RETI DI CALCOLATORI

Autunno 2018

docente: Laura Ricci

laura.ricci@unipi.it

Lezione 7:
IL LIVELLO IP:
FORMATO DEI PACCHETTI
ROUTING PROTOCOLS
18/10/2018

parte di queste slides sono
ricavati da slides pubblicate in corsi
universitari tenuti da colleghi
italiani e stranieri

MATERIALE DIDATTICO

Forouzan

- Addressing
  - paragrafo 4.1
  - paragrafo 4.1.1
  - Paragrafo 4.2.1
- Network protocols
  - parte su RIP ed OSPF nel capitolo 4

CONTROL E DATA PLANE

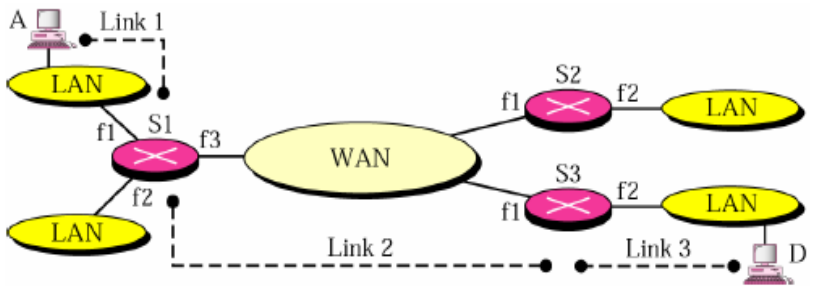
- nelle lezioni precedenti, ci siamo focalizzati soprattutto del control plane del livello IP
  - come i routers scoprono e selezionano i cammini tra due nodi utilizzando protocolli distribuiti
  - attivato ad ogni cambiamento della topologia (ad esempio viene inserito un nuovo router, un router si guasta, il costo di un link è modificato)

- nelle prossime lezioni:
  - routing protocols del control plane, basati sugli algoritmi visti in precedenza.
  - il data plane del livello IP
    - quale è il formato dei pacchetti, in particolare dell'IP header
    - come avviene l'inoltro dei pacchetti nei routers basato sulle informazioni raccolte dal control pane
    - opera su ogni pacchetto ricevuto da un router per determinare il suo inoltro su uno delle interfacce di uscia del router
      - necessario supporto che renda efficiente l'inoltro

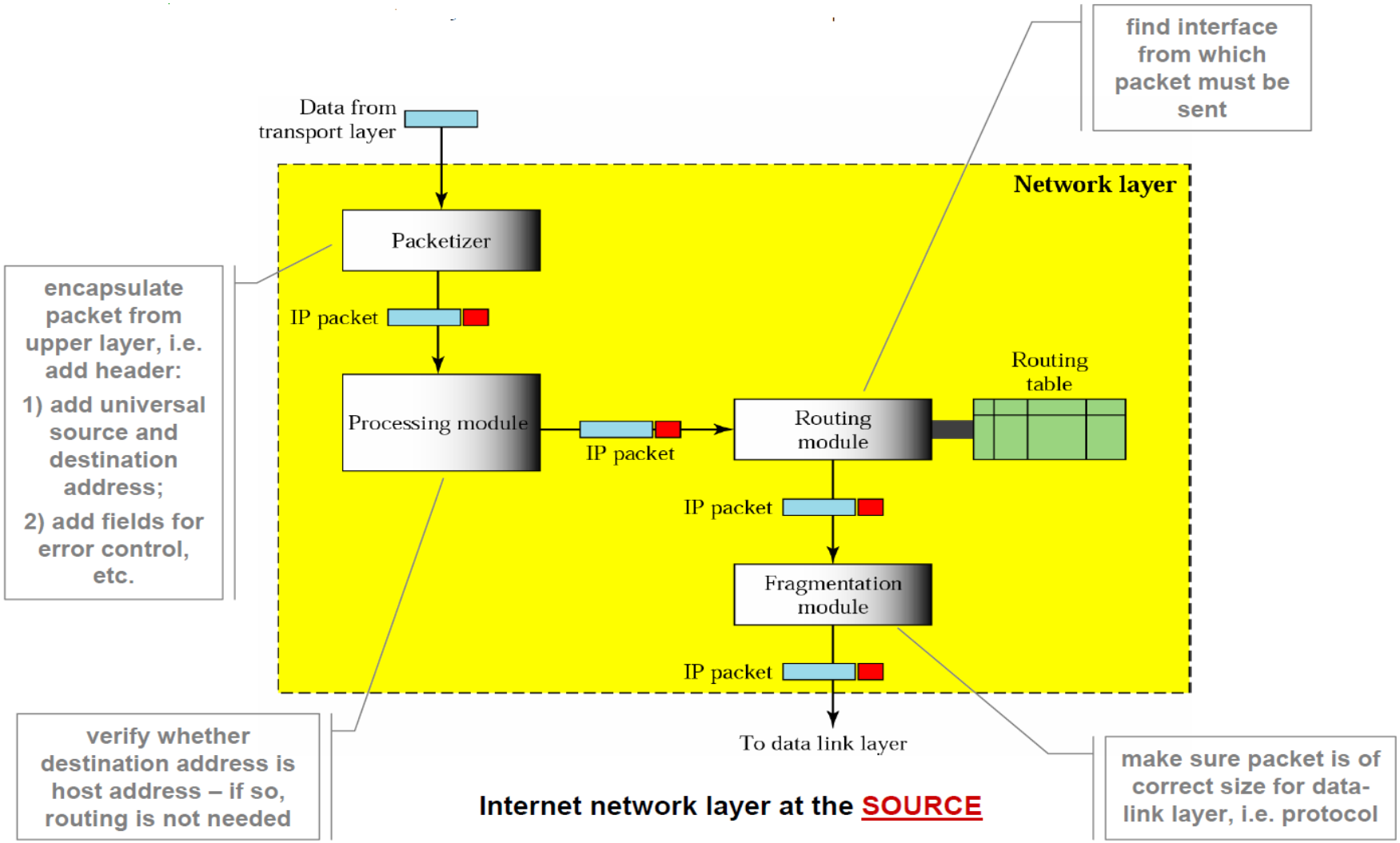
NETWORK LAYER(IP LAYER)

- supervisiona la consegna dei pacchetti host-to-host
  - due host possono essere separati da diverse reti fisiche
  - data link layer supervisione la consegna nodo a nodo, transport layer quella processo-processo

- funzioni più importanti:
  - addressing: identificare ogni device in modo univoco per permettere la comunicazione globale
  - routing: determinare il cammino migliore per spedire un pacchetto da un host ad un altro
  - packetizing: incapsulare i pacchetti ricevuti dal livello superiore
  - fragmenting: deincapsulare i pacchetti provenienti da una rete ed incapsularli per la rete successiva

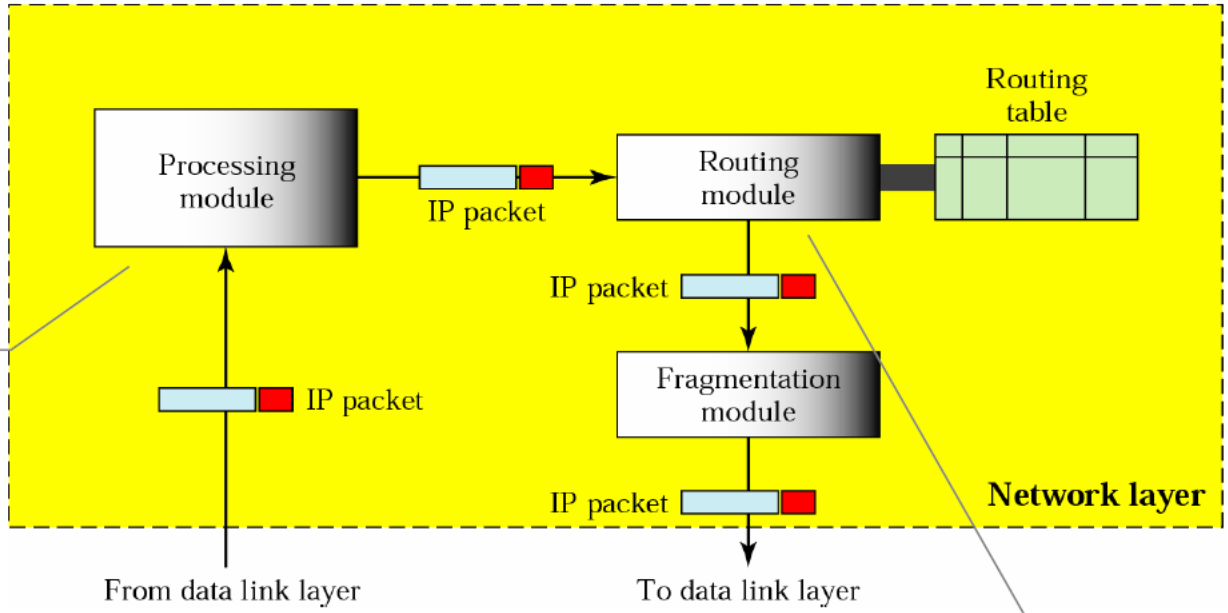


NETWORK LAYER: FUNZIONALITA'



Internet network layer at the SOURCE

NETWORK LAYER: FUNZIONALITA'



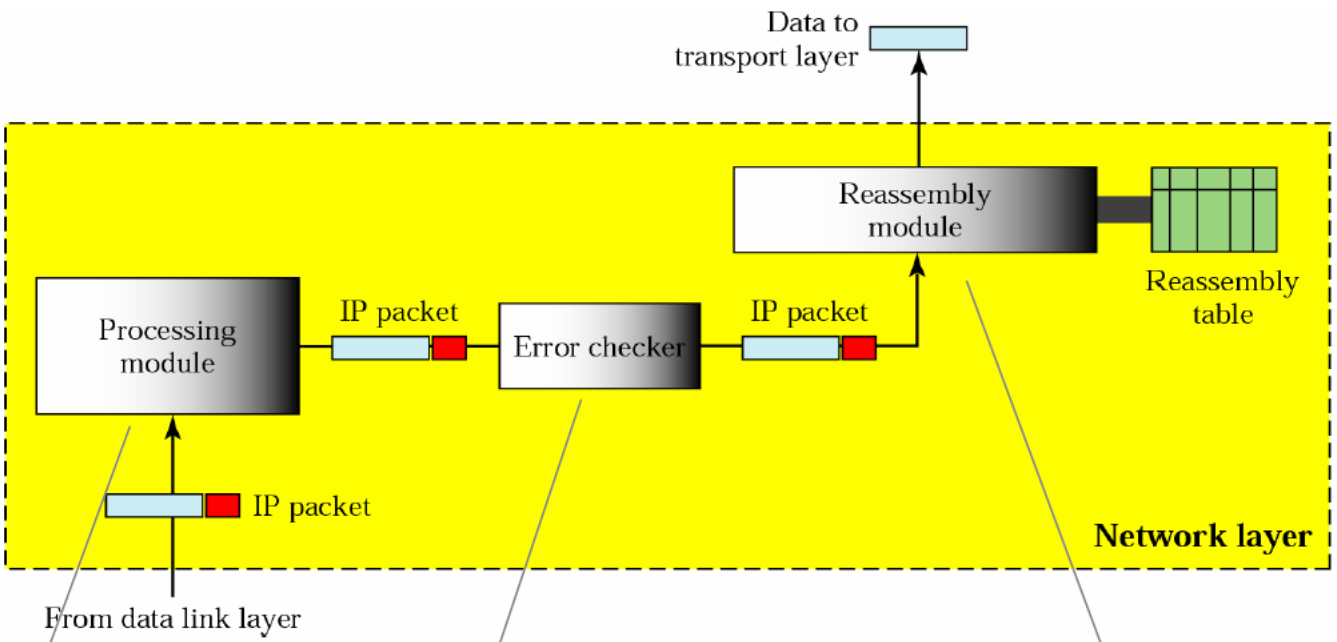
check if packet has reached its final destination or needs to be forwarded

+ header error checking !!!

find interface from which packet must be sent

Internet network layer at a ROUTER

NETWORK LAYER: FUNZIONALITA'



verify whether destination address is host address

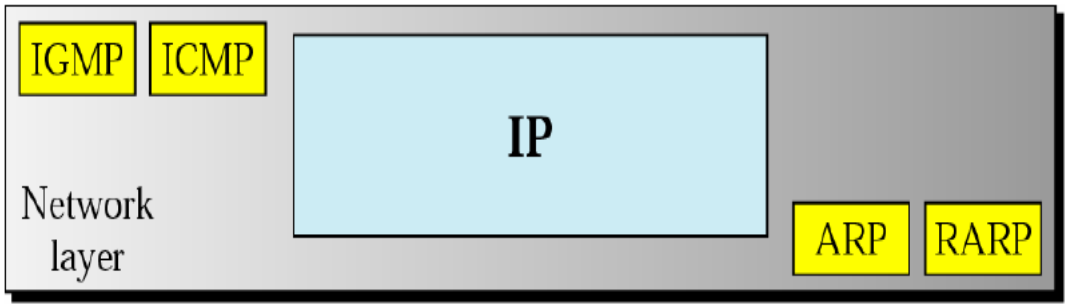
Internet network layer at the DESTINATION

check if packet has been corrupted during transmission

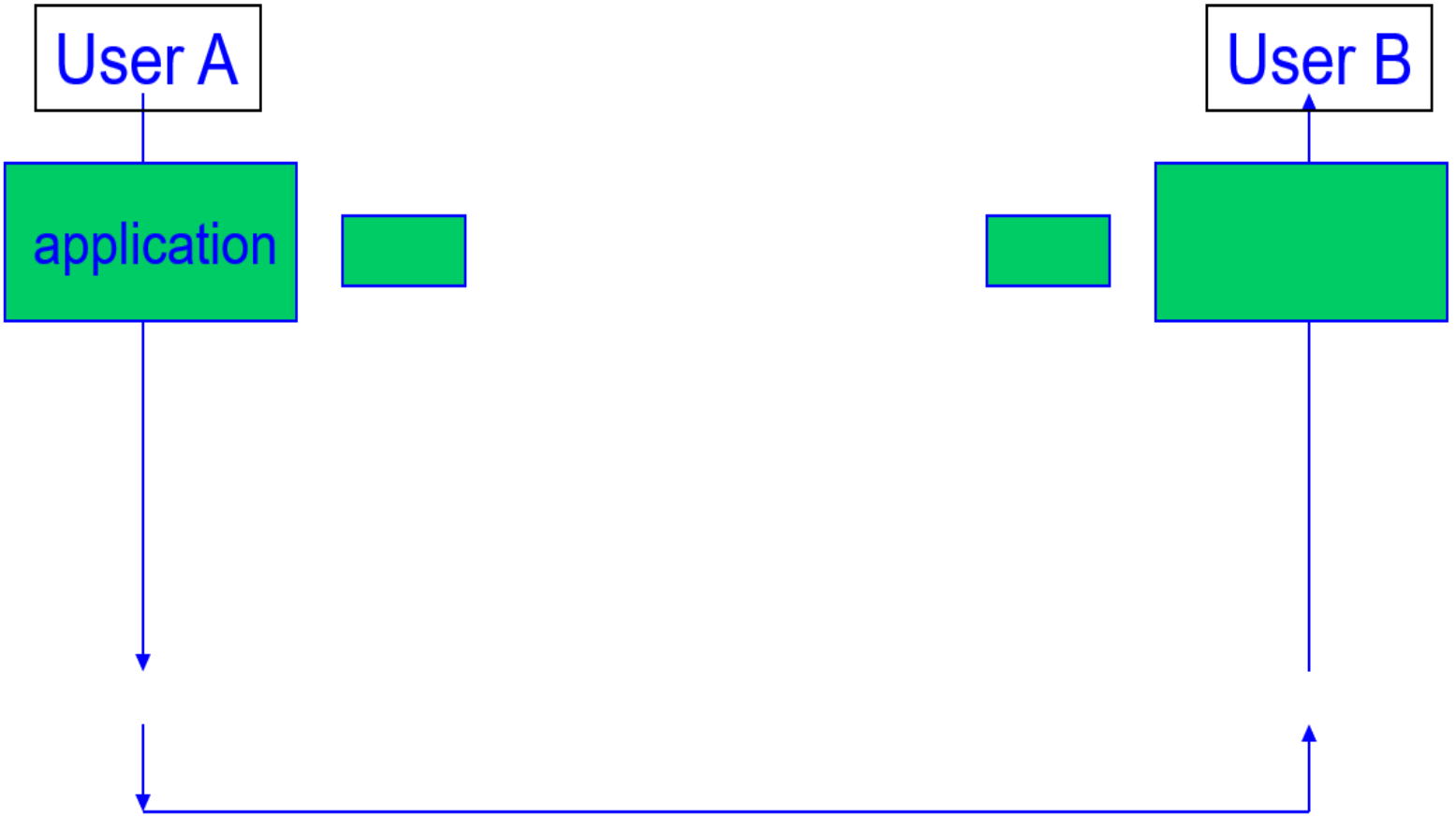
if packet has been fragmented, wait until all fragments have arrived, reassemble them, and then deliver the reassembled packet to transport layer

NETWORK LAYER PROTOCOLS

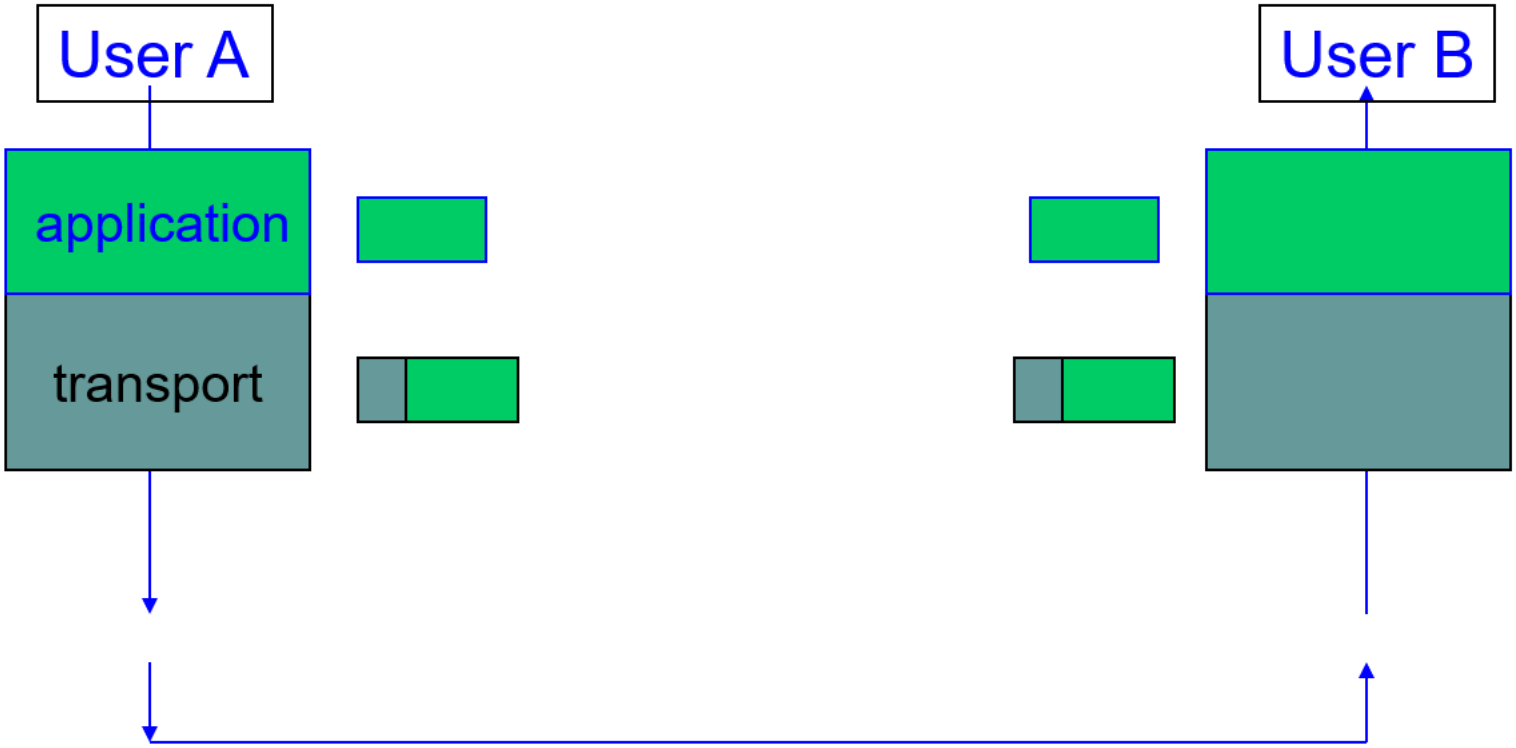
- IP: IP–main protocol, responsabile del delivery "best effort" host-to-host
- ARP: mappa l'indirizzo IP address del next hop nel suo indirizzo fisico (MAC) (usato quando il pacchetto viene passato al livello inferiore, il livello data-link)
- RARP: maps l'indirizzo fisico (MAC) nell'indirizzo IP
- ICMP: usato da host e routers per gestire situazioni di errore, come errori nell'header del pacchetto, host e reti non raggiungibili
- IGMP: usato da host e router per il network-layer multicasting



