv6 il checksum di TCP e UDP, calcolato utilizzando la pseudo-header.

In presenza di un meccanismo di traduzione SIIT, per comunicare con il nodo IPv4-only l'host IPv6 utilizzerà un indirizzo IPv4-translated come proprio local address e vedrà il destinatario come un nodo avente un indirizzo di tipo IPv4-mapped. Per i pacchetti originati invece dall'host IPv4 e indirizzati all'host IPv6 il procedimento è esattamente l'opposto.

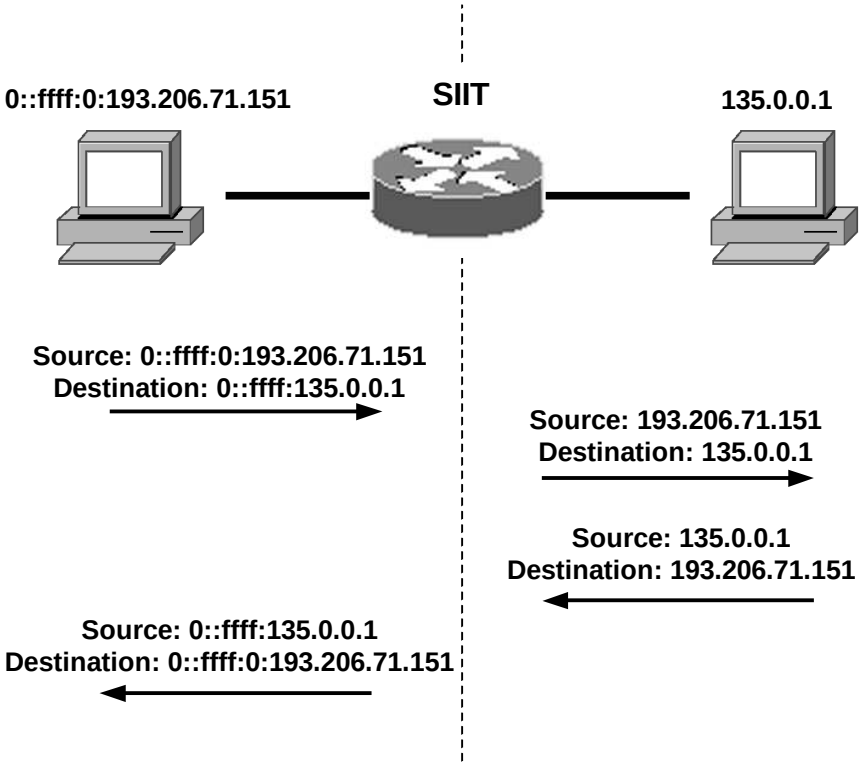

Figura 5.2 Esempio di procedura di traduzione con SIIT

5.2 La traduzione di datagram IP

La traduzione delle header IPv4 in testate IPv6 non presenta particolari problemi: con l'esclusione del Flow_Label, posto zero perché non presente in IPv4, tutti gli altri campi possono essere copiati o ricavati direttamente dalla testata originale. Il campo Traffic_Class ripresenta il valore presente in Type_of_Service così come in Next_Header viene copiato il campo Protocol e in Hop_Limit il campo TTL di IPv4, decrementato però di due punti (poiché il traduttore è un router sia per IPv4 che per IPv6). L'indirizzo mittente, di tipo IPv4-mapped, è ottenuto concatenando il prefisso ::ffff:0:0/96 e i 32 bit dell'indirizzo IPv4 mittente; analogo discorso per l'indirizzo del destinatario, di tipo IPv4-translated generato a partire proprio dall'indirizzo IPv4 di destinazione originariamente indicato. Per i pacchetti che attraversano il traduttore SIIT in senso contrario le operazioni sono ovviamente inverse e l'unico campo che deve essere calcolato appositamente, essendo stato abbandonato con IPv6, è Header_Checksum (ovviamente dovrà essere computato per ultimo, solo dopo che tutti gli altri campi sono stati tradotti o impostati ai valori di default).

Un discorso a parte deve essere fatto per la gestione della frammentazione in presenza di un nodo traduttore IPv4-IPv6. Diversi fattori concorrono infatti a rendere piuttosto complicata la gestione di questa funzionalità: tra queste la differente strategia di gestione dei frammenti in IPv4 (mediante appositi campi) e in IPv6 (mediante un'apposita Extension Header), il processo di calcolo della MTU massima lungo il percorso non obbligatorio in IPv4, la diversa MTU minima richiesta ai link dai due protocolli (68 byte contro 1280 byte), la lunghezza dell'header v6 doppia rispetto alla precedente nonché l'impossibilità per i nodi intermedi IPv6 di frammentare. Per tutti questi motivi si è stabilito che la traduzione dei pacchetti IPv4 deve produrre datagram IPv6 grandi al massimo 1280 byte; in caso contrario, è necessario introdurre la frammentazione del payload in blocchi al più di 1232 byte (1280 meno i 40 della Basic header IPv6 e gli 8 della Fragment Header) prima di effettuare la traduzione vera e propria. La traduzione da IPv6 ad IPv4 è meno complessa poiché attraverso il procedimento di Path MTU Discovery, IPv6 è in grado di determinare la MTU massima supportata dai vari link lungo il percorso; da notare come in presenza di MTU inferiori a 1280, ritrovabili lungo i link IPv4, la pMTU (la MTU dell'intero percorso) viene impostata dal nodo mittente al valore minimo di 1280 byte. Il valore trovato attraverso questo processo è quindi soddisfacente per l'intera rete IPv6 ma potrebbe non esserlo per uno o più link lungo la rete IPv4; il vecchio protocollo di Internet consente tuttavia ai nodi intermedi (e quindi anche al traduttore stesso) presenti lungo il percorso di frammentare ogni qualvolta se ne presenti la necessità.

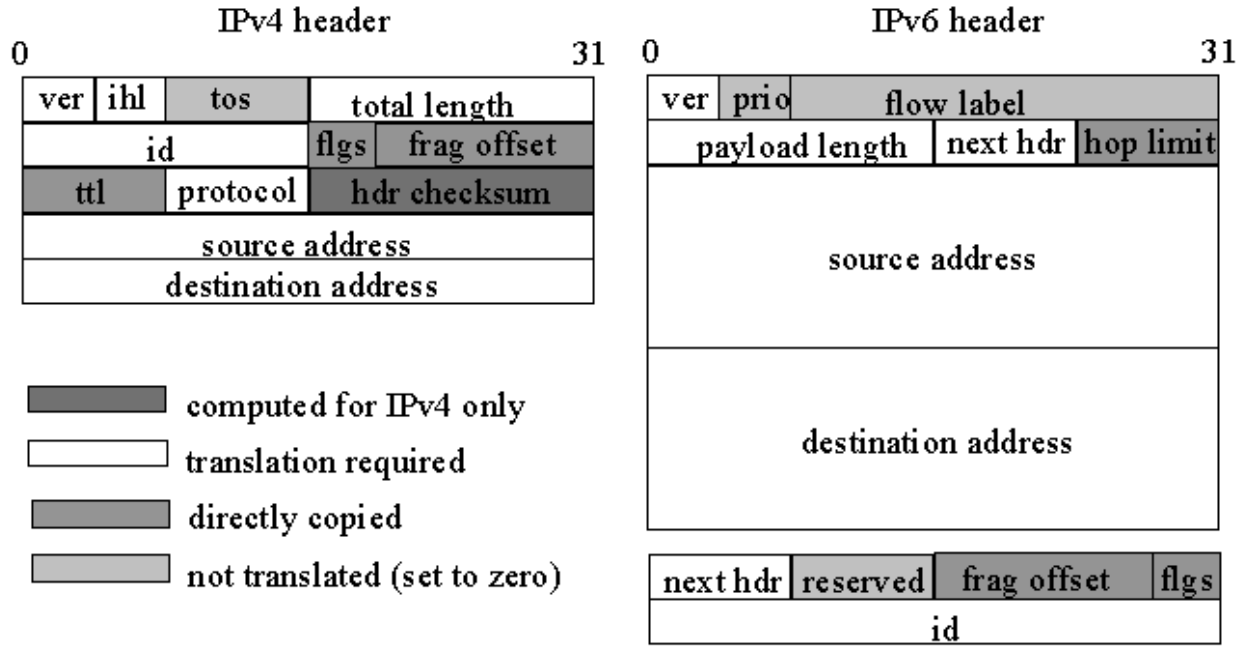

Figura 5.3 Testate IPv4 ed IPv6 a confronto, con le diverse operazioni da effettuare sui campi in caso di traduzione

5.3 La traduzione di messaggi ICMP

In tutti i messaggi ICMP il campo checksum deve essere aggiornato in fase di traduzione poiché ICMPv6, a differenza di ICMPv4, prevede per il calcolo del checksum l'utilizzo di una pseudo-header come accade per UDP e TCP. Inoltre il campo type dei vari pacchetti ICMP deve essere tradotto e, nei messaggi di errori, anche le header IP allegate.

Tutti i messaggi ICMP ricevuti dal traduttore, con l'esclusione di Echo Request and Echo Reply, verranno scartati dal traduttore SIIT; per molti di essi non esiste infatti un corrispondente nell'altra versione del protocollo o si tratta di messaggi destinati solamente ai nodi adiacenti. Per quanto riguarda invece i messaggi di errore, la traduzione è sempre possibile e solamente i messaggi ICMPv4 Redirect e Source Quench devono essere scartati. In particolare è da segnale come, con alcuni accorgimenti, sia possibile effettuare la traduzione anche dei messaggi ICMPv6 "Packet Too Big": sebbene non esista un messaggio analogo in ICMPv4, l'algoritmo SIIT prevede in questi casi la generazione di un messaggio Destination Unreachable di tipo 4, ovvero di richiesta di frammentazione. Inoltre i messaggi ICMP di errore contengono l'header IP del pacchetto che ha generato l'errore, il quale necessita anch'esso di una traduzione come un qualsiasi altro datagram IP. Occorre anche tenere in considerazione il fatto che la traduzione di questi "pacchetti sbagliati" influirà con ogni probabilità anche sulla lunghezza totale del datagram IP incapsulante, rendendo necessario un aggiornamento del campo Total Lenght presente nella testata IPv4.

5.4 La traduzione di messaggi dei protocolli di trasporto

Gli indirizzi di tipo IPv4-translatable sono stati appositamente pensati in modo tale da risultare trasparenti agli algoritmi di controllo di integrità utilizzati a livello Transport; in altre parole il checksum di TCP o UDP ottenuto utilizzando nelle pseudo-header indirizzi IPv4 semplici o indirizzi IPv6 di tipo IPv4-translatable è il medesimo. Nonostante quindi TCP o UDP vìolino il principio di indipendenza dei livelli proprio delle architetture a stati, non devono essere effettuate da SIIT operazioni di traduzione al di sopra del livello Network.

Unica eccezione a quanto detto si ha in presenza di pacchetti UDP IPv4 con checksum pari a zero, ovvero non calcolato. Poiché infatti in IPv6 il campo checksum è necessario, se il pacchetto non è stato frammentato il traduttore deve incaricarsi di calcolarne il valore in fase di traduzione. I pacchetti UDP IPv4 frammentati che non contengono checksum devono invece essere scartati dal traduttore: le operazioni di ricostruzione sarebbero troppo onerose e comunque tradurrebbero con ogni probabilità un pacchetto originato da un malintenzionato (questo tipo di pacchetti è solitamente usato in fase di port scanning).

Capitolo 6
NAT-PT

Come visto prima, SIIT descrive un meccanismo di traduzione che consente la comunicazione tra nodi IPv6-only e nodi IPv4-only effettuando la traduzione dei singoli datagram IPv6 in maniera indipendente, non richiedendo cioè informazioni circa lo stato della sessione. SIIT presuppone che indirizzi IPv4 vengano associati ad host IPv6 senza tuttavia indicare quali debbano essere le strategie da applicare per effettuare l'assegnazione di questi stessi indirizzi. Combinando l'algoritmo definito da SIIT per la traduzione con appositi meccanismi di assegnazione e traduzione dinamica degli indirizzi e appositi ALG, è possibile realizzare una traduzione IPv4-IPv6 completa, non limitandosi ai livelli TCP/UDP e IP ma operando anche a livello Applicativo.
La strategia di migrazione appena descritta è proprio quella alla base di NAT-PT, o Network Address Translation - Protocol Translation (RFC2766), un dispositivo in grado di mantenere traccia delle sessioni e fornire un sistema di routing dei datagram IP completamente trasparente agli host finali. Nello specifico una prima componente di NAT-PT è incaricata di effettuare la traduzione degli indirizzi di livello network (NAT) mentre un secondo modulo gestisce la traduzione dei protocolli (PT), appoggiandosi ad eventuali ALG (Application Level Gateway) per particolari protocolli applicativi.
Come vedremo però nelle pagine seguenti, l'impiego di NAT-PT introduce diverse complicazioni, rendendo di fatto preferibile all'interno delle reti l'utilizzo di strategie di migrazione alternative (doppio stack o tunneling) quando possibile.

