Reti di Calcolatori I

Prof. Roberto Canonico

Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione

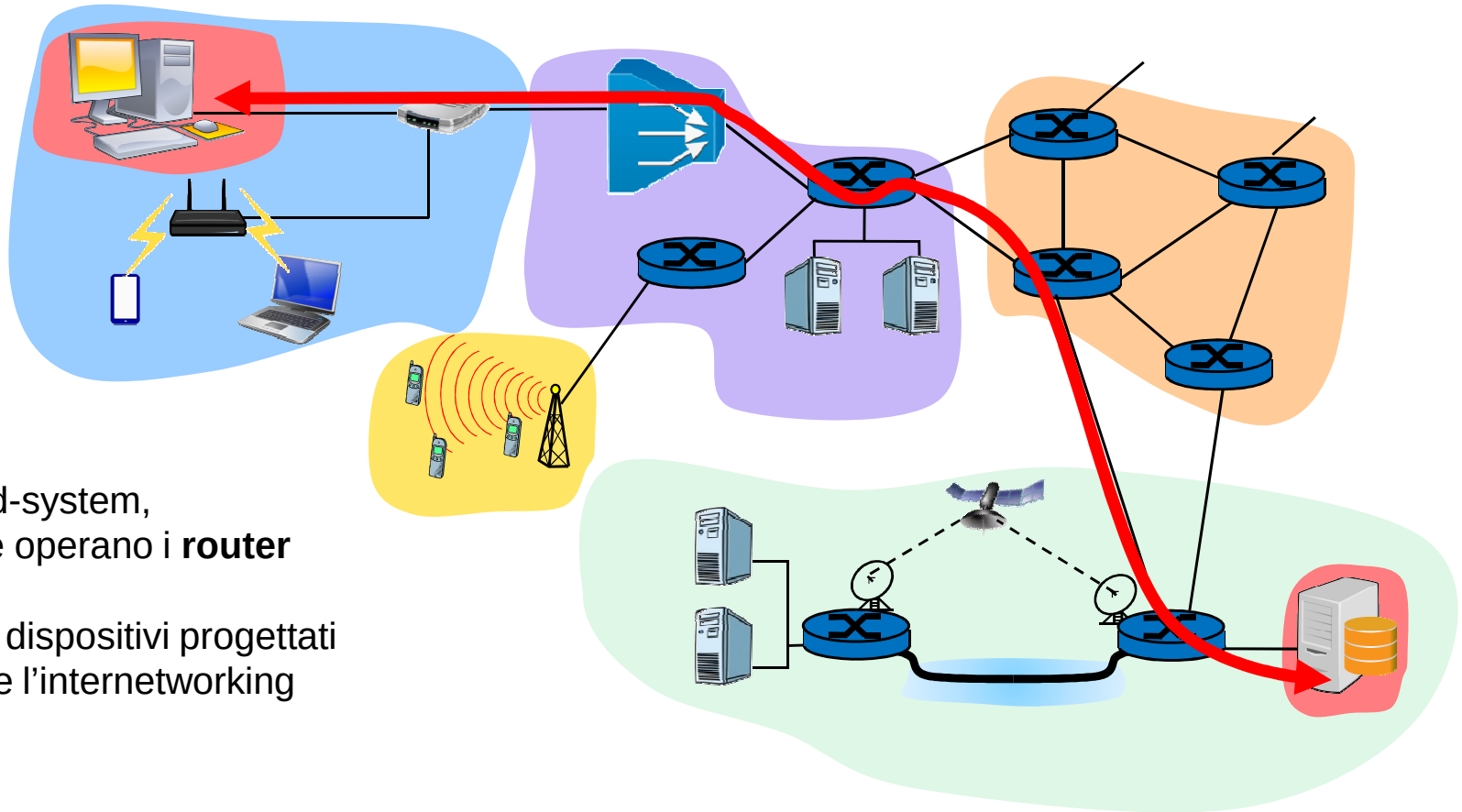
Corso di Laurea in Ingegneria delle Telecomunicazioni

Corso di Laurea in Ingegneria dell'Automazione

A.A. 2017-2018

Il compito del Livello Rete (layer 3)

In una rete di computer ottenuta attraverso la interconnessione di reti distinte (*internetwork*), il compito del ***livello rete*** è quello di definire i percorsi dei pacchetti nel loro transito da host mittente a host destinazione



Oltre agli end-system,
al livello rete operano i **router**

I router sono dispositivi progettati per realizzare l'internetworking

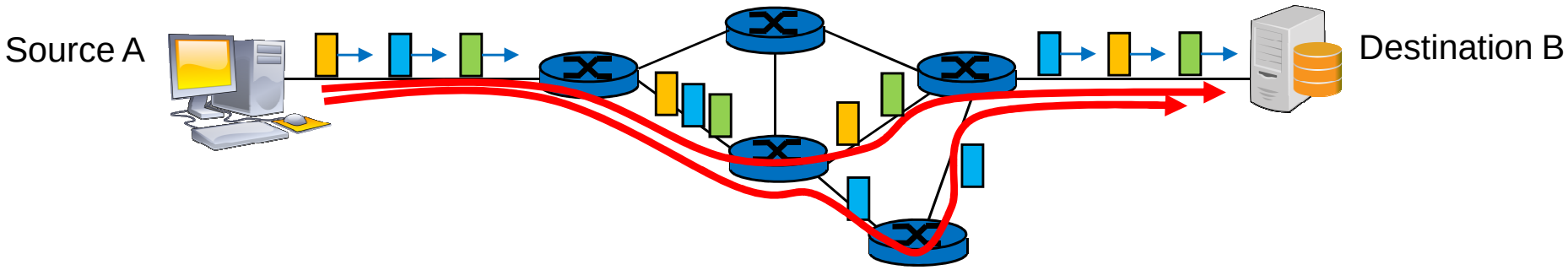
Reti di calcolatori e packet switching

- Le reti di calcolatori operano secondo il modello detto ***packet switching o commutazione di pacchetto***
- In una rete a commutazione di pacchetto l'informazione è trasmessa in ***pacchetti*** formati da una intestazione (***header***) ed un ***payload***
  - l'header contiene informazioni di controllo, tra le quali un indirizzo destinazione che serve ad identificare i terminale al quale il pacchetto deve essere consegnato

- I dispositivi intermedi che operano al livello rete funzionano in una modalità detta ***store-and-forward***
  - ogni pacchetto è ricevuto interamente, se ne controlla l'assenza di errori e se ne opera la ritrasmissione su un link di uscita
  - all'interno dei dispositivi intermedi, i pacchetti sono mantenuti in buffer di memoria gestiti come delle code

Packet switching: modello a datagram

- In una rete a commutazione di pacchetto basata sul ***modello a datagram***, ciascun pacchetto è inoltrato verso la sua destinazione indipendentemente dagli altri
  - Ogni volta che un pacchetto arriva ad un dispositivo intermedio che opera al livello rete (cioè un ***router***), il dispositivo inoltra il pacchetto verso un successivo dispositivo intermedio (o verso il destinatario finale del pacchetto, qualora esso sia direttamente raggiungibile)
  - Pacchetti inviati da un terminale A verso un terminale B in momenti successivi possono seguire percorsi differenti nella rete e, quindi, arrivare a destinazione in ordine diverso da quello con il quale sono stato trasmessi



