IPv6

Luca Veltri

(mail.to: luca.veltri@unipr.it)

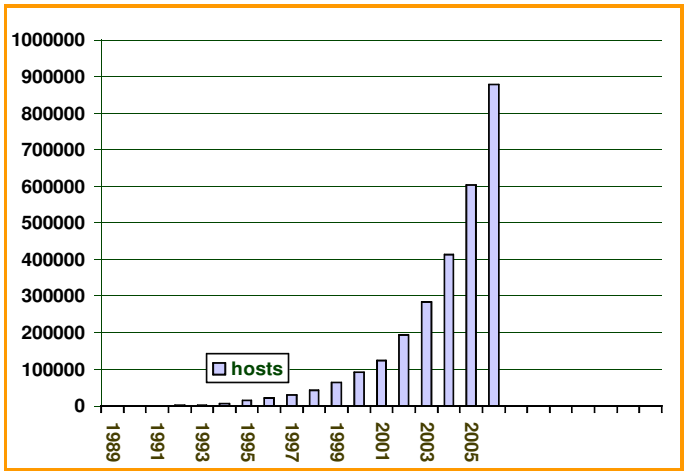
Corso di Reti di Telecomunicazioni C, a.a. 2008/2009

http://www.tlc.unipr.it/veltri

IPv4

- RFC 791 del 1981
- Pressoché inalterato da circa 25 anni
- Maggiori problemi oggi:
  - **Spazio di indirizzamento limitato**
  - **Dimensione delle tabelle di routing**
  - **Configurazione degli host**
  - **Sicurezza**
  - **Qualità di servizio**
  - **Gestione della mobilità**

Internet Host Count 1989-2006



Internet Hosts (000s)

Problemi: Indirizzamento

- Crescente richiesta di nuovi indirizzi IP dovuti a:
- Dispositivi always-on
  - **Connettività diretta (non commutata) ad Internet e.g xDSL, Cable modem, Fiber to the home**
- Apparati mobili always-addressable
  - **GPRS**
  - **UMTS (backbone IMS in IPv6)**
  - **reti 4G? (e.g. WiMAX+WiFi)**
- Paesi in via di sviluppo
  - **e.g. Cina ed altri paesi asiatici**

Problemi: Indirizzamento

- Soluzione attuale: estensivo uso del Network Address Traslator (NAT/NAPT)
  - **Mapping di più indirizzi privati su un numero ristretto di indirizzi pubblici (tipicamente 1 solo indirizzo)**
  - **Riuso degli spazi di indirizzamento privati**
- Aspetti negativi del NAT
  - **Alcune applicazioni si devono gestire ad-hoc tramite ALG (FTP, DNS, ICMP, …)**
  - **Limita l'impiego di nuove applicazioni**
  - **Impedisce l'uso di alcuni meccanismi di sicurezza standard (alcune configurazioni di IPsec)**

Problemi: Tabelle di routing

- I router del backbone di Internet hanno bisogno di una conoscenza completa della topologia della rete (non hanno instradamenti di default)
- Circa 110.000 righe nelle tabelle dei router BGP
- Sarebbero enormemente di più senza il Classless Inter Domain Routing (CIDR - 1992)
- Limiti del CIDR
  - **Organizzazioni multi-homed**
  - **Organizzazioni che cambiano provider senza rinumerare**
  - **Indirizzi assegnati in modo non gerarchico e non restituiti**
- Questo problema è legato al throughput richiesto ai router ed alle prestazioni delle CPU

Problemi: Configurazione degli host

- Spesso manuale
- Necessità di rinumerazione se cambia l'ISP
- Può essere utile il DHCP

Problemi: QoS, Sicurezza, Mobilità

- Sicurezza
  - **Sono state definite differenti meccanismi**
    - IPsec v4
    - SSL
    - SHTTP
  - **Non sono però intrinsechi nel protocollo IP**
- Qualità di servizio
  - **Differenziated services**
  - **Integrated services**
    - Vantaggi e svantaggi per entrambe le soluzioni
    - Complessità
- Mobilità
  - **Mobile IP v4**
- Ogni nuova aggiunta ad IPv4 comporta aumenti di complessità

IPv6: storia & standards

- 1978: Definizione di Internet Protocol
- 1981: Specifiche del protocollo IPv4 (Jon Postel - rfc791)
- 1990: Previsioni di esaurimento di spazio di indirizzamento con IPv4
- 1992: TUBA (TCP and UDP over Bigger Addresses) – Ross Callon
  - **Impiego di ISO/OSI 8473 CLNP**
  - **Indirizzi OSI/N-SAP su 20 byte**
- 1992: Inizio dei lavori per la definizione di un nuovo IP (IPng)
- 1992/1993: IPv7-TP/IX-CATNIP – Robert Ullman
  - **Nuovo protocollo di routing RAP**
  - **Formato unico pacchetti per IP, CLNP ed IPX**
  - **Indirizzi OSI/N-SAP**
- 1992: SIP (Simple IP) – Steve Deering
  - **Impiego di indirizzi su 64 bit**
  - **Eliminazione di alcuni campi obsoleti**

IPv6: storia & standards

- 1993: PIP (Paul's IP) – Paul Francis
  - **Meccanismi di routing più efficenti**
- 1993: SIPP (Simple IP Plus)
  - **Fusione di SIP e PIP**
  - **Indirizzi su 64 bit**
- 1993: IPAE (IP Address Encapsulation) - Dave Crocker
  - **Due livelli di indirizzamento (dorsale mondiale ed aree periferiche)**
  - **Estensione del pacchetto trasportata come dati nel pacchetto IP**
  - **I vecchi host e router rimangono invariati**
  - **Solo i nuovo apparati devono elaborare le estensioni**
- 1994: Nasce IPng, battezzato poi IPv6 (...v5 utilizzato da ST)
- 1995: Le prime specifiche di IPv6 come Internet-Draft

IPv6: storia & standards

- 1995: IPng (RFC 1752) – S. Bradner, A. Mankin
  - **Sancisce la nascita di IPv6**
  - **Deriva dal SIPP**
  - **Incremento degli indirizzi a 128 bit**
- 1995: IPv6 (RFC 1883) – S. Deering, R. Hinden
  - **Ulteriore specifica**
- 1996: IPv6 diventa Proposed Standard - Nasce 6BONE
- 1998: IPv6 (RFC 2460) – S. Deering, R. Hinden
  - **Piccole modifiche (ad es 24->20 bit il campo Flow Label)**
- 1998: IPv6 Protocol Specification è Draft-Standard e poi Standard
- 1998-2000: Raddoppiano i nodi connessi a 6BONE - Si sperimentano funzionalità avanzate (multicast, anycast, mobilità) e tecniche di transizione IPv4-IPv6
- 2000-2005: Altri standard definiscono l'indirizzamento, transizione, etc.

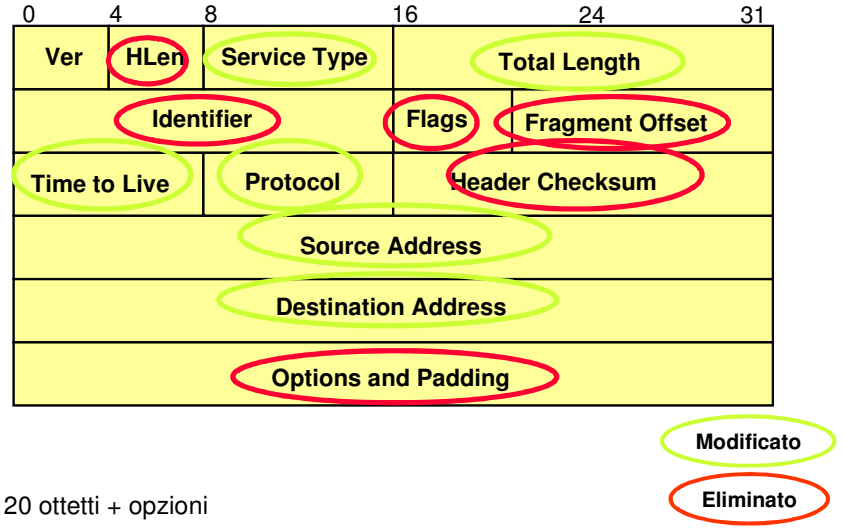
IPv6 vs IPv4

- Formato ottimizzato del datagramma
  - **il formato del datagramma è stato ottimizzato per rendere più efficiente il suo processamento**
  - **sono stati eliminati campi superflui e adottato uno schema di allineamento a 32 bit**
- Identificativo di flusso
  - **rende possibile distinguere i flussi emessi da una sorgente**
  - **apre la possibilità di un trattamento differenziato dei flussi in rete**
- Indirizzamento a 128 bit

IPv6 vs IPv4

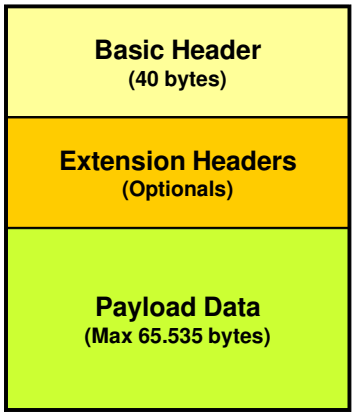
- Eliminazione del header checksum
- Segmentazione effettuata solo dall'host sorgente
  - **i router intermedi non possono segmentare un datagramma**
- Estensioni dell'header
  - **rendono possibile l'implementazione di opzioni**
- Inclusione di procedure di security
  - **supporto di meccanismi di autenticazione e confidenzialità a livello di rete**

Header IPv4



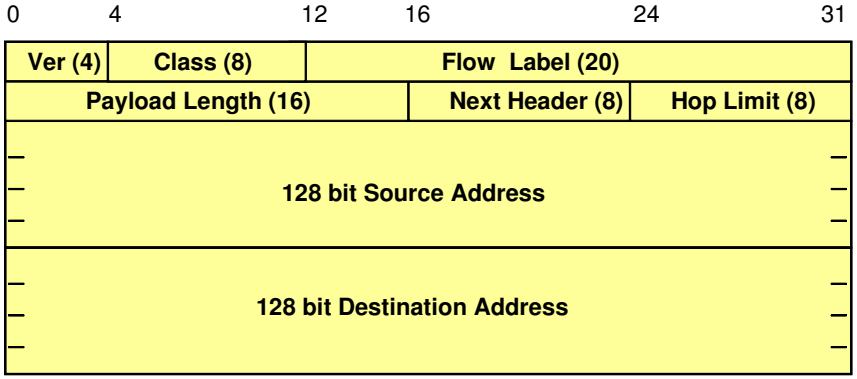
- 20 ottetti + opzioni
- 13 campi obbligatori (di cui 1 composto da 3 bit di flag)

