1. Introduzione al Livello 3

Il livello di rete (layer 3) è responsabile del routing dei pacchetti dalla sorgente alla destinazione, attraversando potenzialmente diverse reti intermedie.

1.1 Funzioni principali

- **Indirizzamento logico** (IP)
- **Routing**: scelta del percorso ottimale tra reti diverse
- **Instradamento** dei pacchetti
- **Frammentazione e riassemblaggio** dei pacchetti
- **Interconnessione** di reti eterogenee
- **Controllo della congestione** della rete

1.2 Posizionamento nel modello ISO/OSI

- Si trova tra il livello di collegamento dati (2) e il livello di trasporto (4)
- È indipendente dalla tecnologia di trasmissione sottostante
- Si occupa della consegna end-to-end di pacchetti attraverso diverse reti

2. Tipi di Routing

2.1 Routing statico

- Le tabelle di routing sono configurate manualmente dall'amministratore
- Nessun adattamento automatico ai cambiamenti della rete

2.1.1 Vantaggi

- Overhead ridotto (nessuno scambio di informazioni di routing)
- Maggiore sicurezza (percorsi controllati)
- Prevedibilità dei percorsi
- Nessun consumo di banda per aggiornamenti di routing

2.1.2 Svantaggi

- Nessuna tolleranza ai guasti automatica
- Richiede riconfigurazioni manuali in caso di cambiamenti
- Non scalabile per reti grandi o dinamiche
- Difficile da gestire in ambienti complessi

2.1.3 Utilizzo

- Reti piccole con topologia semplice
- Connessioni stabili e prevedibili
- Link di backup o percorsi alternativi fissi
- Reti con requisiti di sicurezza elevati

2.1.4 Esempio di configurazione in un router Cisco

```
Router(config)# ip route 192.168.2.0 255.255.255.0 192.168.1.2
```

2.2 Routing dinamico

- Le tabelle di routing vengono create e aggiornate automaticamente
- Adattamento ai cambiamenti della topologia di rete

2.2.1 Vantaggi

- Resilienza ai guasti (ricerca automatica di percorsi alternativi)
- Scalabilità (gestione di reti complesse)
- Adattabilità ai cambiamenti
- Riduzione della gestione manuale

2.2.2 Svantaggi

- Overhead di calcolo e di traffico
- Tempo di convergenza (ritardo nell'adattarsi ai cambiamenti)
- Possibili loop di routing
- Maggiore complessità di configurazione iniziale

2.2.3 Due approcci principali

- **Distance Vector**: basato sulla distanza verso le destinazioni
- **Link State**: basato su una mappa completa della rete

2.3 Confronto tra routing statico e dinamico

Caratteristica	Routing Statico	Routing Dinamico
Configurazione	Manuale	Automatica
Adattabilità ai cambiamenti	Nessuna	Alta
Overhead di rete	Nessuno	Presente
Risorse richieste	Basse	Medio-alte

Caratteristica	Routing Statico	Routing Dinamico
Scalabilità	Bassa	Alta
Complessità di configurazione	Bassa per reti piccole, alta per reti grandi	Media, indipendente dalla dimensione
Tempo di convergenza	Non applicabile	Variabile in base al protocollo

3. Algoritmi di Routing

3.1 Distance Vector (Bellman-Ford)

- Ogni router condivide la propria tabella di routing con i vicini diretti
- Calcolo basato sulla "distanza" (numero di hop o costo) verso ogni destinazione
- Memorizza solo il vettore distanza verso destinazioni note
- Rappresentazione dati: vettore = [destinazione, distanza, next-hop]
- Aggiornamento periodico delle tabelle

3.1.1 Vantaggi

- Implementazione semplice
- Basso overhead computazionale
- Adatto a reti di piccole-medie dimensioni

3.1.2 Svantaggi

- Convergenza lenta
- Problema del "count-to-infinity"
- Decisioni basate su informazioni indirette

3.1.3 Protocolli che lo implementano

- RIP (Routing Information Protocol)
- BGP (Border Gateway Protocol, variante path vector)
- EIGRP (Enhanced Interior Gateway Routing Protocol, ibrido)

