

Università degli Studi di Torino
Scuola di Amministrazione Aziendale

Corso di laurea in
Management dell'informazione e della
comunicazione aziendale

*Corso di
Informatica generale (parte teorica)*

Diego Magro



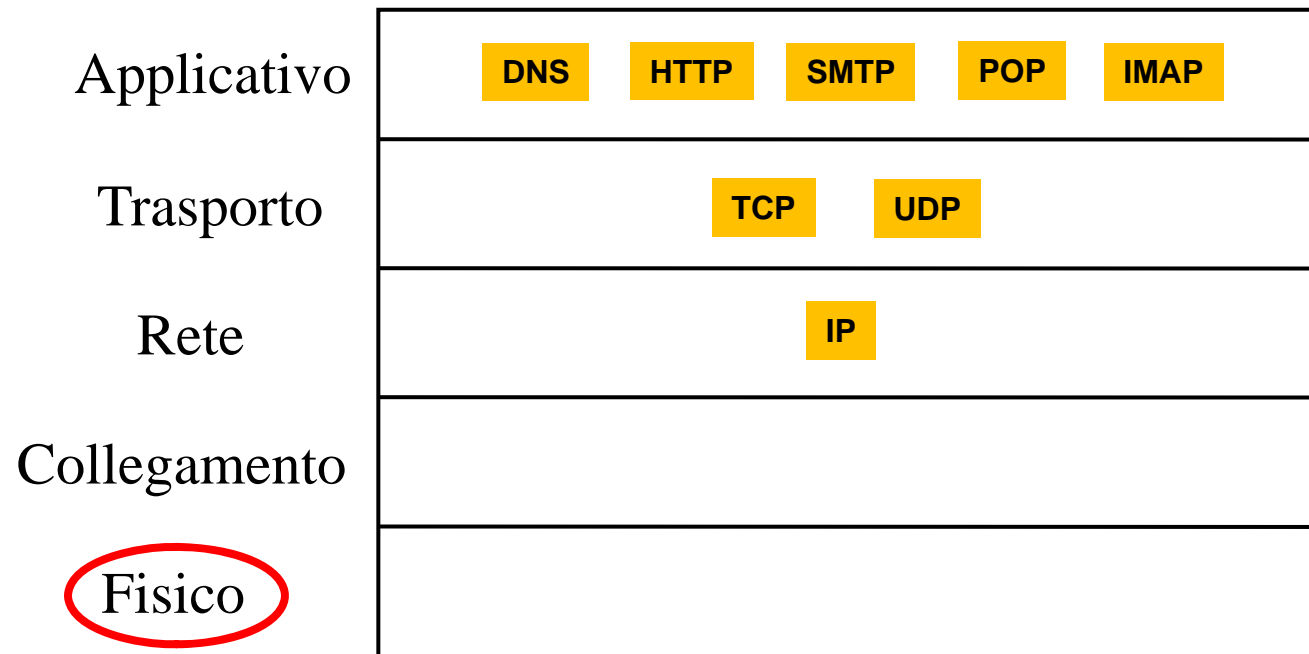
ARGOMENTI DI QUESTO GRUPPO DI LUCIDI

• RETI DI CALCOLATORI

- Livello fisico e livello di collegamento: generalità
- Livello rete: generalità e protocollo IP
- Livello di trasporto: generalità , protocolli TCP e UDP
- Livello applicativo: generalità, DNS, Web, posta elettronica

RETI DI CALCOLATORI (PARTE II)

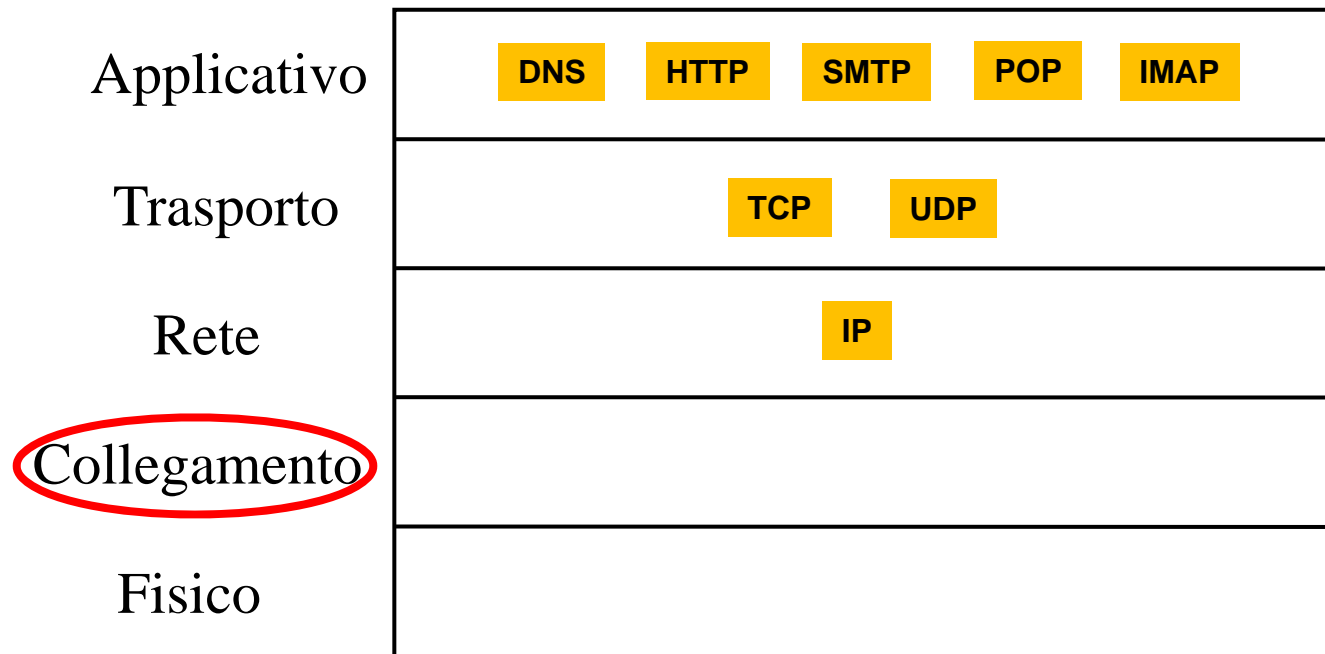
LIVELLO FISICO (I)



LIVELLO FISICO (II)

- **Tutti gli elementi di rete che comunicano fra loro devono implementare il livello fisico**
- **In particolare, fra tre dispositivi di rete citati in precedenza che, in prima battuta, potrebbero sembrare avere funzionalità molto simili, vale a dire gli hub, gli switch e i router, val la pena sottolineare che:**
 - gli hub sono dispositivi che operano a livello fisico e, pertanto, implementano solo quel livello (un hub ritrasmette il segnale in ingresso su tutte le porte di uscita, eventualmente amplificandolo)
 - gli switch sono dispositivi che operano tipicamente a livello di collegamento (anche se ve ne sono che operano anche a livelli superiori), pertanto implementano sia il livello di collegamento sia il livello fisico
 - i router sono dispositivi che operano a livello rete, pertanto implementano il livello rete, il livello di collegamento e il livello fisico

LIVELLO DI COLLEGAMENTO (I)



LIVELLO DI COLLEGAMENTO (II)

- **Principali servizi specificati dai protocolli a livello di collegamento:**
 - trasferimento dei pacchetti attraverso un singolo collegamento fra due nodi [questo era già stato detto]
 - Talvolta, rilevazione di errori nel pacchetto ricevuto ed eventuale correzione (l'interferenza elettromagnetica e l'attenuazione del segnale possono introdurre errori nel pacchetto trasferito)
 - Altri servizi che qui non citiamo

LIVELLO DI COLLEGAMENTO (III)

- **Funzionamento tipico:**

- Sul nodo mittente, un'entità a livello collegamento riceve un pacchetto (detto datagramma) dall'entità a livello rete, vi appone un proprio header, creando un pacchetto di livello collegamento (detto frame [è inglese: si legge "freim"]). In gergo, si dice che essa incapsula il datagramma in un frame. Infine, invoca i servizi del livello fisico per il trasferimento fisico del frame attraverso la connessione
- Sul nodo ricevente, un'entità a livello collegamento riceve un frame attraverso il livello fisico sottostante, esegue eventuali controlli (ad esempio individuazione/correzione errori), estrae dal frame il datagramma e passa quest'ultimo ad una soprastante entità a livello rete

- **N.B.: Le entità di livello di collegamento sono in grado di comprendere solo gli header del loro livello: questa è l'unica parte del frame che esse comprendono e gestiscono. La comprensione del datagramma incapsulato è di competenza delle entità di livello rete!**

LIVELLO DI COLLEGAMENTO (IV)

(Indirizzi MAC – Media Access Control)

- Vi sono situazioni in cui uno stesso collegamento è condiviso fra più nodi. In questo caso, tutti i nodi collegati ricevono un frame trasmesso su tale collegamento, ma solo il vero destinatario deve elaborarlo; gli altri nodi, devono ignorare il frame. Ci deve essere quindi un modo per specificare nel frame il destinatario dello stesso
- In generale, le entità a livello collegamento, in molti casi, hanno la necessità di specificare esplicitamente in un frame il destinatario e il mittente dello stesso: come fare?
- Si usano gli indirizzi MAC (Media Access Control)

LIVELLO DI COLLEGAMENTO (V)

(Indirizzi MAC – Media Access Control)

- Ad ogni nodo (host e router) sono associati degli indirizzi MAC che identificano univocamente tale nodo
- Più precisamente: gli indirizzi MAC (chiamati anche indirizzi LAN o indirizzi fisici) sono associati alle interfacce di rete di un nodo.
- Interfaccia di rete: punto fisico di connessione fra un nodo e la rete, costituito da una scheda hardware, es. scheda Ethernet, scheda WiFi
- Un nodo può avere più di una interfaccia di rete (es. un host con scheda Ethernet e scheda WiFi, inoltre i router hanno sempre almeno due interfacce di rete): ad ogni interfaccia è associato un indirizzo MAC che la identifica in maniera univoca

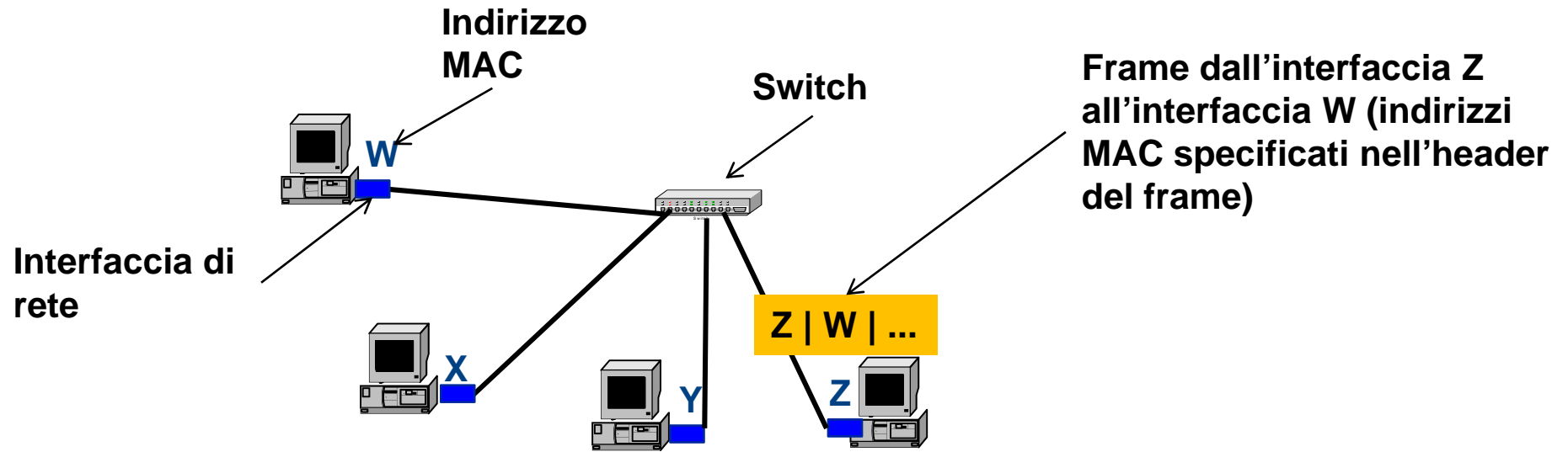
LIVELLO DI COLLEGAMENTO (VI)

(Indirizzi MAC – Media Access Control)

- **N.B. Nel mondo, non esistono due interfacce di rete aventi lo stesso indirizzo MAC!!!**
- **Quando un produttore vuole costruire un'interfaccia di rete, si rivolge ad un'organizzazione internazionale, la IEEE (si legge "I triple E"...è inglese, quindi si pronuncia "ai traipel i") – Institute of Electrical and Electronics Engineers – e acquista un insieme di indirizzi MAC ancora disponibili. LA IEEE garantisce di non vendere mai più di una volta lo stesso indirizzo MAC¹**
- **Ma come è fatto un indirizzo MAC? ...un indirizzo MAC è una sequenza di valori di bit, solitamente lunga 48 bit (cioè 6 byte)**
- **Es.: 110011100010100101010001110001110001110001110010**
- **Quando è necessario specificare il destinatario e il mittente di un frame, nell'header di collegamento di quello stesso frame vengono inseriti gli indirizzi MAC del destinatario e del mittente**

1 E' possibile modificare gli indirizzi MAC via software, ma per gli scopi di questa trattazione, possiamo pensare tali indirizzi come permanenti ed immutabili

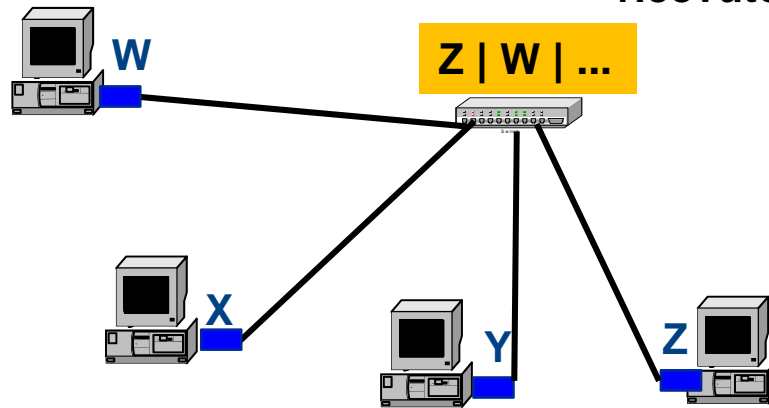
LIVELLO DI COLLEGAMENTO (VII) (Indirizzi MAC – Media Access Control)



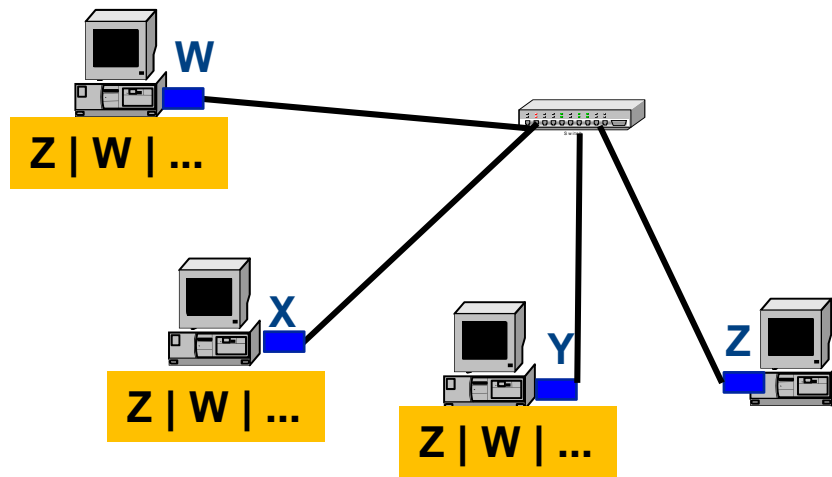
LIVELLO DI COLLEGAMENTO (VII)

(Indirizzi MAC – Media Access Control)

Lo switch riceve il frame e, in questo caso, lo inoltra a tutte le interfacce a cui è connesso (tranne quella da cui lo ha ricevuto)



LIVELLO DI COLLEGAMENTO (VII) (Indirizzi MAC – Media Access Control)

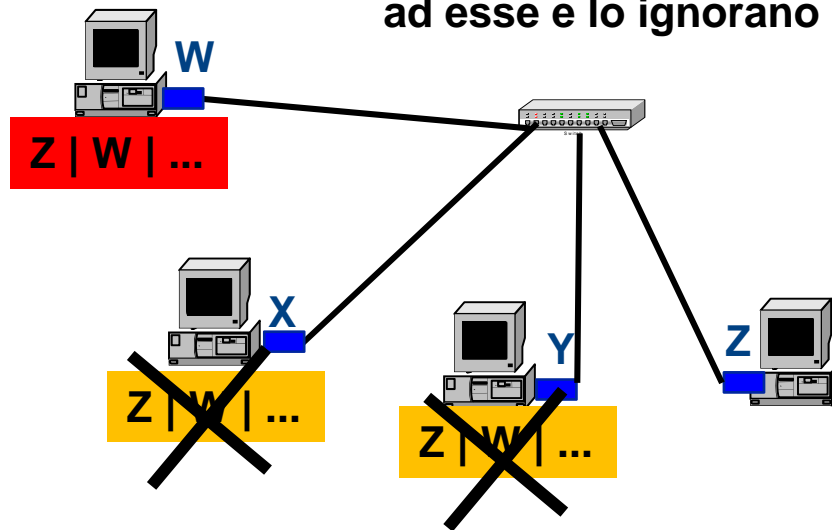


LIVELLO DI COLLEGAMENTO (VII)

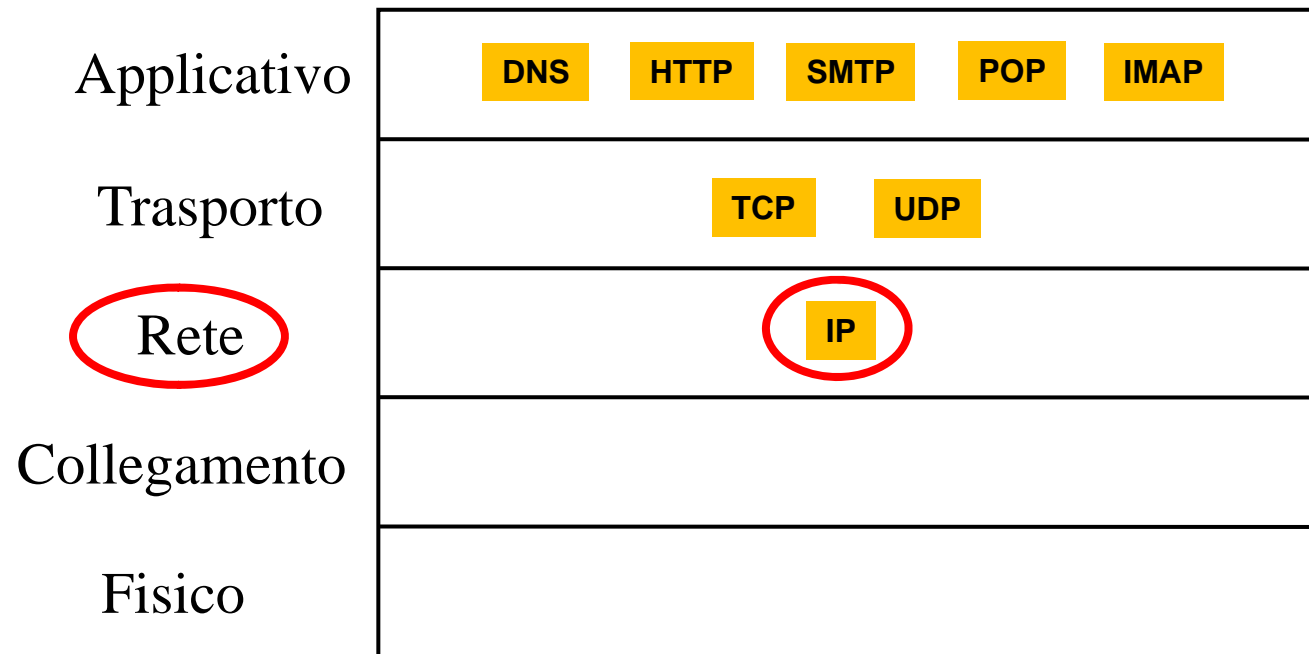
(Indirizzi MAC – Media Access Control)

L'interfaccia W riconosce che il frame è diretto ad essa e lo elabora.

