RETI DI CALCOLATORI

Autunno 2018

docente: Laura Ricci

laura.ricci@unipi.it

Lezione 2:
INTERNET: PRINCIPI DI PROGETTAZIONE
24/09/2018

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

INTERNET: SCELTE DI PROGETTO

Cosa affronteremo in questa lezione?

- analisi delle scelte che hanno guidato inizialmente lo sviluppo di Internet
  - due scelte fondamentali
    - layering
    - end-to-end principle e low-effort interconnection
- come si misurano le prestazioni in Internet
  - Concetti di banda e di latenza

- Cosa studiare sul libro di testo?
  - paragrafo 1.2
  - paragrafo 1.1.5

LAYERING: UNA METAFORA

- CEO A scrive una lettera al CEO B

  caro Elon la tua carriera ha i giorni contati.....

  ----Mark

- Seguiamo il percorso della lettera dall'ufficio di Mark e quello di Elon

LAYERS AND ENCAPSULATION: UNA METAFORA

- CEO A scrive una lettera al CEO B
  - ripiega la lettera e la consegna al suo assistente, a cui indica verbalmente il destinatario
- L'assistente
  - mette una lettera in una busta e vi scrive sopra il nome del CEO B (che tutti conoscono)
  - la consegna a SDA Express Courier
- L'ufficio di SDA Express Courier
  - mette la lettera in una busta più grande
  - scrive indirizzo del destinataro sulla busta
  - consegna la lettera ad uno dei suo corrieri
  - SDA assicura la consegna al CEO dell'altra compagnia.

LAYERS AND ENCAPSULATION: UNA METAFORA

CEO A

ASSISTENTE

SDA EXPRESS

CEO B

ASSISTENTE

SDA EXPRESS

LAYERS AND ENCAPSULATION: UNA METAFORA

CEO A

ASSISTENTE

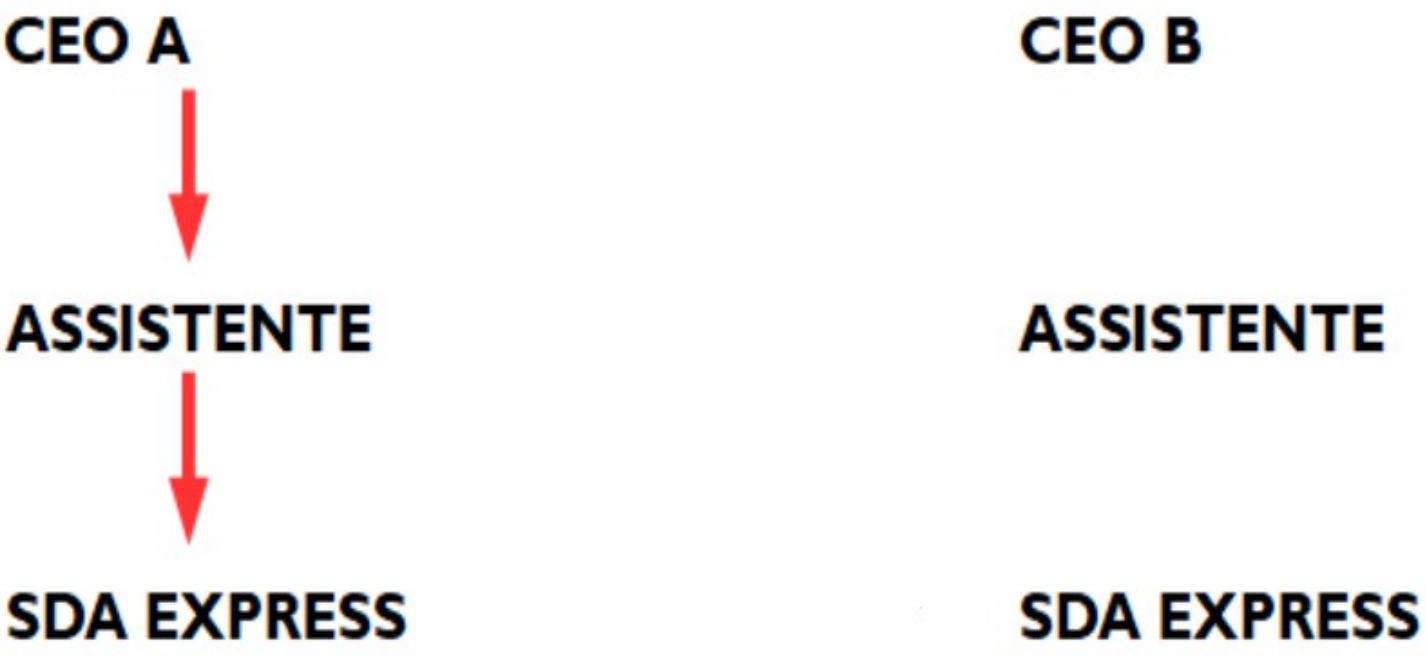
SDA EXPRESS

CEO B

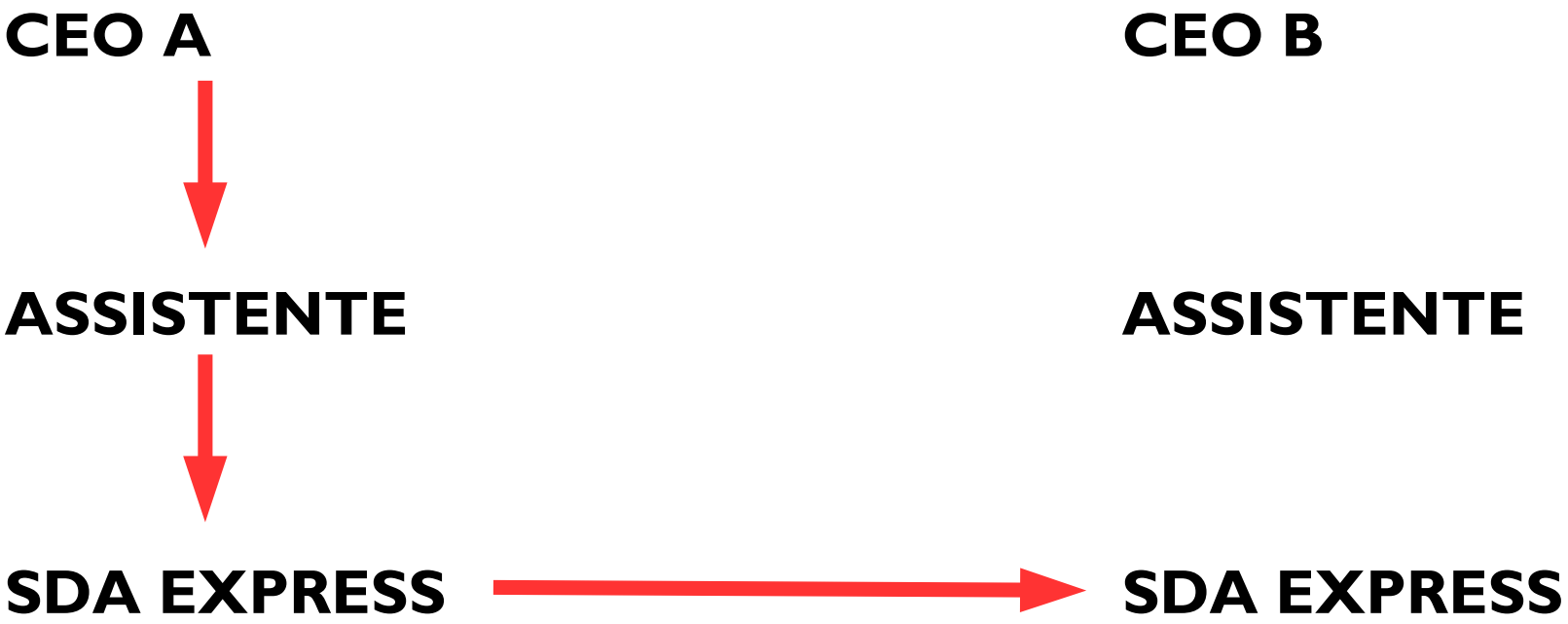
ASSISTENTE

SDA EXPRESS

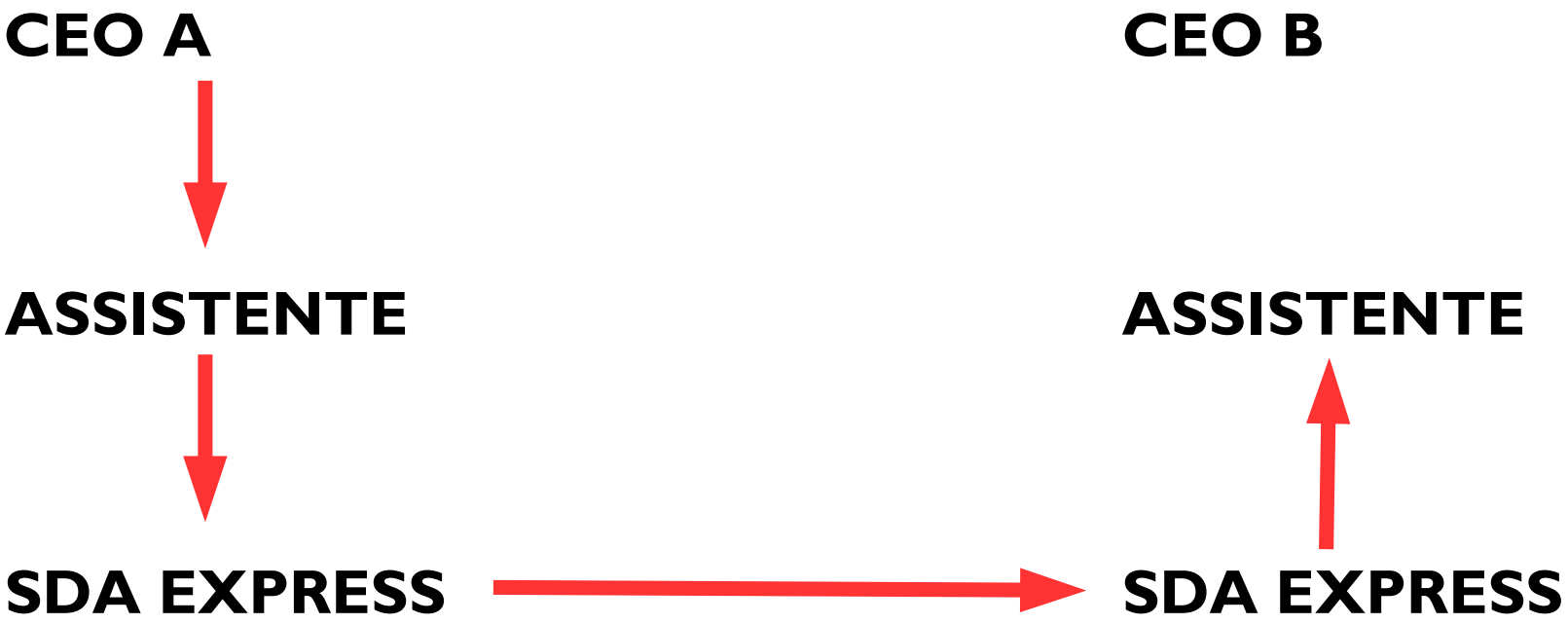
LAYERS AND ENCAPSULATION: UNA METAFORA



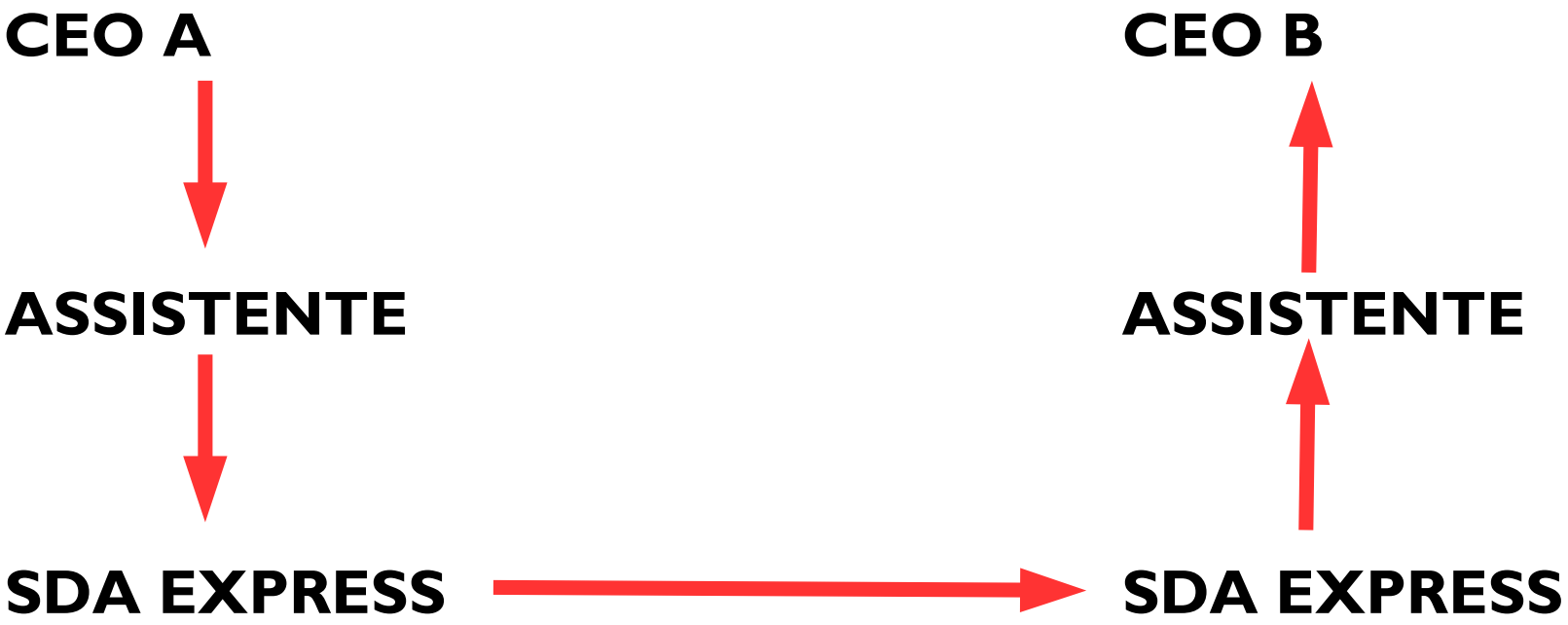
LAYERS AND ENCAPSULATION: UNA METAFORA



LAYERS AND ENCAPSULATION: UNA METAFORA



LAYERS AND ENCAPSULATION: UNA METAFORA



LAYERS AND ENCAPSULATION: UNA METAFORA

CEO A	Lettera	CEO B
ASSISTENTE	Busta	ASSISTENTE
SDA EXPRESS	Busta con indirizzo	SDA EXPRESS

LAYERS AND ENCAPSULATION: UNA METAFORA

CEO A	Contenuto semantico	CEO B
ASSISTENTE	Busta	ASSISTENTE
SDA EXPRESS	Busta con indirizzo	SDA EXPRESS

LAYERS AND ENCAPSULATION: UNA METAFORA

CEO A	Contenuto semantico	CEO B
ASSISTENTE	Identità	ASSISTENTE
SDA EXPRESS	Busta con indirizzo	SDA EXPRESS

LAYERS AND ENCAPSULATION: UNA METAFORA

CEO A	**Contenuto semantico**	**CEO B**
ASSISTENTE	**Identità**	**ASSISTENTE**
SDA EXPRESS	**Locazione**	**SDA EXPRESS**

- le entità "alla pari" sui due lati sono in grado di interpretare lo stesso tipo di infomazione
- nessun altro deve interpretare quell'informazione
- più livelli di involucro nei livelli più bassi

LAYERS AND ENCAPSULATION: UNA METAFORA

CAMIONCINO SDA

CAMIONCINO SDA

UFFICIO SMISTAMENTO

UFFICIO SMISTAMENTO

UFFICIO SMISTAMENTO

AEREOPORTO

AEREOPORTO

AEREOPORTO

LAYERS AND ENCAPSULATION: UNA METAFORA

LAYERS AND ENCAPSULATION: UNA METAFORA

LAYERS AND ENCAPSULATION: UNA METAFORA

LAYERS AND ENCAPSULATION: UNA METAFORA

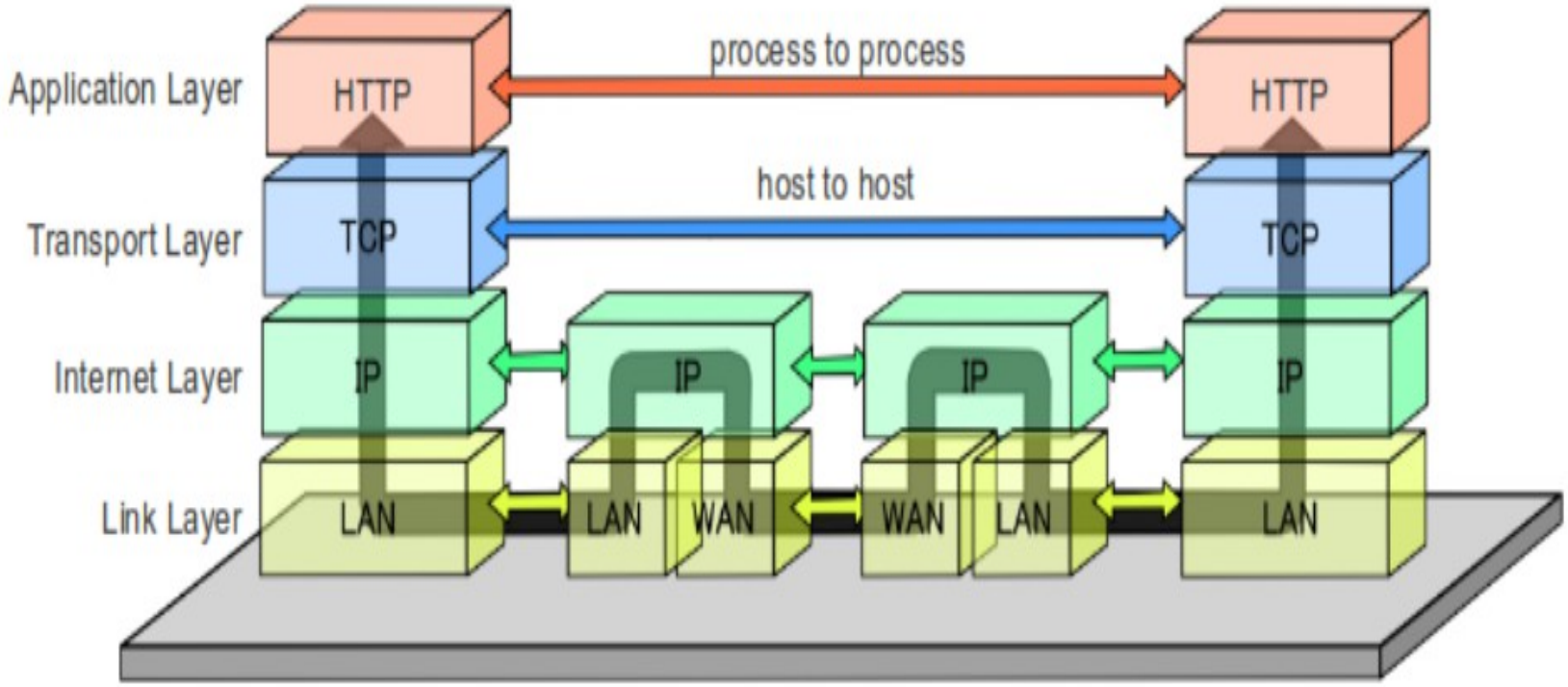
LAYERS AND ENCAPSULATION: UNA METAFORA

LAYERS AND ENCAPSULATION: UNA METAFORA

LAYERS AND ENCAPSULATION: UNA METAFORA

- maggiori livelli di involucro nei livelli più bassi (busta+SDA+PACCO)
- altezza dello "stack" più alta nei punti terminali della comunicazione
- uno "stack" più basso nei livelli intermedi
- routing: avviene negli uffici di smistamento, al livello più alto dello "stack parziale"

INTERNET E LAYERING



Per il momento notate solamente la similitudine tra la Struttura a livelli di Internet e quella dell'esempio precedente

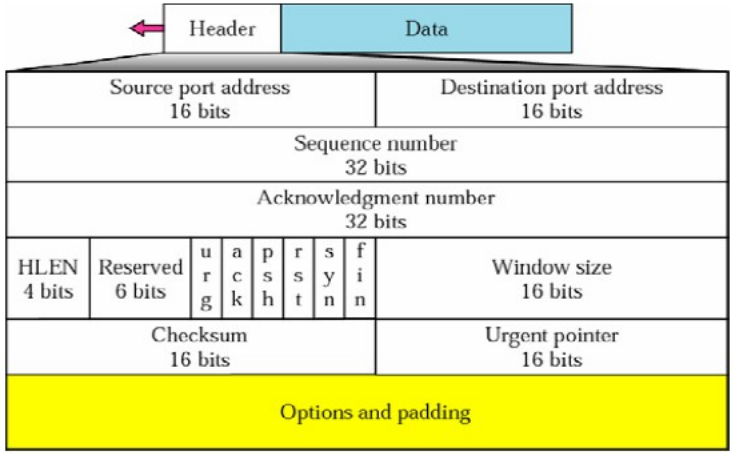
ALCUNE NOZIONI DI BASE

- Entità: ogni modulo capace di inviare e ricevere informazione.
- Protocollo: insieme di regole che controllano la comunicazione tre due o più entità
- Elementi essenziali di un protocollo: sintassi, semantica, temporizzazione
  - Sintassi: formato delle unità di dati scambiate nel protocollo, ad esempio numero campi di un pacchetto e loro sequenza nel pacchetto
    - esempio: 4 bytes nell'header del pacchetto + 12 bytes di dati
  - Semantica: significato associato alle informazioni di controllo
  - Temporizzazione: quando i dati devono essere inviati ed eventualmente, con che frequenza
- Specifica del protocollo: regole condivise tra due o più entità relative ai formati dei messaggi scambiati ed alle azioni da intraprendere quando i messaggi vengono ricevuti

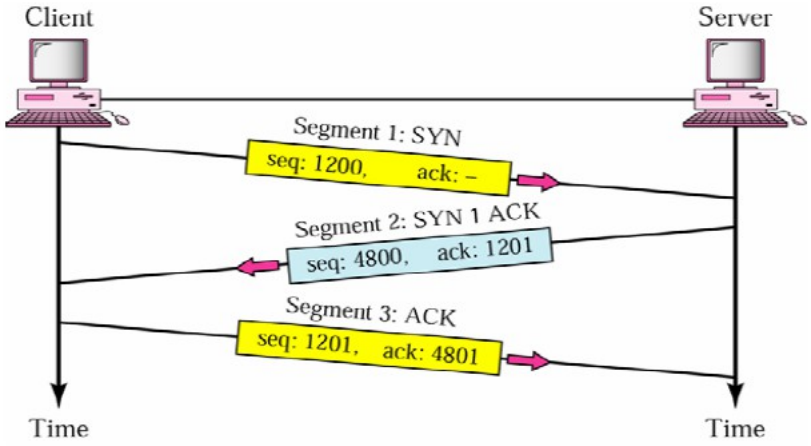
INTERNET PROTOCOL: UN ESEMPIO

TCP: Sintassi, semantica e temporizzazione del protocollo

Sintassi

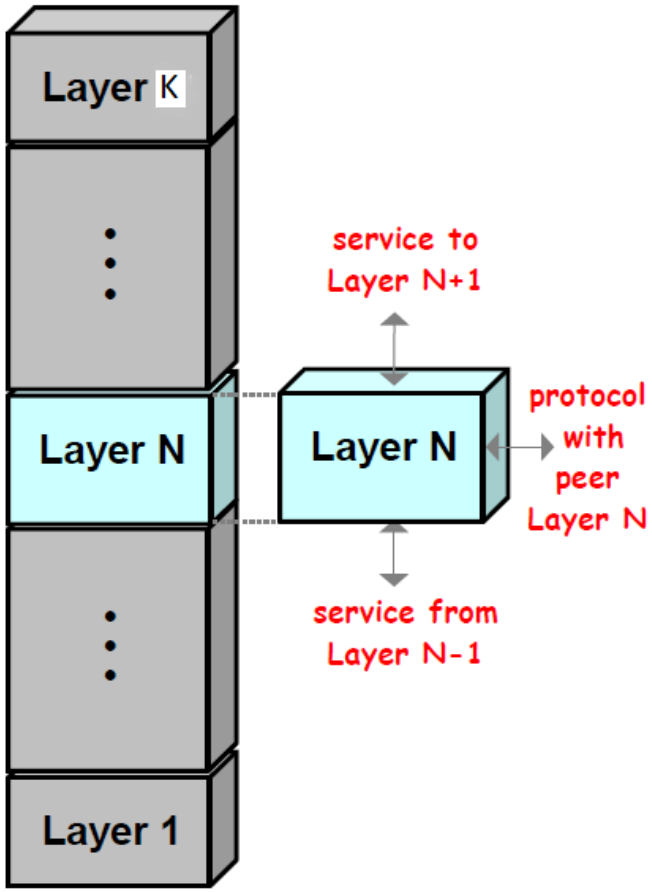


Temporizzazione al momento dell'apertura di una connessione



Semantica: ad esempio il campo HLEN contiene la lunghezza dell'header TCP in words di 32 bits (numero segmenti di 4 bytes nell'header)

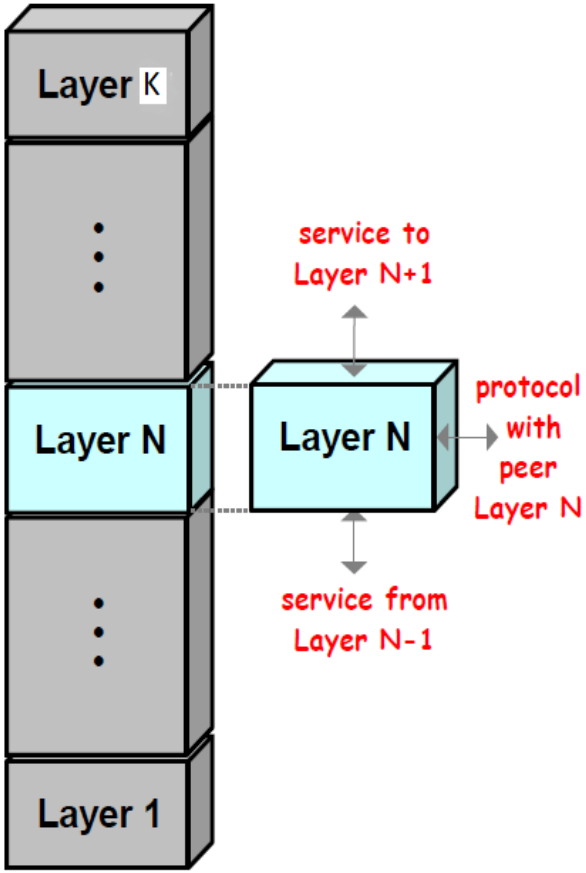
INTERNET PROTOCOL LAYERING



Primo principio fondante di Internet: strutturazione a livelli

- le funzioni relative relative alla comunicazione sono raggruppate in un insieme gerarchico di livelli
- ogni livello
  - svolge un sottoinsieme delle funzioni necessarie alla comunicazione
  - utilizza i servizi offerti dal livello più in basso per svolgere le funzioni di più basso livello, senza conoscere i dettagli di queste
  - fornisce un insieme di servizi al livello più alto
  - implementa un insieme di protocolli per la comunicazione con il lo stesso layer di un'altra entità

INTERNET PROTOCOLS LAYERING



- comunicazione verticale tra layer adiacenti
  - richiede la definizione di interfacce per ogni servizio e/o informazione che il livello sottostante deve fornire al livello sopra
- comunicazione orizzontale tra peer layers:
  - richiede la definizione di protocolli tra elementi software o hardware in esecuzione allo stesso livello su entità diverse

VANTAGGI DELLA ORGANIZZAZIONE A LIVELLI

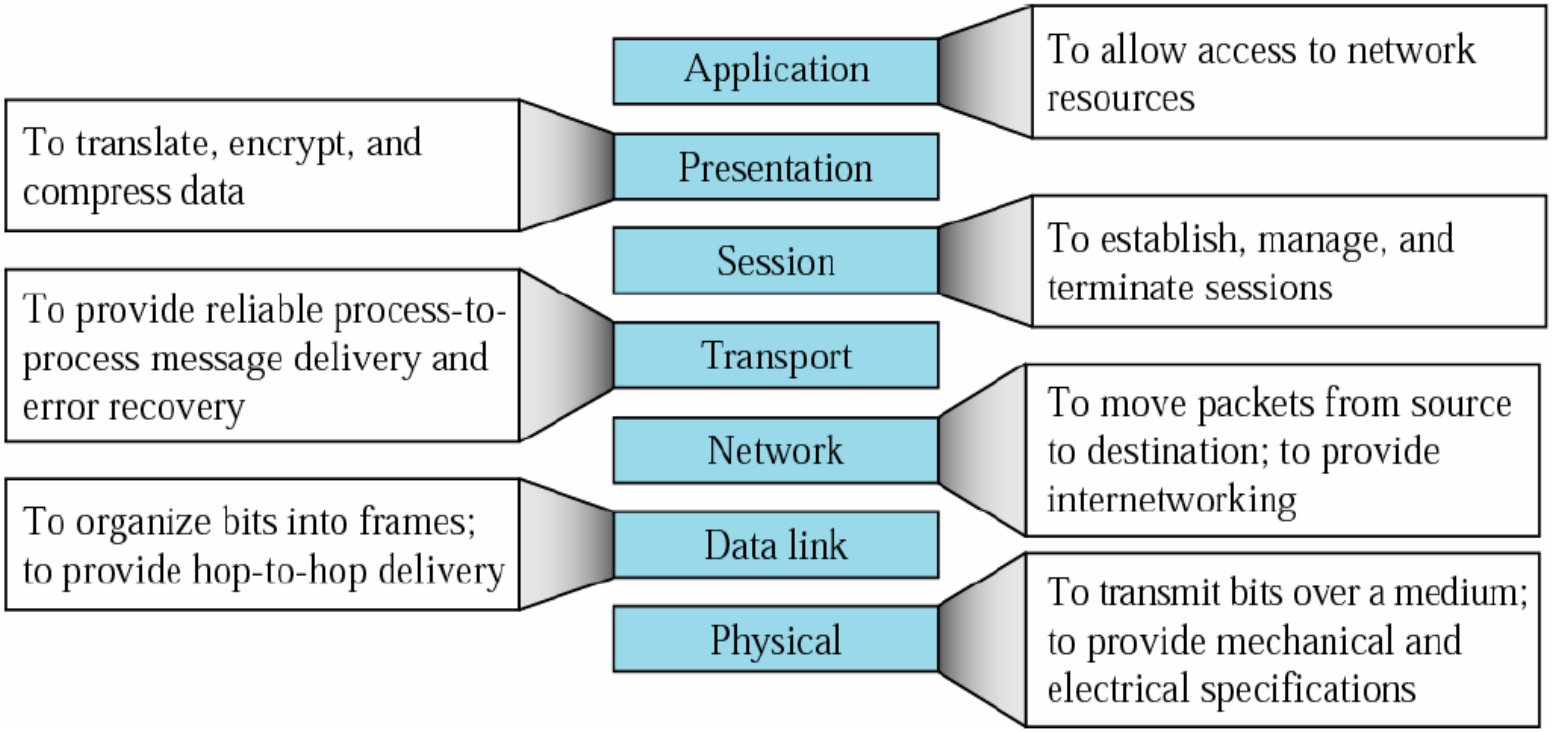
- Modularità: un problema è decomposto in un numero di problemi più piccoli e gestibili più facilmente: maggiore flessibilità nel disegno, modifica della rete
- Riuso: la stessa funzionalità può essere utilizzata da più di un protocollo a livello superiore
- Mascheramento dei dettagli implementativi a livello inferiore, separazione di specifica ed implementazione. Se la specifica di un livello non cambia, l'implementazione può essere cambiata, senza modificare il livello superiore.
- Diverse comunità possono lavorare su livelli diversi, in parallelo

Una organizzazione monolitica del software e dell'hardware è estremamente difficile e costosa da modificare e diventa rapidamente obsoleta

Un approccio a livelli favorisce una rapida modifica incrementale

L'organizzazione a livelli ha contribuito largamente al successo di Internet

OPENED SYSTEM INTERCONNECTION PROTOCOL (OSI)



- standard definitivo pubblicato nel 1984
- superato dal protocollo di Internet TCP/IP
- free distribution di TCP/IP con Berkeley Unix

INTERNET: IL MODELLO "HOURGLASS"

Livello delle Applicazioni

costruito sul livello....