3.1.4 Problema del Count-to-infinity

- Quando un link fallisce, le informazioni errate possono propagarsi
- I router continuano ad incrementare la metrica fino a raggiungere l'infinito
- Soluzioni parziali:
  - **Split horizon**: non annunciare le rotte apprese da un vicino indietro allo stesso vicino

- **Poisoned reverse**: annunciare rotte apprese da un vicino indietro allo stesso vicino ma con metrica infinita
- **Trigger update**: inviare aggiornamenti immediati in caso di cambiamenti
- **Definizione di infinito**: stabilire un valore massimo come "infinito" (es. 16 in RIP)

3.2 Link State (Dijkstra)

- Ogni router crea una mappa dell'intera rete
- Invia informazioni sullo stato dei propri link a tutti i router della rete
- Calcola il percorso più breve usando l'algoritmo di Dijkstra
- Ogni router esegue il calcolo in modo indipendente

3.2.1 Vantaggi

- Convergenza rapida
- Maggiore affidabilità
- Meno soggetto a loop
- Conoscenza completa della topologia

3.2.2 Svantaggi

- Richiede più memoria e potenza di calcolo
- Overhead maggiore in fase iniziale
- Complessità di implementazione
- Maggiore traffico per flooding iniziale

3.2.3 Protocolli che lo implementano

- OSPF (Open Shortest Path First)
- IS-IS (Intermediate System to Intermediate System)

3.2.4 Algoritmo di Dijkstra (pseudocodice)

```
function Dijkstra(Graph, source):
    // Inizializzazione
    for each vertex v in Graph:
        dist[v] := infinity
        prev[v] := undefined
        add v to Q
    dist[source] := 0

    // Algoritmo
    while Q is not empty:
```

```
        u := vertex in Q with min dist[u]
        remove u from Q

        for each neighbor v of u:
            alt := dist[u] + length(u, v)
            if alt < dist[v]:
                dist[v] := alt
                prev[v] := u

    return dist[], prev[]
```

3.2.5 Esempio di tabella di routing

```
Destinazione | Next Hop | Metrica | Interfaccia
-------------|----------|---------|------------
192.168.1.0  | -        | 0       | eth0
10.0.0.0     | 192.168.1.1 | 1    | eth0
172.16.0.0   | 192.168.1.254 | 2  | eth0
0.0.0.0      | 192.168.1.1 | 1    | eth0
```

3.3 BGP (Border Gateway Protocol)

- Utilizzato principalmente tra sistemi autonomi (AS) diversi
- Protocollo path vector (evoluzione del distance vector)
- Tiene traccia del percorso completo verso ogni destinazione
- Permette policy di instradamento basate su accordi economici e politiche

3.3.1 Caratteristiche principali

- Protocollo di routing esterno (EGP)
- Basato su TCP (porta 179)
- Convergenza lenta ma stabile
- Distribuzione di route filtrabili in base a policy
- Distingue tra eBGP (tra AS diversi) e iBGP (all'interno dello stesso AS)

3.3.2 Attributi principali delle route

- **AS_PATH**: lista di AS attraversati per raggiungere la destinazione
- **NEXT_HOP**: indirizzo IP del router di confine
- **LOCAL_PREF**: preferenza locale (usata nell'iBGP)
- **MED** (Multi-Exit Discriminator): suggerisce punto di ingresso preferito

3.4 Protocolli di controllo

3.4.1 ICMP (Internet Control Message Protocol)

- Protocollo di supporto per IP, non usato per trasporto di dati applicativi
- Funzioni principali:
  - Segnalazione errori (es. host irraggiungibile)
  - Diagnostica (es. ping, traceroute)
  - Controllo di flusso

3.4.2 Struttura pacchetto ICMP

- **Type**: tipo di messaggio (es. 0: Echo Reply, 8: Echo Request)
- **Code**: sottotipo del messaggio
- **Checksum**: verifica integrità
- **Data**: dipende dal tipo di messaggio