6.1 Network Address Translation

Network Address Translation è una meccanismo che permette di associare un certo gruppo di indirizzi IP ad un altro gruppo, fornendo un servizio trasparente di routing dei pacchetti (RFC2663). In questo caso specifico con il termine NAT ci si riferisce alle operazioni di traduzione di indirizzi IPv4 in indirizzi IPv6 e viceversa.
Tutti i dispositivi NAT in particolare definiscono una strategia di assegnazione e traduzione trasparente degli indirizzi; la modalità di funzionamento più semplice prevede un mapping statico degli indirizzi, associando ad ogni host presente all'interno della rete privata un corrispondente indirizzo pubblico. Più frequentemente vengono tuttavia preferiti algoritmi di mapping dinamico, dove ad ogni host "nascosto" dal NAT viene associato temporaneamente un indirizzo globale disponibile. Le fasi di Address Translation sono quindi essenzialmente tre e corrispondono ai tre stati in cui può trovarsi una connessione: creazione, mantenimento e abbattimento. Durante la

prima fase, nota come Address Binding, un indirizzo interno viene associato ad un indirizzo esterno per consentire la traduzione; questa prima operazione ha luogo ogni volta che una nuova connessione viene instaurata. A questo punto per ogni successivo pacchetto ricevuto verrà effettuata una ricerca tra tutti gli indirizzi mappati e quindi una traduzione dell'indirizzo di conseguenza (Address Lookup and Translation); in questo modo tutti i pacchetti appartenenti alla stessa sessione verranno tradotti utilizzando sempre lo stesso indirizzo inizialmente associato. Infine, una volta che l'indirizzo non è più utilizzato da alcuna sessione, viene reso di nuovo disponibile per assegnazioni future (Address Unbinding).

Non è casuale che siano stati introdotti i concetti di 'sessione' e di 'memorizzazione dello stato delle associazioni' tra gli indirizzi V4 e V6: a differenza di SIIT, NAT-PT è un meccanismo di traduzione dei protocolli stateful, che richiede cioè informazioni sulle operazioni effettuate in precedenza per effettuare nuove traduzioni.

6.2 Protocol Translation

Le testate IPv4 e ICMPv4 sono simili alle corrispondenti testate nella versione 6 ma, come già illustrato in precedenza, alcuni campi sono stati eliminati o hanno assunto un significato o una dimensione differente. Compito del NAT-PT è quello di tradurre tutte le testate IP/ICMP dalla versione 4 alla 6 e viceversa, rendendo così possibile la comunicazione end-to-end tra due host basati su differenti versioni dell'Internet Protocol. Le operazioni di Protocol Translation descritte dall'algoritmo SIIT sono valide, con alcuni accorgimenti, anche per questo meccanismo di transizione.

In SIIT l'utilizzo di indirizzi IPv6 esclusivamente di tipo IPv4-translated (cioè con prefisso 0::ffff:0:0:0/96) permette infatti di lasciare inalterato il campo checksum dei protocolli di livello superiore TCP e UDP; NAT-PT invece utilizza un generico prefisso di 96 bit per indicare quali pacchetti dovranno essere indirizzati al traduttore, rendendo però di fatto necessario ricalcolare per ogni pacchetto TCP, UDP e ICMP il valore del campo checksum una volta tradotti gli indirizzi.

Inoltre la traduzione degli indirizzi di livello network in NAT-PT non è immediata ma deve essere effettuata sulla base dello spazio di indirizzamento utilizzabile dal traduttore e dello stato dei mapping, cioè delle associazioni tra indirizzi IPv4 ed IPv6 effettuate in precedenza.

6.3 Le tipologie di NAT-PT

Così come per IPv4 sono state definite diverse modalità di funzionamento di NAT, anche NAT-PT non consiste in un unico meccanismo; sono stati invece definiti più algoritmi in grado di rispondere in maniera ottimale alle diverse esigenze. Le diverse tipologie di NAT-PT sono tuttavia

essenzialmente raggruppabili in due grandi famiglie: Traditional NAT-PT e Bi-Directional NAT-PT.

Prima di procedere nell'analisi di questi diversi meccanismi, è opportuno tuttavia illustrare la strategia adottata da tutti i dispositivi NAT-PT per operare all'interno di una rete in maniera trasparente e garantire nel contempo l'effettiva traduzione dei dati. Per far sì che i pacchetti IPv6 indirizzati alla rete V4 vengano correttamente ricevuti dal traduttore, viene infatti inizialmente specificato sugli host un prefisso di 96 bit da concatenare con l'indirizzo IPv4 di destinazione (es. PREFIX::192.168.0.1) da utilizzare per tutti i datagram destinati alla rete esterna. Sarà quindi il NAT-PT stesso a pubblicizzare un'opportuna regola di instradamento, indicando sé stesso come router predefinito al quale inoltrare tutti i pacchetti che necessitano di traduzione (ovvero indirizzati al prefisso prestabilito).

6.3.1 Traditional NAT-PT

Traditional-NAT-PT consente ad host presenti in una rete IPv6 di accedere ad una rete IPv4; tuttavia questa modalità di funzionamento prevede solo sessioni unidirezionali, originate esclusivamente dagli host della rete V6. Esistono in particolare due modalità di funzionamento del Traditional-NAT-PT, chiamate Basic-NAT-PT e NAPT-PT.

6.3.1.1 Basic NAT-PT

Con il Basic-NAT-PT, un blocco di indirizzi IPv4 viene assegnato al nodo traduttore ed è utilizzato per allocare indirizzi per gli host della rete V6 che inizializzano sessioni comunicative verso la rete V4. Se si suppone che gli indirizzi IPv4 disponibili siano sufficienti, è possibile prevedere una associazione uno-ad-uno tra gli host V4 e V6; più comunemente, tuttavia, si hanno più nodi finali IPv6 di quanti indirizzi IPv4 si abbiano a disposizione.

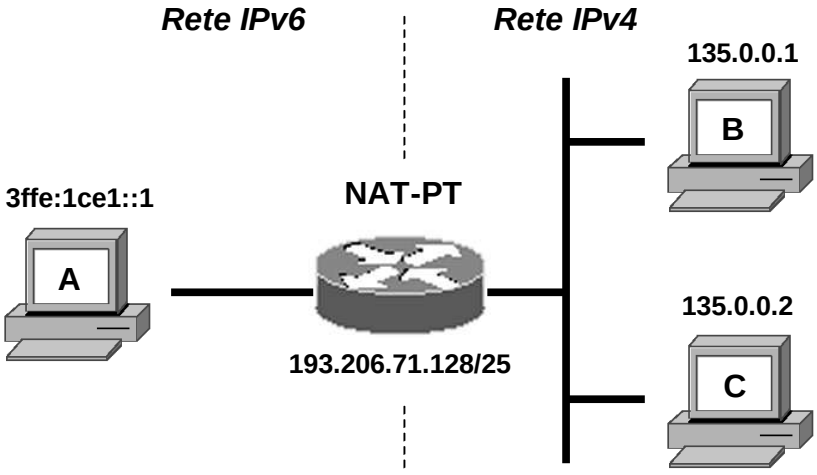

Figura 6.1 NAT-PT utilizzato per interconnettere una rete IPv4 con una rete IPv6

Nella situazione ‘tipica’, illustrata dall’esempio sopra, il dispositivo NAT-PT è posto a cavallo delle due reti da far interoperare. Nel momento in cui il Nodo IPv6 A decide di comunicare con il nodo IPv4 B crea un pacchetto indirizzato all’apposito prefisso, seguito dall’indirizzo IPv4 di destinazione, e quindi lo inoltra al traduttore. Il NAT-PT a questo punto analizza il datagram ricevuto e, sulla base delle informazioni sui mapping effettuati fino a quel momento, stabilisce se il pacchetto ricevuto è di inizializzazione della sessione o se appartiene ad una sessione già esistente. Nel primo caso provvederà, prima di tradurre il pacchetto, ad assegnare un nuovo indirizzo tra quelli disponibili memorizzando quindi il nuovo stato delle associazioni IPv6-mittente e IPv4-pubblico. Il traffico di ritorno verrà tradotto di conseguenza, sulla base del mapping effettuato in precedenza.

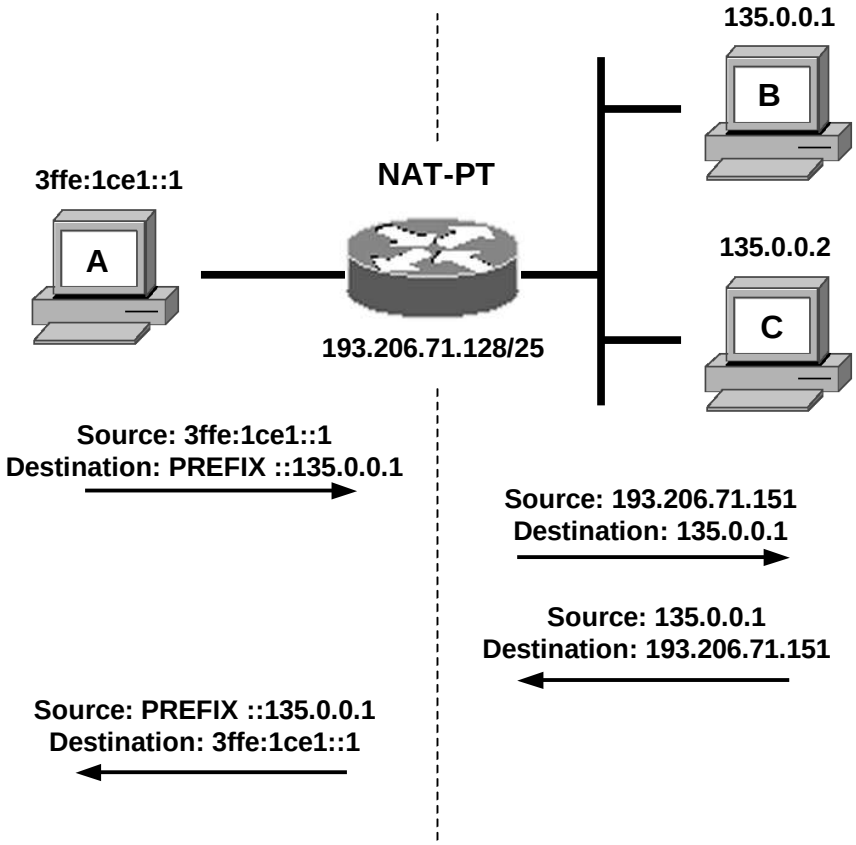

Figura 6.2 Esempio di procedura di traduzione con NAT-PT (sul router viene memorizzata l’associazione 3ffe:1ce1::1 – 193.206.71.151)

Il funzionamento qui brevemente descritto è piuttosto semplice ma presenta alcuni limiti, superati in parte dalle altre tipologie di NAT-PT esistenti. In particolare il primo problema è sicuramente dato dal numero limitato di connessioni effettuabili contemporaneamente, limitate dal numero massimo di indirizzi IPv4 disponibili ai fini della traduzione; se tentano di connettersi verso la rete esterna più host IPv6 di quanti siano gli indirizzi IPv4, il sistema non è in grado di fornire

connettività a tutti. Inoltre la sessione può essere inizializzata solamente da un host IPv6 verso uno V4 e non viceversa, creando non poche difficoltà in talune situazioni reali di lavoro.

6.3.1.2 NAPT-PT

NAPT-PT, acronimo di Network Address Port Translation – Protocol Translation, estende la nozione di traduzione operando anche sugli 'identificatori' di livello transport (i port number TCP o UDP, i codici delle richieste ICMP…) per permettere ad una rete di più host IPv6 di comunicare con una rete IPv4 utilizzando un unico indirizzo V4. I diversi flussi comunicativi originati dai vari host V6 vengono cioè identificati e smistati utilizzando le informazioni contenute a livello di trasporto e il processo stesso di traduzione coinvolge sia gli indirizzi IP che le porte TCP/UDP. NAPT-PT risolve uno dei principali problemi di NAT-PT, soluzione strettamente legata al numero di indirizzi IPv4 disponibili per i fini di traduzione. Quando infatti il pool di indirizzi disponibili per NAT-PT è stato esaurito, nessun ulteriore nodo IPv6 è in grado di aprire nuove sessioni comunicative verso la rete esterna. NAPT-PT invece, pur utilizzando un solo indirizzo, consente un massimo di 64mila sessioni TCP e altrettante UDP prima di esaurire tutte le porte assegnabili.