Livello Trasporto (affidabile o meno...)

costruito sul livello....

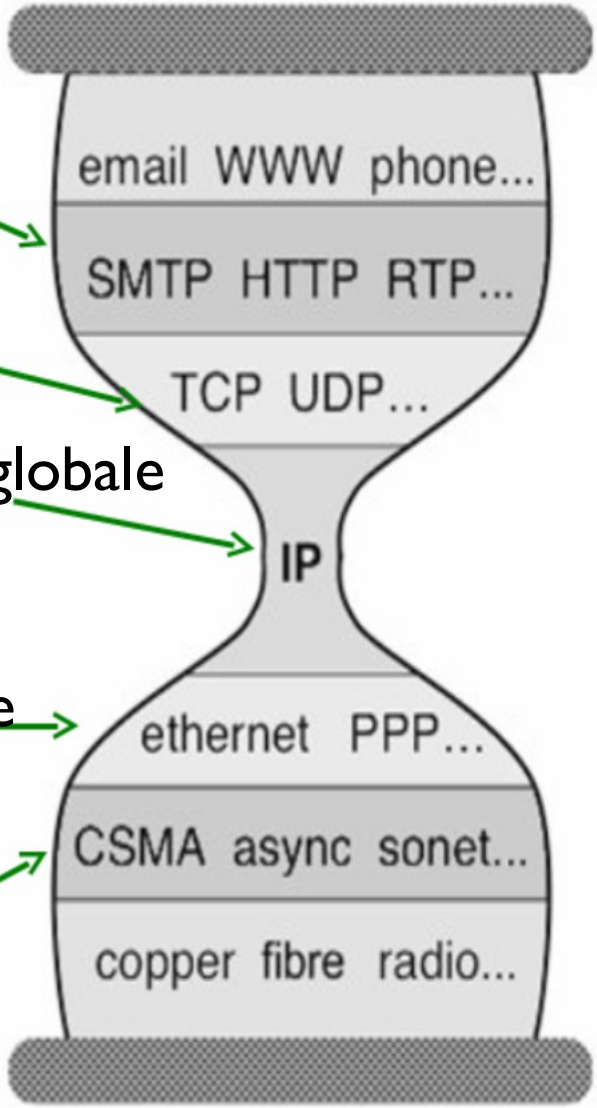
IP: Consegna best effort di pacchetti a livello globale

costruito sul livello....

Consegna best effort di pacchetti a livello locale

costruito sul livello....

Trasferimento fisico

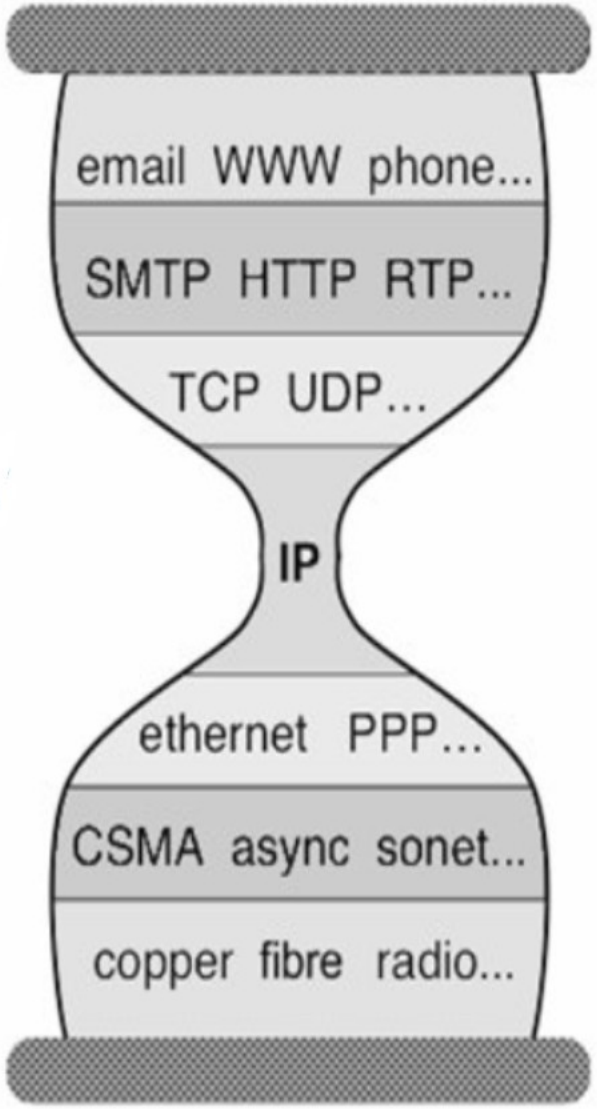


INTERNET: IL MODELLO "HOURGLASS"

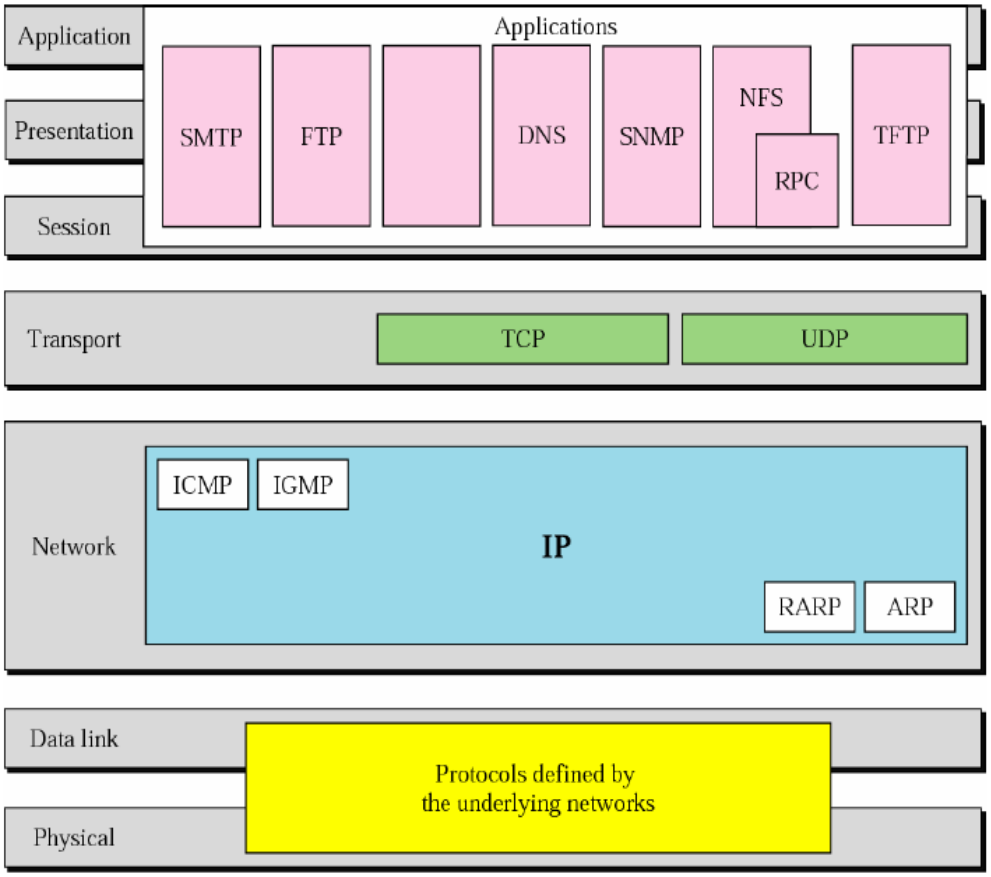
- Ogni livello
  - dipende dal livello sottostante
  - supporta il livello soprastante
  - è indipendente dagli altri livelli

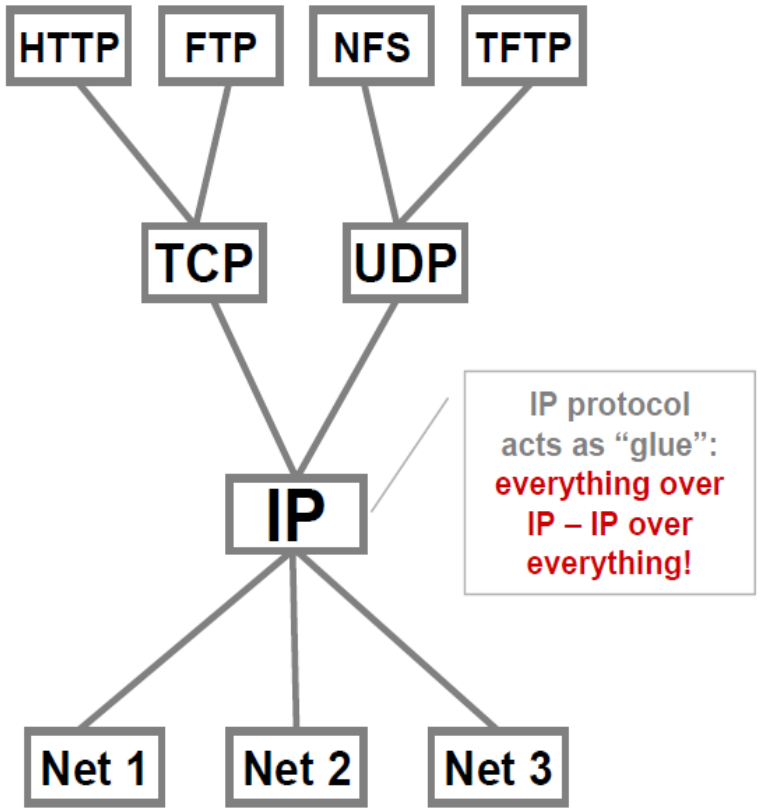
- più protocolli per ogni livello:
  - ogni componente al livello i sceglie il protocollo di livello i-1 più adatto

- L'unico livello che contiene un unico protocollo è il livello IP
  - livello unificante



INTERNET ED OSI





PHYSICAL LAYER

Funzionalità:

- rappresentazione dei bit: definisce come gli 0 e gli I vengono rappresentati da segnali elettrici/ottici (voltaggio, durata,...)
- sincronizzazione dei bit: sincronizzazione dei clock del mittente e del destinatario a livello di trasmissione di bit
- non verrà preso in esame in questo corso

DATA LINK LAYER

- **Framing:** divisione dello stream di bit ricevuto dalla rete in un insieme di data units, chiamati frames
- **Access Control:** se due o più device sono connesse allo stesso link fisico, determina quale device controlla quel link, ad un certo punto.
- **Physical Addressing:** se il frame è destinato ad un host della stessa rete. aggiunge di un header per definire sender e/o receiver del frame. Se è destinato ad una rete diversa rispetto a quella del mittente, l'indirizzo aggiunto nell'header è quello del device che connette la rete all'esterno.
- **Flow Control**: se la frequenza con cui i dati vengono ricevuti dal mittente è inferiore a quella con cui il mittente produce i dati, previene la perdita dei pacchetti
- **Error Control**: rende il link affidabile
  - meccanismi per individuare i frame dnneggiati o perso
  - meccanismi per evitare la duplicazione di frames
  - aggiunta di un trailer alla fine del frame

The data link layer trasforma il link a livello fisico, visto come uno stream di bits, in un link affidabile. Il livello fisico risulta quindi, per i livelli superiori come indenne da errori

NETWORK LAYER (INTERNET LAYER)

- Protocollo IP: deve essere compreso da ogni device presente su Internet
- Funzionalità:
  - Logical Addressing: il data link layer fornisce un indirizzamento a livello di rete locale. Se il pacchetto viene spedito al di fuori di questa, è necessario un nuovo meccanismo di indirizzamento, a livello globale
  - Routing: Fornisce meccanismi per il routing/switching dei pacchetti verso la loro destinazione
  - Forwarding: inoltro di pacchetti
- E' l'unico protocollo che deve essere interpretato da ogni nodo della rete
  - transport protocol solo su end host (per end-to-end principle)
  - application protocols, diversi per ogni applicazione
  - link layer: diversi protocolli, per reti diversi, ad esempio WiFI ed Ethernet

Host to Host Layer

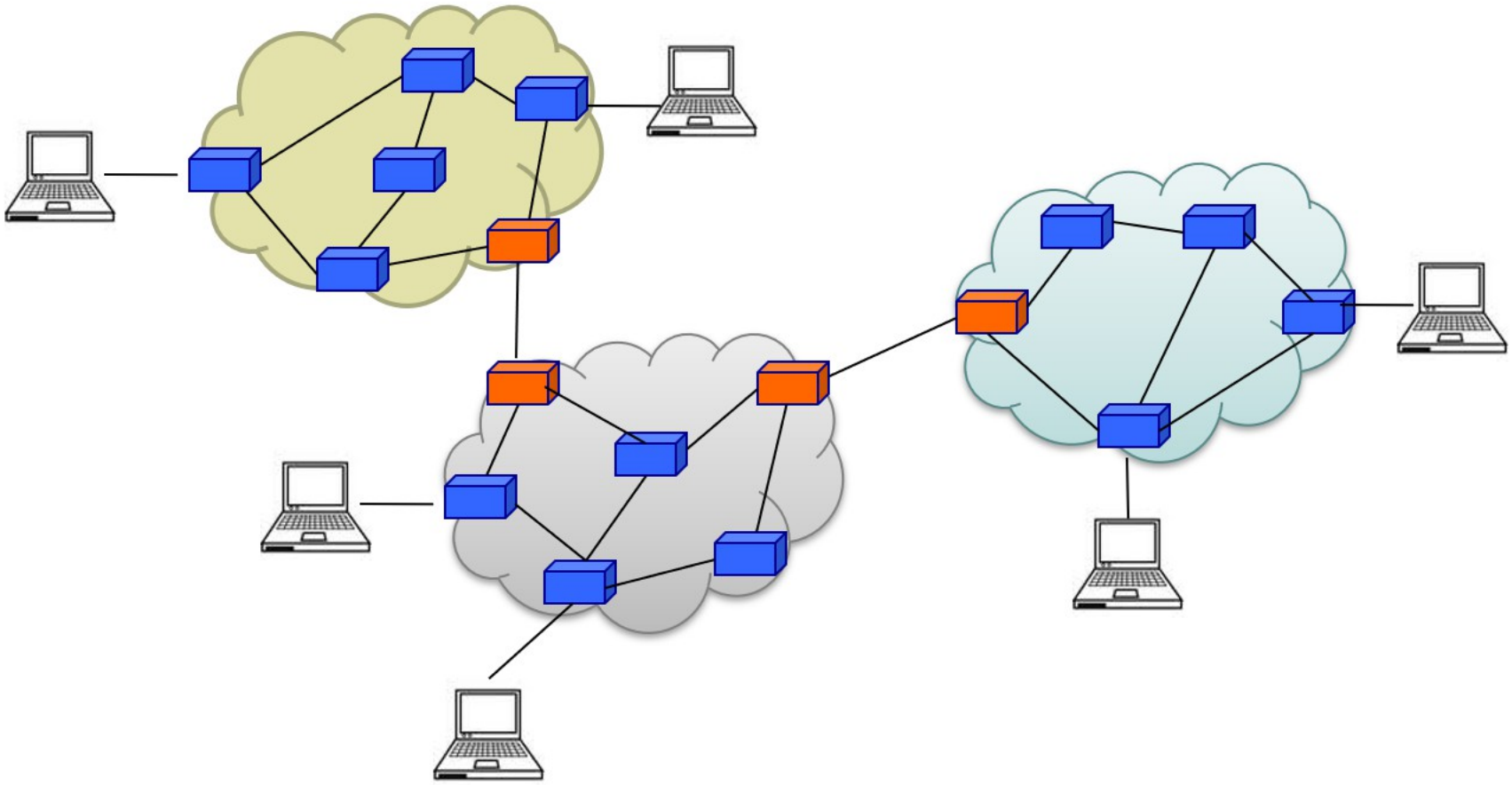
- Indirizzamento mediante porte. Più processi contemporaneamente in esecuzione sullo stesso host. Come distinguere il processo a cui è destinato un messaggi? Indirizzamento tramite il concetto di porta.
- Segmentazione e riassemblaggio. Divisione di un messaggio in segmenti, identificati da numeri di sequenza. Gli identificatori sono utilizzati per identificare e rimpiazzare i pacchetti persi nella trasmissione.
- Controllo della connessione
- Controllo del flusso e degli errori.

APPLICATION LAYER

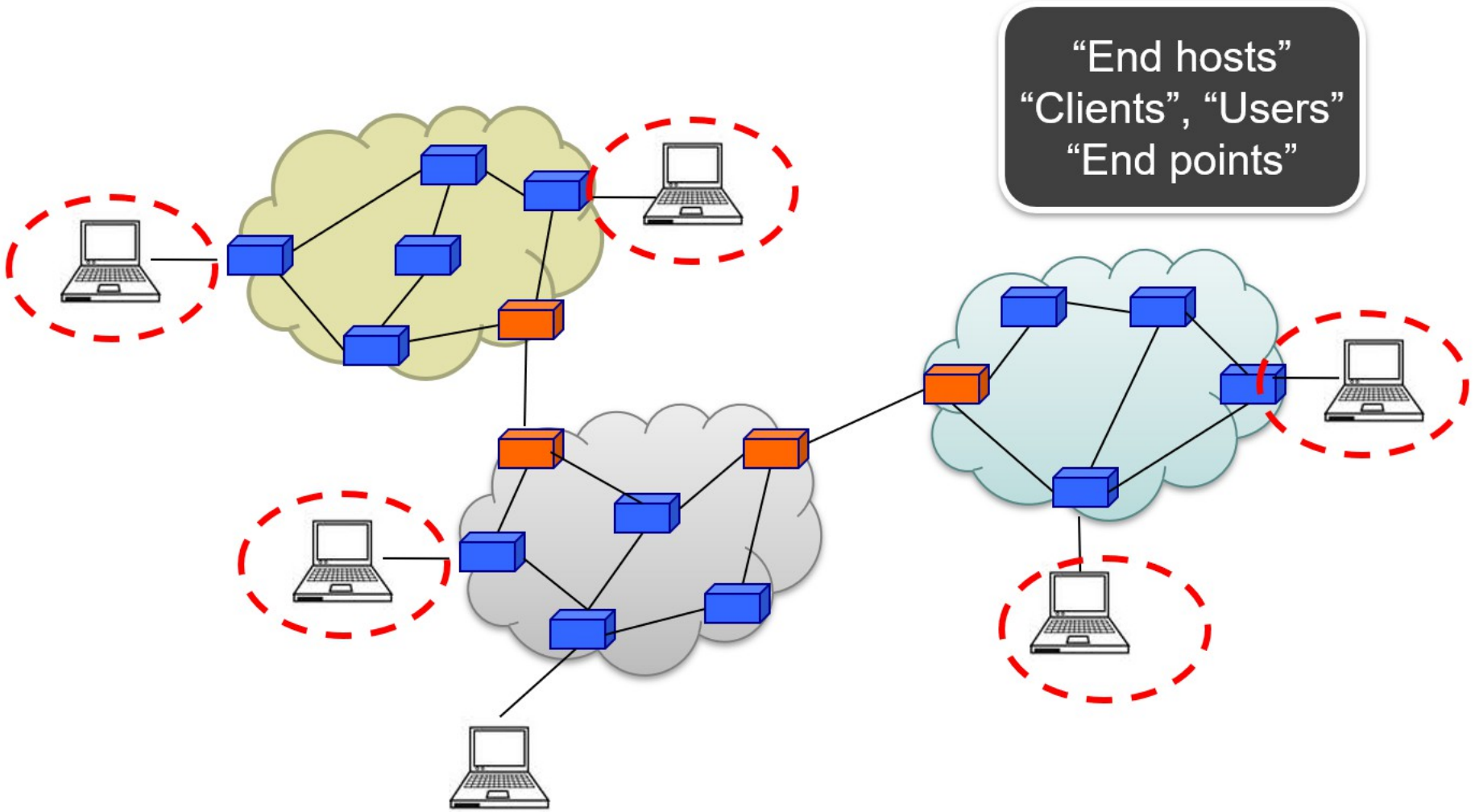
- Diversi tipi di applicazioni sviluppate a questo livello

- Molte realizzate secondo il modello client server
  - Web
  - Posta elettronica
  - FTP
  -
- Alcune realizzate secondo il modello P2P
  - file sharing
  - criptomonete
  - Streaming video/audio

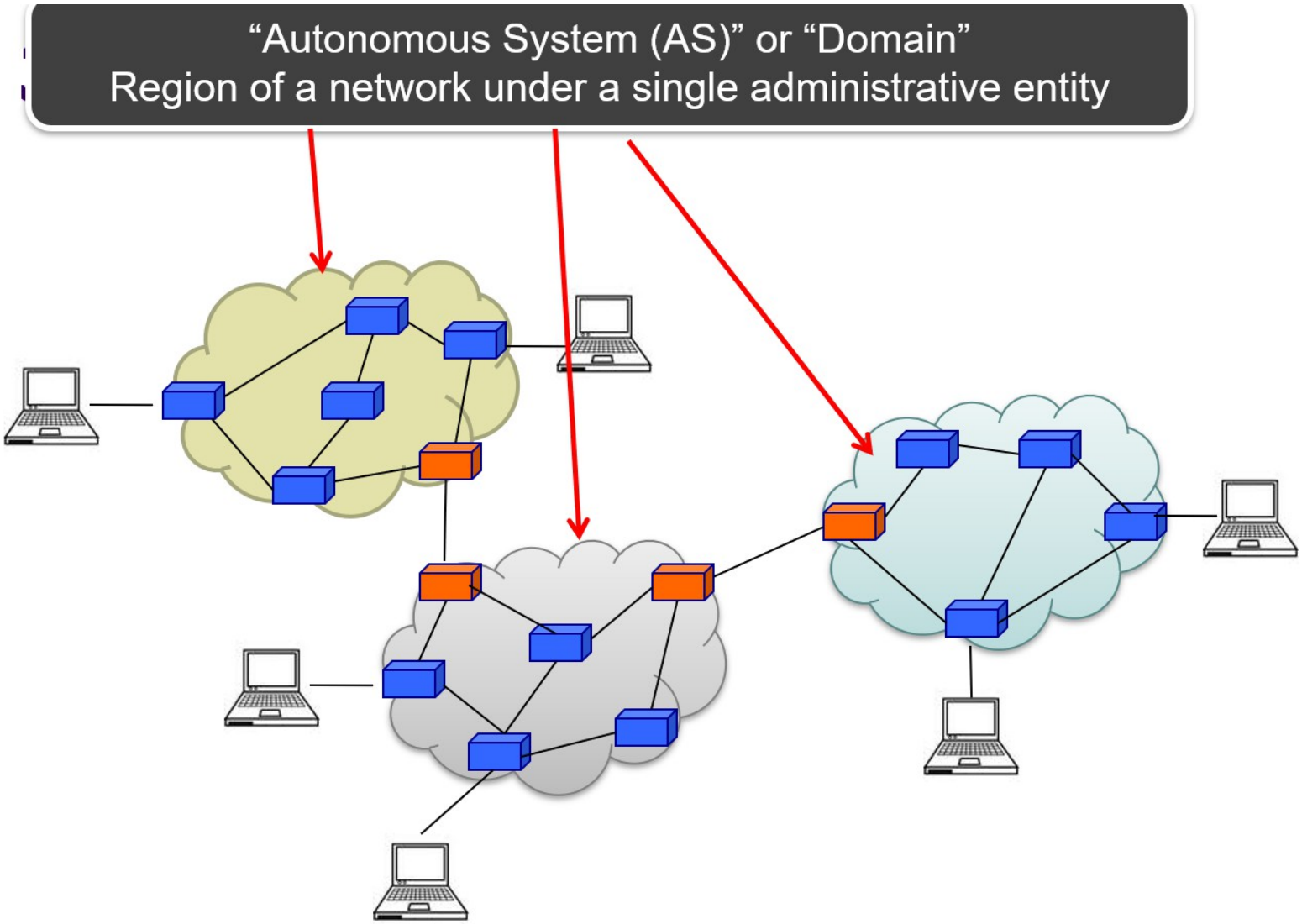
INTERNET: TOPOLOGIA SEMPLIFICATA



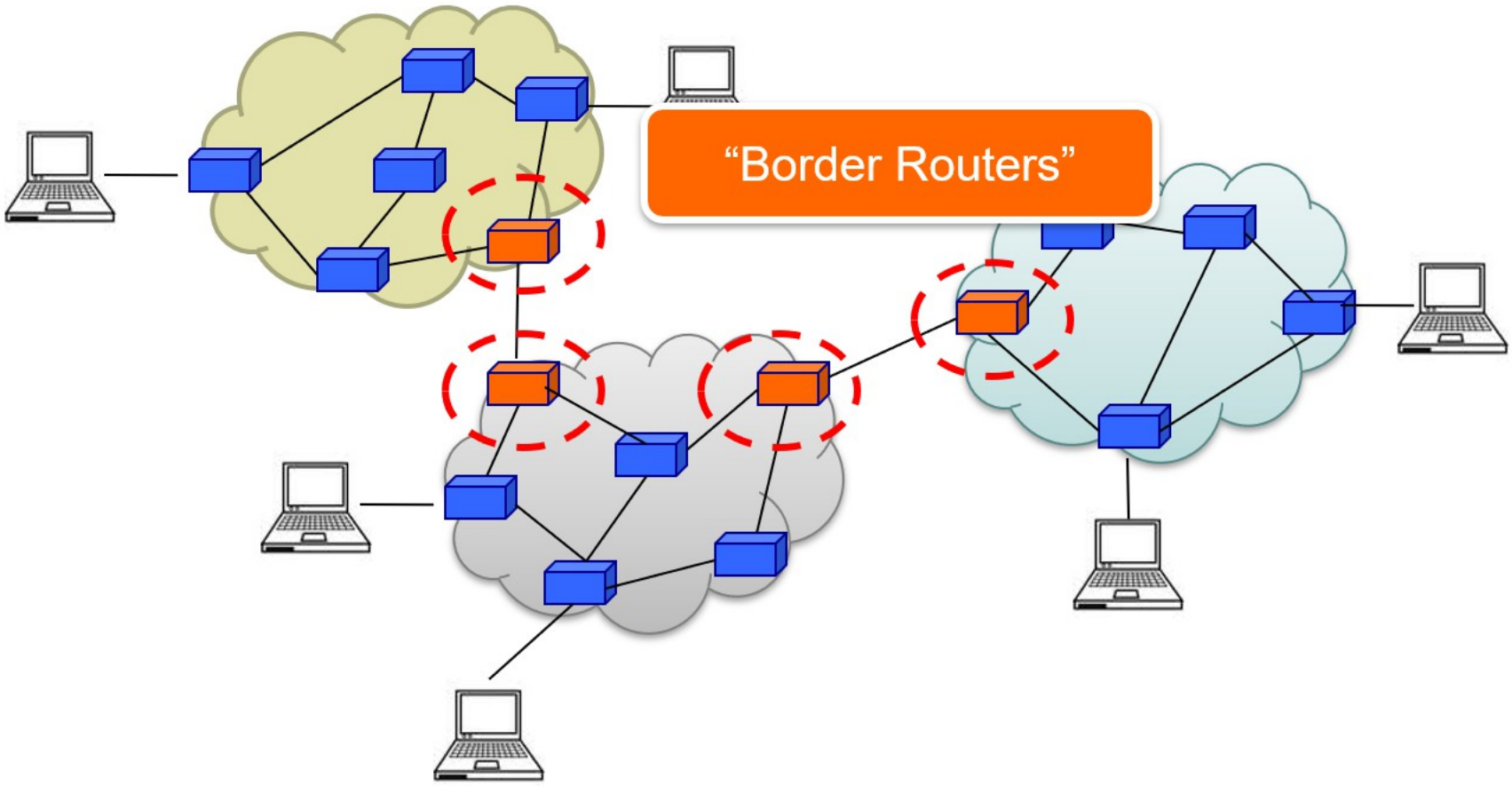
INTERNET: TOPOLOGIA SEMPLIFICATA



INTERNET: TOPOLOGIA SEMPLIFICATA



INTERNET: TOPOLOGIA SEMPLIFICATA



END-TO-END PRINCIPLE

Secondo principio fondante di Internet: end-to-end principle

- scopo principale dei progettisti di Internet: interconnettere un insieme di reti con il minor sforzo possibile
  - favorire la raggiungibilità, da parte di un insieme di reti locali, di una supernetwork globale, in cui ogni host possa raggiungere qualsiasi altro host
  - un insieme di gateways (routers) avrebbero dovuto mettere insieme le diverse reti