Le altre interfacce riconoscono che non è diretto ad esse e lo ignorano



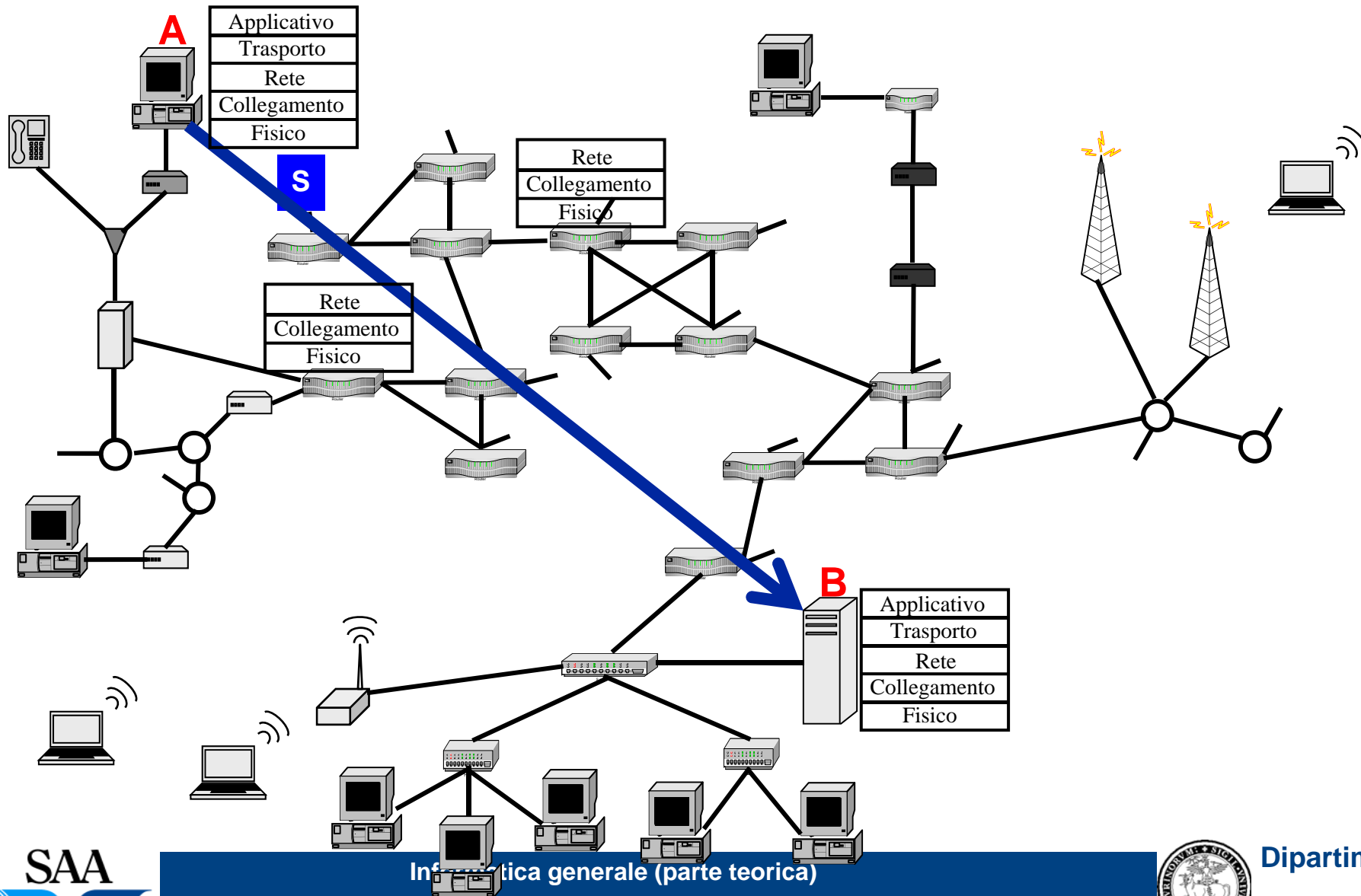
LIVELLO RETE: PROTOCOLLO IP (I)



LIVELLO RETE: PROTOCOLLO IP (II)

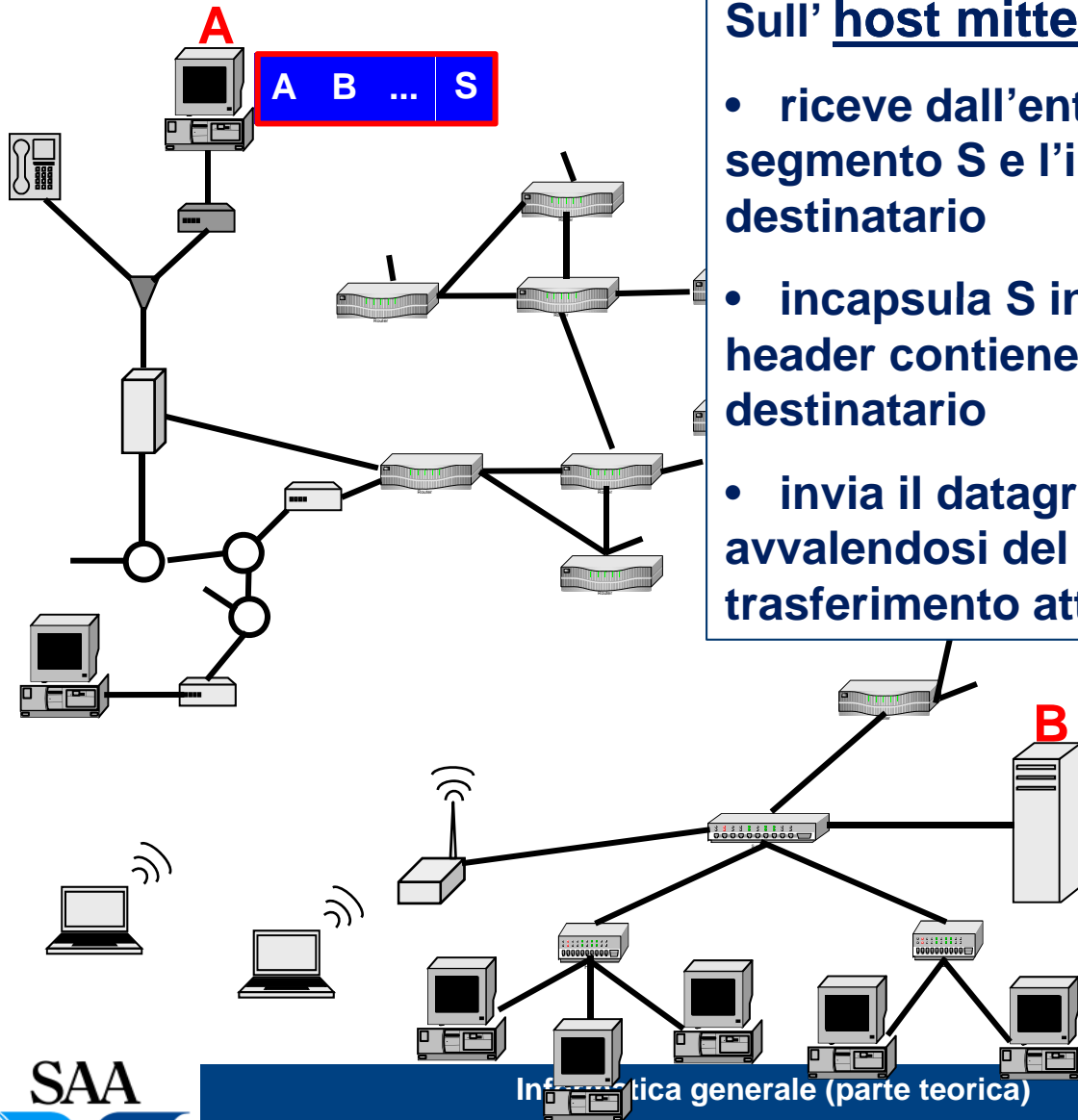
- Il protocollo IP (Internet Protocol) è uno dei principali protocolli Internet
- Al momento, in Internet sono usati due versioni di questo protocollo: la versione 4 (IPv4) e la versione 6 (IPv6)
- Si situa a livello rete e definisce un servizio di comunicazione “host-to-host”, cioè un servizio di trasporto dei pacchetti da un host mittente verso un host di destinazione
- Nell’attuazione di questo protocollo, oltre agli host, giocano un ruolo fondamentale i router
- Mentre gli host implementano tutti i livelli della gerarchia di protocolli (da quello fisico a quello applicativo), i router tipicamente implementano soltanto i 3 livelli più bassi: fisico, di collegamento e di rete

LIVELLO RETE: PROTOCOLLO IP (III)



LIVELLO RETE: PROTOCOLLO IP (IV)

Visione d'insieme

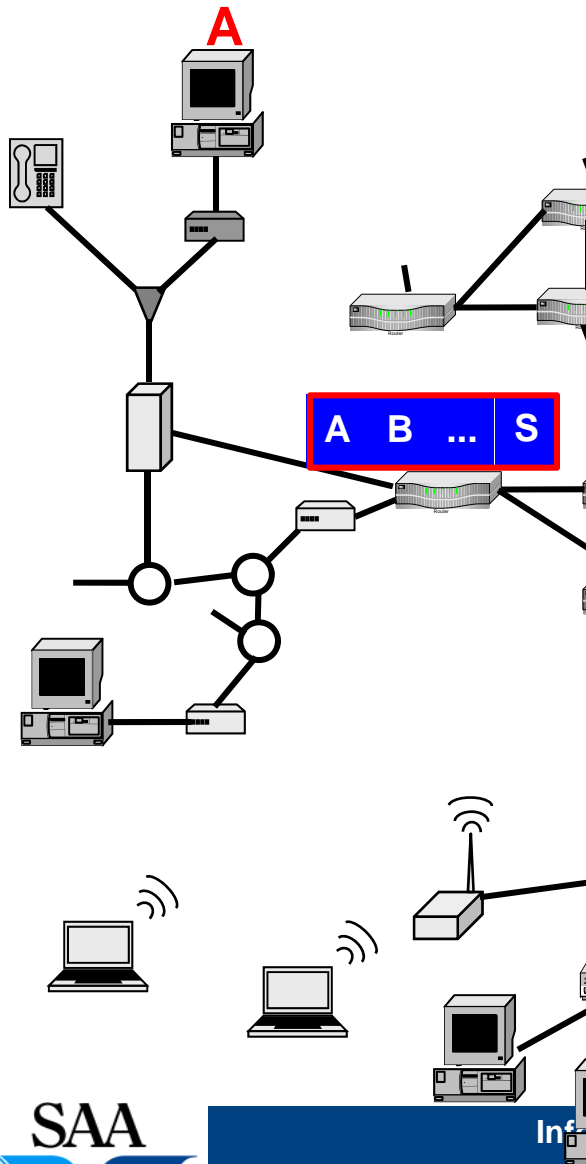


Sull' host mittente, l'entità che realizza IP :

- riceve dall'entità a livello di trasporto il segmento S e l'indicazione di mittente e destinatario
- incapsula S in un datagramma IP il cui header contiene indicazione di mittente e destinatario
- invia il datagramma ad un opportuno router, avvalendosi del servizio di collegamento per il trasferimento attraverso la connessione

LIVELLO RETE: PROTOCOLLO IP (IV)

Visione d'insieme

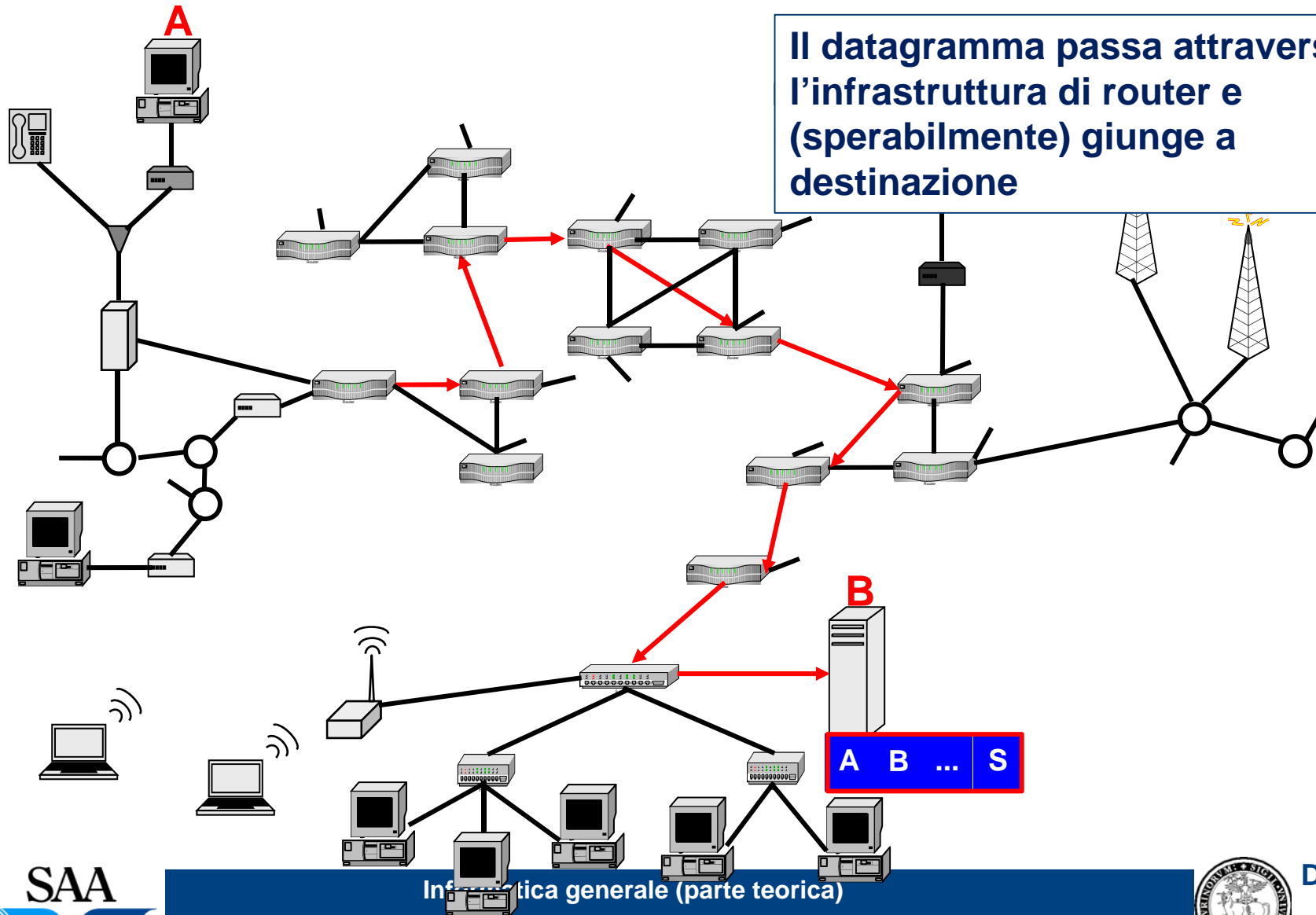


Sul router, l'entità che realizza IP :

- riceve dall'entità a livello di collegamento il datagramma IP
- controlla se esso è in circolazione da troppo tempo e, in tal caso, lo distrugge, altrimenti:
- se l'host destinatario appartiene ad una rete a cui il router è direttamente collegato, si avvale del servizio di collegamento per inoltrare il datagramma nella rete opportuna
- se l'host destinatario non appartiene ad alcuna delle reti a cui, eventualmente, il router è direttamente collegato, esso sceglie (in base alla destinazione finale) un altro router a cui inoltrare il datagramma, fra quelli a cui esso è direttamente collegato, nella speranza di far compiere al datagramma un "salto" nella giusta direzione e si avvale del servizio di collegamento per trasferire il datagramma al router selezionato attraverso la connessione che li collega

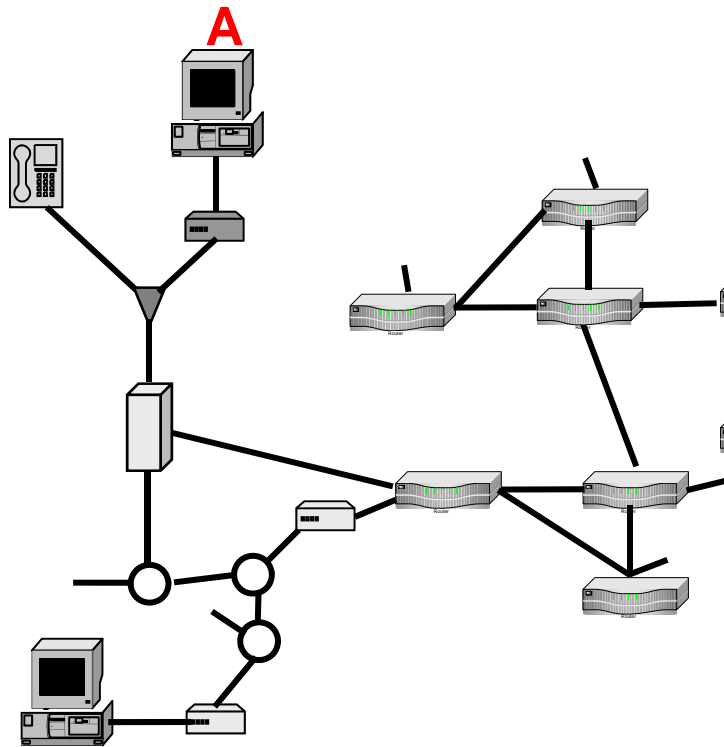
LIVELLO RETE: PROTOCOLLO IP (IV)

Visione d'insieme



LIVELLO RETE: PROTOCOLLO IP (IV)

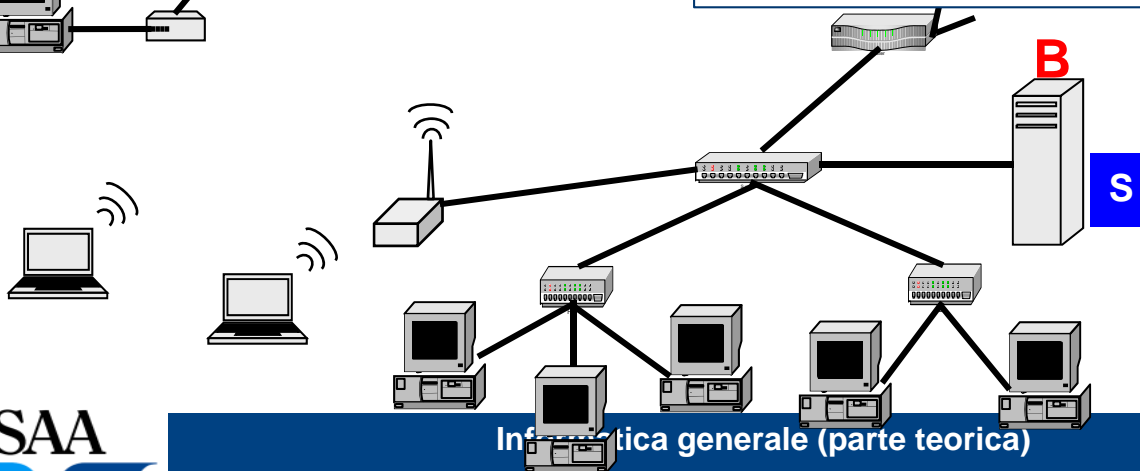
Visione d'insieme



Sull' host destinatario, l'entità che realizza IP :

- riceve dall'entità a livello di collegamento il segmento datagramma IP
- estrae dal datagramma il segmento S e lo passa all'opportuna entità situata a livello di trasporto

N.B.: si ricordi che l'obiettivo era trasferire S da A a B e che il datagramma IP è stato costruito proprio al fine di trasportare S da A a B!