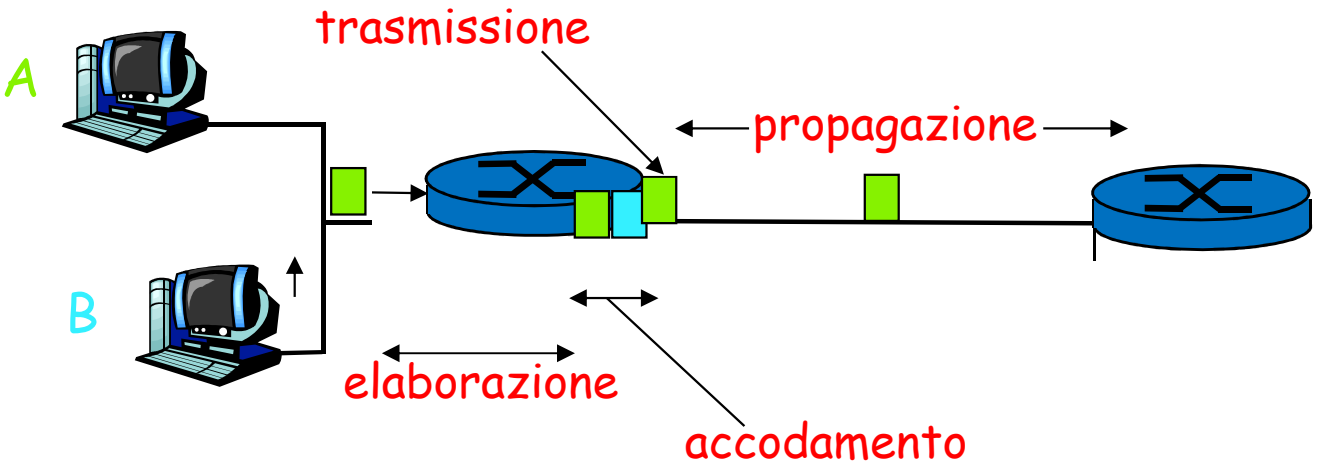
E' possibile che dei pacchetti non arrivino a destinazione

Qualità del Servizio

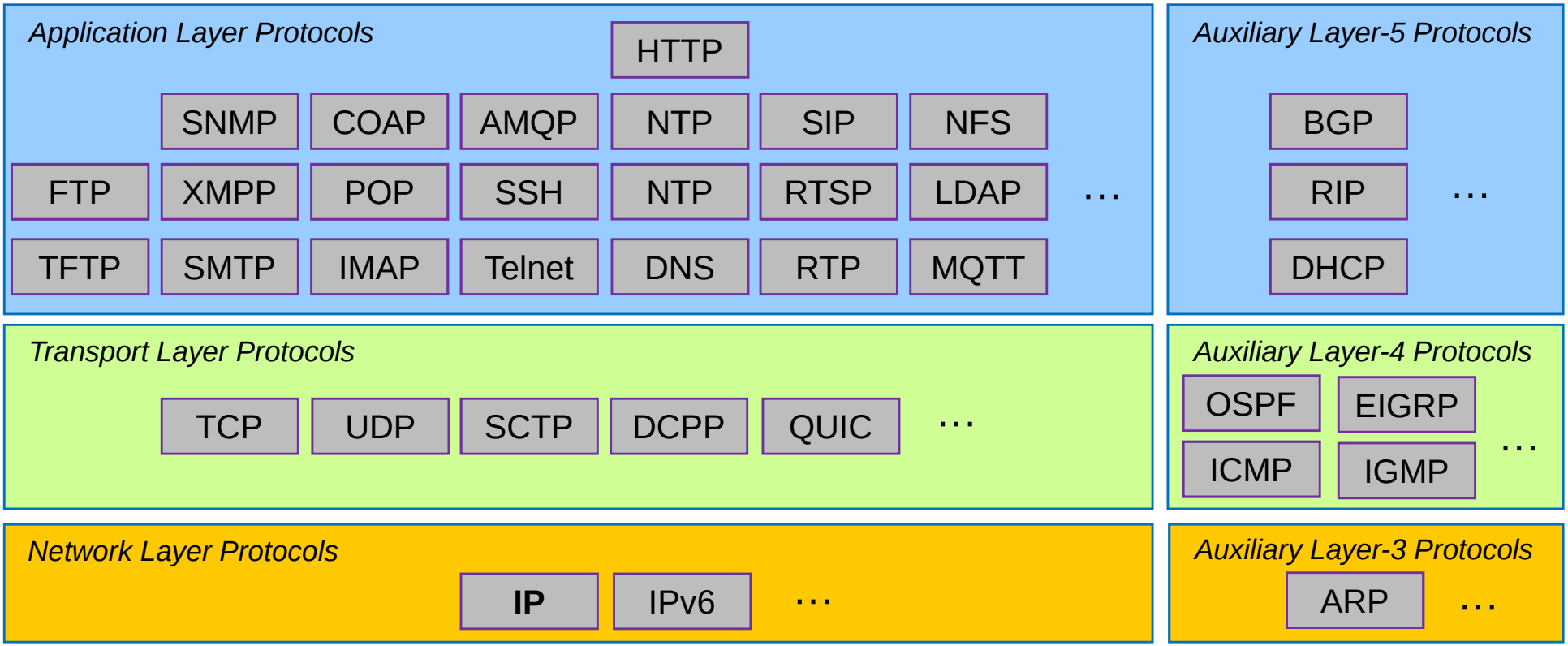
- Il servizio offerto da una rete a commutazione di pacchetto consiste nel recapitare pacchetti da un qualunque terminale mittente ad un qualunque terminale destinatario
- La **Qualità del Servizio (QoS)** di una rete a commutazione di pacchetto è misurata da una molteplicità di “indici di prestazione”
- Relativamente alla comunicazione tra due terminali collegati ad una rete, i parametri di QoS più comunemente utilizzati sono:
  - ***End-to-end delay***: ritardo nella consegna dei pacchetti [s]
  - ***Packet delay variation (PDV):*** variazione temporale del ritardo one-way (spesso anche indicata con il termine ***packet jitter)***
  - ***Throughput***: quantità di bit al secondo che la rete è in grado di trasferire tra i due terminali [b/s]
  - ***Loss-Rate***: probabilità che un pacchetto non venga consegnato a destinazione

Ritardo nelle reti a commutazione di pacchetto

- Il ritardo nella consegna di un pacchetto alla destinazione è determinato da:
  - Tempo di elaborazione nel nodo:
    - controllo di errori, determinazione link di uscita, …
  - Tempo di trasmissione su ciascun link = Lunghezza in bit / velocità in bps
  - Tempo di attesa nelle code dei router (variabile)
  - Tempo di propagazione sulle linee = lunghezza della linea / velocità del segnale



La Internet Protocol suite ed il protocollo IP



- Nella rete Internet, la funzione principale del livello rete è svolta dal protocollo IP
- La versione ancora oggi prevalentemente utilizzata è la versione 4 del protocollo IP
  - IP versione 6 è progressivamente introdotto ed utilizzato
- La caratteristica principale del protocollo IP è quella di offrire un servizio di consegna elementare e senza garanzie (*best effort*) di pacchetti
  - La semplicità rende IP adattabile ad un'ampia varietà di tecnologie di livello inferiore

Chi definisce come funziona Internet: l'IETF

- La rete Internet è una "rete di reti" basata su standard aperti
- I protocolli di comunicazione utilizzati nei livelli Rete, Trasporto ed Applicazione in Internet sono definiti da una comunità aperta di esperti detta

Internet Engineering Task Force (IETF)

- L'IETF è organizzata in gruppi di lavoro (*working groups*) che operano soprattutto tramite mailing list, aperte alla partecipazione di chiunque sia interessato
- Tre volte l'anno l'IETF organizza dei meeting plenari
  - IETF 101 a Londra – Marzo 2018
- I gruppi di lavoro si occupano ciascuno di uno specifico argomento e sono organizzati in aree (protocolli applicativi, sicurezza, ecc...)
- Ogni gruppo produce dei documenti detti RFC (*Request For Comments*) che vengono sottoposti alla IESG (*Internet Engineering Steering Group*) per il loro avanzamento a standard ufficiale
  - Prima di arrivare allo stato di RFC i documenti condivisi nei working group sono denominati *Internet Draft* (I-D)