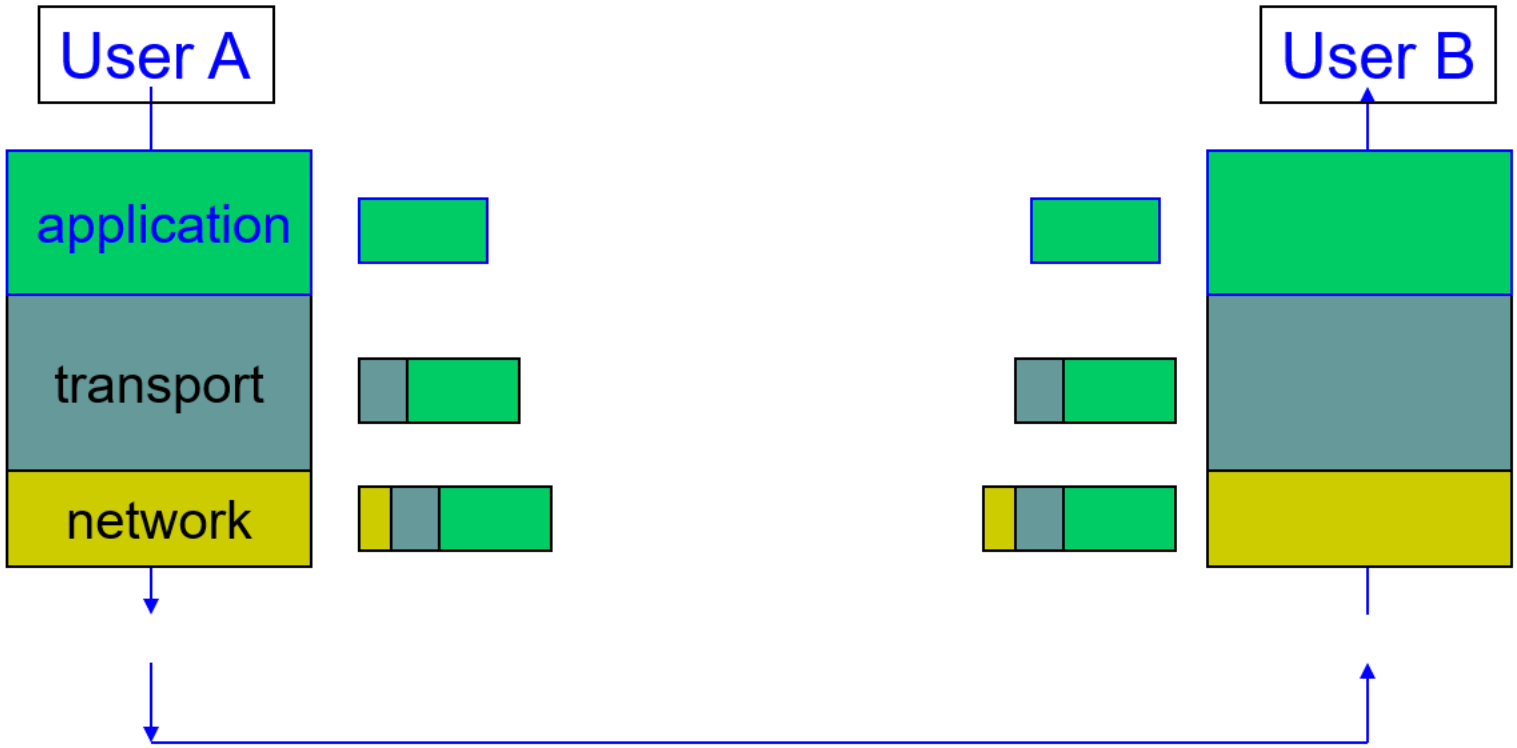
RICHIAMO: LAYER ENCAPSULATION



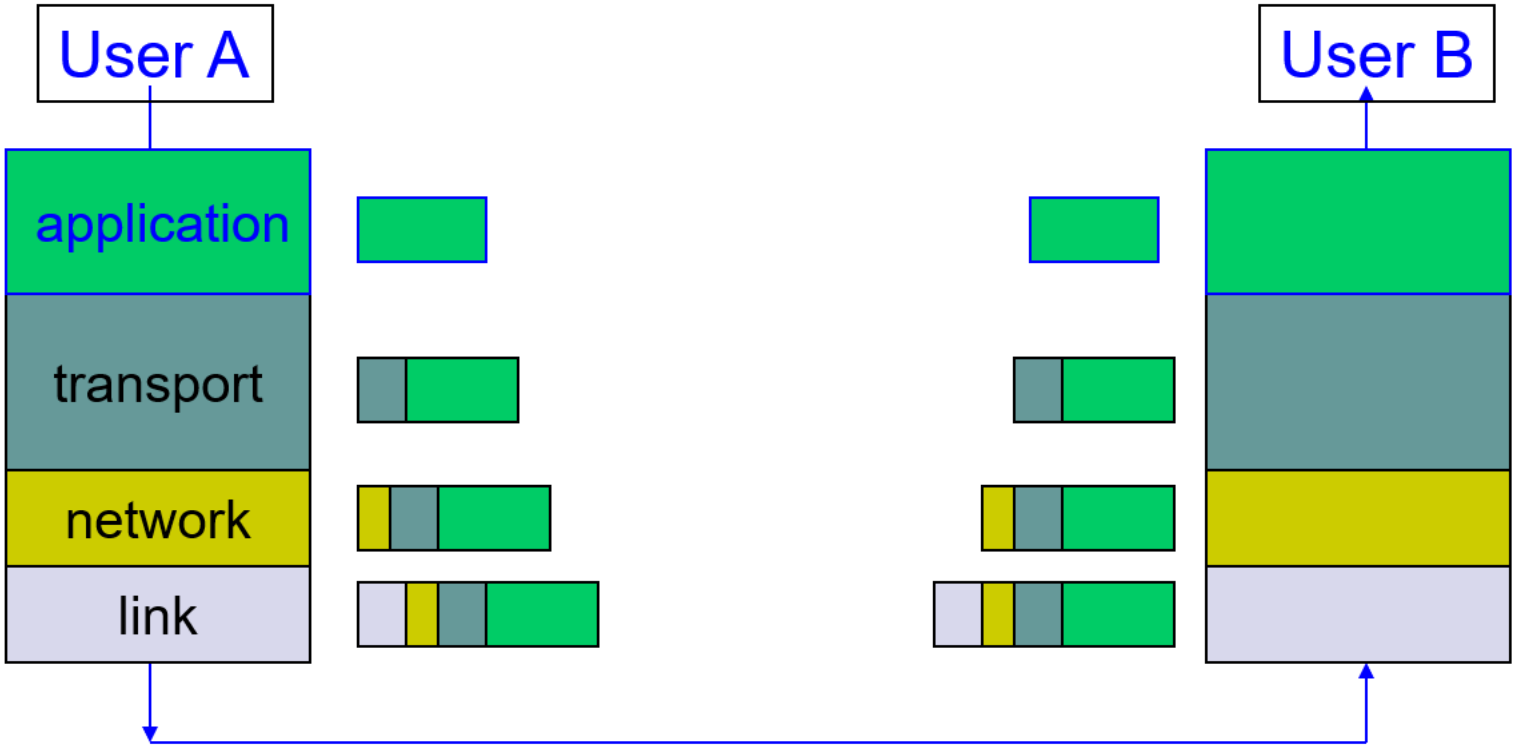
RICHIAMO: LAYER ENCAPSULATION



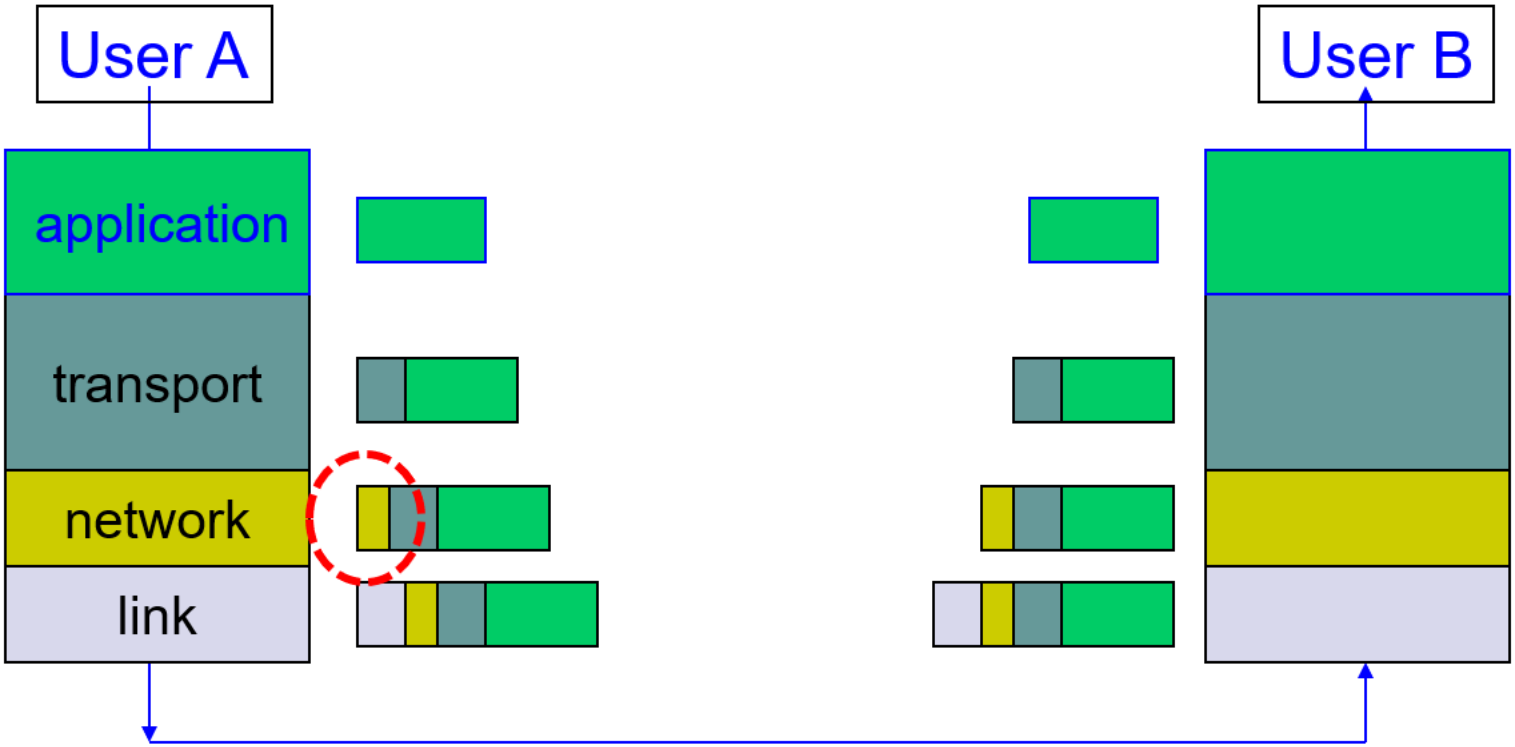
RICHIAMO: LAYER ENCAPSULATION

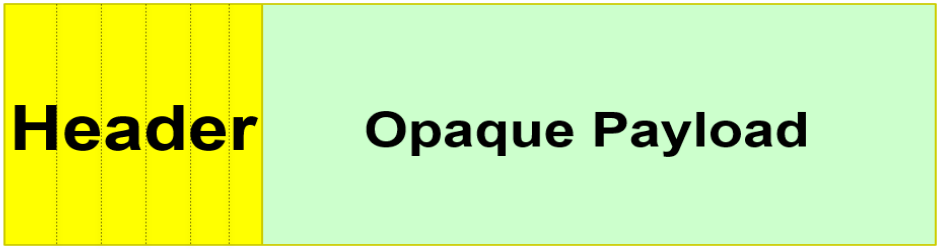


RICHIAMO: LAYER ENCAPSULATION

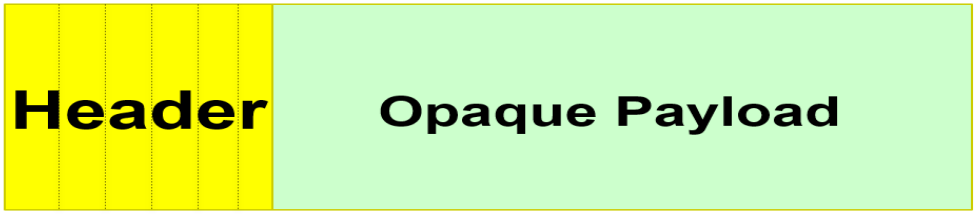


RICHIAMO: LAYER ENCAPSULATION

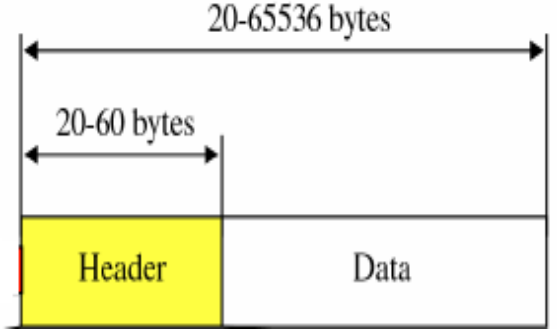




PACKET HEADER



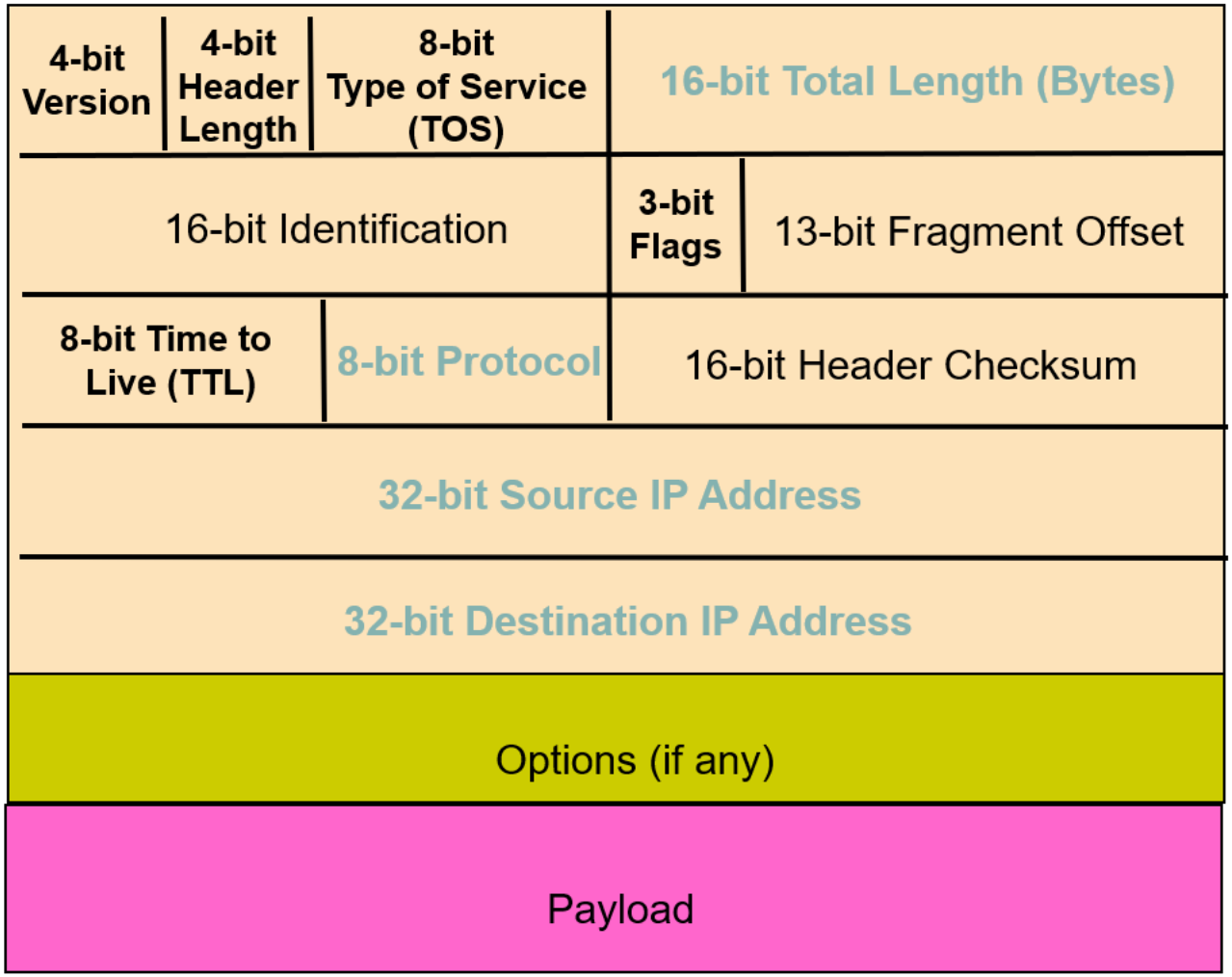
- packet header come interfaccia
  - un modo di passare l'informazione dal pacchetto al router
- cosa deve contenere questa interfaccia?
  - quali elaborazioni si intende fare sul pacchetto?
  - di quali informazioni ho bisogno per elaborare il pacchetto
  - il contenuto dell'header riflette le informazioni necessarie per eseguire questi tasks.