LIVELLO RETE: PROTOCOLLO IP (V)

- **Restano da capire alcuni aspetti importanti:**
 - come vengono identificati i nodi a livello rete? Si usa un meccanismo di indirizzamento basato sui cosiddetti indirizzi IP (vedere in seguito)
 - come fa un router a sapere verso quale altro router inoltrare un datagramma? Usa delle tabelle di inoltro, prodotte dall'esecuzione di algoritmi di routing
- **Infine, resta da esaminare brevemente la struttura di un datagramma IP**
- **Continuiamo...**

LIVELLO RETE: PROTOCOLLO IP (VI)

Indirizzi IP

- Un indirizzo IP è (salvo eccezioni, come nel caso dell'approccio NAT – Network Address Translation – che non tratteremo) un identificatore univoco di nodo (o, meglio: di un'interfaccia di rete di un nodo) in Internet
- Come per gli indirizzi MAC, in Internet, ad ogni interfaccia di rete è associato un indirizzo IP che la identifica univocamente: (tralasciando le suddette eccezioni) possiamo affermare che uno stesso indirizzo IP non è mai assegnato a più di un'interfaccia di rete
- Un nodo può avere più di un'interfaccia di rete (un router ne ha almeno due) → un nodo può avere più indirizzi IP
- Il livello rete usa gli indirizzi IP (e non gli indirizzi MAC!) per identificare i nodi a cui inoltrare e da cui provengono i datagrammi

LIVELLO RETE: PROTOCOLLO IP (VII)

Indirizzi IP: IPv4

- In IPv4, ogni indirizzo IP è una sequenza di 32 valori di bit
- es.: 11010001100010000100111100100010
- Sgradevole da scrivere e difficilino da ricordare, eh?
- Per agevolare le persone nell'uso degli indirizzi IP si usa la dot notation, che è un diverso e più semplice modo per scrivere gli indirizzi, più adatto alle persone (ma al livello rete, i nodi usano solo la sequenza di 32 valori di bit)

LIVELLO RETE: PROTOCOLLO IP (VIII)

Indirizzi IP: IPv4

- Consideriamo ad esempio l'indirizzo IP
11010001100010000100111100100010
- Per trasformarne la notazione da binaria nella corrispondente *dot notation*, si procede così:
 1. si dividono i 32 bit in 4 gruppi da 8 bit:
11010001 10001000 01001111 00100010
 2. per ogni blocco, si considera il corrispondente numero decimale:
209 136 79 34
 3. si scrivono i numeri decimali così trovati separandoli con un punto:
209.136.79.34
- **sempre brutto...ma un po' più facile da usare per chi è una persona e non un host né un router!**

LIVELLO RETE: PROTOCOLLO IP (IX)

Indirizzi IP: IPv4

- A differenza di un indirizzo MAC (che è “piatto”), un indirizzo IP (tranne casi particolari) è strutturato al proprio interno: una parte di esso (una sequenza di valori che inizia a sinistra) è usata per identificare una (sotto)rete e la parte restante è usata per identificare (l’interfaccia di rete di) un nodo all’interno di tale (sotto)rete¹

identificatore di (sotto)rete (in questo esempio, i primi 24 bit)

- Es.: **11010001100010000100111100100010**

identificatore di nodo (i restanti bit: in questo esempio 8 bit)

¹ Facciamo riferimento all’approccio cosiddetto CIDR (Classless Interdomain Routing - si pronuncia “cider”), attualmente molto diffuso

LIVELLO RETE: PROTOCOLLO IP (X)

Indirizzi IP: IPv4

- Il numero di bit usati per l'identificazione della (sotto)rete (e, di conseguenza, anche il numero di quelli usati per l'identificazione di un nodo all'interno di una (sotto)rete) non è fisso; al contrario, esso può variare da un indirizzo all'altro
- Nella dot notation, il numero di bit usati per identificare una (sotto)rete è indicato come nel seguente modo: 209.136.79.0/24

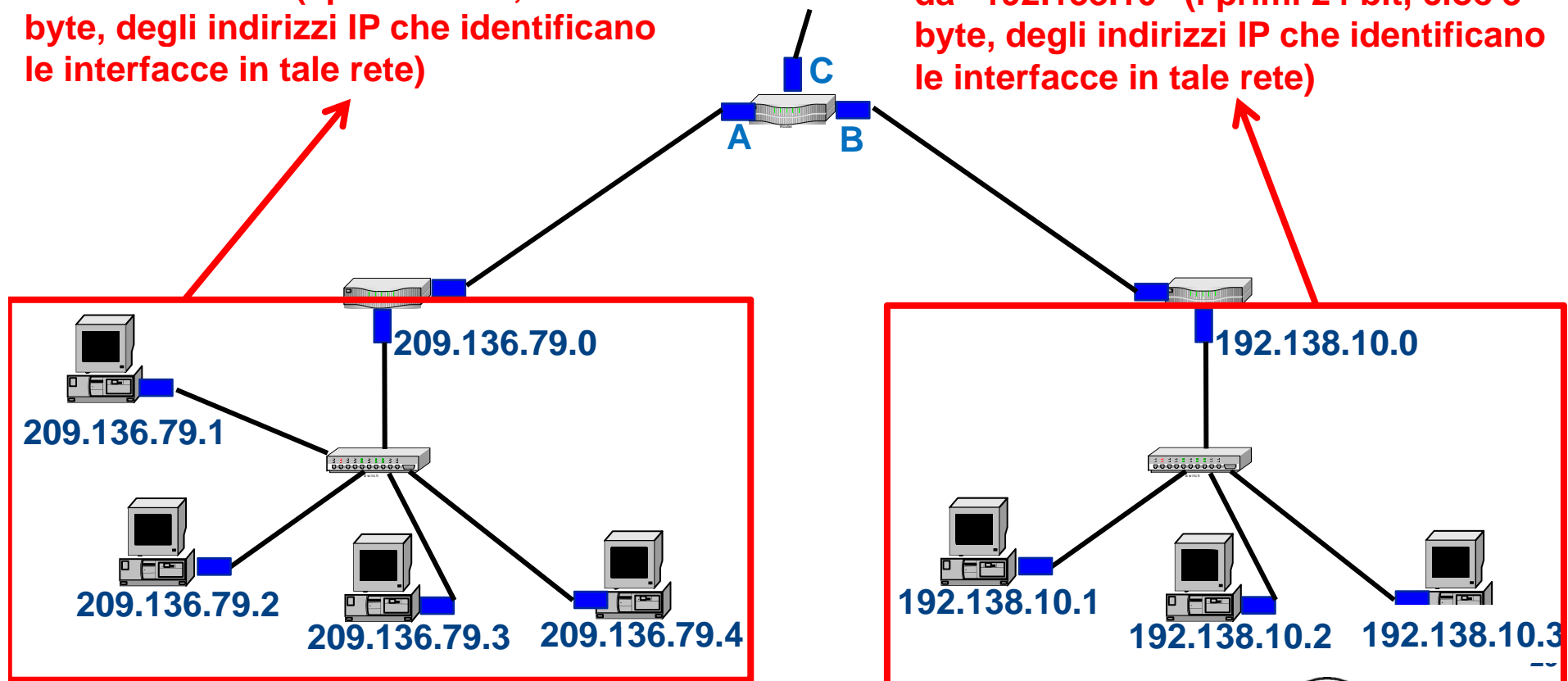
I primi 24 bit di questo indirizzo (e di tutti quelli ottenibili variando l'ultima cifra, corrispondente agli ultimi 8 bit dell'indirizzo) identificano una (sotto)rete

LIVELLO RETE: PROTOCOLLO IP (XI)

Indirizzi IP: IPv4

Rete 209.136.79.0/24, cioè identificata da “209.136.79” (i primi 24 bit, cioè 3 byte, degli indirizzi IP che identificano le interfacce in tale rete)

Rete 192.138.10.0/24, cioè identificata da “192.138.10” (i primi 24 bit, cioè 3 byte, degli indirizzi IP che identificano le interfacce in tale rete)



LIVELLO RETE: PROTOCOLLO IP (XII)

Indirizzi IP: IPv4

- E' frequente il caso in cui una stessa organizzazione ha competenza su più sottoreti, es. una stessa organizzazione potrebbe avere competenza sulle sottoreti 209.136.76.0/24, 209.136.77.0/24, 209.136.78.0/24 e 209.136.79.0/24
- Se dalla dot notation passiamo alla rappresentazione binaria, vediamo che gli identificatori di sottorete sono uguali nei primi 22 bit:

209.136.76.0 = 11010001100010000100110000000000

209.136.77.0 = 11010001100010000100110100000000

209.136.78.0 = 11010001100010000100111000000000

209.136.79.0 = 11010001100010000100111100000000

- La stessa organizzazione ha competenza sulle 4 sottoreti i cui identificatori di 24 bit iniziano tutti con la sequenza di 22 bit 1101000110001000010011: questa sequenza può essere utilizzata per identificare l'insieme delle 4 sottoreti, nel suo complesso
- 209.136.76.0/22 indica che i primi 22 bit di questo indirizzo IP e di tutti quelli ottenibili variando i valori degli ultimi 10 bit identificano una sottorete (che, in questo esempio, è ulteriormente articolata in 4 sottoreti)

LIVELLO RETE: PROTOCOLLO IP (XIII)

Indirizzi IP: IPv4

- **Internet è una rete di reti: i router al di fuori della (sotto)rete di destinazione di un datagramma considerano solo l'identificatore di (sotto)rete [o una parte di esso, sufficiente ad identificare un insieme di sottoreti, solitamente appartenenti alla stessa organizzazione, come nell'esempio precedente] per inoltrare il datagramma verso la (sotto)rete di destinazione. Gli altri bit dell'indirizzo IP sono considerati solo dai router appartenenti alla (sotto)rete di destinazione [o all'insieme di sottoreti a cui appartiene quella di destinazione]**
- **...questo semplifica molto la gestione del meccanismo di inoltro dei datagrammi, in quanto ogni router non ha la necessità di possedere conoscenza su ogni singolo host presente in Internet, ma, per quelli che non appartengono alle proprie reti, necessita solo di sapere come far fare al datagramma un salto (sperabilmente) nella direzione della rete giusta: una volta raggiunta la rete di destinazione, saranno i router locali ad occuparsi dell'inoltro all'host appropriato!**

LIVELLO RETE: PROTOCOLLO IP (XIV)

Tabelle di inoltro

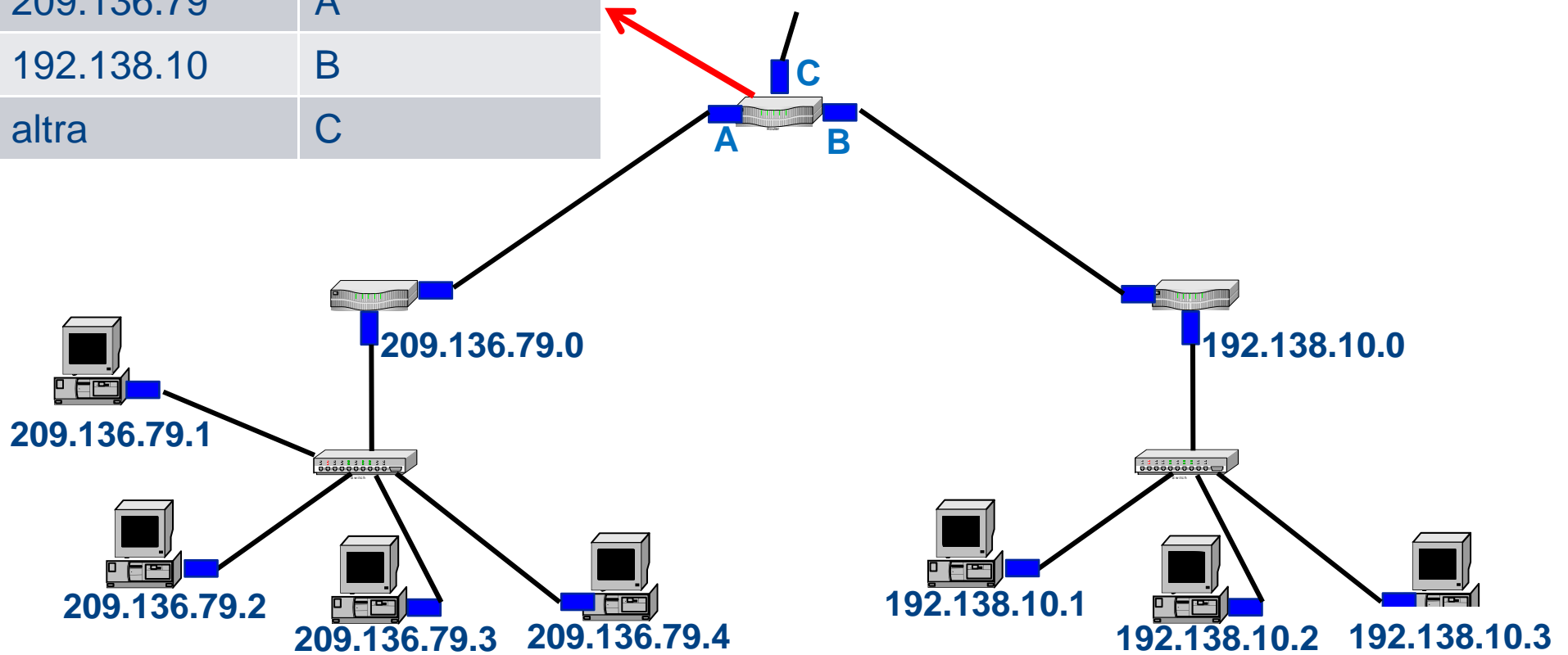
- Quando un router deve inoltrare un datagramma, deve scegliere una delle sue interfacce di rete sulla quale eseguire l'inoltro
- Per operare questa scelta, si basa sull'indirizzo IP del destinatario (tale indirizzo è riportato nell'header del datagramma)
- Se il destinatario appartiene ad una stessa rete a cui appartiene il router, questo inoltro direttamente il datagramma, attraverso l'interfaccia che lo connette a tale rete
- Se il destinatario non appartiene ad alcuna eventuale rete a cui il router è direttamente connesso, allora esso individua un'opportuna interfaccia attraverso cui inviare il datagramma, nella speranza di far compiere al datagramma un passo nella direzione giusta
- Come fa il router ad individuare la giusta interfaccia attraverso cui inviare il datagramma? Utilizza la propria tabella di inoltro

LIVELLO RETE: PROTOCOLLO IP (XV)

Tablelle di inoltr

Tabella di inoltr

RETE	INTERFACCIA
209.136.79	A
192.138.10	B
altra	C



LIVELLO RETE: PROTOCOLLO IP (XVI)

Algoritmi di routing

- Chi inserisce le tabelle di inoltro nei router in Internet?
- Le tabelle di inoltro vengono prodotte dai router stessi, i quali, scambiandosi informazioni ed eseguendo dei complessi programmi, chiamati algoritmi di routing, sono in grado di produrre tali tabelle
- Internet è un sistema dinamico (nuove reti vengono connesse, altre reti possono scomparire, nuovi host vengono aggiunti, ecc.), quindi le tabelle di inoltro vengono aggiornate frequentemente

LIVELLO RETE: PROTOCOLLO IP (XVII)

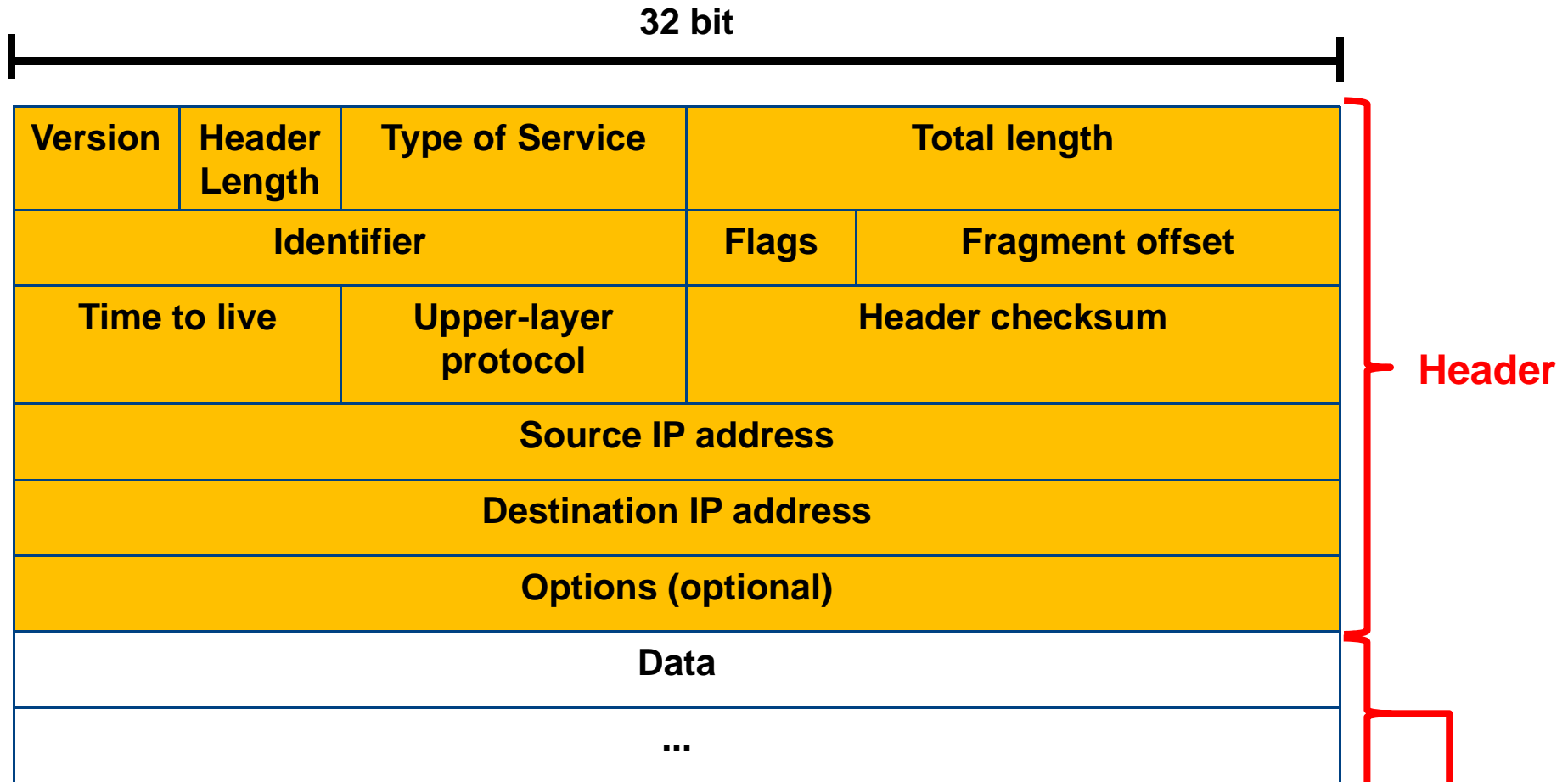
Datagramma IP: IPv4

32 bit

32 bit			
Version	Header Length	Type of Service	Total length
Identifier		Flags	Fragment offset
Time to live	Upper-layer protocol	Header checksum	
Source IP address			
Destination IP address			
Options (optional)			
Data			
...			

LIVELLO RETE: PROTOCOLLO IP (XVII)

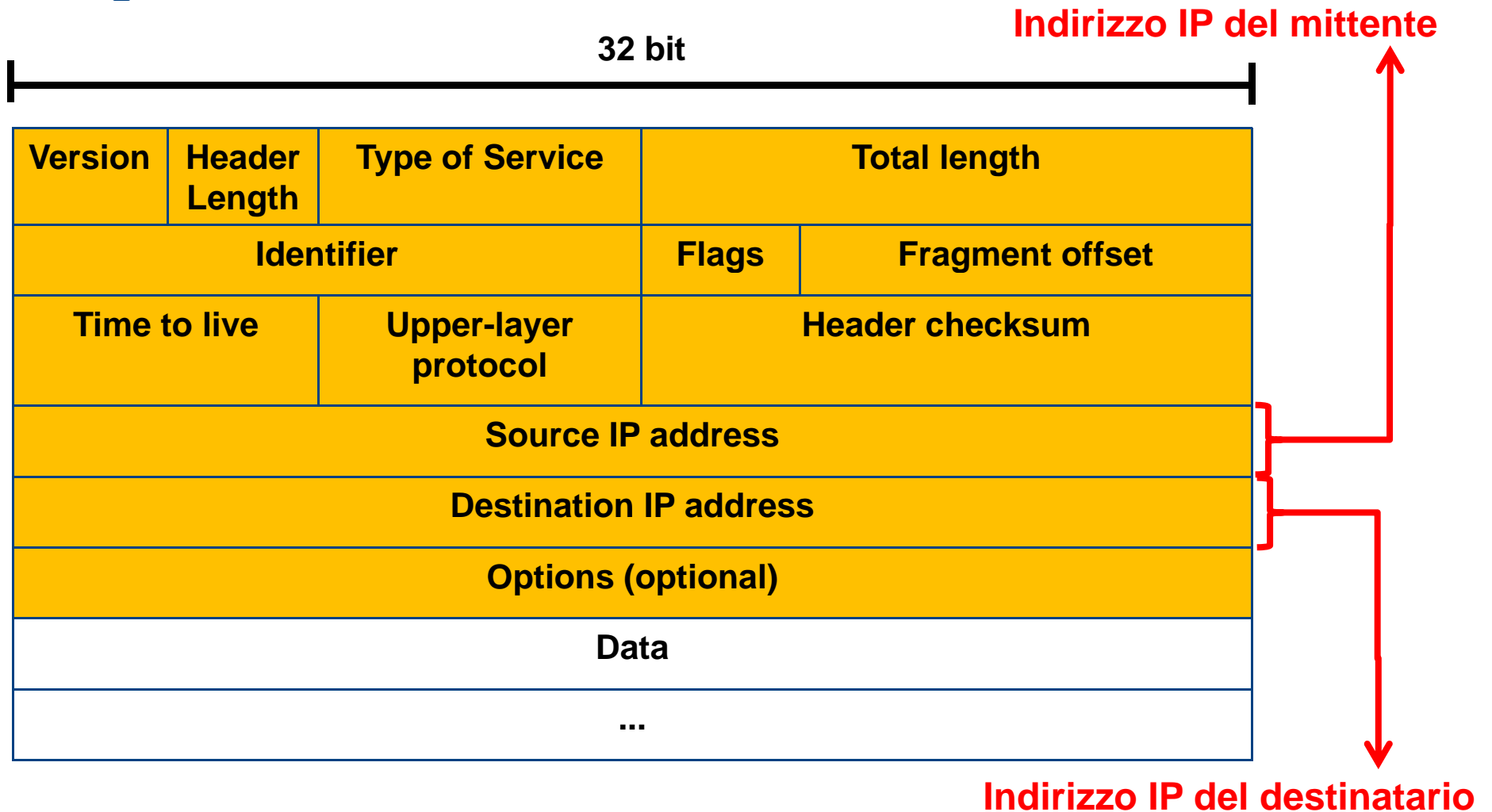
Datagramma IP: IPv4



Parte dati: tipicamente è costituita da un segmento del livello di trasporto (es. segmento UDP o TCP) 36

LIVELLO RETE: PROTOCOLLO IP (XVII)

Datagramma IP: IPv4



LIVELLO RETE: PROTOCOLLO IP (XVII)

Datagramma IP: IPv4

Massima durata della vita del datagramma: quando questo limite è raggiunto, il router che riceve il datagramma lo distrugge. Si evita così di avere datagrammi che saltano all'infinito da un router all'altro

Version	Header Length	Identifier		Flags	Fragment offset
Time to live		Upper-layer protocol	Header checksum		
Source IP address					
Destination IP address					
Options (optional)					
Data					
...					