Formato generale datagramma IPv6



- Basic header
  - **contiene le informazioni comuni a tutti i datagrammi**
- Extension Headers
  - **contengono le opzioni utilizzate dai router intermedi e/o dall'host di destinazione**
- Payload Data
  - **sono i bit informativi elaborati dall'host di destinazione**

IPv6 Basic Header



40 ottetti
8 campi

Spazio di indirizzamento IPv6

- Campi di indirizzo di 128 bit (16 byte)
  - **n. di indirizzi IPv6 totale ≈ 3.40 * 10^{38}**
  - **paria a ≈ 6,65 * 10^{23} indirizzi/m^2 di superficie della Terra**
  - **o anche un indirizzo ogni ≈ 5 cm^3 di spazio nella sfera di raggio pari alla distanza Terra-Sole**

- In realtà, poiché 64 bit sono riservati per l'identificatore dell'interfaccia (indirizzi unicast), si avranno:
  - **n. totale di reti IP ≈ 1.8 * 1019**
  - **paria a ≈ 36.000 reti IPv6 per m^2 di superficie terrestre**

IPv6 Basic Header

- Version (4 bit)
  - **versione del protocollo (6), è possibile la coesistenza di più versioni di IP**

- Traffic Class (8 bit)
  - **Originariamente definito come campo Priority di 4 bit (RFC 1883)**
    - due classi di priorità
      - Congestion Controlled Traffic (livelli 0 - 7)
      - Noncongestion Controlled Traffic (livelli 8 - 15)
    - La relazione di priorità ha valore solo all'interno di una classe
    - Non è definita nessuna relazione di priorità tra datagrammi appartenenti a classi diverse
  - **Ridefinito come DS Field (Diffserv Field) da RFC 2474**
  - **Stabilisce la classe di traffico o più precisamente il PHB (per-hop behavior) con cui deve essere trattato il pacchetto**

IPv6 Basic Header

- Flow label (20 bit)
  - **Originariammente di 24bit (RFC 1883)**
  - **Ha lo scopo di identificare, insieme ai campi source e destination address, un particolare flusso di datagrammi**
  - **E' un numero scelto casualmente dall'host mittente nell'intervallo 1-0xFFFFFF (0 identifica traffico non associato a nessun flusso)**
  - **Dovrebbe aiutare a ridurre i tempi di elaborazione dei datagrammi nei router di rete**
  - **Potrebbe facilitare l'istradamento dei datagrammi in hardware mediante consultazione di tabelle di cache evitando l'applicazione della normale procedura di IP forwarding**
  - **Il concetto di flusso si adatta anche a procedure di riservazione di risorse per traffico con qualità di servizio garantita (protocollo RSVP)**

Flow Label (more..)

- **Traditionally, flow classifiers have been based on the 5-tuple of source and destination addresses, ports, and the transport protocol type**
  - some of these fields may be unavailable due to either fragmentation or encryption, or locating them past a chain of IPv6 option headers may be inefficient
  - if classifiers depend only on IP layer headers, later introduction of alternative transport layer protocols will be easier
- **The usage of the 3-tuple of the Flow Label and the Source and Destination Address fields enables efficient IPv6 flow classification**
  - only IPv6 main header fields in fixed positions are used

IPv6 Basic Header

- Payload Length (16 bit)
  - **indica la lunghezza in byte del datagramma IP (escluso il basic header)**
  - **normalmente la lunghezza massima del payload è 65.535 byte; è possibile l'uso dell'opzione "jumbo payload" (hop-by-hop options header)**
- Next Header (8 bit)
  - **identifica quali header seguono il basic header nel datagramma**
  - **alcuni valori:**

0	Hop-by-hop options header	46	Resource Reservation Protocol
4	Internet Protocol	50	Encapsulating Security Payload
6	Transmission Control Protocol	51	Authentication Header
17	User Datagram Protocol	58	Internet Control Message Protocol
43	Routing Header	59	No Next Header
44	Fragment Header	60	Destination Options Header

IPv6 Basic Header

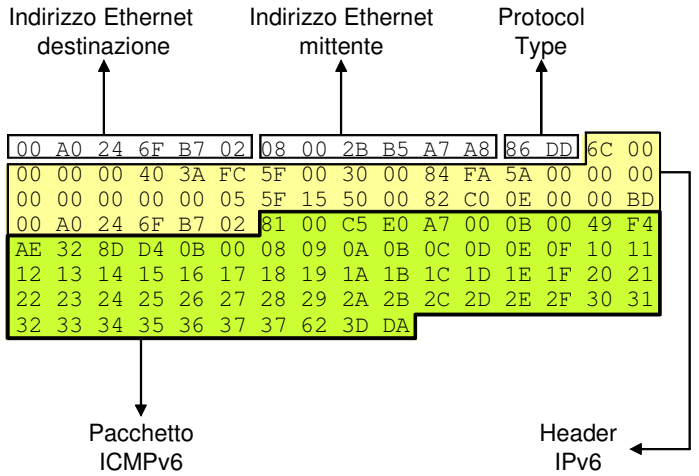
- Hop Limit (8 bit)
  - **l'host sorgente indica il numero massimo di tratte di rete che il datagramma può attraversare**
  - **ogni router decrementa di una unità tale campo**
  - **se il contatore si azzera prima che la destinazione sia raggiunta, il datagramma è scartato**
  - **evita gli effetti di eventuali loop in rete e può essere utilizzato per effettuare delle ricerche di host in rete a distanza prefissata**
- Source Addresses (128 bit)
  - **indica l'indirizzo IP del host sorgente**
- Destination Addresses (128 bit)
  - **indica i'indirizzo IP del/degli host di destinazione**
  - **potrebbe non essere il destinatario finale, se è presente un Routing header**

IPv6: migliori prestazioni

- IPv6 Header di dimensione fissa
  - **Extension headers aggiuntivi se serve (non limitati a 40 bytes come in IPv4)**
- Meno campi nell'intestazione
  - **Elaborazione più veloce nella maggior parte dei casi**
  - **Non c'è più il checksum (che doveva essere ricalcolato per effetto della modifica del TTL)**
- Elaborazione degli "Extension Header"
  - **Per la maggior parte solo nell'host di destinazione**
- Assenza di segmentazione nella rete

MAGGIORE SEMPLICITA' DI ESECUZIONE SU SILICIO

Esempio di pacchetto IPv6



IPv6 Extension Headers

- Meccanismo utilizzato per inviare informazioni addizionali alla destinazione o ai sistemi intermedi
- Sostituiscono le opzioni presenti nel header IPv4
- Tutti gli Ext. Header hanno lunghezza multipla intera di 8 ottetti (64 byte)
- Quando sono presenti più Extension Header il loro ordine non è arbitrario (ma stabilito nella RFC 2460)
- Il contenuto e la semantica di un Extension Header determinano se prcedere ad elaborare il prox header
- Nella maggior parte dei casi sono trattate solo dai nodi estremi

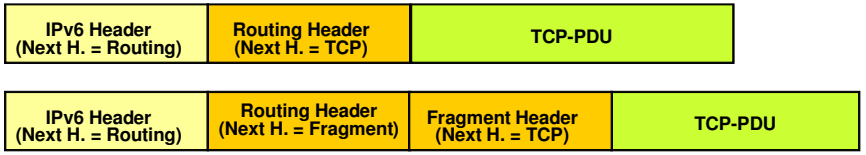
Basic header
Extension header 1 (optional)
. . .
Extension header n (optional)
Upper layer protocol header (e.g. TCP)

IPv6 Extension Headers

- Gli Extension headers sono inseriti tra l'header IPv6 e l'header di protocollo superiore (e.g. TCP)

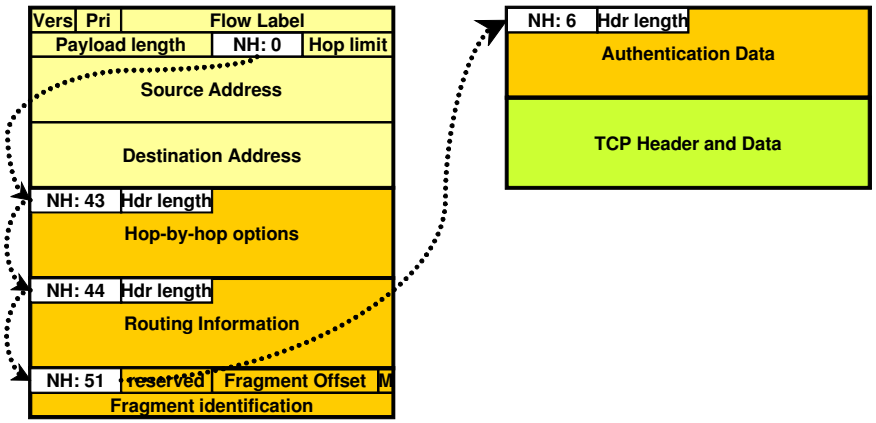
Pacchetto IPv6 "normale"

Esempi di pacchetti IPv6 con "Extension Headers"



IPv6 Extension Headers

- Ogni EH ha una lunghezza uguale a 8*(1+n) byte, ove n è il contenuto del campo Header Extension Length
- Ogni EH contiene un puntatore al prossimo EH (Next header)



IPv6 Extension Headers

- Sono definite i seguenti Extension Header opzionali con questo ordine
  - **Hop-by-hop Options header**
  - **Destination Options header**
    - **Nel caso di opzioni che devono essere elaborate anche dai nodi intermedi indirizzati nel Routing header**
  - **Routing header**
  - **Fragment header (solo destinazione)**
  - **Authentication header (solo destinazione)**
  - **Encapsulation security payload header (solo destinazione)**
  - **Destination Options header (solo destinazione)**
- Ogni EH può apparire solo 1 volta, ad eccezione di Destination Options che può apparire 2 volte