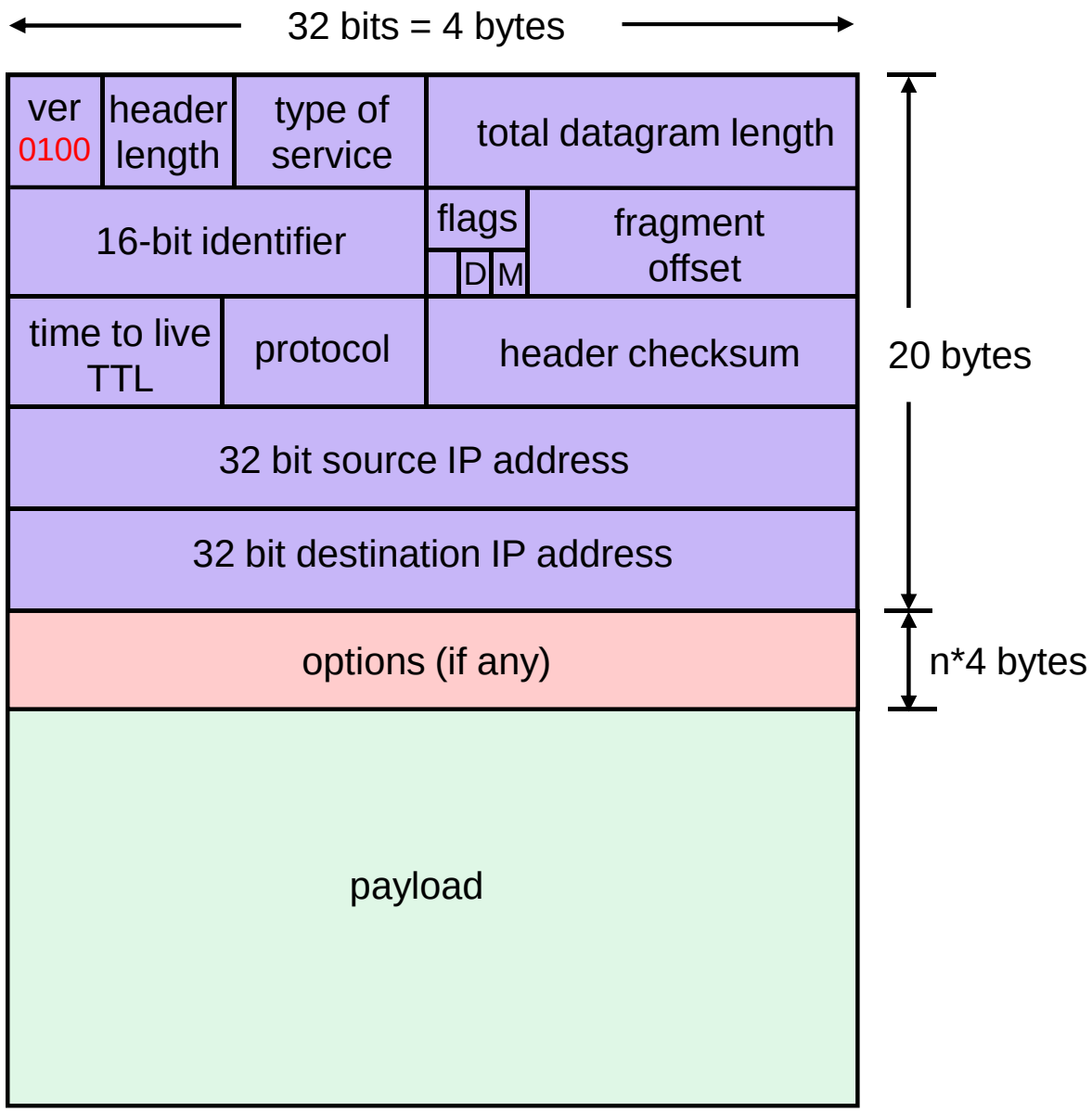
Il protocollo IP versione 4

- Nella rete Internet, la funzione principale del livello rete è svolta dal protocollo IP
  - IPv4 definito in RFC 791 (settembre 1981)
- IP realizza un servizio di consegna best-effort di pacchetti singoli (*datagram*)
- Al di sopra di IP, nello stack TCP/IP (*Internet Protocol Suite*), operano i protocolli di livello trasporto (UDP e TCP)
- Il protocollo IP gestisce indirizzamento, frammentazione, ri-assemblaggio e multiplexing dei protocolli
- E' implementato sia negli end-system (terminali) che nei router
- È responsabile dell'***instradamento*** dei pacchetti, cioè della scelta dell'interfaccia sulla quale un pacchetto deve essere trasmesso per arrivare a destinazione
- Un datagramma IPv4 può avere una dimensione massima di 65535 byte ($2^{16} – 1$) ed è costituito da un header ed un payload
- In IPv4 l'***header*** è costituito da una parte a struttura fissa (20 byte) ed una opzionale
- Il ***payload*** è creato di norma da un protocollo di trasporto (TCP o UDP)
  - In circostanze particolari, il payload di un pacchetto IP può contenere un altro pacchetto IP: *incapsulamento IP in IP*
  - Alcuni protocolli ausiliari (cioè non intesi a supportare la comunicazione di applicazioni eseguite nei terminali) inviano i loro messaggi inserendoli direttamente in un payload IP: ICMP, IGMP, OSPF

IP: servizio best effort

- IP non garantisce di prevenire:
  - pacchetti duplicati
  - consegna ritardata o fuori ordine
  - corruzione di dati
  - perdita di pacchetti
- La consegna affidabile dei messaggi alle applicazioni può avvenire grazie a meccanismi di controllo realizzati nei protocolli di livello superiore (negli end-system)
- Ogni router che riceve un pacchetto IP decide a quale altro nodo inoltrarlo, sulla base dell'indirizzo destinazione contenuto nel pacchetto, in maniera indipendente …
  - rispetto agli altri router
  - rispetto agli altri pacchetti passati in precedenza per lo stesso router
- Il protocollo IP è stato progettato per realizzare un servizio *best-effort*
- Servizio best-effort significa che la rete
  - non fornisce alcuna garanzia sulla consegna di un pacchetto
  - ma non discrimina un pacchetto rispetto ad altri
    - ***network neutrality***

Struttura di un datagram IP versione 4

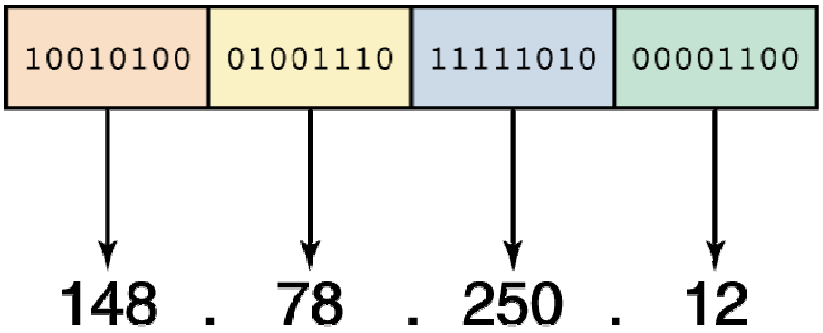


Campi dell'header IP versione 4

- In IPv4 l'***header*** è costituito da una parte a struttura fissa (20 byte) ed una opzionale di lunghezza multipla di 4 byte
- ***IP header length (4 bit):*** lunghezza dell'header, in multipli di 32 bit (max 60 byte)
- ***Type-of-Service (8 bit):*** specifica il tipo di servizio che si richiede alla rete
  - usato, in pratica, per scopi differenti
- ***Total length (16 bit):*** indica la lunghezza in byte dell'intero pacchetto (header+dati)
- ***Time-to-live TTL (8 bit):*** numero residuo di router attraversabili
  - viene decrementato di 1 da ogni router, a 0 il pacchetto viene scartato
  - serve, in caso di percorsi circolari (*loop*), ad evitare che un pacchetto resti perennemente in circolo
- ***Protocol (8 bit):*** indica il protocollo di livello superiore associato al payload
  - il valore 6 indica TCP, 17 indica UDP
  - serve al de-multiplexing dei pacchetti a destinazione
- ***Header checksum (16 bit):*** serve a verificare l'integrità dell'header IP
- ***Source IP Address (32 bit):*** indirizzo IP del nodo mittente del pacchetto
- ***Destination IP Address (32 bit):*** indirizzo IP del nodo destinatario del pacchetto
- ***Identification (16 bit), Flags (3 bit), Fragment Offset (13 bit):*** sono usati in caso di frammentazione del pacchetto da parte di un router
  - consentono al nodo destinatario di ricostruire il pacchetto originario