PROGETTARE L'HEADER

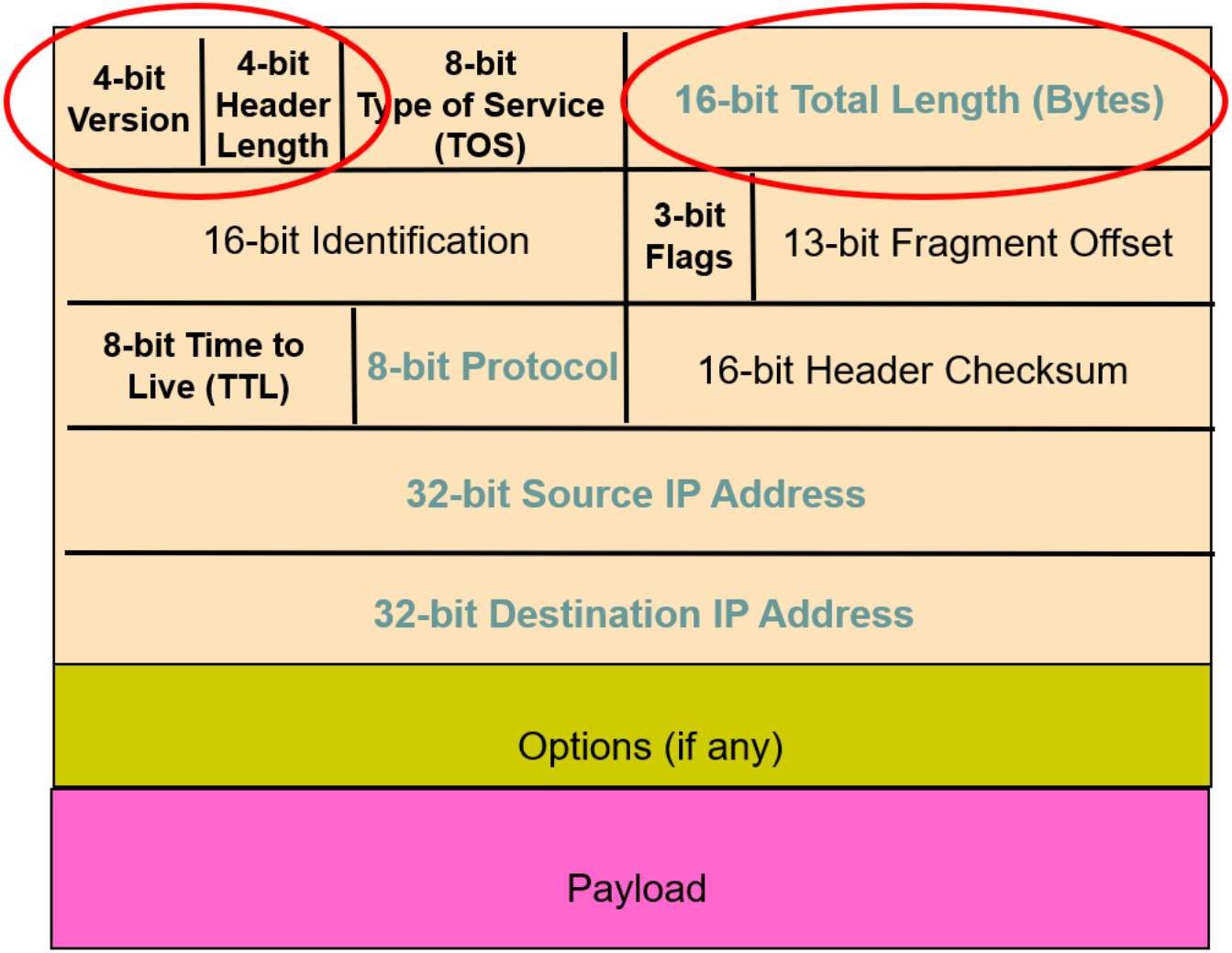
- individuare la struttura del pacchetto nel modo corretto
- quali funzioni?
  - portare il pacchetto a destinazione ed eventualmente inoltrare una risposta al mittente
  - trasportare i dati
  - comunicare al nodo destinatario cosa deve fare con il pacchetto
  - specificare eventuali operazioni particolari riguardanti il pacchetto
  - gestire eventuali problemi che possano sorgere nella trasmissione del pacchetto

20 BYTES PER L'HEADER STANDARD + LE OPZIONI



4 bytes (una parola) per ogni riga, 5 righe + eventuali opzioni

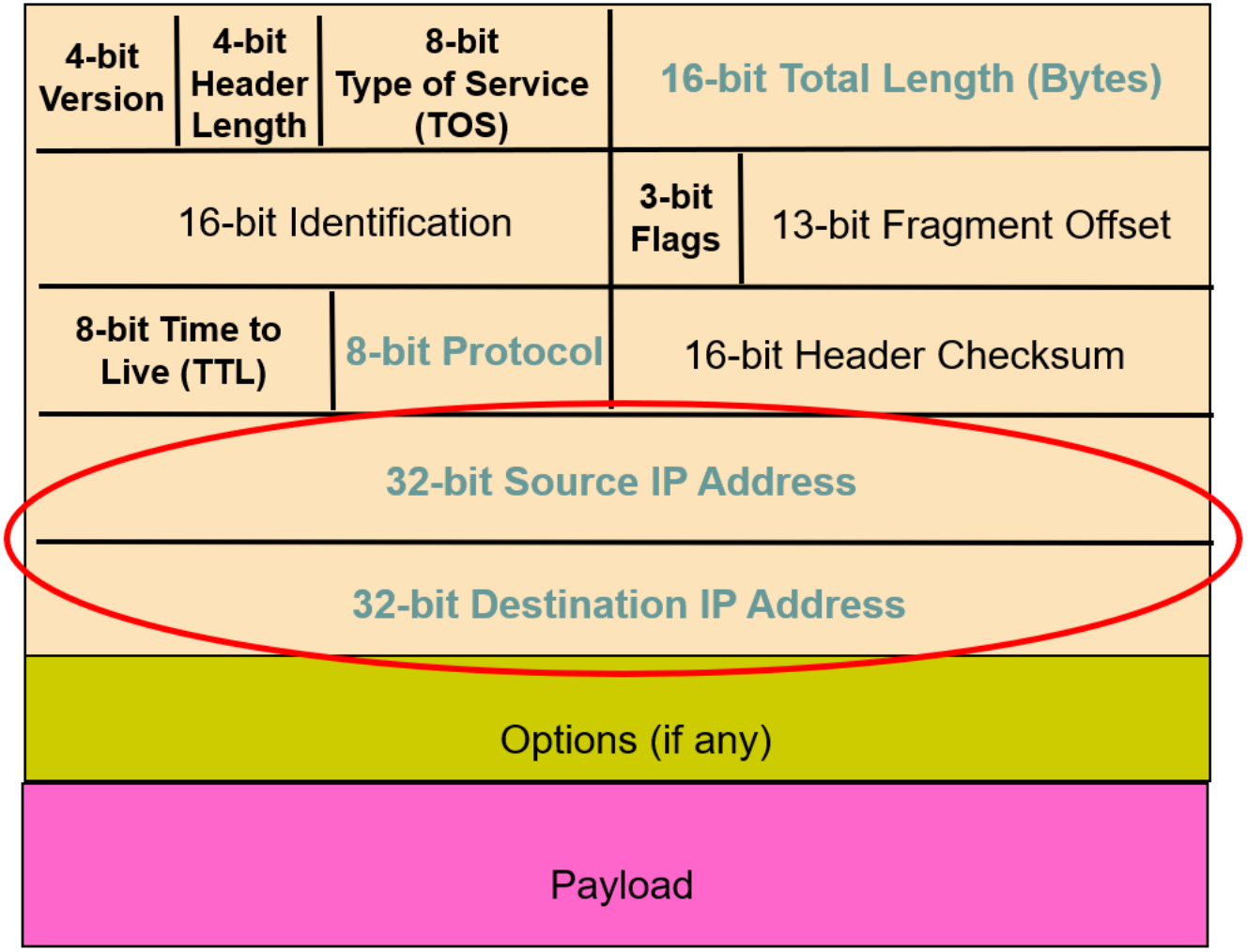
INDIVIDUAZIONE DELLA STRUTTURA DEL PACCHETTO



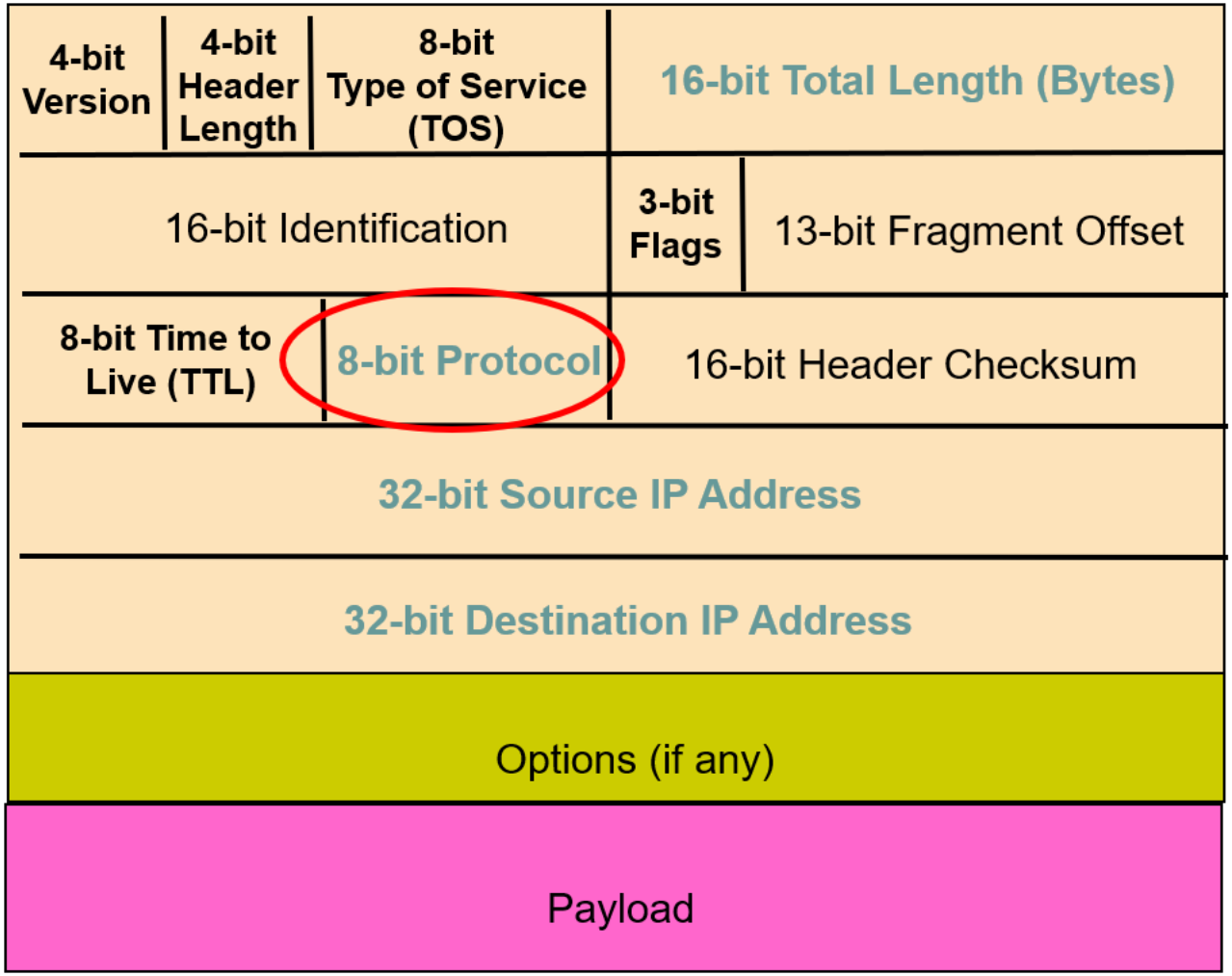
INDIVIDUAZIONE DELLA STRUTTURA DEL PACCHETTO

- version number (4 bits)
  - indica la versione del protocollo IP
    - in genere contiene "4" (per IPv4), qualche volta "6" (for IPv6)
  - analizzando questo campo, il router può interpretare correttamente la restante parte del datagram
- lunghezza dell'header (4 bits)
  - lunghezza dell'header in words (4 bytes)
  - in genere "5" (per un header IPv4 di 20-byte)
  - può contenere un valore più grande se vengono usate le IP option
- lunghezza totale (16 bits)
  - numbero di bytes totali nel pacchetto
  - dimensione massima è 65,535 bytes (216 -1)
  - anche se i link sottostanti possono accettare pacchetti più piccoli

CAMPI PER INSTRADARE IL PACCHETTO

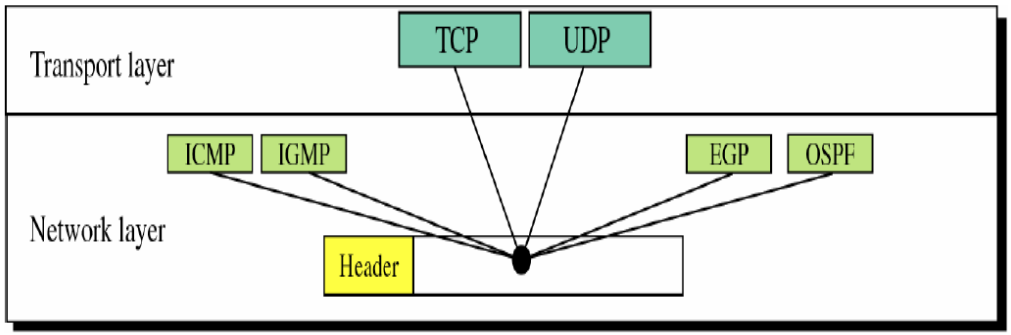


INDICAZIONE PER IL DESTINATARIO: COME GESTIRE IL PACCHETTO



PROTOCOL: COME GESTIRE IL PACCHETTO

- campo di 8-bit indica lo specifico protocollo a cui il payload di questo datagram IP deve essere passato
- usato nell'host destinazione finale del pacchetto per guidare il processo di demultiplexing
- numero di protocollo:
  - "colla" che collega il livello rete ed il livello trasporto
  - l'indicazione della porta invece la "colla" che collega i livelli trasporto ed applicazione
- valori: 1 –ICMP, 2 –IGMP, 6 –TCP, 17 –UDP
  - notare TCP, UDP protocolli del livello superiore
  - ICMP e IGMP protocolli dello stesso livello IP

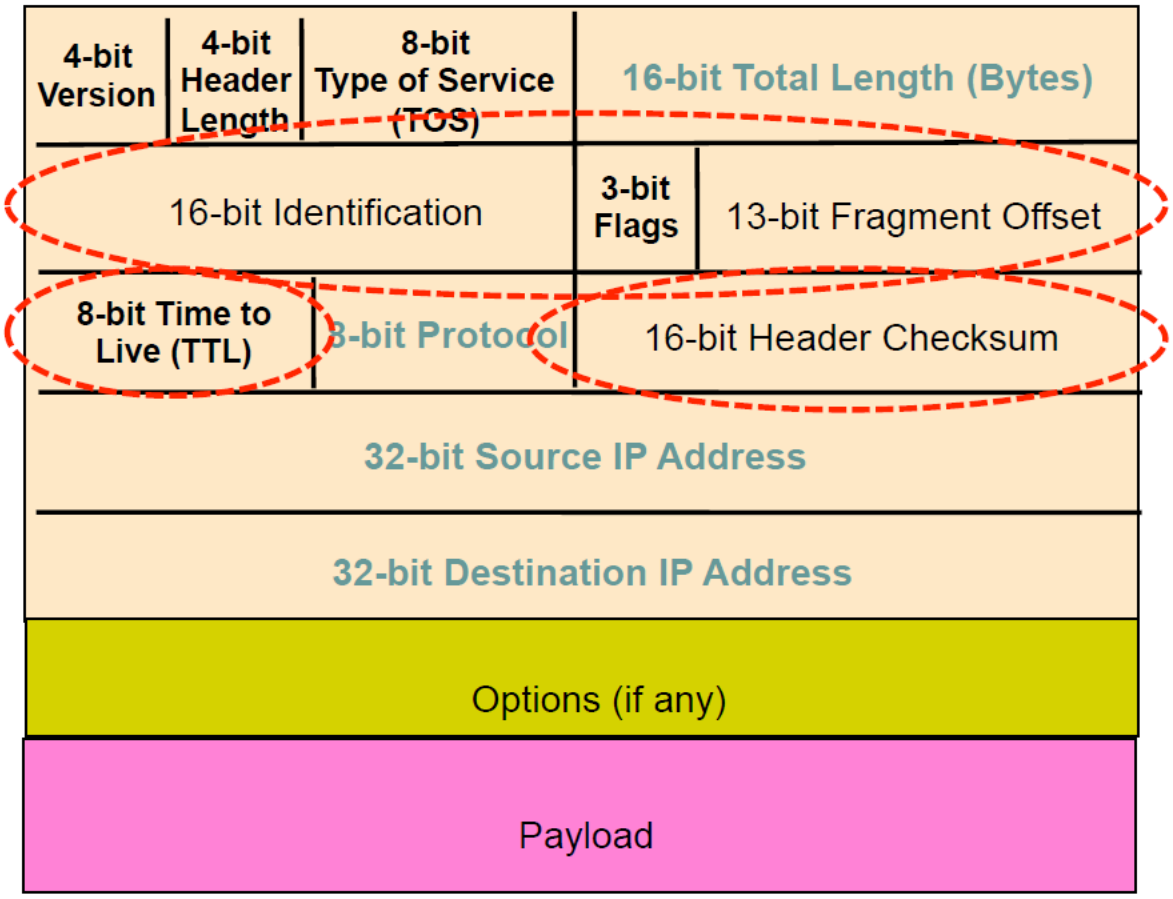


FORMATO DEI PACCHETTI: QUESITI

- un datagramma IP arriva ad un router ed i primi 8 bits sono 01000010. Il ricevente scarta il pacchetto. Perchè?
  - Soluzione: Il pacchetto deve essere corrotto. Infatti i 4 bit (0010) più a sinistra mostrano la versione, che è corretta. I successivi 4 bits mostrano la lunghezza dell'header in words (2 X 4=8), che è sbagliata. Il minimo numero di bytes dell'header è 20

- in un pacchetto IP, il valore di HLEN è1000 in binario. Il acchetto contiene campi opzionali?
  - Soluzione: il valore di HLEN è 8, che significa che il numero totale di bytes dell'header è 8 X 4= 32 bytes. I primi s0 bytes sono il main header, i successivi 12 bytes sono utilizzati per le opzioni.

GESTIRE LA TRASMISSIONE



HEADER CORRUPTION: CHECKSUM

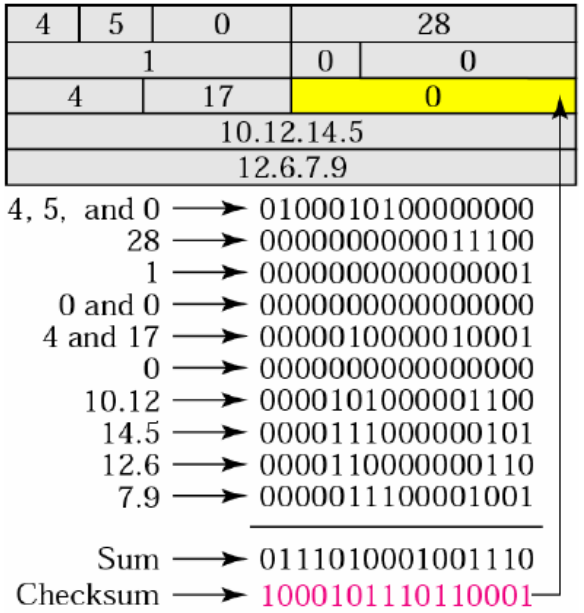
- verificato mediante una forma particolare di checksum (16 bit) calcolata sull'header del pacchetto
- checksum ricalcolata e memorizzata da ogni router
  - TTL ed alcuni campi opzionali dell'header possono cambiare ad ogni hop
- se il pacchetto non è corretto, il router scarta il pacchetto, evitando di elaborare informazione corrotta
- quando un router:
  - invia un pacchetto calcola la checksum del pacchetto, escludendo il campo checksum stesso. Quindi inserisce il valore calcolato nel campo checksum del pacchetto
  - riceve un pacchetto, ricalcola la checksum e verifica che sia uguale a quella inviata
- checksum basata su somma in complemento a 1 e complemento a 1 del risultato, per rendere iù semplice il processo di verifica.