Per comprendere meglio il meccanismo di funzionamento di questa soluzione, riprendiamo l'esempio illustrato in precedenza installando sul router NAPT-PT anziché NAT-PT e mappando tutti gli host V6 sull'unico indirizzo IPv4 193.206.71.151.

Come nel caso precedente, tutti i datagram destinati alla rete V4 esterna verranno inviati ad un indirizzo IPv6 opportunamente costruito concatenando l'indirizzo IPv4 destinatario ed il prefisso di 96 bit indicato dal NAT-PT stesso. Supponiamo che il nodo IPv6 A voglia instaurare una sessione TCP con il nodo IPv4 C: se il routing interno alla rete è stato opportunamente configurato, i dati destinati alla rete V4 verranno inoltrati al traduttore NAT-PT. Questi provvede quindi ad assegnare una delle porte TCP disponibili per modificare di conseguenza la coppia Indirizzo-porta sorgente del pacchetto ricevuto ed infine lo inoltra verso il nodo C.

A questo punto è stata memorizzata nel NAPT-PT l'associazione tra l'indirizzo e la porta mittente e la corrispondente accoppiata IPv4 pubblica: tutto il traffico originato successivamente dal nodo IPv6 nonché quello di ritorno verranno identificati come appartenenti alla stessa sessione e tradotti di conseguenza rispettando l'associazione effettuata in fase di instaurazione della connessione.

Adottando questa soluzione è possibile anche consentire l'instaurazione di sessioni originate dall'esterno, limitandosi tuttavia ad un mapping statico delle associazioni host-porta. Questo significa che le sessioni che raggiungeranno dall'esterno il NAPT-PT aventi una determinata porta di destinazione verranno tutte ridirezionate verso lo stesso host IPv6.

La soluzione NAPT-PT può essere ulteriormente estesa combinandola con il Basic-NAT-PT, in modo che non uno solo bensì un pool di indirizzi IPv4 possano essere utilizzati insieme alle porte in fase di traduzione.

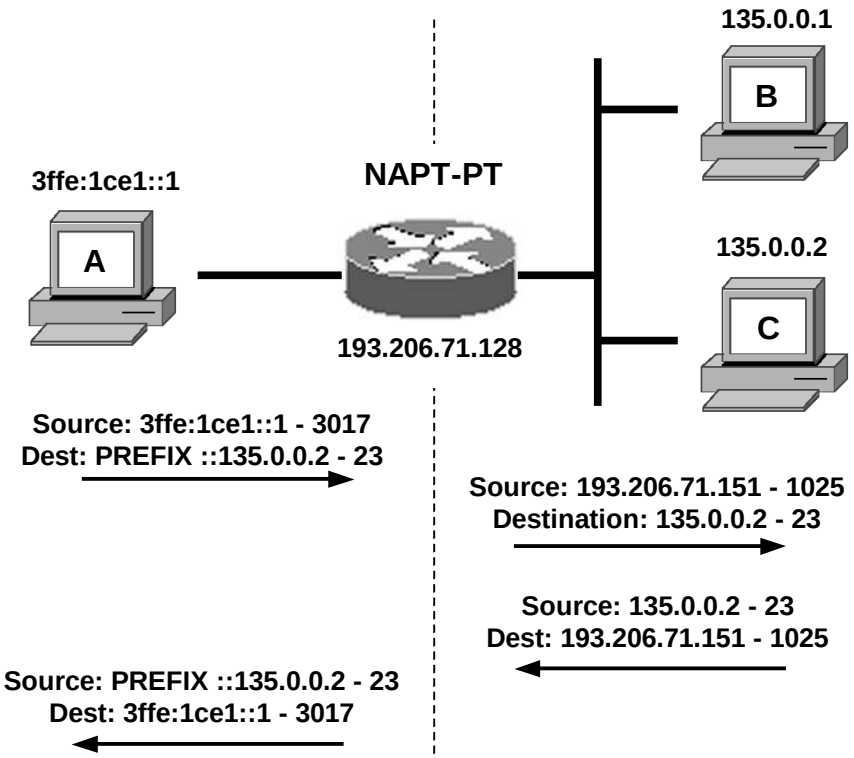

Figura 6.3 Esempio di procedura di traduzione in presenza di NAPT-PT (sul router viene memorizzata l'associazione 3ffe:1ce1::1-3017 – 193.206.71.151-1025)

6.3.2 Bi-directional NAT-PT

A differenza di quanto accade con il Basic NAT-PT, utilizzando il meccanismo Bi-Directional NAT-PT le sessioni possono essere inizializzate sia da host della rete IPv4 che della rete IPv6 e i nodi presenti nelle due aree possono comunicare tra loro utilizzando DNS per la risoluzione degli indirizzi. In questo caso un apposito ALG per DNS deve essere utilizzato per facilitare il mapping tra nomi e indirizzi. Più nello specifico, il DNS-ALG deve essere in grado di modificare Query DNS traducendo gli indirizzi in esse contenute in base alle associazioni effettuate dal NAT-PT.

6.4 Gli ALG

Gli Application Level Gateway (ALG) sono degli agenti pensati appositamente per una applicazione, in grado di consentire ad un nodo V6 di comunicare con la controparte V4 e viceversa. Alcune applicazioni infatti inseriscono indirizzi di livello IP nel payload dei dati

trasmessi; NAT-PT, operando a livello TCP/IP, non è in grado di tradurli. Per questo i dispositivi traduttori NAT operano spesso insieme ad appositi ALG per fornire una comunicazione trasparente end-to-end a molte applicazioni di questo tipo: tra queste sicuramente vanno ricordate DNS e FTP. Ogni ALG opera in maniera indipendente ed è studiato per uno specifico protocollo e, in base alla modalità di funzionamento e alle esigenze del protocollo stesso, può utilizzare o meno informazioni sullo stato del traduttore. In ogni caso tutti gli ALG, per individuare i pacchetti da tradurre, utilizzano gli indentificatori di livello TCP/UDP: in base quindi al numero di porta utilizzato dall'applicazione (es. 53 per DNS, 21 per FTP etc..), il NAT-PT è in grado di identificare a quale ALG passare il payload da tradurre.

6.4.1 DNS

Poiché DNS è un servizio di livello applicativo che prevede lo scambio di messaggi contenenti indirizzi di livello Network, l'utilizzo di IPv4 piuttosto che IPv6 sui nodi interessati alla comunicazione non è affatto trasparente a DNS. In particolare, come visto in precedenza, sono state definite due diverse tipologie di record conservati all'interno dei server DNS: gli indirizzi IPv4 sono memorizzati all'interno di record di tipo A, mentre AAAA è utilizzato in presenza di indirizzi IPv6. In linea generale l'Application Level Gateway per il protocollo DNS è quindi incaricato di tradurre i messaggi scambiati tra le due reti, modificando opportunamente gli indirizzi scambiati e garantire così la connettività tra host IPv4-only e IPv6-only.

6.4.1.1 DNS-ALG per le connessioni in ingresso (da V4 a V6)

Poiché il NAT-PT si trova a bordo del router che interconnette le due reti, i datagram contenenti tutte le richieste di risoluzione di indirizzi originate da host IPv4 e destinate alla rete V6 passeranno attraverso il nodo traduttore. Per rendere compatibile la Query DNS originata dal nodo IPv4 con il dominio di destinazione (la rete IPV6), l'ALG traduce il tipo di record 'A' in 'AAAA' nelle richieste di risoluzione dell'indirizzo a partire dal nome mentre, nelle Query di risoluzione "inverse", la stringa "IN-ADDR.ARPA" viene sostituita con "IP6.INT" e gli ottetti dell'indirizzo IPv4 posti davanti ad essa vengono sostituiti da quelli del corrispondente indirizzo V6 mappato, se esiste. Anche le risposte DNS originate dal server nella rete V6 e destinate al nodo IPv4 richiedente attraversano il router NAT-PT e vengono intercettate dal ALG; in questo caso i record di tipo A vengono tradotti in record AAAA e l'indirizzo IPv4 risolto viene sostituito dall'indirizzo IPv4 mappato internamente dal traduttore. In entrambi i casi, se non è presente alcun mapping per l'indirizzo da tradurre, viene effettuata e memorizzata una nuova associazione IPv4-IPv6 per l'indirizzo in questione.

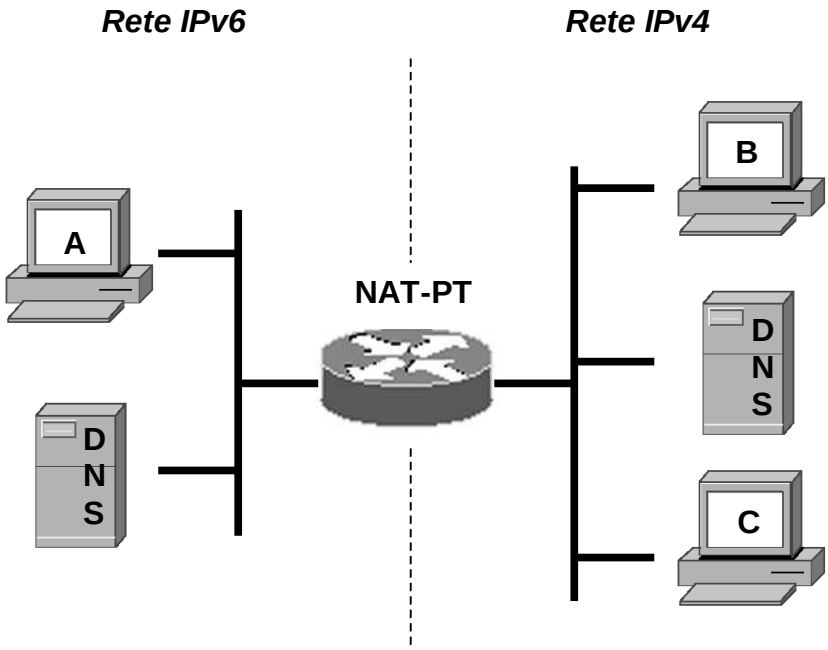

Figura 6.4 Esempio di reti IPv4 e IPv6 dotate di server DNS locali e interconnesse con NAT-PT

Facendo riferimento alla rete sopra presentata, supponiamo ad esempio che il nodo IPv4 C cerchi di inizializzare una sessione con il nodo A effettuando una Query DNS (record di tipo A) per il nodo A. La richiesta verrà quindi indirizzata al server locale e poi propagata al server DNS della rete IPv6, attraversando il traduttore NAT-PT. Sarà proprio l'ALG installato su quest'ultimo nodo ad intercettare e tradurre la query di tipo A in AAAA, inoltrandola poi al server DNS posto all'interno della rete V6. Analogamente, il tipo AAAA e l'indirizzo IPv6 presenti nella risposta verranno rispettivamente convertiti in A e nell'indirizzo IPv4 associato. All'interno del DNS-ALG inoltre viene memorizzato il mapping tra l'indirizzo V6 dell'host A e quello V4 temporaneamente assegnato in fase di traduzione della DNS Reply. A questo punto tutti i pacchetti successivi inviati dall'host C all'indirizzo IPv4-mapped indicato verranno tradotti di conseguenza dal NAT-PT e consegnati all'host A e la comunicazione potrà procedere normalmente.

Poiché i mapping effettuati dal traduttore variano nel tempo, il valore del campo TTL presente in tutti i RR DNS che attraversano il NAT-PT dovrebbe essere posto a 0 in maniera tale che i server e i client DNS non inseriscano nella cache gli indirizzi assegnati temporaneamente. In presenza invece di indirizzi mappati staticamente, il problema non sussiste e il valore di TTL non dovrebbe essere modificato dall'ALG.

6.4.1.2 DNS-ALG per le connessioni in uscita (da V6 a V4)

I nodi IPv6 possono ricavare l'indirizzo di nodi V4 utilizzando i server DNS presenti nel dominio IPv4 o interni alla rete V6. Nel caso in cui questi ultimi contengano in memoria l'indirizzo associato al nome ricercato, la Query DNS non deve uscire dal dominio IPv6 attraverso il traduttore e quindi non è necessario l'intervento del DNS-ALG, come avviene invece nel caso in cui non sia noto l'indirizzo. Per questo motivo è consigliabile che i server DNS presenti nel dominio IPv6 non si limitino a memorizzare in cache i mapping per gli host interni ma, possibilmente, anche per qualche nodo esterno.