IPv6 Extension Headers

- Hop-by-hop Options (Type=0)
  - **Racchiude opzioni che coinvolgono tutti i router attraversati: es. Jumbo Option (per pacchetti >65.575 ottetti)**
- Routing (Type=43)
  - **Specifica una lista di router da attraversare (Strict o Loose)**
- Fragment (Type=44)
  - **I pacchetti senza questo header non possono essere frammentati. Se eccedono la MTU del link (MTU minima = 576 bytes in IPv6!!) devono essere scartati e ne viene data comunicazione alla sorgente, la quale potrà**
    - i) diminuire la lunghezza dei successivi pacchetti o
    - ii) inviare pacchetti con l' EH Fragment
- Destination Options (Type=60)
  - **Racchiude opzioni che devono essere elaborate solo nel/nei nodo/nodi di destinazione**
- Authetication (Type=51)
- Encapsulating Security Payload (ESP) (Type=50)

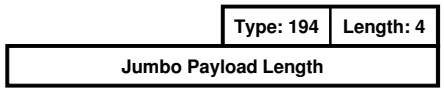
Hop-by-Hop Options

- Contiene le opzioni per ogni sistema intermedio sul percorso del datagramma
  - **E' elaborato anche nei nodi intermedi**
- E' costituito da un numero variabile di opzioni codificate come triplette TLV (Type-Lenght-Value)
  - **Type (8 bit): indica il tipo di opzione**
  - **Length (8 bit): indica la lunghezza del campo value**
  - **Value: trasporta il valore dell'opzione e alcune indicazioni per il router utili per l'elaborazione dell'opzione**
- Opzioni definite:

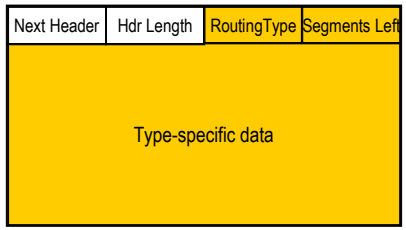
Type	Option	Size	Allineamento
0	Pad 1	1 byte	nessuno
1	Pad N	2+n bytes	nessuno
194	Jumbo Payload Length	2+4 bytes	4·n + 2

Hop-by-Hop Options Header

- Pad1 e PadN options
  - **servono a inserire degli ottetti di riempimento nell'header per questioni di allineamento**
  - **consentono di aumentare la velocità di elaborazione dei router**
    - Pad1: 1 byte: [type=0]
    - Pad2: 2+n bytes: [type=1 (1 byte), len (1 byte), N bytes]
- Jumbo Payload Length Option
  - **serve ad aumentare la lunghezza massima del datagramma rispetto a quanto consentito dal basic header**
  - **nel caso di tale opzione sia utilizzata il campo payload length del basic header deve contenere il valore 0**

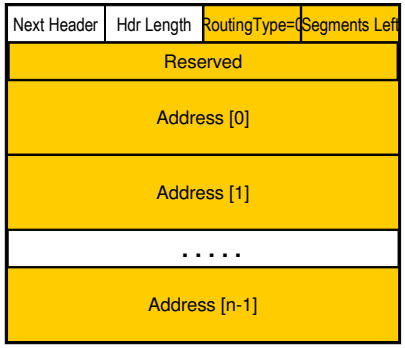


Routing Header



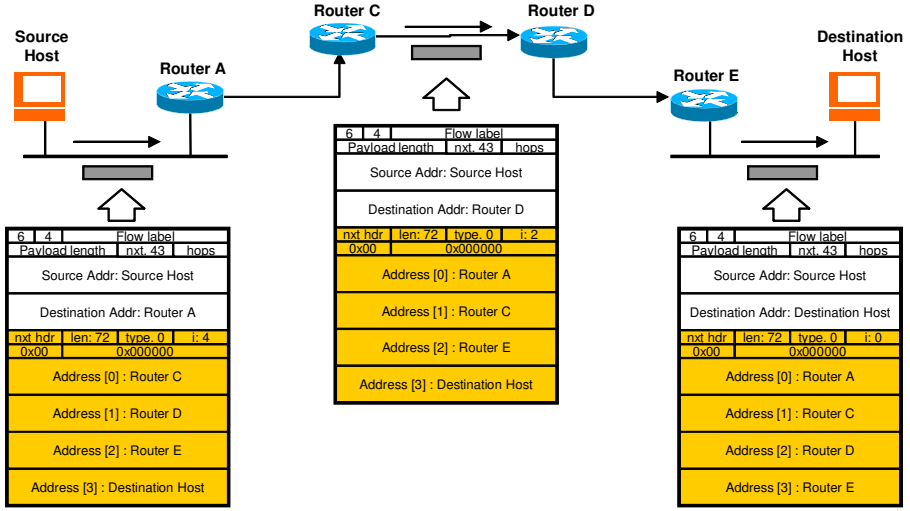
- Fornisce ai router indicazioni per l'instradamento del datagramma, forzando l'uso di un particolare cammino
- Routing Type
  - **identifica il tipo di routing**
- Segments left
  - **indica il numero di indirizzi che devono essere ancora elaborati**
  - **ogni router indirizzato decrementa tale valore**
- Type-specific data
  - **campo di lunghezza variabile multipla di 8 bytes**
  - **dipende dal Routing Type**

Routing Type 0



- Address [i]
  - **i = n - (Segments left)**
  - **i-esimo router da utilizzare come destinazione intermedia lungo il percorso**
  - **l'i-esimo address viene scambiato con il destination address nel Header**

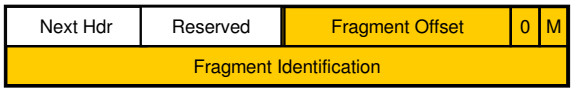
Routing Header



Fragment Header

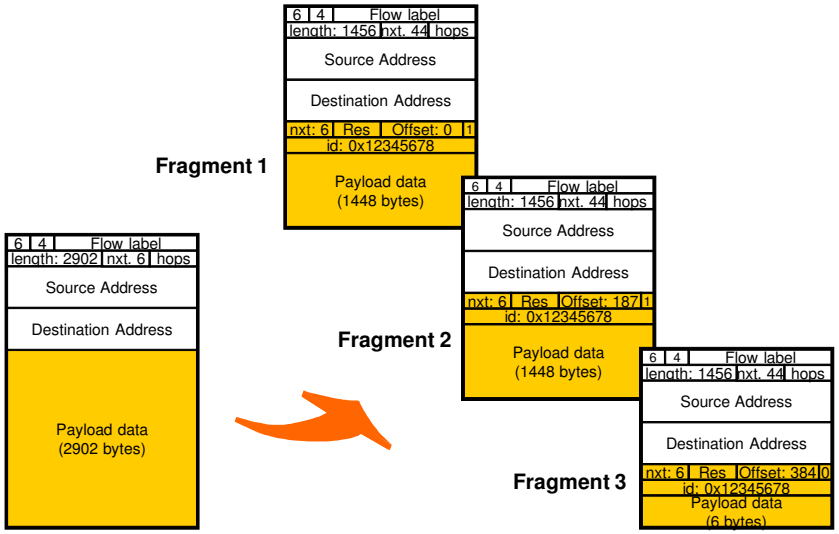
- Header di lunghezza fissa pari a 64bit
- Usato per la segmentazione e ricostruzione dei datagrammi
- In IPv6 solo il nodo mittente può effettuare la frammentazione di un datagramma
  - **Quando un nodo intermedio riceve un datagramma di lunghezza superiore alla MTU lo scarta e notifica l'errore al mittente tramite ICMP**
- Ogni datagramma è composto da una parte non frammentabile e da una parte frammentabile
  - **La parte non frammentabile è formata dagli header che devono poter essere elaborati anche nei nodi intermedi (Basic Header, Hop by Hop Header e Routing Header)**
- Tutti i frammenti tranne l'ultimo hanno lunghezza multipla di 8 bytes

Fragment Header



- Fragment Offset
  - **specifica la posizione (in unità di 8 bytes) del primo byte del frammento nel datagramma originale**
- Bit M (More Fragments)
  - **il suo valore è posto a 1 in tutti i frammenti del datagramma tranne l'ultimo in cui è posto ad 0**
- Fragment identification
  - **identifica il datagramma a cui il frammento appartiene**

Fragment Header

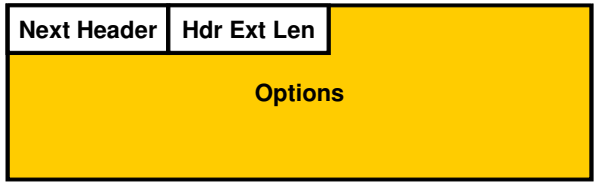


Destination Options Header

- Trasporta informazioni che devono essere lette dal destinatario (o eventualmente da alcuni router intermedi)
  - **Se le Destination options sono effettivamente per l'utente finale, questa extension header è l'ultima**
  - **Se sono dirette ad un router intermedio, tale opzione è usata in unione con l'opzione routing header e precede quest'ultima**
- Si possono inserire due destination options per distinguere le informazioni dirette ai router intermedi da quelle dirette all'utente finale

Destination Options Header

- Il campo Options ha lunghezza variabile e contiene una o più opzioni codificate come sequenze di triplette TLV (Type-Lenght-Value)
  - **Attualmente la RFC 2460 definisce solo le opzioni Pad1 e PadN (le stesse dell'Hop by Hop Extension Header)**



- Nota: nuove informazioni per il destinario possono essere inserite come
  - **Destination Options**
  - **Extension Header separato**

Indirizzamento

<draft-ietf-ipv6-addr-arch-v4-04.txt> "IP Version 6 Addressing Architecture" (May 2005)
(will obsolate RFC 3513 (2003)
that obsolates RFC 2373 (1998)
that has obsolated RFC 1884 (1995))

RFC 3587 (2003) "IPv6 Global Unicast Address Format"
(obsolates RFC 2374 (1998) that has obsolated RFC 2073 (1997))

Indirizzamento

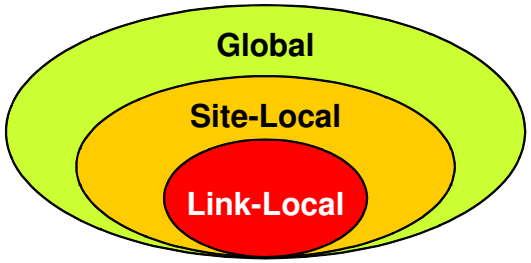
- Schema di assegnazione degli indirizzi gerarchico mirato a minimizzare le dimensioni delle tabelle di instradamento sui router
- Indirizzi globali per Internet e locali per Intranet
- Indirizzi associati alle interfacce e possibilità di avere più indirizzi per ogni interfaccia