IL CAMPO CHECKSUM: INVIO

- porre il campo checksum del pacchetto a 0
- dividere l'header in sezioni di 2 bytes
- calcolare la somma di tutte le sezioni in complemento ad 1
- calcolare il complemento ad 1 del risultato, per ottenere la checksum
- inserire il risultato nel campo checksum del pacchetto ed inviarlo

Each intermediate router must:

compute checksum for every outgoing packet

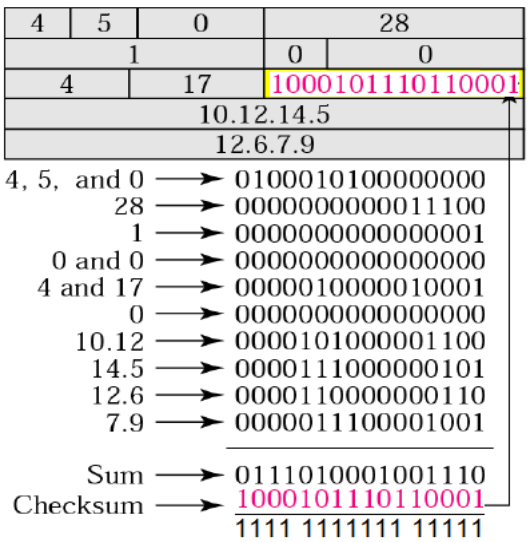


IL CAMPO CHECKSUM: RICEZIONE

- ricalcola la somma in complemento ad 1 di tutto l'header, includendo anche il campo checksum
  - pacchetto corretto se il risultato totale è una sequenza di 16 valori uguali a 1
- infatti, se il pacchetto non è corrotto
  - la somma di tutti i campi, esclusa la checksum, è uguale alla checksum calcolata dal mittente, se il pacchetto non è corrotto
  - se aggiungo il complemento ad 1 della checksum, ricevuto col pacchetto, ottengo una sequenza di 1

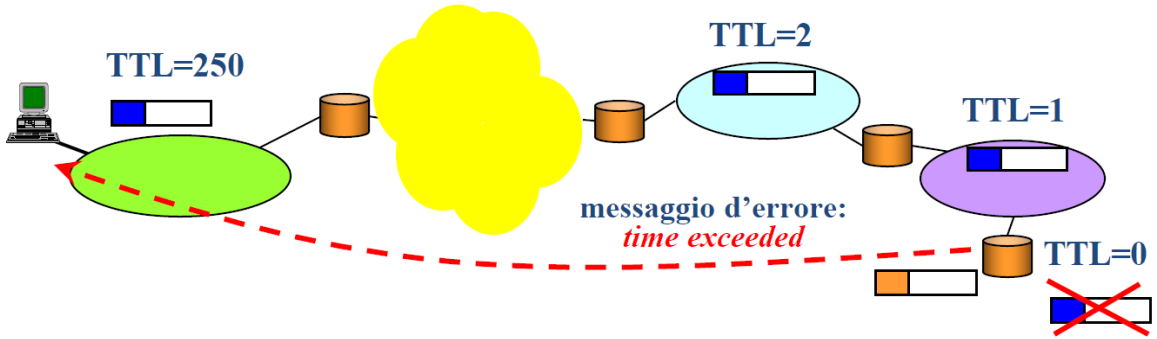
Each intermediate router must:

verify / recompute checksum on every incoming packet

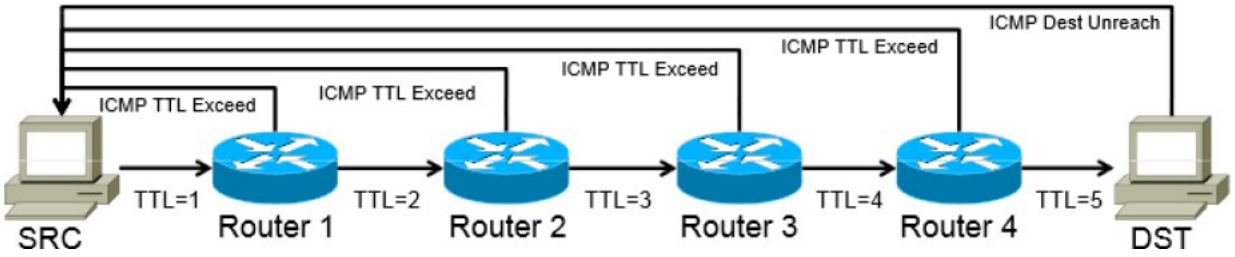


TTL: TIME TO LIVE

- campo a 8-bit: controlla il massimo numero di hops di un datagram
- decrementato ogni volta che il datagram è elaborato da un router
- quando raggiunge il valore 0, il datagram, viene eliminato
- assicura che
  - il datagram non sia inoltrato in un loop infinito
  - di limitare la zona in cui un datagram viene inoltrato



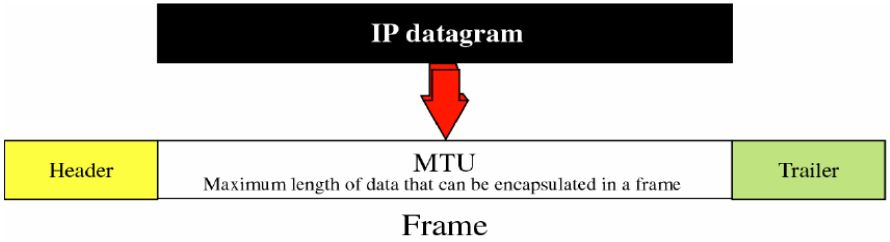
TTL in Traceroute:



IP DATAGRAM FRAGMENTATION

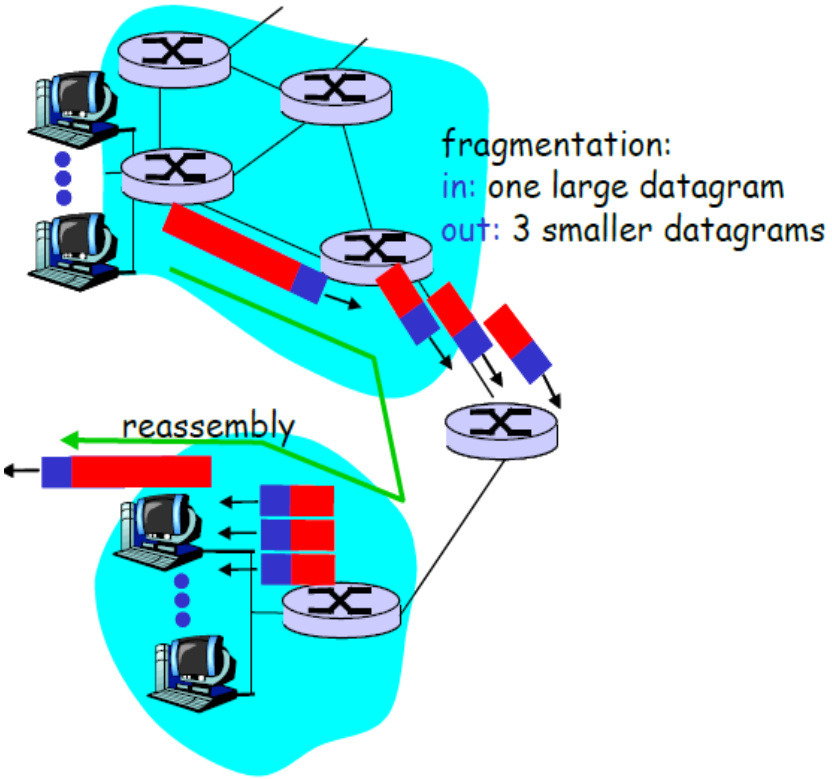
MAXIMUM TRANSFER UNIT (MTU)

- quantità massima di dati che possono essere trasportati da un collegamento a livello link
- limite sulla dimensione del datagram IP che può essere trasportato a livello fisico
- diverso per un protocollo data-link rispetto ad un altro
  - Token Ring (4 Mbps): MTU = 4,464 bytes
  - Ethernet: MTU = 1,500 bytes
  - PPP: MTU = 296 bytes



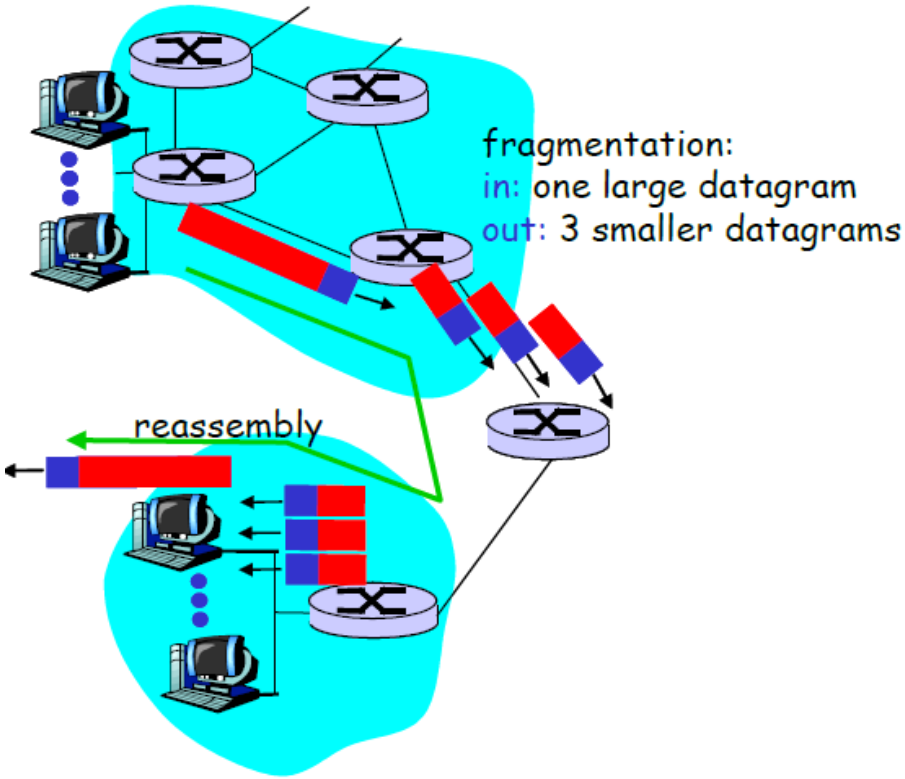
Hard limit on IP datagram size is not a problem.
What is a problem is that each of the links along the route between sender and receiver can use different link-layer protocols,
and each of these protocols can have different MTUs.

IP DATAGRAM FRAGMENTATION



- processo di divisione di un datagram in fragmenti più piccoli che soddisfino i vincoli di MTU del protocollo link-layer
- il datagram può essere fragmentato
  - dall'host sorgente
  - da un qualsiasi router sul cammino verso la destinazione
- il riassemblaggio del datagram deve essere fatto solo dall'host destinazione
  - parti del datagram fragmentato possono seguire cammini diversi

IP DATAGRAM FRAGMENTATION



- un datagram fragmentato può essere ulteriormente fragmentato, se sul cammino verso la destinazione esiste una rete con un MTU minore
- in un datagram fragmentato
  - ogni fragmento ha il proprio header
  - l'host o il router che effettua la fragmentazione cambia il valore di tre campi
    - flags
    - fragmentation offset
    - total length
  - a parte la checksum , gli altri campi sono invariati

IP DATAGRAM FRAGMENTATION

Identification: campo di 16-bit

- identifica univocamente il datagram per ogni host sorgente
- per garantire l'univocità, IP usa un contatore per etichettare ogni datagram
  - invio di un datagram: incremento del contatore e copia del valore corrente nel campo identification
  - quando un datagram viene fragmentato, l'identificatore viene copiato in tutti i pacchetti fragmento
- identification number utilizzato dalla destinazione per riassemblare i pacchetti

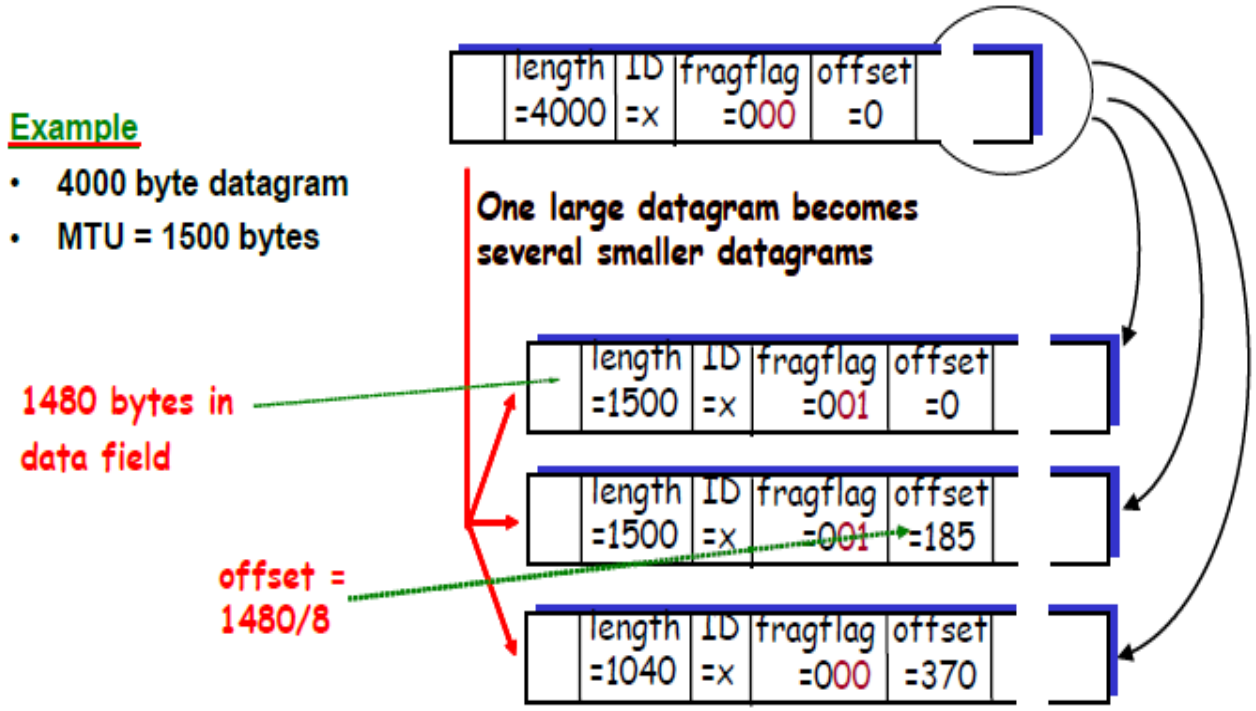
Flags: campo di 3 bits

- bit 1: riservato
- bit 2: do not fragment bit
  - se vale 1, il segmento non deve essere fragmentato, vale 0 altrimenti
  - se non soddisfa MTU, il segmento viene scartato dal router e viene inviato un messaggio ICMP
- bit 3: more fragment bit
  - se vale 1, non è l'ultimo fragmento del datagram, vale 0 altrimenti

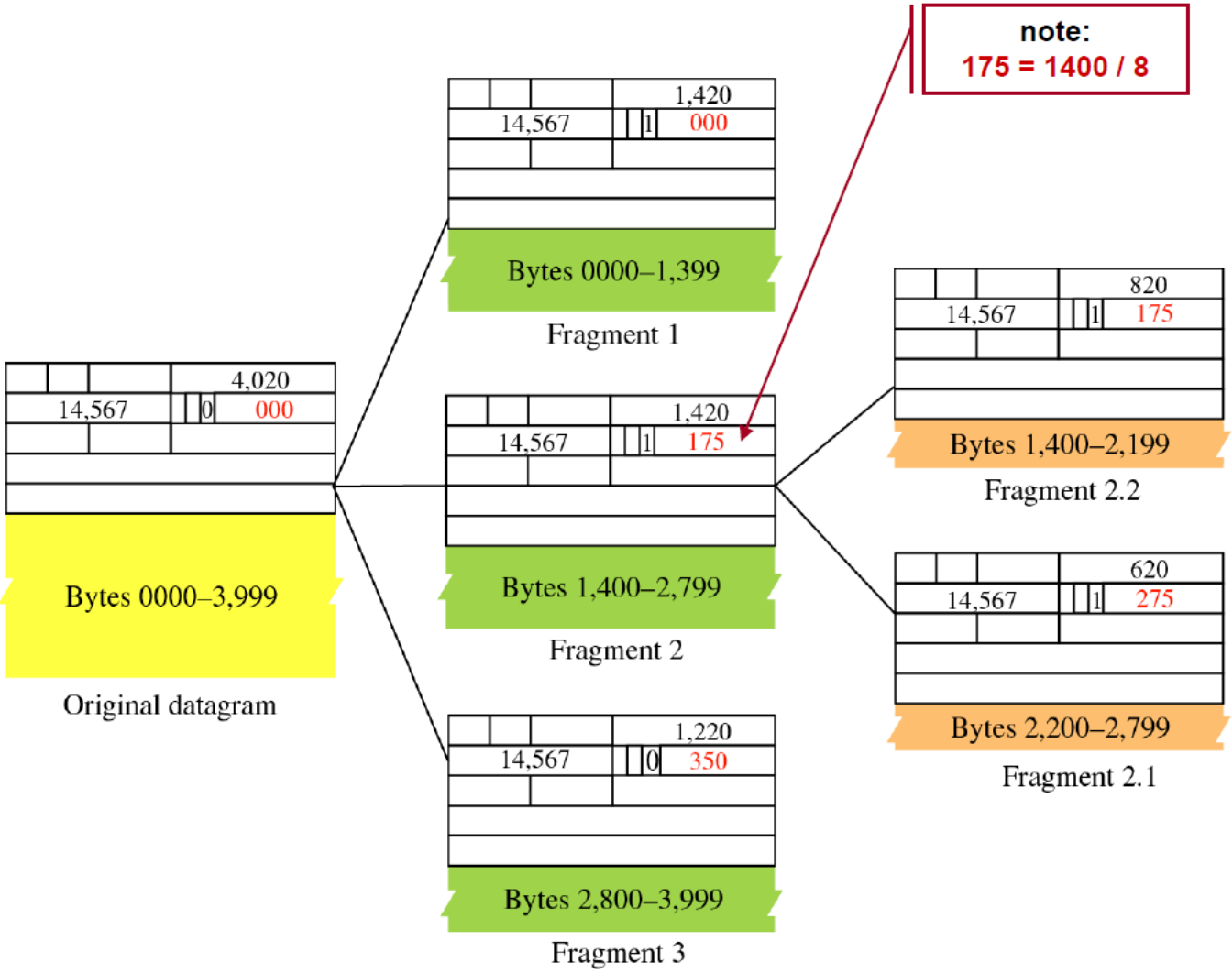
IP DATAGRAM FRAGMENTATION

Offset: campo di 13-bit:

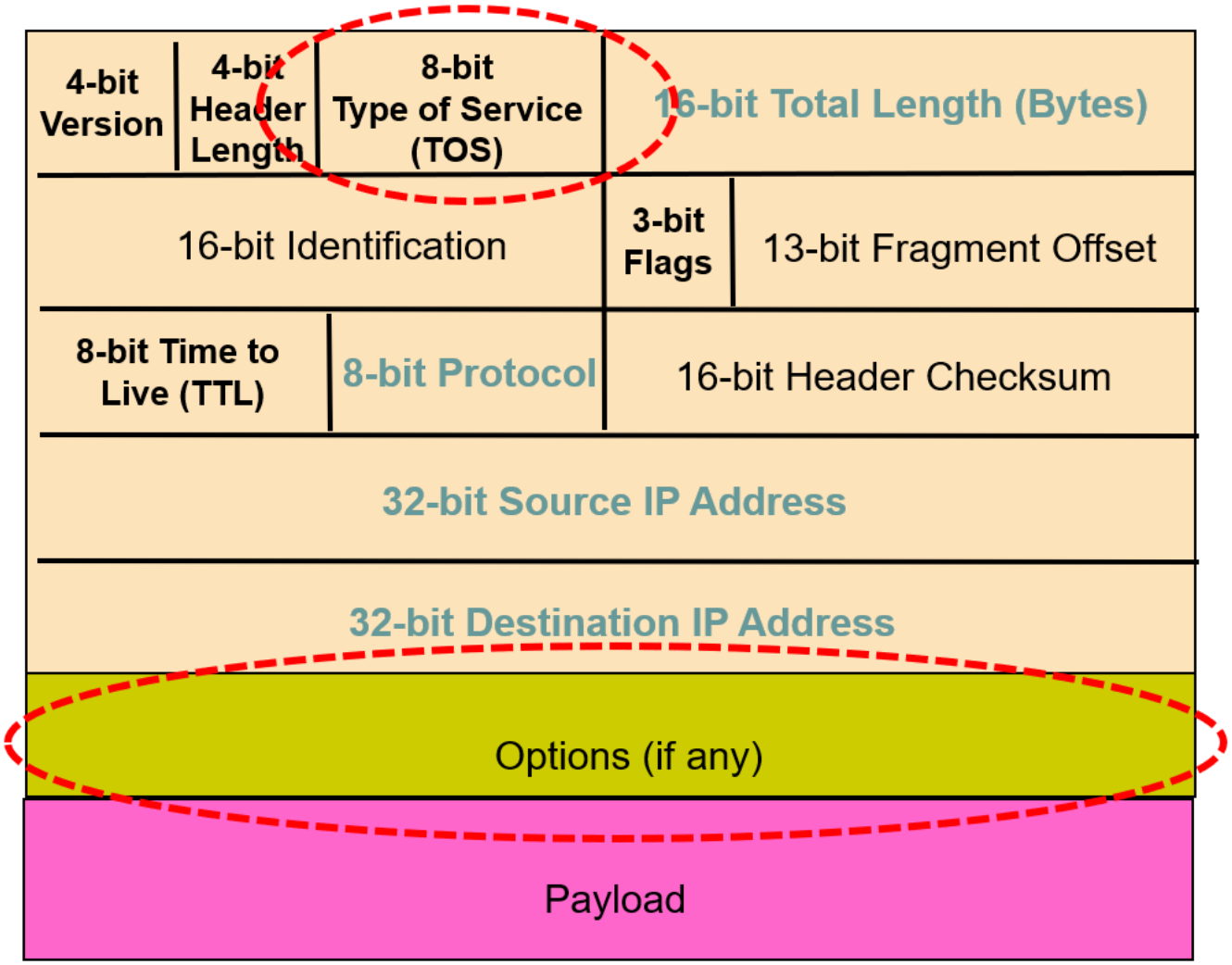
- posizione del fragmento all'interno del datagram
- misurato in bytes
- host e routers devono scegliere una lunghezza del fragmento divisibile per 8



FRAGMENTAZIONE DI UN FRAGMENTO



GESTIONE SPECIALE: TOS

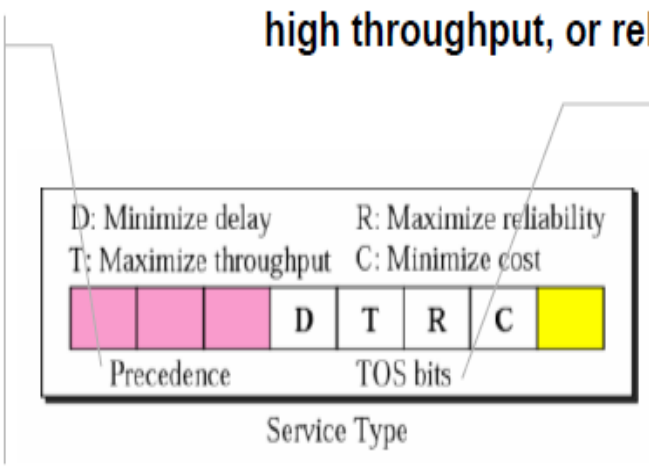


GESTIONE SPECIALE: TOS (TYPE OF SEVICE)

"Type of Service", or "Differentiated Services Code Point (DSCP)" (8 bits)

- permette di trattare in modo diverso pacchetti caratterizzati che richiedono gestioni diverse
- ad esempio: basso delay per audio, alta banda per bulk transfer
- ridefinito più volte

Precedence defines the priority of datagram in case of congestion. If a router is congested and needs to discard some datagrams, those datagrams with lowest precedence are discarded first. Network management datagrams have the highest precedence!