LIVELLO RETE: PROTOCOLLO IP (XVIII)

IPv6

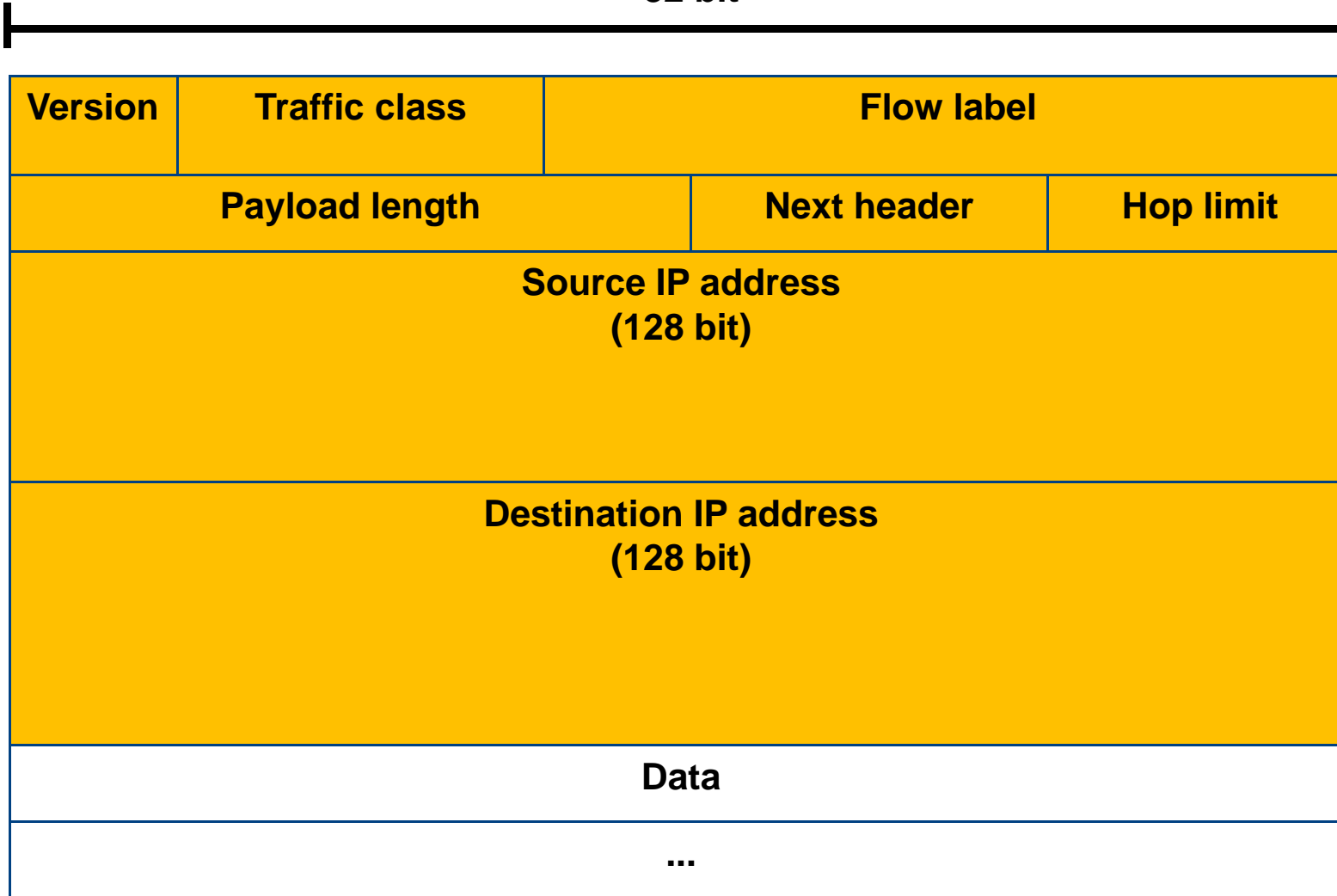
- Accanto ad IPv4 è attualmente in uso in Internet anche la versione 6 del protocollo IP: IPv6
- La principale ragione che ha spinto allo sviluppo di IPv6 è stata la carenza di indirizzi IPv4
- IPv4 ha indirizzi a 32 bit e ciò, in linea teorica, consentirebbe di indirizzare circa 2^{32} – vale a dire più di 4 miliardi – (interfacce di) nodi
- Tuttavia, la rapida crescita di Internet e l'uso non ottimale degli indirizzi IPv4 hanno prodotto carenza di indirizzi

- IPv6 specifica indirizzi di 128 bit (...e 2^{128} è un numero veramente grande!)
- Il datagramma IPv6 risulta semplificato rispetto ad IPv4, come si può vedere di seguito

LIVELLO RETE: PROTOCOLLO IP (XIX)

Datagramma IP: IPv6

32 bit



LIVELLO RETE: PROTOCOLLO IP (XX)

Inaffidabilità di IP

- **IMPORTANTISSIMO!!! IP non è un protocollo affidabile!**
- ...nel senso che IP non garantisce che un datagramma arrivi effettivamente a destinazione (infatti, un datagramma può superare il proprio limite di vita ed essere distrutto, oppure può essere perso da parte di un router “sovraccarico”), né che vi arrivi integro (infatti IP non si preoccupa di controllare che eventuali interferenze elettromagnetiche o indebolimento del segnale o altro non abbiano introdotto errori nel datagramma), né che i datagrammi arrivino nella sequenza in cui sono stati spediti (infatti, essi possono seguire percorsi diversi e, di conseguenza, impiegare tempi diversi per giungere a destinazione)
- L’attuazione di servizi affidabili è demandata ai livelli superiori nella gerarchia di protocolli

LIVELLO RETE: PROTOCOLLO IP (XXI)

A chi richiedere gli indirizzi IP?

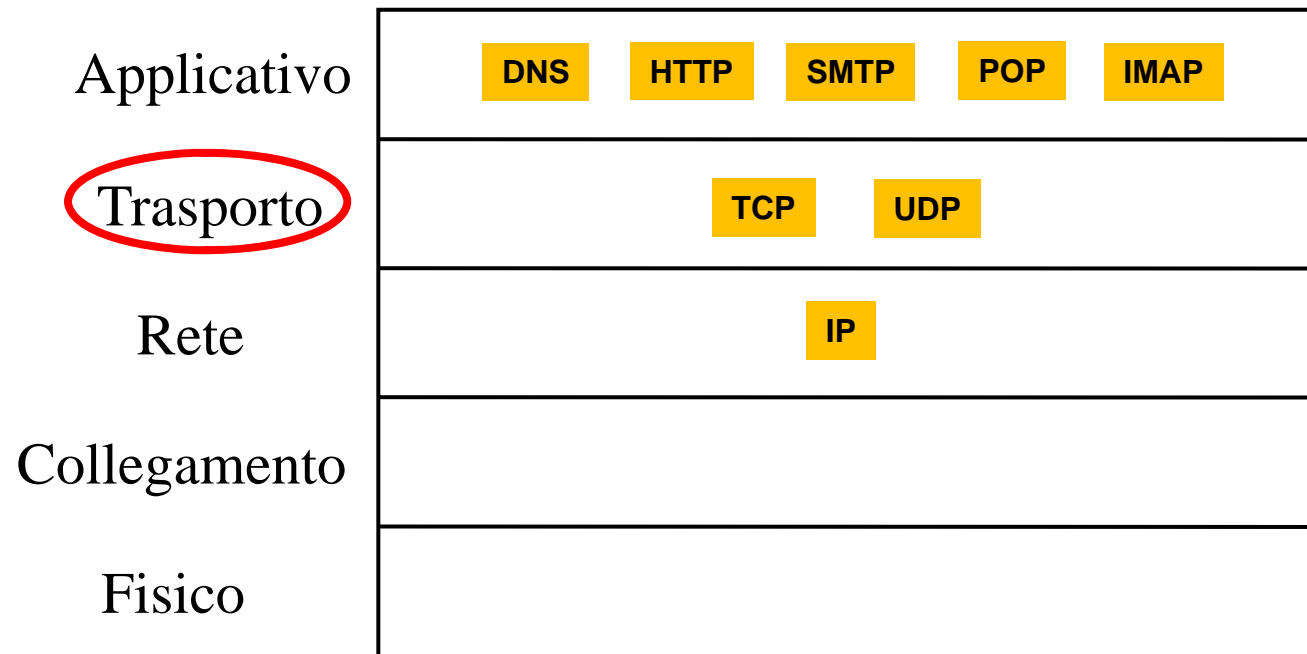
- Un'organizzazione che necessita di un blocco di indirizzi IP per connettere una propria rete ad Internet, può contattare un ISP (Internet Service Provider)
- Un ISP assegna un blocco di indirizzi IP ad ogni suo cliente, “pescando” opportunamente fra gli indirizzi che gli sono stati a sua volta assegnati
- Chi assegna gli indirizzi IP agli ISP?
- In ultima analisi, gli indirizzi IP sono amministrati dall'ICANN (Internet Corporation for Assigned Names and Numbers), un ente internazionale no-profit con sede negli Stati Uniti

LIVELLO RETE: PROTOCOLLO IP (XXII)

Indirizzi IP vs Indirizzi MAC

- **Perché abbiamo bisogno sia degli indirizzi IP che degli indirizzi MAC? Non sarebbe sufficiente un solo tipo di indirizzi?**
- **No: abbiamo bisogno di entrambi!**
- Infatti, abbiamo la necessità di identificare univocamente una singola interfaccia di rete, indipendentemente dalla rete a cui è connessa (questo ruolo è giocato dagli *indirizzi MAC*, che non variano quando un nodo migra da una rete all'altra...*sono analoghi al codice fiscale per le persone*); ma abbiamo anche la necessità di un meccanismo che associ ad un'interfaccia di rete un identificatore che dipenda dalla rete a cui essa è connessa (questo ruolo è giocato dagli indirizzi IP, infatti, un nodo che migri da una rete all'altra deve cambiare gli indirizzi IP delle proprie interfacce, ma non il loro indirizzo MAC...*gli indirizzi IP sono analoghi agli indirizzi di residenza per le persone*)
- Inoltre, le interfacce di rete devono poter funzionare anche in reti costruite con tecnologie diverse da Internet, quindi devono avere un meccanismo di identificazione che non assuma l'uso del protocollo IP
- Infine, gli indirizzi MAC sono “piatti” e non contengono l'indicazione della rete a cui una certa interfaccia è connessa: il loro uso a livello rete complicherebbe molto il lavoro dei router e le loro tabelle di inoltra!

LIVELLO DI TRASPORTO



LIVELLO DI TRASPORTO (I)

- Come detto in precedenza, il protocollo IP offre un servizio (non affidabile) di trasporto pacchetti da un host ad un altro host (comunicazione host-host)
- Per realizzare applicazioni Internet questo servizio è necessario, ma non sufficiente: infatti non basta che i pacchetti raggiungano l'host corretto...su un host possono risiedere più processi di livello applicativo, quindi è necessario che sull'host di destinazione i pacchetti vengano recapitati al giusto processo applicativo (comunicazione processo-processo)
- Ad esempio, nell'applicazione Web, è necessario che i pacchetti provenienti da un browser siano consegnati al Web server di destinazione (e viceversa); nell'applicazione di posta elettronica, occorre che i pacchetti del client di posta raggiungano il corretto server di posta (e viceversa), ecc.

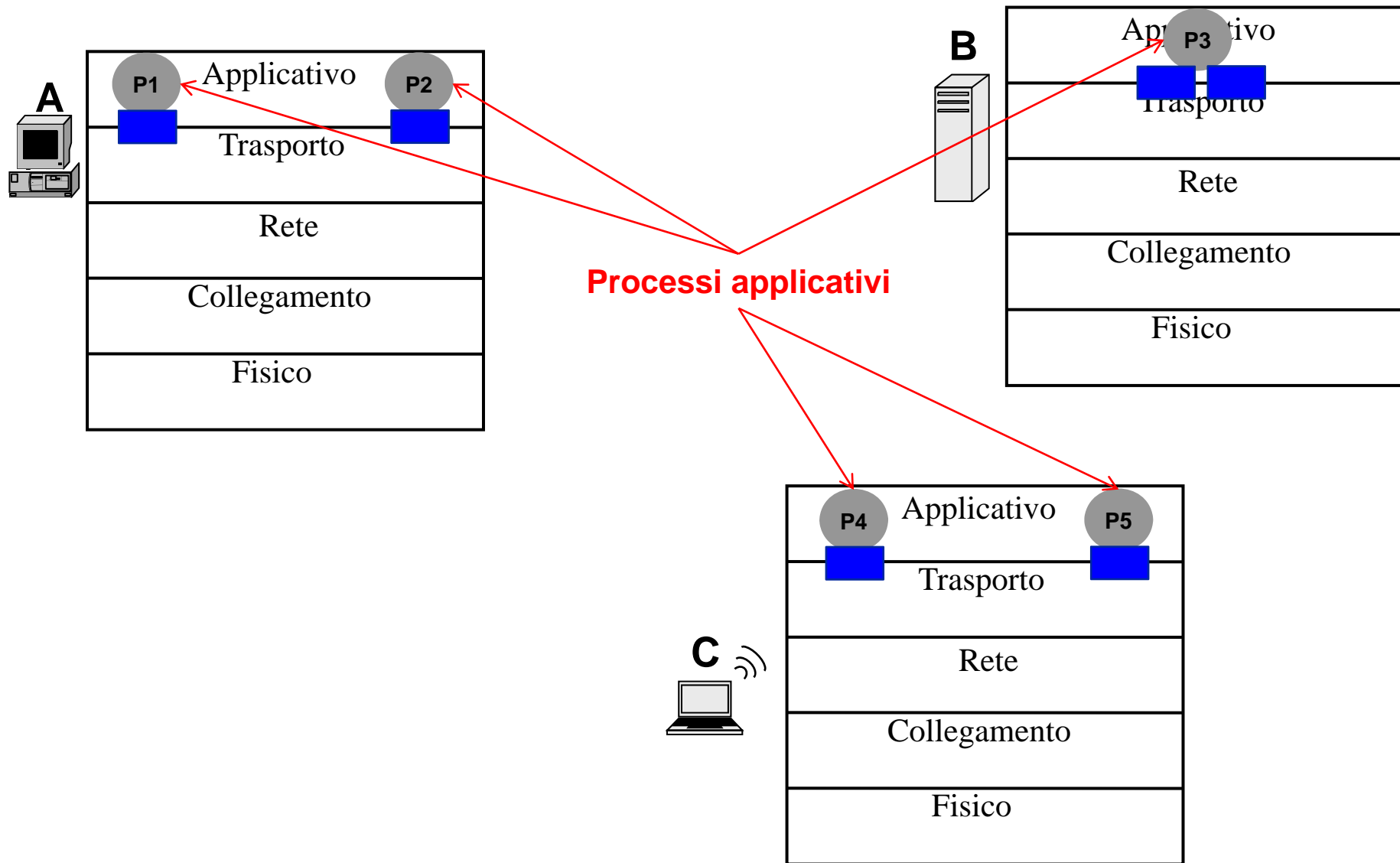
LIVELLO DI TRASPORTO (II)

- I protocolli di trasporto si occupano proprio di questi aspetti
- **N.B.: I protocolli di trasporto sono implementati sugli host, ma non sui router!**
- → c'è bisogno di un modo che consenta all'entità per il protocollo di trasporto di recapitare il pacchetto al processo a cui esso è indirizzato
- Tecnicamente, il fulcro di questo meccanismo è il concetto di **socket**

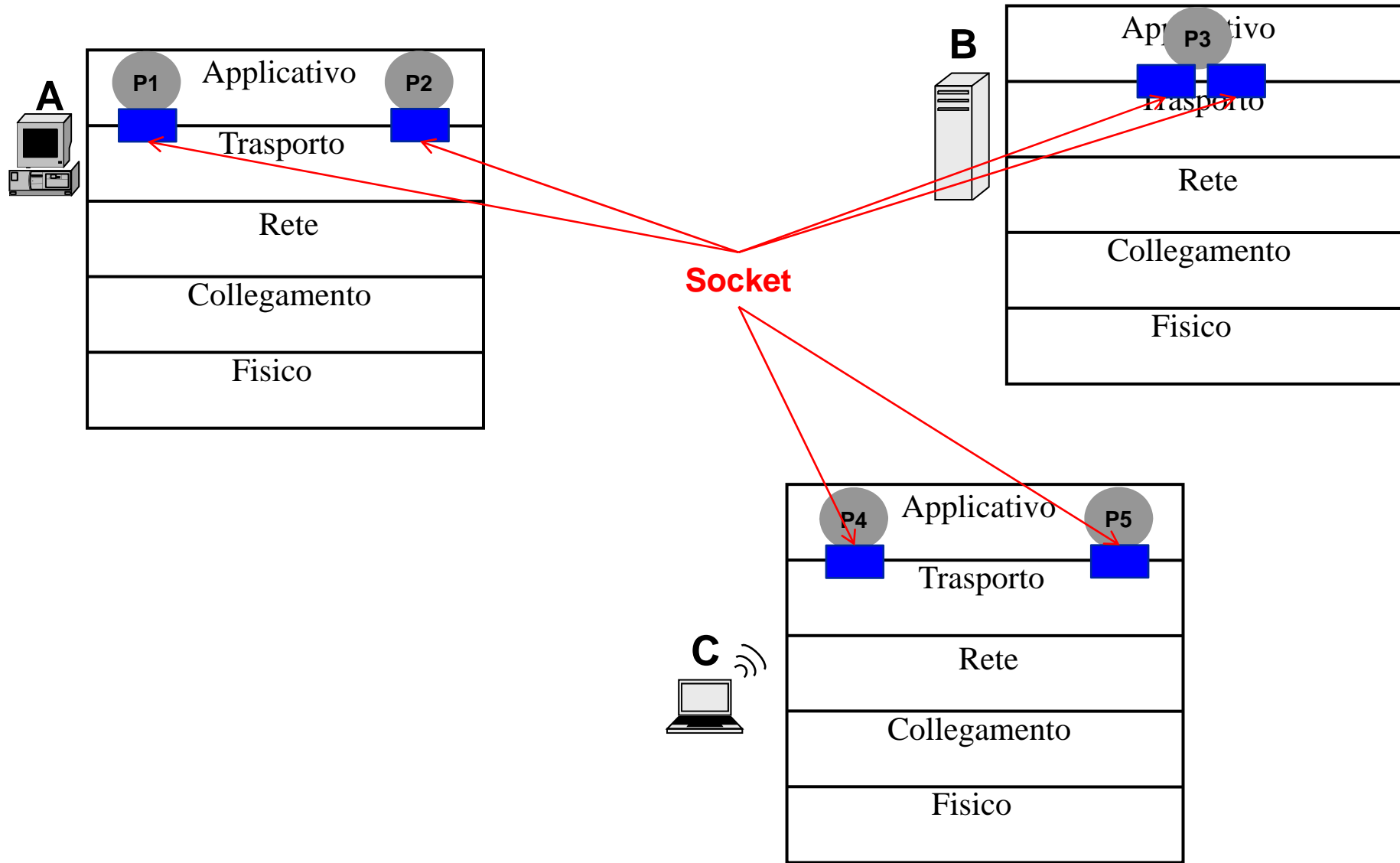
- Possiamo pensare ad un socket come ad un' "apertura" (in senso figurato, ovviamente) che, su un singolo host, mette in contatto un processo applicativo con il livello di trasporto e attraverso cui passano i pacchetti da e per quel processo applicativo