Indirizzi IP

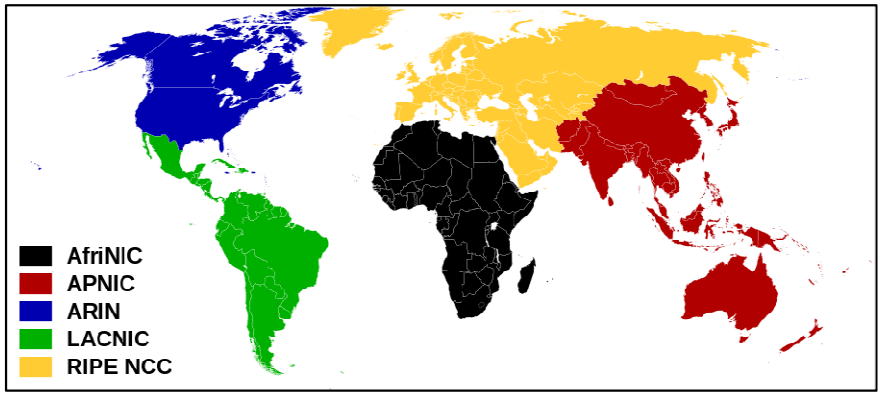
- Un indirizzo IP è una sequenza di 32 bit
- Un pacchetto IP ha, nell'header, l'indirizzo IP del mittente e quello del destinatario
- In forma testuale, per un uso da parte di un utente umano, un indirizzo IP è solitamente rappresentato nella ***notazione dotted decimal***:
  - i 32 bit sono decomposti in 4 byte, il valore di ciascuno dei quali è riportato in decimale come numero naturale tra 0 e 255
  - i quattro numeri decimali sono scritti in sequenza separati dal punto



- In una rete IP (ad esempio, la rete Internet) un indirizzo IP serve ad identificare univocamente un'interfaccia di rete di un dispositivo
  - Un end system può avere una sola interfaccia di rete, un router almeno due
  - I terminali moderni hanno diverse interfacce di rete (***multi-homed***) e dunque diversi indirizzi IP (es. interfaccia Ethernet, WiFi, Bluetooth, ecc.)

Chi assegna gli indirizzi IP

- L'assegnazione degli indirizzi IP avviene attraverso un sistema gerarchico di autorità
- Il gestore globale dell'intero spazio di indirizzamento è IANA
  - IANA - Internet Assigned Numbers Authority
  - In origine IANA era una persona: Jon Postel
- IANA dipartimento di ICANN (*Internet Corporation for Assigned Names and Numbers*)
- IANA delega la gestione degli indirizzi IP a cinque autorità regionali (RIR)
  - In Europa opera come *Regional Internet Registry* il RIPE NCC



- I registry regionali assegnano blocchi di indirizzi agli Internet Service Provider (ISP) ed alle grosse organizzazioni
- Questi, a loro volta, sono responsabili della assegnazione unica degli indirizzi di loro pertinenza ai singoli dispositivi delle proprie reti

Funzioni di un router: forwarding e routing

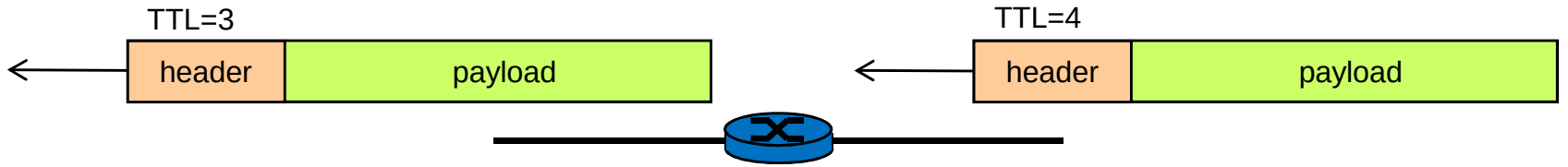
- Un router è un dispositivo dotato di più interfacce di rete che serve a collegare due o più reti tra di loro
- Un router ha il compito di inoltrare pacchetti nella rete verso la destinazione finale
- All'interno del router sono esplicate due funzioni fondamentali: **forwarding** e **routing**
- La funzione di ***forwarding*** consiste nell'inoltrare ciascun pacchetto che entra da un'interfaccia verso un'altra interfaccia
    - L'azione di forwarding effettuata dai router deve essere coordinata, in modo da far sì che un pacchetto, generato da un qualunque host mittente, possa arrivare verso un qualsiasi host destinatario
- La funzione di ***routing*** ha il compito di determinare i percorsi (*path*)
- Le due funzioni sono svolte contemporaneamente da due distinte sezioni del router:

Forwarding:	funzione esplicata dal ***data plane***
Routing:	funzione esplicata dal ***control plane***

- Il data plane deve essere in grado di operare alla velocità dei link
    - La funzione di forwarding è tipicamente realizzata mediante hardware specializzato
- Il control plane può operare a velocità più bassa (le scelte di percorso cambiano nell'ordine dei secondi)
    - La funzione di routing è tipicamente realizzata mediante software eseguito da CPU

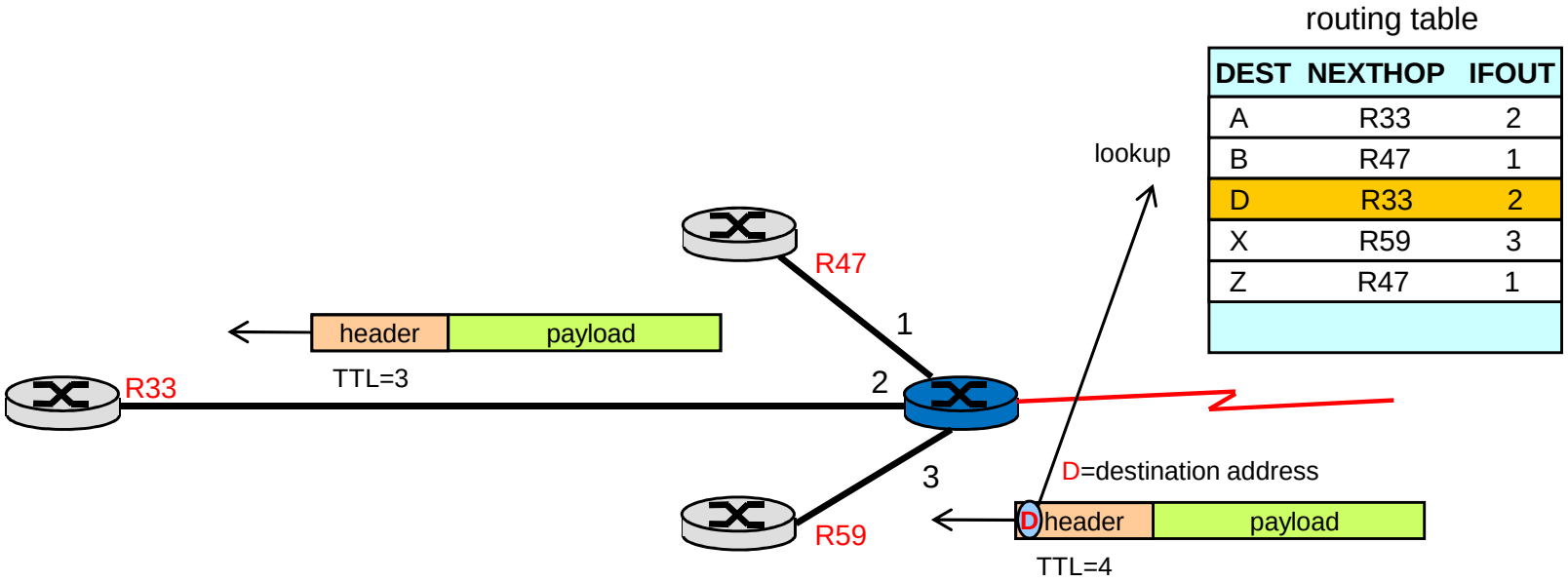
Funzioni di un router IP: forwarding (1)