Abbiamo visto che in generale un nodo IPv6 utilizza, per comunicare con la rete esterna, uno specifico prefisso di 96 bit anteposto all'indirizzo IPv4 destinatario. Con il meccanismo descritto di seguito ed utilizzato dai DNS-ALG posti a bordo dei router NAT-PT, è possibile sfruttare questo prefisso senza alcuna configurazione sui nodi finali.

Supponiamo ad esempio che il nodo A, per iniziare una sessione comunicativa con il nodo C, cerchi di risolvere l'indirizzo di quest'ultimo effettuando una Query DNS (ovviamente di tipo AAAA, dal momento che A è un host IPv6-only). Poiché l'interfaccia del nodo C potrebbe avere assegnati sia indirizzi IPv4 che IPv6, l'ALG del traduttore provvede ad inoltrare la query AAAA senza modificarla e quindi invia anche una seconda richiesta di tipo A verso lo stesso nodo. Se esiste nei server DNS del dominio un record di tipo AAAA, questo verrà ritornato al NAT-PT e quindi inoltrato al host richiedente senza modifiche. Se invece per il nodo C viene ritornato un RR di tipo A, il DNS-ALG traduce la risposta in una di tipo AAAA, aggiungendo all'indirizzo IPv4 riportato il prefisso appropriato ed inoltrandola quindi verso il nodo A. A questo punto l'host A può utilizzare questo indirizzo per le comunicazioni come se stesse comunicando con un qualsiasi host IPv6 e i server DNS V6 possono persino inserire nella cache questo mapping per l'host C.

6.4.2 FTP

Poiché la sessione di controllo FTP contiene, all'interno dei messaggi scambiati, l'indirizzo IP e le porte TCP utilizzate per lo scambio dati, un ALG apposito è richiesto a bordo dei router NAT-PT per fornire una comunicazione trasparente a livello applicativo per questa applicazione così diffusa. Come spiegato in precedenza, i comandi PORT e PASV specifici di IPv4 sono stati sostituiti dai comandi EPRT e EPSV (si veda a riguardo RFC2428), aventi la medesima funzione ma compatibili sia con IPv4 che con IPv6.

6.4.2.1 FTP-ALG per le connessioni in ingresso (da IPv4 a IPv6)

Un host IPv4 può avere o meno le estensioni EPRT e EPSV implementate nel client FTP.

Se la sessione FTP è originata utilizzando i comandi PORT o PASV, il gateway di livello applicativo deve tradurre i comandi rispettivamente con EPRT e EPSV prima di inoltrarli verso il nodo IPv6 e, analogamente, le risposte EPSV originate dal nodo V6 devono essere tradotte in risposte di tipo EPSV. Il comando PORT di un nodo V4 viene tradotto invece con EPRT, seguito dal numero 2 per indicare il tipo di protocollo da utilizzare (IPv6) e dall'indirizzo V6 mappato dal NAT-PT. Il numero di porta TCP indicato dal comando PORT viene anch'esso tradotto in decimale e indicato nel comando EPRT. Il comando PASV viene invece tradotto in EPSV 2, dove il numero indica sempre la versione del protocollo utilizzata (IPv6). Anche le risposte EPSV inviate dal nodo IPv6 vengono tradotte in risposte PASV prima di essere inoltrate all'host destinatario V4.
Se invece l'host V4 utilizza i comandi estesi per le sessioni FTP, l'ALG è solamente incaricato di tradurre i parametri dei comandi e non i comandi stessi. Il protocol number deve quindi essere cambiato da 1 in 2 e gli indirizzi IP V4 negli indirizzi V6 corrispondenti assegnati dal NAT-PT. Le porte indicate devono invece essere tradotte esclusivamente in presenza di un dispositivo NAPT-PT

6.4.2.2 FTP-ALG per le connessioni in uscita (da IPv6 a IPv4)

Se la sessione FTP viene inizializzata da un host V6, l'ALG ha due possibili approcci da utilizzare. Nel primo caso, l'FTP-ALG non deve modificare i comandi estesi EPRT e EPSV, traducendo semplicemente gli indirizzi, le porte ed i numeri di protocollo indicati come argomento; lo stesso dicasi per le risposte EPSV inviate dai noti V4. Questa modalità di funzionamento è quella preferibile, poiché più semplice e pienamente compatibile con le nuove specifiche del protocollo FTP. Tuttavia questo approccio impone agli host V4 un aggiornamento delle applicazioni FTP per supportare le estensioni EPRT e EPSV.
Per evitare questo aggiornamento "forzato", è possibile prevedere che l'ALG traduca i comandi EPRT e EPSV con PORT e PASV e i rispettivi parametri in base ai mapping effettuati dal nodo NAT-PT e lo stesso dicasi, analogamente, per le risposte PASV originate dai nodi V4. Tuttavia il comando esteso EPSV ALL, non essendo traducibile, potrebbe essere ricevuto dal nodo V4 e portare quindi ad errori in grado di far terminare prematuramente la sessione.

6.4.2.3 FTP-ALG ed i livelli sottostanti

Poiché tutte le traduzioni dei payload FTP considerate sono effettuati su dati codificati in ASCII, è probabile che il processo porti a nuovi pacchetti aventi dimensioni differenti da quelle iniziali. Se la dimensione non è variata, solamente il checksum TCP deve essere ricalcolato in seguito alla traduzione del payload FTP. Se invece la nuova dimensione del pacchetto è differente, anche i

numeri di sequenza TCP devono essere modificati per riflettere le modifiche alla lunghezza del payload, nonché il campo Packet_length nella testata IPv4 o Payload_length nell'header IPv4. In quest'ultimo caso una tabella deve essere utilizzata dall'ALG a bordo del NAT-PT, per tener traccia e correggere di conseguenza i numeri di sequenza e di acknowledgement nelle testate TCP dei pacchetti della connessione di controllo. All'interno di questa tabella devono essere memorizzati gli indirizzi e le porte del mittente e del destinatario, insieme al delta (differenza) dei numeri di sequenza dei pacchetti uscenti e di quelli entranti.

6.5 Le problematiche introdotte da NAT-PT

L'utilizzo del meccanismo di transizione NAT-PT introduce alcune problematiche all'interno delle reti in cui è utilizzato; alcune questioni sono riscontrabili anche nel NAT per IPv4 e sono proprie di ogni meccanismo di Network Address Translation, mentre altre sono specifiche di questa strategia di migrazione verso IPv6.

6.5.1 Limiti alla topologia della rete

Essendo NAT-PT un meccanismo stateful, è necessario che tutte le richieste e le risposte appartenenti ad una stessa sessione vengano inoltrate attraverso il medesimo router. Per garantire che questo avvenga sempre, viene solitamente realizzata un'unica connessione tra le due reti da mettere in comunicazione, in modo che il NAT-PT risulti l'unico router utilizzabile dai pacchetti che devono essere tradotti.

6.5.2 Fault tolerance e prestazioni

L'utilizzo di un unico router NAT-PT attraverso cui far transitare tutti i dati di una stessa connessione non impone solamente limiti alla topologia della rete ma introduce anche alcuni problemi legati all'affidabilità e alle prestazioni. Se si considera infatti che per ogni pacchetto il NAT-PT deve effettuare alcune operazioni matematiche, nonché ricercare voci in un database delle associazioni, è facile intuire come, in presenza di traffico comunicativo elevato, questo nodo possa diventare il "collo di bottiglia" dell'intera rete. Tuttavia NAT-PT non presenta solo problemi in termini di prestazioni ma anche di affidabilità della rete stessa: in caso di guasto del dispositivo infatti, nessuno dato diretto alla rete esterna sarà in grado di raggiungere il destinatario poiché non esistono strade alternative.

6.5.3.Complessità di Protocol Translation

Come più volte è stato ricordato, la traduzione da IPv4 a IPv6 (o viceversa) non è immediata poiché alcuni campi presenti in una delle due testate non trovano un corrispondente nell'altra o, se

esiste, hanno lunghezze o significati diversi. Inoltre informazioni di livello Transport vengono utilizzate per operare traduzioni anche ai livelli al di sopra di Internet Protocol. Nello specifico le porte TCP o UDP vengono utilizzate sia per distinguere le diverse sessioni comunicative (NAPT-PT) che per identificare il protocollo di livello applicativo trasportato (da tradurre eventualmente mediante l'apposito ALG).

6.5.4 Esaurimento dello spazio di indirizzamento e attacchi DDOS

Se il numero di host IPv6 è superiore al numero di indirizzi pubblici disponibili per il traduttore, un alto numero di connessioni può rapidamente portare all'esaurimento dello spazio di indirizzamento e, di conseguenza, alla mancanza di connettività per alcuni host. La soluzione NAPT-PT illustrata supera questa limitazione del NAT-PT tradizionale per quanto riguarda le connessioni uscenti dalla rete V6 ma non per quelle entranti. Inoltre l'utilizzo di un mapping dinamico degli indirizzi presenta alcune insidie in fase di Address Unbinding. Per il nodo traduttore infatti è piuttosto semplice riconoscere se un pacchetto appartiene ad una connessione già esistente o se è una richiesta di instaurazione di una nuova sessione, semplicemente effettuando una ricerca nella tabella dei binding esistenti. Più difficile è invece riconoscere quando una connessione non è più attiva e le risorse ad essa assegnate possono essere pertanto liberate; poiché UDP non prevede infatti il concetto di sessione e i pacchetti TCP di RST della stessa potrebbero andar perduti, viene solitamente impostato un tempo massimo di attesa, un Time-out dopo il quale considerare la comunicazione conclusa e de-allocare gli indirizzi assegnati.
Una strategia di questo tipo risulta molto utile anche in caso di attacchi sferrati ad un Bi-directional NAT-PT da parte di malintenzionati. Un attaccante potrebbe infatti voler interrompere il servizio di connettività verso l'esterno effettuando un attacco di tipo DOS; eseguendo infatti un gran numero di Query DNS, al solo scopo di richiedere continuamente nuovi binding, è possibile esaurire lo spazio di indirizzamento a disposizione del traduttore. Per contrastare questo tipo di attacchi è necessario verificare l'effettivo utilizzo degli binding tra indirizzi e deallocare quelli non utilizzati per un certo periodo di tempo. Inoltre, in presenza di NAPT-PT, è consigliabile riservare sempre un indirizzo IPv4 per le sessioni uscenti per minimizzare gli effetti di questo tipo di attacchi sulle sessioni originate dalle rete V6.

6.5.5 Incompatibilità con meccanismi di sicurezza end-to-end

Una delle limitazioni principali del meccanismo NAT-PT consiste nell'impossibilità di utilizzare meccanismi di sicurezza end-to-end a livello Network. Inoltre, poiché solitamente questi sistemi si basano sugli indirizzi IP, anche alcuni meccanismi di livello transport o application non

funzionano in presenza di NAT-PT; in una comunicazione end-to-end basata su questo meccanismo di transizione non è quindi utilizzabile DNSSec né il più diffuso IPSec.

6.5.6 Frammentazione

Un aspetto che non è stato volutamente affrontato riguarda la frammentazione dei pacchetti in presenza di NAT-PT e, soprattutto, NAPT-PT. Come ricordato in RFC3027, i semplici traduttori di porte sono in grado solamente di tradurre il primo frammento di un pacchetto frammentato poiché in quelli successivi il numero di porta utilizzato non è più riportato. Questo impedisce quindi ai dispositivi NAPT-PT di tradurre pacchetti frammentati; in teoria sarebbe possibile memorizzare lo stato dei frammenti in transito, a patto però che il primo pacchetto arrivi sempre prima degli altri. Un problema per tutti i NAT-PT inoltre si ha in presenza di pacchetti UDP IPv4 frammentati con il checksum di livello transport non calcolato: poiché in IPv6 il campo è stato reso obbligatorio, la traduzione dei frammenti non è possibile.

Una soluzione ad entrambi questi problemi potrebbe consistere nel riassemblare sul nodo traduttore NA(P)T-PT tutti i pacchetti frammentati provenienti sia dalla rete V4 che V6; questo introdurrebbe tuttavia notevoli ritardi nella comunicazione end-to-end ed un carico di lavoro aggiuntivo per il nodo intermedio.

Capitolo 7

La realizzazione pratica del nodo traduttore NAT-PT

Scopo dell'attività di laboratorio svolta nel Laboratorio Reti dell'Università, attivo presso la sede di Mantova, era quello di verificare l'effettiva realizzabilità di un nodo traduttore di protocolli ed, eventualmente, di verificarne il funzionamento. Diverse ore sono state quindi trascorse navigando su Internet alla ricerca di informazioni sulle implementazioni NAT-PT esistenti. In un secondo momento i diversi software individuati sono stati quindi installati e, se possibile, provati realizzando una semplice rete all'interno del laboratorio stesso.