Indirizzi IPv6: Categorie

- Gli indirizzi IPv6 possono essere:
  - **Unicast**
    - analogo ad IPv4
  - **Multicast**
    - analogo ad IPv4
  - **Anycast**
    - Un indirizzo IPv6 anycast è un indirizzo assegnato a più di una interfaccia
    - Un pacchetto inviato ad un indirizzo anycast viene instradato verso la più vicina interfaccia con quell'indirizzo, in accordo alla distanza rilevata dal protocollo di routing
- Sono stati eliminati gli indirizzi broadcast
  - **al loro posto sono utilizzati gli indirizzi multicast**

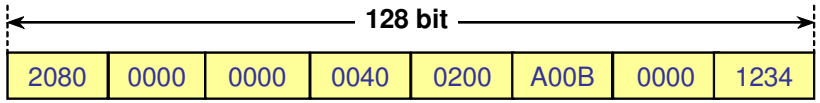
Indirizzi IPv6: Visibilità

- Come in IPv4
  - **gli indirizzi sono assegnati alle interfacce**
  - **un'interfaccia può avere più indirizzi**
- A differenza di IPv4
  - **Gli indirizzi hanno un ambito di validità**
    - Link Local
    - Site Local
    - Global



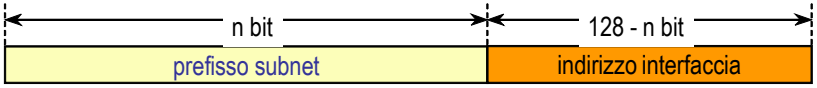
- Ogni interfaccia deve avere almeno un indirizzo unicast "link-local"

Indirizzi IPv6: Notazione esadecimale



- Rappresentati in esadecimale come 8 blocchi da 16 bit separati da ":" 2080:0000:0000:0040:0200:A00B:0000:1234
- Semplificazioni:
  - **in ogni blocco si possono omettere gli zero iniziali**
    - 2080:0:0:40:200:A00B:0:1234
  - **si può sostituire una SINGOLA serie di più uno o più blocchi consecutivi da 16 bit tutti a zero con "::"**
    - 2080::40:200:A00B:0:1234
- Gli indirizzi di compatibilità IPv4 si scrivono:
  - **0:0:0:0:0:0:A00:1**
  - **::A00:1**
  - **::10.0.0.1**

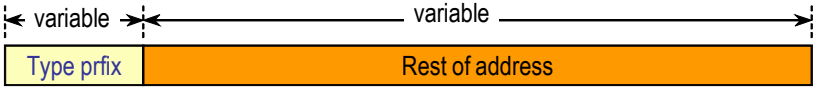
Address Prefix



- Scompare il concetto di Netmask, sostituito da quello di "Prefix" che indica:
  - **Il tipo di indirizzo**
  - **La sottorete a cui appartiene l'interfaccia**
- Il prefix si indica aggiungendo ad un indirizzo "/N", dove N è la lunghezza in bit del prefix
- Esempio:
  - **FEDC:0123:8700::1:2:3:4 /36**

Indirizzi IPv6: Spazio di indirizzi

- Gli indirizzi IPv6 iniziano con un primo campo di lunghezza variabile chiamato "Type prefix"



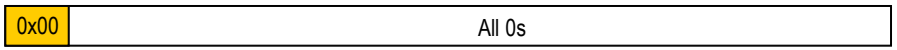
- Il "Type prefix" identifica la tipologia di indirizzo

Address type	*Binary prefix*	*IPv6 notation*
Unspecified	00...0 (128 bits)	::/128
Loopback	00...1 (128 bits)	::1/128
Multicast	11111111	FF00::/8
Link-local unicast	1111111010	FE80::/10
Site-local unicast	1111111011	FEC0::/10
Global unicast	(everything else)	

- Anycast addresses are taken by unicast addresses spaces

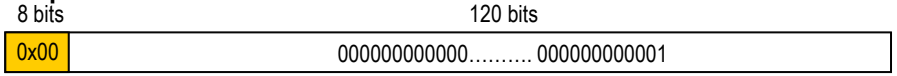
Reserved Addresses with prefix 0x00

Unspecidfied address (only as source address)



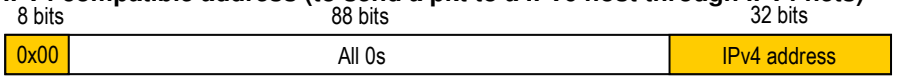
::0

Loopback address



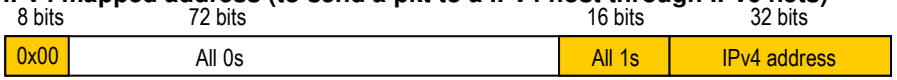
::1

IPv4 compatible address (to send a pkt to a IPv6 host through IPv4 nets)



::x.y.z.w

IPv4 mapped address (to send a pkt to a IPv4 host through IPv6 nets)



::FFFF:x.y.z.w

Unspecified and Loopback Addresses

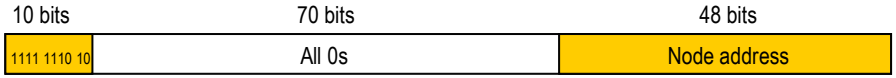
- Unspecified address 0:0:0:0:0:0:0:0
  - **::**
  - **It must never be assigned to any node**
  - **It indicates the absence of an address**
  - **e.g. as src addr of any pkts sent by an initializing host**
  - **must not be used as the dest addr of pkts**
  - **a packet with a src addr :: must never be forwarded by an IPv6 router**
- Loopbackaddress 0:0:0:0:0:0:0:1
  - **::1**
  - **It may be used by a node to send an IPv6 packet to itself**
  - **It must not be assigned to any physical interface**
  - **It is treated as having link-local scope (of a virtual interface called "loopback interface")**
  - **must not be used as src addr in pkts sent outside of a single node**
  - **a packet with dest addr of loopback must never be sent outside**

IPv6 Addresses with Embedded IPv4 Addresses

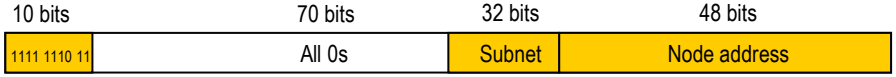
- IPv4-Compatible IPv6 Address
  - **was defined to assist in the IPv6 transition (to send a pkt to a IPv6 host through IPv4 nets)**
  - **is now deprecated because the current IPv6 transition mechanisms no longer use these addresses**
- IPv4-Mapped IPv6 Address
  - **IPv6 address that holds an embedded IPv4 address is defined**
  - **is used to represent the addresses of IPv4 nodes as IPv6 addresses (when sending a pkt to a IPv4 host through IPv6 nets)**

Local Addresses

- Link local address (only within a subnet, e.g. LAN)



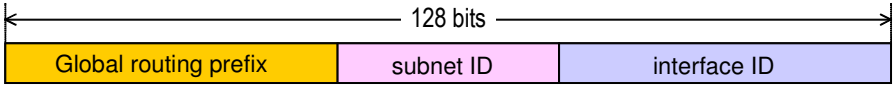
- Site local address (only though private subnets)



- Note:
  - **Site-local addresses were originally designed to be used for addressing inside of a site without the need for a global prefix**
  - **Site-Local addresses are now deprecated as defined in**

Global Unicast Addresses

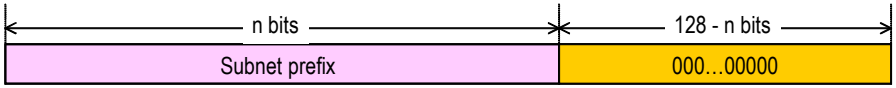
- The general format for IPv6 global unicast addresses is as follows



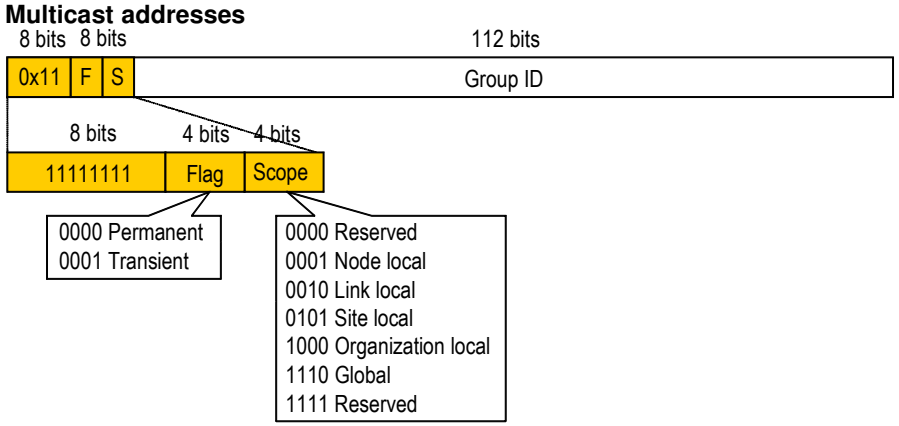
- where
  - **global routing prefix is a (typically hierarchically-structured) value assigned to a site (a cluster of subnets/links)**
  - **subnet ID is an identifier of a link within the site**
  - **interface ID uniquely identify interfaces on a link; it can be derived directly from that interface's link-layer address**
- All global unicast addresses other than those that start with binary 000 have a 64-bit interface ID field (i.e., n + m = 64),
- Global unicast addresses that start with binary 000 have no such constraint on the size or structure of the interface ID field

Indirizzi Anycast

- Formalmente identici agli indirizzi Unicast
- Assegnati simultaneamente a più interfacce
  - **Indicano il server più vicino al mittente che fornisce un dato servizio**
  - **Sono instradati dai router come indirizzi unicast**
  - **Nell'ambito della area topologica identificata dal "subnet prefix" ogni singola interfaccia appartenente al gruppo è annunciata individualmente dai protocolli di routing**
  - **I nodi a cui sono assegnati indirizzi anycast sono esplicitamente configurati per sapere che il loro indirizzo è di tipo anycast**
- Sino ad ora è stato definito un solo indirizzo anycast:
  - **subnet router anycast address:**

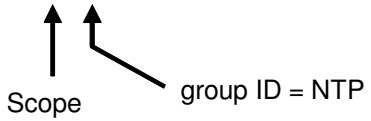


Multicast Addresses



Multicast Addresses: Esempio

- NTP: Network Time Protocol
  - **FF01::43 - tutti i server NTP sul nodo mittente**
  - **FF02::43 - tutti i server NTP sullo stesso link del mittente**
  - **FF05::43 - tutti i server NTP nello stesso sito del mittente**
  - **FF08::43 - tutti i server NTP nell'organizzazione mittente**
  - **FF0E::43 - tutti i server NTP presenti sulla rete**