- Un router IP è un dispositivo dotato di più interfacce di rete che serve a collegare due o più reti tra di loro
- A ciascuna interfaccia di un router è assegnato un indirizzo IP appartenente alla subnet associata alla rete a cui l'interfaccia si collega
- Internamente, il router identifica le proprie interfacce mediante degli identificatori locali come fa0, eth0, eth1, ecc.
- La funzione di ***forwarding*** svolta da un router IP è la seguente:
  - Per ciascun pacchetto, viene determinata l'interfaccia di uscita sulla base dell'indirizzo IP destinazione contenuto nel pacchetto
  - Prima della ritrasmissione, il campo TTL (*time-to-live*) nell'header del pacchetto inoltrato viene decrementato di 1
    - Se il TTL diventa zero, il pacchetto non è inoltrato ma viene eliminato
  - La modifica del TTL impone il ricalcolo del valore del campo *header checksum*



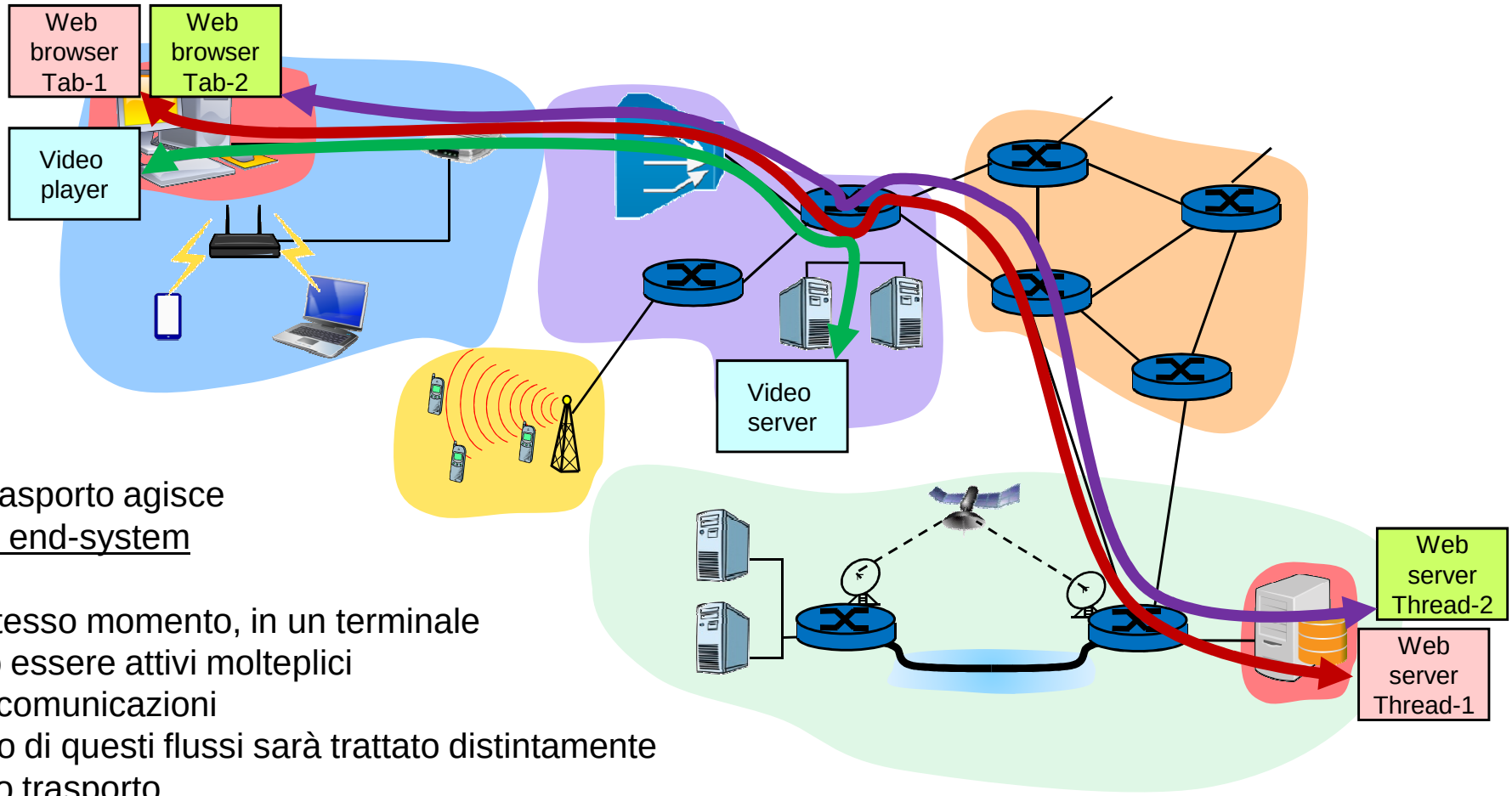
Funzioni di un router IP: forwarding (2)

- La scelta dell'interfaccia verso la quale il router realizza la ritrasmissione è determinata dall'indirizzo IP del destinatario del pacchetto
- Tale scelta è operata sulla base delle regole di instradamento contenute in una tabella: la ***tabella di routing***
- Ogni volta che il router deve inoltrare un pacchetto, viene consultata la tabella di routing per determinare l'interfaccia di uscita del pacchetto
- Il router effettua un'operazione di ricerca nella tabella (*lookup*) per determinare la regola da applicare



Il compito del Livello Trasporto (layer 4)

In una rete di computer, il compito del ***livello trasporto*** è quello di consentire alle molteplici applicazioni eseguite in un end-system lo scambio di informazioni attraverso il servizio offerto dal livello rete