7.1 Le implementazioni esistenti

In Internet sono disponibili diverse implementazioni software di traduttori NAT-PT, sviluppate prevalentemente in ambiti universitari o di ricerca. Due aspetti in particolare accomunano però quasi tutti questi programmi e pongono alcuni interrogativi circa il reale utilizzo in ambito produttivo di questa soluzione. Se si esclude la soluzione commerciale fornita da Cisco all'interno dei propri dispositivi router, tutte le altre implementazioni sono infatti piuttosto instabili e incomplete; inoltre, negli ultimi due anni, nessuna è stata ulteriormente aggiornata, migliorata o modificata. Proprio per questi motivi le difficoltà incontrate durante le prove in laboratorio sono state numerose e non è sempre stato possibile ovviare agli inconvenienti che si sono presentati.

7.1.1 Cisco IOS

Cisco (http://www.cisco.com) è un'azienda leader nel settore del networking e le sue apparecchiature sono utilizzate in tutto il mondo. Tutti i router Cisco utilizzano un sistema operativo che, seppur presentando funzionalità differenti in base alla versione installata, offre agli amministratori di sistema un interfaccia di configurazione testuale decisamente completa ed uniforme. In particolare, a partire dalla versione 12.2 del sistema operativo IOS i dispositivi Cisco supportano la funzionalità di NAT-PT, includendo anche gli ALG per DNS e FTP.

7.1.2 NAT-PT module for Click

Click (http://pdos.csail.mit.edu/click) è un progetto sviluppato dal MIT, funzionante sotto Linux, che permette di realizzare router e traduttori modulari. Utilizzando l'apposito linguaggio di Click è infatti possibile realizzare un file di configurazione in cui definire il funzionamento del router, ovvero il percorso che ogni pacchetto deve seguire e le diverse operazioni che devono essere compiute (traduzioni, visualizzazione dei dati, memorizzazione in un buffer…). Il punto di forza di

questa soluzione consiste nella sua modularità: il programma è cioè composto da tanti sottoprogrammi, ciascuno con una ben definita funzionalità, che possono essere combinati tra loro con estrema semplicità per ottenere un router completo. In particolare un modulo software chiamato GT64, scritto nel 2001, consente di realizzare alcune operazioni fondamentali per un traduttore NAT-PT attraverso tre componenti basilari: un traduttore di protocollo IP dalla versione 4 alla versione 6 (PT46) e un traduttore da IPv6 ad IPv4 (PT64), basati sulle specifiche di SIIT, ed un traduttore di indirizzi e porte (Address and Port Translator, APT). Nel 2003 due studenti colombiani hanno realizzato due moduli in grado di effettuare la traduzione del protocollo FTP e delle richieste DNS e, combinandoli con GT64, hanno realizzato un traduttore NAT-PT completo e funzionante. Sebbene questo codice non sia stato più aggiornato e presenti diversi problemi di instabilità, si è tuttavia dimostrato essere l'unico effettivamente installabile e funzionante ed è stato pertanto utilizzato nelle nostre prove in laboratorio.

7.1.3 Kame project

KAME (http://www.kame.net) è un progetto Giapponese nato con lo scopo di realizzare un valido stack di rete IPv6 per i sistemi operativi Unix BSD, con un occhio di riguardo per gli aspetti legati alla sicurezza. La qualità del prodotto realizzato ha convinto gli sviluppatori a proseguire nel lavoro e attualmente lo stack IPv6 nei sistemi operativi FreeBSD e NetBSD deriva direttamente dal progetto KAME. Insieme al software incluso nel nucleo del sistema, sono stati sviluppati anche numerosi strumenti per il nuovo protocollo; per quel che riguarda in particolare i meccanismi di transizione, con KAME/FreeBSD è possibile attivare sia il doppio stack che il tunneling. Inoltre è disponibile faithd, un demone Unix che implementa l'algoritmo di traduzione TRT. Anche NAT-PT è stato inizialmente implementato ma, a partire dal 2003, il software non è più stato aggiornato e non è pertanto più in grado di funzionare con le versioni attuali di FreeBSD.

7.1.4 Microsoft NAPT (University of Washington)

Sviluppato da tre ricercatori del dipartimento di Computer Science and Engineering dell'Università di Washington, NAPT (http://www.cs.washington.edu/research/networking/napt/) è stata la prima implementazione funzionante mai realizzata basata sul meccanismo di transizione da noi studiato. Presentato per la prima volta nel 1998 durante la conferenza annuale del settore USENIX, questa implementazione ha permesso di mostrare l'effettivo funzionamento del meccanismo di transizione NAT-PT ed è stato altresì possibile effettuare alcune misurazioni in termini di prestazioni. Funzionante sotto Windows NT, questo software è stato successivamente acquisito da Microsoft ed inserito all'interno del progetto della casa di Redmond per lo sviluppo delle tecnologie legate ad IPv6. Attualmente non sono più disponibili on-line i sorgenti originali.

7.1.5 Ultima (BT - British Telecom)

Ultima (http://www.ipv6.btexact.com) è un software funzionante sotto FreeBSD 4 scritto da British Telecom. Le notizie a riguardo purtroppo sono scarse e il sito della compagnia britannica non fornisce particolari informazioni. Dalle notizie ricavate da uno studio pubblicato nel 2002 dalla facoltà di scienze dell'Università di Bruxelles si può evincere che l'installazione di Ultima richiede la modifica un file del kernel di FreeBSD e che dispone persino di una interfaccia Web per la configurazione. Abbiamo richiesto a BT il sorgente di Ultima per poterlo testare in laboratorio, ma ci è stato purtroppo risposto che il programma non viene più distribuito.

7.1.6 ETRI NAT-PT (Korea)

L'Istituto Coreano di Ricerca Elettronica e Telecomunicazioni (Korean ETRI, http://www.ipv6.or.kr) ha utilizzato il codice di Ultima come punto di partenza per realizzare una nuova implementazione di NAT-PT funzionante sotto Linux in User mode. Scritto per operare su una macchina configurata come router e basata sul vetusto Kernel 2.4.0-test9, questo software non è più stato aggiornato ed è rimasto sempre ad uno stato sperimentale. Le novità introdotte dal kernel 2.6, la scarsa stabilità del software e la difficoltà di configurazione hanno reso impossibile la compilazione e il funzionamento di questo programma sulle macchine presenti in laboratorio.

7.1.7 ISPRAS NAT-PT

Nel biennio 2002-2003 un progetto di ricerca dell'ISPRAS (Institute for System Programming Russian Academy of Science) di Mosca ha realizzato due software per FreeBSD 4 che implementano rispettivamente gli algoritmi SIIT e NAT-PT (http://ipv6.ispras.ru/linux_nat-pt.html). I tentativi di compilazione sulle versioni attuali di FreeBSD non sono però andati a buon fine; purtroppo il codice non è più stato aggiornato e le pagine, completamente scritte in cirillico, non sono state di grande aiuto.

7.2 Le prove in laboratorio con Click

Per testare le funzionalità della soluzione NAT-PT è stata realizzata in laboratorio una rete estremamente semplice ed è stato installato e configurato appositamente un nodo traduttore basato su Click. Come visto sopra infatti, tra tutte le soluzioni esistenti Click è stata l'unica soluzione in grado di funzionare con i sistemi operativi attualmente diffusi ed utilizzati. In particolare la versione corrente 1.4.3 del software si integra perfettamente con Linux 2.4 ma, proprio in questi mesi, gli sviluppatori stanno lavorando per portare il software anche su Linux 2.6 e FreeBSD. Il pacchetto principale Click include tutti i componenti software fondamentali per la gestione e lo smistamento del traffico di rete, la ricezione e l'invio di messaggi nonché l'instradamento dei

pacchetti sia IPv4 che IPv6. Un secondo archivio contiene invece alcuni packages aggiuntivi per la gestione della qualità del servizio, l'analisi dei tracciati dei flussi TCP o un traduttore NAT-PT comprensivo di ALG per FTP e DNS. Il funzionamento del router, le modalità di utilizzo dei vari moduli aggiuntivi e la loro integrazione vengono definiti invece in un file di configurazione apposito, scritti in un linguaggio dedicato estremamente potente ma non altrettanto immediato.

7.2.1 Installazione

Una volta scaricati i pacchetti click e click-packages, è necessario decomprimerli in una directory utilizzando, in sequenza, i comandi:

gzip d nomefile.gz

tar xf nomefile.tar

Dopo aver decompresso entrambi i pacchetti, è necessario copiare alcuni file dalla cartella packages a quella principale di click. Nello specifico:

- i file dnsalg.* ftpportmapper6.* e tcpaddresstranslator.* vanno copiati nella sottodirectory elements/ip6.
- il file rfc1036.h va copiato nella sottodirectory include/clicknet
- il file dnsmessage.cc va copiato nella sottodirectory lib
- il file dnsmessage.hh va copiato nella sottodirectory include/click.

A questo punto, dalla directory di base in cui è stato decompresso click occorre lanciare il comando

./configure --enable-ip6

Viene così avviata la procedura di configurazione completamente automatica che dovrebbe concludersi in breve tempo e con successo. Al termine è necessario tuttavia modificare il file elements.conf presente nella sottodirectory userlevel, inserendo la riga

../elements/ip6/dnsalg.cc "../elements/ip6/dnsalg.hh" DNSAlg

Inoltre, all'interno del file userlevel/Makefile.in, occorre aggiungere dnsmessage.o all'elenco associato a GENERIC_OBJS; la stessa procedura va ripetuta per il file linuxmodule/Makefile.in

A questo punto, per avviare la compilazione dei sorgenti e la successiva installazione vera e propria del programma, occorre lanciare il comando make seguito, infine, dal comando make install

7.2.2 Configurazione e Utilizzo

Per poter testare le funzionalità del nodo traduttore NAT-PT installato, una macchina funzionante sotto Linux è stata configurata come nodo IPv4-only ed un secondo sistema FreeBSD è stato invece predisposto come host IPv6-only. I PC sono stati poi connessi a due segmenti Ethernet e,

quindi, a due differenti interfacce del router NAT-PT. Infine, tramite opportune regole di instradamento, è stato impostato il nodo NAT-PT come router predefinito per l'accesso alla rete esterna.

Per la configurazione del nodo traduttore è stato invece creato un file di configurazione di Click apposito, riportato (opportunamente commentato) in Appendice A. Dopo aver assegnato gli indirizzi alle interfacce, è stato utilizzato un elemento apposito (Classifier) per analizzare le frame Ethernet ricevute e distinguere quindi i diversi traffici. In particolare sono stati filtrati e gestiti i messaggi ARP di Request e Reply, per quanto riguarda la rete V4, e i messaggi ICMPv6 Neighbor Discovery e Advertisement, effettuando la risoluzione di indirizzi quando necessario o rispondendo alle richieste degli host. Al processo di traduzione vero e proprio, effettuato grazie alle primitive fornite dal modulo software GT64, sono stati invece sottoposti i datagram IP e l'eventuale contenuto. La traduzione degli indirizzi è resa possibile dall'elemento at (Address Translator), funzionante in entrambe le direzioni, mentre la traduzione completa del protocollo IP da V6 a V4 o da V4 a V6 è stata realizzata rispettivamente dagli elementi PT64 e PT46. Per quel che riguarda gli ALG, FTPPortMapper e TCPAlg effettuano la traduzione delle sessioni FTP, mentre DNSAlg è incaricato di gestire le richieste e le risposte DNS.