ICMPv6

<draft-ietf-ipngwg-icmp-v3-07.txt> "ICMPv6 for the IPv6 Specification" (2005)
(will obsolete RFC 2463 (1998) that obsoletes RFC 1886 (1995))

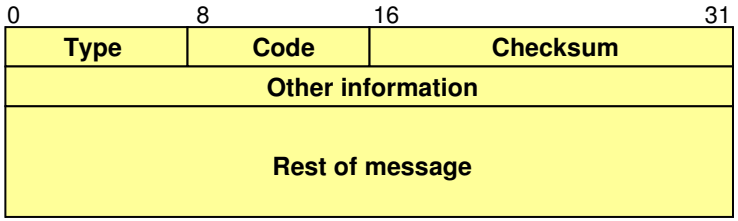
<draft-ietf-ipv6-2461bis-05.txt> "Neighbor Discovery for IPv6" (2005)
(will obsolate RFC 2461 that obsoletes RFC 1970)

Messaggi ICMPv6

- Internet Control Message Protocol per IPv6, Ha tre impieghi principali:
  - **diagnostica**
  - **neighbor discovery**
  - **gestione dei gruppi multicast**
- Svolge funzionalità che in IPv4 erano principalmente svolte da:
  - **ICMP**
  - **ARP (Address Resolution Protocol)**
  - **IGMP (Internet Group Membership Protocol)**
- Prevede due classi di messaggi
  - **messaggi di errore**
  - **messaggi informativi**
    - diagnostica
    - neighbor/router discovery
    - gestione dei gruppi multicast

Formato del pacchetto ICMP

- Il messaggio ICMPv6 è trasportato in un pacchetto IPv6 ed è indicato dal valore 58 nel campo Next Header
  - **La dimensione complessiva di un pacchetto ICMPv6 non deve superare 576 bytes (MTU garantita da qualunque link)**



- Type (8 bits)
  - **indica il tipo di messaggio ICMPv6**
- Code (8 bits)
  - **specifica la ragione del particolare messaggio**

Messaggi ICMPv6

- Error messages (types from 0 to 127)
  - **Destination Unreachable**
  - **Packet Too Big**
  - **Time Exceeded**
  - **Parameter Problem**
  - **Redirection**
- Informational messages (types from 128 to 255)
  - **Echo Request and Reply**
  - **Router Solicitation and Advertisement**
  - **Neighbor Solicitation and Advertisement**
  - **Group management (Multicast Listener Discovery Protocol)**

Neighbor Discovery

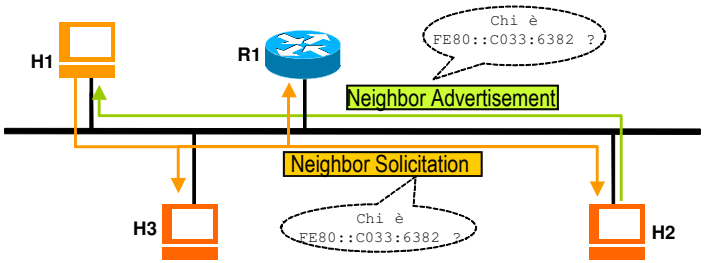
- It solves a set of problems related to the interaction between nodes attached to the same link
  - **Router Discovery**
    - how hosts locate routers that reside on an attached link
  - **Prefix Discovery**
    - how hosts discover the set of address prefixes that define which destinations are on-link for an attached link
  - **Parameter Discovery**
    - How a node learns such link parameters as the link MTU or such Internet parameters as the hop limit value to place in outgoing packets
  - **Address Autoconfiguration**
    - Introduces the mechanisms needed in order to allow nodes to automatically configure an address for an interface

Neighbor Discovery (cont.)

- **Address resolution**
  - How nodes determine the link-layer address of an on-link destination (e.g., a neighbor) given only the destination's IP address
- **Next-hop determination**
  - The algorithm for mapping an IP destination address into the IP address of the neighbor to which traffic for the destination should be sent
  - The next-hop can be a router or the destination itself
- **Neighbor Unreachability Detection**
  - How nodes determine that a neighbor is no longer reachable
  - For neighbors used as routers, alternate default routers can be tried
- **Duplicate Address Detection**
  - How a node determines that an address it wishes to use is not already in use by another node
- **Redirect**
  - How a router informs a host of a better first-hop node to reach a particular destination

Neighbor Discovery

- Neighbor Solicitation
  - **Inviato da un host per determinare l'indirizzo "Link Layer" di un vicino, o per verificare che il vicino sia ancora raggiungibile**
  - **Usato anche per Duplicate Address Detection**
  - **Indirizzo:** FF02:0000:0000:0000:0000:0001:xxxx:yyyy
    - dove xxxx:yyyy sono i 32 bit meno significativi dell'indirizzo IPv6 sollecitato
- Neighbor Advertisement
  - **Risposta al messaggio di Neighbor Solicitation**
  - **Può essere inviato anche per notificare un cambiamento di indirizzo**



Address Resolution

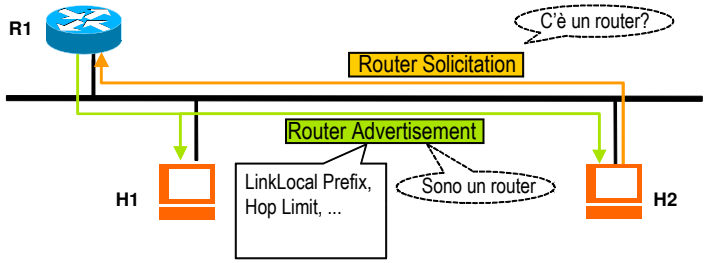
- Consente di tradurre un indirizzo IP nel corrispondente indirizzo Link Layer (e.g. MAC address)
- In IPv4 è realizzato tramite ARP
- In IPv6 è realizzato dallo strato IP tramite ICMP
- Si basa su IP multicast
  - **consente un risparmio di elaborazione rispetto al broadcast**
- Per risolvere un indirzzo IP nel corrispondente indirizzo MAC
  - **viene inviata una richiesta ICMP di tipo Neighbor Solicitation**
  - **il messaggio contiene il proprio indirizzo IP, il proprio indirizzo MAC e quello IP della destinazione**
  - **viene inviato ad 1 indirizzo multicast ottenuto facendo seguire a un prefisso multicast convenzionale (prefisso FF02::1/96) gli ultimi 32 bit dell'in-dirizzo IP della destinazione**
  - **il destinatario risponde con un Neighbor Advertisement in cui è inserito il proprio indirizzo MAC**
- Le risposte sono memorizzate in una memoria cache

Router Discovery

- Router Discovery
  - **Individuare i router attestati al collegamento e i prefissi di rete su di esso configurati**
  - **permettere la Stateless Address Autoconfiguration dei nodi del link**
  - **permettere ai nodi di determinare per ogni loro interfaccia lo stato di "on link" ovvero "off link"**
- I router periodicamente inviano in rete messaggi di Router Advertisement
  - **Questi messaggi contengono diverse informazioni, tra cui**
    - indirizzo link-local del router
    - la lista dei prefissi di rete configurati sul collegamento
    - MTU sul link
    - vari parametri (timeout..)
- Le macchine terminali possono così individuare quali destinazioni sono direttamente raggiungibili
- I router possono essere "sollecitati" tramite ICMP Router Solicitation

Router Discovery

- Router Solicitation
  - **Inviato da un host che attiva un'interfaccia; ha come destination address l'indirizzo multicast "all-router"**
- Router Advertisement
  - **Il router annuncia la sua presenza, fornisce i prefissi da usare sul link, suggerisce il parametro di hop limit, MTU, etc.**
  - **Periodicamente o in risposta ad un Router Solicitation**



Host Autoconfiguration

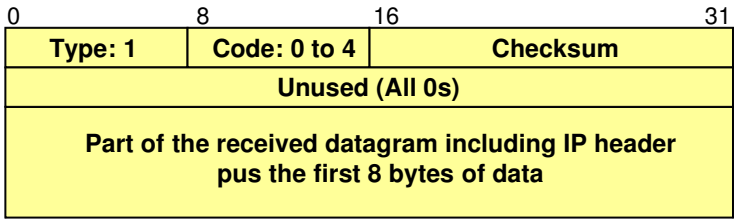
- IPv6 consente differenti forme di autoconfigurazione, che permettono di interconnettere gli host in modo Plug-and-Play
- 2 approcci Plug-and-Play:
  - **Stateful**
    - DHCPv6 (Dynamic Host Configuration Protocol) simile ad IPv4
    - prevede lo scambio di informazioni di configurazione tra il nodo che deve configurare l'indirizzo e un server secondo il protocollo DHCPv
    - Permette di controllare in modo più diretto l'assegnazione degli indirizzi
  - **Stateless**
    - Colloquio tra Host e Router
    - il prefisso di rete è ottenuto tramite una procedura che strutta messaggi ICMP Router Solicitation/Advertisement
    - il nodo ricostruisce il proprio indirizzo da solo, concatenando il prefisso di rete all'indirizzo PH/DL (e.g. MAC address)
    - Usata quando non ci sono particolari esigenze di attribuire specifici indirizzi agli host
  - **Le due modalità possono coesistere**

Multicast Listener Discovery Protocol (MLDv2)

- Multicast Listener Discovery Protocol (MLD)
- Used by IPv6 routers to
  - **discover the presence of multicast listeners (i.e., nodes that wish to receive multicast packets) on their directly attached links, and**
  - **discover specifically which multicast addresses are of interest to those neighboring nodes**
- MLDv2 is a translation of the IGMPv3 protocol for IPv6 semantics

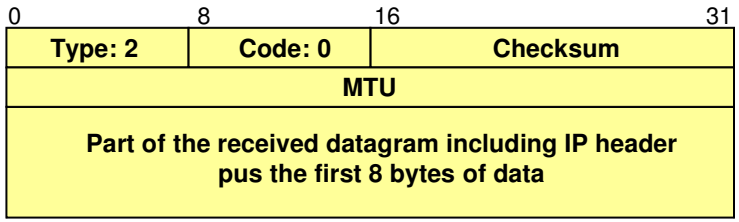
Destination unreachable

- Destination unreachable (type=1)
  - **no path to destination (code=0)**
  - **communication is prohibited (code=1)**
  - **strict source routing is impossible (code=2)**
  - **destination address is unreachable (code=3)**
  - **port is not available (code=4)**