3.4.3 Tipi di messaggi ICMP comuni

Type	Code	Significato
0	0	Echo Reply (risposta al ping)
3	0-15	Destination Unreachable
8	0	Echo Request (ping)
11	0-1	Time Exceeded (usato da traceroute)
5	0-3	Redirect (cambia next-hop)

3.4.4 Comandi che utilizzano ICMP

- **ping**: verifica connettività verso un host

```
ping 192.168.1.1
```

- **traceroute** (Linux/macOS) o **tracert** (Windows): determina il percorso verso un host

```
traceroute google.com
```

4. Algoritmi di Congestione

Gli algoritmi di congestione sono usati per prevenire e gestire la congestione nelle reti.

4.1 Choke Packet

- Pacchetti di notifica generati dai router congestionati
- Inviati alle sorgenti per ridurre il tasso di trasmissione
- Simile al messaggio ICMP Source Quench (ora deprecato)
- Funzionamento:
  1. Il router rileva congestione (buffer che si riempiono)
  2. Genera choke packet verso le sorgenti
  3. Le sorgenti riducono il tasso di trasmissione
  4. La congestione diminuisce

4.2 Leaky Bucket

- Algoritmo che regola la velocità con cui i pacchetti vengono inoltrati
- Funzionamento:
  1. I pacchetti entrano nel "secchio" (buffer)
  2. Escono a velocità costante
  3. Se il secchio è pieno, i nuovi pacchetti vengono scartati
- Analogia: secchio con un foro sul fondo, acqua (pacchetti) esce a velocità costante
- Livella i burst di traffico in un flusso regolare

4.2.1 Vantaggi

- Implementazione semplice
- Garantisce un traffico regolare
- Previene sovraccarichi downstream

4.2.2 Svantaggi

- Limitazione della velocità anche con rete non congestionata
- Possibile perdita di pacchetti
- Non adattivo alle condizioni della rete

4.3 Token Bucket

- Più flessibile rispetto al Leaky Bucket
- Funzionamento:
  1. Token vengono generati a velocità costante
  2. Ogni pacchetto richiede un token per essere trasmesso
  3. Se non ci sono token, il pacchetto attende
  4. I token possono accumularsi fino a un massimo
- Permette burst controllati di traffico

4.3.1 Vantaggi

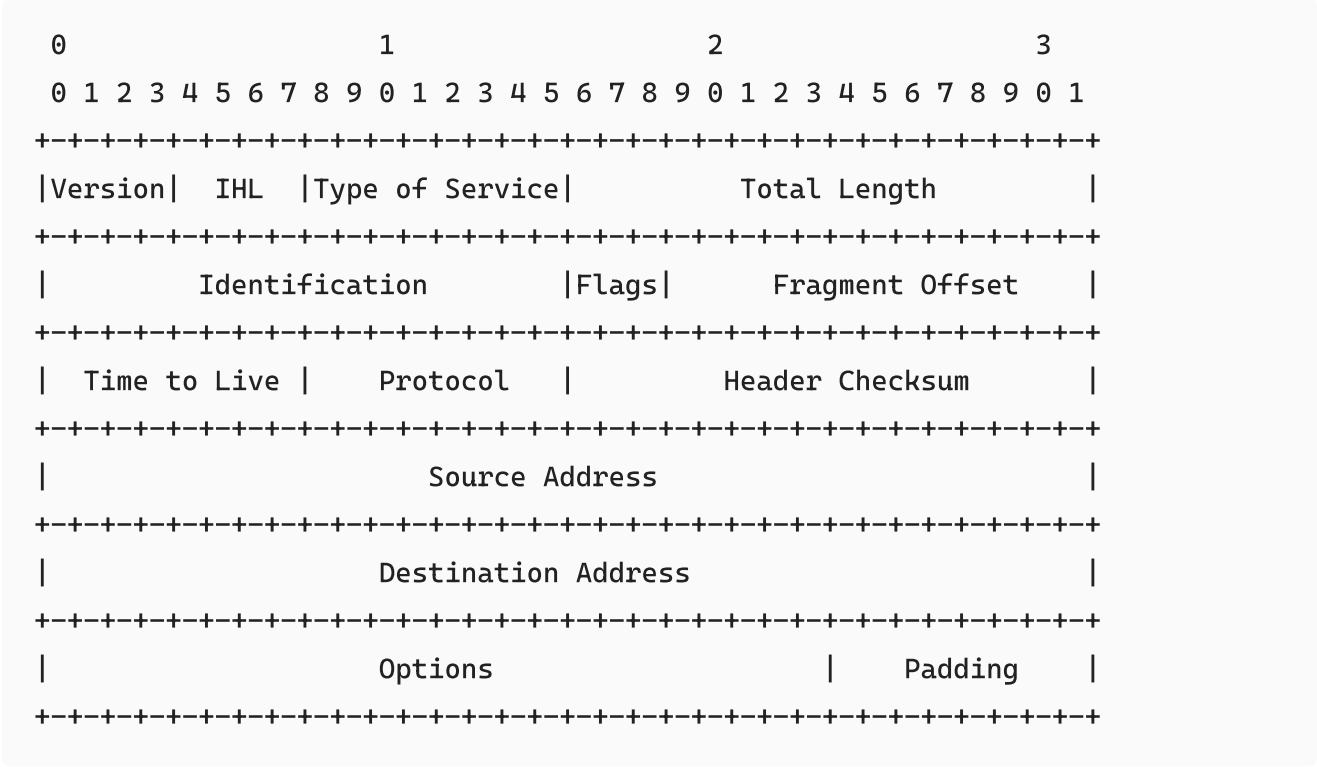
- Permette picchi temporanei di traffico
- Efficiente in termini di utilizzo della rete
- Adatto a traffico a raffica