- End-to-end principle: "se qualche funzionalità può essere eseguita dagli end-host, non coinvolgere i router, ma eseguila sull'end-host"
  - funzionalità minime per i routers/gateways:
    - solo forwarding dei dati da una rete ad un'altra rete
    - affidare una funzionalità ai router solo se non esiste altra alternativa
  - tutte le altre funzionalità affidate agli end-hosts

END-TO-END PRINCIPLE: UN ESEMPIO

- Reliable Data Transfer:
  - assicura che i dati siano consegnati al ricevente nonostante la possibilità che vengano persi nella rete
  - richiede la ritrasmissione dei messaggi, se essi vengono persi
- Due possibilità
  - end-to-end reliability: l'end point (mittente) aspetta un acknowledge per ogni blocco di dato e li ritrasmette, se persi
  - link-to-link reliability: ogni router sul cammino tra il mittente ed il destinatario attende un acknowledge dal router successivo, e se il dato è perso ritrasmette il dato.
    - l'end point può dimenticarsi dei dati una volta che ha ricevuto un acknowledge dal primo router
- In Internet: end-to-end reliability

END-TO-END PRINCIPLE: UN ESEMPIO

- In Internet: end-to-end reliability
- motivazioni:
  - minimalità funzioni dei router: i router non devono essere necessariamente coinvolti, esiste un'altra alternativa: esiste una soluzione implementabile negli end-host
  - il controllo effettuato solo nei router non è sufficiente
    - dati persi nella trasmissione tra l'ultimo router ed il destinatario
    - dati corrotti nella memoria receiver prima di essere elaborati
    - questi problemi possono essere evitati nel caso in cui l'acknowledge sia effettivamente inviato dal ricevente, una volta che ha ricevuto e memorizzato il dato

DISTRIBUZIONE DEI LIVELLI SUI NODI

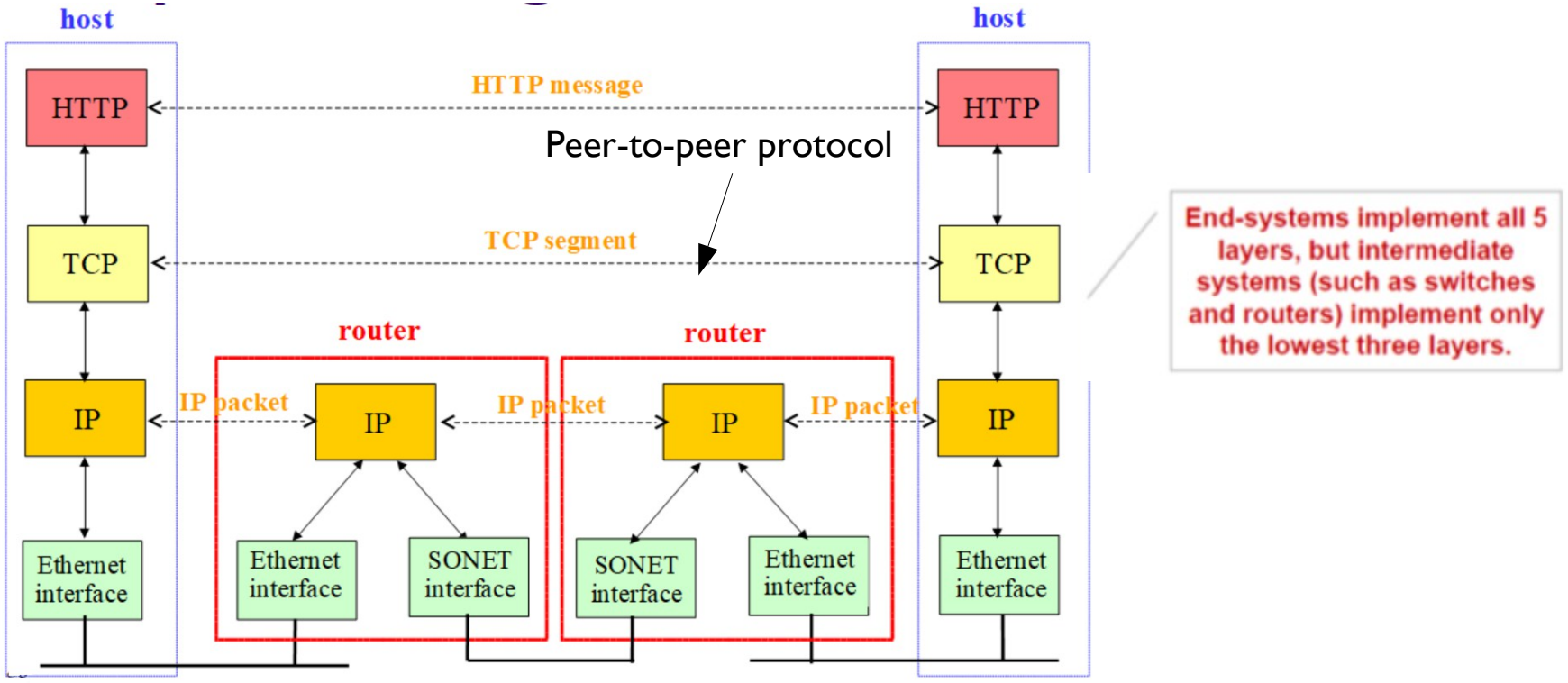
- l'organizzazione a livelli risulta semplice se implementato su un host singolo
  - uno stack di moduli in cui ogni modulo interagisce con il precedente ed il successivo
- ma in Internet lo stack deve essere implementato in modo distribuito, su macchine diverse
  - end host
  - routers
  - switch
- cosa implementare e dove implementarlo?
  - applicare l'end-to-end-principle
  - distribuzione dei livelli è una conseguenza di questo principio

DISTRIBUZIONE DEI LIVELLI SUI NODI

- End host: a partire dai bit che arrivano su una connessione, deve consegnare i messaggi alla applicazione
  - tutti i livelli dello stack presenti sugli end host
- Routers
  - gestioscono i bit in arrivo su una connessione: Physical Layer
  - partecipano alla consegna dei pacchetti:
    - sulla rete globale: network layer (IP)
    - sulla rete locale i pacchetti devono essere consegnati al prossimo host (next hop delivery): Data Link layer
  - non forniscono una consegna affidabile dei pacchetti
    - non necessari transport layer ed application layer
- Switches
  - stessa funzione dei routers, ma non partecipano alla distribuzione globale dei pacchetti
  - Physical e Data Link supportati
  - Network, transport ed application Layer non supportati

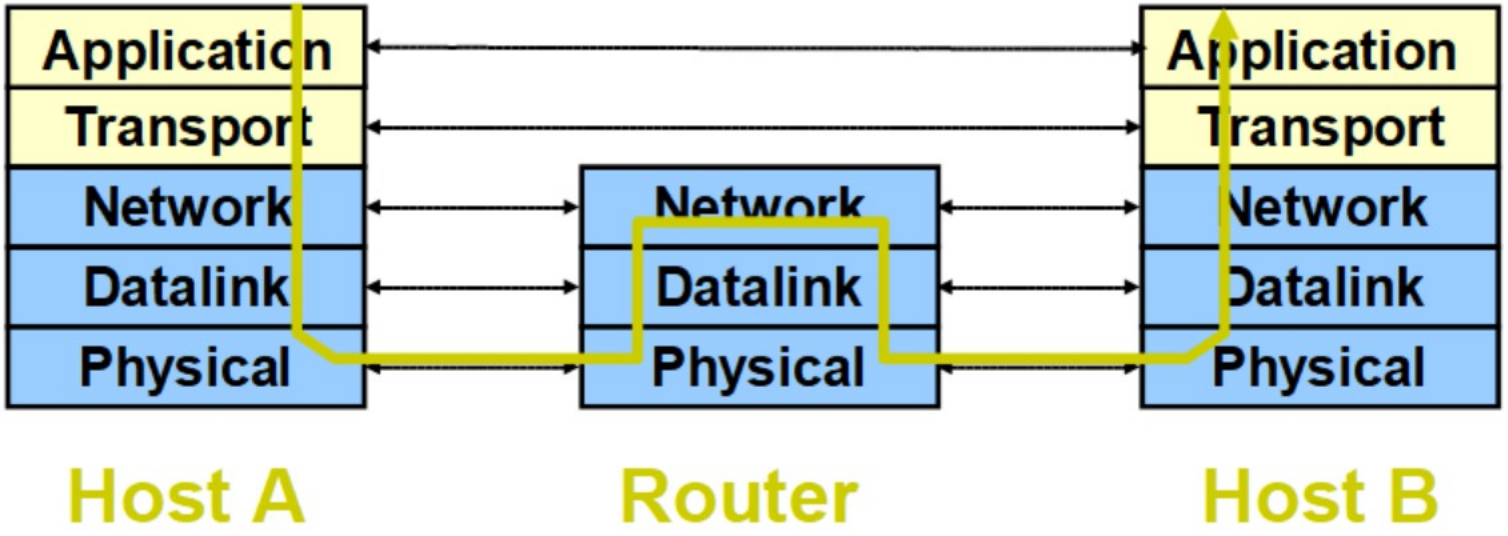
DISTRIBUZIONE DEI LIVELLI SUI NODI

- I livelli più bassi sono implementati su tutti i nodi (non mostrato, in figura, il livello fisico)
- I livelli più alti solo sugli end host
- da ogni livello l'interazione logica è con il livello corrispondente sull'altro host

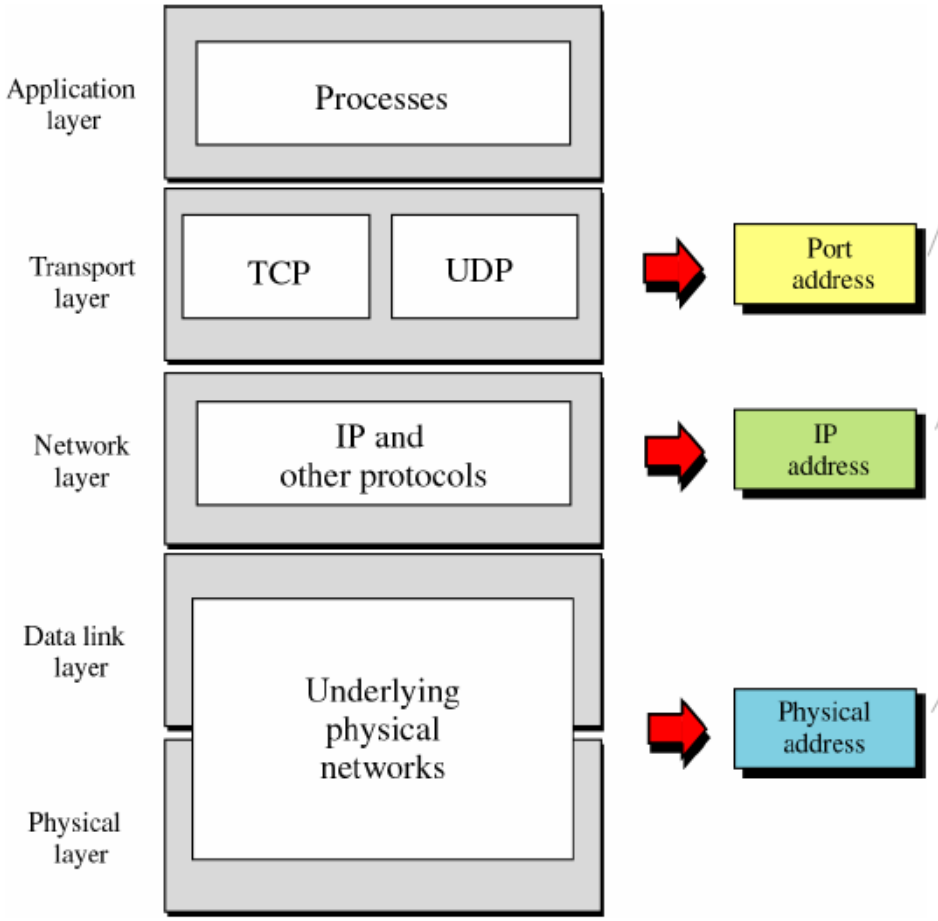


FLUSSO DEI MESSAGGI NEI LIVELLI

- I messaggi "scendono" fino al livello fisico
- vengono inviati sulla rete
- "risalgono" fino al livello opportuno



LAYERS ED INDIRIZZAMENTO

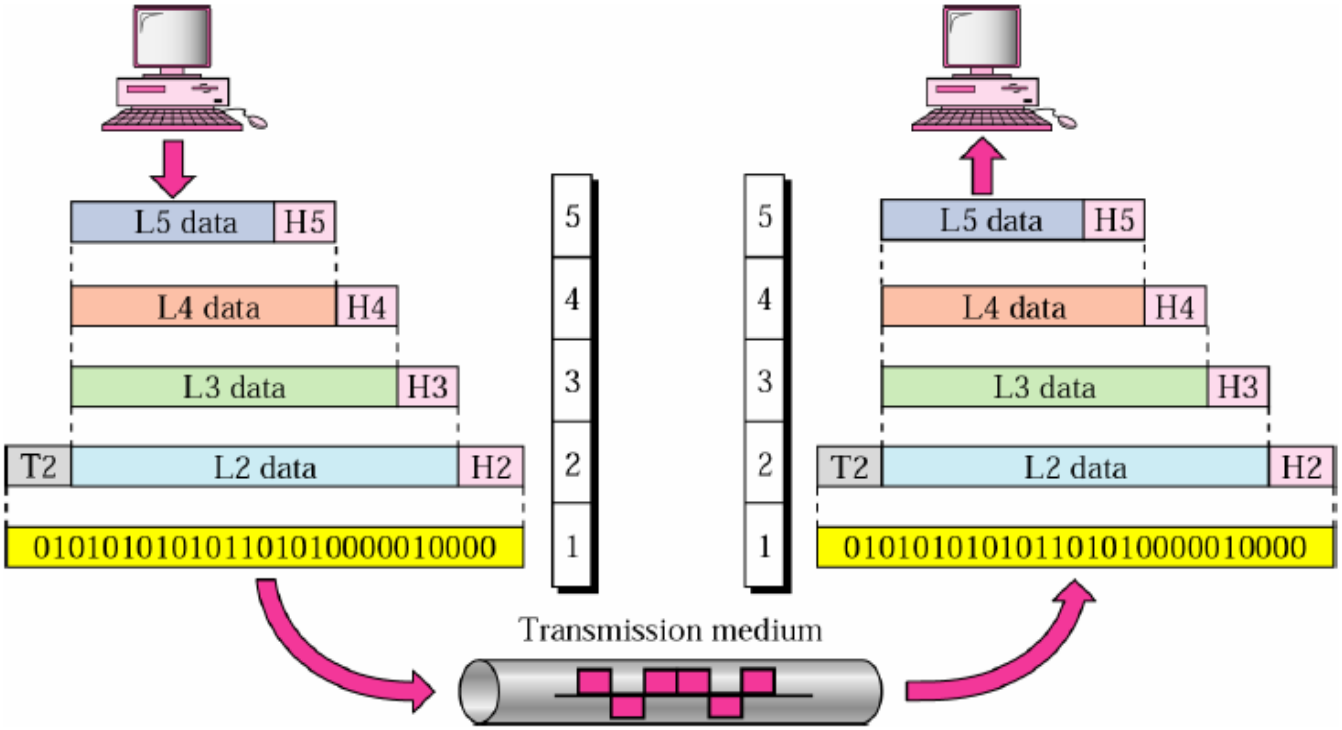


- locally unique logical address used to differentiate between applications sharing the same IP address

- globally unique logical address used to located corresponding node in the entire Internet
- hierarchical addresses that can be easily aggregated in routing tables ⇒ **fast routing !**

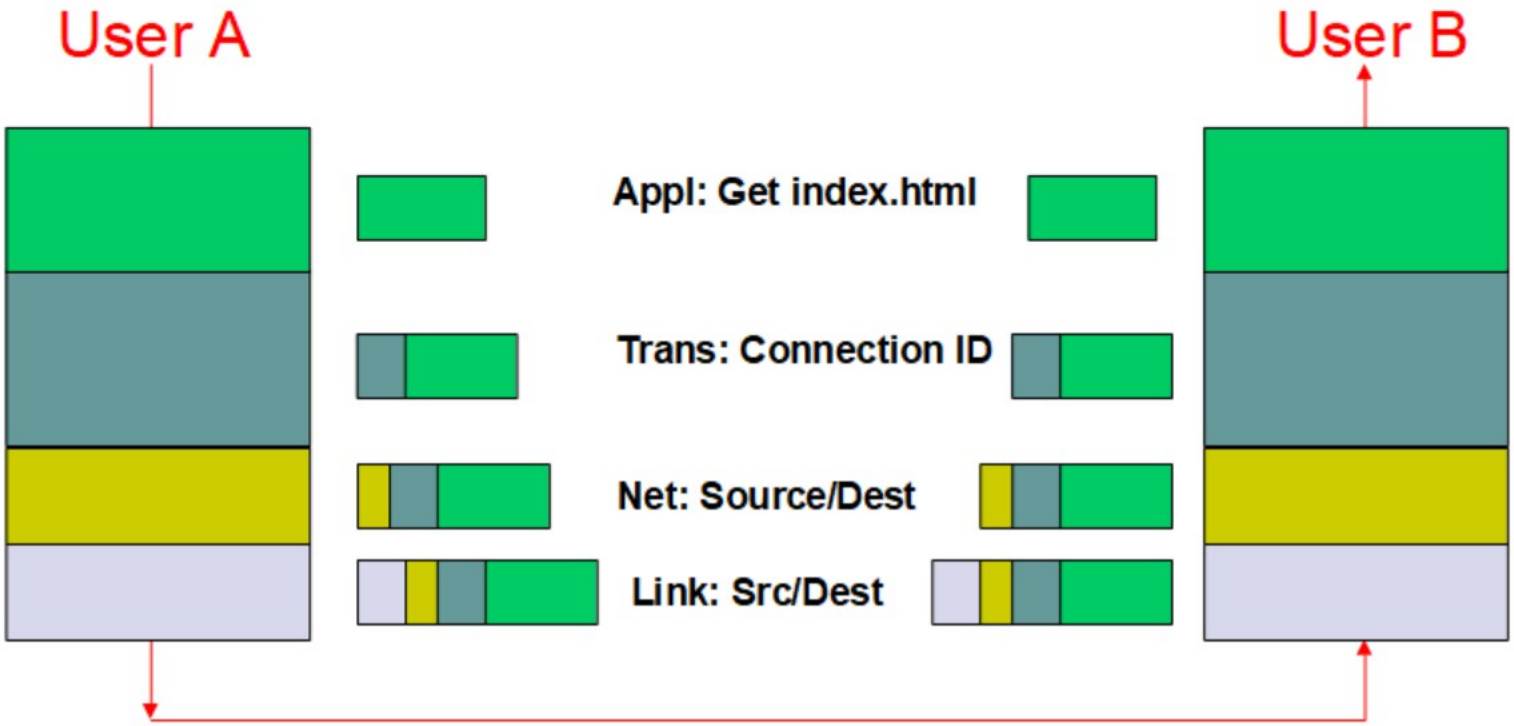
- globally unique NIC address used to located corresponding node on a LAN
- each NIC on a subnetwork may have different manufactures ⇒ we cannot aggregate physical addresses in routing tables ⇒ large networks cannot use these addresses to identify hosts !

INCAPSULAMENTO DEI MESSAGGI



NOTE: At each layer, a header is added to the data unit.
At layer 2, a trailer is added as well.

INCAPSULAMENTO: HEADERS



Common case: 20 bytes TCP header + 20 bytes IP header
+ 14 bytes Ethernet header = *54 bytes overhead*

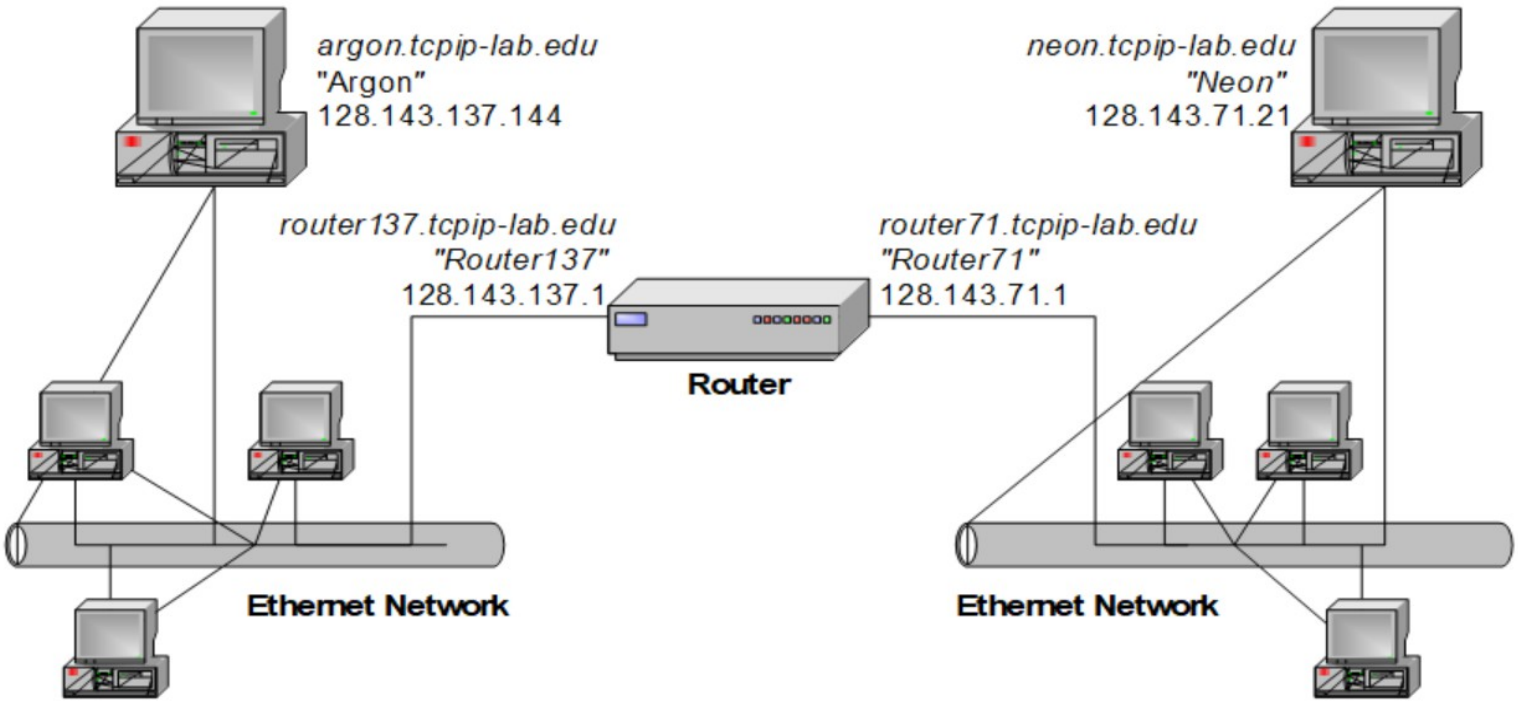
UN ESEMPIO

- come esempio riassuntivo, si mostra un esempio che illustra come tutti i livelli dello stack protocollare TCP/IP sono interessati nell'invio di un messaggio tra due nodi

- non preoccupatevi se alcuni concetti sono nuovi: saranno approfonditi nel corso delle prossime lezioni

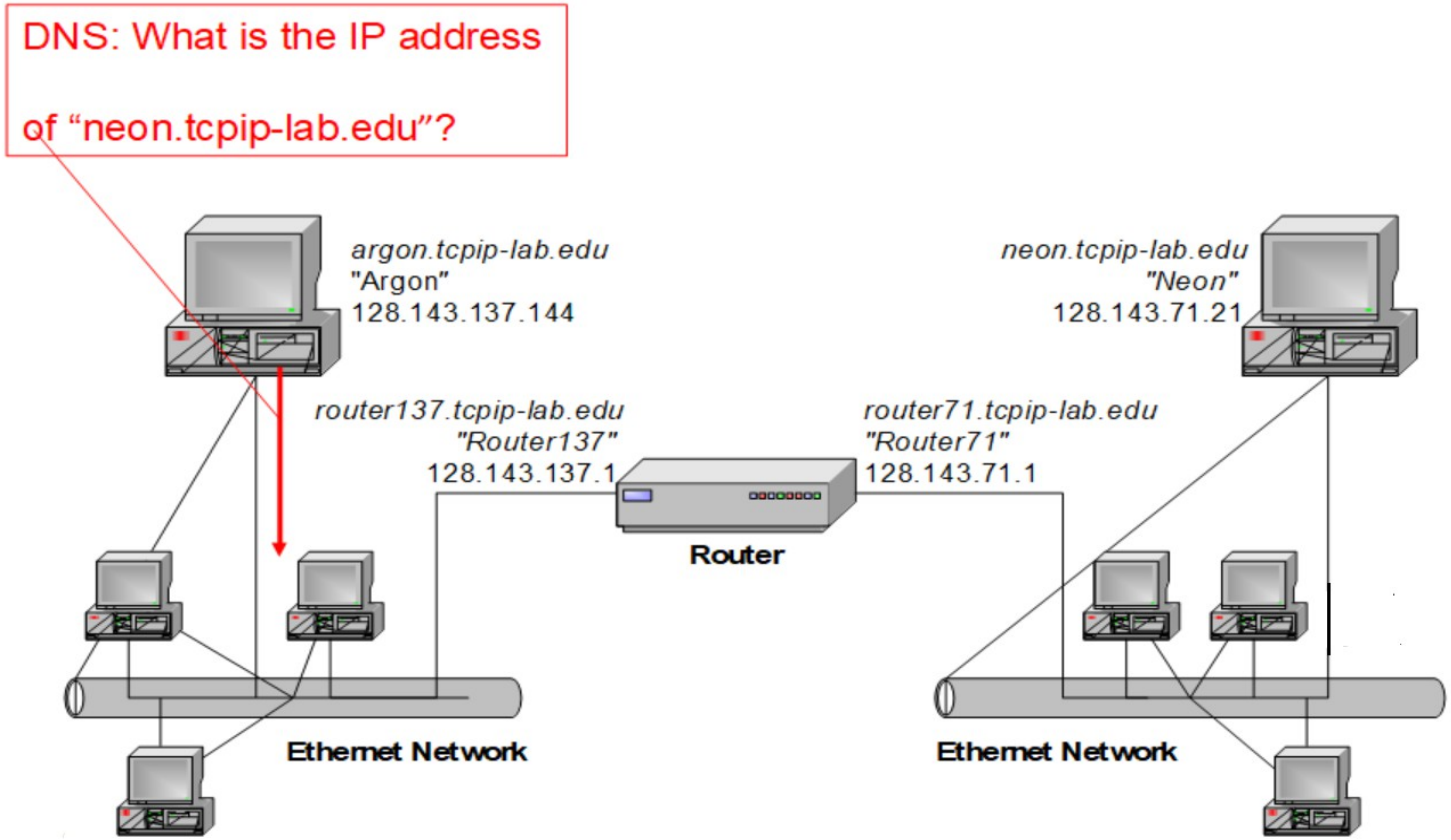
- scopo: dare una visione complessiva del funzionamento del "sistem Interne"

UN ESEMPIO



- l'host di nome argon invia un messaggio all'host di nome neon
  - ad esempio la richiesta di accedere ad una pagina web (protocollo HTTP, a livello applicazione)
- ogni host possiede due indirizzi: uno simbolico, l'altro è l'indirizzo IP