- Ad ogni processo applicativo è associato sempre almeno un socket (ma possono esservene associati anche più d'uno)

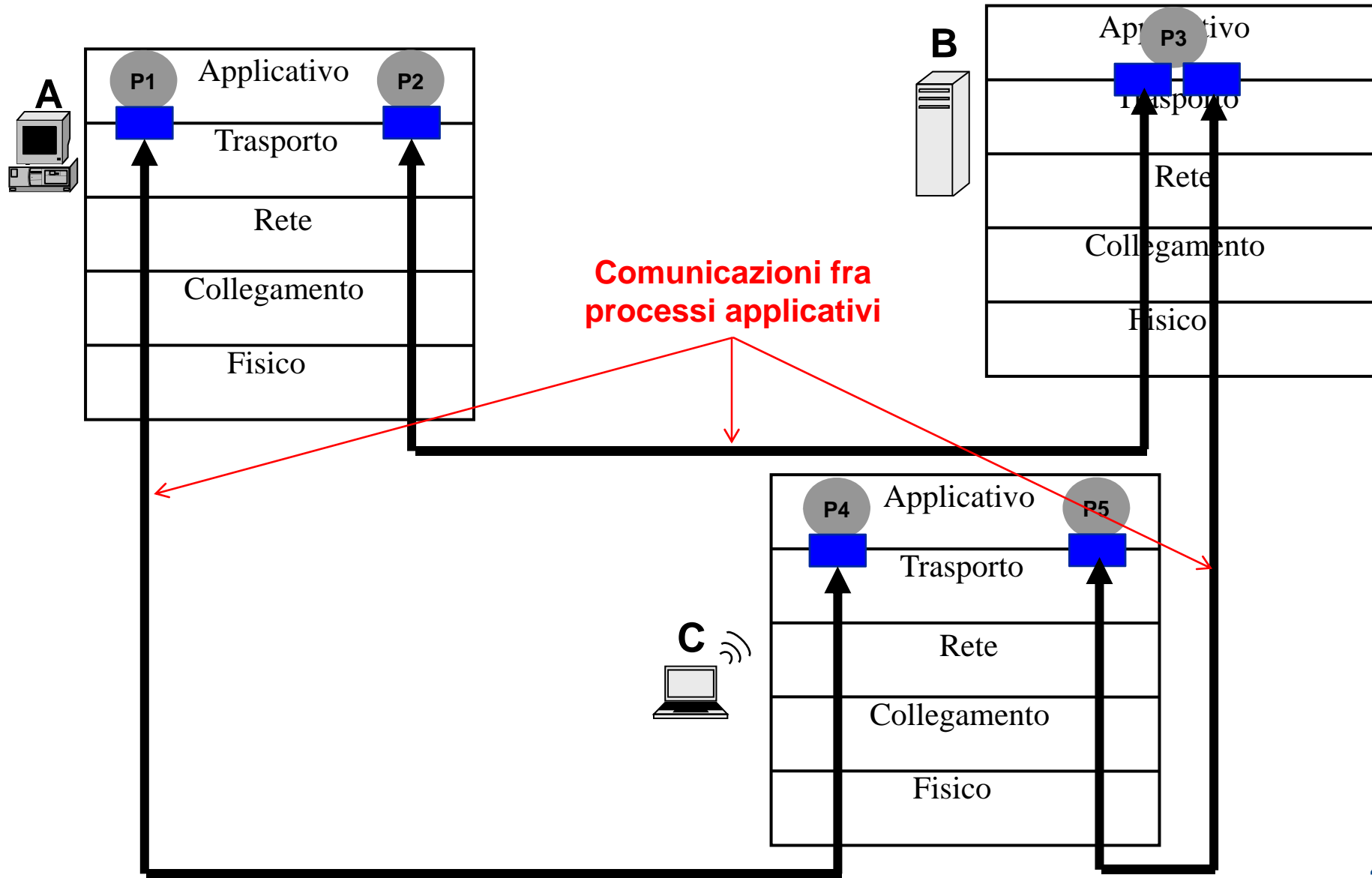
LIVELLO DI TRASPORTO (III)



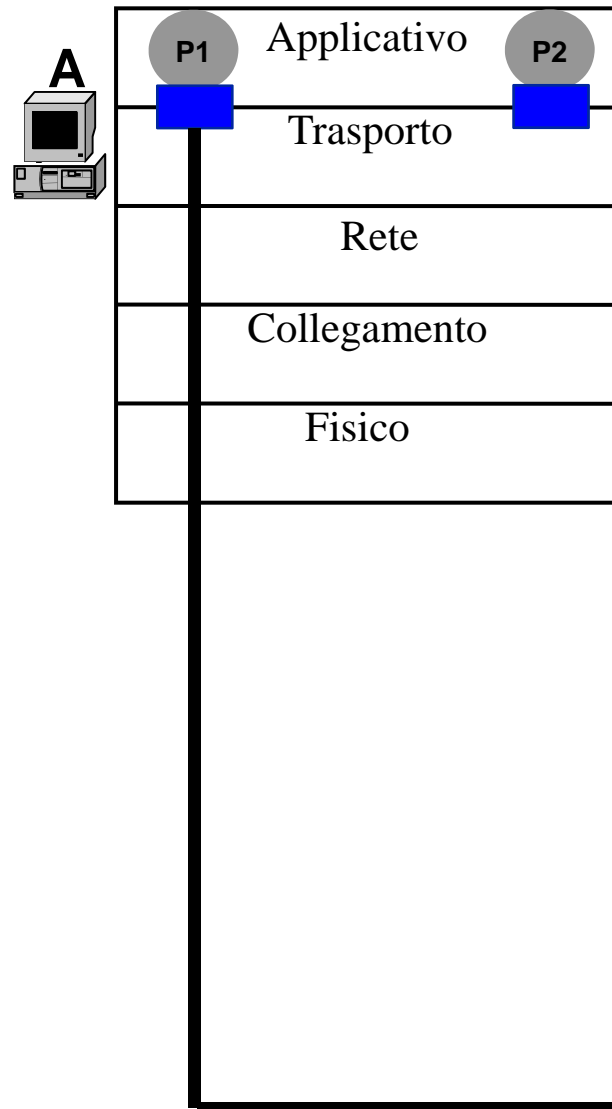
LIVELLO DI TRASPORTO (IV)



LIVELLO DI TRASPORTO (V)



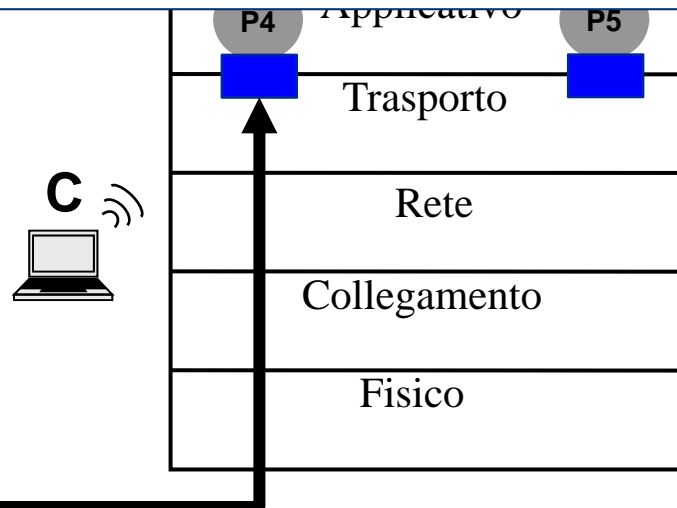
LIVELLO DI TRAS



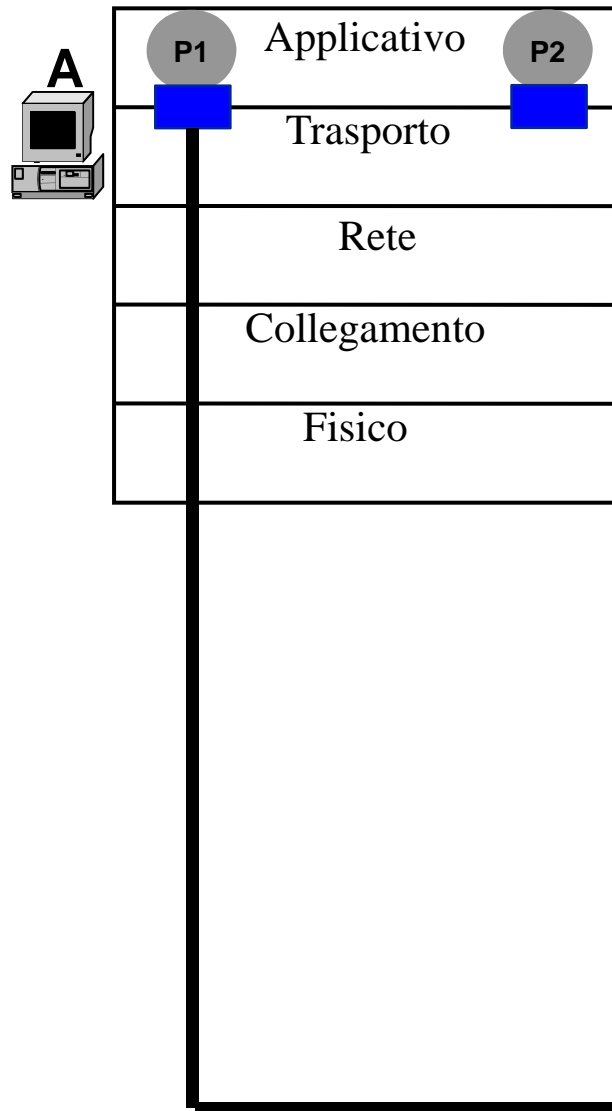
Comunicazione P1 → P4

Sull' host mittente, l'entità che realizza il protocollo di trasporto :

- riceve attraverso il socket dal processo applicativo P1 il messaggio m , con l'indicazione del destinatario P4 e del mittente P1
- eventualmente spezza m in blocchi più piccoli (p_1, \dots, p_n)
- incapsula m o ogni blocco p_i in un segmento di trasporto S , il cui header contiene (una parte dei) dati necessari ad identificare gli opportuni socket di P1 e P4
- passa S all'entità che realizza il protocollo di rete (IP), con l'indicazione degli indirizzi IP di mittente e destinatario



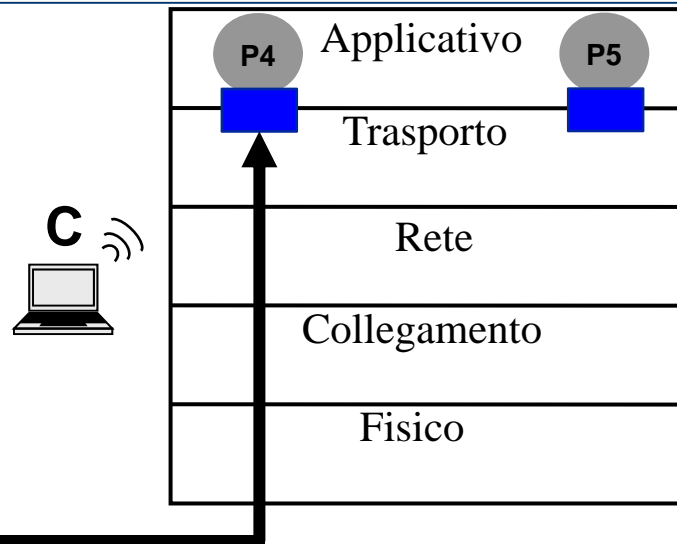
LIVELLO DI TRASPORTO (VII)



Comunicazione P1 → P4

Sull' host destinatario, l'entità che realizza il protocollo di trasporto :

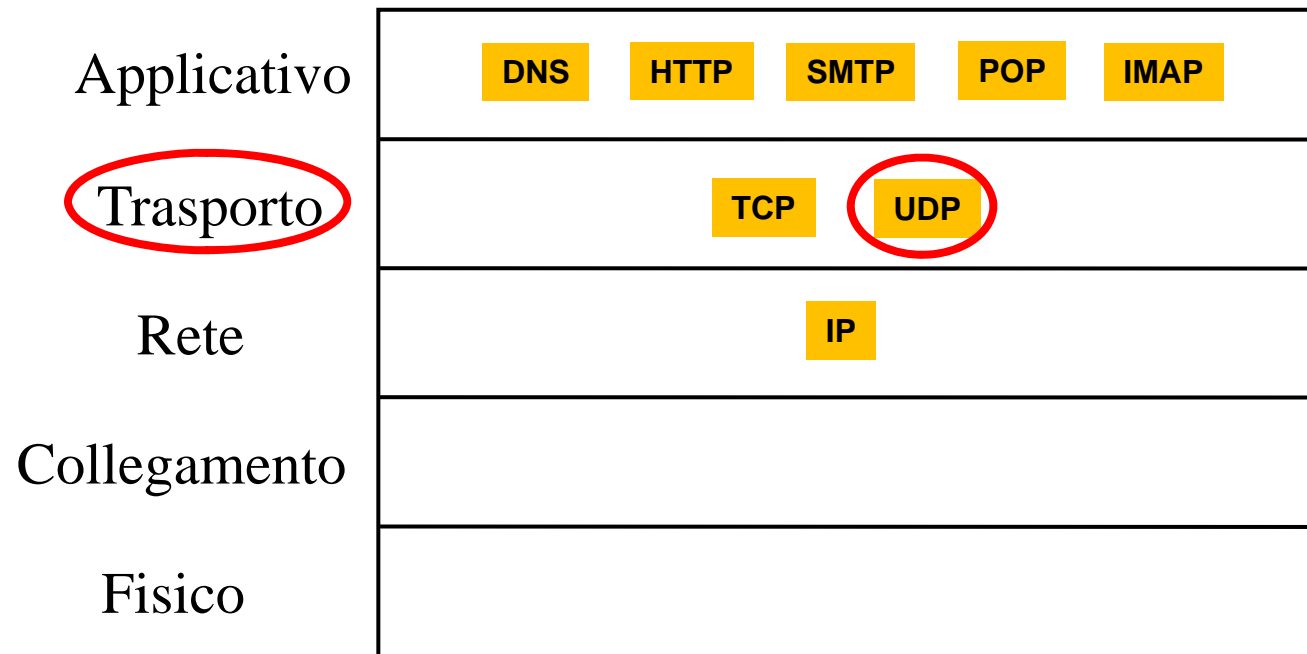
- riceve il segmento S dall'entità che realizza il protocollo IP
- individua il socket opportuno attraverso cui recapitare S
- recapita S attraverso il socket opportuno
- tramite il socket, i dati applicativi raggiungono il processo applicativo P4



LIVELLO DI TRASPORTO (VIII)

- Come fa l'entità che realizza il protocollo di trasporto sull'host destinazione ad individuare il socket opportuno attraverso cui recapitare il segmento?
- Si basa sulle informazioni contenute nel segmento...fra queste, un ruolo di primo piano è giocato dai cosiddetti numeri di porta
- Un numero di porta è un intero di 16 bit (quindi un numero compreso fra 0 e 65535): nei casi più semplici (come nel caso del protocollo UDP), su un host il numero di porta è direttamente associato ad un socket e, quindi tale numero è sufficiente per identificare il socket opportuno nell'host destinazione; in casi più complicati, esso è usato in combinazione con altri dati (es.: TCP usa indirizzo IP e num. di porta del mittente + num. porta del destinatario per identificare l'opportuno socket su un dato host)

LIVELLO DI TRASPORTO: PROTOCOLLO UDP (I)

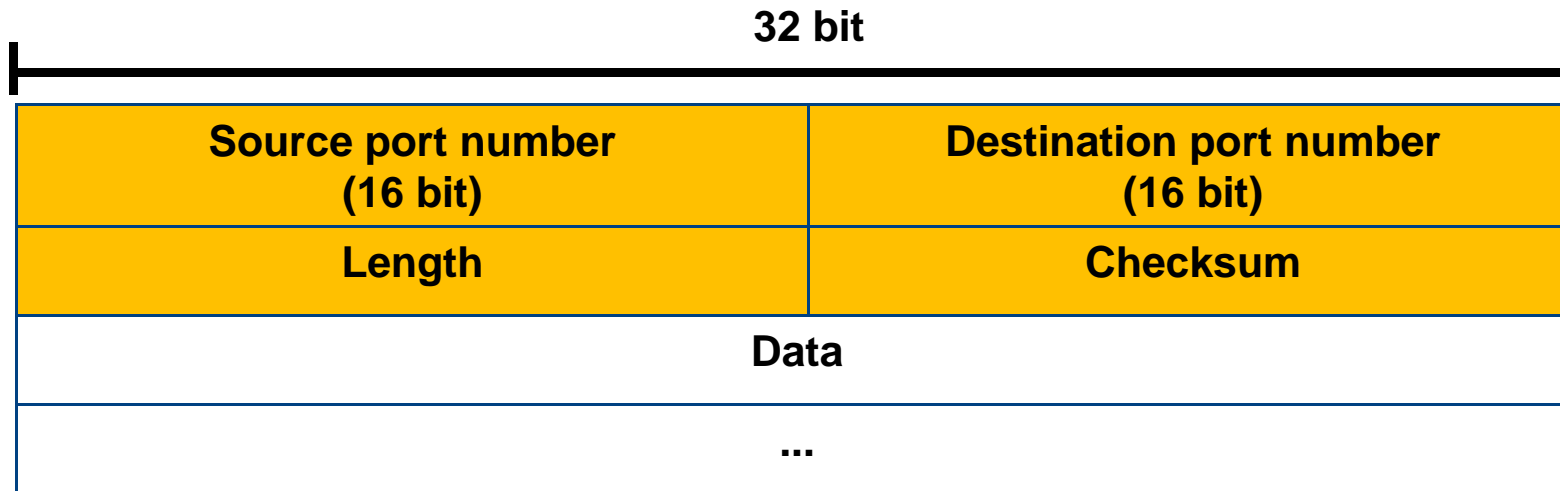


LIVELLO DI TRASPORTO: PROTOCOLLO UDP (II)

- **UDP (User Datagram Protocol)** è il più semplice protocollo Internet a Livello Trasporto
- Sostanzialmente è semplicemente l'estensione del servizio specificato dal IP per il recapito pacchetti da host a host con la specifica di un meccanismo che consente l'invio dei pacchetti da processo applicativo a processo applicativo
- → **anche UDP, come IP NON offre un servizio affidabile!**
- Realizzare un servizio affidabile di comunicazione processo applicativo-processo applicativo ha un costo (in termini di risorse computazionali come tempo, banda, ecc.)
- alcune applicazioni non necessitano di affidabilità assoluta per alcune di esse è più importante la velocità...ad esempio, le applicazioni di streaming, il DNS e altre utilizzano UDP

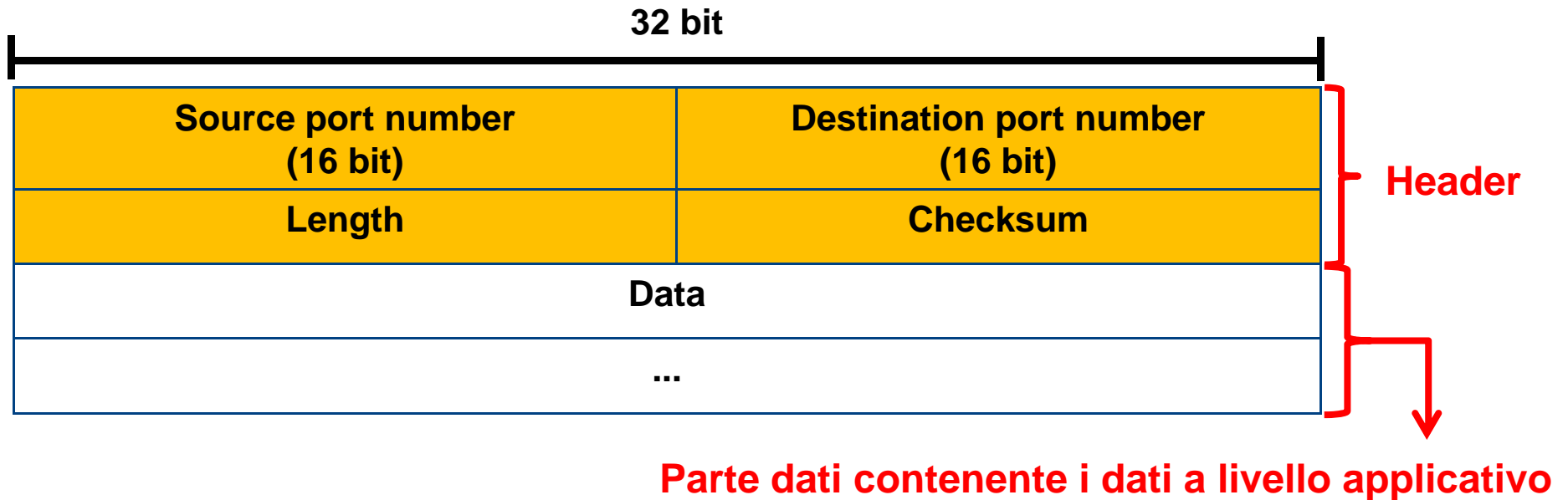
LIVELLO DI TRASPORTO: PROTOCOLLO UDP (III)