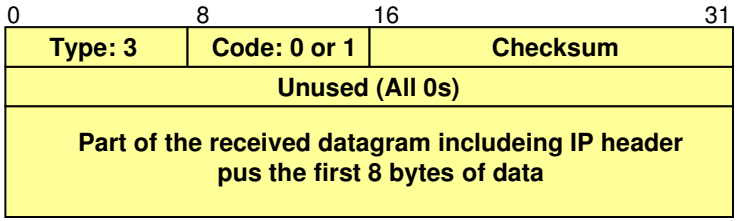
Packet too big

- packet too big (type=2)
  - **code=0**



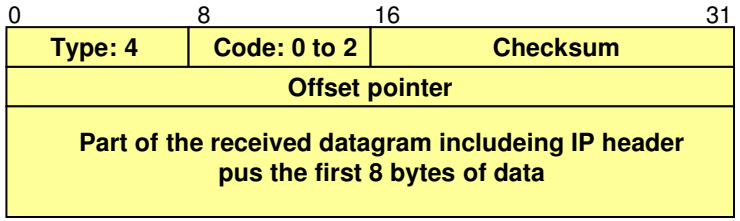
Time exceeded

- Time exceeded (type=3)
  - **code=0 when a datagram is discarded**
  - **code=1 when a fragment is discarded**



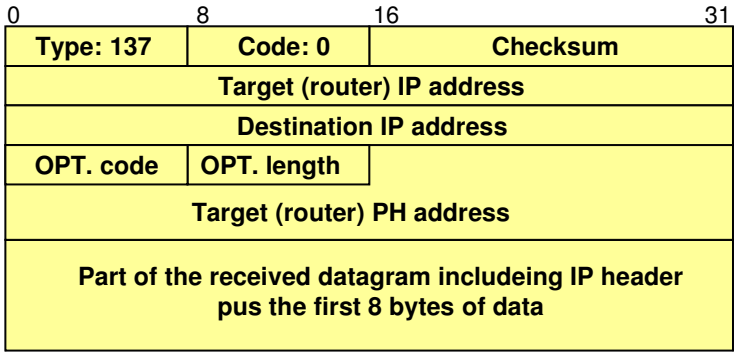
Parameter problems

- Parameter problems (type=4)
  - **error or ambiguity in one header field (code=0)**
  - **unrecognized extension header (code=1)**
  - **unrecognized option (code=2)**



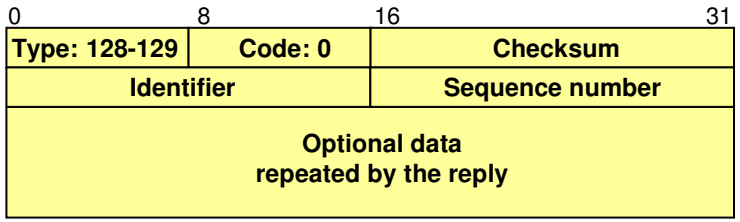
Redirection

- Redirection (type=137)
  - **code=0**



Echo request and reply

- Echo request and reply (type=128 and 129)
  - **Echo request (type=128, code=0)**
  - **Echo reply (type= 129, code=0)**



Router solicitation and advertisement

- Router solicitation (type= 133, code=0)

Type: 133	Code: 0	Checksum
Unused (All 0s)		
OPT. Code:1	OPT. length	
Host PH address		

- Router advertisement (type= 134, code=0)

Type: 134	Code: 0	Checksum
Max hop	M O Unused	Router lifetime
Reachability lifetime		
OPT. Code: 1	OPT. length	
Router PH address		
OPT. Code: 1	OPT. length	Unused (All 0s)
MTU size		

Neighbor solicitation and advertisement

- Neighbor solicitation (type=135, code=0)

Type: 135	Code: 0	Checksum
Unused (All 0s)		
Target IP address		
OPT. Code: 2	OPT. length	
Solicitor PH address		

- Neighbor advertisement (type=136, code=0)

Type: 136	Code: 0	Checksum
R S	Unused (All 0s)	
Target IP address		
OPT. Code: 2	OPT. length	
Target PH address		

Group management

- Query (type=130, code=0)

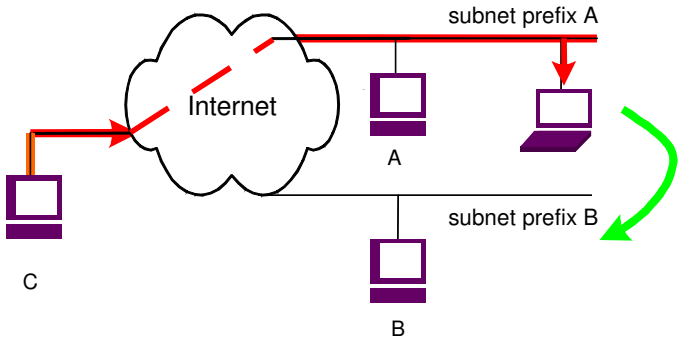
Type: 130	Code: 0	Checksum
Maximum response delay		Reserved
IP multicast address		

- Report (type=131, code=0)
- Termination (type=132, code=0)

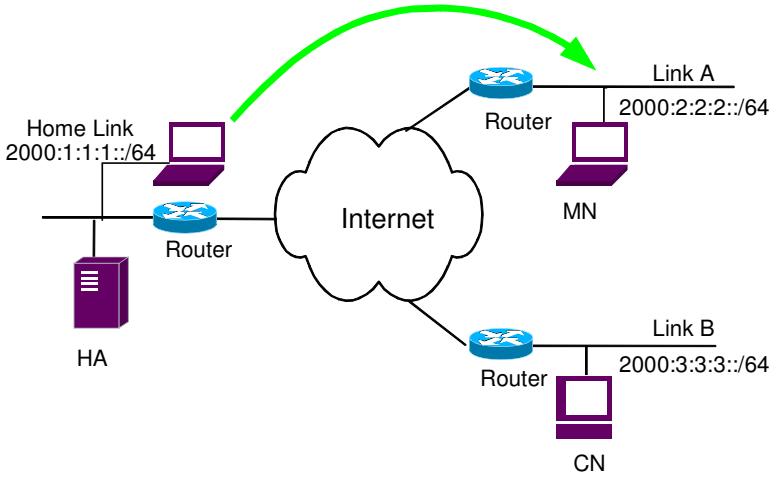
Type: 131-132	Code: 0	Checksum
Reserved		
IP multicast address		

Mobilità (MIPv6)

IP mobility



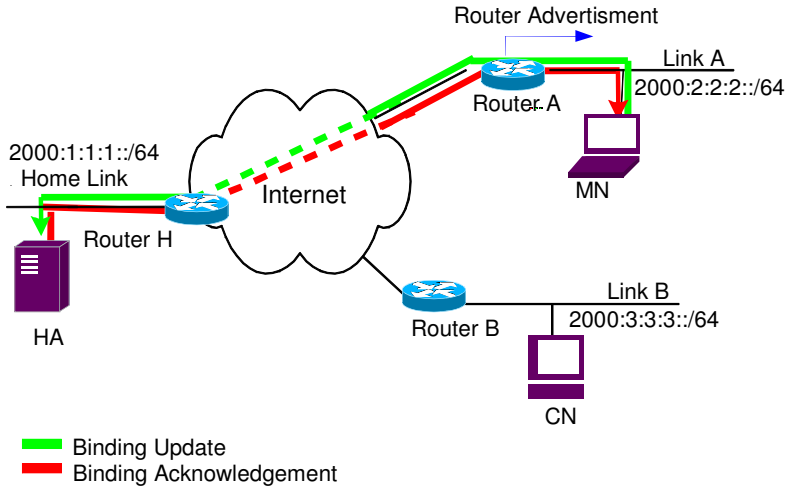
MIPv6 overview



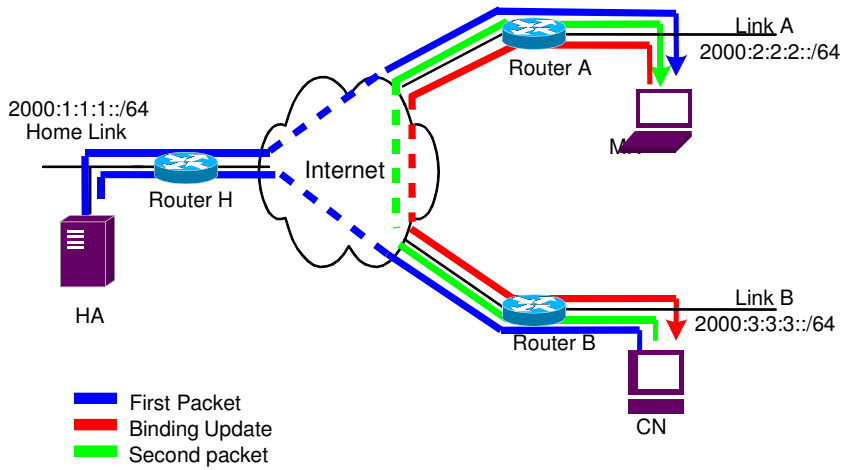
MIPv6 overview

- Advertisement from local router contains routing prefix
- Seamless Roaming: mobile node always uses home address (HAddr)
- Address autoconfiguration for care-of address (CoAddr)
- Binding Updates sent to Home Agent (HA) & correspondent nodes
- Mobile Node “always on” by way of home agent

Registration with HA



Triangolar routing and route optimization



Sicurezza

Sicurezza in IPv6

- Il protocollo IPv6 comprende funzionalità per assicurare
  - **l'autenticazione dei dati scambiati, cioè che il pacchetto**
    - non sia stato alterato durante il transito in rete
    - sia stato emesso effettivamente dal sender indicato nel datagramma
  - **la confidenzialità dello scambio informativo, ovvero la sicurezza che i dati non siano utilizzabili da altri se non il destinatario**
- IPv6 utilizza due appositi Extension Headers:
  - **Authentication Header (AH)**
  - **Encapsulating Security Payload (ESP)**