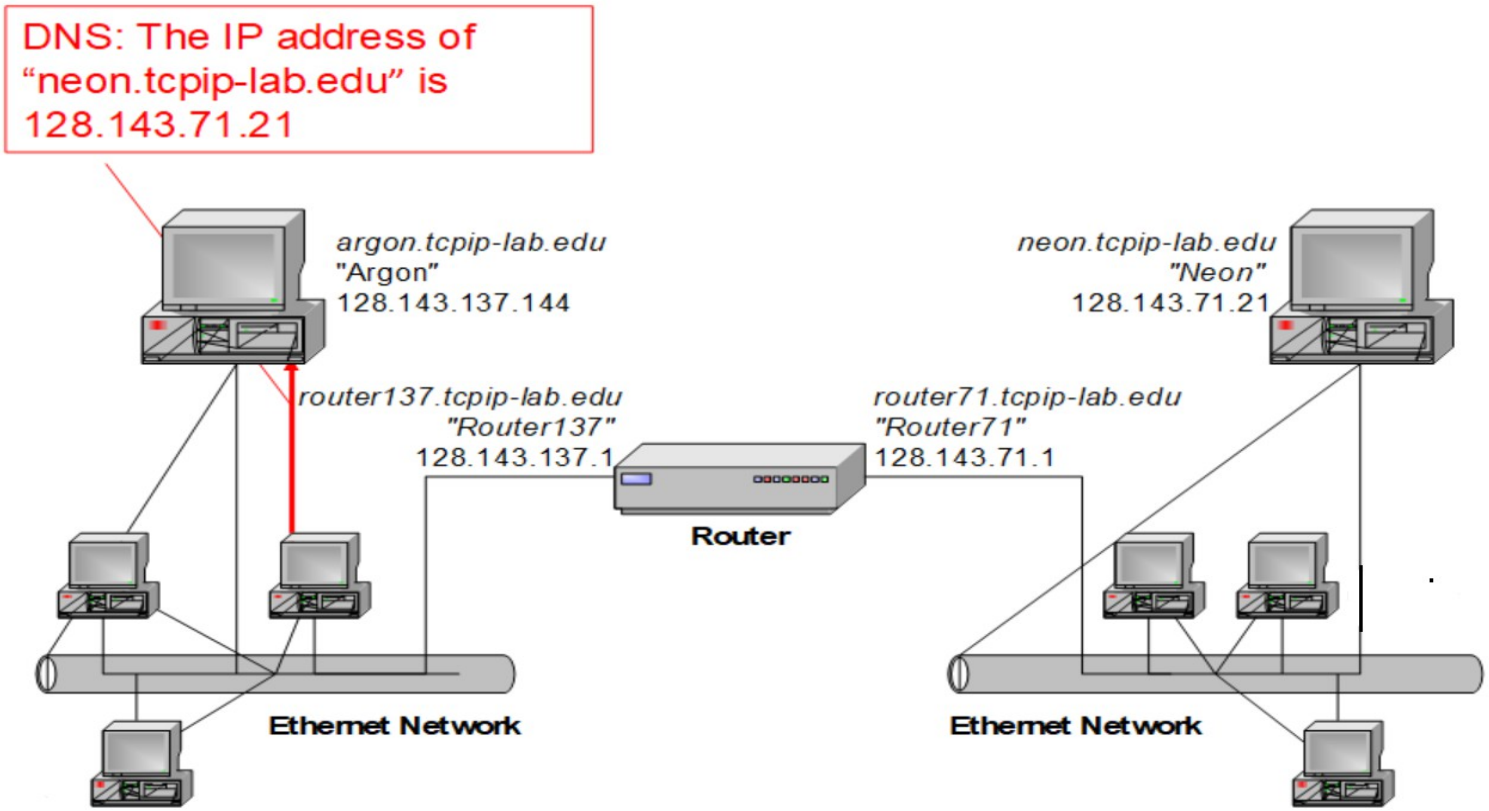
UN ESEMPIO



Primo passo

- l'applicazione presenta un nome simbolico, che deve essere tradotto in un indirizzo IP, comprensibile al protocollo IP

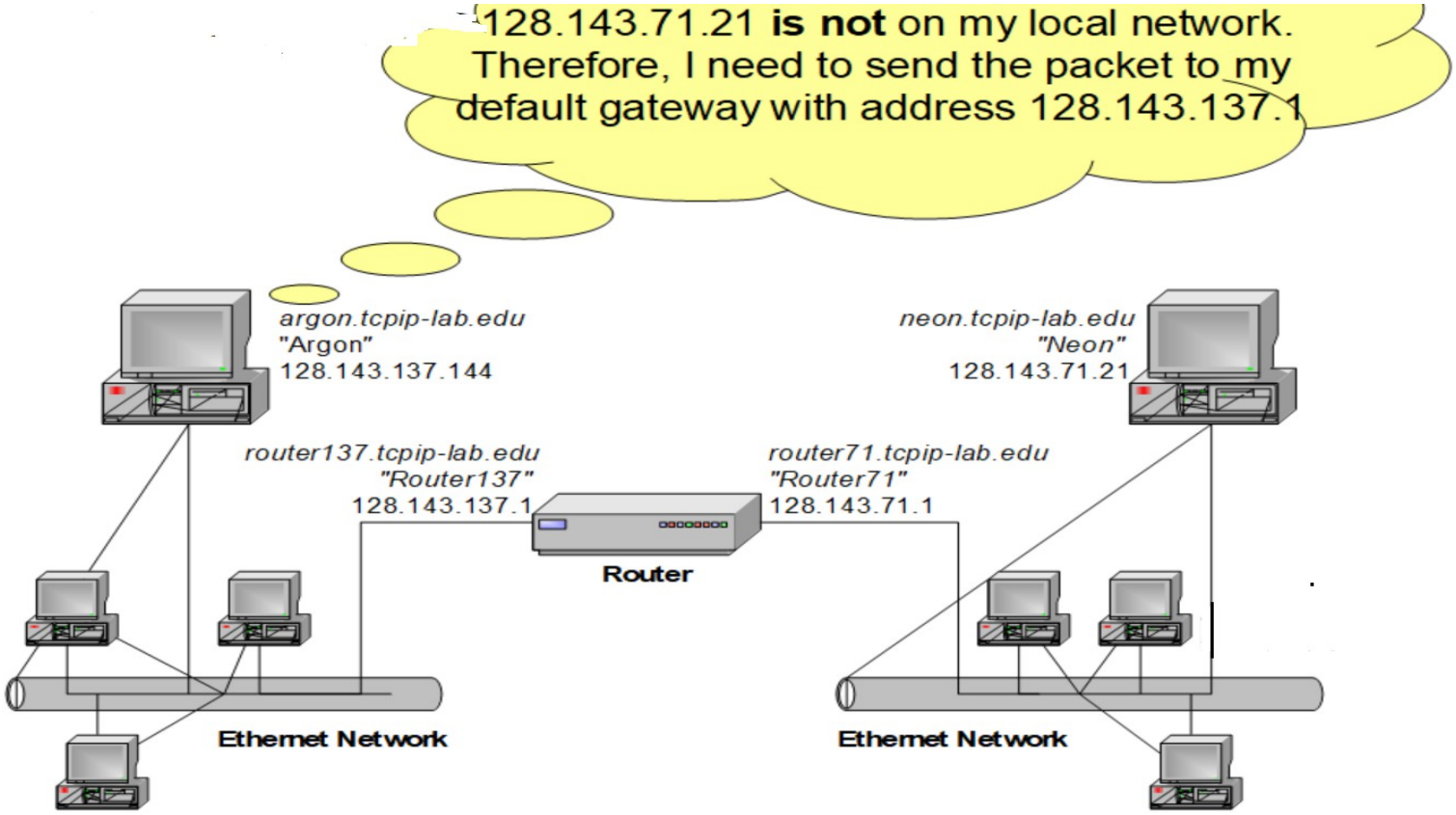
UN ESEMPIO



Secondo passo

- DNS: Domain Name System: servizio del livello applicazione per la traduzione dei nomi simbolici in indirizzi IP
- il DNS risponde con l'indirizzo IP dell'host neon

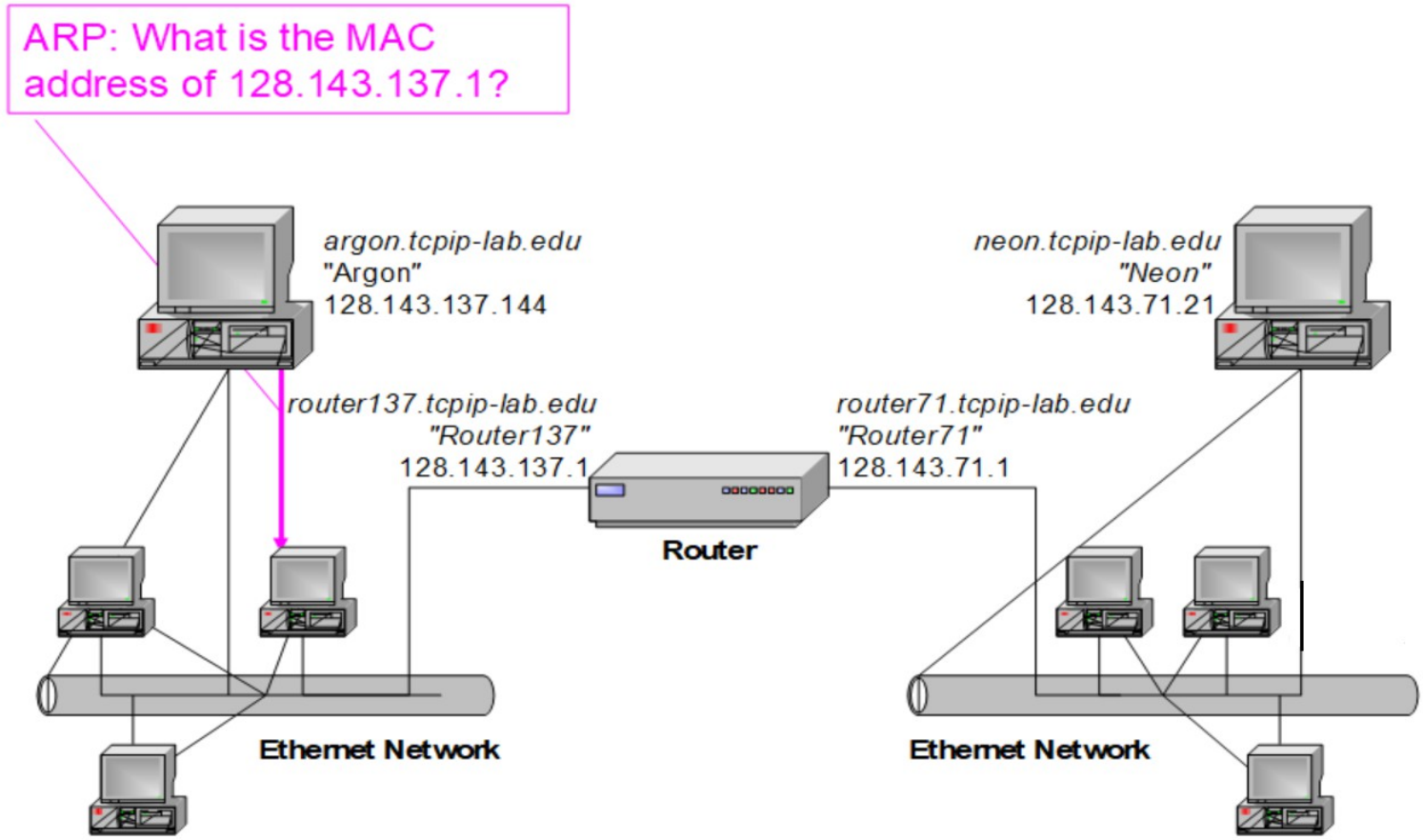
UN ESEMPIO



Secondo passo

- L'indirizzo IP non è nella stessa rete locale

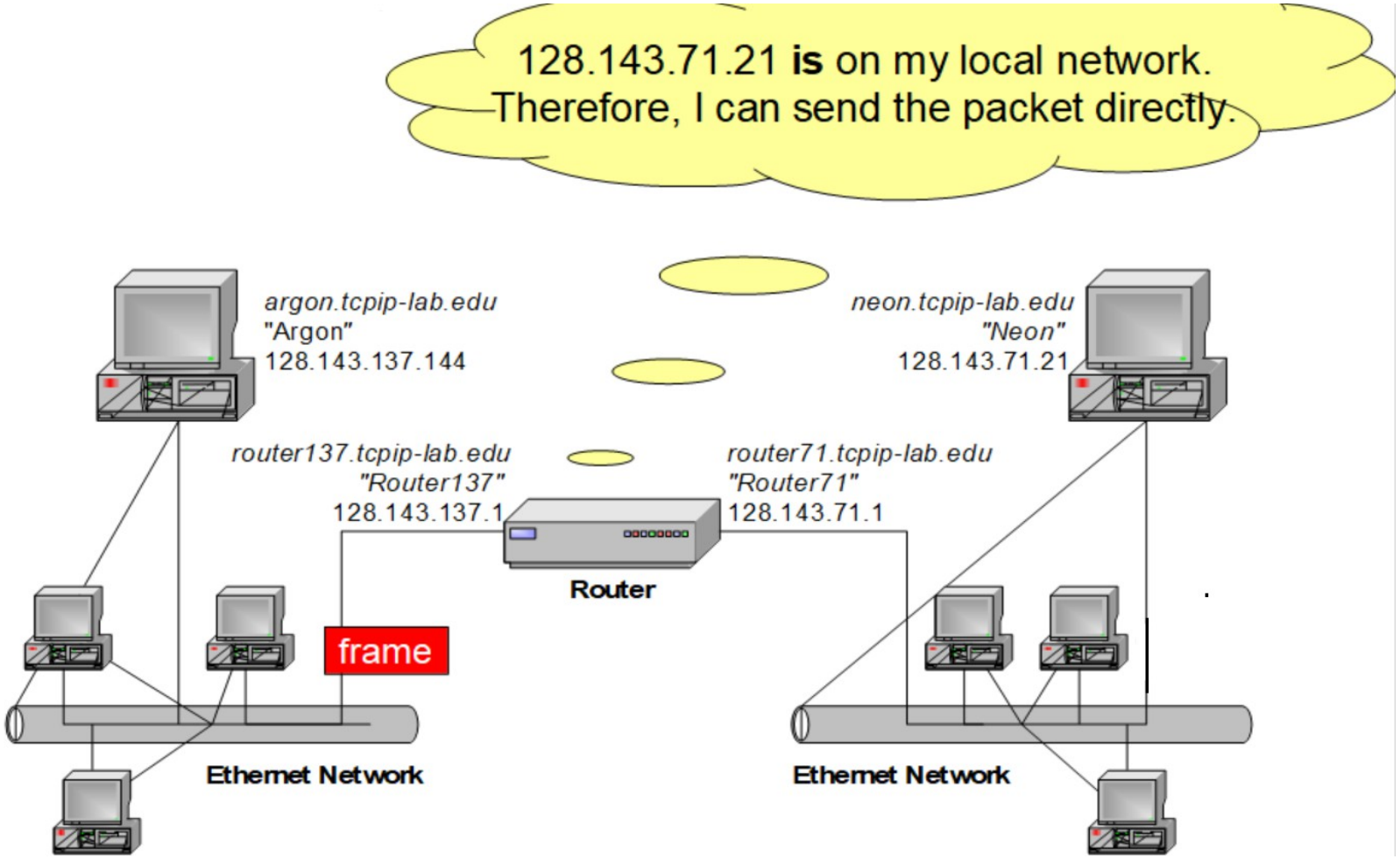
UN ESEMPIO



Terzo passo

- MAC Address: indirizzo a livello di rete locale
- Address Resolution Protocol (ARP), traduce indirizzi IP in indirizzi MAC

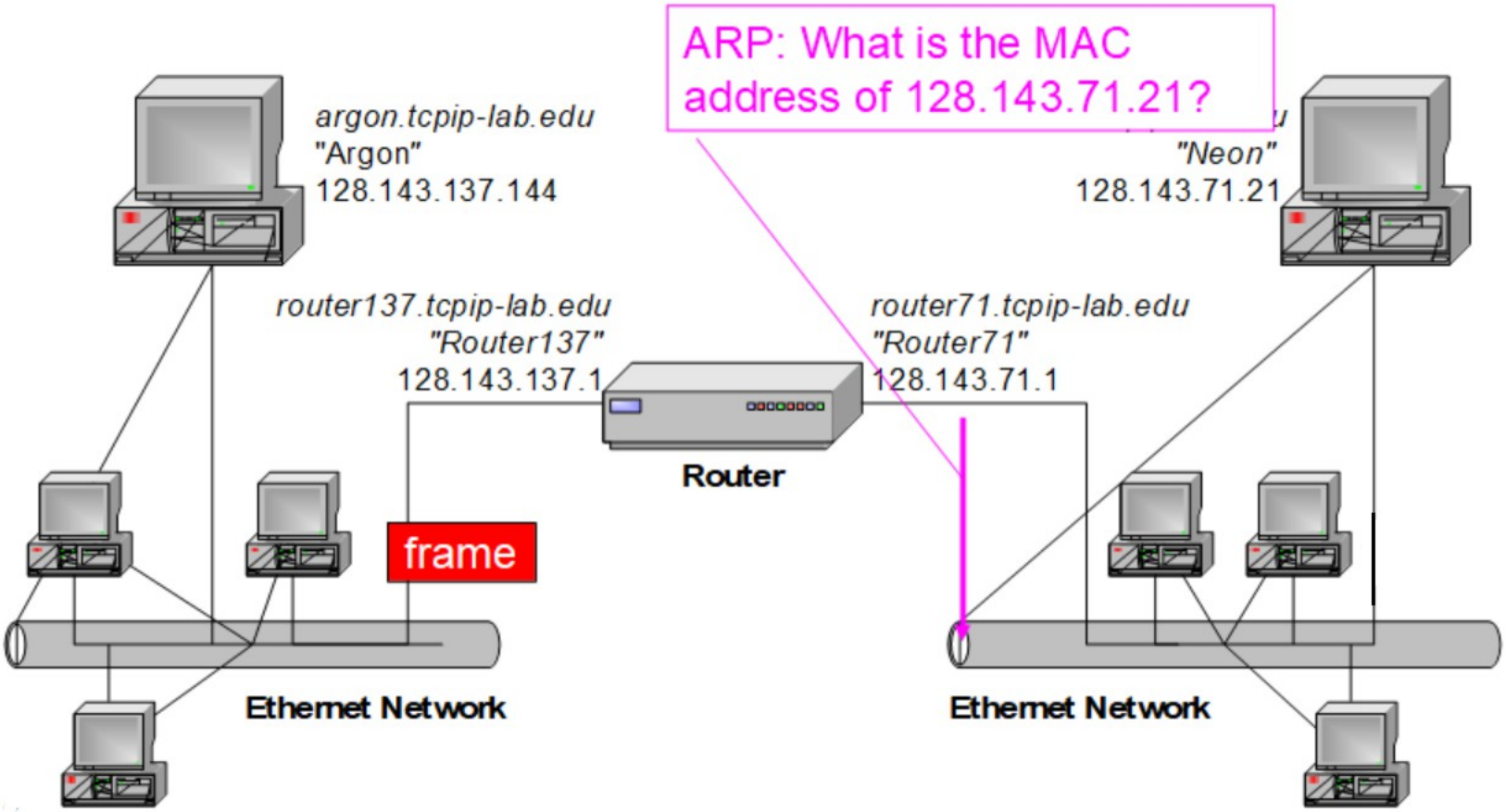
UN ESEMPIO



Quarto passo

- invio di un frame al router locale
- il router capisce che è collegato direttamente alla rete locale del destinatario

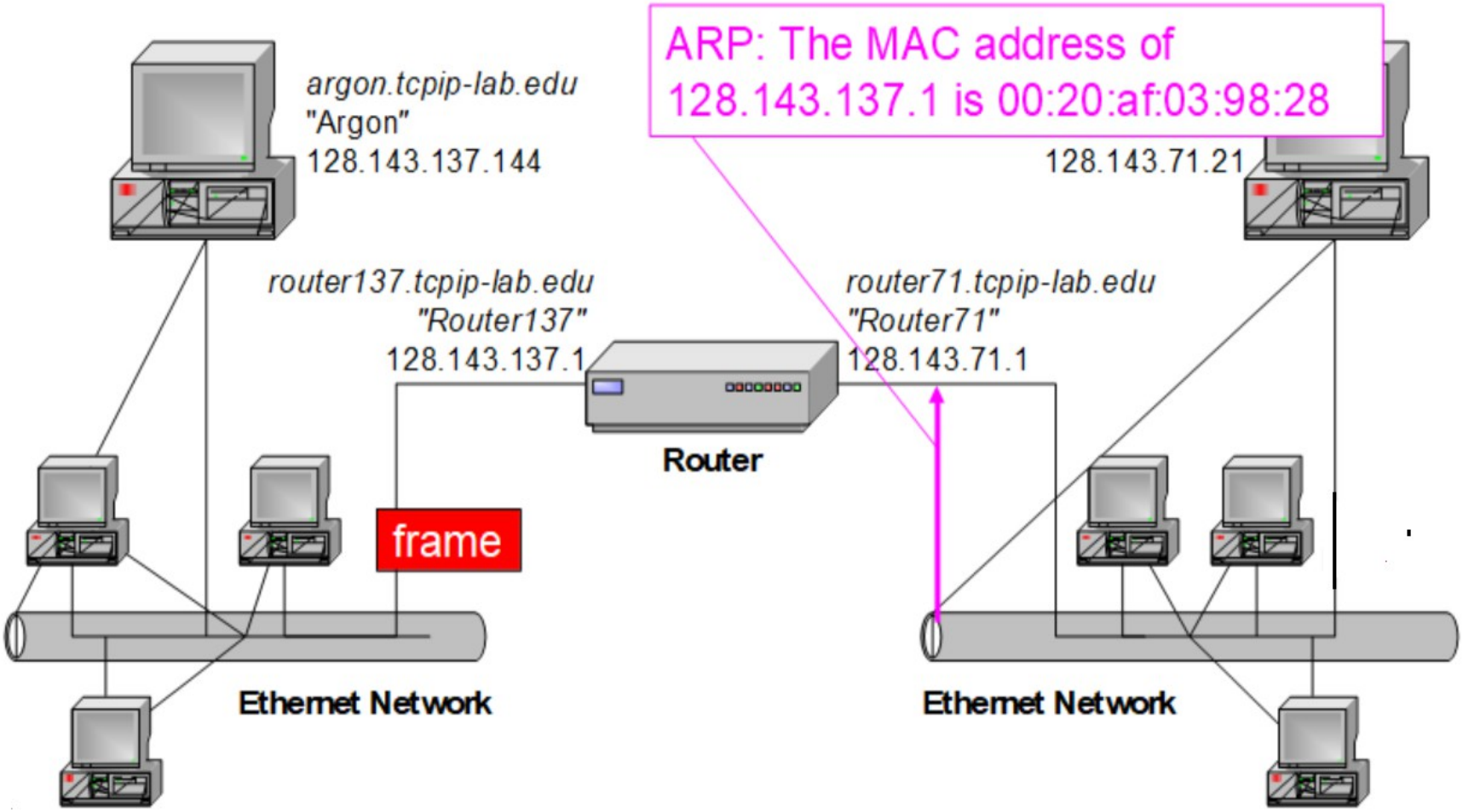
UN ESEMPIO



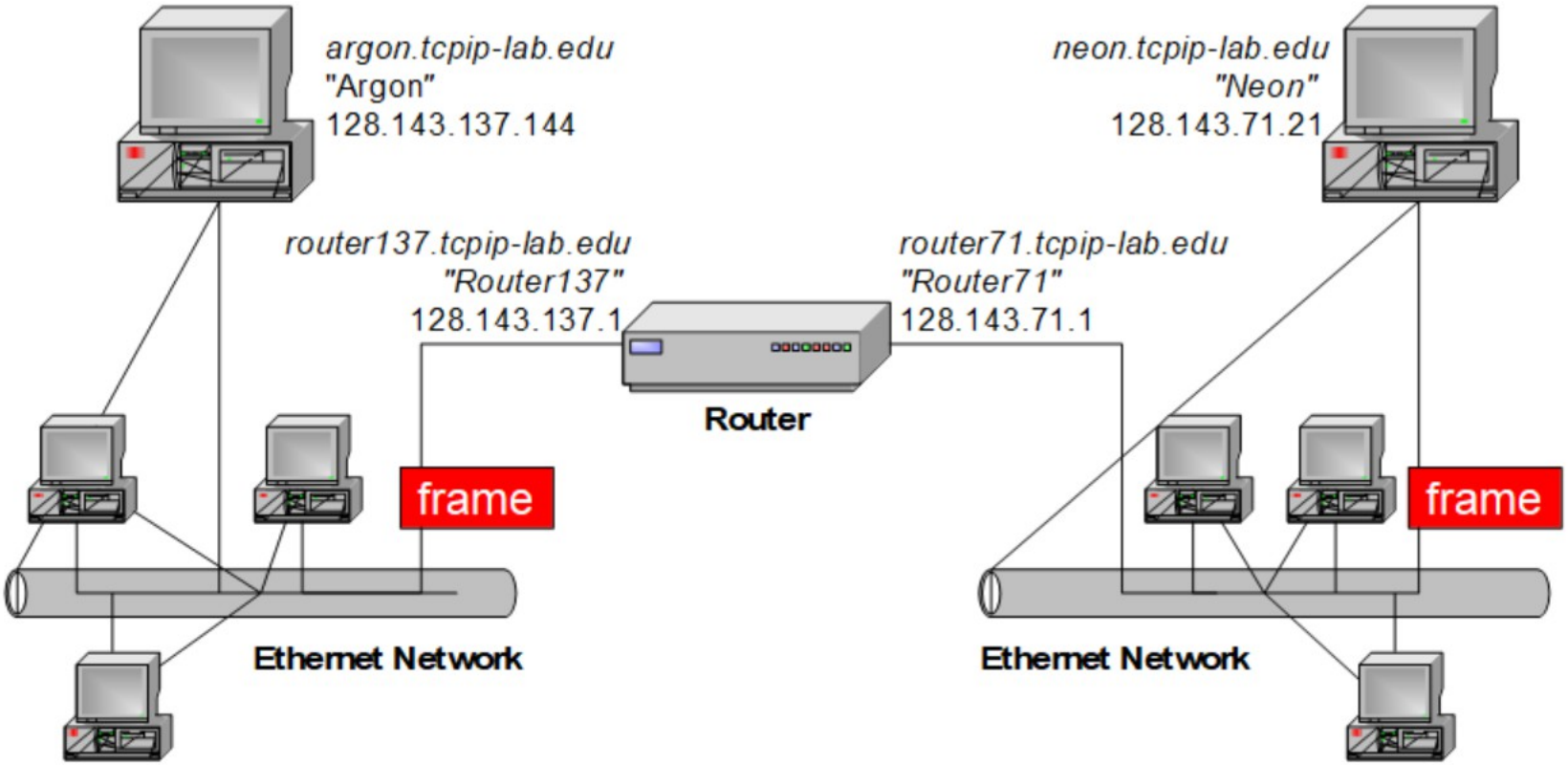
Quinto passo

- Procedimento inverso per l'ARP dell'altra rete: da indirizzo IP a MAC address

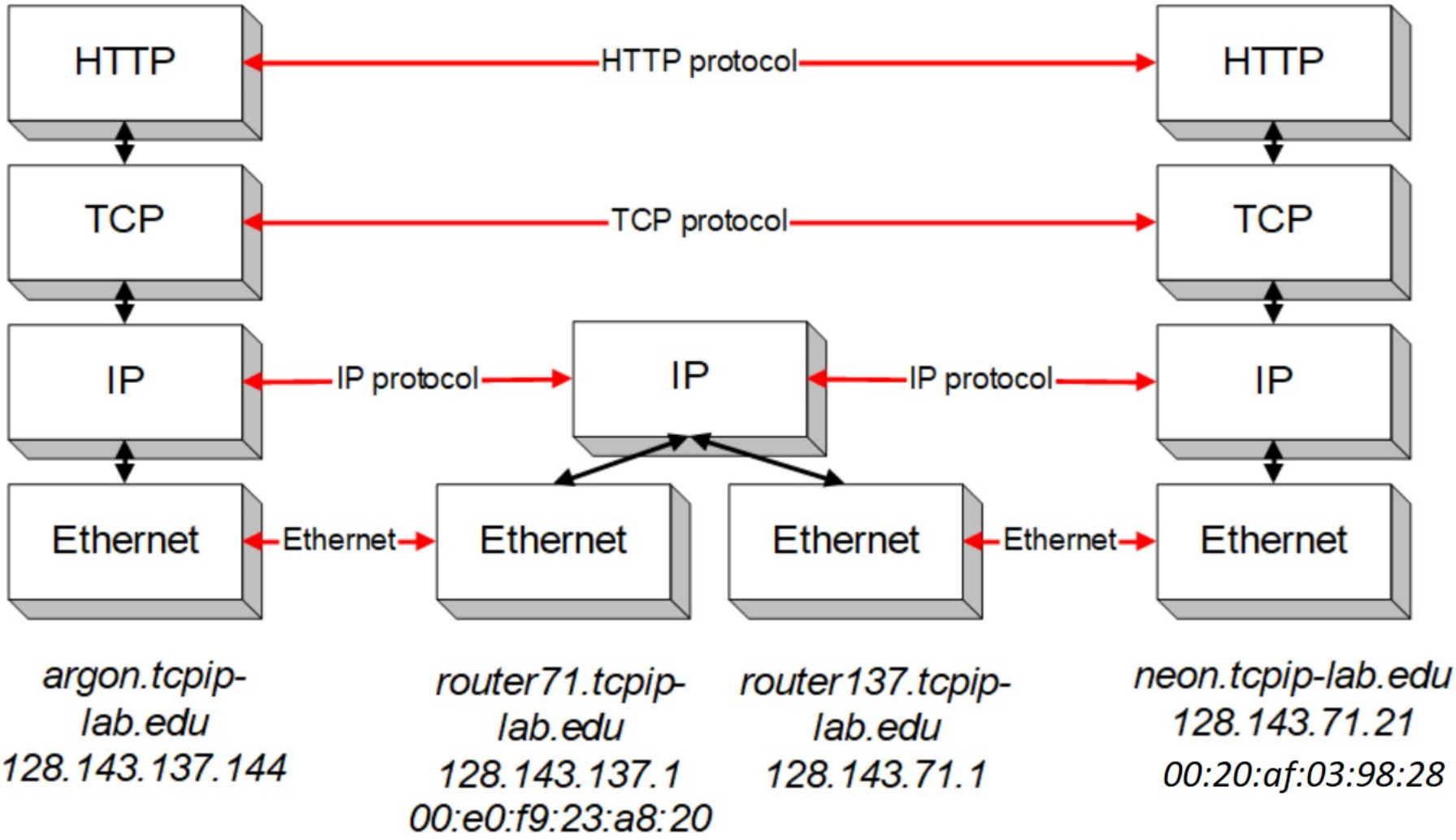
UN ESEMPIO


ARP: The MAC address of
128.143.137.1 is 00:20:af:03:98:28
argon.tcpip-lab.edu
"Argon"
128.143.137.144
128.143.71.21
router137.tcpip-lab.edu
"Router137"
128.143.137.1
router71.tcpip-lab.edu
"Router71"
128.143.71.1
Router
frame
Ethernet Network
Ethernet Network