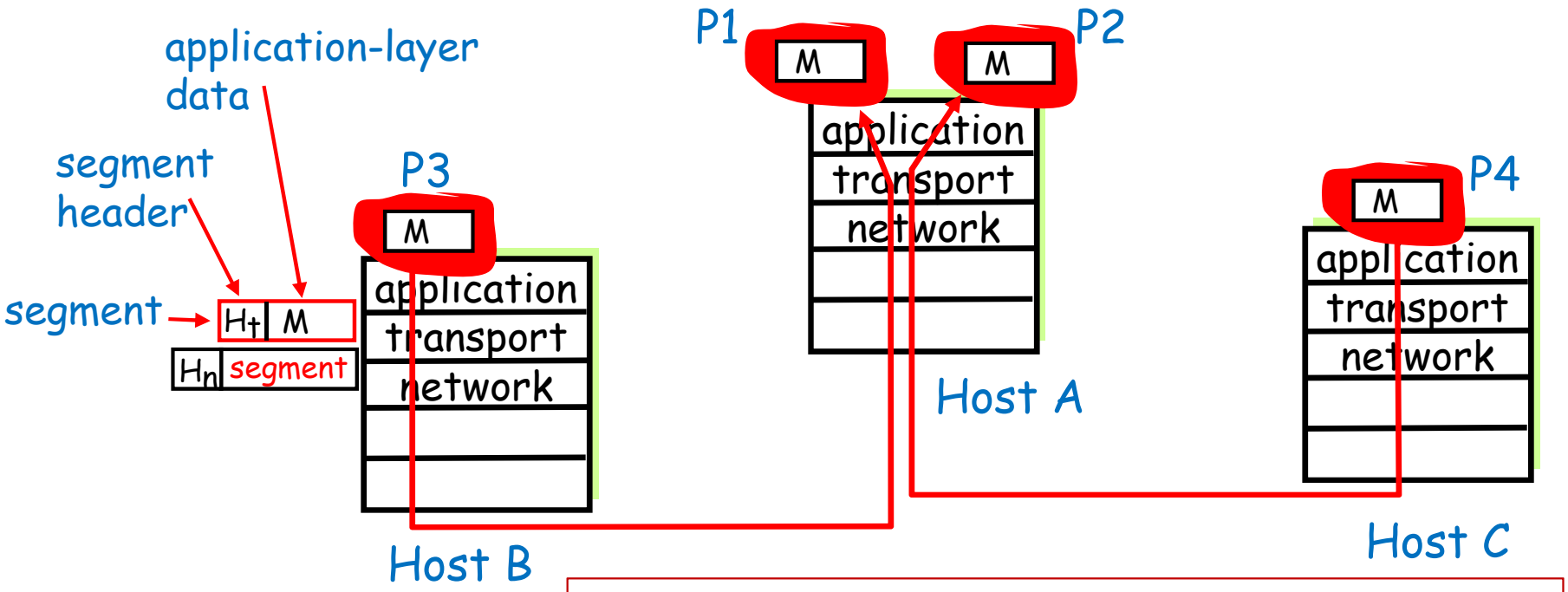


Il livello trasporto agisce
<u>solo negli end-system</u>

- In uno stesso momento, in un terminale possono essere attivi molteplici flussi di comunicazioni
- Ciascuno di questi flussi sarà trattato distintamente dal livello trasporto

Multiplexing/demultiplexing a livello trasporto

- I diversi programmi in esecuzione concorrente all'interno di un terminale (end-system) sono detti ***processi*** (nella terminologia dei sistemi operativi)
- Il compito del livello trasporto è quello di consegnare i messaggi applicativi contenuti nei pacchetti che arrivano dal livello rete al processo al quale sono destinati
- Questa funzione di smistamento (*demultiplexing*) è esplicata al livello trasporto attraverso un'informazione ausiliaria contenuta nell'header dei protocolli di livello trasporto: i ***port number***



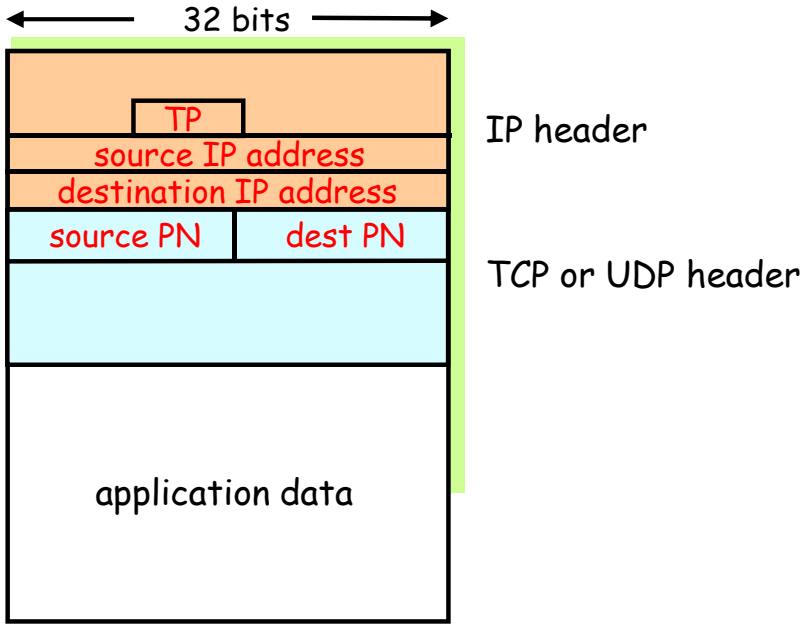
Esempio: due flussi di comunicazione contemporanei in A
1. tra i processi P1@A e P3@B
2. tra i processi P2@A e P4@B

Demultiplexing a livello trasporto: port number (1)

- Un pacchetto in arrivo ad un host è associato ad uno specifico flusso di informazione mediante la quintupla:

Transport_protocol, Source_IP_address, Source_port_number, Destination_IP_address, Destination_port_number

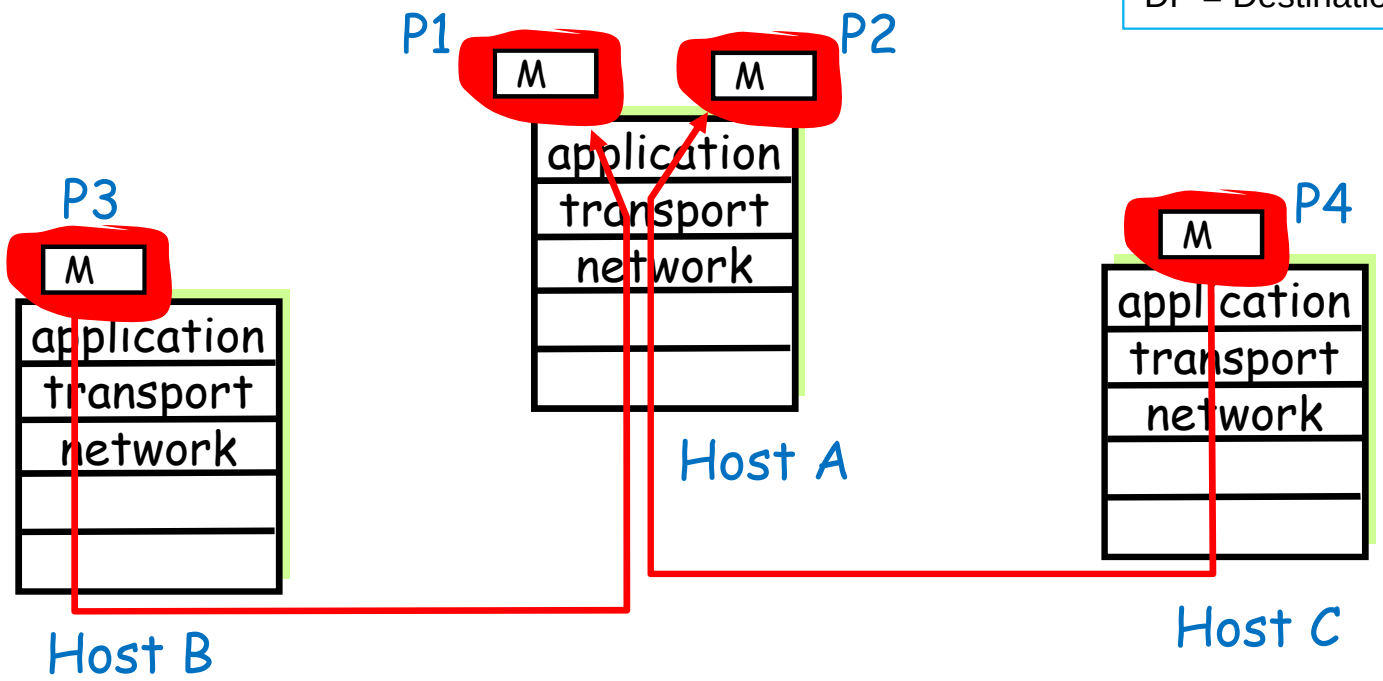
- ***Transport_protocol*** (TP) è un numero naturale rappresentato su 8 bit che identifica il protocollo di livello trasporto ed è collocato nell'header di livello rete (IP):
  - 6 per TCP, 17 per UDP (valori specificati in RFC 790)
- ***Source_IP_address*** e ***Destination_IP_address*** sono gli indirizzi IP (32 bit) del mittente e destinatario del pacchetto
- ***Source_port_number*** e ***Destination_port_number*** sono numeri naturali espressi su 16 bit che servono ad identificare il flusso di informazione e sono collocati nell'header TCP e UDP
  - Source_port number è scelto dall'host mittente
  - Destination_port_number è scelto dall'host destinatario