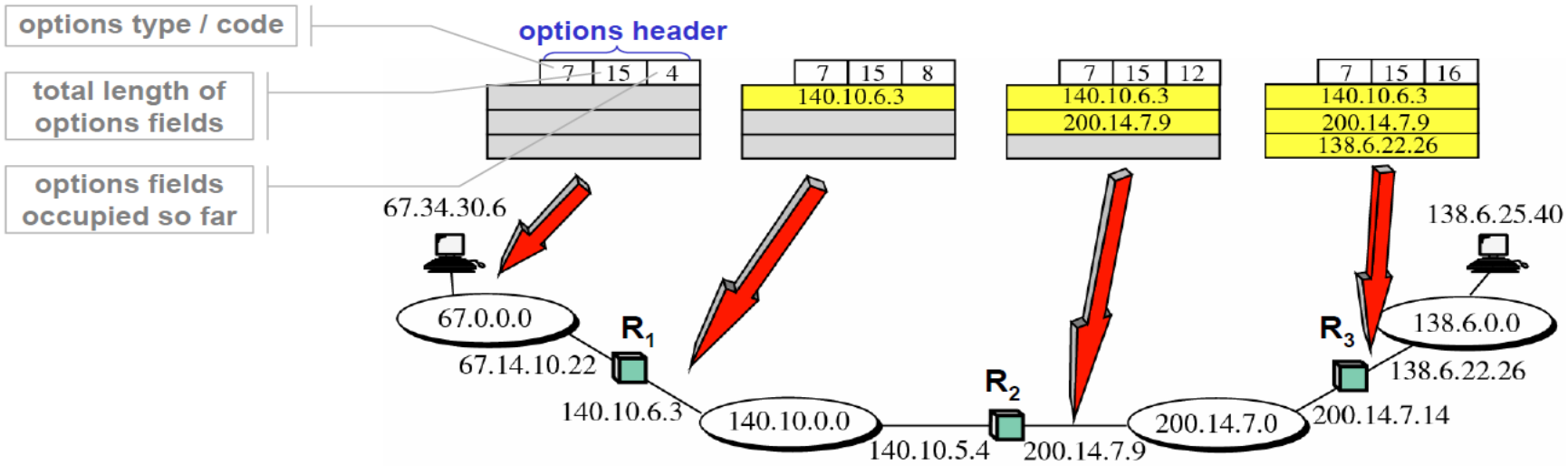
Although each TOS bit has a special meaning, only one bit can be set to 1 in each datagram.
0000 – normal type of service
0001 – minimize cost
0010 – maximize reliability
0100 – maximize throughput
1000 – minimize delay

OPTIONS

- campo opzionale di 32 bits: permette l'espansione dell'header per scopi speciali

1) record route option: utilizzato per tracciare la rotta di un datagram

- il mittente crea un campo vuoto per una sequenza di indirizzi IP
- massimo 9 slot: (40 bytes per le opzioni –4 bytes per l'header del campo opzione) / 4 bytes per ogni indirizzo IP
- ogni router inserisce l'indirizzo di uscita nel campo option
- traccia dei routers attraversati da un messaggio di PING (copiati poi nella risposta ICMP)



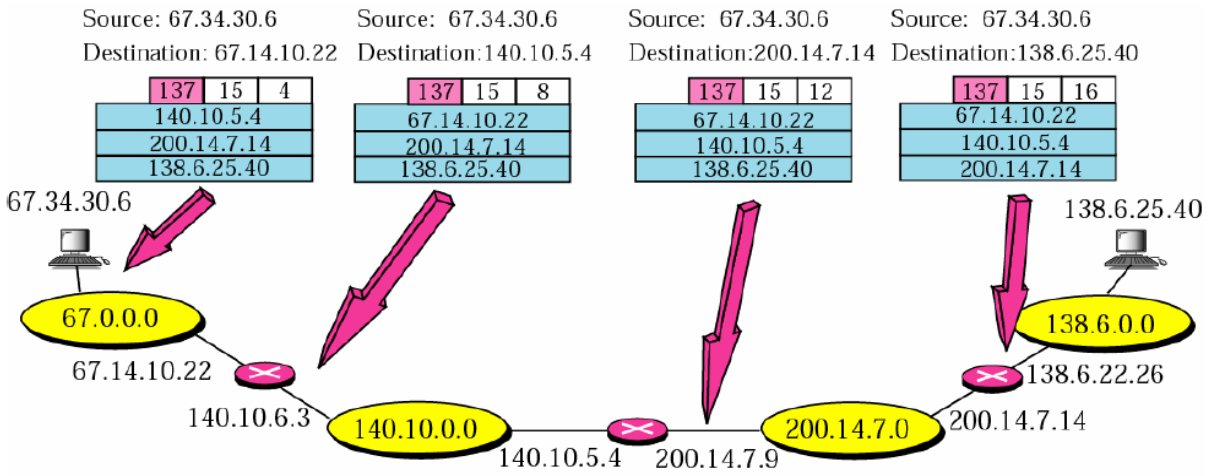
OPTIONS

2) timestamp option

- simile ad record route + registra il tempo di elaborazione di un datagram da parte di un router, in millisec

3) strict source route option:

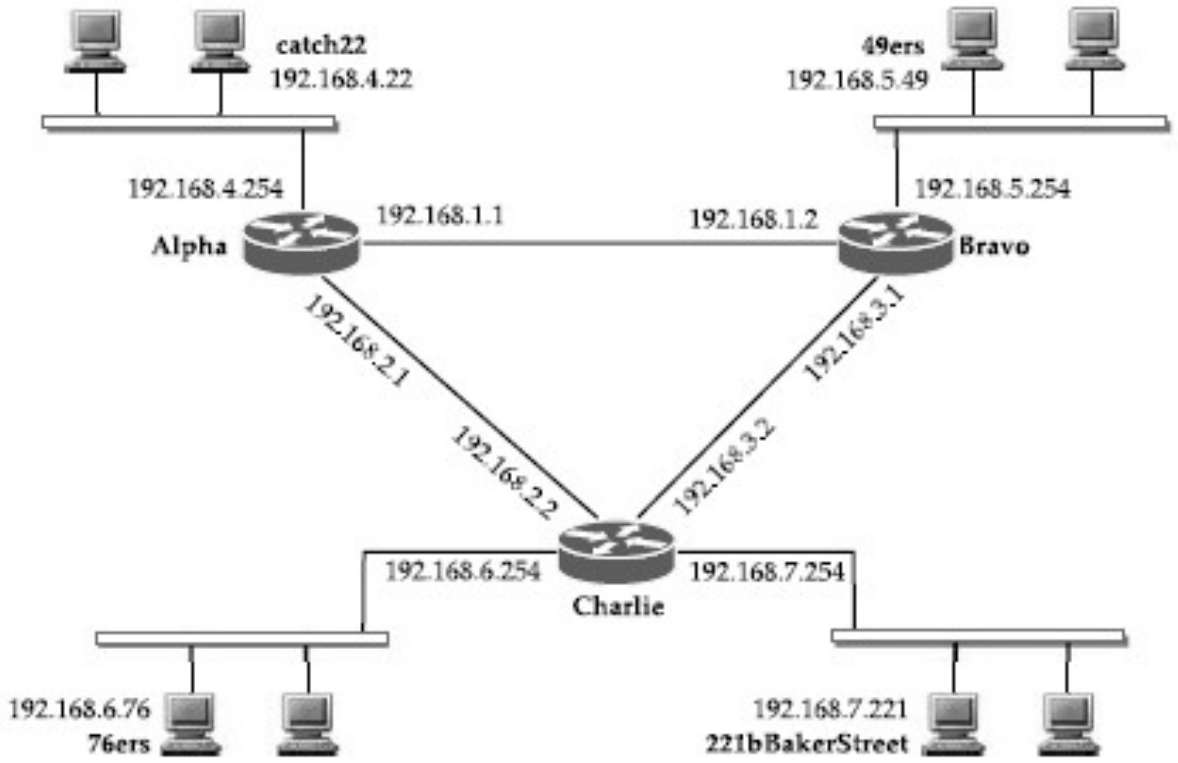
- usata dal mittente per predeterminare la rotta di un datagramma
- il nodo sorgente fornisce una lista di indirizzi IP, una sequenza di routers che il datagram deve attraversare nel suo cammino verso la destinazione
- esiste la versione loose source route option ogni router nella lista deve essere attraversato, ma il datagram può attraversare anche altri routers



PROTOCOLLI DI ROUTING IN RETI IP

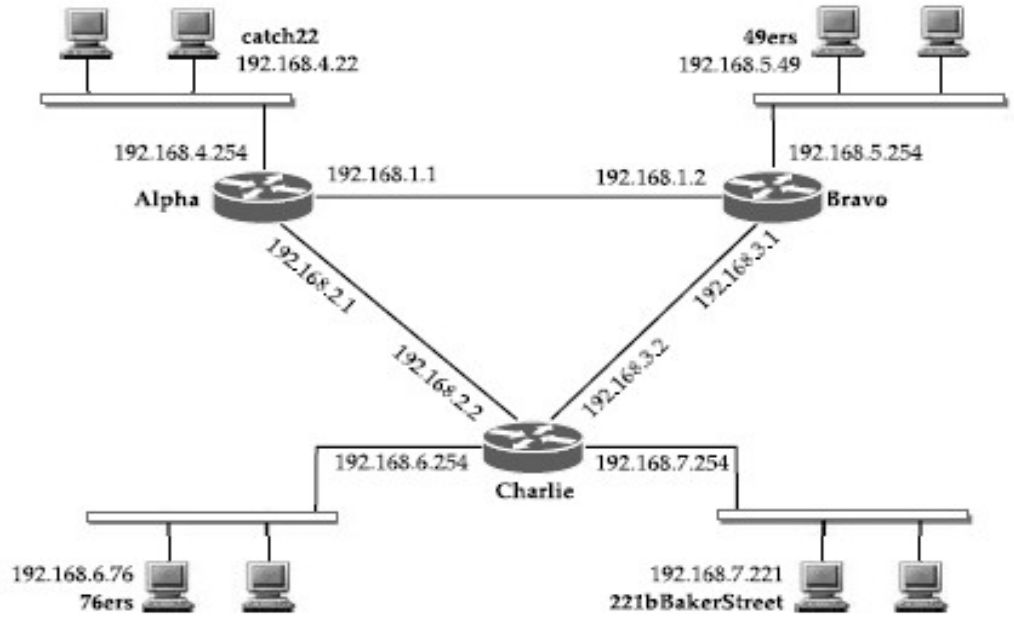
- nelle lezioni precedenti, abbiamo analizzato diversi algoritmi di routing, in questa lezione analizzeremo alcuni protocolli di routing, e come sono implementati in una rete IP
- Protocolli di routing strettamente collegati a come i nodi vengono indirizzati in un a rete IP
  - IP addressing
- IP network
  - nodi sono routers ed end host
  - link identificati da indirizzi IP delle interfacce di nodi e dei router
- Diversi tipi di pacchetti sulla rete IP
  - messaggi per l'implementazione dei protocolli di routing:
    - origine e destinazione sono i routers
  - traffico tra end-host
  - traffico a livello IP per implementare protocolli di gestione

ROUTING IN RETE IP



- per semplicità, classfull addressing e reti di classe C, subnet mask /24, fissa
- notare: specifici indirizzi IP/subnet per il collegamento diretto tra due routers
- più interfacce per ogni router: router identificati da un router ID che può essere gli indirizzi di una delle interfacce o un identificatore diverso

ROUTING IN RETE IP E TABELLE DI ROUTING



Router: Alpha		Router: Bravo		Router: Charlie	
Network/Mask	Next Hop	Network/Mask	Next Hop	Network/Mask	Next Hop
192.168.1.0/24	direct	192.168.1.0/24	direct	192.168.1.0/24	192.168.2.1
192.168.2.0/24	direct	192.168.2.0/24	192.168.1.1	192.168.2.0/24	direct
192.168.3.0/24	192.168.1.2	192.168.3.0/24	direct	192.168.3.0/24	direct
192.168.4.0/24	direct	192.168.4.0/24	192.168.1.1	192.168.4.0/24	192.168.2.1
192.168.5.0/24	192.168.1.2	192.168.5.0/24	direct	192.168.5.0/24	192.168.3.1
192.168.6.0/24	192.168.2.2	192.168.6.0/24	192.168.3.2	192.168.6.0/24	direct
192.168.7.0/24	192.168.2.2	192.168.7.0/24	192.168.3.2	192.168.7.0/24	direct

ROUTING IN RETE IP

- l'esempio precedente può essere generalizzato al caso di classless routing
- CIDR: aggiunta di network mask esplicite nelle tabelle di routing
- la tabella di routing contiene:
  - indirizzi di sottoreti
  - può contenere anche indirizzi di host, se un host è collegato direttamente al router: sconsigliato perchè può aumentare in maniera consistente la dimensione della tabella di routing, con impatto negativo sul processo di inoltro

ROUTING IN RETE IP

- un aspetto importante di TCP/IP è che tutti i tipi di comunicazione devono avvenire all'interno di TCP stesso
- nessuna rete separata o canale dedicato per la comunicazione di messaggi di controllo o di routing
  - viaggiano sulla stesse rete dove transitano i messaggi scambiati tra end host
- i protocolli di routing sfruttano quindi funzionalità a livelli diversi dello stack TCP/IP
  - RIP protocol: messaggi di routing utilizzano UDP, a livello trasporto
  - BGP: utilizza TCP, a livello trasporto
  - OSPF utilizza direttamente IP, per cui è necessaria una identificazione a livello di protocollo

PROTOCOLLI DI ROUTING

	RIP v1	RIP v2	IGRP	EIGRP	OSPF	IS-IS	BGP
Interior/Exterior?	Interior	Interior	Interior	Interior	Interior	Interior	Exterior
Type	Distance Vector	Distance Vector	Distance Vector	Hybrid	Link-state	Link-state	Path Vector
Default Metric	Hopcount	Hopcount	Bandwidth/Delay	Bandwidth/Delay	Cost	Cost	Multiple Attributes
Administrative Distance	120	120	100	90 (internal) 170 (external)	110	115	20 (external) 200 (internal)
Hopcount Limit	15	15	255 (100 default)	224 (100 default)	None	None	EBGP Neighbors: 1 (default) IBGP Neighbors: None
Convergence	Slow	Slow	Slow	Very Fast	Fast	Fast	Average
Update timers	30 seconds	30 seconds	90 seconds	Only when change occurs	Only when changes occur; (LSA table is refreshed every 30 minutes, however)	Only when changes occur	Only when changes occur
Updates	Full table	Full table	Full table	Only Changes	Only Changes	Only changes	Only changes
Classless	No	Yes	No	Yes	Yes	Yes	Yes
Supports VLSM	No	Yes	No	Yes	Yes	Yes	Yes
Algorithm	Bellman-Ford	Bellman-Ford	Bellman-Ford	DUAL	Dijkstra	Dijkstra	Best Path Algorithm
Update Address	Broadcast	224.0.0.9	224.0.0.10	224.0.0.10	224.0.0.5 (All SPF Routers) 224.0.0.6 (DR's and BDR's)		Unicast
Protocol and Port	UDP port 520		IP Protocol 9	IP Protocol 88	IP Protocol 89		TCP port 179

RIP: STORIA

- Anni '60: utilizzo dei Distance Vectors in ARPANET
- Metà anni '70: XNS (Xerox Network system) routing protocol è il precursore del RIP di IP
- 1982: Release del routing software per BSD Unix
- 1988: RIPv1 (RFC 1058)
  - supporta solo classful routing
- 1993: RIPv2 (RFC 1388)
  - aggiunge subnet masks in ogni route entry della tabella di routing
  - Permette classless routing
- 1998: Versione corrente di RIPv2 (RFC 2453)

PROTOCOLLI DI ROUTING

In questo corso analizzeremo

- RIP – Routing Internet Protocol
- OSPF – Open Short Path First

RIP V1 (VERSIONE 1): CARATTERISTICHE

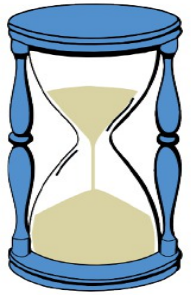
- un semplice protocollo intra-dominio
- implementa il protocollo distance vector basato sull'algoritmo di Bellman-Ford
  - con i noti problemi
    - slow convergence
    - funziona con reti di dimensioni limitate
  - implementazione molto vicino alla versione teorica
  - differenza: indirizzamento IP (utilizzavamo identificatori unici)
- metrica per la distanza : # di hop–ogni hop ha costo 1
  - valori tra 1 e 16
- costo max di un cammino = 15
  - di conseguenza RIP è usato in autonomous systems con un grafo di diametro inferiore a 15
    - (cammino = 16 hops: host unreachable)
- caratterizzato da un insieme di timers

RIP V1: CARATTERISTICHE

- dimensione massima dei messaggi RIP: 512 bytes
  - al più 25 blocchi per messaggio
  - se è necessario notificare più di 25 destinazioni: più messaggi RIP
- non prevista la trasmissione di una subnet mask
  - un router deve conoscere la struttura dell'indirizzo (classfull addressing) per distinguere separazione tra network prefix e host address
  - non adatto a CIDR
  - Superato da RIP v2

RIP V1: TEMPISTICA DEI MESSAGGI

- periodic routing update timer (25-30 secondi)
  - utilizzato per inviare periodicamente i DV
  - scelto casualmente nell'intervallo 25-35 secondi
  - Per evitare che tutti i router inviino un messaggio allo st o e generino picchi di traffico