Segmento UDP



LIVELLO DI TRASPORTO: PROTOCOLLO UDP (III)

Segmento UDP

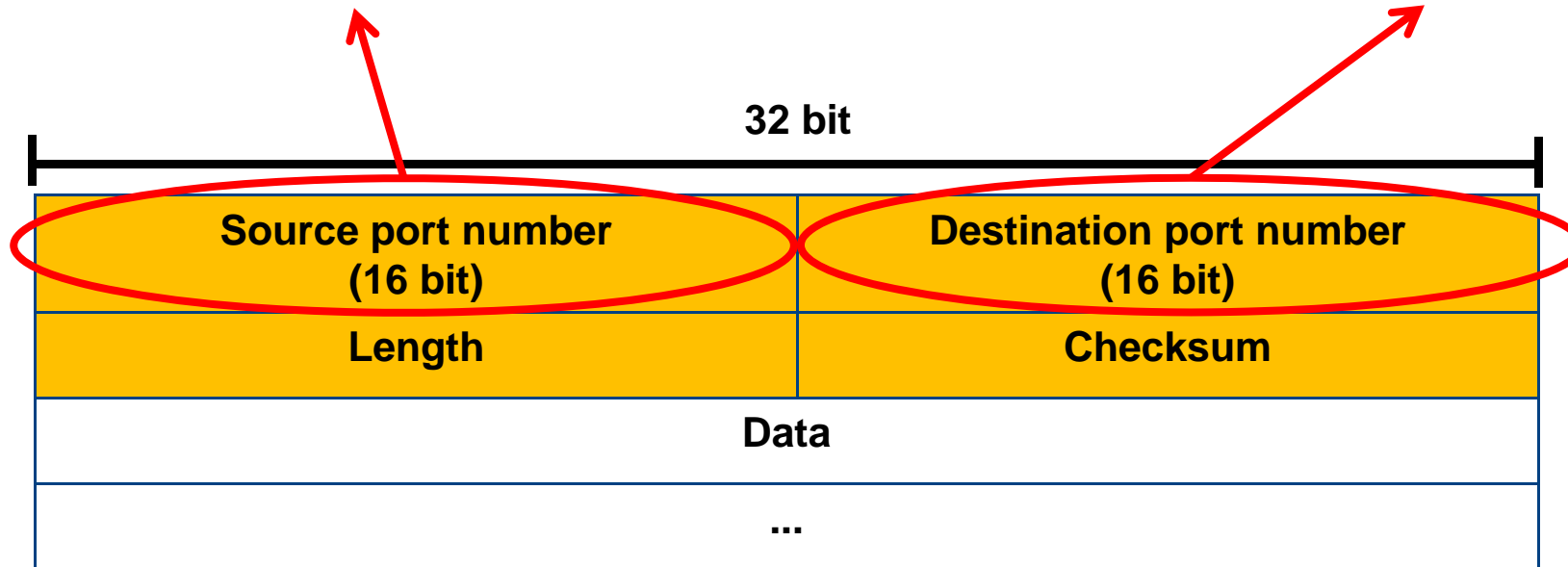


LIVELLO DI TRASPORTO: PROTOCOLLO UDP (III)

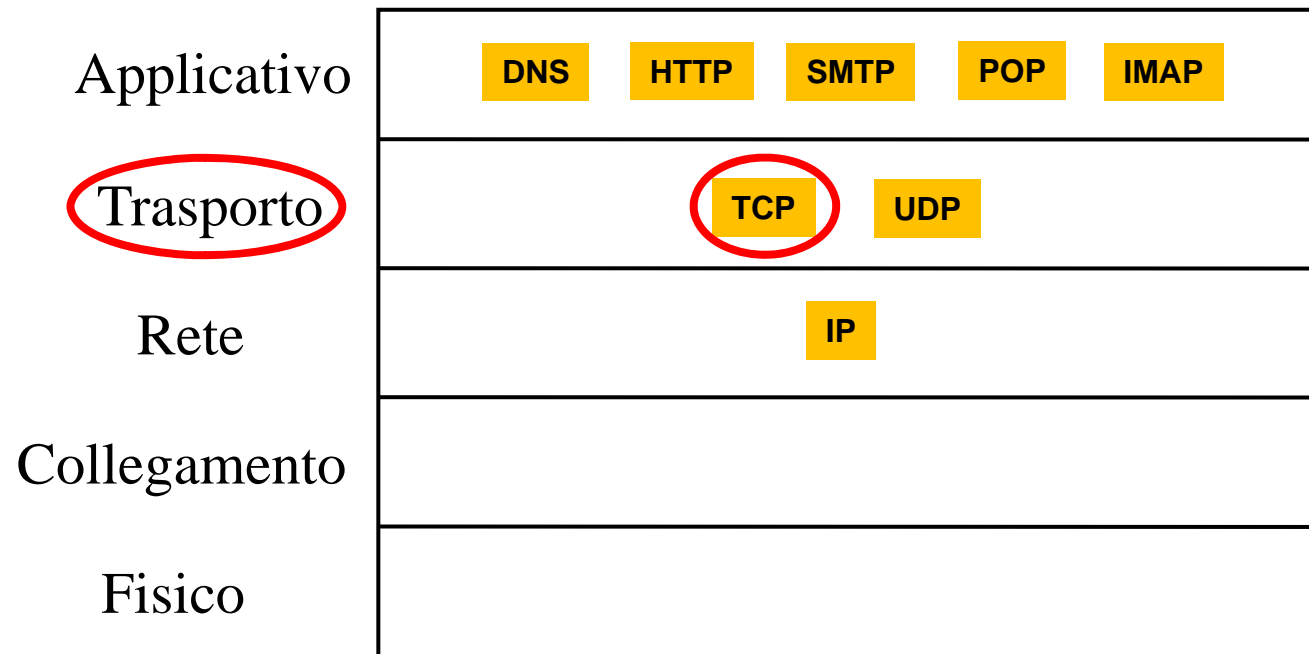
Segmento UDP

Numero di porta del mittente

Numero di porta del destinatario



LIVELLO DI TRASPORTO: PROTOCOLLO TCP (I)



LIVELLO DI TRASPORTO: PROTOCOLLO TCP (II)

- **TCP (Transmission Control Protocol)** è un protocollo a livello di trasporto che sfrutta il servizio offerto dal sottostante protocollo IP per specificare un servizio di invio dei pacchetti da processo applicativo a processo applicativo **affidabile**, inoltre specifica anche un meccanismo di controllo di congestione della rete, per evitarne il sovraccarico

- TCP assicura che, se tutto funziona correttamente, i pacchetti non vanno persi, arrivano integri e nell'ordine in cui sono stati spediti

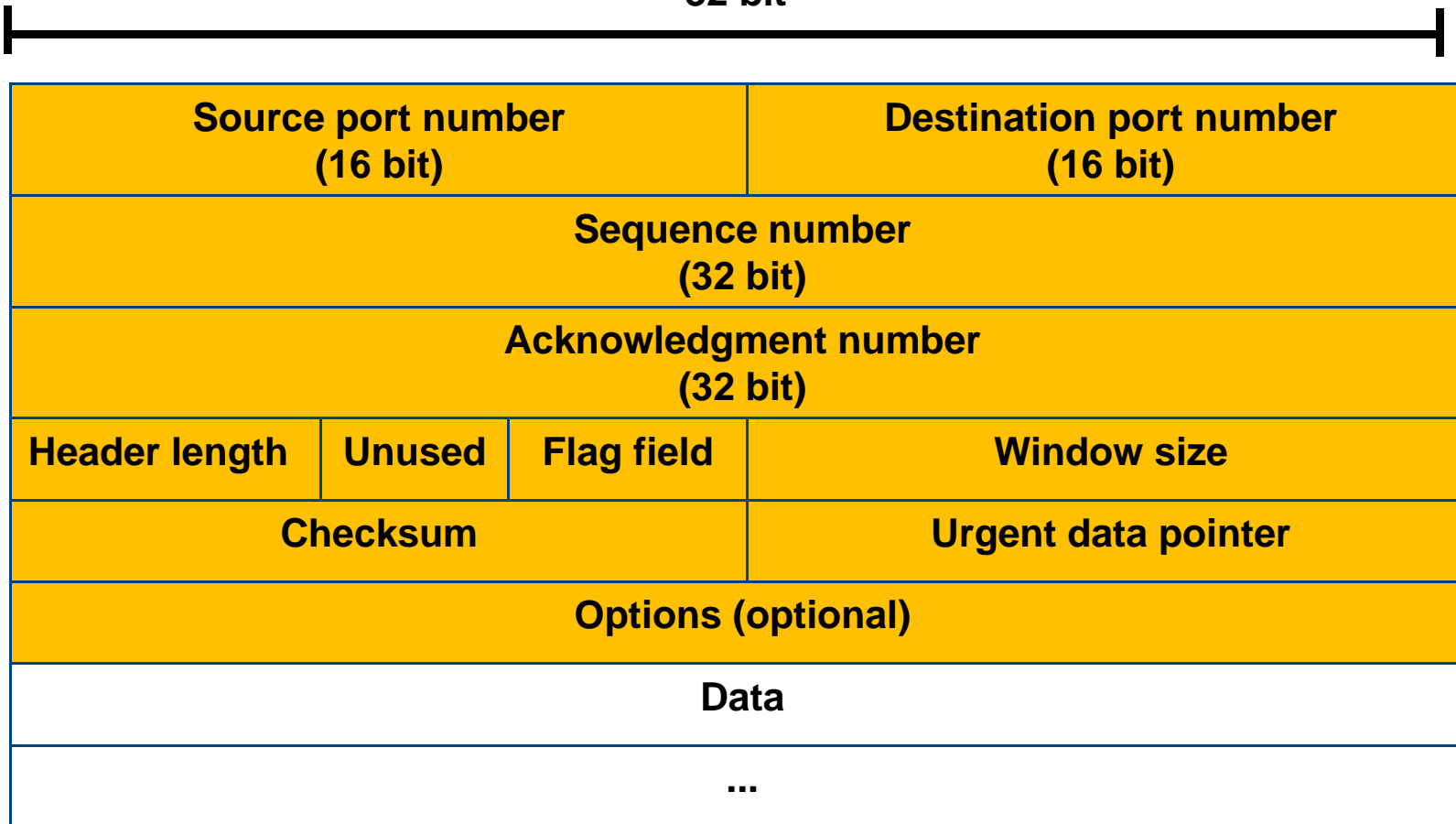
LIVELLO DI TRASPORTO: PROTOCOLLO TCP (III)

- I meccanismi per realizzare un servizio affidabile di comunicazione da processo applicativo a processo applicativo, sfruttando un servizio non affidabile di recapito pacchetti da host a host (si ricordi che IP non è affidabile) sono piuttosto complicati (ad esempio, prevedono l'invio al mittente di messaggi di ricevuta da parte del destinatario che riceve un segmento, il reiterno dell'invio dei segmenti da parte del mittente, quando necessario, lo scarto dei messaggi duplicati da parte del destinatario, ecc.): non esamineremo questi aspetti
- Molte applicazioni Internet ben note, per il trasporto dei pacchetti, utilizzano il protocollo TCP, ad es.: il Web (protocollo HTTP), la posta elettronica (protocolli SMTP, POP, IMAP), ecc.

LIVELLO DI TRASPORTO: PROTOCOLLO TCP (IV)

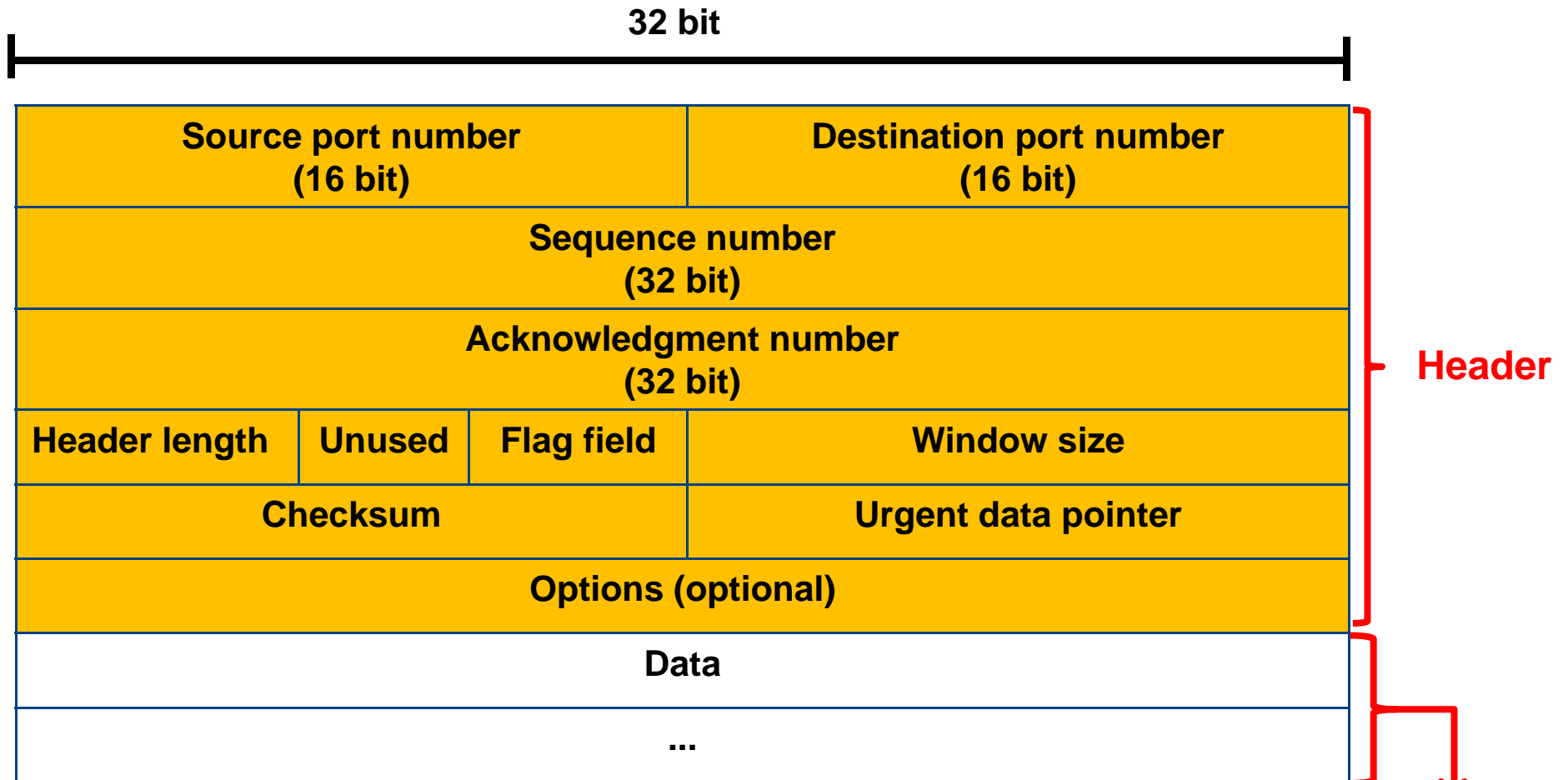
Segmento TCP

32 bit



LIVELLO DI TRASPORTO: PROTOCOLLO TCP (IV)

Segmento TCP



Parte dati contenente i dati a livello applicativo

62

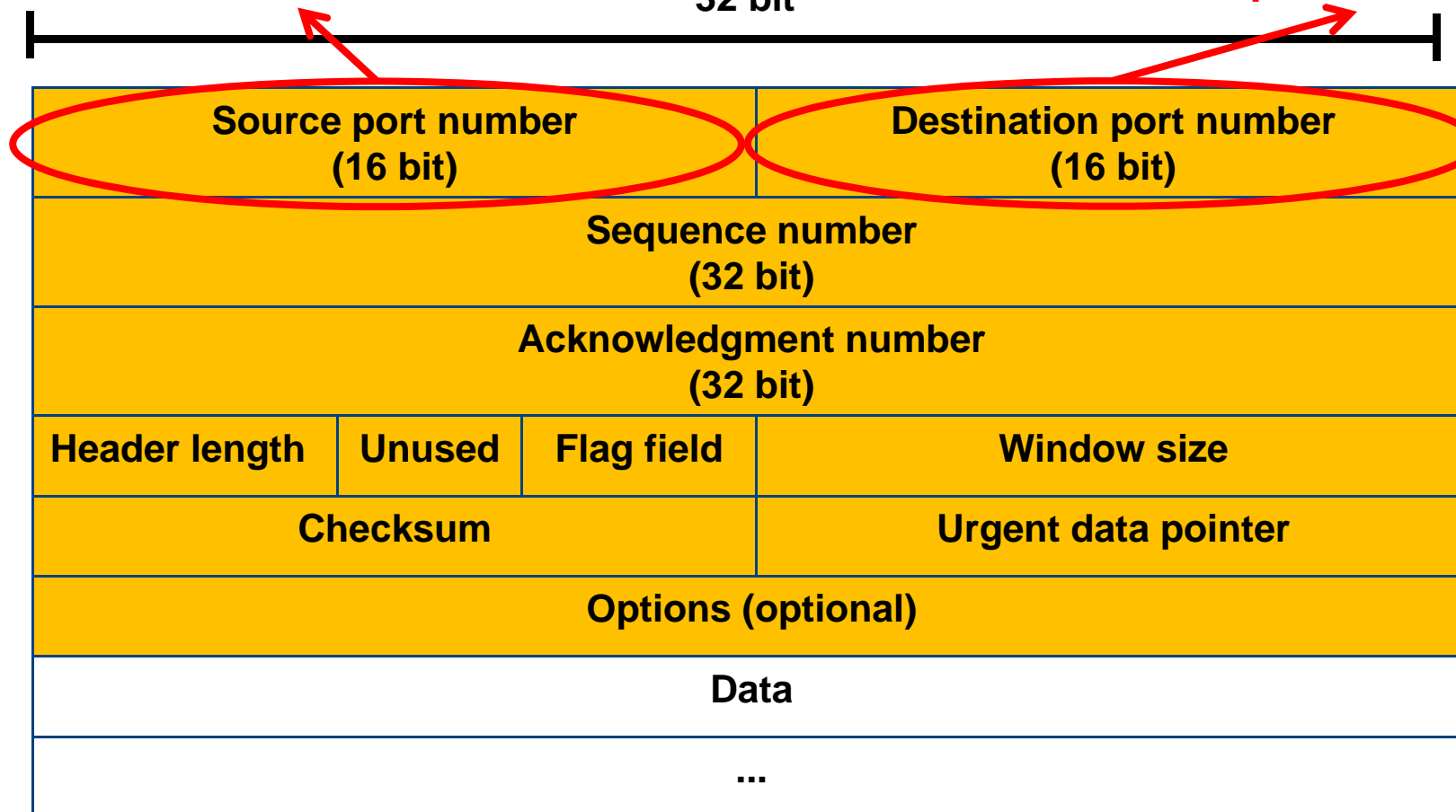
LIVELLO DI TRASPORTO: PROTOCOLLO TCP (IV)