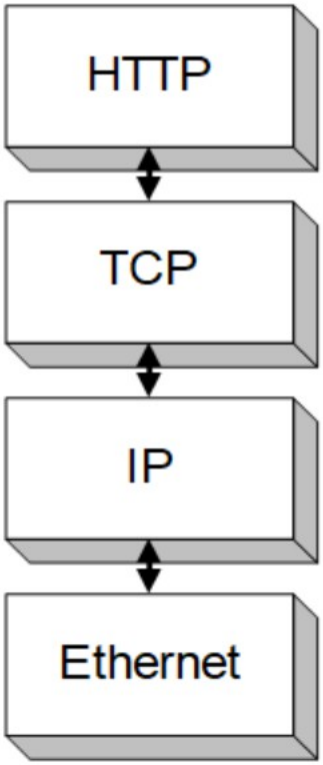
UN ESEMPIO


argon.tcpip-lab.edu
"Argon"
128.143.137.144
neon.tcpip-lab.edu
"Neon"
128.143.71.21
router137.tcpip-lab.edu
"Router137"
128.143.137.1
router71.tcpip-lab.edu
"Router71"
128.143.71.1
Router
frame
frame
Ethernet Network
Ethernet Network

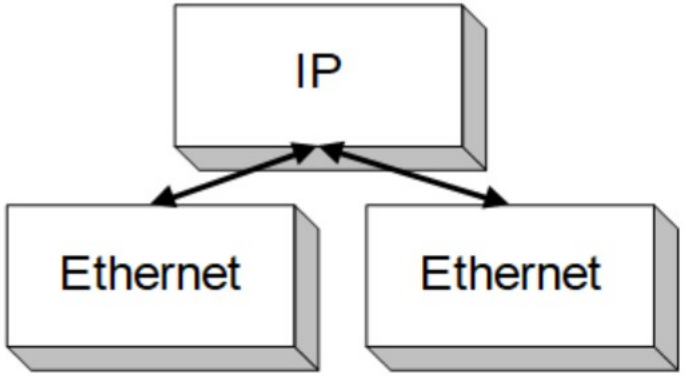
LAYERS NELL'ESEMPIO



LAYERS NELL'ESEMPIO

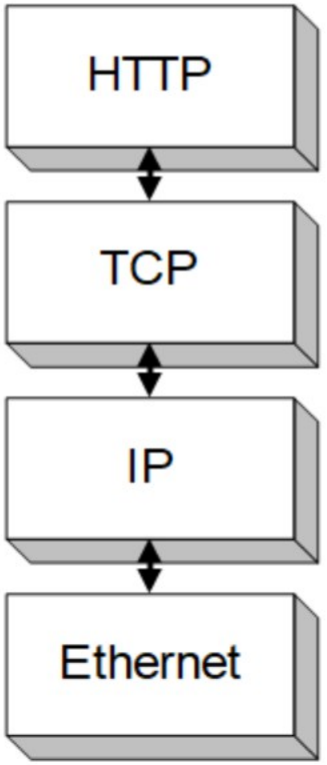


argon.tcpip-lab.edu
128.143.137.144



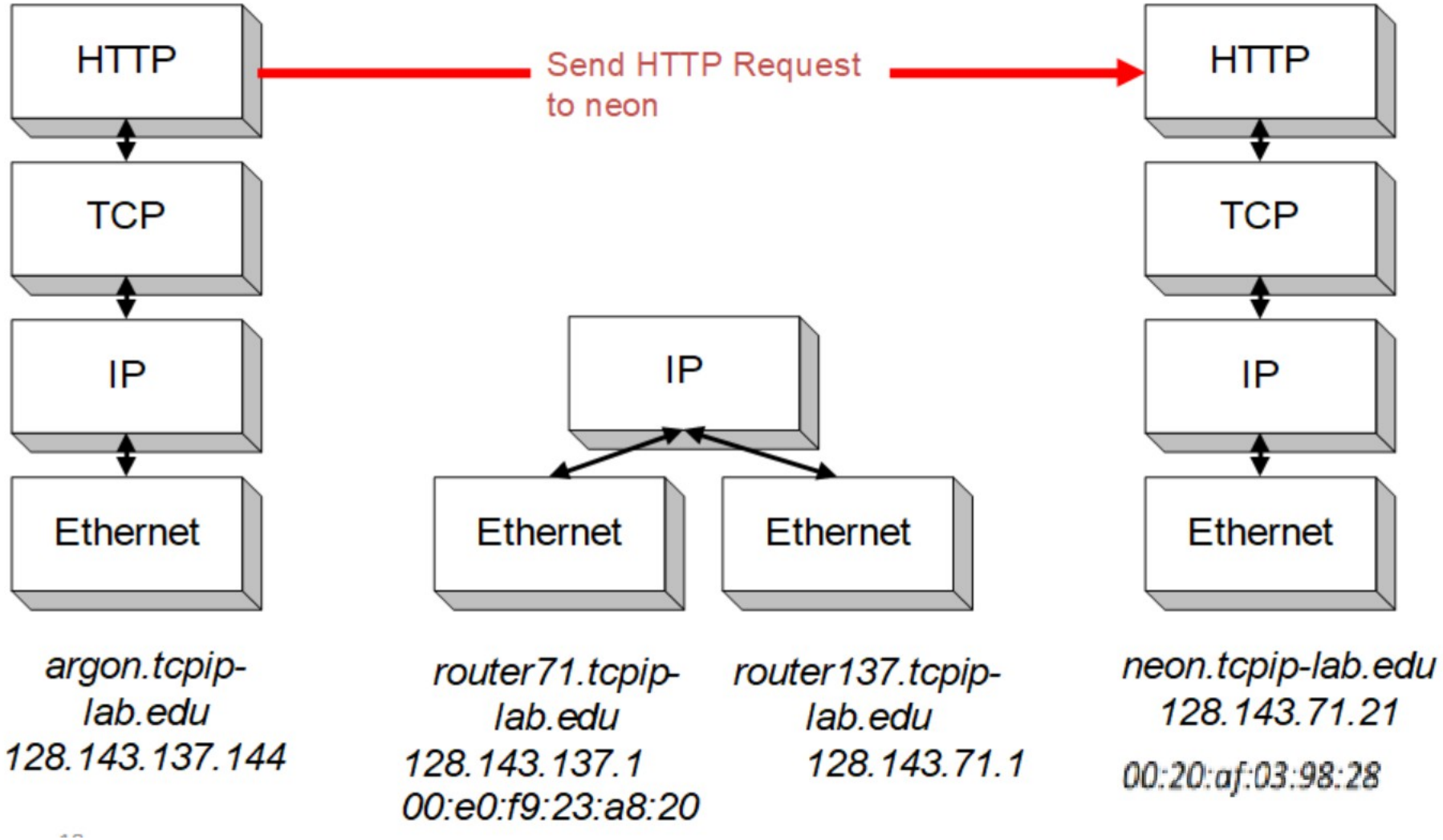
router71.tcpip-lab.edu
128.143.137.1
00:e0:f9:23:a8:20