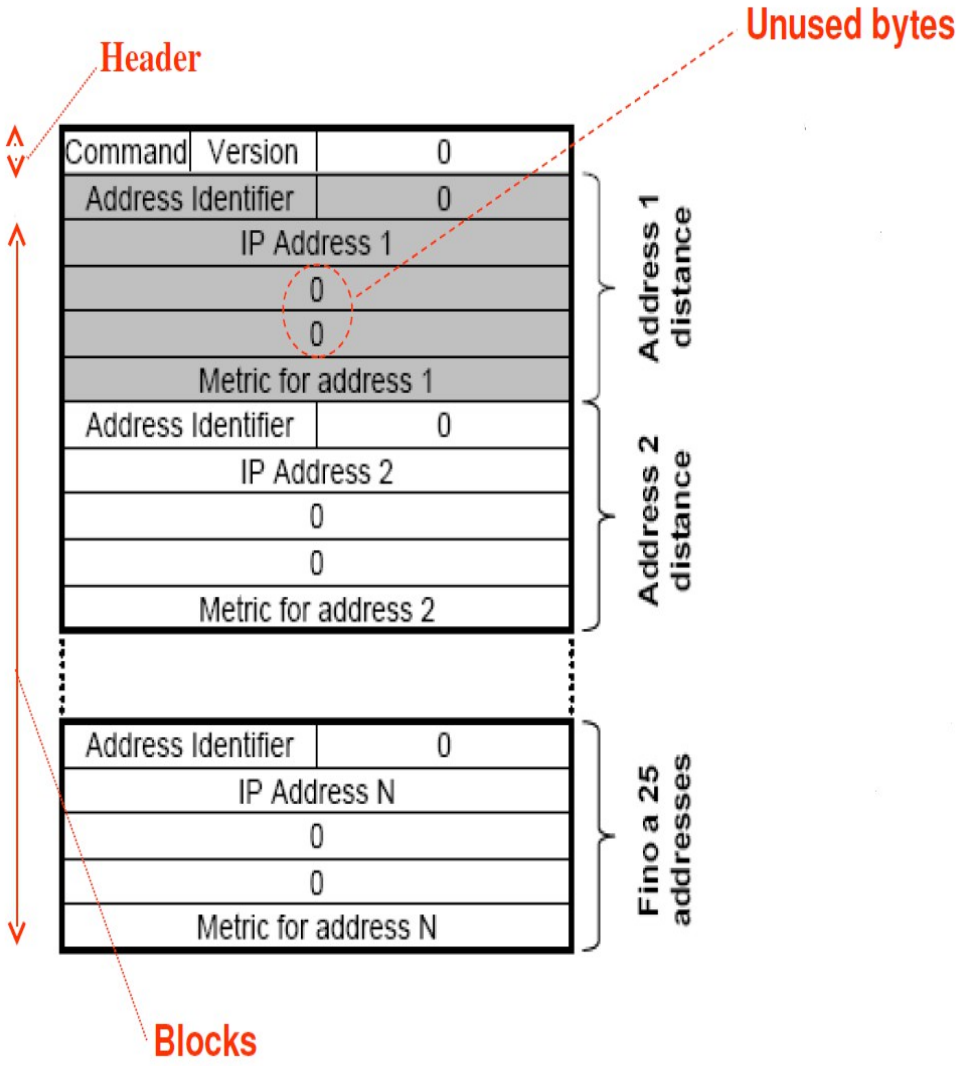


- Route invalid (o duration) timer (tipicamente 180 s)
  - se non si ricevono aggiornamenti per una certa entrata per 180 s, la route è dichiarata non valida
  - la route viene comunque annunciata con una distanza posta a 16
- Garbage collection timer (tipicamente 60-120 s)
  - una entrata non valida viene marcata, ma non immediatamente cancellata
  - per 120 secondi il router annuncia questa rotta con distanza infinita
  - dopo 120 secondi, la route invalida è cancellata

Triggered update (non collegata a nessun timer)

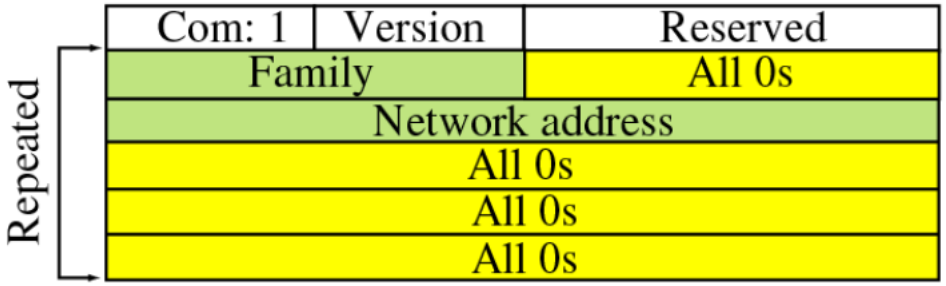
- in caso di cambiamento del valore di una metrica per una route, viene inviato subito un DV con solo le entry cambiate

RIP V1: FORMATO DEI PACCHETTI



- Header
  - command: tipo
    - richiesta di invio di Distance Vectors (codice 1)
    - risposta (stimolate o meno) (codice 2)
  - version: versione RIP
- Block
  - address identifier: identifica la famiglia di indirizzi usati (2=IP)
  - IP address
    - rete, sotto-rete, host
  - metric
    - distanza dalla rete indicata dall'indirizzo IP

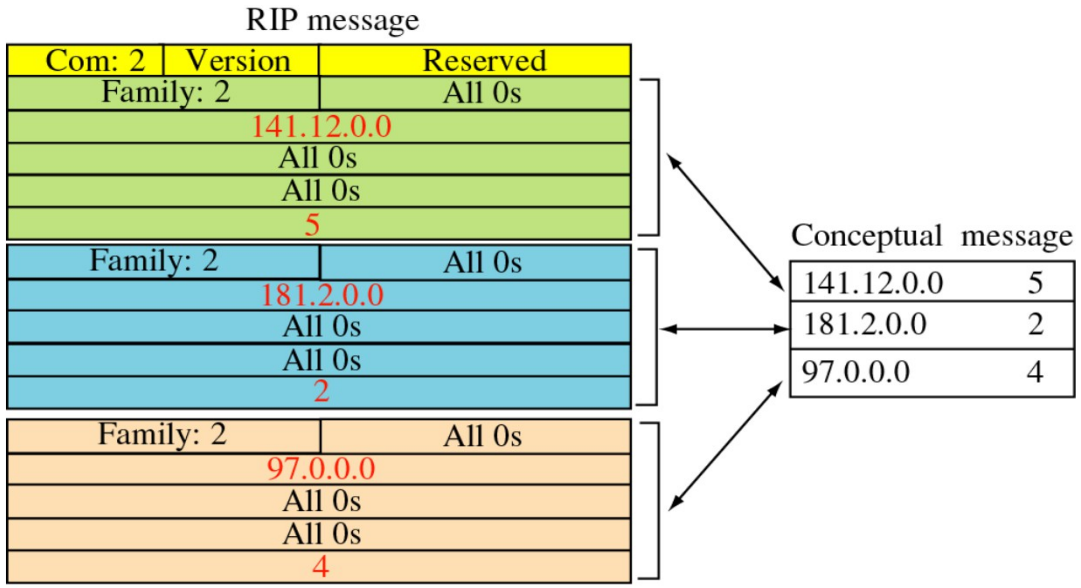
RIP V1: RICHIESTA INVIO



richiesta di invio di Distance Vectors

- inviato da un router ad un vicino per richiedere tutte le rotte in suo possesso o una parte di esse
- possono provenire da
  - un router appena attivato
    - invia una richiesta su ogni interfaccia e interroga tutti i router adiacenti
  - un router con qualche destinazione "in scadenza"
- possono riguardare
  - tutte le destinazioni
  - una destinazione in particolare

RIP V1: RISPOSTA

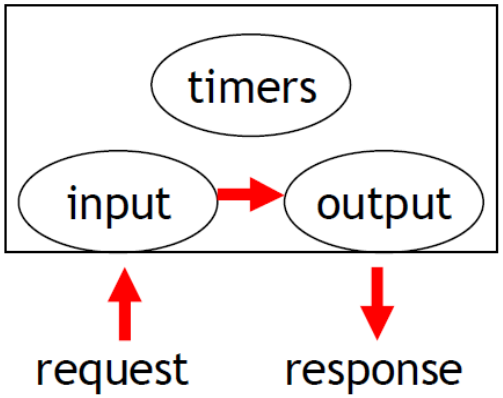


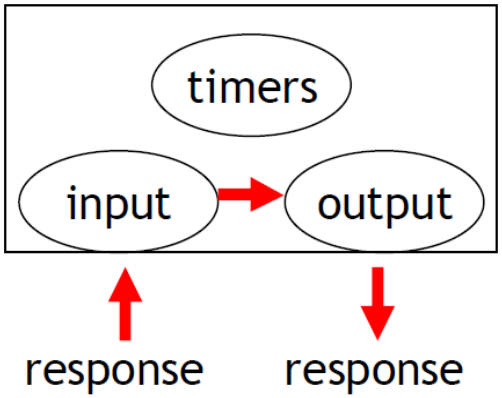
- possono arrivare da routers che
  - eseguono updates regolari
  - rispondono a specifiche richieste
  - triggered updates
- contengono il DV
  - al massimo 25 rotte per risposta
  - >25 rotte trasferite con più pacchetti UDP

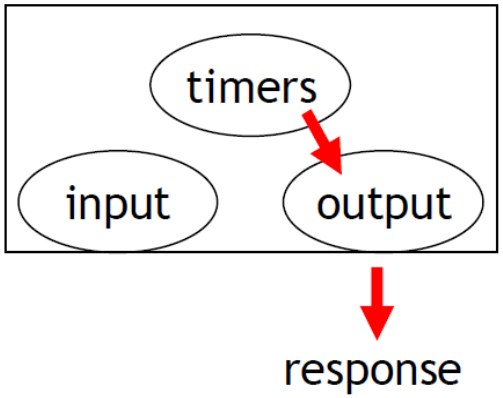
RIP: GENERAZIONE DEI MESSAGGI

Output generati

- quando il router viene attivato
- se arriva un input
- routing updates regolari



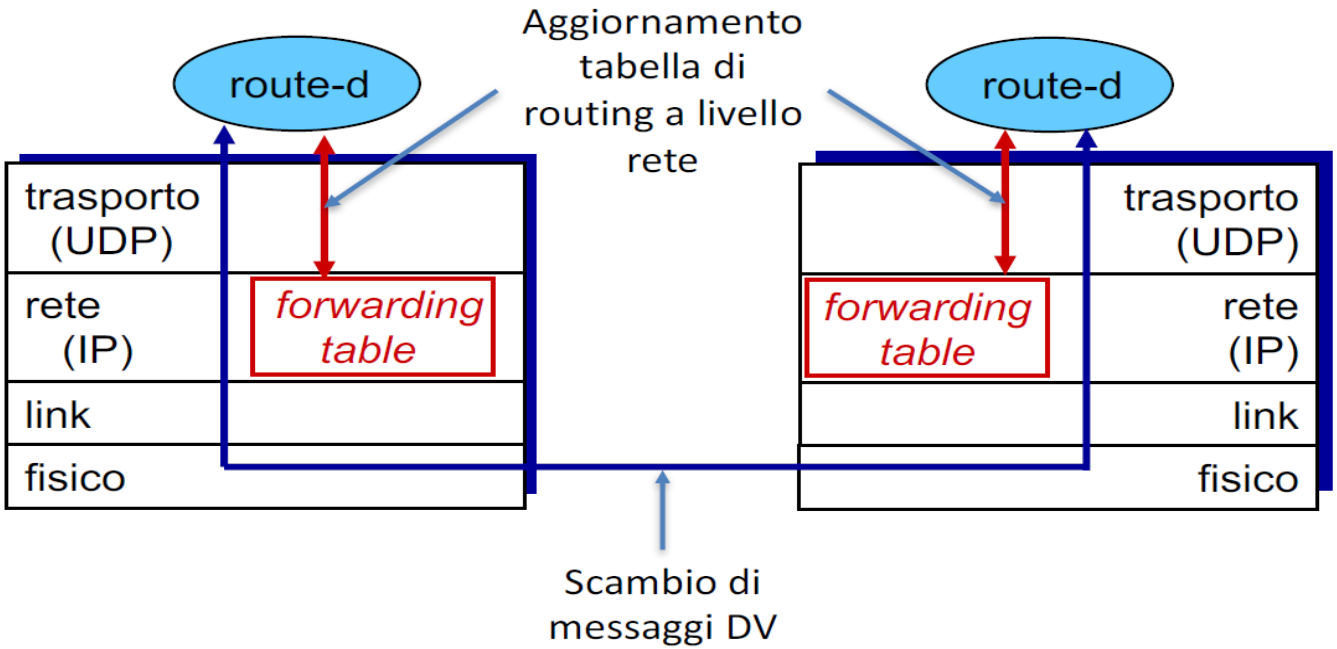




RIP V1: DEMONE SU UDP

- le tabelle di routing di RIP sono gestite da un processo applicativo (route-d), un demone Unix.

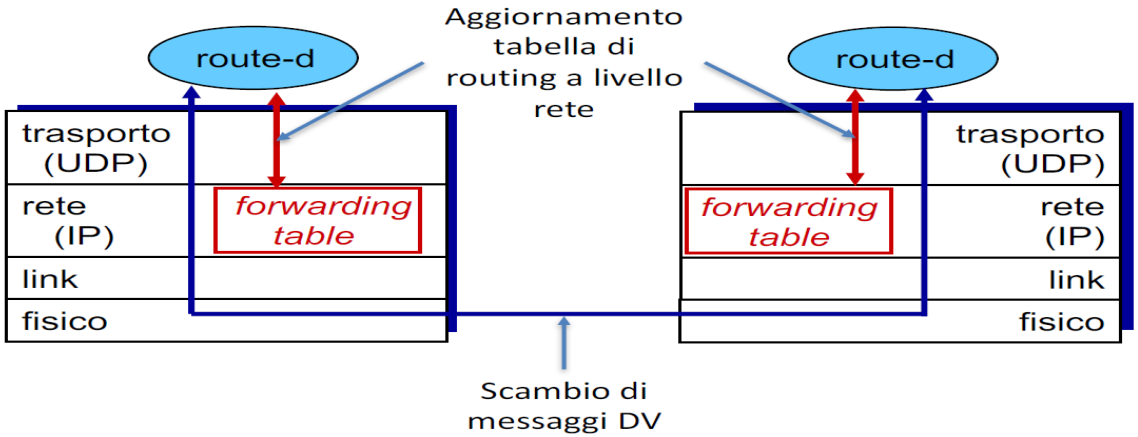
- gli advertisement sono incapsulati in pacchetti UDP
  - un RIP message (advertisement) trasmesso su UDP (port 520)
  - non richiesta trasmissione affidabile, poichè gli updates sono inviati periodicamente.



RIP V1: DEMONE SU UDP

Domanda spontanea:

- ma non avevamo detto che i routers implementavano solo i primi 3 livelli dello stack TCP/IP?
- consideriamo un router come una "macchina astratta" per l'inoltro dei messaggi (data plane)
  - quella macchina astratta ha bisogno solo dei primi 3 livelli
  - però sullo stesso hardware su cui è implementato il router si possono implementare altre funzionalità
  - posso avere dei demoni a livello applicativo per supportare i protocolli di routing (control plane)



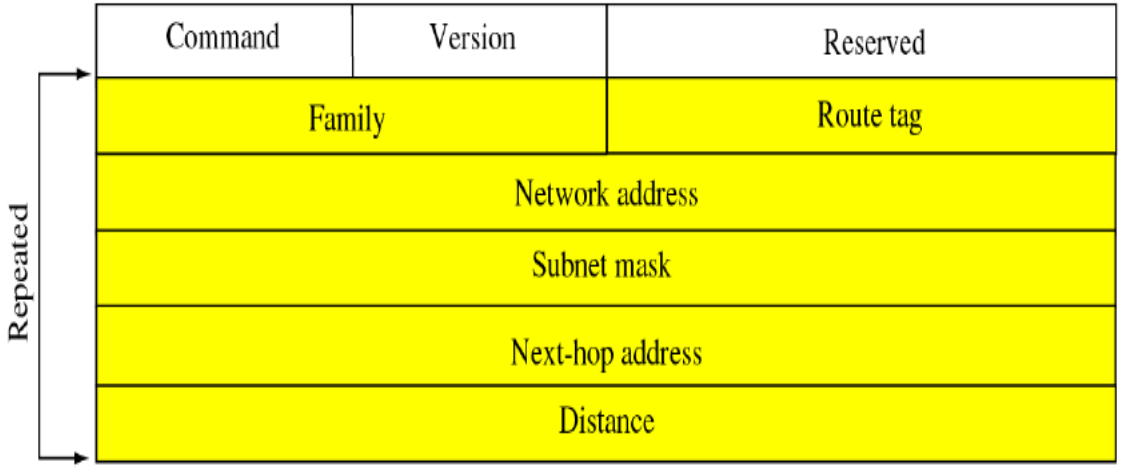
RIP V1: LIMITAZIONI

- hop count è una metrica troppo semplice!
  - considerare informazioni più interessanti
    - lunghezza della coda ad ogni hop
    - ritardo di attraversamento
    - percentuale d'errore del collegamento
    - etc…
- funziona per reti medio-piccole
  - due nodi possono distare al massimo 15 nodi
- convergenza lenta dovuta ai lunghi timer
- superato con l'introduzione di CIDR

RIP V2

- standardizzato in RFC 1723
- estende le funzionalità di RIP V1
  - supporta coesistenza di router di diverso tipo nelle stessa subnet
  - info sulla connettività
    - indicazione esplicita del next hop
    - route tag: permette di marchiare le rotte secondo la provenienza (scoperte da RIP, scoperta da altro IGP, indicate da EGP, ecc)
  - classless routing! supporta subnetting e supernetting (CIDR)
    - subnet mask
  - supporta autenticazione!
  - multicasting: usa l'indirizzo 224.0.0.9 come destinazione dei pacchetti
- backward compatibile con RIP v1
- stessa metrica di RIP v1
  - intero da 1 a 16
- usa i bytes "riservati" nel pacchetto RIP v1per implementare nuove funzionalità

RIP V2: FORMATO DEI PACCHETTI



- Standardizzato in RFC 1723
- Funzionalità aggiuntive
  - Info sulla connettività
    - Indicazione esplicita del next hop
    - route tag: permette di marchiare le rotte secondo la provenienza (scoperte da RIP, scoperta da altro IGP, indicate da EGP, ecc)
  - Autenticazione
  - Classless routing (subnet mask)!
  - Distance: come in RIPV1

PROTOCOLLI DI ROUTING

In questo corso analizzeremo

- RIP – Routing Internet Protocol
- OSPF – Open Short Path First

LINK STATE: RIASSUNTO IN BREVE

- ogni router deve:
  - scoprire i propri vicini
  - effettuare una misura del "costo" del link che lo connette al nodo vicino
    - generalmente delay, calcolato mediante un messaggio ICMP
    - ma può anche essere la banda, o altro
  - costruire un pacchetto che contenga l'informazione sui link a cui è connesso
  - inviare il pacchetto agli altri routers

- quando un router riceve un pacchetto:
  - controlla se il pacchetti è nuovo oppure una copia,
    - se non è nuovo, lo distrugge per evitare duplicazioni e di sprecare banda di rete
    - se è nuovo, lo invia su tutti i link, eccetto quello da cui proviene
  - mediante la valutazione di ogni pacchetto ricevuto, il router ricostruisce dinamicamente la topologia della rete
  - una volta che la topologia è stata scoperta, calcolo dello shortest path via Dijkstra e costruzione delle routing tables.