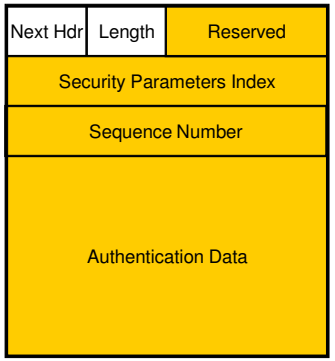
Sicurezza in IPv6

- Sia AH che ESP sfruttano il concetto di Security Association (SA) che specifica:
  - **l'algoritmo di encryption (default: Message digest 5 - MD5)**
  - **le chiavi di codifica**
  - **la durata limite dell'associazione**
  - **tipo di protezione (e.g. secret, top secret, etc.)**
- Per ogni comunicazione sicura in corso il nodo IPv6 gestisce una SA diversa, che è individuata da un SPI (Security Parameter Index)
- La negoziazione di una SA (e quindi del relativo SPI) è parte integrante dell'algoritmo di scambio delle chiavi

Authentication Header

- Assicura che il datagramma sia autentico
  - **non sia stato alterato durante il transito in rete**
  - **sia stato emesso effettivamente dal sender indicato nel datagramma, in modo da evitare attacchi di:**
    - IP spoofing (falsificazione dell'indirizzo IP del mittente)
    - Connection Hijacking (inserimento in una comunicazione in corso)

Authentication Header



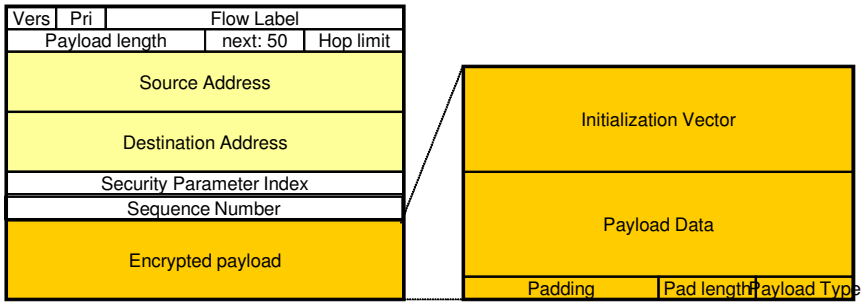
- Security Parameter Index (SPI) (32 bit)
  - **definisce la security association**
  - **è usato in associazione con l'indirizzo di destinazione**
- Authentication data (multiplo di 32 bit)
  - **Se è utilizzato l'algoritmo MD5 la lunghezza è di 16 bytes**
  - **I 128 bit di questo campo sono calcolati dall'algoritmo MD5 sulla base di**
    - i bit del datagramma
    - la chiave segreta di encryption del sender
- il ricevente verifica l'autenticità del datagramma, compiendo l'operazione inversa mediante la stessa chiave

Encapsulating Security Payload Header

- Assicura la confidenzialità dello scambio informativo
  - **Viene utilizzato per evitare attacchi di Packet Sniffing**
- E' sempre l'ultimo header nella catena
  - **Viene utilizzato per crittografare il contenuto del payload che lo segue immediatamente**
  - **Esso stesso è solo parzialmente in chiaro**

Encapsulating Security Payload Header

- è costituito dal SPI (in chiaro) e dal payload crptato
- L'Encrypted Payload è preceduto dal vettore di inizializzazione
- Il padding è usato per portare la lunghezza complessiva ad un multiplo di 64 byte
- Il campo payload type ripete il contenuto del campo Next Header



Transizione verso IPv6

RFC 4213 “Basic Transition Mechanisms for IPv6 Hosts and Routers”
(October 2005)
(obsoletes RFC 2893 (2000) that has bsoleted RFC1933 (1996))

RFC 2766 “NAT-PT” (2000)

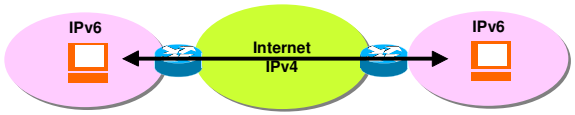
RFC 2529 “6over4” (1999)

RFC 3056 “6to4” (2001)

La transizione ad IPv6

- Elemento chiave per una transizione con successo ad IPv6 è la compatibilità con la base istallata IPv4
- Durante la fase di transizione sono necessari meccanismi per:
  - **Permettere il colloquio tra nuovi host IPv6 e la Internet preesistente IPv4**

  - **Realizzare connettività tra isole IPv6 utilizzando l’infrastruttura IPv4**



- Inoltre, reti IPv6 verranno utilizzate per intercconnettere reti IPv4 (e.g. backbone UMTS)

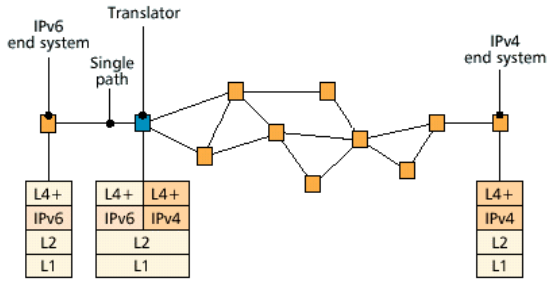
Transition mechanisms

- Transition mechanisms generally come in one one three forms
  - **Dual stack**
  - **Translation**
  - **Tunneling**

- Dual stack
  - **two protocol stacks are separately maintained and operate in parallel**
  - **the device can operate via either protocols**
  - **can be implemented in both end systems and network nodes**
    - in end systems: it enables both IPv4 and IPv6 applications to operate on the same node
    - in network nodes: it allows handling of both IPv4 and IPv6 packets

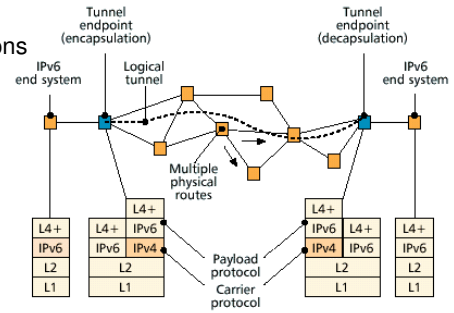
Transition mechanisms (cont.)

- Translation
  - **direct conversion of protocols**
  - **can occur at different layer in the protocol stack**
  - **often results in feature loss where there is not clear mapping (e.g. QoS)**
  - **can be either stateless or stateful**
  - **can be performed by end-systems or network nodes**
  - **in the latter case, it is considered “transparent”**



Transition mechanisms (cont.)

- Tunneling
  - **used to bridge compatible nodes (e.g. IPv6) across incompatible networks (e.g. IPv4)**
  - **IPv6-over-IPv4 or IPv4-over-IPv6**
  - **main problem is the configuration of tunnels (on tunnel endpoints)**
  - **tunnel endpoint addresses are generally attained:**
    - manually or by tools (e.g. tunnel broker)
    - through existing services such as DNS or DHCP options
    - by embedding information in the link layer addresses or IP addresses

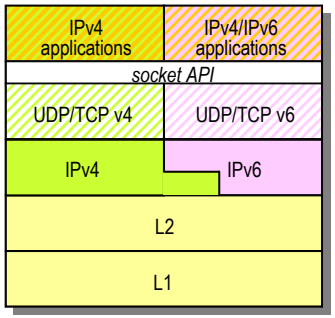


Transition mechanisms (cont.)

Name	Connectivity	Type	Location (End System/Network Device)
Dual stack	4-to-4 over 4, 6-to-6 over 6	Dual stack	In single ES or ND
SIIT	6-to-4, 4-to-6	Translator	In single ES or ND
Bump-in-Stack (BIS)	4-to-6	Translator	In single ES
Bump-in-API (BIA)	4-to-6	Translator	In single ES
NAT-PT	6-to-4, 4-to-6	Translator	In single ND
MTP	4-to-6,4-to-6 (multicast)	Translator	In single ND
TRT	6-to-4	Translator	In single ND
SOCKS64	4-to-6, 4-to-6	Translator	Between ES and ND
6over4	6-to-6 over 4	Tunnel	Between ES and ND
ISATAP	6-to-6 over 4	Tunnel	Between ES and ND
DSTM	4-to-4 over 6	Tunnel	Between ES and ND
Configured IP-in-IP	6-to-6 over 4, 4-to-4 over 6	Tunnel	Between ES and ND, two NDs or two ESs
6to4	6-to-6 over 4	Tunnel	Between two NDs

Dual stack

- [RFC 4213]
- A node installs both IPv4 and IPv6 stacks in parallel
  - **IPv4 applications use IPv4 stack**
  - **IPv6 applications use IPv6 stack**
- Flow decisions are based on
  - **IP header version field, for receiving**
  - **destination address type, for sending**
- Address types typically come from DNS lookups (maintains both IPv4 and IPv6 addresses)



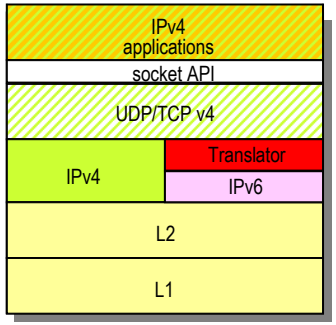
- Many OSs already provide dual IP stacks
- Dual stack only enables comunications between compatible nodes
  - **IPv6-to-IPv6 or IPv4-to-IPv4**
- Often used in combination with other mechanisms (e.g. tunnels)

Translation mechanisms

- Stateless IP/ICMP Translation (SIIT) [RFC 2765]
  - **basically consists on converting IP and ICMP packets**
  - **bidirectional translation between IPv4-IPv6 and ICMPv4-IVMPv6**
  - **ignores many IPv6 extension headers and IPv4 options**
  - **translation mechanism does not affect UDP and TCP pseudo-header checksums**
  - **Three SIIT mechanisms are BIS (Bump in the Stack), BIA (Bump in the API) , and NAT-PT (see later)**
- SIIT makes IPv6-only nodes interoperate with IPv4-only nodes
- Translation can occur in network nodes or in the end systems
- In case of end systems, translation may occur at different layers
  - **e.g. BIS (Bump in the Stack) or BIA (Bump in the API)**

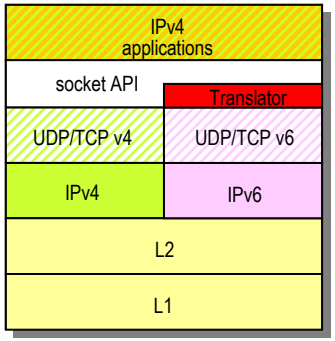
Translation mechanisms: Bump-in-the-Stack