router137.tcpip-lab.edu
128.143.71.1

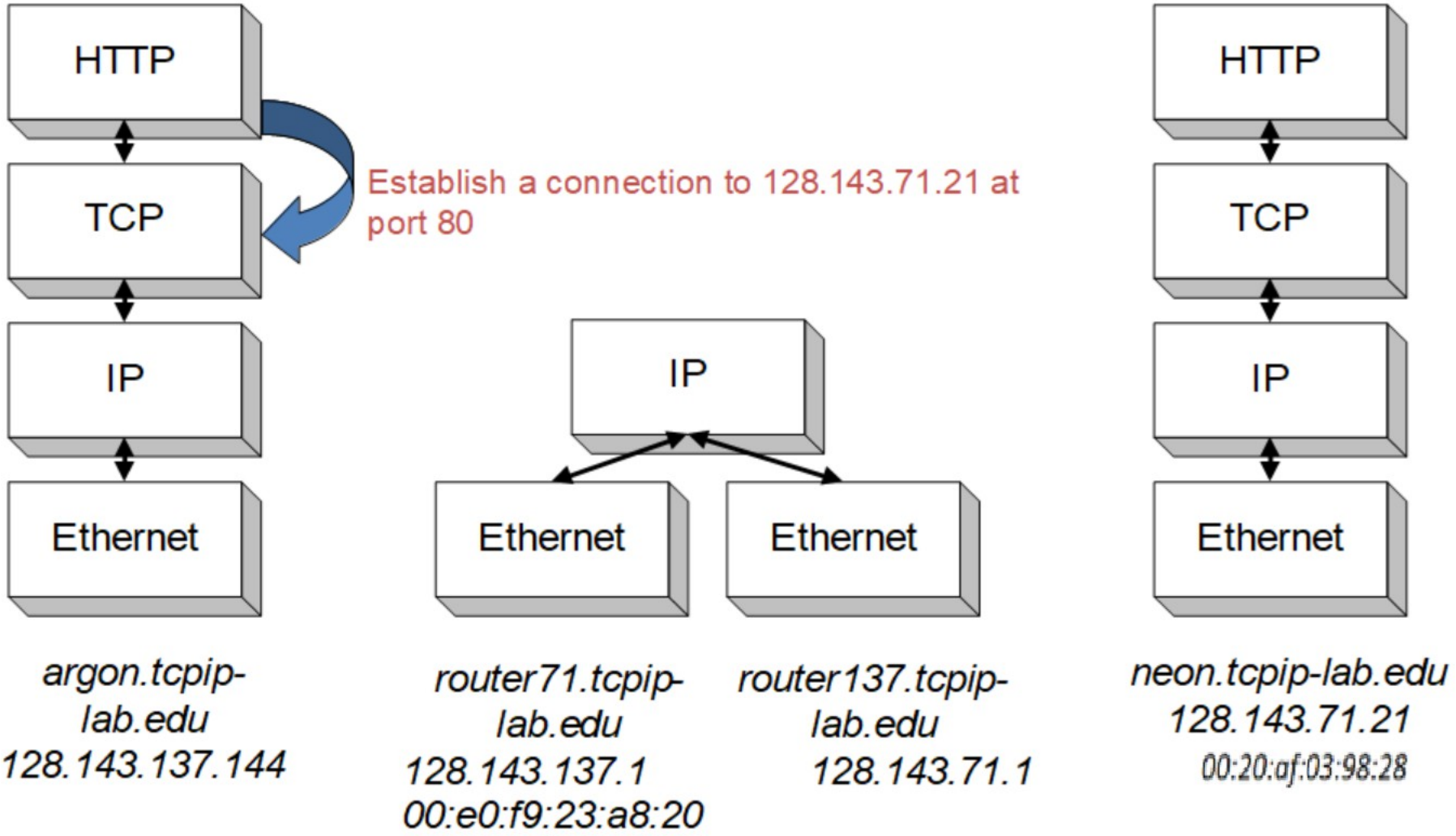


neon.tcpip-lab.edu
128.143.71.21
00:20:af:03:98:28

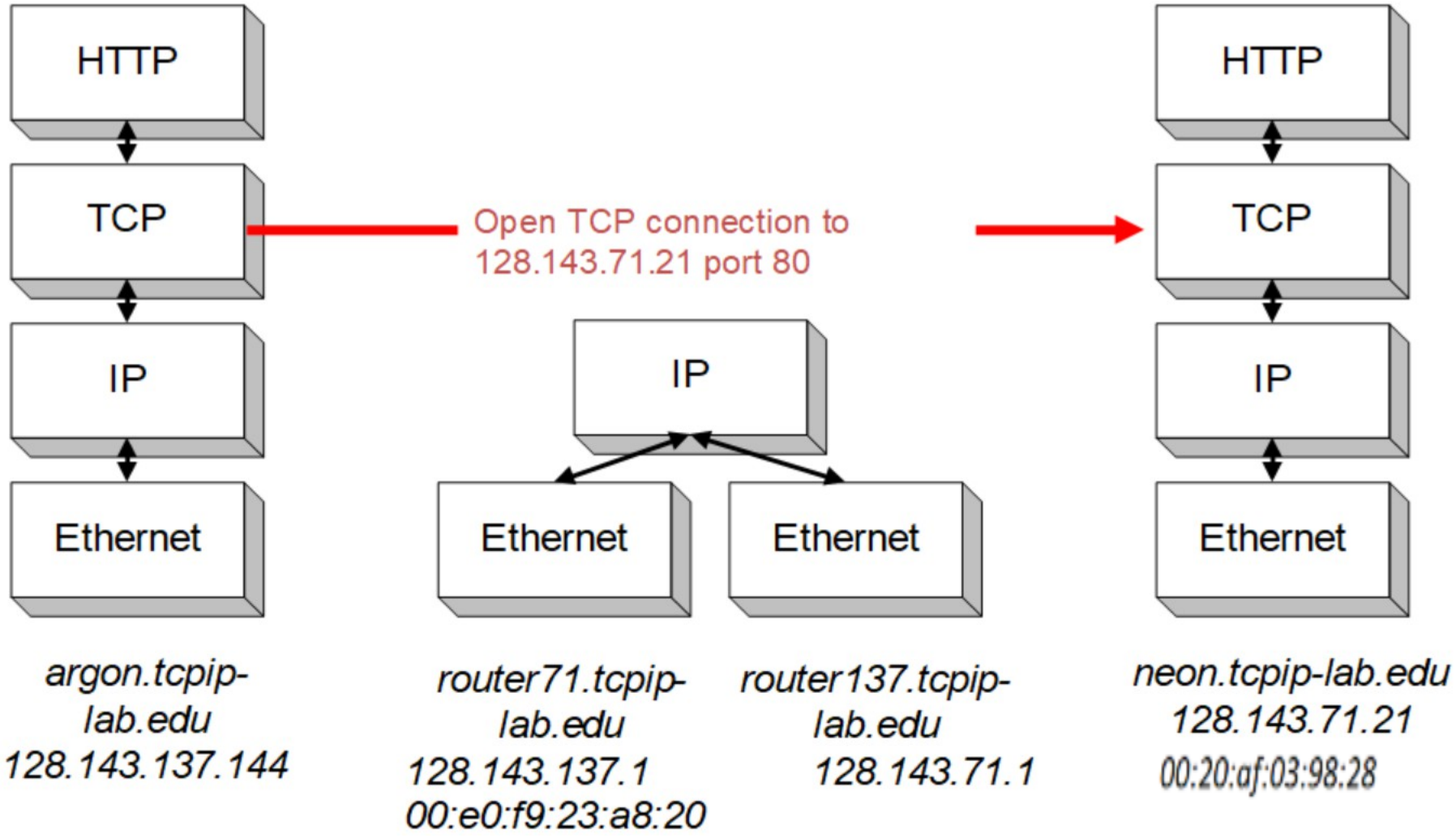
LAYERS NELL'ESEMPIO



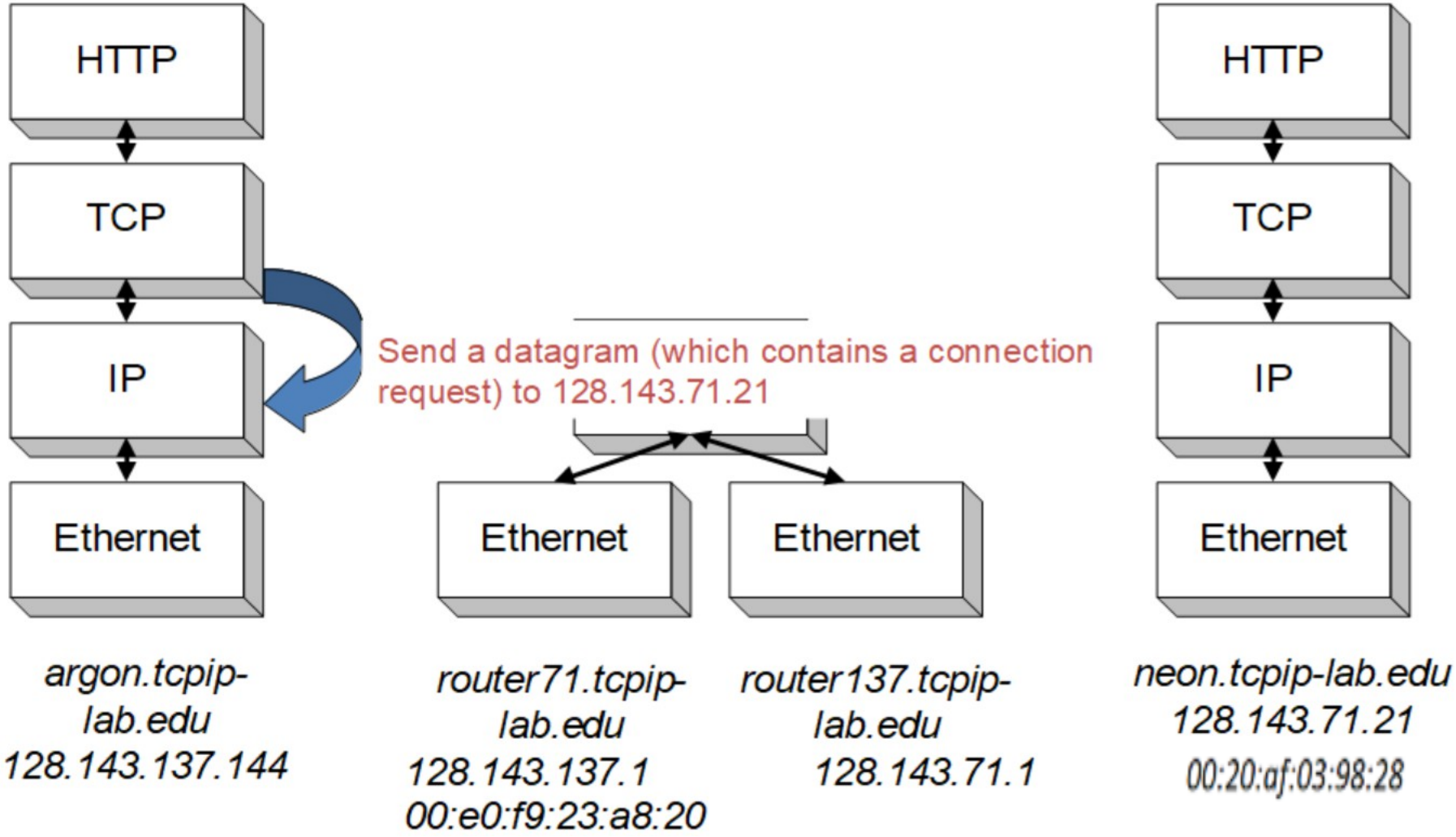
LAYERS NELL'ESEMPIO



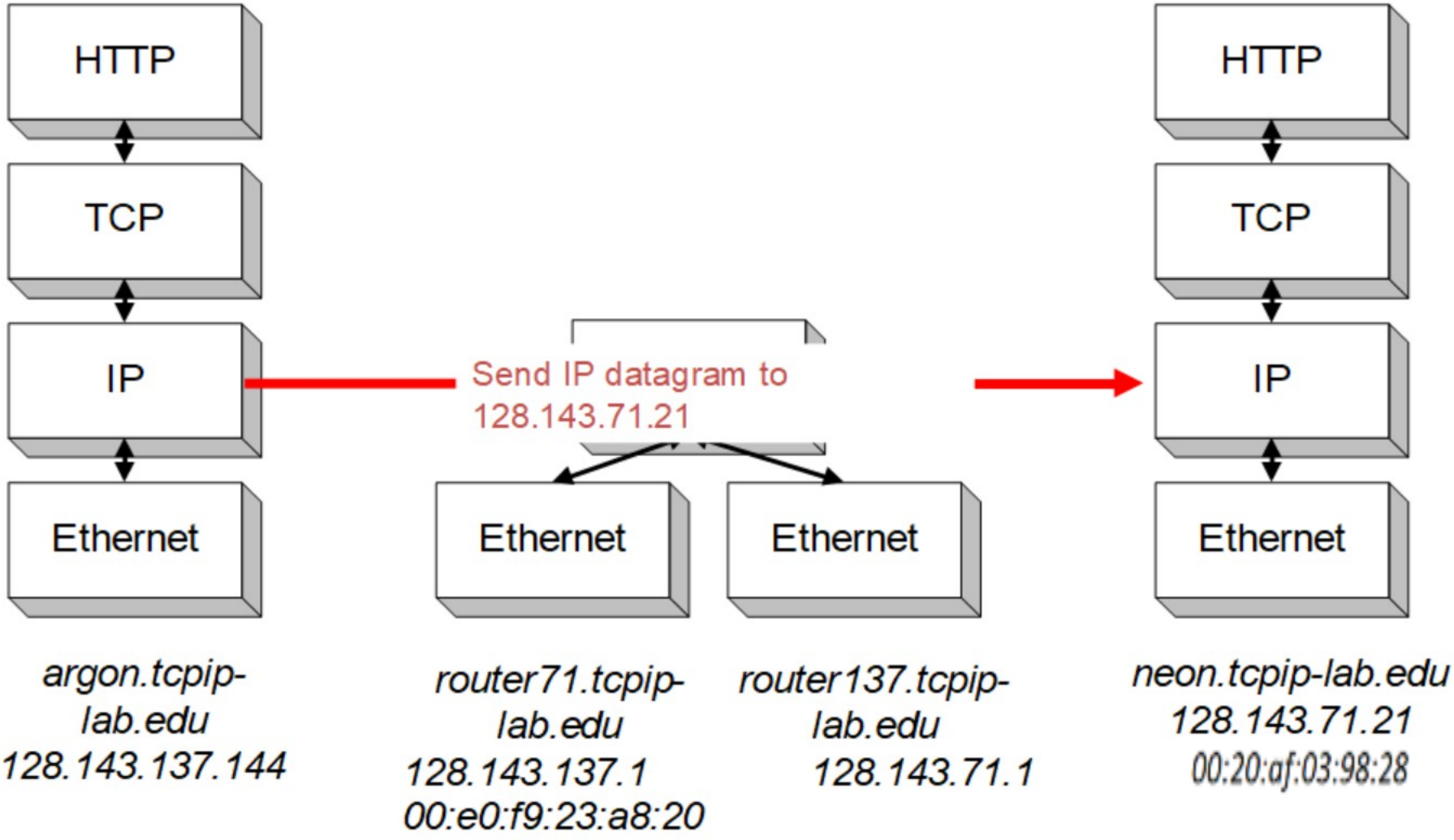
LAYERS NELL'ESEMPIO



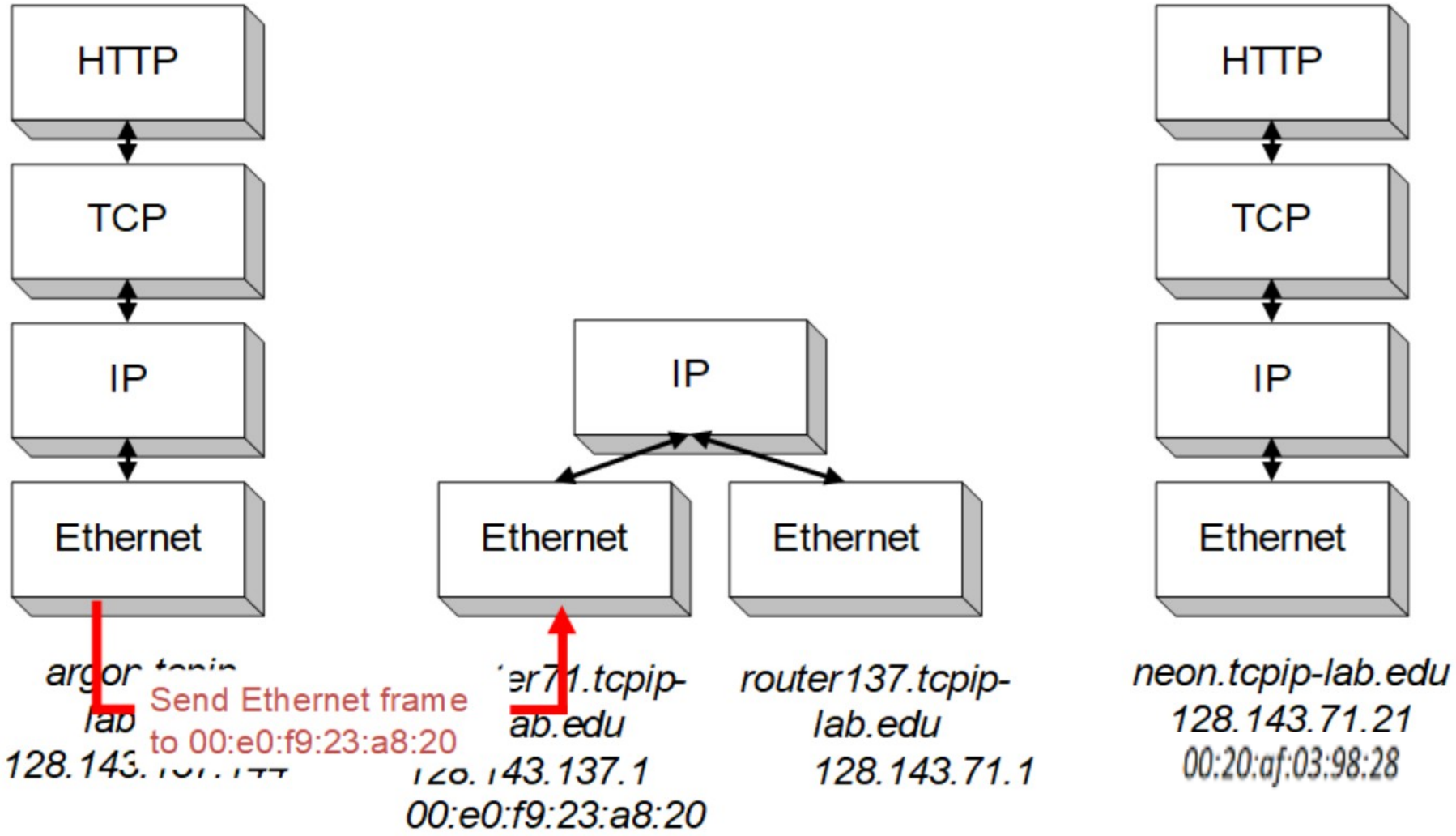
LAYERS NELL'ESEMPIO



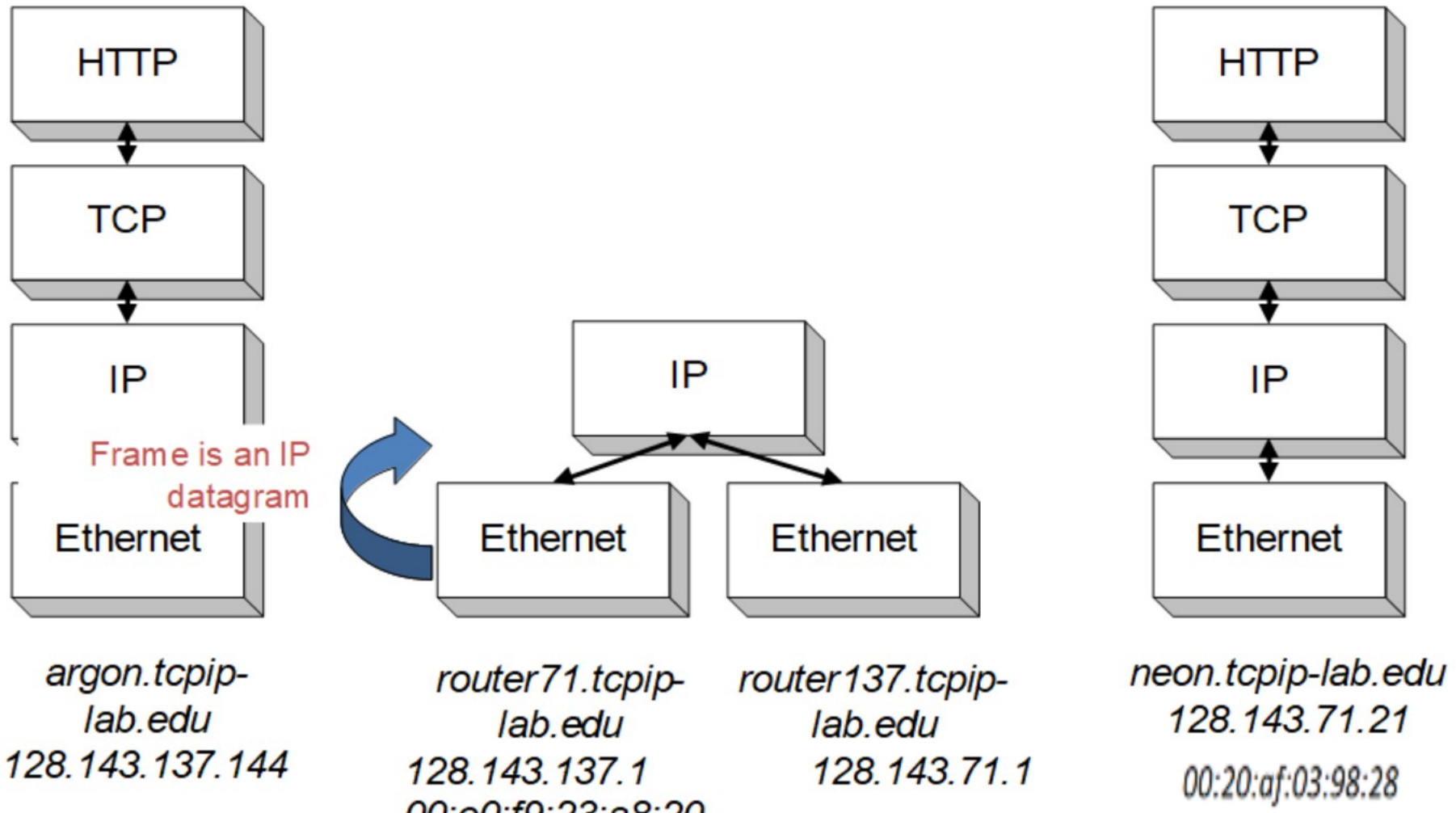
LAYERS NELL'ESEMPIO



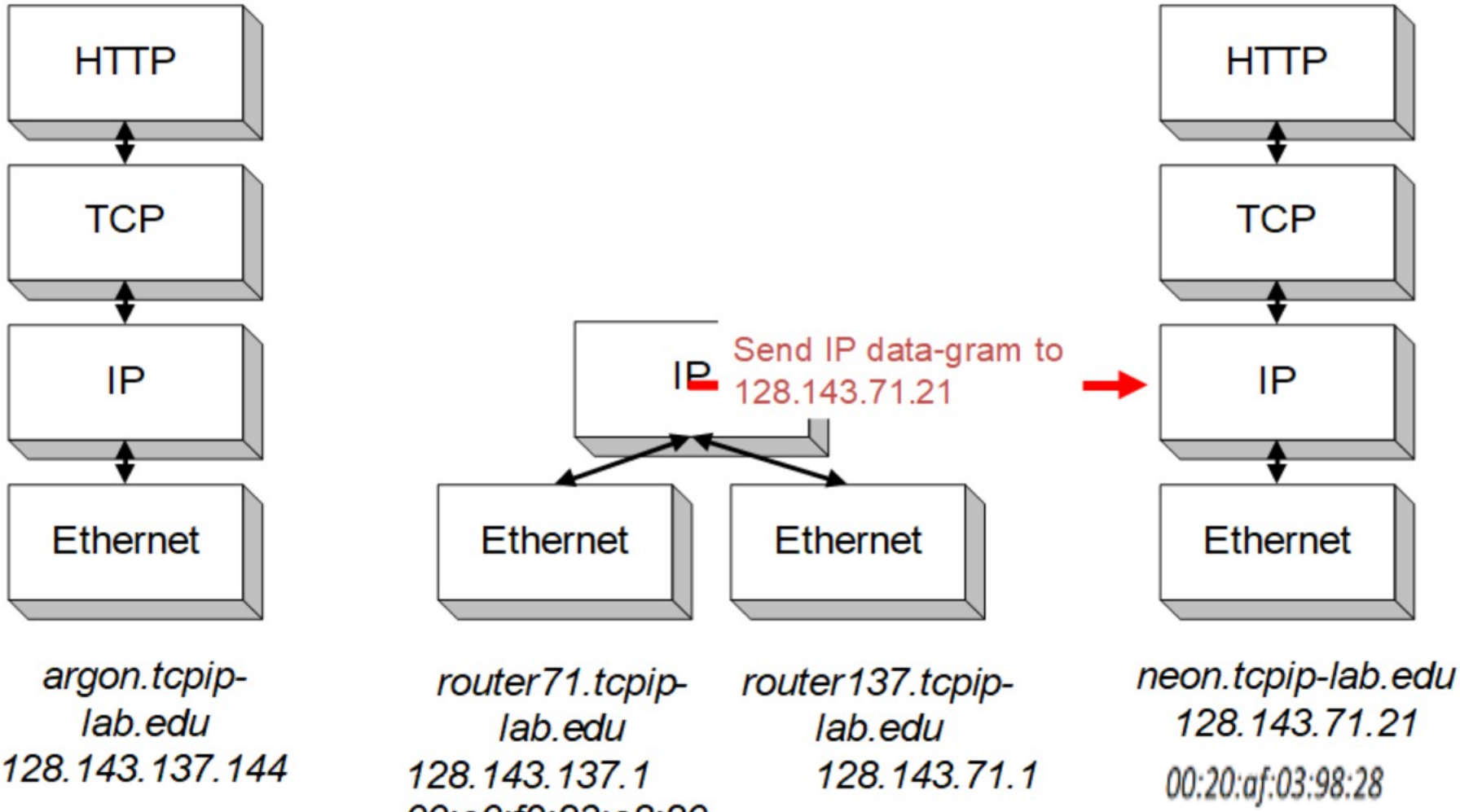
LAYERS NELL'ESEMPIO



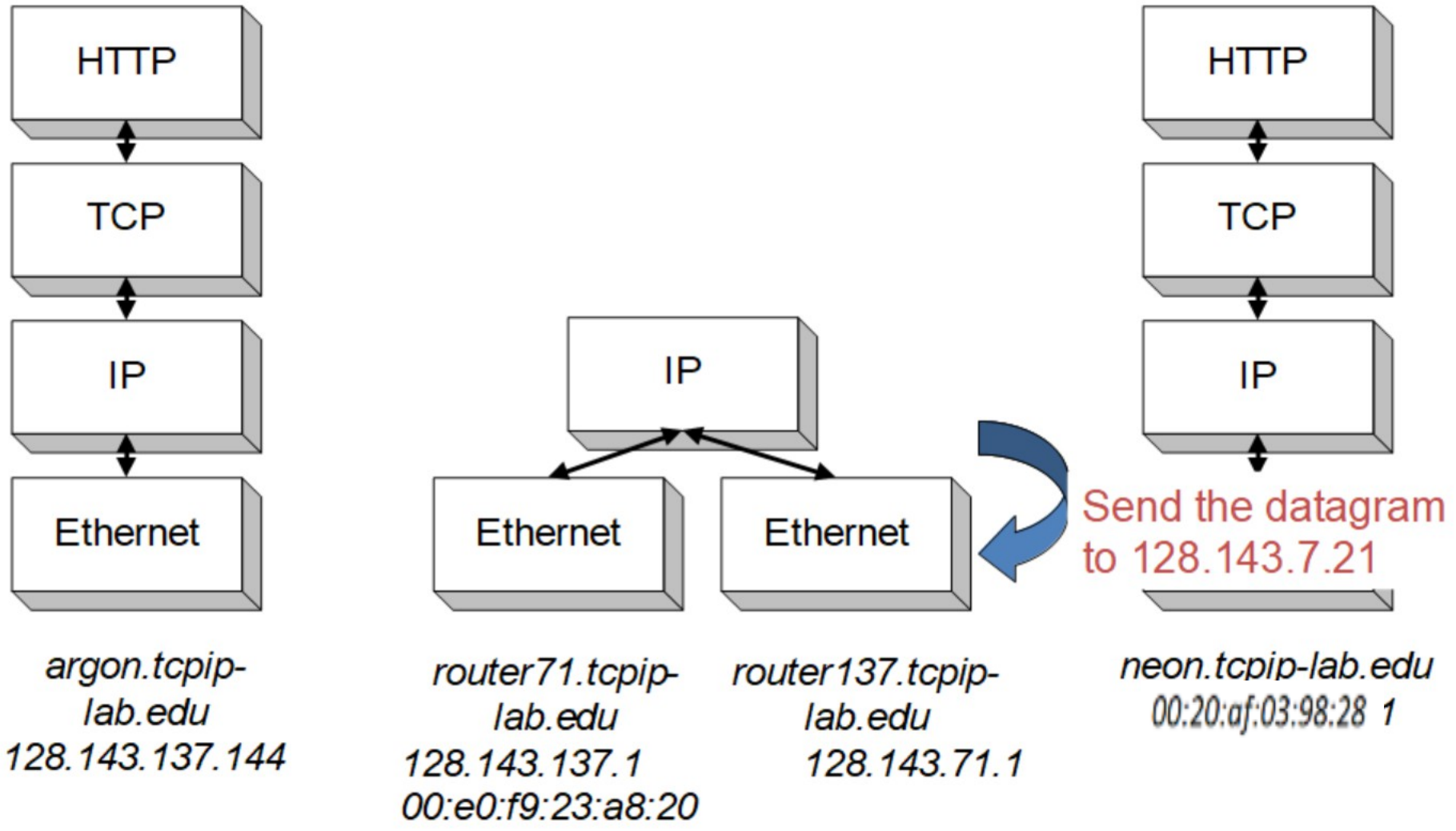
LAYERS NELL'ESEMPIO



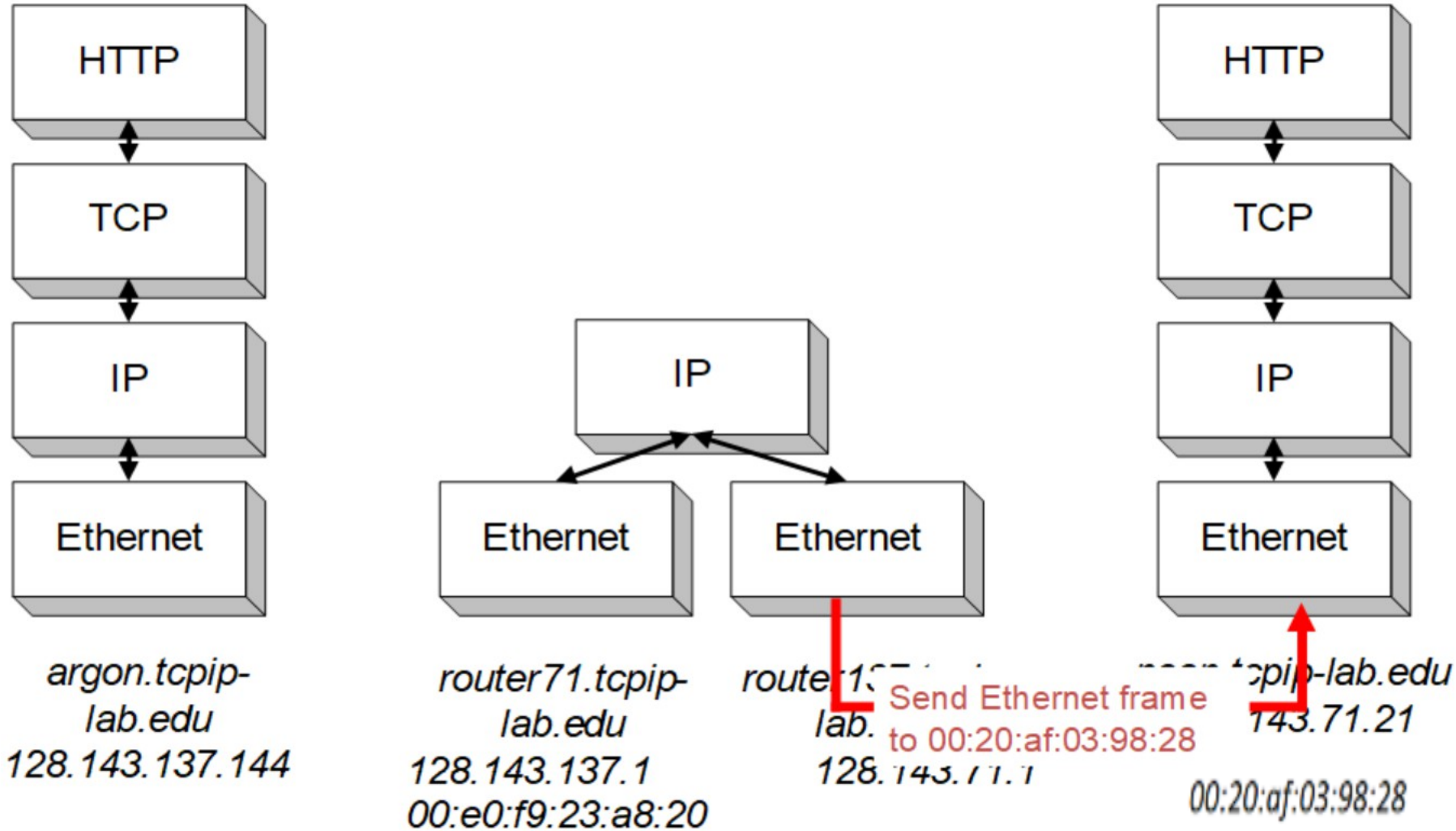
LAYERS NELL'ESEMPIO



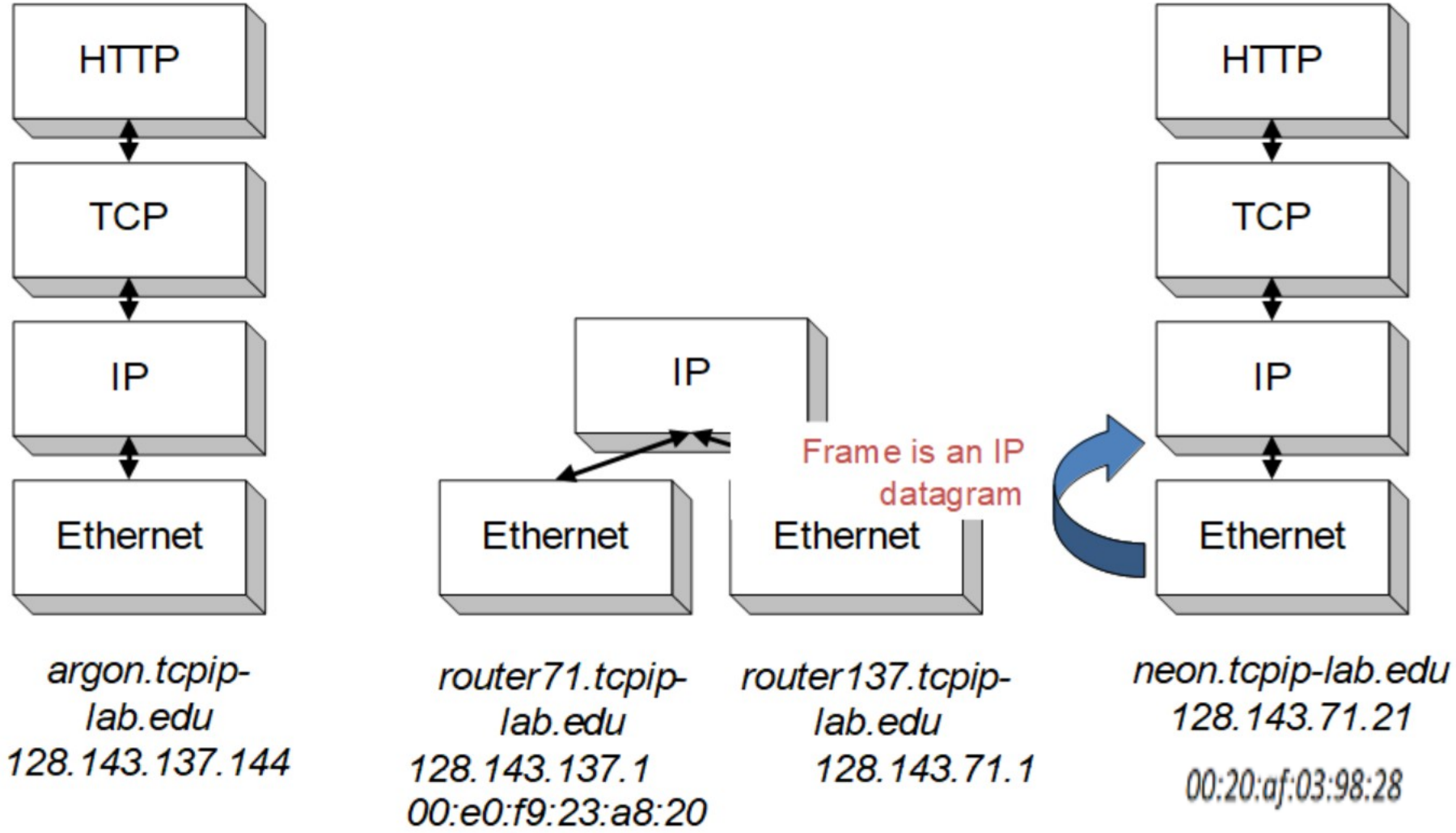
LAYERS NELL'ESEMPIO



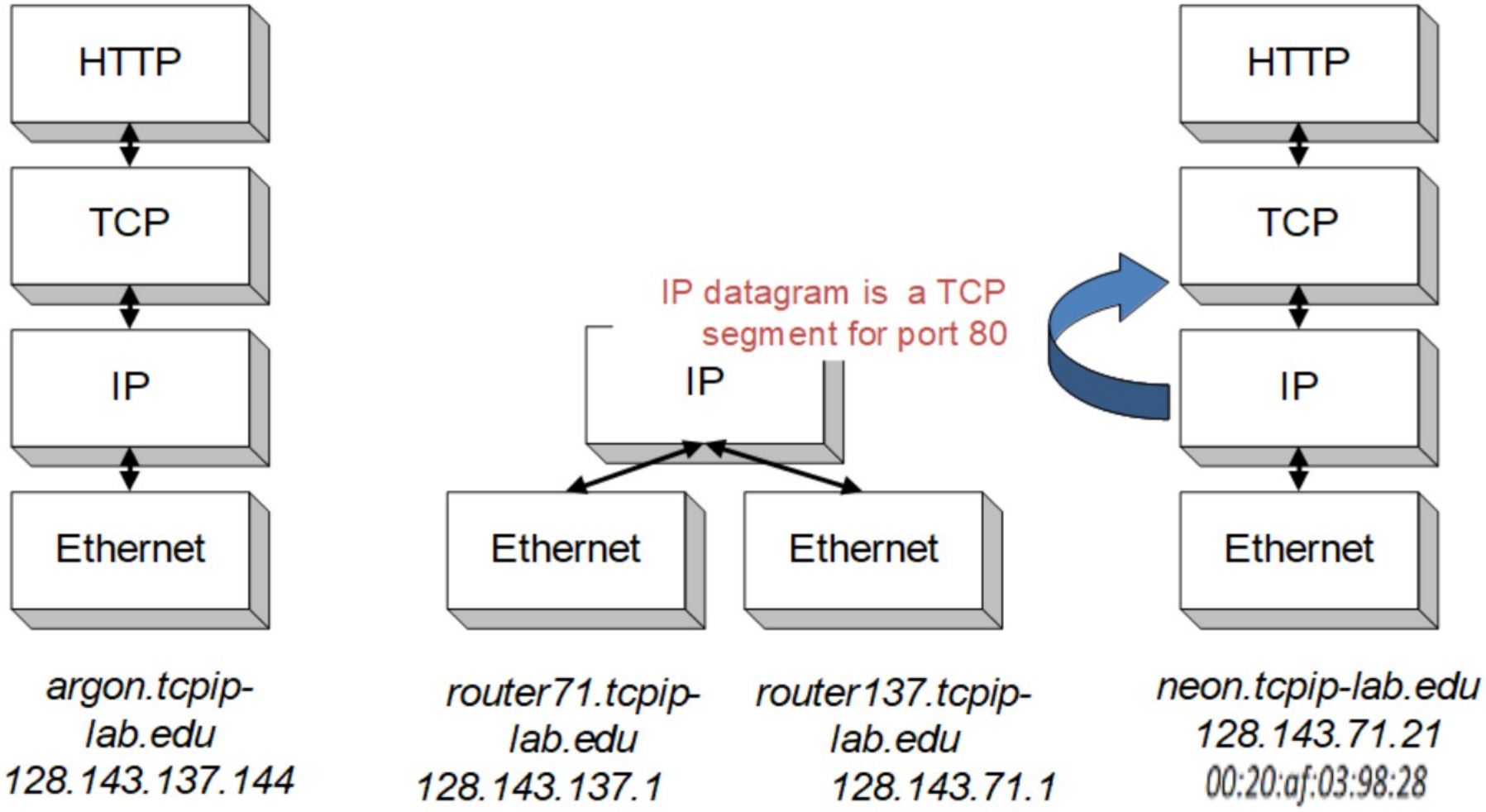
LAYERS NELL'ESEMPIO



LAYERS NELL'ESEMPIO



LAYERS NELL'ESEMPIO

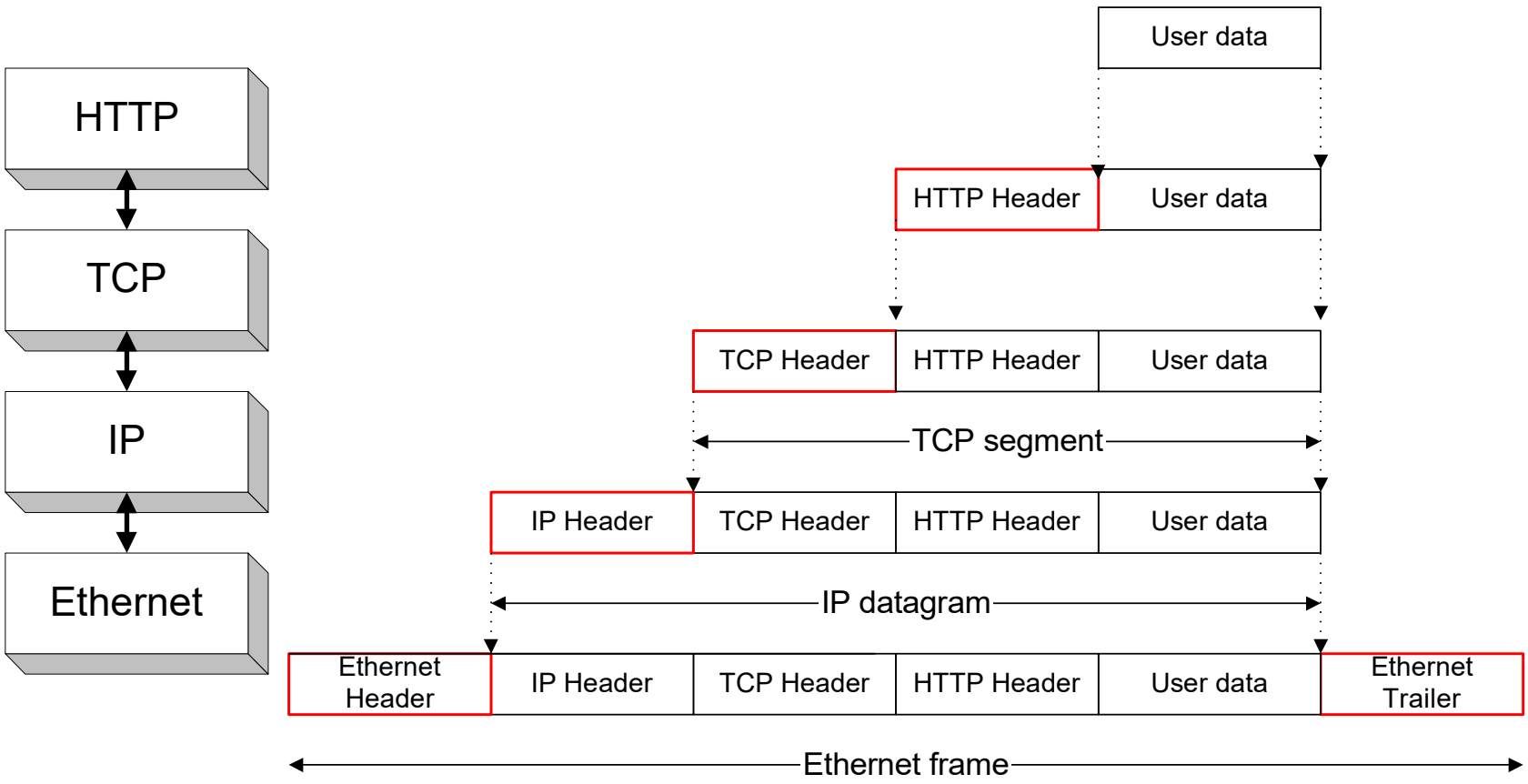


LAYERS E SERVIZI

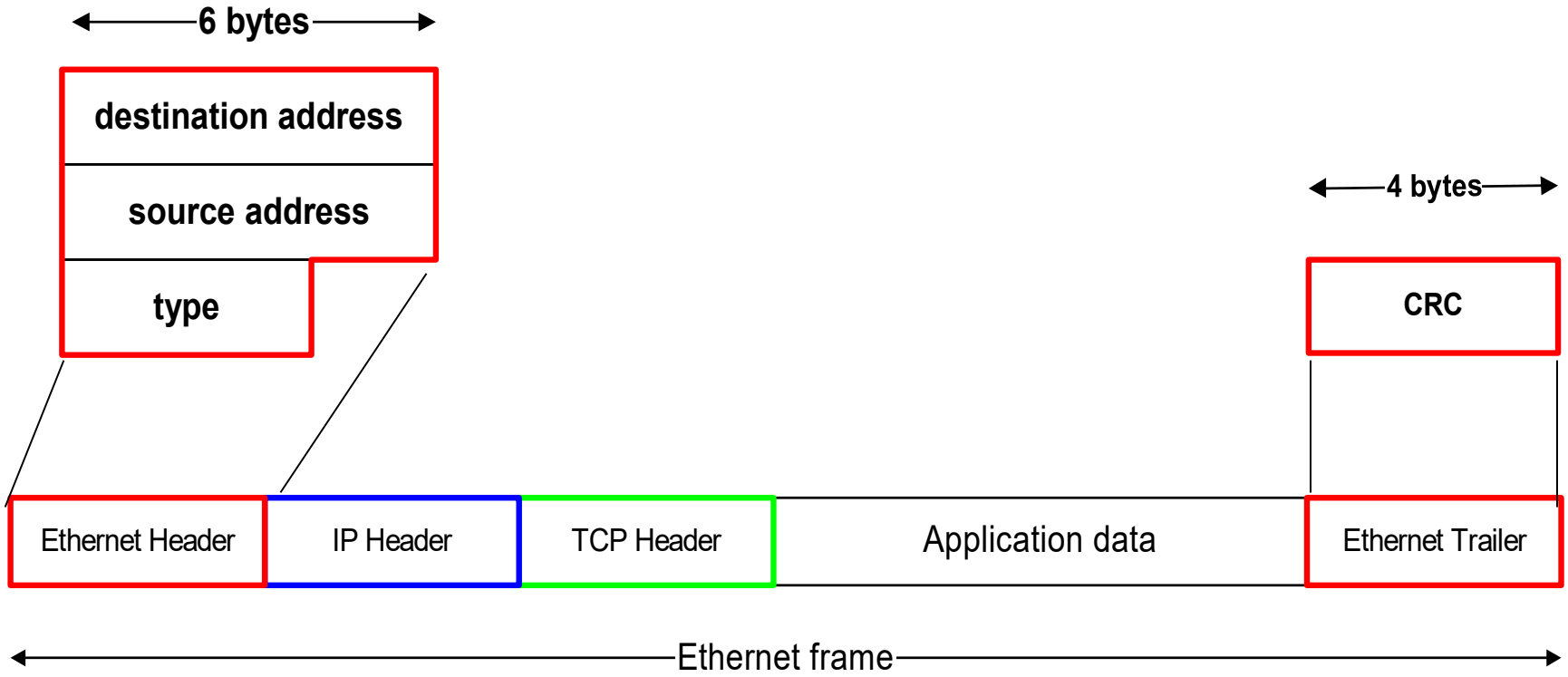
- Servizio fornito da TCP a HTTP:
  - trasmissione affidabile di dati su una connessione logica
- Servizio fornito da IP a TCP:
  - trasmssione non affidabile di datagrammi IP datagrams su una rete IP
- Servizio fornito da Ethernet ad IP
  - trasmissione di un frame su un segmento Ethernet
- Altri servizi:
  - DNS: traduzione nomi di dominio indirizzi IP
  - ARP: traduzione indirizzi IP indirizzi MAC

INCAPSULAMENTO E DEMULTIPLEXING

- i dati si muovono nello stack di protocolli, ogni protocollo aggiunge informazione di controllo specifica di quel livello