4.3.2 Svantaggi

- Più complesso da implementare
- Richiede parametrizzazione attenta
- Possibile inattività in presenza di token inutilizzati

5. Livello IP (Internet Protocol)

5.1 Struttura del pacchetto IPv4



5.1.1 Campi principali

- **Version**: Versione del protocollo IP (4 per IPv4)
- **IHL** (Internet Header Length): Lunghezza dell'header in parole da 32 bit
- **Type of Service**: Priorità del pacchetto (oggi DSCP e ECN)
- **Total Length**: Lunghezza totale del pacchetto (header + dati)
- **Identification**: Identificatore per i frammenti appartenenti allo stesso datagramma
- **Flags**: Controllo frammentazione
  - DF: Don't Fragment
  - MF: More Fragments

- **Fragment Offset**: Posizione del frammento nel datagramma originale
- **Time to Live (TTL)**: Numero massimo di hop prima di scartare il pacchetto
- **Protocol**: Protocollo di livello superiore

5.2 Classi di indirizzi IP

Gli indirizzi IPv4 sono divisi in classi basate sui primi bit dell'indirizzo:

5.2.1 Classe A

- **Primo bit**: 0
- **Range**: 0.0.0.0 - 127.255.255.255
- **Maschera di rete**: 255.0.0.0 (/8)
- **Formato**: N.H.H.H (N=Network, H=Host)
- **Host per rete**: 2^24 - 2 = 16,777,214
- **Utilizzo**: Grandi organizzazioni e enti governativi

5.2.2 Classe B

- **Primi bit**: 10
- **Range**: 128.0.0.0 - 191.255.255.255
- **Maschera di rete**: 255.255.0.0 (/16)
- **Formato**: N.N.H.H
- **Host per rete**: 2^16 - 2 = 65,534
- **Utilizzo**: Medie e grandi imprese

5.2.3 Classe C

- **Primi bit**: 110
- **Range**: 192.0.0.0 - 223.255.255.255
- **Maschera di rete**: 255.255.255.0 (/24)
- **Formato**: N.N.N.H
- **Host per rete**: 2^8 - 2 = 254
- **Utilizzo**: Piccole reti aziendali e LAN

5.2.4 Classe D (Multicast)

- **Primi bit**: 1110
- **Range**: 224.0.0.0 - 239.255.255.255
- **Non utilizzata per indirizzare host**
- **Uso**: Trasmissioni multicast (da uno a molti)
- **Esempi**: Video conferenza, streaming, protocolli di routing