Segmento TCP

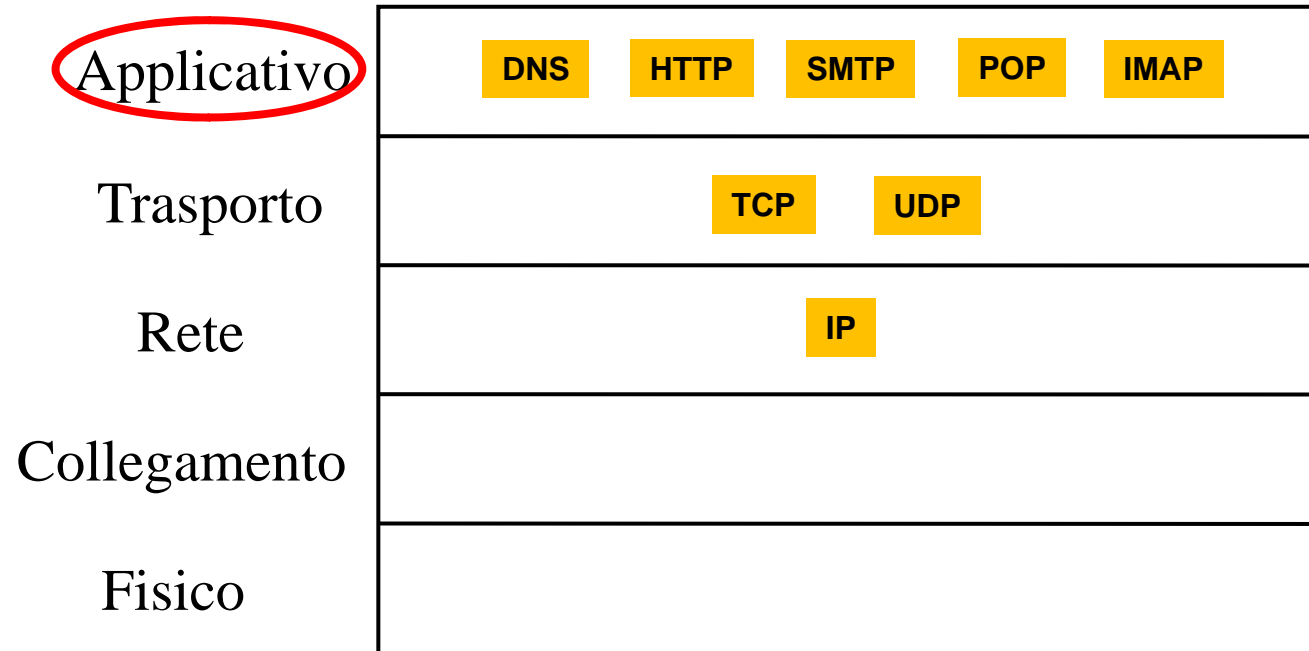
Numero di porta del mittente

32 bit

Numero di porta del destinatario



LIVELLO APPLICATIVO (I)



LIVELLO APPLICATIVO (II)

- Vi sono due principali modelli architetturali di riferimento per le applicazioni Internet: il modello client-server ed il modello peer-to-peer (P2P)
- Solitamente ogni applicazione Internet si conforma ad uno di questi due modelli, anche se vi sono applicazioni ibride che “mescolano” tali modelli
- **Il modello client-server** prevede l’esistenza su un host di un processo applicativo server, sempre attivo ed in attesa di richieste di servizi da parte di processi applicativi client (i client possono essere sempre attivi oppure no). Tipicamente l’interazione client-server vede la richiesta da parte del client di un servizio amministrato dal server a cui segue l’erogazione del servizio da parte del server (con messaggio di risposta) o l’invio al client da parte del server di un messaggio di rifiuto

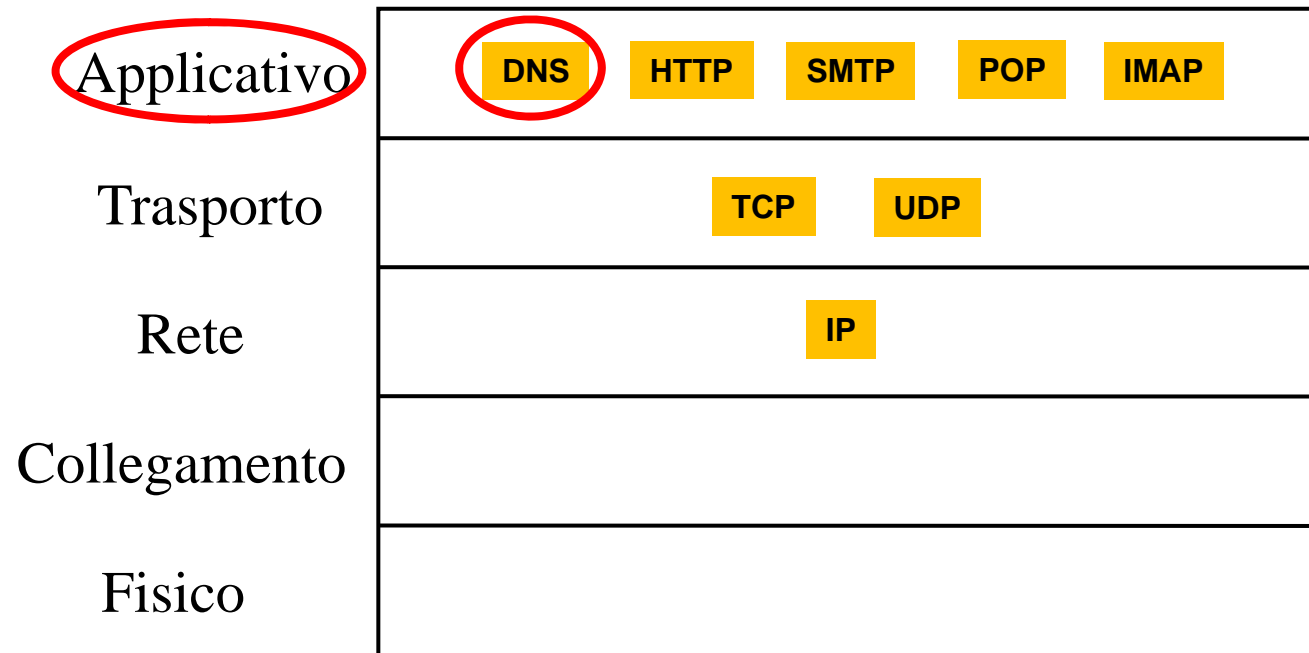
LIVELLO APPLICATIVO (III)

- **Nell'interazione client-server i due processi applicativi giocano ruoli diversi**
- **Tipicamente, in questo tipo di applicazioni i server devono sopportare un notevole carico di lavoro e sono gestiti da aziende o organizzazioni**
- **Molte applicazioni Internet sono realizzate secondo il paradigma client-server, es. il servizio DNS, Web, la posta elettronica, ecc.**

LIVELLO APPLICATIVO (IV)

- **Nel modello peer-to-peer (P2P), come suggerisce il nome, il rapporto non è più di tipo client-server, ma è un rapporto tra pari. La comunicazione avviene tra processi applicativi dello stesso tipo (chiamati “peers”, cioè “pari”). A tali processi applicativi non è richiesto di essere sempre attivi e pronti a rispondere ad eventuali richieste (come avviene, invece, per i server)**
- **Solitamente i processi P2P risiedono sugli host degli utenti finali e non su potenti server di aziende o organizzazioni**
- **Vi sono varie applicazioni Internet realizzate secondo il paradigma P2P, es.: applicazioni di telefonia (cosiddetta “VoIP”, Voice over IP) come Skype, applicazione per la condivisione e distribuzione di file, come BitTorrent, eMule, ecc.**

LIVELLO APPLICATIVO: DNS (I)



LIVELLO APPLICATIVO: DNS (II)

- **DNS: Domain Name System** è un protocollo di livello applicativo che, tipicamente usa UDP come servizio di trasporto
- Siccome ricordare gli indirizzi IP a 32 o 128 bit è pressoché impossibile e, per quelli a 32 bit, ricordarli nella forma “dot notation” è comunque difficile, esiste un servizio Internet che consente di usare nomi simbolici (molto più facili da ricordare per le persone) per identificare gli host in Internet
- **es.: pippo.acquisti.acme.com** è più facile da ricordare di
11010001100010000100111100100010 e anche di
209.136.79.34

LIVELLO APPLICATIVO: DNS (III)

- Il servizio DNS traduce un nome di host nel corrispondente indirizzo IP (che è quello effettivamente utilizzato dal protocollo IP)
- pippo.acquisti.acme.com → 11010001100010000100111100100010
- ...in realtà il DNS offre anche altri servizi, ma quello di traduzione da indirizzi nomi di host ad indirizzi IP è sicuramente quello più significativo

LIVELLO APPLICATIVO: DNS (IV)

- Internet è diffusa a livello planetario → non è pensabile un unico spazio di nomi di host
- → approccio gerarchico basato sul concetto di dominio

LIVELLO APPLICATIVO: DNS (V)

Dominio. Gli elaboratori sono raggruppati in domini e ad ognuno di essi e' assegnato un nome logico, che deve essere unico solamente all'interno del dominio (quindi elaboratori in domini diversi possono condividere lo stesso nome logico).

Anche ai domini sono assegnati dei nomi logici e a loro volta i domini possono essere raggruppati in domini a livello gerarchico superiore.

Se il dominio *A* viene inserito nel dominio *B* allora si dice che *A* è un *sotto-dominio* di *B*.

LIVELLO APPLICATIVO: DNS (VI)

Esempio: definiamo il dominio *acme* e i sotto-domini *acquisti* e *vendite* e poi assegniamo dei nomi agli elaboratori dei due sotto-domini (reparti) in modo indipendente. Così potremo avere che i due elaboratori nel dominio *acquisti* vengono chiamati *pippo* e *pluto* e così pure nel dominio *vendite*.

```
pippo.acquisti.acme  
pippo.vendite.acme
```

Si parte cioè dal nome locale dell'elaboratore, specificando il dominio cui appartiene e così via verso i domini di livello gerarchico superiore.

LIVELLO APPLICATIVO: DNS (VII)

DOMINI DI PRIMO LIVELLO: Un dominio per ogni nazione: come **it** (Italia), **fr** (Francia), **us** (Stati Uniti), **uk** (United Kingdom), **ch** (Svizzera), **de** (Germania), **br** (Brasile), ...

una serie di domini trans-nazionali, legati al tipo di attività svolta dalle organizzazioni/enti/aziende in rete,

edu, per enti di tipo educativo (università, scuole, ...)

int, per organizzazioni internazionali

net, per aziende di servizi in rete

org, per organizzazioni non commerciali

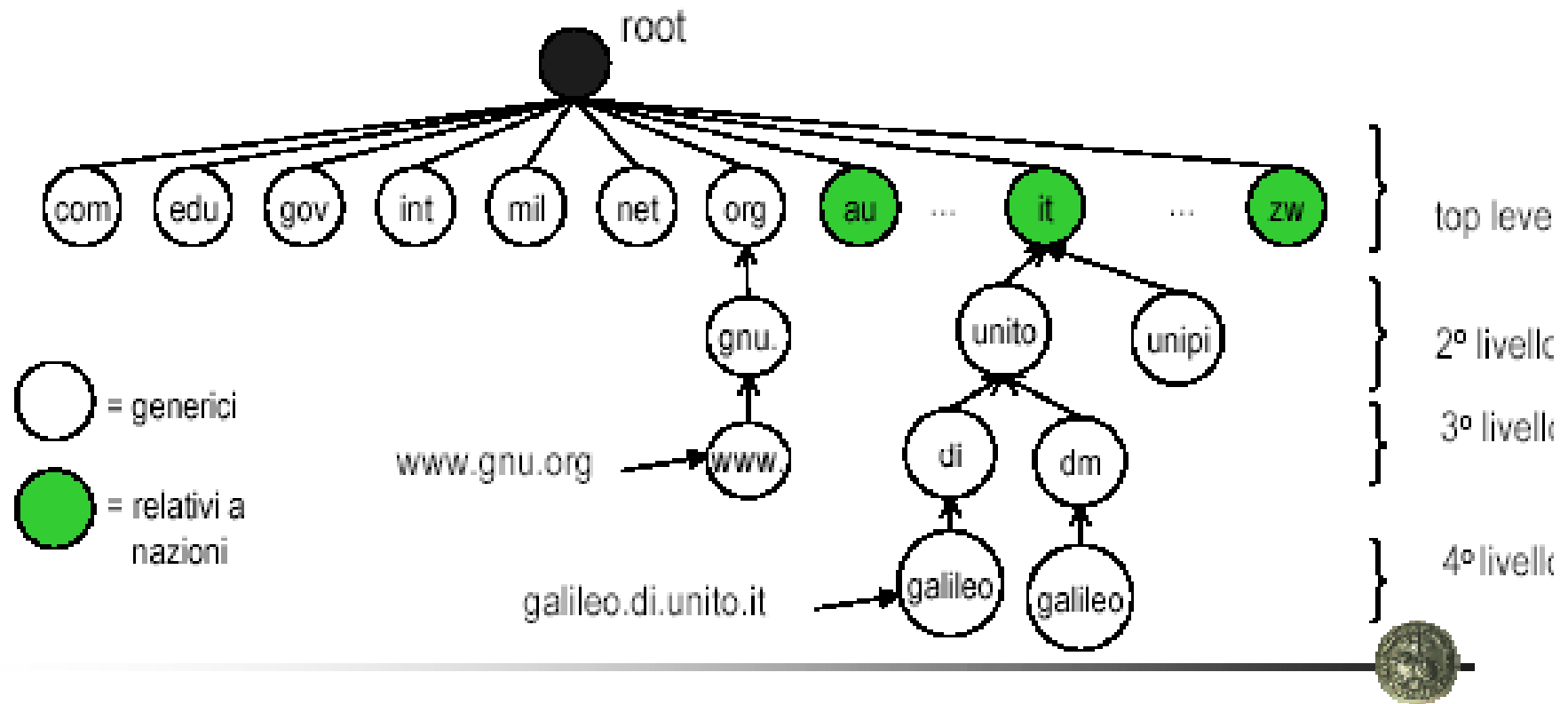
com, per aziende commerciali

gov, per enti di tipo governativo

mil, per enti militari

a questi si sono aggiunti poi via via altri domini di primo livello, ad esempio **biz** (business), ...

LIVELLO APPLICATIVO: DNS (VIII)



LIVELLO APPLICATIVO: DNS (IX)

www.unito.it

*Indirizzo dell'elaboratore **www** nel sotto-dominio **unito** (Università di Torino), nel dominio **it**.
Convenzionalmente in Italia le Università sono identificate dal dominio formato dal prefisso **uni** seguito da un identificativo dell'Ateneo, spesso la provincia (**unige, unina, unive, ...**).*

doc.ic.ac.uk

*Indirizzo dell'Imperial College di Londra. Il dominio **uk** indica le reti inglesi, il sotto-dominio **ac** indica i nodi accademici inglesi, il sotto-dominio **ic** indica l'Imperial College di Londra, ed infine il sotto-dominio **doc** indica il Department of Computing.*

LIVELLO APPLICATIVO: DNS (X)

- **Esistono organizzazioni apposite (nazionali ed internazionali) che gestiscono i domini di primo livello**
- **La gestione dei domini di livello inferiore è delegata all'organizzazione cui fa capo il dominio; es.: il dominio di secondo livello `unito.it` è gestito dall'Università di Torino che, fra le altre cose, ha la facoltà di suddividerlo in ulteriori (sotto)domini ed, eventualmente, delegare la gestione di tali (sotto)domini. Es. `unito.it` contiene i (sotto)domini `di.unito.it` e `dm.unito.it` (e altri) che sono gestiti, rispettivamente, dal dipartimento di Informatica e da quello di Matematica dell'Università di Torino**
- **N.B. Non c'è alcuna corrispondenza diretta fra i domini e le reti di elaboratori di Internet: elaboratori della stessa rete fisica possono trovarsi in domini diversi; lo stesso dominio può raggruppare elaboratori di diverse reti fisiche**

LIVELLO APPLICATIVO: DNS (XI)

- Come avviene la traduzione da indirizzi logici a indirizzi fisici?
- Il servizio di traduzione è offerto da un gran numero di server organizzati gerarchicamente (su ogni nodo che funge da server DNS esiste un sw server DNS)
- su ogni nodo esiste un sw client DNS
- nessun server conosce tutte le possibili traduzioni, ma puo` chiedere ad altri server quel che non sa

LIVELLO APPLICATIVO: DNS (XII)

- Es. l'elaboratore elios.unito.it chiede una connessione a www.media.mit.edu: come si determina l'indirizzo fisico di quest'ultimo elaboratore?

RISALITA NELLA GERARCHIA

1. Il client DNS di elios.unito.it si rivolge al server DNS che gli è stato specificato, supponiamo che sia quello del suo dominio "unito.it"
2. il server DNS di "unito.it" si rende conto che non è un elaboratore del suo dominio
3. il server DNS di "unito.it" si rivolge al (cioè "diventa client del") server DNS del suo sopra-dominio "it"

LIVELLO APPLICATIVO: DNS (XIII)

2. il server DNS di “it” si rende conto che non si tratta di un indirizzo del suo dominio e si rivolge al server DNS della radice

DISCESA NELLA GERARCHIA

2. il server DNS della radice passa la richiesta al server DNS di “edu”
3. il server DNS di “edu” riconosce l’indirizzo come di sua competenza, ma, non essendo macchina locale, passa la richiesta al server DNS di “mit.edu”
4. il server DNS di “mit.edu” passa la richiesta al server DNS di “media.mit.edu”;

LIVELLO APPLICATIVO: DNS (XIV)

5. il server DNS di “media.mit.edu” riconosce l’indirizzo come quello di una sua macchina locale e recupera in una tabella l’indirizzo IP corrispondente

RISALITA NELLA GERARCHIA

6. il server DNS di “media.mit.edu” comunica la traduzione al server DNS di “mit.edu”
7. il server DNS di “mit.edu” comunica la traduzione al server DNS di “edu”
8. il server DNS di “edu” comunica la traduzione al server DNS della radice

LIVELLO APPLICATIVO: DNS (XV)

DISCESA NELLA GERARCHIA

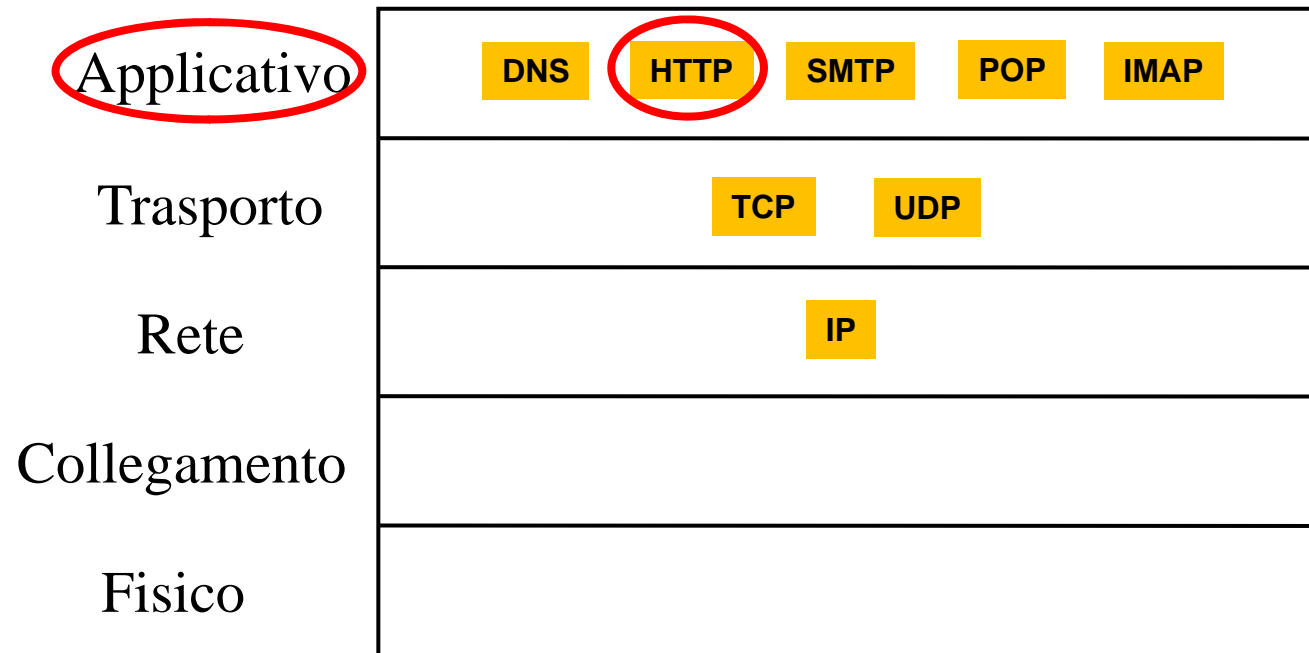
9. il server DNS della radice comunica la traduzione al server DNS di “it”
10. server DNS di “it” comunica la traduzione al server DNS di “unito.it”
11. il server DNS di “unito.it” comunica la traduzione al client DNS di elios.unito.it
12. i protocolli TCP e IP su elios.unito.it possono utilizzare l'indirizzo fisico così ottenuto

LIVELLO APPLICATIVO: DNS (XVI)

N.B.: il precedente esempio costituisce una semplificazione della situazione reale, in quanto:

- assume che vi sia un server DNS per ogni (sotto)dominio (es. un server per “di.unito.it”, uno per “unito.it”, uno per “it”, uno per la radice, ecc): in realtà, uno stesso server può gestire direttamente un dominio e parte delle gerarchie dei suoi sotto-domini
- descrive un funzionamento “ricorsivo” (in cui un server fornisce al client la risposta, dandogliela esso stesso, se è in grado, o chiedendola ad un altro server): esiste anche un funzionamento “iterativo”, in cui quando un server non è in grado di fornire direttamente la risposta, indica al client quale altro server contattare: nella realtà sono frequenti comportamenti ibridi (ricorsione+iterazione)
- non considera il meccanismo di “caching”: nella realtà, per velocizzare il processo, si utilizzano delle “cache” sia nei client che nei server DNS

LIVELLO APPLICATIVO: IL WEB - PROTOCOLLO HTTP (I)



LIVELLO APPLICATIVO: IL WEB - PROTOCOLLO HTTP (II)

- Come il DNS, HTTP (HyperText Transfer Protocol) e` un protocollo applicativo realizzato secondo il modello client/server
- usa TCP come servizio di trasporto
- E` alla base del Web (WWW, World Wide Web)
- Il Web e` un enorme ipertesto multimediale distribuito le cui pagine sono memorizzate in milioni di server sparsi in tutto il mondo e collegati attraverso la rete Internet
- Attenzione! Internet e Web sono due cose distinte!