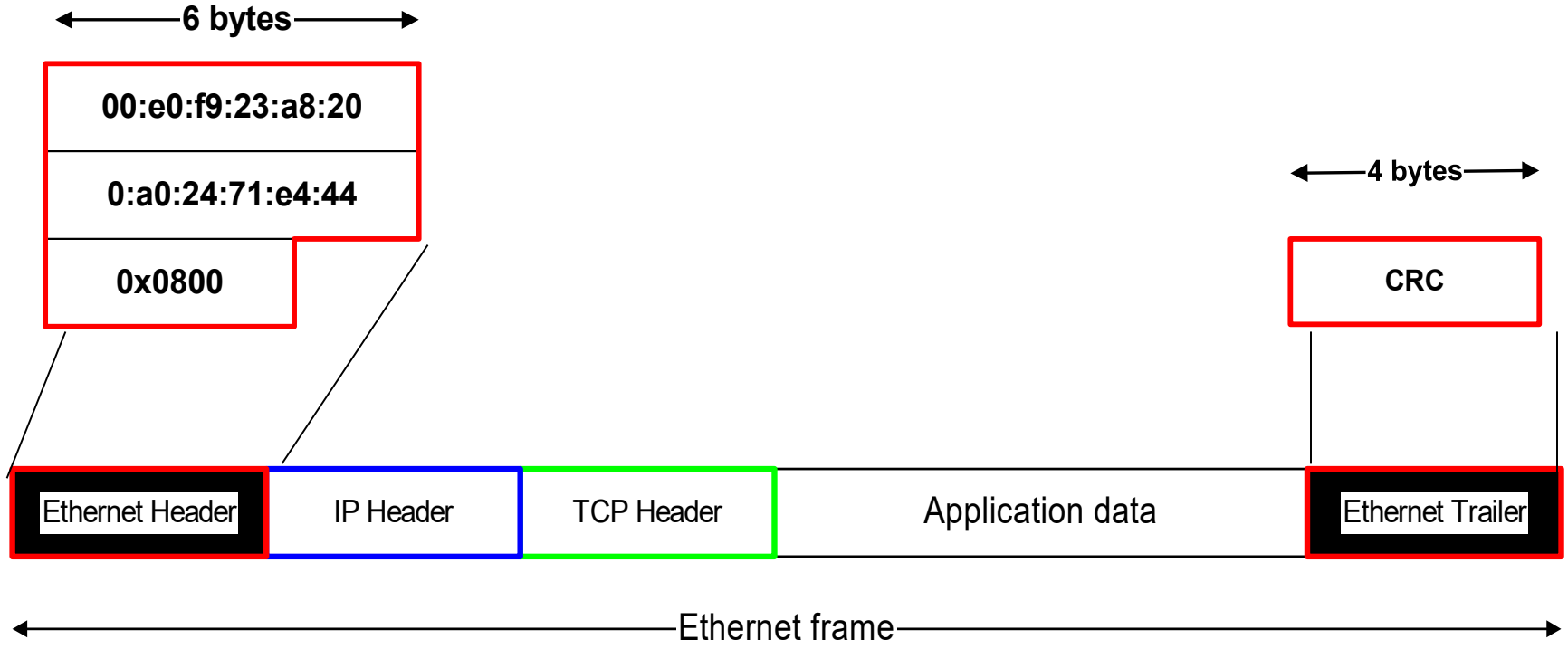


INCAPSULAMENTO E DEMULTIPLEXING

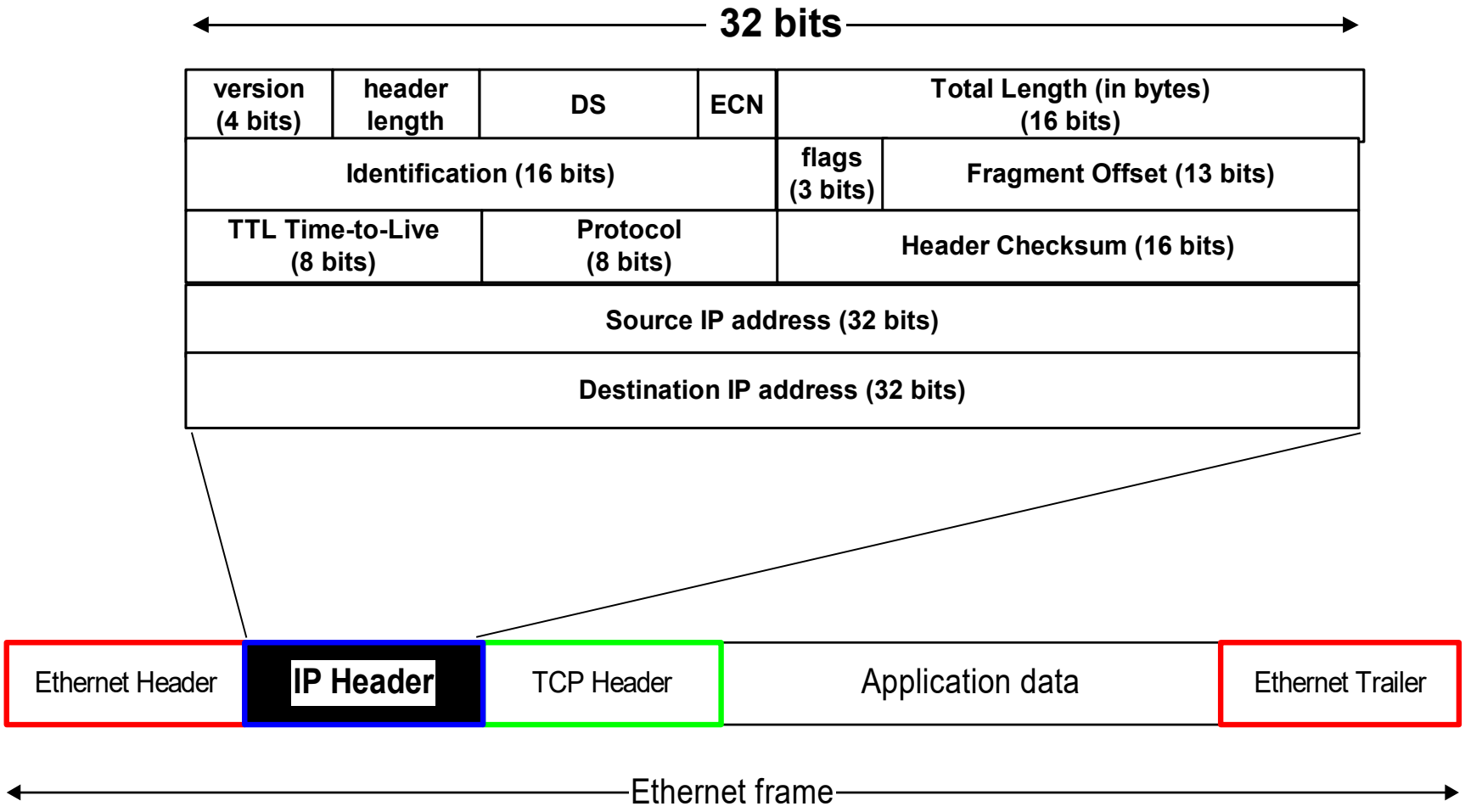


79

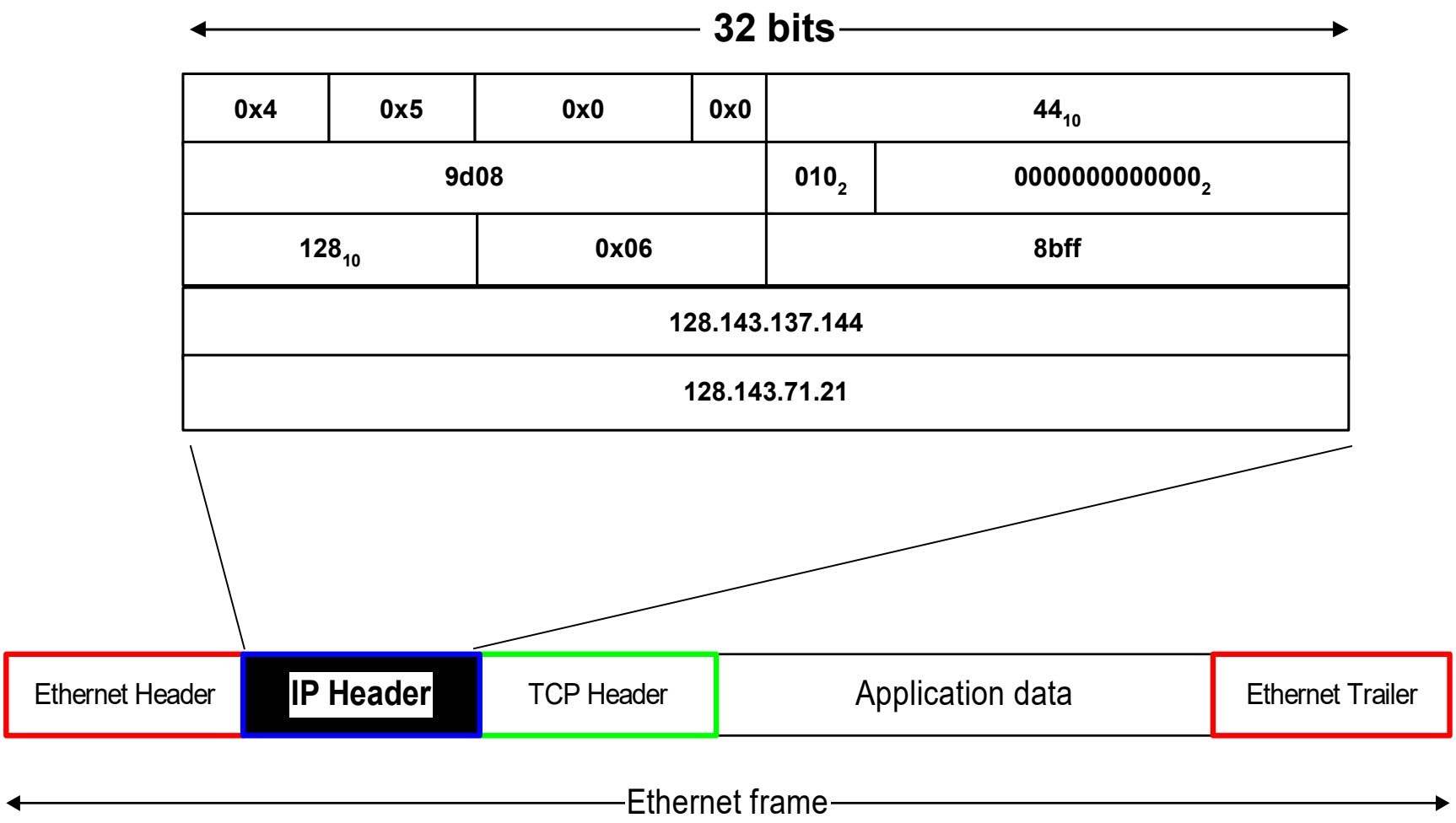
IP HEADER


6 bytes
00:e0:f9:23:a8:20
0:a0:24:71:e4:44
0x0800
4 bytes
CRC
Ethernet Header
IP Header
TCP Header
Application data
Ethernet Trailer
Ethernet frame

IP HEADER

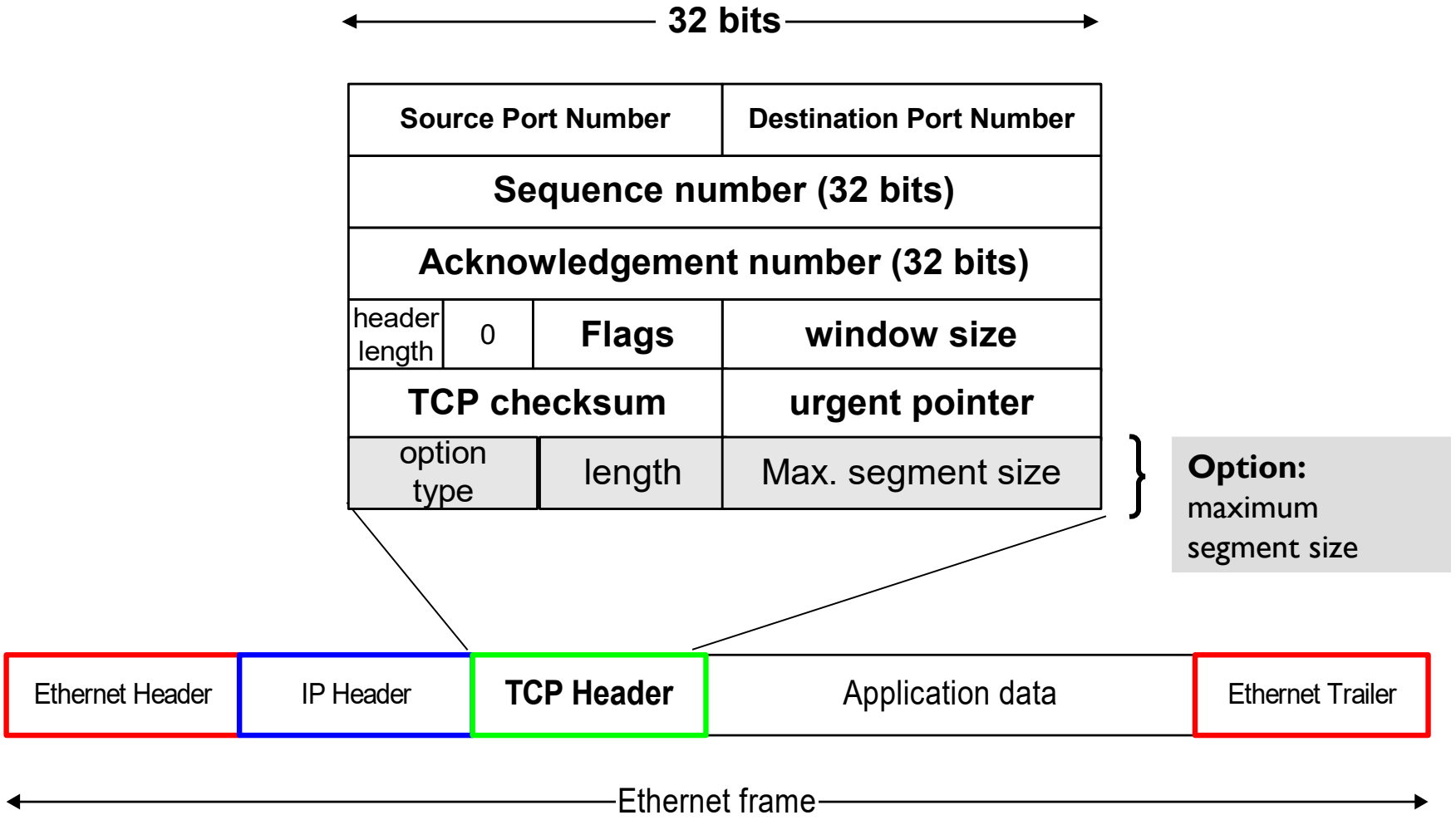

32 bits
version (4 bits)
header length
DS
ECN
Total Length (in bytes) (16 bits)
Identification (16 bits)
flags (3 bits)
Fragment Offset (13 bits)
TTL Time-to-Live (8 bits)
Protocol (8 bits)
Header Checksum (16 bits)
Source IP address (32 bits)
Destination IP address (32 bits)
Ethernet Header
IP Header
TCP Header
Application data
Ethernet Trailer
Ethernet frame

IP HEADER

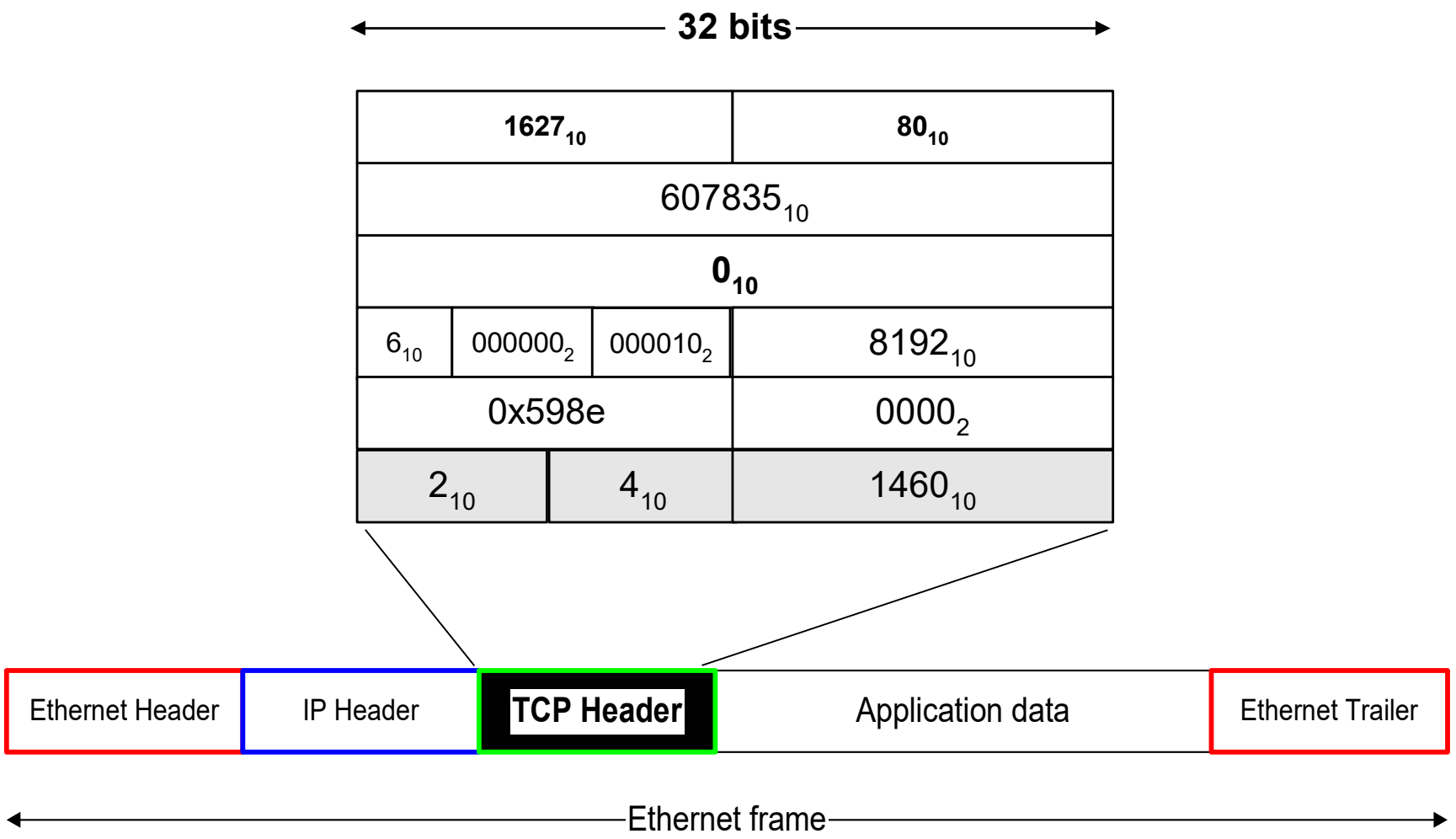

32 bits
0x4
0x5
0x0
0x0
44_{10}
9d08
010_2
0000000000000_2
128_{10}
0x06
8bff
128.143.137.144
128.143.71.21
Ethernet Header
IP Header
TCP Header
Application data
Ethernet Trailer
Ethernet frame

82

TCP HEADER


32 bits
Source Port Number
Destination Port Number
Sequence number (32 bits)
Acknowledgement number (32 bits)
header length
0
Flags
window size
TCP checksum
urgent pointer
option type
length
Max. segment size
Option:
maximum segment size
Ethernet Header
IP Header
TCP Header
Application data
Ethernet Trailer
Ethernet frame

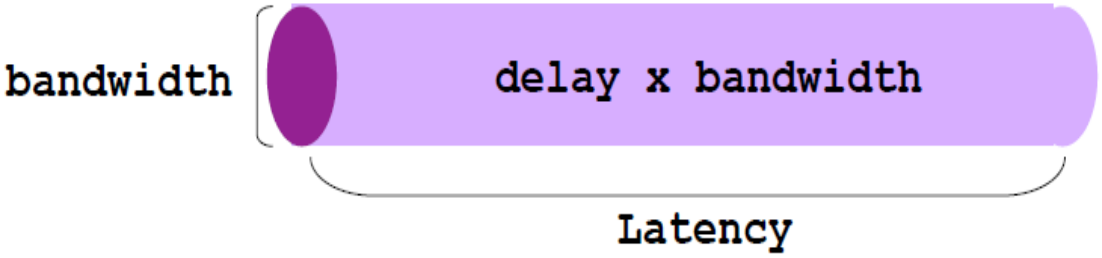
TCP HEADER


32 bits
1627 10
80 10
607835 10
0 10
6 10
000000 2
000010 2
8192 10
0x598e
0000 2
2 10
4 10
1460 10
Ethernet Header
IP Header
TCP Header
Application data
Ethernet Trailer
Ethernet frame

VALUTAZIONE DELLA PERFORMANCE

- fino a questo punto abbiamo discusso gli aspetti funzionali della rete
- un altro aspetto riguarda la performance
  - come misurare le prestazioni della rete?
  - quali sono le metriche più adatte?
- importante, perchè da esse si può dedurre l'effettiva fruibilità di una applicazione
  - impatto sulla "user experience"
- le applicazioni possono avere requisiti diversi

LINK PERFORMANCE: BANDA E LATENZA

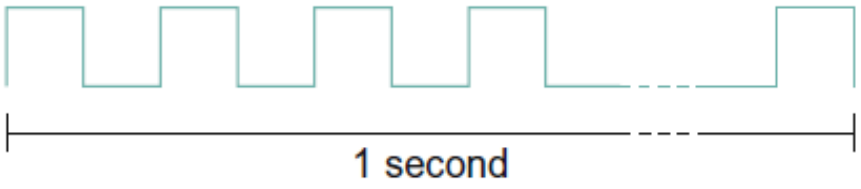


- **Bandwidth** (**Banda, capacità**): proporzionale alla "ampiezza" del link
  - quantità di dati (bits) che possono essere trasmessi ("inseriti nella linea") o ricevuti nell'unità di tempo (bits/secondo or bps)
- **Propagation delay (talvolta riferito come latency)**: proporzionale alla "lunghezza" del link
  - tempo di propagazione dei dati sul link (secondi)
  - talvolta Round Trip Time: tempo che un messaggio impiega a raggiungere una destinazione + tempo della risposta
- **Bandwidth-Delay Product (BDP)**: "volume" del link
  - latenza (o round trip time) × bits/time = bits totali sul link
  - quantità massima di dati "in transito" in ogni istante

BANDA

- una rete con una banda di 10 milioni di bits/secondo (Mbps), è in grado di inviare sul mezzo trasmissivo 10 milioni di bit per ogni secondo
  - ad esempio, per trasmettere un bit, su una rete a 10 Mbps occorrono 0.1 microsecondi (μs)

- un altro modo di vedere la banda:
  - immaginare un secondo di tempo come un segmento: la banda è "quanti bits possono essere contenuti in quel segmento di tempo"
  - ogni bit un impulso con una ampiezza



bits trasmessi a
1 Mbps (ogni bit ha "ampiezza" 1μs);

bits trasmessi a
10 Mbps (ogni bit ha "ampiezza" 0.5μs);

DELAY TOTALE

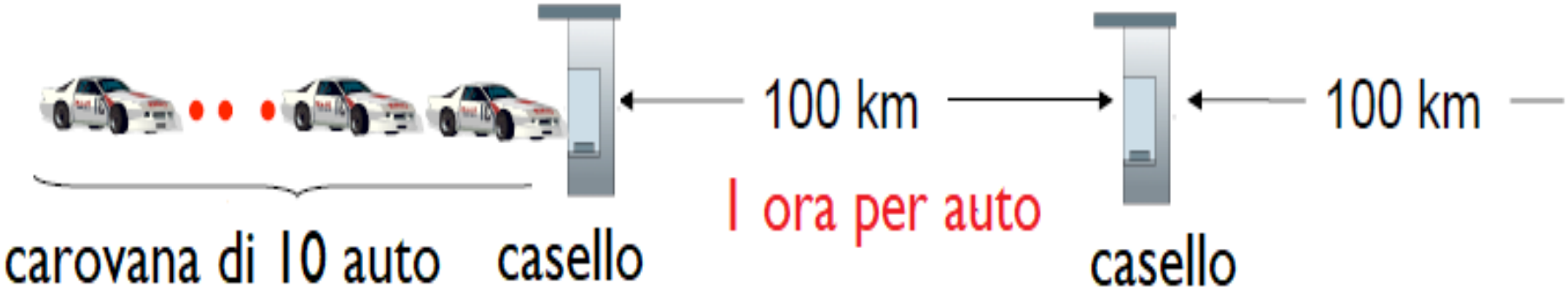
- Per calcolare il delay totale (talvolta indicato come latenza) su un link fisico:

  Latency=Transmission delay+Propagation delay

  - **Propagation delay = Distance/Speed of light**
    - limitato dalla velocità della luce.
    - velocità della luce diversa in diversi mezzi fisici (fibra ottica, cavo di rame,...)
  - **Trasmission delay: Packet Size/Bandwidth**
    - tempo richiesto per trasmettere una unità di dati.

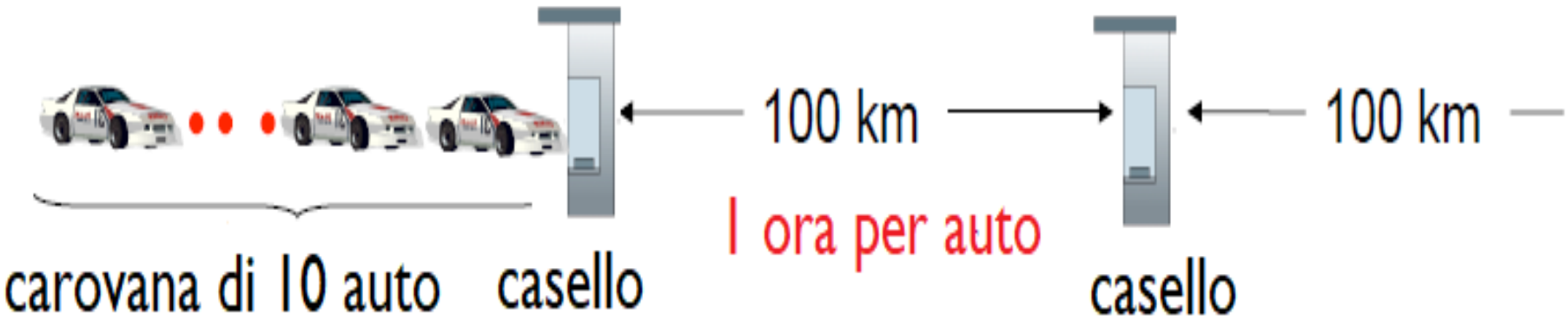
- Latency in packed switched networks, occorre considerare
  - pacchetto può attraversare segmenti di rete con caratteristiche diverse
  - ritardi dovuto alle code nei nodi intermedi

PROPAGATION E TRASMISSION DELAY :UNA METAFORA



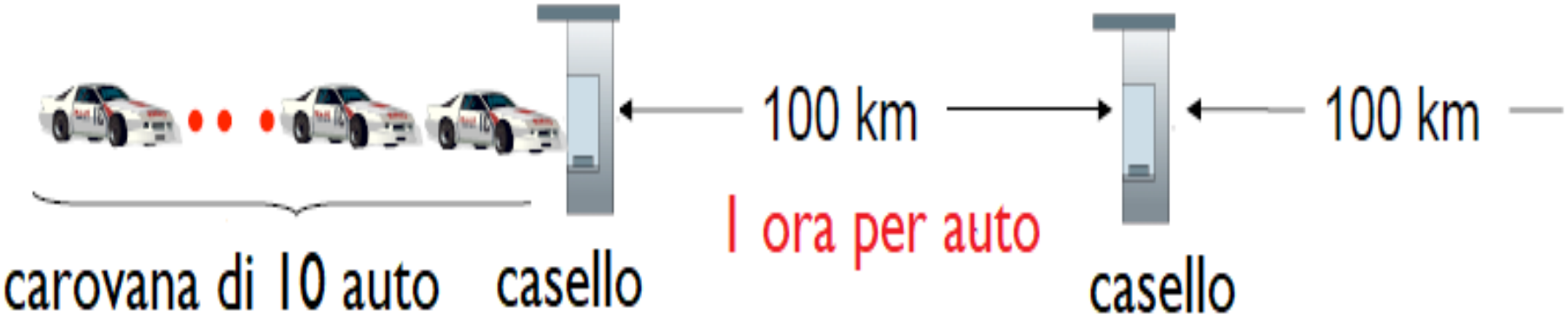
- velocità delle auto: 100 km/h (tempo di propagazione dell'auto tra un casello e l'altro)
- tempo di servizio di un casello per una singola auto: 12 secondi (tempo di trasmissione)
- auto~bit; carovana ~ pacchetto;
- quanto tempo occorre prima che tutta la carovana raggiunga il secondo casello? (Latency)

PROPAGATION E TRASMISSION DELAY :UNA METAFORA



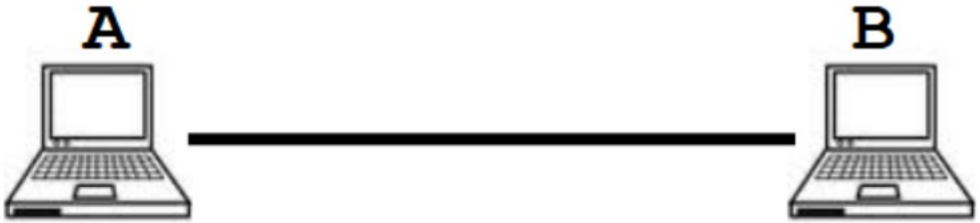
- tempo per "far passare" l'intera carovana di auto attraverso il primo casello ed immetterla nell'autostrada: 12 * 10 =120 secondi (banda del casello)
- tempo che impiega l'ultima auto per raggiungere il secondo casello a 100 km/h= 1h (Propagaton delay)
- tempo totale (latency) =62 minuti

PROPAGATION E TRASMISSION DELAY :UNA METAFORA

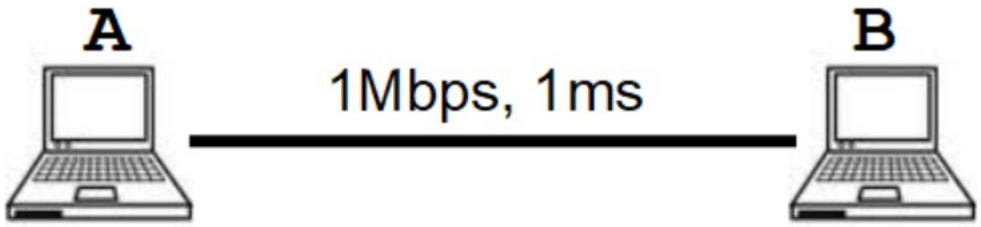


- supponiamo che le auto viaggino a 1000 km/ora (!!), e che il casello richieda un minuto a servire una auto
- ci sono auto che arrivano al secondo casello prima che tutte le auto siano state servite al primo casello?
  - Sì, dopo 7 minuti, la prima auto arriva al secondo casello, mentre tre auto sono ancora al primo casello

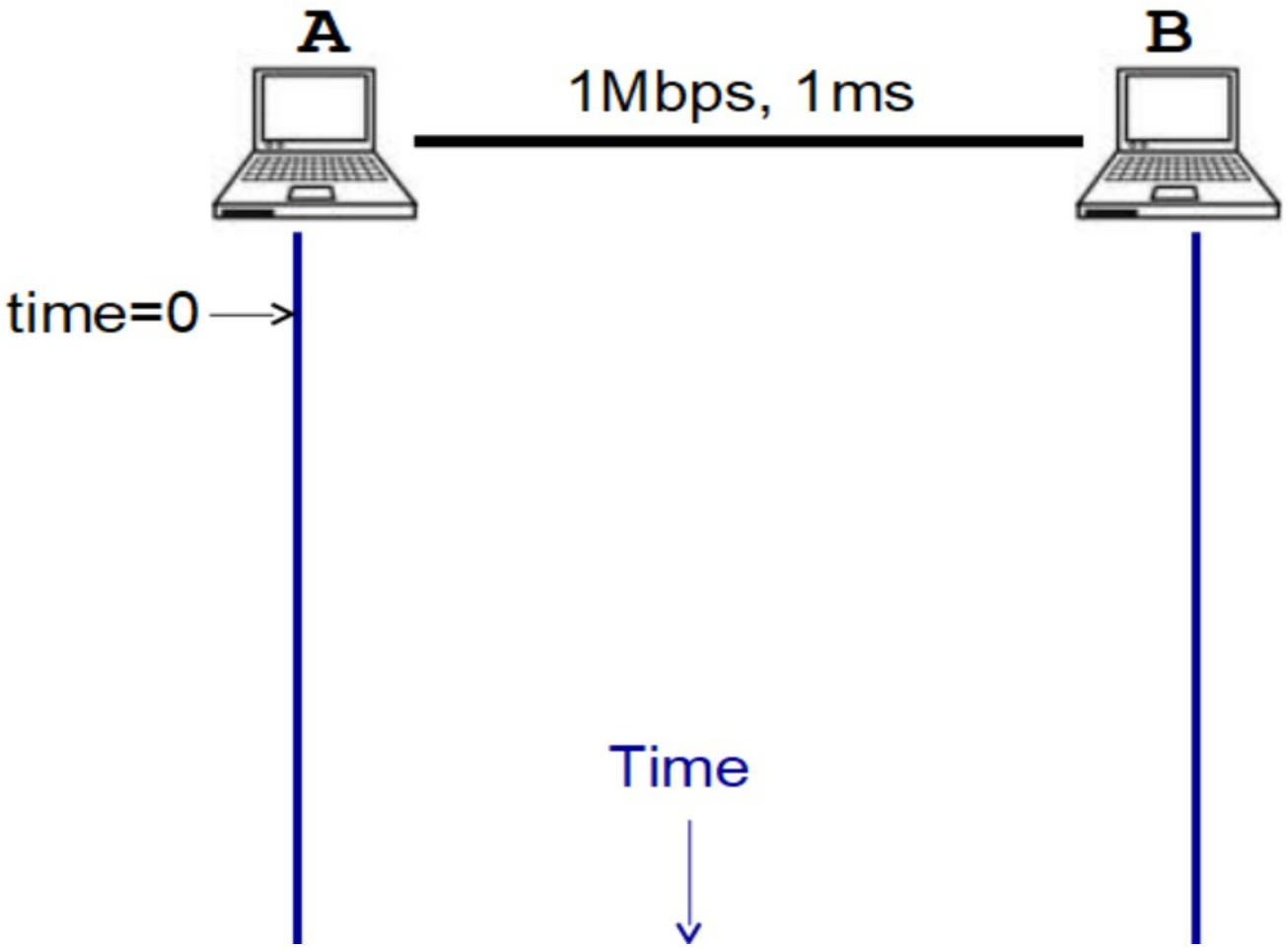
Quanto tempo per spedire un pacchetto di 100Bytes *da A a B*?