5.2.5 Classe E (Riservata/Sperimentale)

- **Primi bit**: 1111
- **Range**: 240.0.0.0 - 255.255.255.255
- **Uso**: Riservata per utilizzo futuro e sperimentazione
- **Non utilizzabile in Internet**

5.3 Indirizzi speciali

- **Loopback**: 127.0.0.0/8 (in particolare 127.0.0.1)
  - Utilizzato per test e comunicazione interna alla macchina
  - Qualsiasi pacchetto inviato a un indirizzo in questo range viene reindirizzato a localhost
- **Broadcast locale**: xxx.xxx.xxx.255
  - Broadcast limitato alla rete locale
  - Esempio: 192.168.1.255 in una rete 192.168.1.0/24
- **Broadcast di rete**: tutti i bit host a 1
  - Inviato a tutti gli host di una specifica rete
- **Indirizzo di rete**: tutti i bit host a 0
  - Identifica la rete stessa, non un host
  - Esempio: 192.168.1.0 in una rete 192.168.1.0/24
- **Indirizzi privati** (non instradabili su Internet):
  - 10.0.0.0/8 (Classe A)
  - 172.16.0.0/12 (Classe B)
  - 192.168.0.0/16 (Classe C)
  - Utilizzati nelle reti locali e tradotti tramite NAT
- **APIPA**: 169.254.0.0/16
  - Automatic Private IP Addressing
  - Assegnazione automatica quando DHCP fallisce
  - Utilizzato in Windows e altri sistemi
- **Multicast speciali**:
  - 224.0.0.1: tutti gli host del segmento
  - 224.0.0.2: tutti i router del segmento
  - 224.0.0.5: tutti i router OSPF
  - 224.0.0.251: mDNS (Multicast DNS)

5.4 Subnetting e CIDR

5.4.1 Subnet Mask

- Maschera binaria che identifica la parte di rete e di host di un indirizzo

- Esempi di subnet mask comuni:
  - 255.0.0.0 = /8 (Classe A)
  - 255.255.0.0 = /16 (Classe B)
  - 255.255.255.0 = /24 (Classe C)
  - 255.255.255.240 = /28 (16 indirizzi per subnet)

5.4.2 CIDR (Classless Inter-Domain Routing)

- Supera il concetto di classi di indirizzi rigide
- Notazione: indirizzo/prefisso
  - Esempio: 192.168.1.0/24
- Vantaggi:
  - Utilizzo efficiente dello spazio di indirizzi
  - Flessibilità nella suddivisione delle reti
  - Consente aggregazione di rotte (supernetting)
- Ha contribuito a ritardare l'esaurimento degli indirizzi IPv4

5.4.3 Calcolo del subnetting

1. Determinare quante subnet sono necessarie
2. Determinare quanti host per subnet sono richiesti
3. Calcolare il numero di bit da prendere in prestito dagli host
4. Calcolare la nuova subnet mask
5. Calcolare gli indirizzi di rete, broadcast e il range di host per ogni subnet

Esempio:

Suddividere 192.168.1.0/24 in 4 subnet

1. Per 4 subnet servono 2 bit (2^2 = 4)
2. La nuova subnet mask è /26 (24+2)
3. Le subnet risultanti sono:
   - 192.168.1.0/26 (host: 192.168.1.1 - 192.168.1.62, broadcast: 192.168.1.63)
   - 192.168.1.64/26 (host: 192.168.1.65 - 192.168.1.126, broadcast: 192.168.1.127)
   - 192.168.1.128/26 (host: 192.168.1.129 - 192.168.1.190, broadcast: 192.168.1.191)
   - 192.168.1.192/26 (host: 192.168.1.193 - 192.168.1.254, broadcast: 192.168.1.255)

5.4.4 VLSM (Variable Length Subnet Mask)

- Permette di utilizzare subnet mask di lunghezza variabile all'interno della stessa rete
- Vantaggi:
  - Utilizza lo spazio di indirizzi in modo più efficiente
  - Adatta le dimensioni delle subnet alle esigenze specifiche