"REAL LIFE" PROBLEMS

- necessario gestire tutti i tipi di fallimento
  - crash dei link e perdita dei pacchetti nel flooding
    - numeri di sequenza, "età" dei pacchetti
  - notifiche inconsistenti
    - modifica di stato mentre il protocollo è in esecuzione
  - scalabilità: problematica per link state
- link state Internet protocols:
  - OSPF
  - OSPF-2
  - IS-IS (Intermediate System – Intermediate System)

OPEN SHORTEST PATH FIRST

- RFC 1247, 1583, 2328
- protocollo Link State
  - esecuzione algoritmo Dijkstra in ogni nodo
- supporta routing gerarchico
  - per aumentare la scalabilità
- permette di utilizzare metrica generica
  - costo di attraversamento di interfaccia configurabile dall'amministratore
    - link cost= 1, minimum hop routing
    - link cost inversamente proporzionale alla capacità dei link
      - disincentivare l'uso di low-bandwidth links
  - OSFP fornisce meccanismi per determinare i cammini di costo minimo per un certo insieme di costi sui link

OPEN SHORTEST PATH FIRST

- trasportato direttamente su IP (Protocol = 89)
  - deve implementare funzioni livello di trasporto
    - messaggi ACK
- supporta autenticazione
- molte tipologie di messaggi scambiati
- supporta percorsi multipli verso la destinazione
  - rotte con la stessa "lunghezza" vengono usate per bilanciare il carico
- permette di usare metriche diverse in aree diverse

OPEN SHORTEST PATH FIRST: ROUTING GERARCHICO

- una delle funzionalità più interessanti di OSPF è la possibilità di dividere una rete intradominio (un sistema autonomo) in sottodomini, indicati come aree

- in reti di grandi dimensioni
  - il numero delle entrate nei database dei diversi nodi cresce
  - il tempo per il calcolo dello shortest path aumenta
  - il traffico per spedire le notifiche relative ai link aumenta

- OSPF risolve il problema usando un routing gerarchico:
  - rete organizzata in aree corrispondenti a porzioni indipendenti della rete
    - ogni area corrisponde ad un gruppo di reti contigue
      - un pacchetto può passare da una qualsiasi rete ad un altra senza uscire dall'area

- aumento della scalabilità:
  - database disgiunti per le varie aree
  - meccanismi di flooding indipendenti

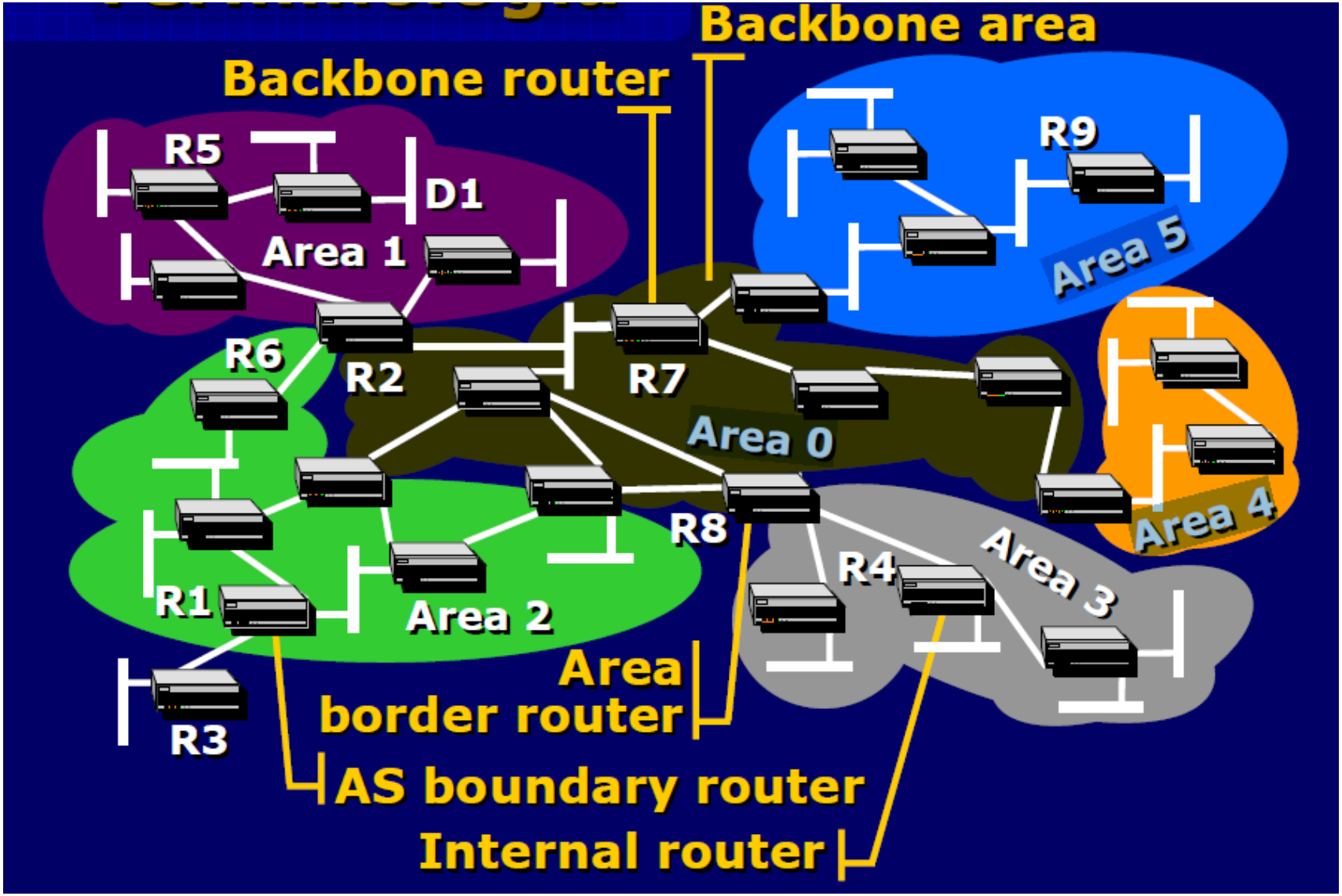
OSPF: STABILIRE IL NUMERO DI AREE

- le aree vengono configurate dall'amministratore di sistema
- la decisione di definire più aree di routing dipende da:
  - dal numero di routers e dalla complessità della rete
  - dalla capacità computazionale dei router
  - dal fatto che l'ente che gestisce l'autonomous system sia o meno in larga espansione
    - previsione esigenze future
- idea generale:
  - tante aree disgiunte connesse ad un'area di transito (backbone) che le connette

OSPF: CLASSIFICAZIONE ROUTERS

- Backbone area: area di transito tra le altre aree
  - interfaccia un'area verso le altre, esponendo una astrazione della topologia
- Backbone router:
  - router che si trova nel backbone
- Internal Router
  - router che fa parte di una sola area
- Area Border Router
  - router che si affaccia su più aree, di cui una è la backbone area
    - garantire che il percorso di un pacchetto tra aree transiti necessariamente nel backbone prima di entrare nell'area di destinazione
  - esegue una copia dell'algoritmo per ogni area
  - si occupa della propagazione delle informazioni tra più aree
- AS boundary router: diffonde informazioni al di fuori dell'autonomous system

OSPF: ESEMPIO DI CONFIGURAZIONE



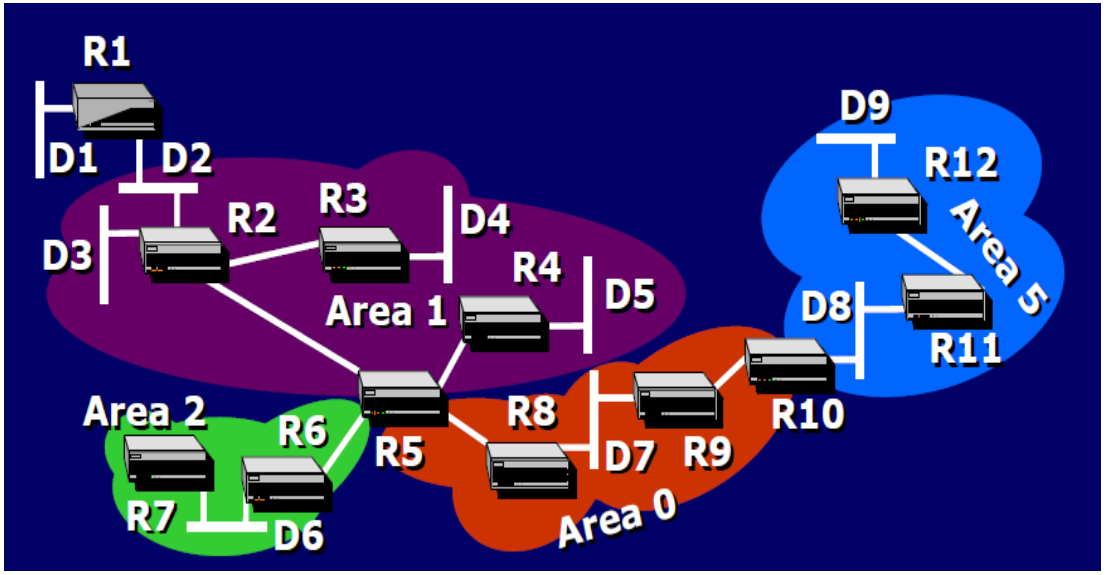
OSPF: INFORMAZIONI TOPOLOGICHE

- LSA (Link State advertisement) contengono informazioni topologiche
  - sofisticato meccanismo di propagazione : selective flooding
- propagazione interna: gli LSA generati da un router in un'area sono notificati a tutti i router di quell'area
  - tutti i router di un'area posseggono le stesse informazioni relative alla topologia di quell'area
- area border routers
  - diffondono in un'area un riassunto delle informazioni raccolte nelle altre aree su cui si affaccia
  - informazioni provenienti dall'esterno dell'area consentono all'internal router di scegliere un punto di uscita ottimale
- vantaggi: scalabilità, minor quantità di informazione da
  - memorizzare
  - comunicare
  - elaborare

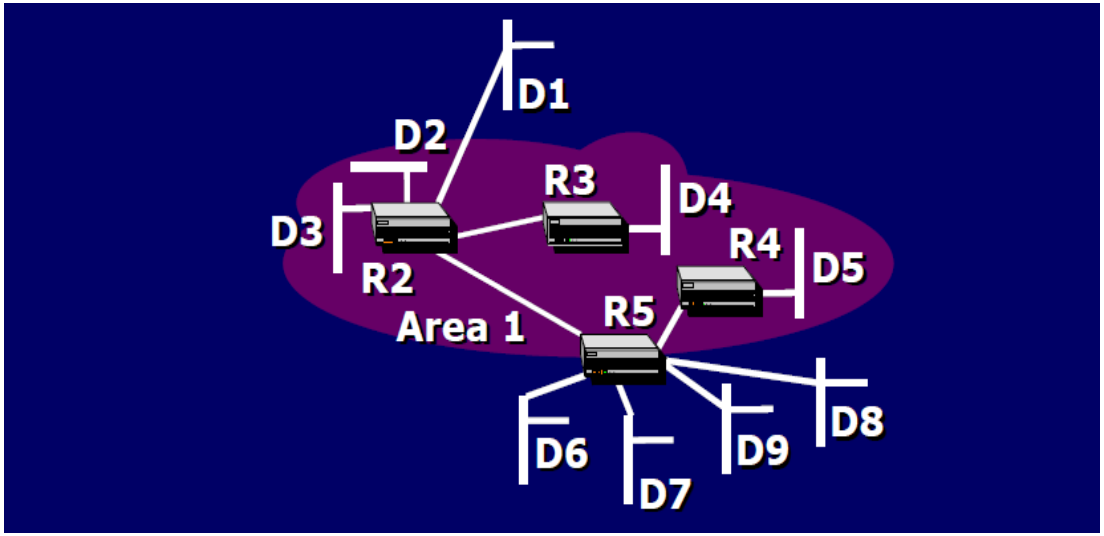
OSPF: AGGREGAZIONE DEGLI INDIRIZZI

- area border routers sono configurati per aggregare gli indirizzi
  - all'esterno dell'area sono diffusi address summaries
  - ad esempio:
    - un'area contiene indirizzi di forma 5.12.*.* e 5.7.*.*
    - l'area border router dichiara di essere collegato a 5.*.*.
- eliminazione del dettaglio
  - presenza di router e collegamenti in un'area non sono rilevanti all'esterno
- riduzione della dimensione delle tabelle di routing dei routers esterni
- Il flooding è limitato entro ciascuna area e entro il backbone, quindi la suddivisione limita fortemente il numero di messaggi

OSPF: AGGREGAZIONE DEGLI INDIRIZZI



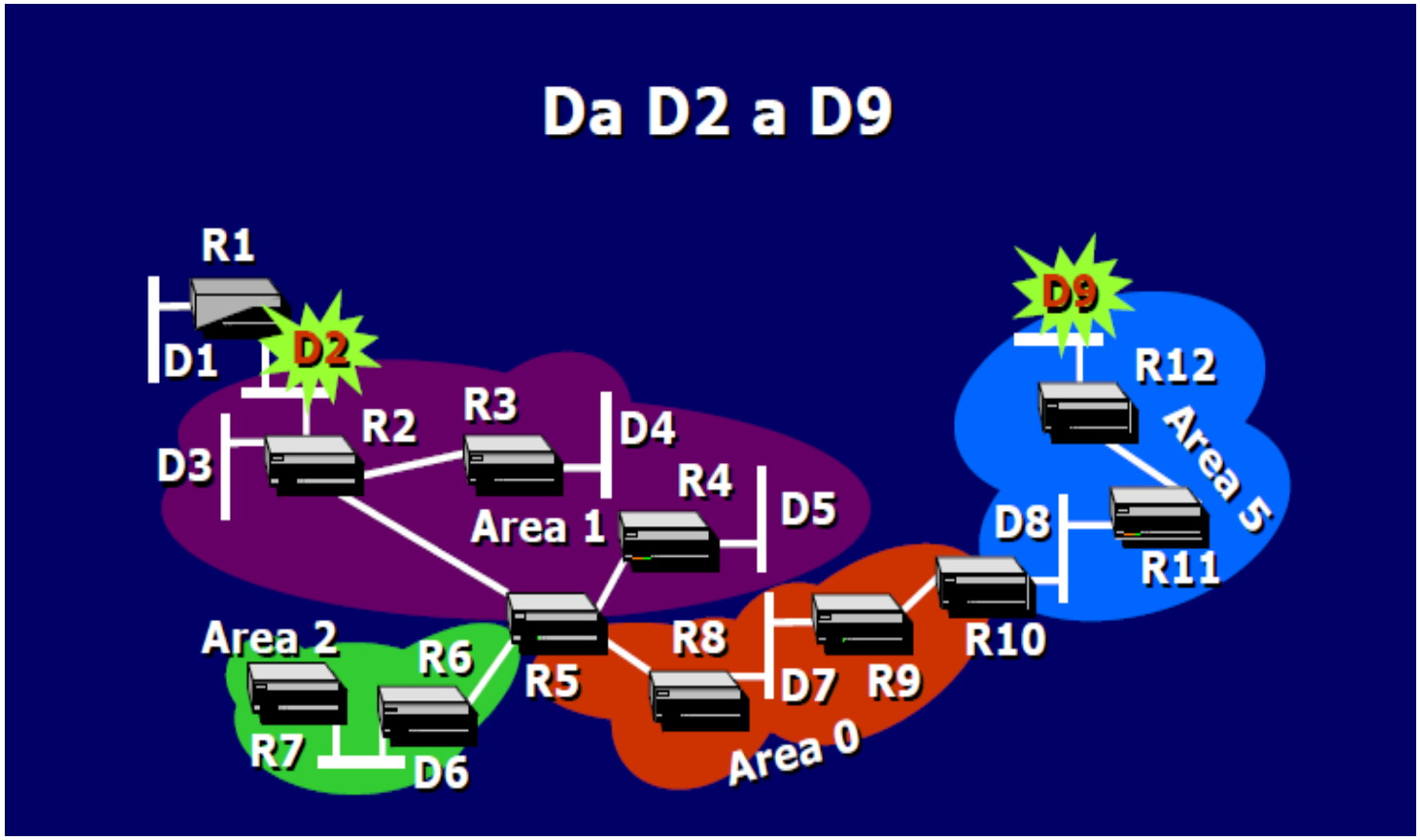
Visione della rete da parte dei routers dell'area 1



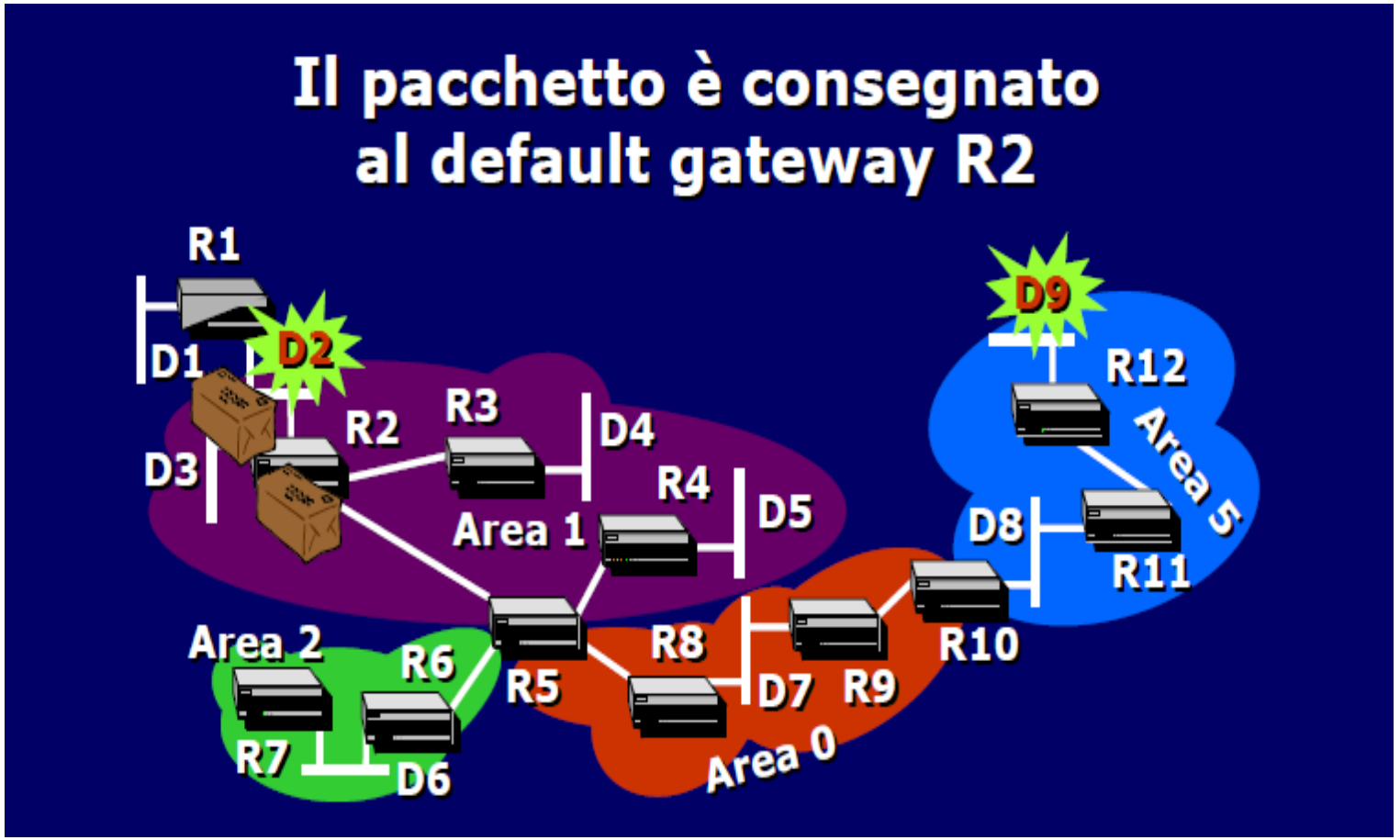
IL CAMMINO DI UN PACCHETTTO

- Il cammino di un pacchetto tra due nodi appartenenti a due aree che non siano il backbone viene quindi suddiviso in tre tronchi:
  - il cammino intra-area dal nodo di partenza sino al router di confine della sua area;
  - il cammino entro il backbone tra l'area di partenza e l'area di destinazione;
  - il cammino intra-area nell'area di destinazione.
- I router di confine scambiano tra di loro informazioni sulla raggiungibilità delle reti interne
- Utilizzando questi dati, insieme all'informazione sui link interni al backbone, ciascun router è in grado di calcolare il prossimo hop sulla route migliore per ciascuna delle reti interne alle diverse aree.
- Il router di confine per uscire dall'area di partenza viene scelto in base alla sua distanza dall'area di destinazione.