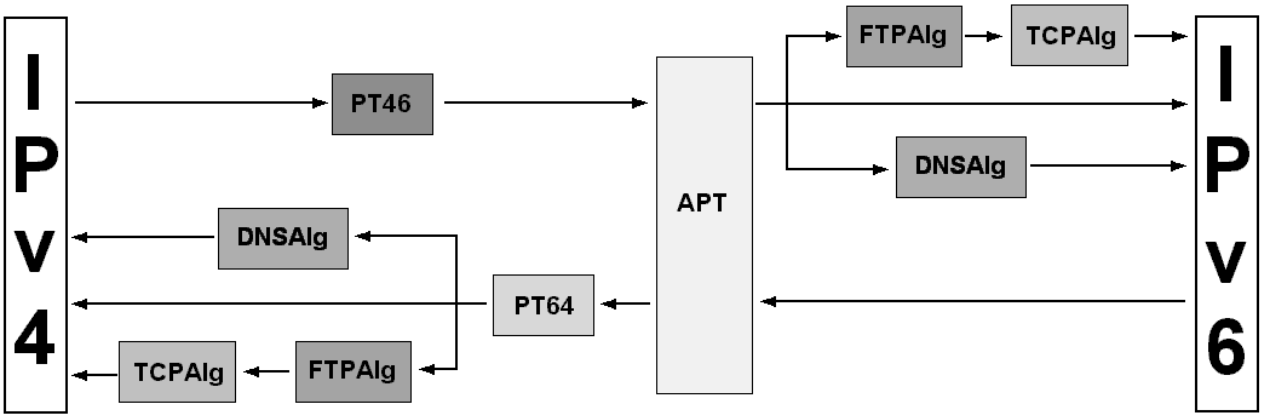

Figura 7.1 Schema rappresentante il percorso seguito dai pacchetti ricevuti dal router Click

Vista la semplicità della rete presa in considerazione, il mapping tra indirizzi IPv4 ed IPv6 è stato impostato staticamente all'interno del file di configurazione stesso e anche le regole di instradamento sugli host sono state definite manualmente. Per quel che riguarda gli indirizzi, all'interno della rete V6 l'host destinatario V4 è stato indirizzato utilizzando il corrispondente indirizzo di tipo IPv4-mapped; il nodo IPv6 è stato invece mappato virtualmente sul NAT-PT utilizzando l'indirizzo IPv4 1.0.0.1.

La configurazione realizzata ha permesso quindi di realizzare un router Bi-directional NAT-PT funzionante, in grado teoricamente di tradurre messaggi ICMP e IP, richieste e risposte di risoluzione dei nomi e persino pacchetti appartenenti a sessioni di controllo FTP.

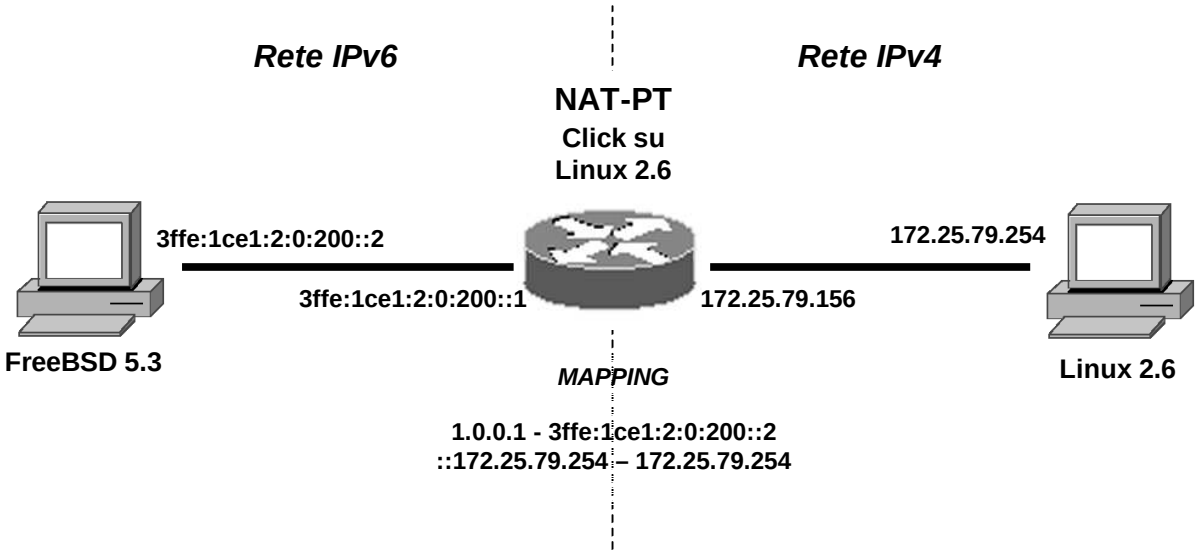

Figura 7.2 Schema rappresentante la situazione di test realizzata in laboratorio

7.2.3 Problematiche riscontrate

Purtroppo il modulo NAT-PT per Click, benché si sia dimostrato essere l'unica implementazione libera attualmente funzionante, non può certo essere considerato un prodotto stabile ed affidabile. Durante le varie prove svolte in laboratorio il sistema infatti si è ripetutamente bloccato, rendendo di fatto impossibile realizzare prove complesse o anche solo valutazioni in termini prestazionali di questa soluzione. E' da escludere che il problema sia legato alla macchina o alla particolare distribuzione Linux utilizzata, poiché gli stessi problemi sono stati riscontrati su computer completamente differenti tra loro; più probabilmente gli errori di 'Segmentation Fault' sono imputabili ad un'errata gestione delle letture e scritture in memoria da parte di qualche funzione specifica di gt64. La traduzione di messaggi ICMP, peraltro già realizzata con successo nell'ambito di un altro progetto di ricerca ospitato dal Laboratorio, non ha presentato particolari problemi; la traduzione di sessioni Ftp complete ha avuto invece scarso successo proprio a causa dei continui blocchi del sistema.

Capitolo 8
Conclusioni

8.1 Presente e futuro di IPv6

Nonostante l'iniziale euforia, l'introduzione di IPv6 ha subito, tuttavia, una battuta di arresto e la sua diffusione non è ancora ad oggi tale da poter considerare IPv4 defunto. A questo ha sicuramente contribuito il mancato esaurimento in tempi brevi degli indirizzi IPv4 disponibili: nel 1994 era stato indicato il 2006 (vedi RFC 1744) come data ultima prima dell'esaurimento dell'intero spazio di indirizzamento V4; in realtà l'introduzione negli anni di diverse strategie (tra le quali il NAT ha sicuramente giocato il ruolo principale) ha permesso di contenere il tasso di crescita della richiesta di indirizzi, rendendo di fatto meno urgente l'introduzione di IPv6. Inoltre occorre considerare che il processo di standardizzazione non è stato sicuramente dei più semplici: mentre gli aspetti basilari di IPv6 sono ormai consolidati da almeno 6-7 anni, altri importanti funzionalità quali il DHCP o il supporto alla Mobilità hanno richiesto molto tempo, giungendo solo di recente alla stesura definitiva (DHCPv6 risale a luglio 2003, il supporto alla Mobilità in IPv6 a giugno 2004).

Ora che la tecnologia è però disponibile, rimangono comunque molti ostacoli da superare. Al momento tutti i moderni sistemi operativi supportano già da alcuni anni IPv6 ma grossi problemi si avranno con tutte quelle applicazioni scritte per IPv4 che utilizzano le socket, ovvero l'interfaccia fornita dal sistema operativo per la comunicazione via rete: essendo infatti le socket strettamente legate allo specifico protocollo utilizzato, moltissime applicazioni dovranno essere modificate per poter supportare nativamente IPv6. In aggiunta a questo, diversi produttori di hardware di rete e fornitori di servizi rimangono ancora oggi riluttanti di fronte all'idea di modificare pesantemente le apparecchiature e l'infrastruttura operativa attuale solamente per supportare un protocollo ancora poco diffuso e richiesto da pochi utenti e clienti. Infine, aspetto forse triviale ma sicuramente da non sottovalutare, è la mancanza sul mercato di una vera ‘killer application’ basata su IPv6, in grado di attirare un gran numero di utenti e superare quella “massa critica” che sarebbe con ogni probabilità in grado di decretare il successo definitivo di questo protocollo.

Alcuni fattori potrebbero però concorrere ed accelerare la diffusione del nuovo protocollo, rappresentando un momento cruciale nel processo di migrazione. L’interesse nei confronti di IPv6 dell’Europa e, soprattutto, delle nazioni tecnologicamente “emergenti” potrebbe contribuire ad raggiungere un numero di utenti considerevole; in particolare India, Cina e gli altri paesi della Asia stanno già da alcuni anni lavorando con IPv6 e sembrano intenzionati ad utilizzarlo sempre più

massicciamente in futuro. Inoltre alcune reti di tecnologia mobile, tra cui la futura rete di Terza Generazione (3G), sono state pensate proprio basandosi sulle funzionalità offerte da IPv6; una diffusione di questo tipo di tecnologie rappresenterebbe sicuramente un ulteriore incentivo per la diffusione del nuovo protocollo. Inoltre IPv6 è stato studiato appositamente per garantire elevati standard di sicurezza end-to-end e dispone di un supporto notevole per i dispositivi wireless e mobili in generale.

Per lungo tempo ancora probabilmente IPv4 ed IPv4 continueranno a coesistere ma, con gli attuali ritmi di sviluppo della tecnologia e dello stesso Internet, il vecchio protocollo è destinato prima a poi a dover cedere il passo al nuovo IP versione 6. I tempi di questa migrazione rimangono però ancora un'incognita.

8.2 Presente e futuro di NAT-PT

NAT-PT era stato pensato originariamente come un meccanismo di transizione temporaneo e sicuramente non perfetto, ma necessario in vista di una rapida transizione della rete mondiale da IPv4 ad IPv6. In realtà i tempi di migrazione più lunghi del previsto hanno permesso a tutti gli sviluppatori di iniziare gradualmente a supportare nativamente IPv6 all'interno del proprio software. Da diversi anni ormai i più diffusi sistemi operativi (Windows, Linux, BDS, Solaris…) integrano IPv6 così come i browser, i client e i server ftp e le applicazioni multimediali presenti sui desktop di tutto il mondo. La lunga, e forzata, convivenza di IPv4 ed IPv6 all'interno delle stesse reti ha ovviamente favorito l'adozione di strategie alternative a NAT-PT; si sono così diffusi i sistemi Dual-stack reti ed i tunnel sono stati largamente utilizzati per mettere in comunicazione reti isolate. E' lecito aspettarsi che ancora per diversi anni la vecchia e la nuova versione del protocollo di Internet convivranno e, nel momento in cui IPv4 non sarà più necessario, non saranno probabilmente più funzionanti molti sistemi IPv4-only. Non è un caso che l'interesse verso i traduttori NAT-PT e simili sia andato nel tempo scemando: con ogni probabilità non sarà quasi mai necessaria la loro adozione all'interno delle reti, salvo alcune situazioni particolari (es. sistemi o dispositivi necessari ma non aggiornabili ad IPv6). NAT-PT inoltre, come ricordato in precedenza, presenta diversi limiti e reintroduce all'interno delle reti problematiche tipiche dei dispositivi NAT che, nelle intenzioni dei progettisti, avrebbero dovuto essere superate proprio con l'introduzione di IPv6. In questi ultimi mesi è in fase di stesura un Internet Draft proprio a questo riguardo (draft-ietf-v6ops-natpt-to-exprmntl-01): in particolare gli autori hanno riassunto tutti i problemi derivanti dall'utilizzo di NAT-PT e hanno concluso che l'utilizzo di questo meccanismo di traduzione è sconsigliabile. Alla luce di quanto detto fino ad ora, e sulla base anche delle prove effettuate in laboratorio, mi sento personalmente di appoggiare questa tesi: per il futuro conviene puntare su soluzioni alternative, meccanismi Dual-Stack *in primis*.

8.3 La sperimentazione IPv6 nell'Università degli Studi di Pavia

6Bone è una rete IPv6 sperimentale, nata nel 1996, alla quale sono interconnesse sottoreti localizzate in quasi tutto il mondo. Si tratta di una rete costituita dall'interconnessione di isole IPv6, realizzata in gran parte come rete sovrapposta all'attuale Internet IPv4 utilizzando la tecnica del tunneling statico. Nata come iniziativa spontanea di diversi istituti di ricerca coinvolti nella sperimentazione delle prime implementazioni del protocollo IPv6, *6Bone* è a tuttoggi la rete dove hanno luogo le più interessanti sperimentazioni geografiche. Tali attività sono coordinate dall'IETF con lo scopo di aiutare i lavori di specifica tecnica e l'assestamento delle implementazioni del protocollo, sulla base dell'esperienza delle prove in campo. Anche il laboratorio Reti dell'Università degli Studi di Pavia, operativo presso la sede di Mantova, è connesso alla dorsale mondiale sperimentale *6Bone* attraverso il TILAB (Telecomitalia Lab) di Torino. Tale connessione, attivata alla fine del 2002, ha come obiettivo il test del nuovo protocollo e della sua interoperabilità con le attuali reti IPv4; lo spazio di indirizzamento IPv6 di *6Bone* assegnato al Laboratorio Reti è 3FFE:1001:1C0::/48. Tuttavia è bene ricordare che la rete *6Bone*, poiché realizzata esclusivamente per fini sperimentali, è solo temporanea: il 6 Giugno 2006 infatti cesserà di funzionare definitivamente (RFC3701) e tutti gli indirizzi assegnati in questi anni verranno ritirati.