- Esempio di utilizzo:
  - Link punto-punto: richiedono solo 2 indirizzi (/30)
  - Piccoli uffici: poche decine di indirizzi (/27 o /28)
  - Grandi LAN: centinaia di indirizzi (/24 o più grande)
- Richiede protocolli di routing che supportano il CIDR (OSPF, EIGRP, BGP)

5.5 IPv6

IPv6 è la nuova versione del protocollo IP, progettata per sostituire IPv4 a causa dell'esaurimento degli indirizzi.

5.5.1 Caratteristiche principali

- **Spazio di indirizzamento**: indirizzi a 128 bit (2^128 indirizzi disponibili)
- **Header semplificato**: meno campi, più efficiente
- **Supporto integrato per sicurezza** (IPsec)
- **Nessuna frammentazione a livello di router**
- **Nessun checksum nell'header** (delegato ai livelli superiori)
- **Configurazione automatica** degli indirizzi (SLAAC)
- **Multicast** integrato e migliorato
- **Eliminazione del broadcast** (sostituito da multicast)
- **Introduzione di Anycast** (indirizzo condiviso tra più interfacce con routing verso la più vicina)

5.5.2 Formato degli indirizzi IPv6

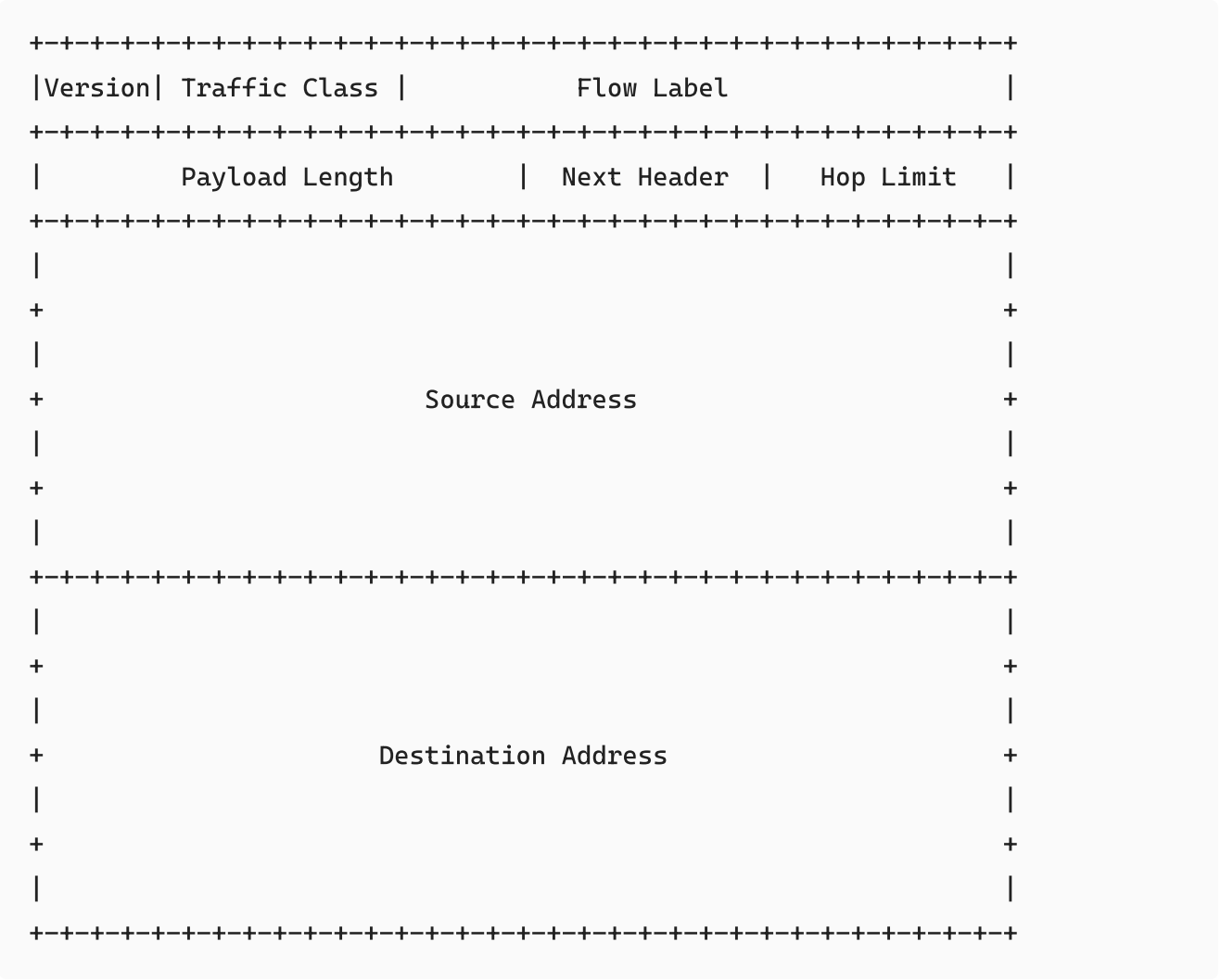
- 8 gruppi di 4 cifre esadecimali separati da ":"
- Esempio: 2001:0db8:85a3:0000:0000:8a2e:0370:7334
- Regole di semplificazione:
  - I gruppi di zeri possono essere omessi: 2001:0db8:85a3::8a2e:0370:7334
  - Gli zeri iniziali in ogni gruppo possono essere omessi: 2001:db8:85a3::8a2e:370:7334
  - Solo una sequenza di zeri può essere abbreviata con "::"