Demultiplexing a livello trasporto: port number (2)

Esempio: due flussi di comunicazione contemporanei in A
1. tra i processi P1@A e P3@B con TCP
2. tra i processi P2@A e P4@B con TCP

Legenda:
TP = Transport Protocol
SA = Source IP Address
DA = Destination IP Address
SP = Source Port Number
DP = Destination Port Number



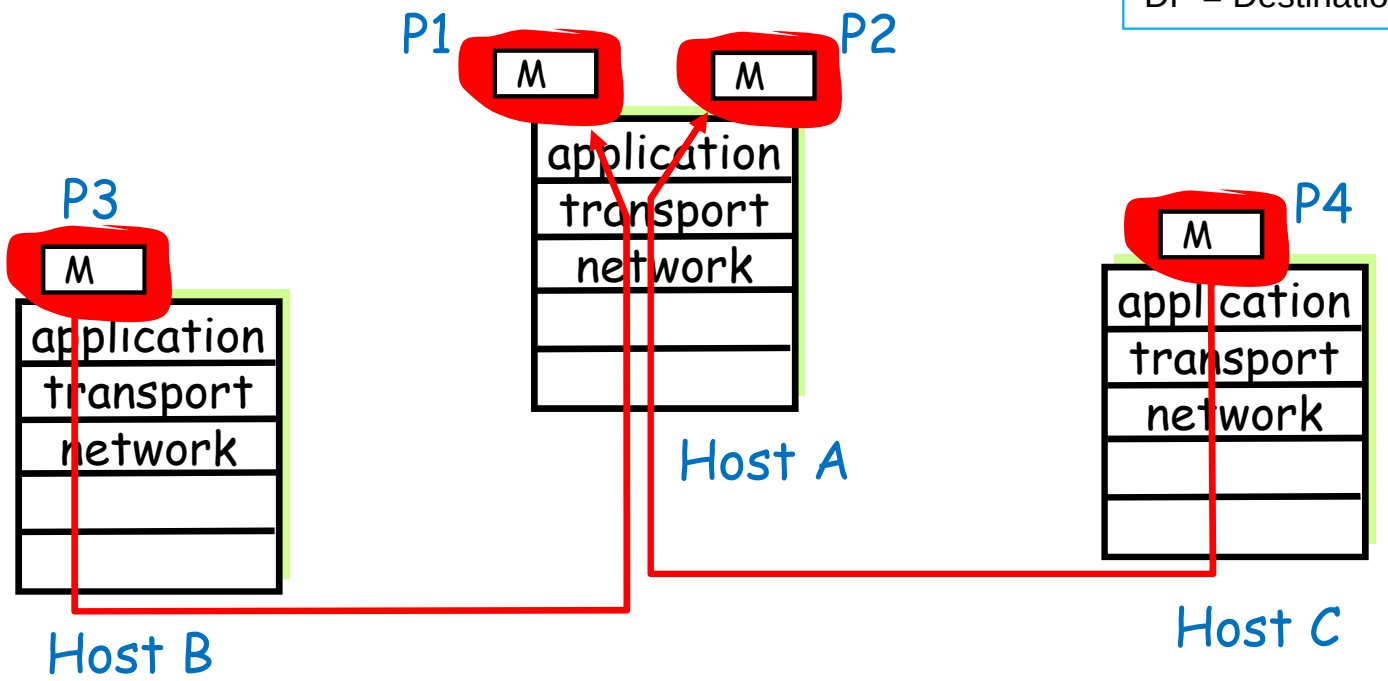
- Flusso P1@A → P3@B: TP=TCP; SA=A; DA=B; SP=80; DP=3127
- Flusso P3@B → P1@A: TP=TCP; SA=B; DA=A; SP=3127; DP=80

- Flusso P2@A → P4@C: TP=TCP; SA=A; DA=C; SP=80; DP=1984
- Flusso P4@C → P2@A: TP=TCP; SA=C; DA=A; SP=1984; DP=80

Demultiplexing a livello trasporto: port number (3)

Esempio: due flussi di comunicazione contemporanei in A
1. tra i processi P1@A e P3@B con TCP
2. tra i processi P2@A e P4@B con TCP

Legenda:
TP = Transport Protocol
SA = Source IP Address
DA = Destination IP Address
SP = Source Port Number
DP = Destination Port Number

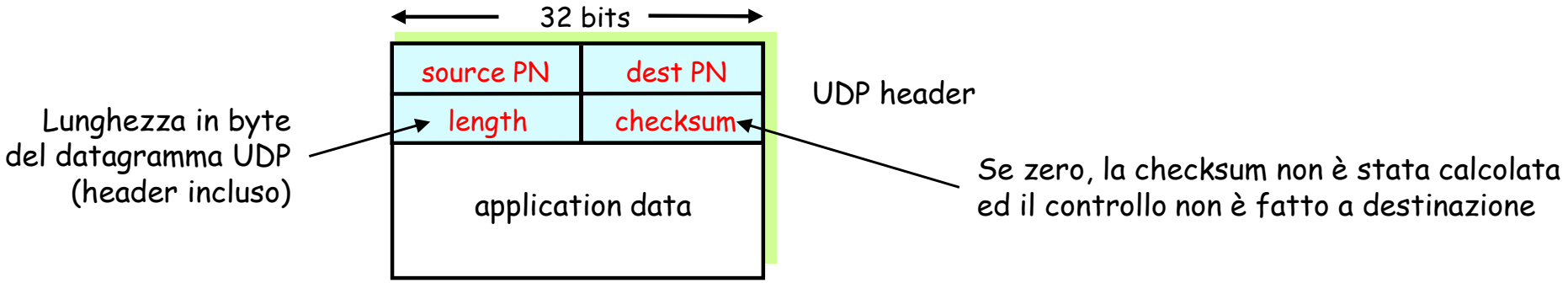


- I due host B e C possono eventualmente scegliere lo stesso valore di Port Number
- Non si genera ambiguità perche i flussi entranti in A sono distinti in base al Source IP address

- Flusso P3@B → P1@A: TP=TCP; SA=B; DA=A; SP=3127; DP=80
- Flusso P4@C → P2@A: TP=TCP; SA=C; DA=A; SP=3127; DP=80

Protocollo UDP: modello di servizio e header

- Offre alle applicazioni un servizio di consegna best-effort ma senza garanzie di messaggi
- UDP forma un datagram IP per ciascun messaggio applicativo ricevuto dall'applicazione
- Non è garantita né la consegna né il rispetto dell'ordine di sequenza
- In sostanza aggiunge al servizio di IP solo il multiplexing attraverso il port number
- Opzionalmente, UDP effettua un controllo di correttezza del datagramma ricevuto tramite una checksum; se il controllo da esito negativo, il datagramma UDP è scartato
- Le applicazioni che usano UDP devono, se lo ritengono opportuno, implementare meccanismi che implementino contromisure nel caso di perdita di datagrammi o di consegna fuori ordine
- Le applicazioni che usano UDP sono orientate a semplici interazioni richiesta-risposta