LIVELLO APPLICATIVO: IL WEB - PROTOCOLLO HTTP (III)

- Lo standard Web prevede la costruzione di pagine e ipertesti mediante il linguaggio HTML (HyperText Markup Language)
- I documenti HTML sono visualizzati dai browser (che sono programmi applicativi, es. Netscape Communicator, Internet Explorer)
- Quando un browser su un nodo A vuole recuperare un documento HTML su un diverso nodo B, lo richiede al server Web del nodo B → interazione client/server fra browser e server Web
- N.B: Una cosa è il browser, altra cosa è il protocollo HTTP: i diversi browser possono differenziarsi anche notevolmente gli uni dagli altri, ma essi, per poter navigare il Web, devono tutti attenersi al protocollo HTTP

LIVELLO APPLICATIVO: IL WEB - PROTOCOLLO HTTP (IV)

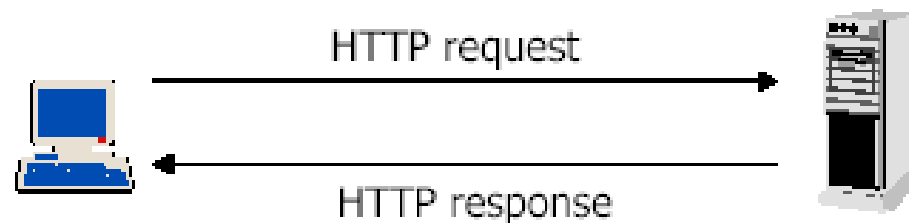
- Come avviene l'interazione fra browser (client) e server Web?
- Essa si fonda sul protocollo HTTP

PROTOCOLLO HTTP

- Il cuore di un server Web è il server HTTP
- Con questo protocollo, client HTTP (browser) e server HTTP si scambiano richieste e risposte

LIVELLO APPLICATIVO: IL WEB - PROTOCOLLO HTTP (V)

- L'utente richiede una pagina residente su un server
- Il browser richiede una connessione TCP con il server
- Il server accetta la connessione iniziata dal browser
- Il browser ed il server si scambiano messaggi
- La connessione viene chiusa



LIVELLO APPLICATIVO: IL WEB - PROTOCOLLO HTTP (VI)

- Una richiesta HTTP e` formata da 2 parti: una parte dati (quasi sempre vuota, tranne nel caso di invio informazioni, ad esempio, a seguito di compilazione modulo on line) piu` un header

LIVELLO APPLICATIVO: IL WEB - PROTOCOLLO HTTP (VII)

HEADER RICHIESTA HTTP

es.:

Header

```
GET /~magro/teaching.html HTTP/1.1
User-agent: Mozilla/4.0
Accept: text/html, image/gif, image/jpeg
Host: pianeta.di.unito.it
...
```

LIVELLO APPLICATIVO: IL WEB - PROTOCOLLO HTTP (VIII)

- Una risposta HTTP è formata da due parti: una parte dati (contenente il documento richiesto, se è stato possibile inviarlo) più un header

LIVELLO APPLICATIVO: IL WEB - PROTOCOLLO HTTP (IX)

RISPOSTA HTTP

es.:

Header

```
HTTP/1.1 200 OK  
Date: Friday, 10 Nov 2011, 12:00:12 GMT  
Server: Apache/1.3.0 (Unix)  
Last-Modified: ...  
Content-Type: text/html  
...
```

Body

```
<HTML>  
<HEAD>  
...  
</HEAD>  
...
```

LIVELLO APPLICATIVO: IL WEB - PROTOCOLLO HTTP (X)

I codici di risposta notificano al browser le informazioni relative al successo o al fallimento della connessione:

1xx: informazione (richiesta ricevuta, l'elaborazione sta continuando)

2xx: successo (l'azione è stata ricevuta con successo, compresa e accettata)

3xx: redirectione (per portare a termine la richiesta sono richieste ulteriori azioni)

4xx: errore del client (la richiesta contiene errori di sintassi o non può essere soddisfatta)

5xx: errore del server (il server non è stato in grado di soddisfare una richiesta apparentemente valida)

[categorie di codici tratte da L.L. Peterson, B. S. Davie, *Reti di calcolatori*, Apogeo, 2004]

LIVELLO APPLICATIVO: IL WEB - PROTOCOLLO HTTP (XI)

- **URL (Uniform Resource Locator):** e` l'indirizzo specificato nella barra indirizzi del browser
- è formato da 3 parti:
 1. il protocollo che si vuole utilizzare per la connessione, che identifica il tipo di server cui ci si vuole connettere;
 2. l'indirizzo (logico o IP) del server;
 3. la specifica risorsa che si richiede.
- **<http://www.unito.it/~magro/teaching.html>**
- - il server cui connettersi è un server HTTP
- - l'indirizzo del server è **www.unito.it**
- - la risorsa richiesta è il documento (la pagina Web) **/~magro/teaching.html**

LIVELLO APPLICATIVO: IL WEB - PROTOCOLLO HTTP (XII)

- **Memoria cache del browser:** I browser attuali hanno una memoria cache (in RAM e/o sul disco) in cui sono memorizzate le ultime pagine caricate dall'utente. Quando l'utente richiede il caricamento di una pagina, il browser prima di richiederla al server verifica se essa è già disponibile nella cache. In questo modo le pagine ad accesso frequente vengono caricate velocemente, con il rischio, tuttavia, di caricare versioni non aggiornate (l'utente può comunque forzare il caricamento dal server, inoltre, il server può indicare, per ogni pagina inviata, la data di scadenza).

LIVELLO APPLICATIVO: IL WEB - PROTOCOLLO HTTP (XIII)

Proxy server

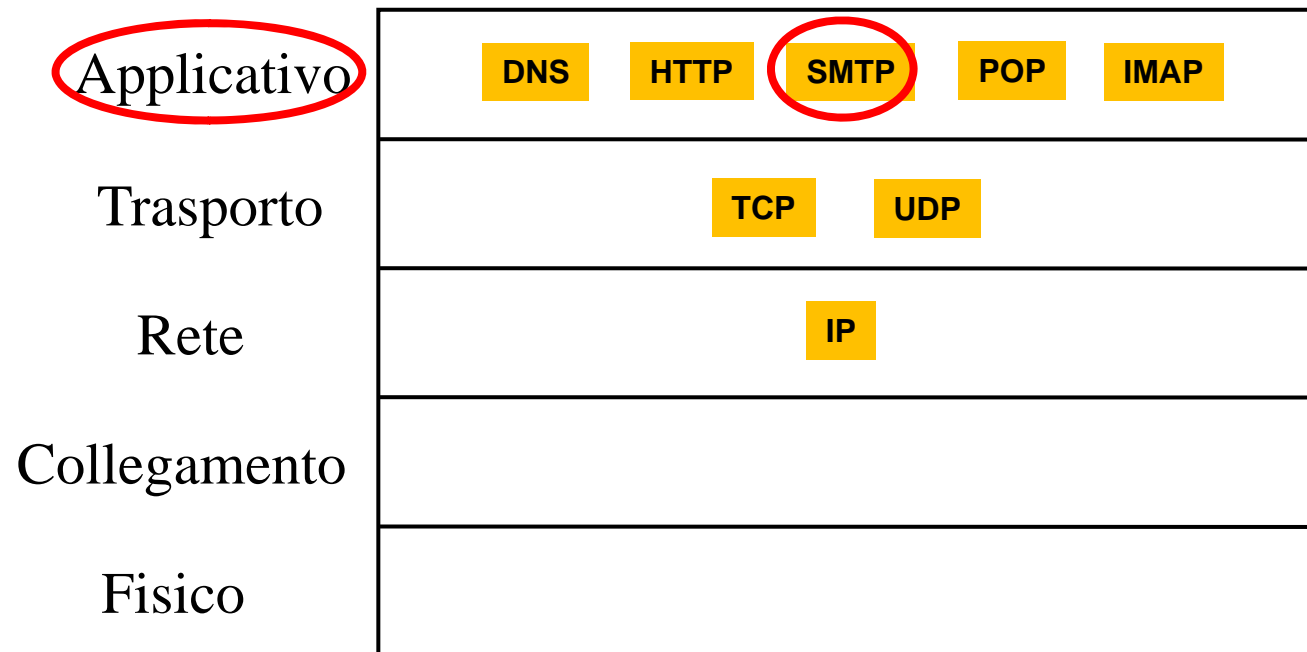
- I proxy server sono dei nodi speciali all'interno della rete che possono svolgere il ruolo di ponti nella connessione in rete. In condizioni normali una richiesta di connessione da un client ad un server porta ad una connessione diretta tra client e server per il trasferimento dei dati. Nel caso in cui il client decida di usare un proxy server, la connessione viene effettuata in due passi
- 1.- il client si connette al proxy server cui delega il compito di scaricare le informazioni;
- 2.- il proxy server fungerà quindi da server per il client dell'utente (il proxy effettua caching, quindi se dispone dell'informazione richiesta, la invia al client, senza contattare il server di destinazione) e da client verso il server da cui si deve prelevare l'informazione che viene passata al client (se non dispone dell'informazione, il proxy la richiede al server di destinazione).

LIVELLO APPLICATIVO: IL WEB - PROTOCOLLO HTTP (XIV)

Cookies

- Un cookie è un file di testo che molti server producono e memorizzano sul calcolatore client (quindi sull'elaboratore con cui si naviga in rete) e che, in un certo senso, identifica l'utente. Quando un utente torna su un server che ha già visitato e che ha scritto un cookie, nella richiesta HTTP il valore del cookie viene inviato automaticamente al server che è così in grado di "riconoscere" l'utente, magari presentando informazioni personalizzate. Si noti che i cookie vengono memorizzati in una cartella all'interno del proprio elaboratore e possono essere facilmente rimossi.

LIVELLO APPLICATIVO: LA POSTA ELETTRONICA – PROTOCOLLO SMTP (I)



LIVELLO APPLICATIVO: LA POSTA ELETTRONICA – PROTOCOLLO SMTP (II)

- **SMTP: Simple Mail Transfer Protocol**
- **Consente comunicazione una-a-uno (o uno-a-molti) fra utenti Internet, attraverso invio e ricezione di messaggi di posta elettronica (e-mail)**
- **Messaggi inviati da mittente vengono recapitati nella “casella postale” (elettronica) del destinatario che può leggerli, rispondere, inviarne altri, cancellarli, organizzarli in directory**
- **Come DNS e HTTP, SMTP è realizzato secondo il paradigma client/server**
- **usa TCP come servizio di trasporto**

LIVELLO APPLICATIVO: LA POSTA ELETTRONICA – PROTOCOLLO SMTP (III)

Mailbox e indirizzi

- Per poter ricevere e-mail, ad un utente deve essere assegnata una casella postale elettronica (*mailbox*) che è un luogo (es. un file su disco) in cui i messaggi a lui/lei destinati vengono depositati
- Il servizio di posta elettronica può essere attivato in qualunque dominio (es. unito.it, di.unito.it, ...), in cui vi sia una macchina su cui gira un *mail server*
- Affinché un messaggio di posta elettronica possa raggiungere il destinatario (o i destinatari), occorre specificare l'indirizzo della mailbox di tale destinatario
- Un indirizzo di posta elettronica ha generalmente il seguente formato:
nomemailbox@nomedominio
es: magro@di.unito.it, diego.magro@unito.it

LIVELLO APPLICATIVO: LA POSTA ELETTRONICA – PROTOCOLLO SMTP (IV)

Formato dei messaggi e-mail

- Ogni messaggio di e-mail è un testo di caratteri ASCII, composto da due parti: *header* e *body*
- Il body contiene il messaggio vero e proprio
- Lo header contiene informazioni sul messaggio

LIVELLO APPLICATIVO: LA POSTA ELETTRONICA – PROTOCOLLO SMTP (V)

Formato dei messaggi e-mail

Es.:

From: caio.rossi@acme.com

To: diego.magro@unito.it

Date: Mon, 11 Jul 05 10:33:54 EST

Subject: riunione progetto

Header

Ciao,

sono disponibile per la riunione
sia il 13 che il 14.

Fammi sapere.

Caio

Body

LIVELLO APPLICATIVO: LA POSTA ELETTRONICA – PROTOCOLLO SMTP (VI)

Formato dei messaggi e-mail

- Originariamente, le e-mail potevano contenere solo testo (in codice ASCII) e SMTP, ancor oggi, assume che i messaggi siano di tipo testuale
- Tuttavia, attualmente, siamo in grado di spedire in una e-mail anche *allegati (attachments)* di tipo non testuale (immagini, documenti Word, file audio, ecc.): come è possibile ciò?
- Si “nasconde” al protocollo SMTP la presenza di allegati, “fingendo che il messaggio sia soltanto testuale”: i dati non testuali vengono codificati in forma testuale. Le applicazioni che implementano SMTP non interpretano questi dati, ma li trattano come semplice testo: saranno applicazioni specifiche ad interpretare tali dati (e, ad esempio, a riconoscere che una parte di messaggio, che per SMTP era semplice testo, è in realtà un’immagine e, quindi, a visualizzarla)

LIVELLO APPLICATIVO: LA POSTA ELETTRONICA – PROTOCOLLO SMTP (VII)

Formato dei messaggi e-mail

- L'estensione che consente il “trucco” di codificare elementi non testuali in sequenze di caratteri è lo standard MIME (Multipurpose Internet Mail Extension)
- Questo standard prevede che il body di un messaggio possa essere suddiviso in varie parti, ciascuna contenente informazioni codificate secondo certe regole
- Nello header viene specificato che il messaggio si attiene allo standard MIME (più altre informazioni, quale la sequenza di caratteri usata per separare le varie parti del messaggio), nel body alcune righe aggiuntive specificano quale codifica è stata usata (in modo che le applicazioni opportune possano effettuare la decodifica)

LIVELLO APPLICATIVO: LA POSTA ELETTRONICA – PROTOCOLLO SMTP (VIII)

Es.: MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="- - - - -324F9G7AB8HFFFDJ1"
From: caio.rossi@acme.com
To: diego.magro@unito.it
Date: Fri, 15 Jul 05 11:10:27 EST
Subject: ecco la foto!

Header

- - - - -324F9G7AB8HFFFDJ1
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7 bit

ecco la foto dell'ultima riunione
Ciao, Caio

Body

- - - - -324F9G7AB8HFFFDJ1
Content-Type: image/jpeg
Content-Transfer-Encoding: base64

<codifica dell'immagine jpeg allegata>

LIVELLO APPLICATIVO: LA POSTA ELETTRONICA – PROTOCOLLO SMTP (IX)

Trasferimento e-mail

- Vediamo un esempio semplificato di un trasferimento di e-mail fra il mittente caio.rossi@acme.com e il destinatario diego.magro@unito.it
1. Caio Rossi compone l'e-mail usando un'applicazione come Eudora, Microsoft Outlook, Mozilla Thunderbird, pine, elm,...
 2. L'applicazione invia l'e-mail al server SMTP del mittente (utilizzando il protocollo SMTP) su acme.com
 3. l'e-mail viene mantenuta nell' *area di spool* su acme.com
 4. il server SMTP su acme.com interroga il DNS per ottenere l'indirizzo IP del server SMTP nel dominio unito.it
 5. il server SMTP su acme.com stabilisce una connessione TCP/IP con il server SMTP su unito.it
 6. l'e-mail viene trasferita a destinazione (ciò avviene con una sequenza di messaggi scambiati fra server SMTP su acme.com e server SMTP su unito.it, secondo il protocollo...SMTP (guarda caso!))

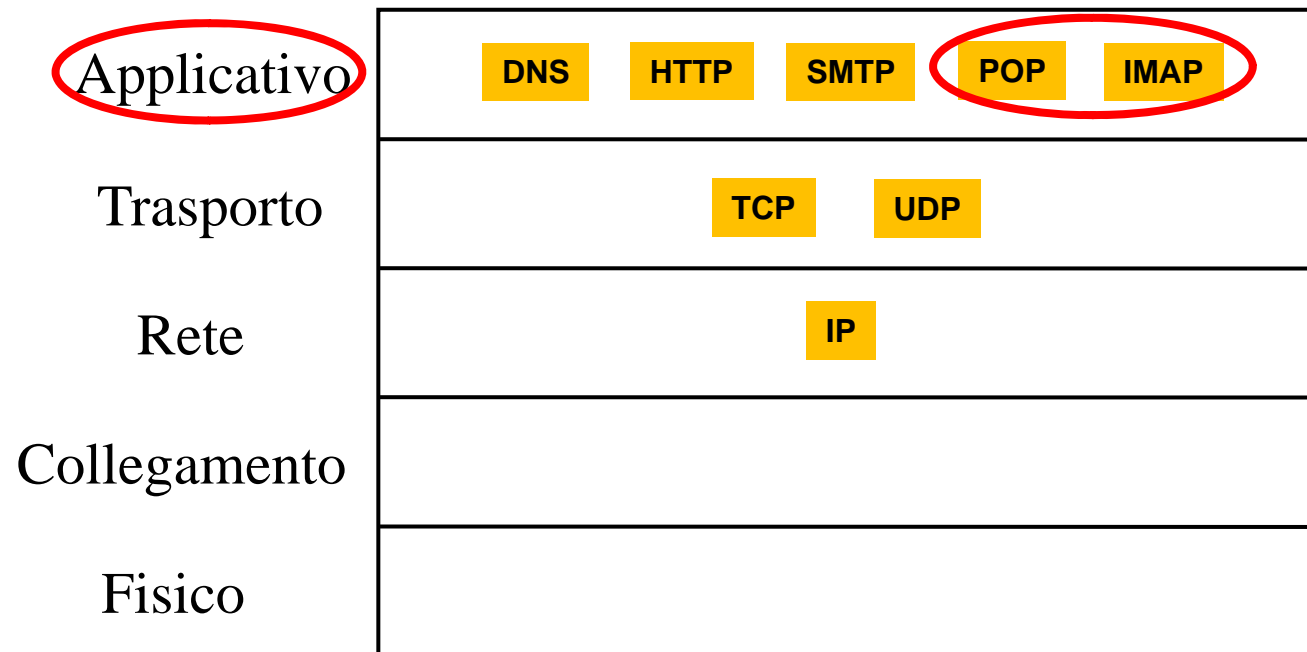
LIVELLO APPLICATIVO: LA POSTA ELETTRONICA – PROTOCOLLO SMTP (X)

Trasferimento e-mail (cont.)

7. il server SMTP su unito.it deposita l'e-mail nella mailbox del destinatario (Diego Magro)
8. la connessione TCP/IP viene chiusa
9. il server SMTP su acme.com elimina l'e-mail dalla propria area di spool

... se qualcosa non funziona e non è possibile inviare l' e-mail a destinazione, l'e-mail non viene eliminata dall'area di spool e il server SMTP su acme.com continua a tentare l'invio per un certo periodo di tempo (solitamente 3 giorni); se, in quel periodo temporale, nessun tentativo va a buon fine, il server SMPT notifica il fallimento al mittente (Caio Rossi)

LIVELLO APPLICATIVO: LA POSTA ELETTRONICA – PROTOCOLLI POP E IMAP(I)



LIVELLO APPLICATIVO: LA POSTA ELETTRONICA – PROTOCOLLI POP E IMAP(I)

- La mailbox risiede sulla macchina su cui gira il server SMTP e, in genere, questa macchina non coincide con quella da cui l'utente legge e scrive le proprie e-mail
- → l'utente, generalmente, accede alla propria mailbox avvalendosi di un'applicazione software che gira su una macchina connessa in rete a quella in cui risiede la propria mailbox
- L'applicazione per la lettura delle e-mail deve dialogare “con la macchina” su cui risiede la mailbox...più precisamente, l'applicazione per la lettura delle mail diventa client di un server che risiede sulla macchina in cui si trova la mailbox
- “client”, “server”, “dialogo” → protocollo! Sono due i possibili protocolli (cui corrispondono due server) con cui l'applicazione per la lettura delle mail può esplorare il contenuto della mailbox: POP3 e IMAP4
- POP3: consente di “scaricare” le e-mail in locale (e-mail disponibili anche quando l'utente non è connesso in rete, ma possibile dispersione delle mail se si usano più macchine per leggerle)
- IMAP4: mantiene i messaggi presso il server (evita la dispersione delle e-mail, ma queste sono disponibili solo quando l'utente è connesso)