5.5.3 Tipi di indirizzi IPv6

- **Unicast**: identifica una singola interfaccia
  - **Global Unicast**: equivalenti agli indirizzi pubblici IPv4, iniziano con 2000::/3
  - **Link-Local**: validi solo nel link locale, iniziano con fe80::/10
  - **Unique Local**: equivalenti agli indirizzi privati IPv4, iniziano con fc00::/7
  - **Loopback**: ::1/128 (equivalente a 127.0.0.1 in IPv4)
- **Multicast**: identifica un gruppo di interfacce, iniziano con ff00::/8

- **Anycast**: identifica un gruppo di interfacce, ma il pacchetto viene inviato solo alla più vicina

5.5.4 Header IPv6



- **Version**: Sempre 6 per IPv6
- **Traffic Class**: Priorità del pacchetto (simile al Type of Service in IPv4)
- **Flow Label**: Etichetta per identificare pacchetti dello stesso flusso
- **Payload Length**: Lunghezza del payload (escluso header principale)
- **Next Header**: Tipo di header seguente (estensione o protocollo di livello superiore)
- **Hop Limit**: Equivalente al TTL in IPv4
- **Source Address**: Indirizzo IPv6 sorgente (128 bit)
- **Destination Address**: Indirizzo IPv6 destinazione (128 bit)

5.5.5 Extension Header

In IPv6, funzionalità aggiuntive sono implementate tramite header di estensione:

- **Hop-by-Hop Options**: opzioni per ogni nodo nel percorso
- **Routing**: routing source-routed
- **Fragment**: informazioni di frammentazione

- **Authentication (AH)**: integrità e autenticazione (IPsec)
- **Encapsulating Security Payload (ESP)**: cifratura (IPsec)
- **Destination Options**: opzioni solo per il nodo destinazione

5.5.6 Differenze principali tra IPv4 e IPv6

Caratteristica	IPv4	IPv6
Dimensione indirizzo	32 bit	128 bit
Notazione	Decimale puntata	Esadecimale con separatori ":"
Header	Variabile (20-60 byte)	Fisso (40 byte)
Checksum	Presente	Assente
Frammentazione	Router e host	Solo host
ARP	Richiesto	Sostituito da NDP
Configurazione	Manuale o DHCP	SLAAC, DHCPv6 o manuale
Broadcast	Supportato	Non supportato (usa multicast)
IPsec	Opzionale	Integrato
NAT	Comune	Generalmente non necessario