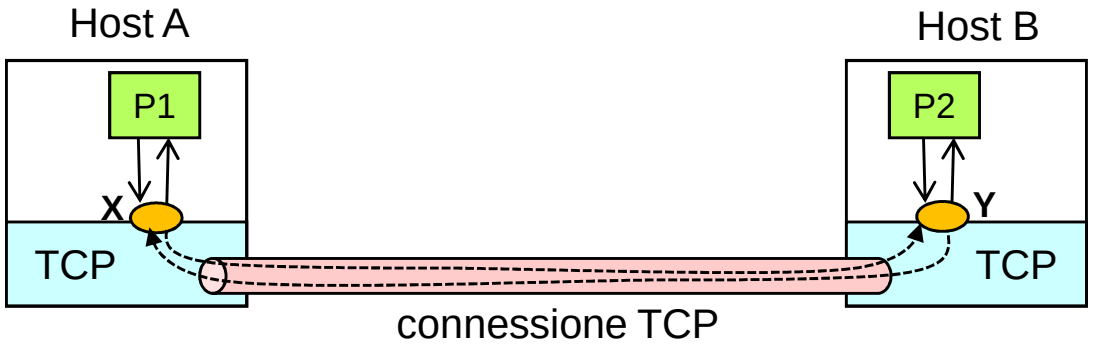


Protocollo TCP: modello di servizio

- TCP offre alle applicazioni un servizio di consegna senza perdite e con rispetto dell'ordine di uno stream di byte bidirezionale tra due endpoint
- Il servizio è di tipo ***connection-oriented***, cioè, prima di iniziare la trasmissione dei dati, occorre stabilire una associazione (*connessione*) tra i due endpoint attraverso una procedura di *connection establishment*
  - E' anche prevista una procedura di *connection teardown* per l'eliminazione di una connessione precedentemente stabilita tra due endpoint
- Una connessione TCP tra due endpoint X ed Y consente un flusso di dati bidirezionale (da X ad Y e viceversa)
- Un pacchetto è associato ad una connessione TCP mediante la quintupla:

Transport_protocol, Source_IP_address, Source_port_number, Destination_IP_address, Destination_port_number

- Negli end-system, i processi si "agganciano" agli endpoint di una connessione TCP mediante opportuni strumenti di comunicazione previsti dal sistema operativo (*socket*)

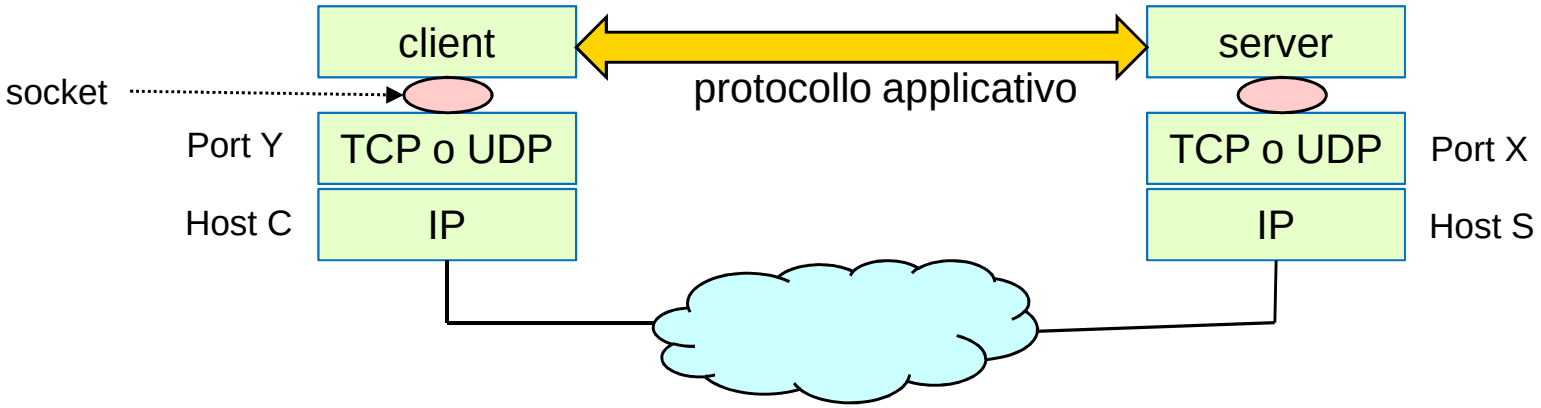


Applicazioni client/server e protocolli applicativi

- Il modello comportamentale a cui si ispira la maggioranza delle applicazioni distribuite è il ***modello client-server***:
  - un ***processo server*** è in esecuzione su un host S, e si rende disponibile alla comunicazione attraverso il protocollo TCP o UDP, ad un numero di porta X noto a priori (*well-known*) ai client
  - un ***processo client*** viene eseguito su un host C, acquisisce dal sistema operativo in esecuzione sull'host C la disponibilità all'uso del numero di porta Y e "chiede" al sistema operativo di stabilire una comunicazione tramite TCP o UDP con il processo server, del quale conosce a priori l'indirizzo IP S ed il port number X

- L'interazione tra le componenti client e server delle applicazioni è governata da ***protocolli di livello applicativo***

- Un protocollo applicativo definisce il formato dei messaggi ed i pattern di interazione



Protocolli applicativi e protocolli di livello trasporto

- La suite TCP/IP mette a disposizione delle applicazioni due servizi di livello trasporto:
  - Connectionless, senza garanzie, orientato ai messaggi: tramite UDP
  - Connection-oriented, con garanzie di consegna in ordine e senza perdite: tramite TCP

- La scelta tra TCP o UDP dipende dai requisiti dell'applicazione
  - Per applicazioni che si basano su semplici interazioni richiesta-risposta è conveniente usare il protocollo UDP
  - Per applicazioni che devono realizzare trasferimenti di dati significativi tra due processi è conveniente usare il protocollo TCP
  - Per classi di applicazioni con requisiti particolari sono stati definiti ulteriori protocolli di trasporto
    - Ad esempio, per le applicazioni che trasmettono flussi di dati time-sensitive è stato realizzato il protocollo di trasporto RTP, che è implementato "su UDP" (l'header RTP è inserito nel payload di un datagramma UDP)

- La definizione di protocolli applicativi standard consente di svincolare i servizi da una specifica implementazione
  - Ad esempio, il servizio World-Wide Web è accessibile da una molteplicità di client (*browser*)
    - Internet Explorer, Firefox, Chrome, Opera, ecc.
  - Lo stesso servizio WWW è erogabile da una molteplicità di server:
    - Apache Web Server, Microsoft IIS, NGINX, ecc.

Protocolli applicativi standard

- Uno sviluppatore che realizzi un'applicazione client/server è libero di definire ed implementare il proprio protocollo applicativo (purché sviluppi sia la componente client che quella server)

- I servizi offerti tramite Internet si attengono a protocolli applicativi standard definiti dall'IETF e definiti in specifici documenti RFC (*Request For Comments*)

Applicazione	**Protocollo applicativo**		**Protocollo di trasporto**
world-wide web	HTTP	[RFC 2068,2616]	TCP
DNS	DNS	[RFC 1034,1035]	UDP
file transfer	FTP	[RFC 959]	TCP
e-mail (invio)	SMTP	[RFC 821]	TCP
E-mail (lettura)	POP3	[RFC 1939]	TCP
remote file server	NFS	[RFC 1813]	TCP o UDP
remote terminal access	TELNET	[RFC 854]	TCP
VOIP (segnalazione)	SIP	[RFC 3261]	UDP
VOIP (flussi multimediali)	-		RTP su UDP
network management	SNMP	[RFC 3416]	UDP