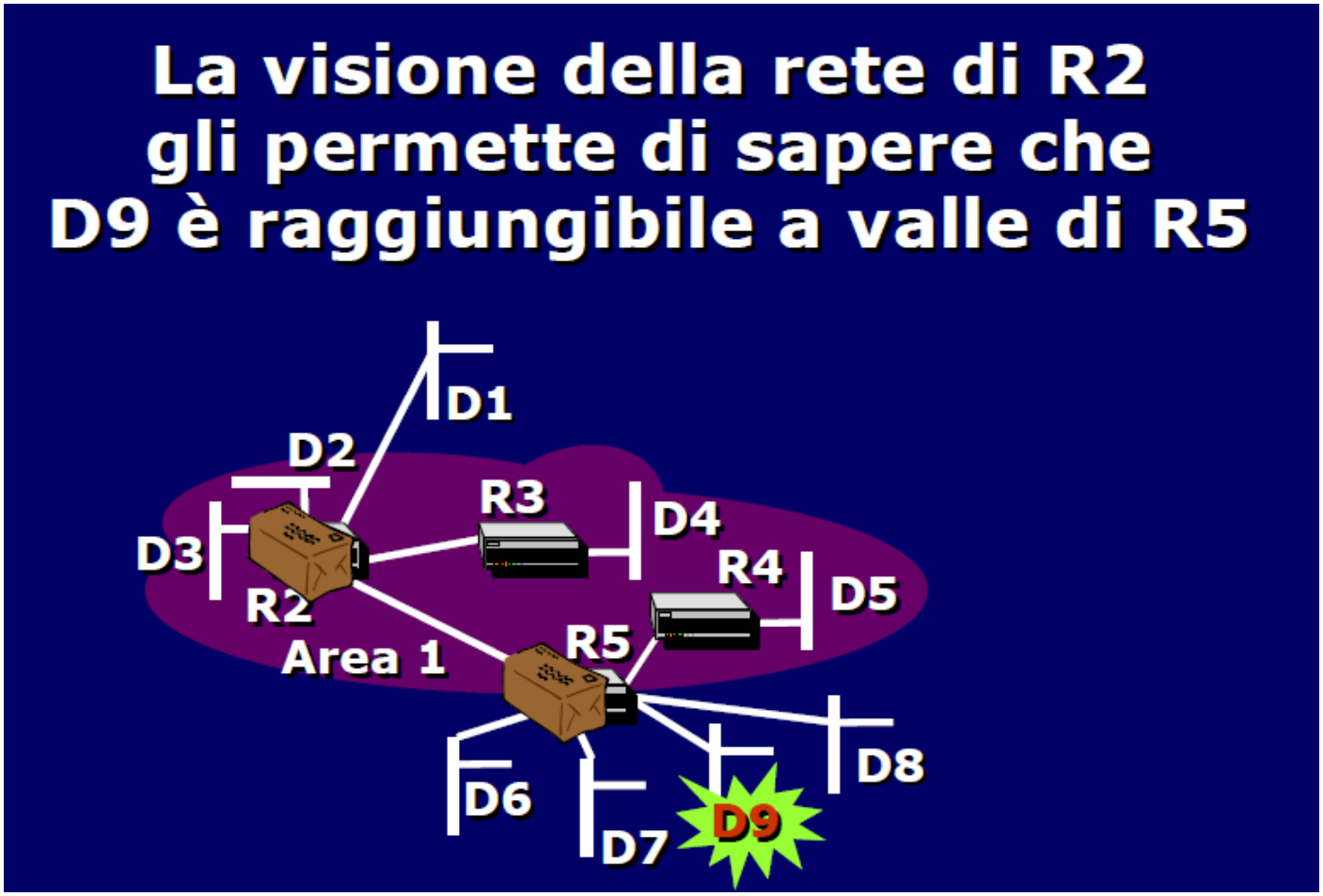
OSPF: INOLTRO DEI PACCHETTI



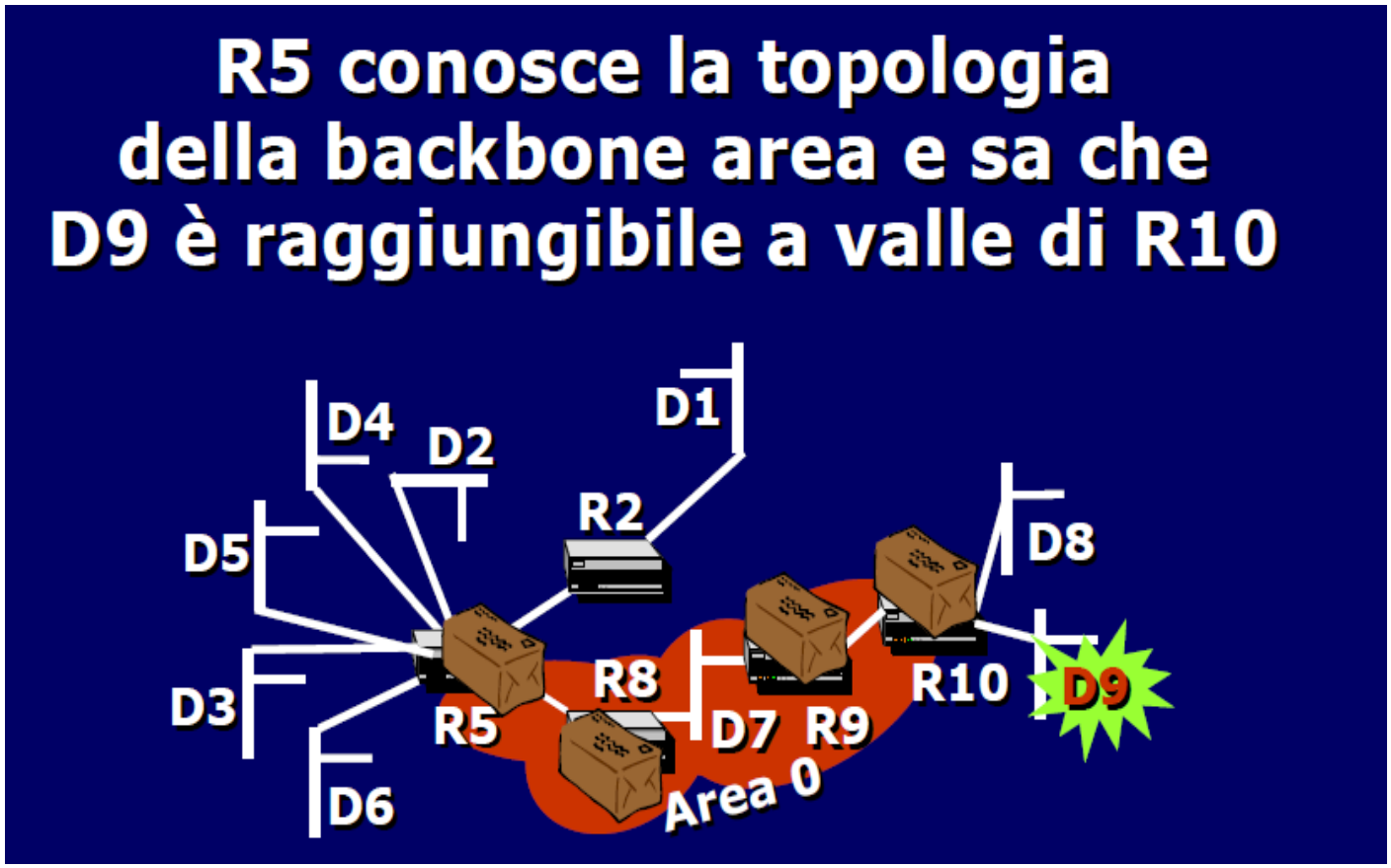
OSPF: INOLTRO DEI PACCHETTI



OSPF: INOLTRO DEI PACCHETTI

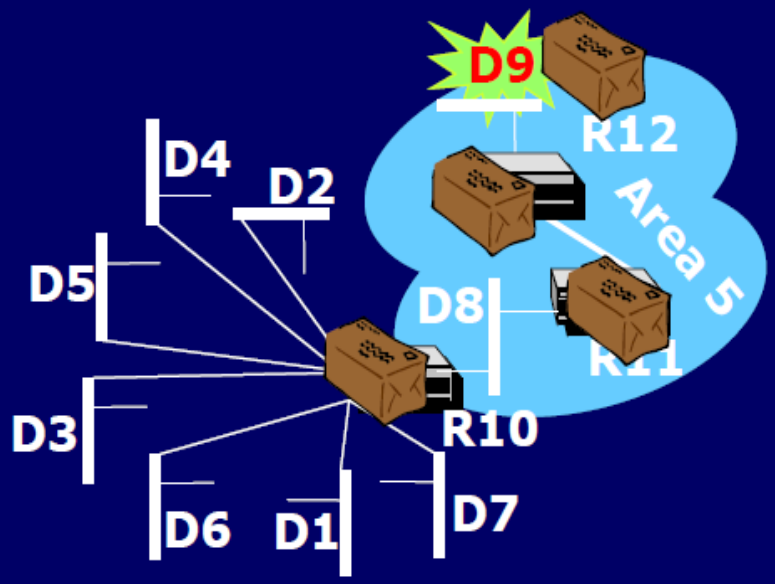


OSPF: INOLTRO DEI PACCHETTI



OSPF: INOLTRO DEI PACCHETTI

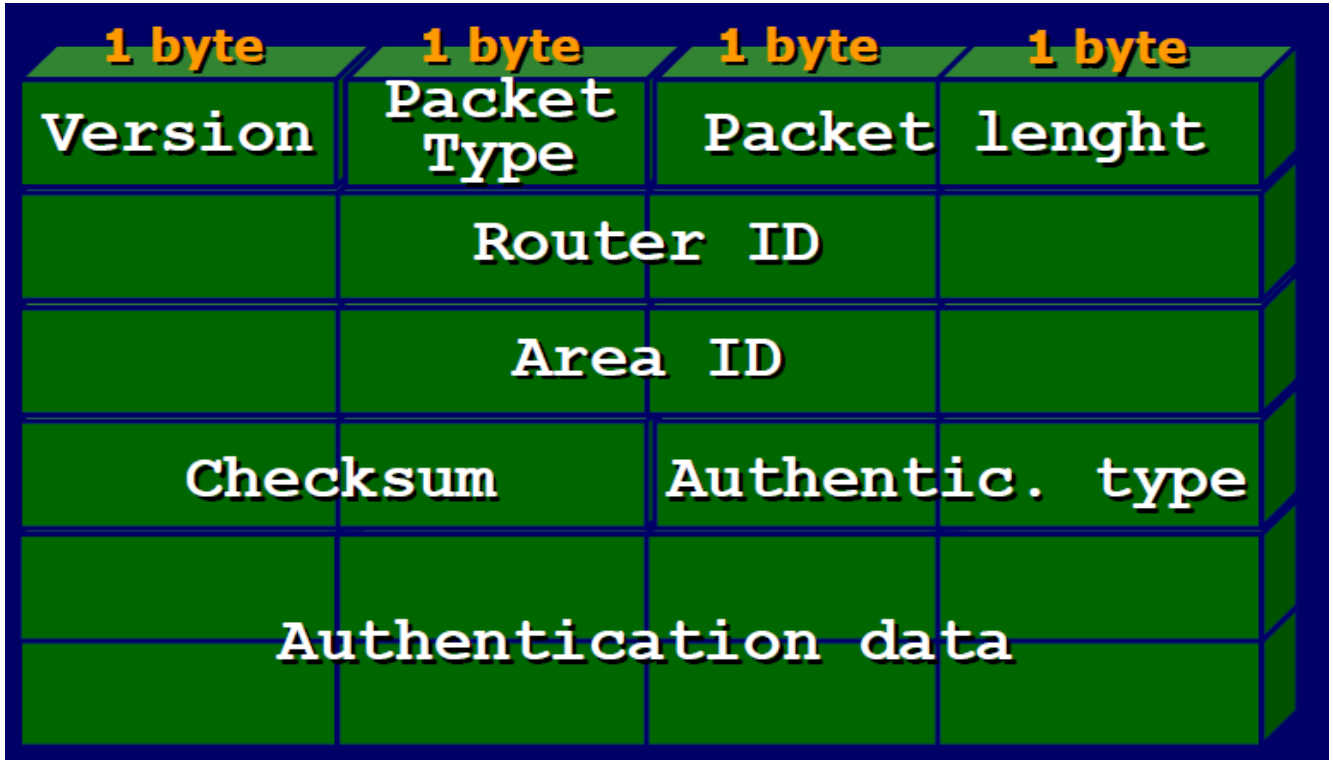
R10 conoscendo i dettagli dell'area 5 inoltra il pacchetto alla destinazione



OPEN SHORTEST PATH FIRST IN BREVE

- I pacchetti OSPF
  - Hello
  - Database description
  - Link State Request
  - Link State Update (advertisement)
  - Link State Acknolewdgement
- intestazione comune
- "imbustati" in IP

INTESTAZIONE OSPF



OPEN SHORTEST PATH FIRST: MESSAGGI

- Link failure detection
  - utilizza messaggi HELLO
    - dopo 4 HELLO intervals (40 secondi), il link viene considerato fallito
- La prima volta che due routers si contattano
  - scambiano il loro link state database
- Sincronizzazione dinamica di due link state databases
  - refresh information ogni 30 minuti
- Un router verifica un cambiamento nello stato di un link
  - trigger di un link state update ai vicini
  - calcolo dei nuovi cammini minimi
- Gli altri routers propagano l'aggiornamento a tutta le rete
  - numeri di sequenza per individuare updates ridonanti
  - conferma di updates (Link state Acknowledge)
  - ricalcolo dei cammini minimi

OPEN SHORTEST PATH FIRST IN BREVE

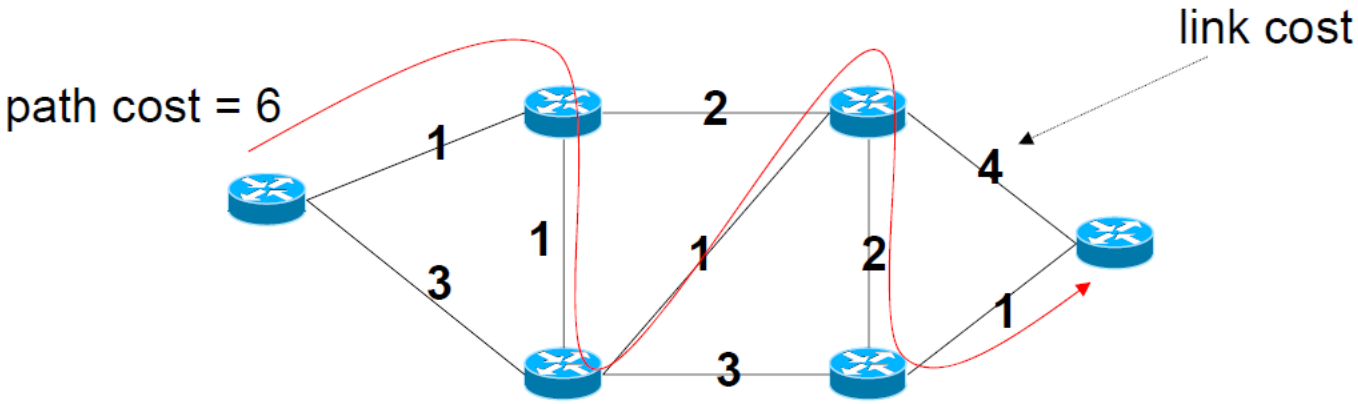
- LSA generati dai router
  - nuove copie generate a seguito di cambiamenti della rete e periodicamente
  - numeri di sequenza
- un LSA deve essere ricevuto da tutti i router in una LAN
- problema: OSPF implementato su IP (raw socket), non utilizza il protocollo IP
  - necessari timer, messaggi di acknowledgment e ritrasmissioni
  - elevata complessità
- soluzione:
  - eleggere un router (designated router) che si occupa di distribuire gli LSA
    - rappresentare l'intera rete nel protocollo OSPF, semplificandolo il protocollo
  - per questioni di tolleranza ai guasti presenza di un secondo router (backward designated router)

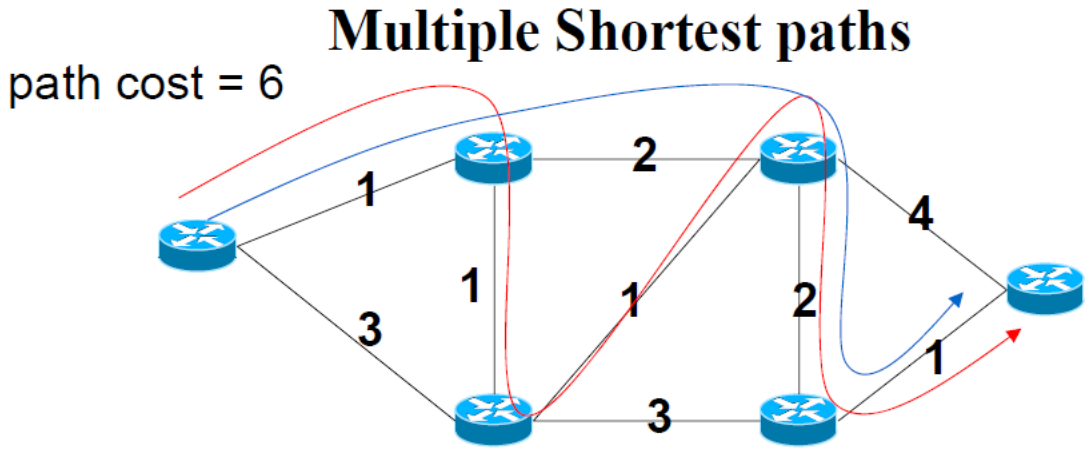
OPEN SHORTEST PATH FIRST IN BREVE

Il router designato

- è abilitato ad eseguire una operazione di broadcast o multicast, se consentito dalla LAN
  - esempio di rete di questo genere è Ethernet
- rappresentare l'intera rete nel protocollo OSPF, semplificandolo ulteriormente
- diversi tipi di LSA
- protocollo molto complesso

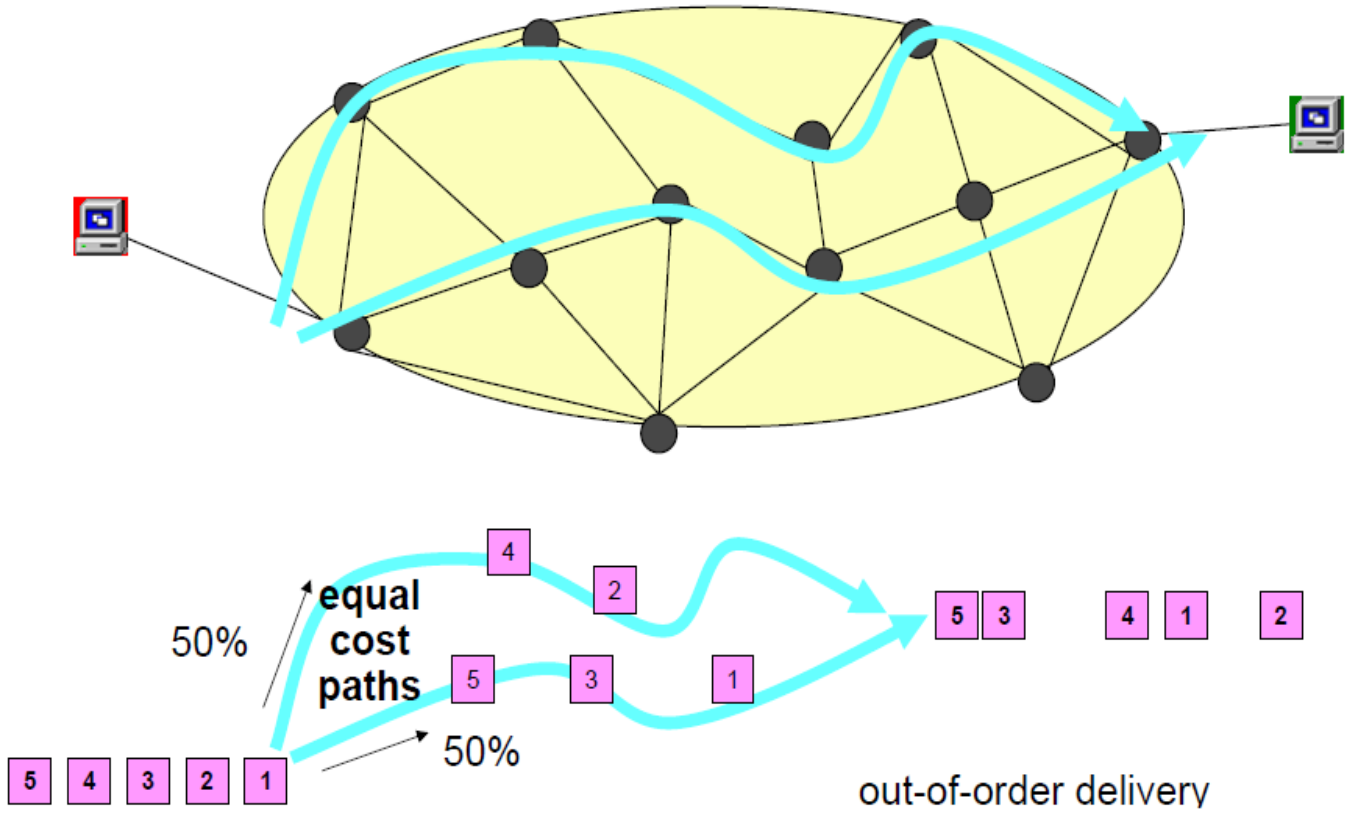
OSPF: MULTIPLE SHORTEST PATHS





Possibilità di bilanciare il carico lungo i due cammini

OSPF: BILANCIAMENTO DEL CARICO



CONFRONTO RIP- OSPF

Feature	RIP	OSPF
Algorithm	Vector-distance	Link-state
Maximum Hops	15 hops. 16 hops is considered to be infinity, implying that the destination is unreachable	Limited only by size of routing tables within routers
Subsystem Segmentation	Treats the autonomous system as a single subsystem	Breaks the autonomous system into one or more areas with two levels of routing algorithms, intra-area, and inter-area.
Metric	Destination/hop	Destination/cost/link identifier
Integrity	No authentication in RIP-1, Authentication has been added to RIP-2	Supports Authentication. Several authentication algorithms are available ranging from simple password operations to more complex cryptographic algorithms.
Complexity	Relatively Simple	More Complex. Several more PDUs and exchanges are defined in the protocol. Routing tables are large and include not only destinations, but also a tree representation of local network.
Acceptance	Widely Available, BSD routed supports RIP	Newer, published in RFCs
Route Options	Identifies a single route to a destination	Supports multiple routes to a single destination. Facilitates load-balancing traffic distribution
Types of Routes	Host, network. RIP-2 adds the ability to transfer subnetwork route entries	Host, network, and subnetwork routes