- Bump-in-the-Stack (BIS) [RFC 2767]:
  - **comprises a TCP/IPv4 module and a translator module**
  - **packets from IPv4 applications flow into the TCP/IPv4 module**
  - **here, packets are translated into IPv6 and forwarded into the IPv6 module**
- Three bump components:
  - **extension name resolver**
    - snoops DNS lookups to decide whether the peer node is IPv6-only
  - **address mapper**
    - allocates a temporary IPv4 address for the IPv6 peer and caches the address mapping
  - **translator**
    - translates packets between IPv4 and IPv6
- Since temporary IPv4 address are not visible outside the node, it doesn't work with address-dependent applications



Translation mechanisms: Bump-in-the-API

- Bump-in-the-API (BIA) [RFC 3338]:
  - **a translator module inserted directly within the socket API**
  - **in such way it avoids translation of IP packets (allowing, for example, IP level security)**
  - **avoids modification of the OS kernel (IPv4/IPv6 stacks)**
- Three bump components:
  - **name resolver**
    - similar to BIS
  - **address mapper**
    - similar to BIS
  - **function mapper**
    - intercepts IPv4 socket function
    - calls and translates them to the equivalent IPv6 socket calls
- As with BIS, BIA doesn't work with address-dependent applications

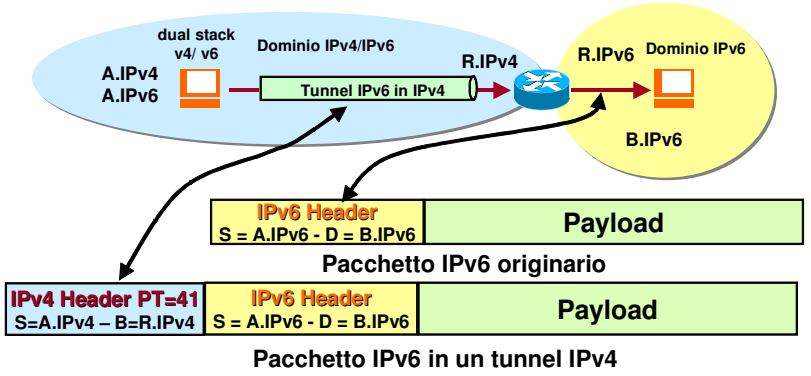


Translation mechanisms: NAT-PT

- Network Address Translation-Protocol Translation (NAT-PT) [RFC2766]
  - **SIIT IPv4/IPv6 translator**
  - **implemented in the network**
    - acts as a communication proxy towards IPv4 peers
    - avoiding increased complexity in end systems (scalable solution for end systems)
    - due to the heavy process load, it is suitable at network edge
  - **serves multiple IPv6 nodes allocating a temporary IPv4 address to each**
  - **allocation is triggered either**
    - by the first outbound IPv6 packet (using an IPv4-compatible IPv6 des-tination address) or
    - by the inbound IPv4 DNS lookup (from the peer) arriving at a co-located Application Level Gateway (ALG)

Tunneling di IPv6 su IPv4

- Il pacchetto IPv6 è inserito in un pacchetto IPv4 con campo *protocol = 41*



Tunneling mechanisms: "6over4"

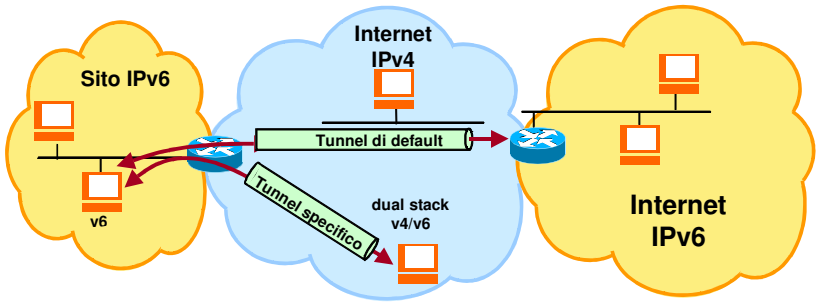
- 6over4 [RFC 2529]
  - **an IPv4 multicast domain (network) is view as IPv6 link (local network)**
  - **6over4 embeds IPv4 addresses in the IPv6 address link layer identifier part (i.e., last 64 bits) and**
  - **it defines Neighbor Discovery (ND) over IPv4 by using organization-local multicast**
- In 6over4, IPv4 network behaves as a virtual LAN
  - **a sender resolves the IPv6 target address (i.e., that of the offlink router or isolated end system) on the virtual LAN via ND**
  - **the resulting address bears the destination tunnel endpoint's IPv4 address**
- 6over4 maintains all of the features of IPv6
  - **including end-to-end security and stateless auto-configuration**
  - **supports multicast by defining a mapping between IPv6 multicast addresses and IPv4 organization-local multicast addresses**

Tunneling mechanisms: Configured Tunneling

- Configured IP-in-IP Tunneling
  - **nodes within the network are statically configured to perform tunneling**
  - **tunneling parameters are managed either**
    - through manual data entry or
    - via some automated service provided by a tunnel broker
- Tunnel brokers alleviate the management effort required
  - **their services are generally provided through Web-based applications**

Tunneling Statico

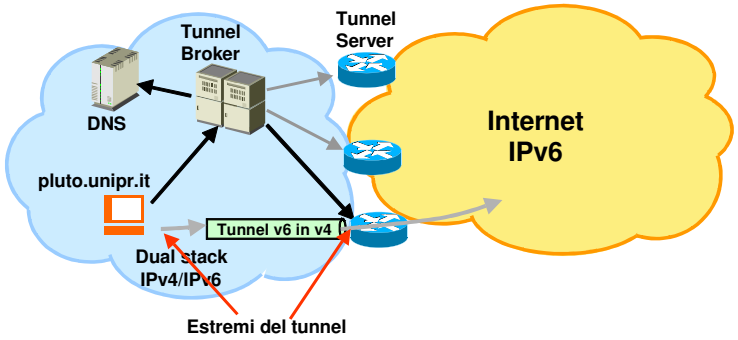
- Gli estremi del tunnel sono definiti attaverso configurazione



- Un router di bordo di un sito IPv6 può avere configurato un *Tunnel di Default* che gli permette di raggiungere il resto della rete IPv6
- Si possono configurare altri tunnel per raggiungere destinazioni specifiche
- *Grosso onere di gestione per gli amministratori*

Tunnel Broker

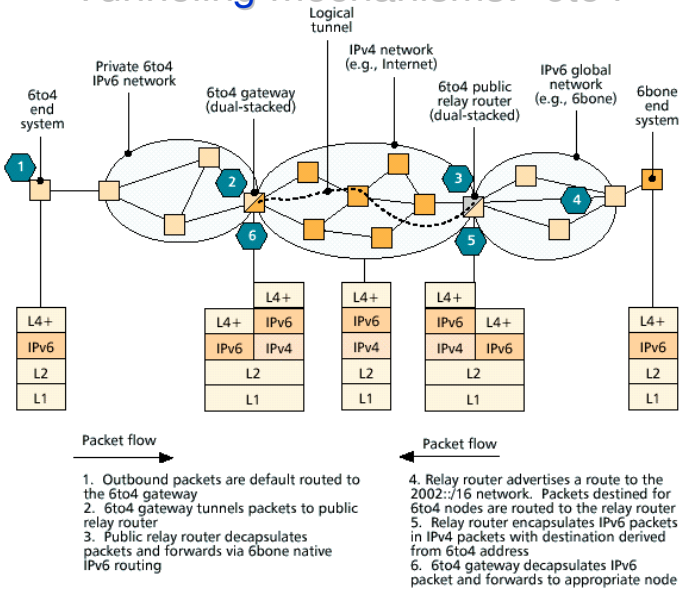
- I Tunnel Broker (TB) [RFC 3053] come ISP IPv6 virtuali
- La lista dei TB disponibili pubblicizzata su "well known" web page
- L'utente sceglie un TB (il più vicino, economico ..)
- L'utente si registra sul TB che gestisce la creazione del tunnel
- Meccanismo utile soprattutto in caso di host isolati



Tunneling mechanisms: "6to4"

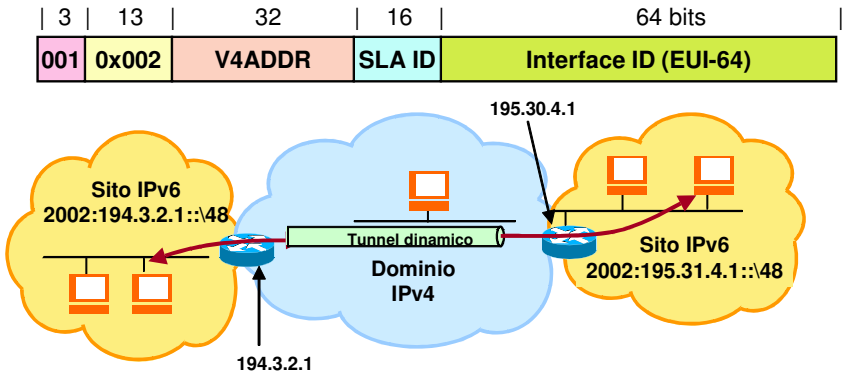
- 6to4 Automatic Tunneling [RFC 3056]
  - **the most widely used automatic tunneling technique**
  - **it tunnels IPv6 traffic over IPv4 networks among isolated 6to4 net-works**
  - **tunnel configuration is performed without the need for explicit management**
- Each 6to4 network
  - **is connected to the rest of the IPv6 network through a local 6to4 gateway and a remote relay router (both are dual-stacked)**
  - **assumes a special prefix that embeds the IPv4 address of its 6to4 gateway (2002:V4ADDR::/48)**
  - **tunnel endpoint addresses are easily obtained and do not need involvement of any IPv6 administrative body**
  - **Aal IPv6 packets, except for those destined to local addresses, are directed to the gateway**
  - **traffic in the reverse direction, destined for the 6to4 network**
    - is first forwarded to a nearby relay router (advertising the 2002::/16 prefix)
    - this then tunnels the traffic to the appropriate 6to4 gateway using the embedded IPv4 address

Tunneling mechanisms: "6to4"



Tunneling mechanisms: "6to4"

- I siti IPv6 che supportano 6to4 hanno *prefissi derivati dagli indirizzi IPv4* dei router 6to4



Vantaggi di "6to4"

- I siti che vogliono iniziare ad impiegare IPv6 non hanno bisogno di chiedere indirizzi IPv6 ai registri regionali
- Molto utile in assenza di ISP IPv6 perché non richiede la configurazione manuale dei tunnel