5.5.7 Transizione da IPv4 a IPv6

Tecniche per la coesistenza e migrazione:

- **Dual Stack**: supporto simultaneo di IPv4 e IPv6
  - Entrambi i protocolli attivi sulla stessa interfaccia
  - Il sistema operativo sceglie quale usare
- **Tunneling**: incapsulamento di pacchetti IPv6 in pacchetti IPv4
  - **6to4**: automatico, usa prefisso 2002::/16
  - **6in4**: tunnel configurato manualmente
  - **Teredo**: attraversa NAT, usa prefisso 2001::/32
- **Translation**: traduzione diretta tra pacchetti IPv4 e IPv6
  - **NAT64/DNS64**: permette a client IPv6 di comunicare con server IPv4
  - **464XLAT**: combinazione di traduzione locale e centralizzata

5.6 Protocolli ausiliari di livello 3

5.6.1 ARP (Address Resolution Protocol)

- **Funzione**: Risolve un indirizzo IP in un indirizzo MAC
- **Funzionamento**:
  1. Il mittente invia un broadcast ARP ("Chi ha questo IP?")

2. Il destinatario risponde ("Io ho questo IP, questo è il mio MAC")
3. Il mittente memorizza l'associazione IP-MAC nella sua cache ARP

- **Struttura pacchetto ARP**:
  - Hardware Type (Ethernet = 1)
  - Protocol Type (IPv4 = 0x0800)
  - Hardware Address Length (6 per MAC)
  - Protocol Address Length (4 per IPv4)
  - Operation (1 = request, 2 = reply)
  - Sender Hardware Address (MAC mittente)
  - Sender Protocol Address (IP mittente)
  - Target Hardware Address (MAC destinatario)
  - Target Protocol Address (IP destinatario)

5.6.2 DHCP (Dynamic Host Configuration Protocol)

- **Funzione**: Assegna automaticamente indirizzi IP e altre configurazioni
- **Processo in 4 fasi**:
  1. **DISCOVER**: client broadcast per trovare server DHCP
  2. **OFFER**: server offre un indirizzo IP
  3. **REQUEST**: client richiede l'indirizzo offerto
  4. **ACK**: server conferma l'assegnazione
- **Informazioni fornite**:
  - Indirizzo IP
  - Subnet mask
  - Gateway predefinito
  - Server DNS
  - Lease time (tempo di validità dell'assegnazione)
- **Vantaggi**:
  - Configurazione automatica
  - Gestione centralizzata
  - Prevenzione di conflitti di indirizzi
  - Recupero di indirizzi non più utilizzati

5.6.3 NAT (Network Address Translation)

- **Funzione**: Permette a una rete privata di condividere un singolo indirizzo IP pubblico
- **Funzionalità**:
  - Conservazione indirizzi IP pubblici
  - Sicurezza (nasconde struttura interna)
  - Facilita cambio di ISP (solo indirizzi pubblici cambiano)
- **Tipi**:

- **Static NAT**: mappatura 1:1 tra IP privati e pubblici
- **Dynamic NAT**: pool di indirizzi pubblici assegnati dinamicamente
- **PAT/NAPT**: mappatura molti:1 usando diverse porte (più comune)

PAT (Port Address Translation)

- Variante del NAT più diffusa
- Usa porte TCP/UDP per mappare più host interni su un singolo IP pubblico
- Funzionamento:
  1. Host interno invia pacchetto verso Internet
  2. Router NAT memorizza informazioni di sessione (IP+porta sorgente, IP+porta destinazione)
  3. Sostituisce IP+porta sorgente con IP pubblico + porta unica
  4. Al ritorno, inverte la traduzione usando la tabella di stato
- Vantaggi:
  - Efficienza nell'uso di indirizzi pubblici
  - Livello base di sicurezza (indirizzi interni nascosti)
  - Economico e facile da implementare
- Svantaggi:
  - Problemi con alcuni protocolli (che includono IP nei dati)
  - Difficoltà nell'hosting di servizi
  - Tracciabilità e logging complessi