6Net è invece un progetto Europeo, nato nel 2002, per la sperimentazione di IPv6 nel vecchio continente. In Italia il progetto *6Net* prevede la realizzazione di una rete nativa IPv6, parallela all'attuale rete di produzione di GARR. La Rete GARR (il cui acronimo significa "Gestione Amplimento Rete Ricerca") è composta da tutte le Entità che rappresentano la Comunità Accademica e della Ricerca Scientifica in Italia. GARR segue direttamente lo sviluppo del protocollo IPv6 e della tecnologia ad esso collegata, partecipando ai gruppi di lavoro di RIPE e di IETF e coordinando in Italia le sperimentazioni del progetto europeo *6Net*. Nel progetto sono coinvolti attivamente diversi enti, tra cui il CASPUR, il CNR e le Università di Milano, Torino, Bologna, Napoli e Roma Tre. Nel Febbraio 2005 anche l'Università degli Studi di Pavia è stata inserita nell'elenco degli enti coinvolti nel progetti di sperimentazione 6NET-GARR. Lo spazio di indirizzamento IPv6 assegnato a Pavia per l'intera rete di Ateneo è 2001:760:2000::/48. Questo blocco di indirizzi è stato a sua volta suddiviso dal Centro di Calcolo dell'Università per realizzare più sottoreti; al Laboratorio Reti, attivamente impegnato all'interno dell'Ateneo nella sperimentazione del nuovo protocollo, è stato assegnato lo spazio di indirizzamento 2001:760:2000:1000::/56. La connessione IPv6 tra il Centro di Calcolo e la sede di Mantova è stata realizzata mediante Tunneling IPv6 incapsulato in IPv4; in particolare, il router utilizzato all'interno del laboratorio come end-point del tunnel consiste in un computer Dual-Stack con installato il sistema operativo FreeBSD 5. Questa macchina funge, all'interno della rete locale del

Laboratorio, da gateway predefinito per il traffico sia IPv4 che IPv6; inoltre, utilizzando il servizio di Routing Advertisement nativamente implementato in FreeBSD, la configurazione degli host IPv6 connessi è effettuata in maniera automatica.

Infine, proprio in questi mesi è in fase di realizzazione il nuovo sito Web del Laboratorio (http://labreti-mn.unipv.it), che verrà poi ospitato su un server Dual-Stack dedicato. Questa macchina, installata sempre all'interno del laboratorio, potrà quindi essere visitata contemporaneamente da utenti IPv4 ed IPv6.

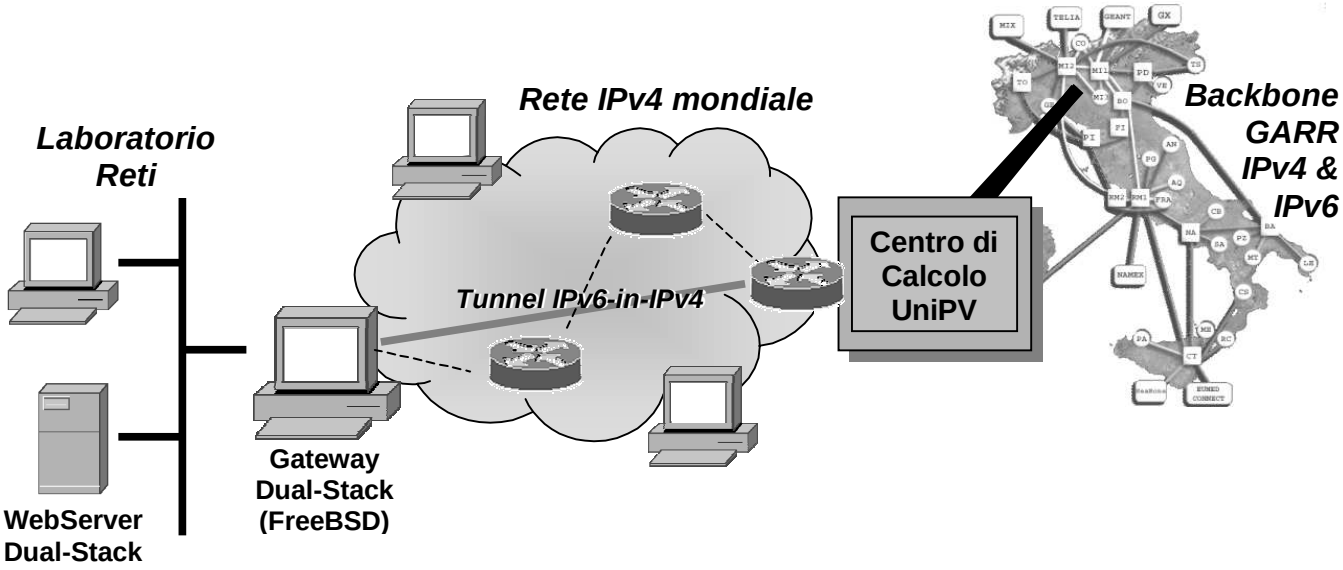

Figura 8.1 Sistema di connessione del Laboratorio Reti alla rete sperimentale GARR-6Net

Bibliografia

A.A.V.V. (2003), “Test Specification: IPv6 conformance test for NAT-PT”, Tahi, <http://www.tahi.org>

A.A.V.V. (2005) “Implementing NAT Protocol Translation”, *Implementig IPv6 for Cisco IOS Software*, pp. 333-354

A.A.V.V., “The FreeBSD Handbook”, Cap. 26.10, <http://www.freebsd.org>

Aiello A., Cardone L. (2001), “IPv6 l’internet che verrà…”, *Linux & C.*, 18, pp. 44-47

Allman M. et al., “FTP Extensions for IPv6 and NATs”, RFC 2428, Internet Engineering Task Force, Settembre 1998

Anselmi L., Goldoni E. (2005) “Manuale di configurazione del protocollo IPv6 nel sistema operativo Linux”, Laboratorio di Reti di Calcolatori, Università degli Studi di Pavia

Aoun C., Davies E., “Reasons to move NAT-PT to Experimental”, IETF Draft draft-ietf-v6ops-natpt-to-exprmntl-01, Internet Engineering Task Force, Luglio 2005

Atwood J. W. et al. (2002) “IPv4/IPv6 Translation”, Concordia University, Montréal

Atwood J. W. et al. (2003) “NAT-PT : providing IPv4/IPv6 and IPv6/IPv4 Address Translation”, Technical Report, Ericsson Open System Lab.

Baker F., “Requirements for IP Version 4 Routers”, RFC 1812, Internet Engineering Task Force, Giugno 1995

Baptiste J. L., Gonzalez F. (2003) “Enrutamiento transparente entre redes IPv4/IPv6”, Ph.D. Thesis, Pontificia Universidad Javeriana de Sandafé de Bogotà

Bellovin S., “Firewall-Friendly FTP”, RFC 1579, Internet Engineering Task Force, Febbraio 1994

Bieringer P. (2005) “Linux IPv6 how to Linux IPv6 HOWTO”, The Linux Documentation Project.

Borman D. et al, “IPv6 Jumbograms”, RFC 2675, Internet Engineering Task Force, Agosto 1999

Capisani L. (2004) “Analisi comparativa dello stack dei protocolli TCP/IPv4 e TCP/IPv6 e problematiche di Protocol Translation”, Tesi di Laurea, Università degli Studi di Pavia

Capisani L. (2004) “Guida alla configurazione di base della rete nel sistema operativo Linux”, Laboratorio di Reti di Calcolatori, Università degli Studi di Pavia

Carpenter B., Jung C., “Transmission of IPv6 over IPv4 Domains without Explicit Tunnels”, RFC 2529, Internet Engineering Task Force, Marzo 1999

Chen W. et al. (2003) “NCTU SLT, A Socket-layer translator for IPv4-IPv6 translation”, National Chiao Tung University

Conta A., Deering S., “Internet Control Message Protocol (ICMPv6) for the Internet Protocol version 6 (IPv6) specification”, RFC 2463, Internet Engineering Task Force, Dicembre 1998

Crawford M., “Transmission of IPv6 Packets over Ethernet Networks”, RFC 2464, Internet Engineering Task Force, Dicembre 1998

Deering S., Hinden R., “Internet Protocol, version 6 (IPv6) specification”, RFC 2460, Internet Engineering Task Force, Dicembre 1998

Droms R. et al, “Dynamic Host Configuration Protocol for IPv6 (DHCPv6)”, RFC 3315, Internet Engineering Task Force, Luglio 2003

Durand A. et al, “IPv6 Tunnel Broker”, RFC 3053, Internet Engineering Task Force, Gennaio 2001

Durand A., “Dual stack vs NAT-PT”, IETF Draft draft-durand-v6ops-dualstack-vs-natpt-00, Internet Engineering Task Force, Febbraio 2003

Fan P. (2001), “A design and implementation of a general-purpose translator for IPv6/IPv4 (GT64)”, Ph.D. Thesis, Massachussets Institute of Technology

Fiuczynski M. et al. (1998) “The design and implementation of an IPv6/IPv4 Network Address and Protocol Translator”, University of Washington

Grossman D., “New Terminology and Clarifications for Diffserv”, RFC 3260, Internet Engineering Task Force, Aprile 2002

Guardini I. (2000) “Migrating from IPv4 to IPv6: planning an effective IPv6 transition”, <http://carmen.cselt.it/papers/globalIPsummit-v6trans/home.html>

Gurbani V., Boulton C., “ Recommendations on the use of IPv6 in the Session Initiation Protocol (SIP)”, IETF Draft draft-gurbani-sipping-ipv6-sip-00, Internet Engineering Task Force, Luglio 2005

Haddad I. (2003) “IPv6: the essential you must know”, *Linux User and Developer*, 06/2003

Hagino J., Yamamoto K., “An IPv6-to-IPv4 Transport Relay Translator”, RFC 3142, Internet Engineering Task Force, Giugno 2001

Holdrege M., Srisuresh P., “Protocol Complications with the IP Network Address Translator”, RFC 3027, Internet Engineering Task Force, Gennaio 2001

Huston G., “Observations on the Management of the Internet Address Space”, RFC 1744, Internet Engineering Task Force, Dicembre 1994

Kent S., Atkinson R., “Security Architecture for the Internet Protocol”, RFC 2301, Internet Engineering Task Force, Novembre 1998

Kohler E. (2000) “The Click modular router”, Ph.D. Thesis, Massachussets Institute of Technology

Kohler E. et al (200) "Modular components for network address translation", Technical Report, Massachussets Institute of Technology

Lee S. et al, “Dual Stack Hosts Using ‘Bump-in-the-API’ (BIA)”, RFC 3338, Internet Engineering Task Force, Ottobre 2002

Mockapetris P., “Domain Names – implementation and specification”, RFC 1035, Internet Engineering Task Force, Novembre 1987

Narten T. et al, “Neighbor Discovery for IP version 6 (IPv6)”, RFC 2461, Internet Engineering Task Force, Dicembre 1998

Nordmark E., “Stateless IP/ICMP Translation Algorithm (SIIT)”, RFC 2765, Internet Engineering Task Force, Febbraio 2000

Olson S. et al, “Support for IPv6 in Session Description Protocol (SDP)”, RFC 3266, Internet Engineering Task Force, Giugno 2002

Postel J., “DoD Standard Transmission Control Protocol”, RFC 761, Internet Engineering Task Force, Gennaio 1980

Postel J., “Internet Protocol (IP)”, RFC 791, Internet Engineering Task Force, Settembre 1981

Postel J., “Transmission Control Protocol (TCP)”, RFC 793, Internet Engineering Task Force, Settembre 1981

Postel J., “User Datagram Protocol (UDP)”, RFC 768, Internet Engineering Task Force, Agosto 1980

Postel J., J. Reynolds, “File Transfer Protocol (FTP)”, RFC 959, Internet Engineering Task Force, Ottobre 1985

Postel J., Reynolds J., “Telnet protocol specification”, RFC 854, Internet Engineering Task Force, Maggio 1983

Rajahalme J. et al., “IPv6 Flow Label Specification”, RFC 3697, Internet Engineering Task Force, Marzo 2004

Ribak J., “Active FTP vs. Passive FTP, a definitive explanation”, <http://slacksite.com/other/ftp.html>

Rosenberg J. et al, “SIP: Session Initiation Protocol”, RFC 3261, Internet Engineering Task Force, Giugno 2002

Rossi G. F., Lucidi del corso di Reti di Calcolatori, Università degli Studi di Pavia, 2005

Rossi G. F., Lucidi del corso di Reti Telematiche, Università degli Studi di Pavia, 2005

Schulzrinne H. et al, “Real Time Streaming Protocol (RTSP)”, RFC 2326, Internet Engineering Task Force, Aprile 1998

Schulzrinne H. et al, “RTP: A Transport Protocol for Real-Time Applications”, RFC 3550, Internet Engineering Task Force, Luglio 2003

Shin M. et al., “Application aspects of IPv6 transition”, RFC 4038, Internet Engineering Task Force, Marzo 2005

Srisuresh P. et al, “DNS extensions to Network Address Translators (DNS_ALG)”, RFC 2694, Internet Engineering Task Force, Settembre 1999

Srisuresh P., Egevang K., “Traditional IP Network Address Translator (Traditional NAT)”, RFC 3022, Internet Engineering Task Force, Gennaio 2001

Srisuresh P., Holdrege M., “IP Network Address Translator (NAT) terminology and considerations”, RFC 2663, Internet Engineering Task Force, Agosto 1999

Thomson S. et al, “DNS extensions to support IP version 6”, RFC 3596, Internet Engineering Task Force, Ottobre 2003

Thomson S., Narten T., “IPv6 Stateless address autoconfiguration”, RFC 2462, Internet Engineering Task Force, Dicembre 1998

Tsirtsis G., Srisuresh P., “Network Address Translation – Protocol Translation (NAT-PT)”, RFC 2766, Internet Engineering Task Force, Febbraio 2000

Tsuchiya K. et al, “Dual Stack Hosts using the ‘Bump-In-the-Stack’ Technique (BIS)”, RFC 2767, Internet Engineering Task Force, Febbraio 2000

Waddington D., Chang F. (2002) “Realizing the transition to IPv6”, *IEEE Communication Magazine*, 06/2002, pp.138-148

Yin F. (2002) “NAT-PT study and test report”, Technical Report, Université libre de Bruxelles

Appendice A: File di configurazione per il router NAT-PT

```
//
// File in linguaggio Click di configurazione per un traduttore
// NAT-PT con supporto per il protocollo FTP mediante apposito ALG.
//
// Utilizzabile in modalità user level, effettua una traduzione
// statica degli indirizzi; per cambiare questa impostazione è
// modificare l'elemento at (Addres Translator). Si vedano a
// riguardo le pagine del manuale di Click presenti on-line
// (http://pdos.csail.mit.edu/click)
//
// La rete utilizzata in laboratorio è stata configurata
// secondo lo schema sotto riportato:
//
//               Rete v4                              Rete v6
// Host IPv6 -------------------- Traduttore -------------------- Host IPv4
// 3ffe:1ce1:2:0:200::2     3ffe:1ce1:2:0:200::1              172.25.79.254
// (mappato come 1.0.0.1)       172.25.79.156
//
//
// Autore originale: Juan Luis Baptiste M. <juancho@linuxmail.org>
// Modificato da: Emanuele Goldoni <emanuele.goldoni01@ateneopv.it>
//


elementclass GatewayDevice {
  $device |
  from :: FromDevice($device)
      -> output;
  input -> q :: Queue(1024)
      -> to :: ToDevice($device);
  ScheduleInfo(from .1, to 1);
}

intern_dev :: GatewayDevice(eth0);
extern_dev :: GatewayDevice(eth1);


// Configurazione degli elementi per la comunicazione nelle rete IPv6
// dedicati alla gestione di messaggi Neighbor solicitation e advertisement

intern_nda::IP6NDAdvertiser(3ffe:1ce1:2:0:200::1/128 00:E0:7D:E1:BB:E0,
                      3ffe:1ce1:2::/80 00:E0:7D:E1:BB:E0);
intern_nds::IP6NDSolicitor(3ffe:1ce1:2:0:200::1, 00:E0:7D:E1:BB:E0);


// Configurazione degli elementi per la comunicazione nelle rete IPv4
// dedicati alla gestione di messaggi ARP request e reply

extern_arp::ARPQuerier(172.25.79.156, 00:10:5A:1C:86:15);
extern_arr::ARPResponder(172.25.79.156 00:10:5A:1C:86:15);


// Routing table statiche IPv4

ipv4rt :: StaticIPLookup(

      172.25.79.156/32 0,
      172.25.79.255/32 0,
      172.25.79.0/32 0,
      192.168.1.2/32 172.25.79.220 1,
```

```
      172.25.79.220/32 172.25.79.220 1,
      1.0.0.1/32 1.0.0.1 2,  // pacchetto da tradurre
      0.0.0.0/0 172.25.79.156 3);


// Routing table IPv6

ipv6rt :: LookupIP6Route(
      3ffe:1ce1:2::2/128 ::0 0,
      3ffe:1ce1:2:0:200::1/128 ::0 0,
      3ffe:1ce1:2:0:200::2/128 3ffe:1ce1:2:0:200::2 1,
      3ffe:1ce1:2::/80 ::0 2,
      3ffe:1ce1:2:0:200::/80 ::0 2,
      ::0/96 ::0 3,  // pacchetto da tradurre
      ::0/0 ::c0a8:1 4);


// Classificazione del traffico proveniente dalla rete IPv4

extern_class :: Classifier(
      12/0806 20/0001,  // Messaggi di ARP query (output 0)
      12/0806 20/0002,  // Messaggi di ARP reply (output 1)
      12/0800 30/01000001,  // Pacchetti IPv4 (output 2)
        -); // Altri pacchetti...


// Classificazione del traffico proveniente dalla rete IPv6

intern_class :: Classifier(
      12/86dd 20/3aff 54/87, //Messaggi di Neighboor solicitation (output 0)
      12/86dd 20/3aff 54/88, //Messaggi di Neighboor advertisement (output 1)
      12/86dd,  // Pacchetti IPv6 (output 2)
      -); // Altri pacchetti


// Configurazione del elemento Address Translator
// (è impostato un solo mapping e di tipo statico)

at :: AddressTranslator(
      1,
      0,
      3ffe:1ce1:2:0:200::2 ::1.0.0.1,
      0,
      0,
      0);

tcpAddr:: TCPAddressTranslator(at);


// Configurazione degli elementi incaricati di effettuare
// la Protocol Translation (da IPv6 a IPv6 e da IPv4 a IPv6)

pt64 :: ProtocolTranslator64();
pt46 :: ProtocolTranslator46();


// ALG FTP

ftp6:: FTPPortMapper6(tcpAddr);


// Gestione diversi tipi di traffico classificati
// provenienti dalla rete da rete 'interna' IPv6 (eth0)
```

```
intern_dev
      -> intern_class;

// Neighbor solicitation
intern_class[0]
      -> [0]intern_nda;

// Neighbor advertisement
intern_class[1]
      -> [1]intern_nds;

// Pacchetto IPv6 'normale'
intern_class[2]
      -> Strip(14)
      -> CheckIP6Header(3ffe:1ce1:2:0:200::ffff 3ffe:1ce1:2::ffff)
      -> GetIP6Address(24)
      -> ipv6rt;

// Altro...
intern_class[3]
      -> Discard;


// Gestione diversi tipi di traffico classificati
// provenienti dalla rete IPv4 (eth1)

extern_dev
      -> extern_class;

// ARP Query
extern_class[0]
      -> extern_arr;

// ARP Response
extern_class[1]
      -> [1]extern_arp;

// Pacchetto IPv4 'normale'
extern_class[2]
      -> Strip(14)
      -> CheckIPHeader(172.25.79.255)
      -> GetIPAddress(16)
      -> ipv4rt;

// Altro...
extern_class[3]
      ->Discard;


// Istruzioni per il routing dei pacchetti
// IPv6 in base alle regole indicate sopra
// nella 'routing table'

ipv6rt[0]
        -> Discard;

ipv6rt[1]
      -> intern_dh1:: DecIP6HLIM
      -> [0]intern_nds;

ipv6rt[2]
      -> intern_dh2:: DecIP6HLIM
```

```
      -> Discard;

ipv6rt[3]
      -> [0]at;

ipv6rt[4]
        -> Discard;


// Istruzioni per il routing dei pacchetti
// IPv4 in base alle regole, sopra indicate
// nella 'routing table'

ipv4rt[0]
      -> Discard;

ipv4rt[1]
      -> DropBroadcasts
            -> dt1 :: DecIPTTL
            -> fr1 :: IPFragmenter(1500)
      -> [0]extern_arp;

ipv4rt[2]
      -> [0]pt46;

ipv4rt[3]
      -> Discard;


// Elemento incaricato di effettuare la traduzione degli
// indirizzi IPv4 <--> IPv6

at[0]
      -> [0]pt64;

at[1]
      -> CheckIP6Header(3ffe:1ce1:2:0:200::ffff 3ffe:1ce1:2::ffff)
      -> [1]ftp6;


// Elemento per la traduzione del
// protocollo IP da v6 a v4

pt64[0]
      -> CheckIPHeader(172.25.79.255 1.255.255.255)
      -> [0]ftp6;


// Elemento per la traduzione del
// protocollo IP da v4 a v6

pt46[0]
      -> [1]at;


// Elemento per la traduzione del
// protocollo FTP (FTP_ALG)

ftp6[0]
      -> [0]tcpAddr;
ftp6[1]
      -> [1]tcpAddr;
```

```
// Elemento per la traduzione del
// protocollo TCP

tcpAddr[0]
      -> GetIPAddress(16)
      -> [0]ipv4rt;

tcpAddr[1]
      -> GetIP6Address(24)
      -> [0]ipv6rt;


intern_dh1[1]
      -> ICMP6Error(3ffe:1ce1:2:0:200::1, 3, 0)
      -> [0]intern_nds;


// Configurazione delle reti di destinazione
// per i messaggi rispettivamente ICMPv4 e ICMPv6

extern_arp[0]
      -> extern_dev;

extern_arr[0]
      -> extern_dev;


intern_nds[0]
      -> intern_dev;

intern_nda[0]
      -> intern_dev;
```

Appendice B: Algoritmo di funzionamento dell'ALG di Click per FTP

Traduzione IPv6→IPv4

1. Tradurre la testata IP
2. Calcolare l'offset del punto in cui sono inseriti i dati del pacchetto FTP e, a partire da questo punto, cercare la stringa "EPRT", "EPSV |2|" o "229 Entering extended passive mode".
    a. Se la stringa incontrata è "EPSV |2|" modificarla con "PASV"
    b. Se la stringa è ""229 Entering extended passive mode (|||porta|)"
        i. Estrarre il numero di porta indicato nella parte finale del comando
        ii. Ricavare l'indirizzo mittente del nuovo pacchetto IPv4
        iii. Creare la nuova risposta per il comando PASV con l'indirizzo e la porta ottenuti
    c. Se la stringa "EPRT"
        i. Estrarre il numero di porta indicato nella parte finale del comando
        ii. Ricavare l'indirizzo mittente del nuovo pacchetto IPv4
        iii. Creare la nuova risposta per il comando PORT con l'indirizzo e la porta ottenuti
3. Creare il nuovo pacchetto
    a. Copiare il nuovo comando nel pacchetto nuovo
    b. Copiare la testata IPv4 tradotta nel nuovo pacchetto
    c. Aggiornare il campo Packet_Length nella testata IPv4 se necessario
    d. Aggiornare di conseguenza il valore del campo Checksum della testata IPv4 (utilizzando la formula specificata in RFC1624)
    e. Copiare la testata TCP nel nuovo pacchetto
    f. Aggiornare i numeri di sequenza
    g. Calcolare la nuova lunghezza della testata TCP
    h. Calcolare il valore del campo Checksum della testata TCP

Traduzione IPv4→IPv6

1. Tradurre la testata IP
2. Calcolare l'offset del punto in cui sono inseriti i dati del pacchetto FTP e, a partire da questo punto, cercare la stringa "PORT", "PASV" o "227 Entering passive mode".
    a. Se la stringa incontrata è "PASV", modificarla con "EPSV |2|"
    b. Se la stringa è ""227 Entering extended passive mode …"

i. Estrarre il numero di porta indicato nella parte finale del comando
ii. Creare la nuova risposta per il comando EPSV con l'indirizzo e la porta ottenuti, utilizzando il formato (|||porta|).

c. Se la stringa "PORT"
i. Estrarre il numero di porta indicato nella parte finale del comando
ii. Ricavare l'indirizzo mittente del nuovo pacchetto IPv6
iii. Creare la nuova risposta per il comando EPRT con l'indirizzo IPv6 e la porta ottenuti

3. Creare il nuovo pacchetto
   a. Copiare il nuovo comando nel pacchetto nuovo
   b. Copiare la testata IPv6 tradotta nel nuovo pacchetto
   c. Aggiornare il campo Packet_Length nella testata IPv6 se necessario
   d. Copiare la testata TCP nel nuovo pacchetto
   e. Calcolare i nuovi numeri di sequenza
   f. Calcolare la nuova lunghezza della testata TCP
   g. Calcolare il valore del campo Checksum della testata TCP