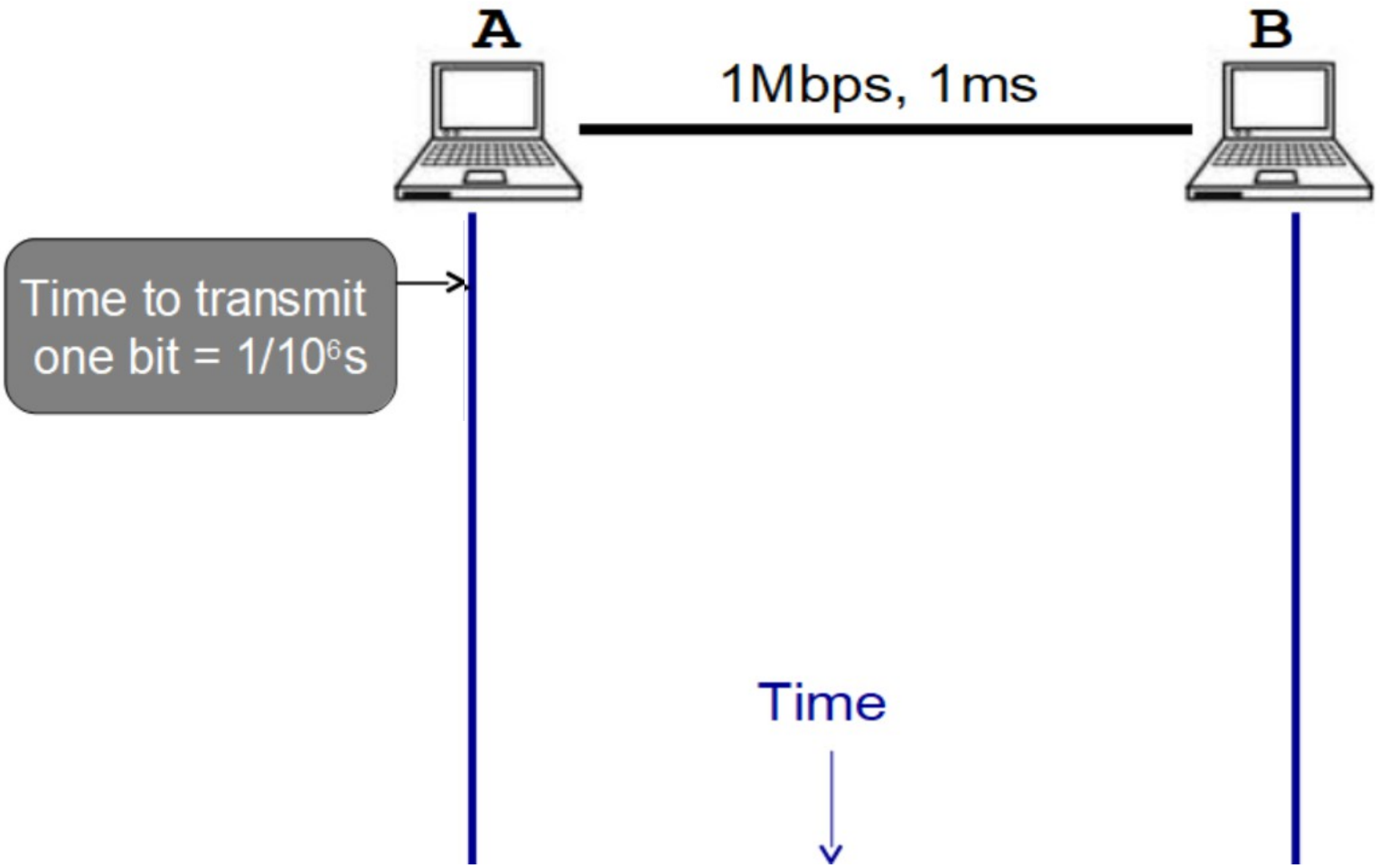
Quanto tempo per spedire un pacchetto di 100Bytes *da A a B?*



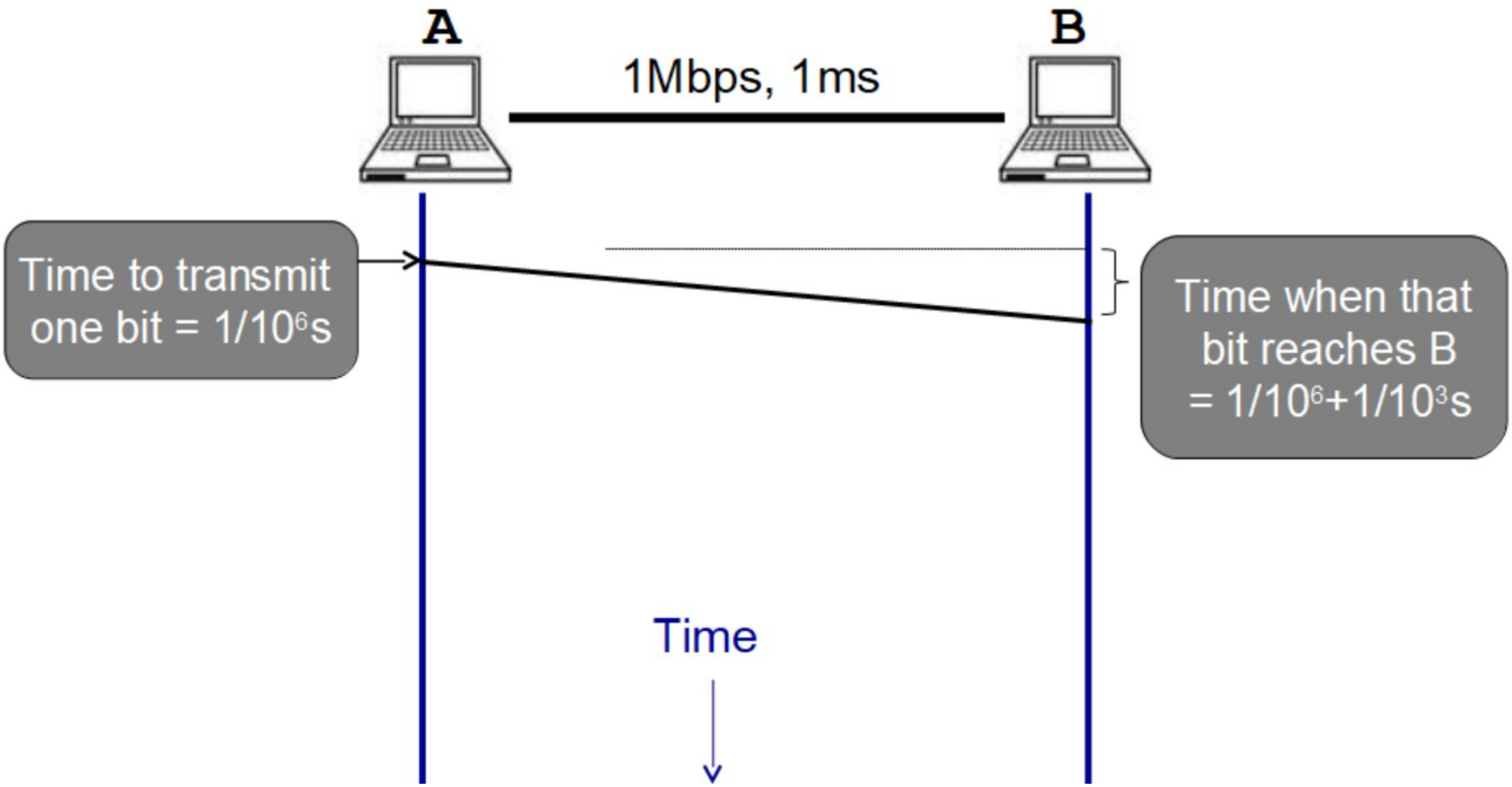
Quanto tempo per spedire un pacchetto di 100Bytes *da A a B*?



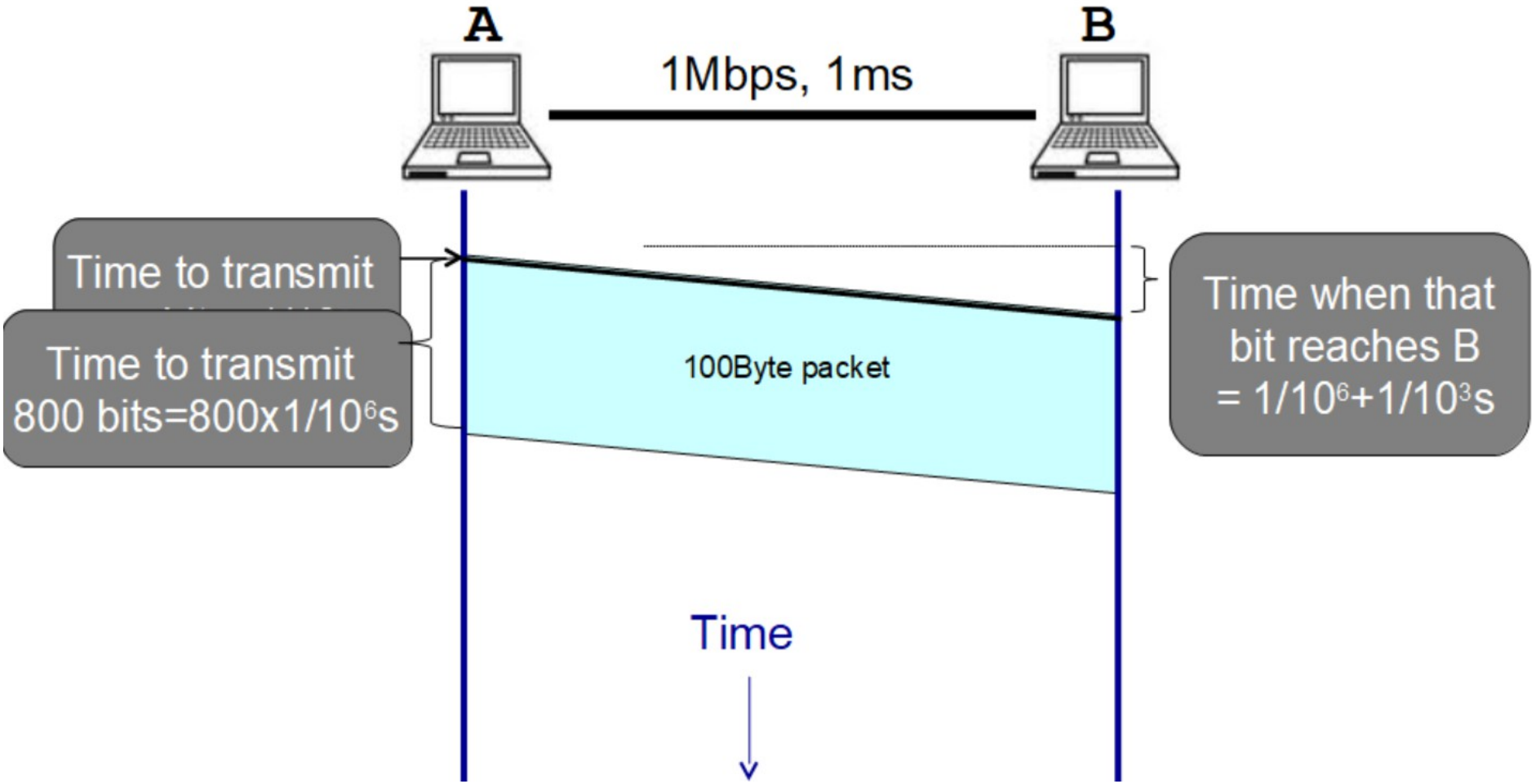
Quanto tempo per spedire un pacchetto di 100Bytes *da A a B*?



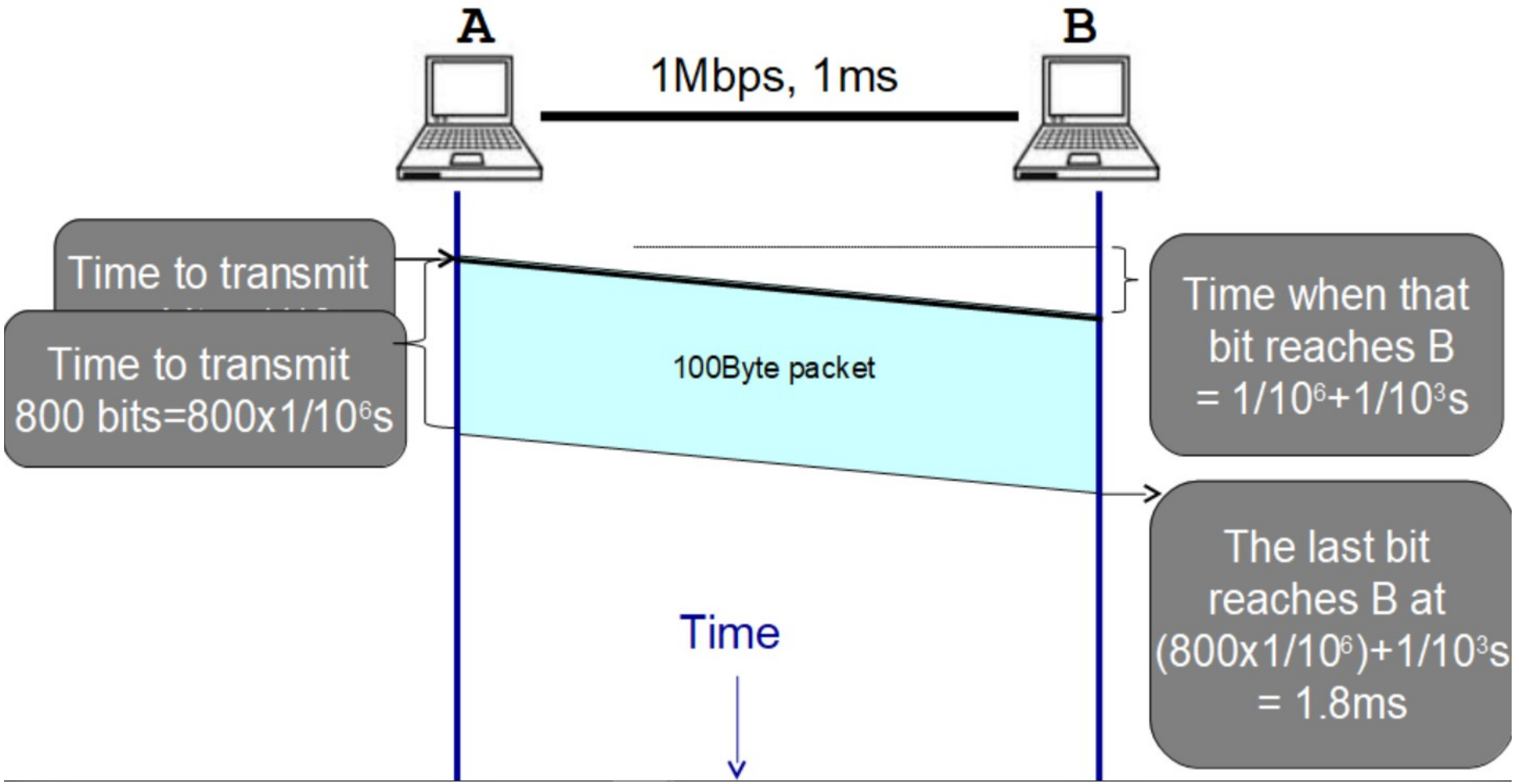
Quanto tempo per spedire un pacchetto di 100Bytes *da A a B*?



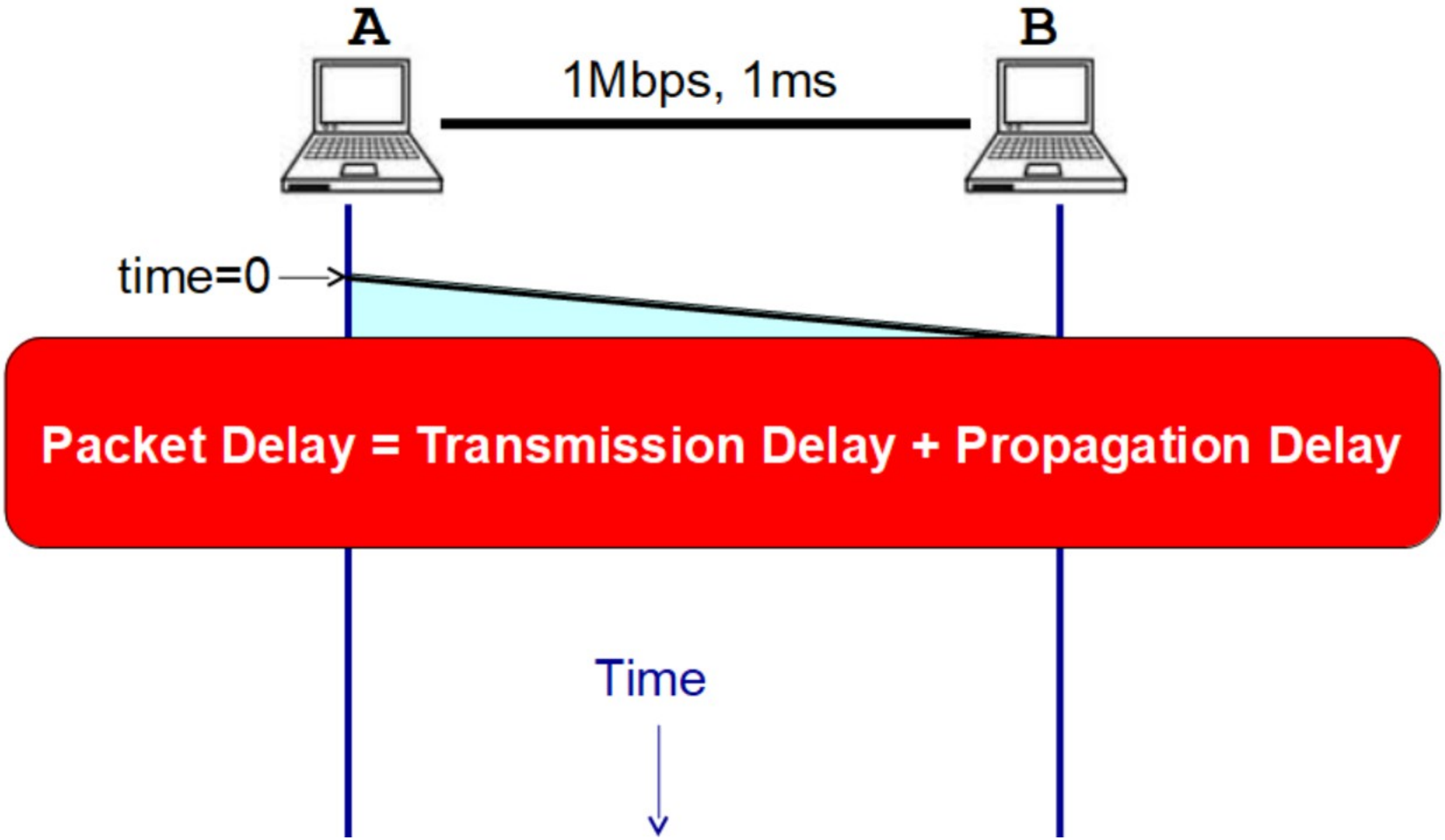
Quanto tempo per spedire un pacchetto di 100Bytes *da A a B*?



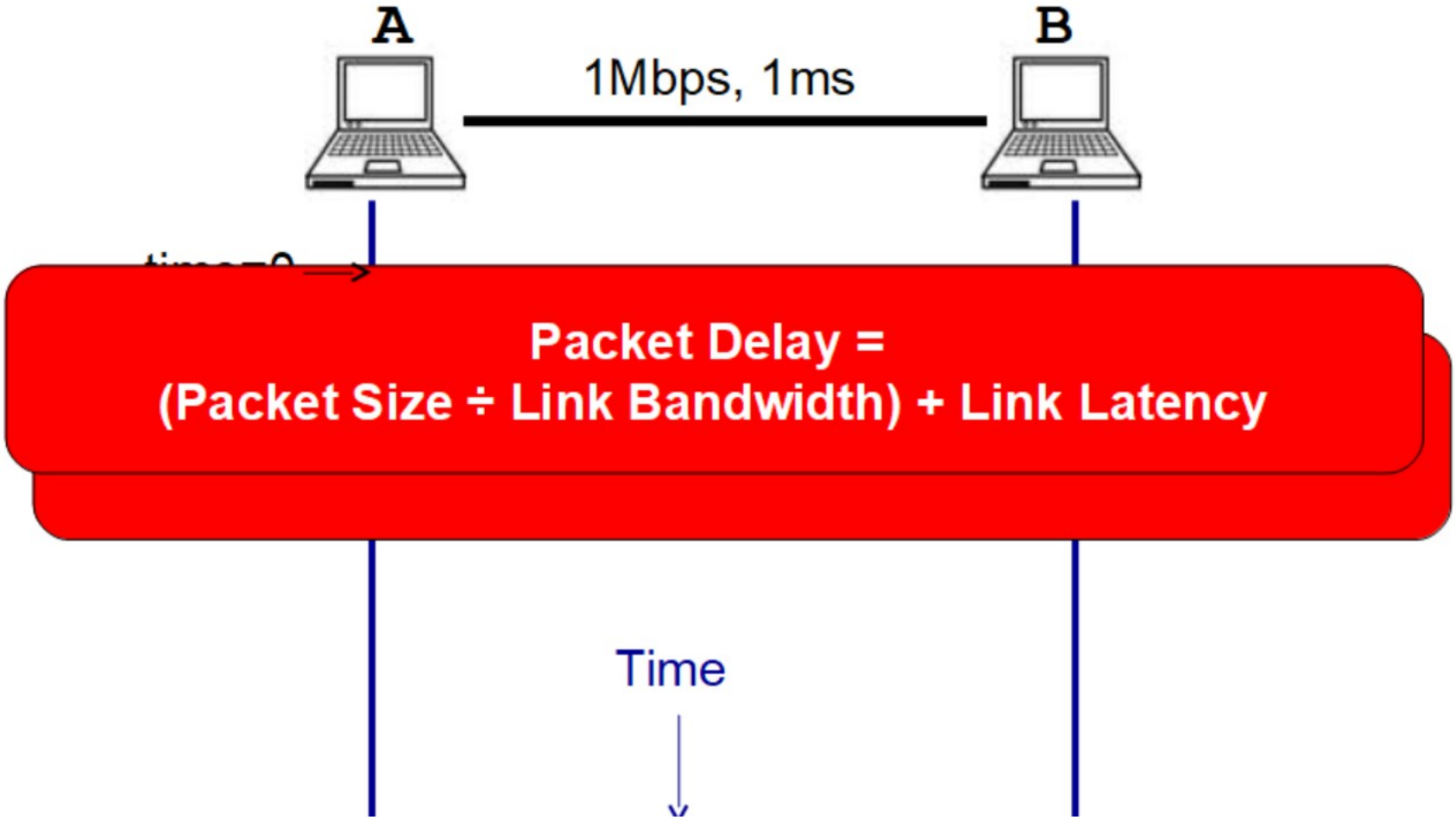
Quanto tempo per spedire un pacchetto di 100Bytes *da A a B*?



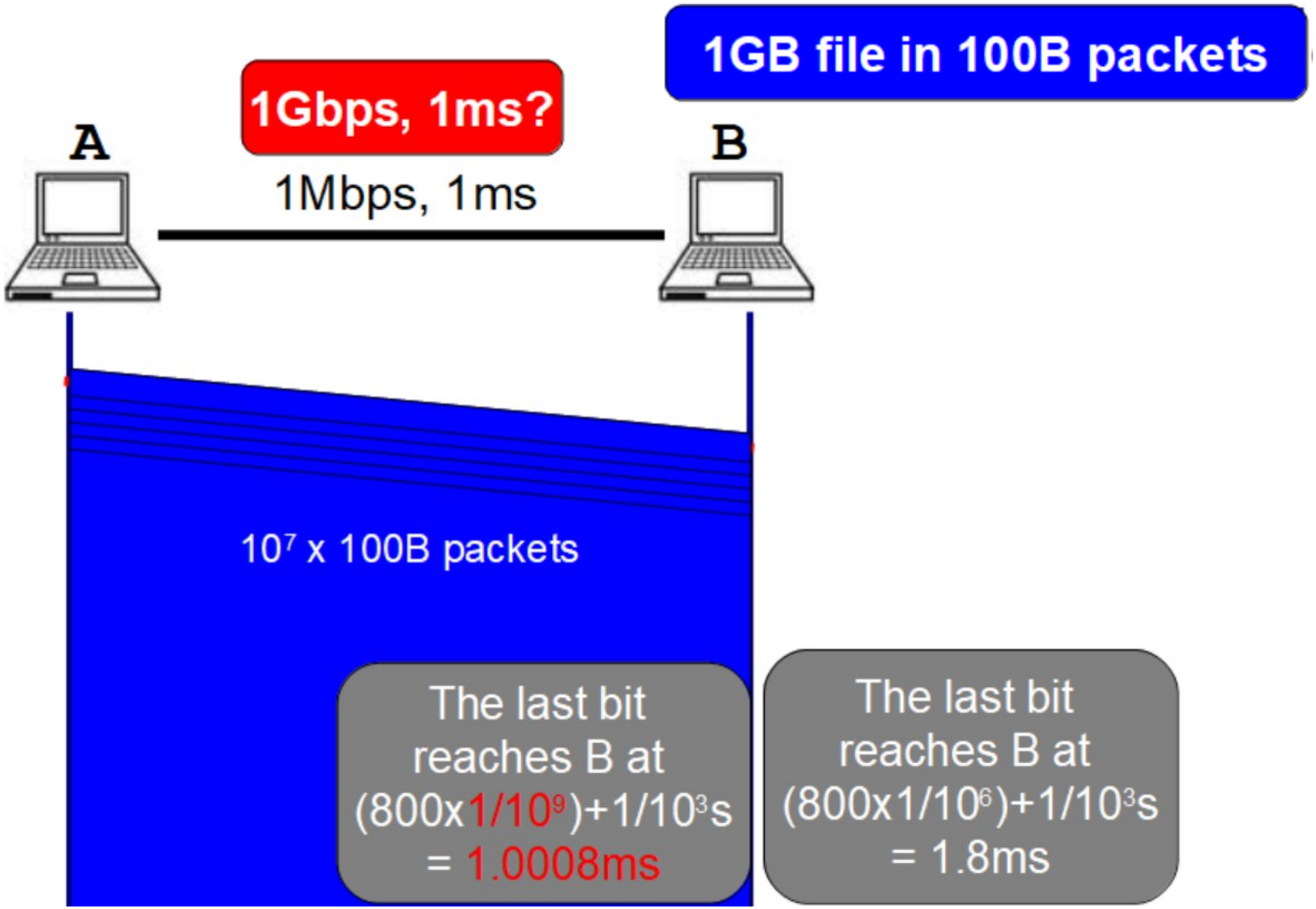
Quanto tempo per spedire un pacchetto di 100Bytes *da A a B*?



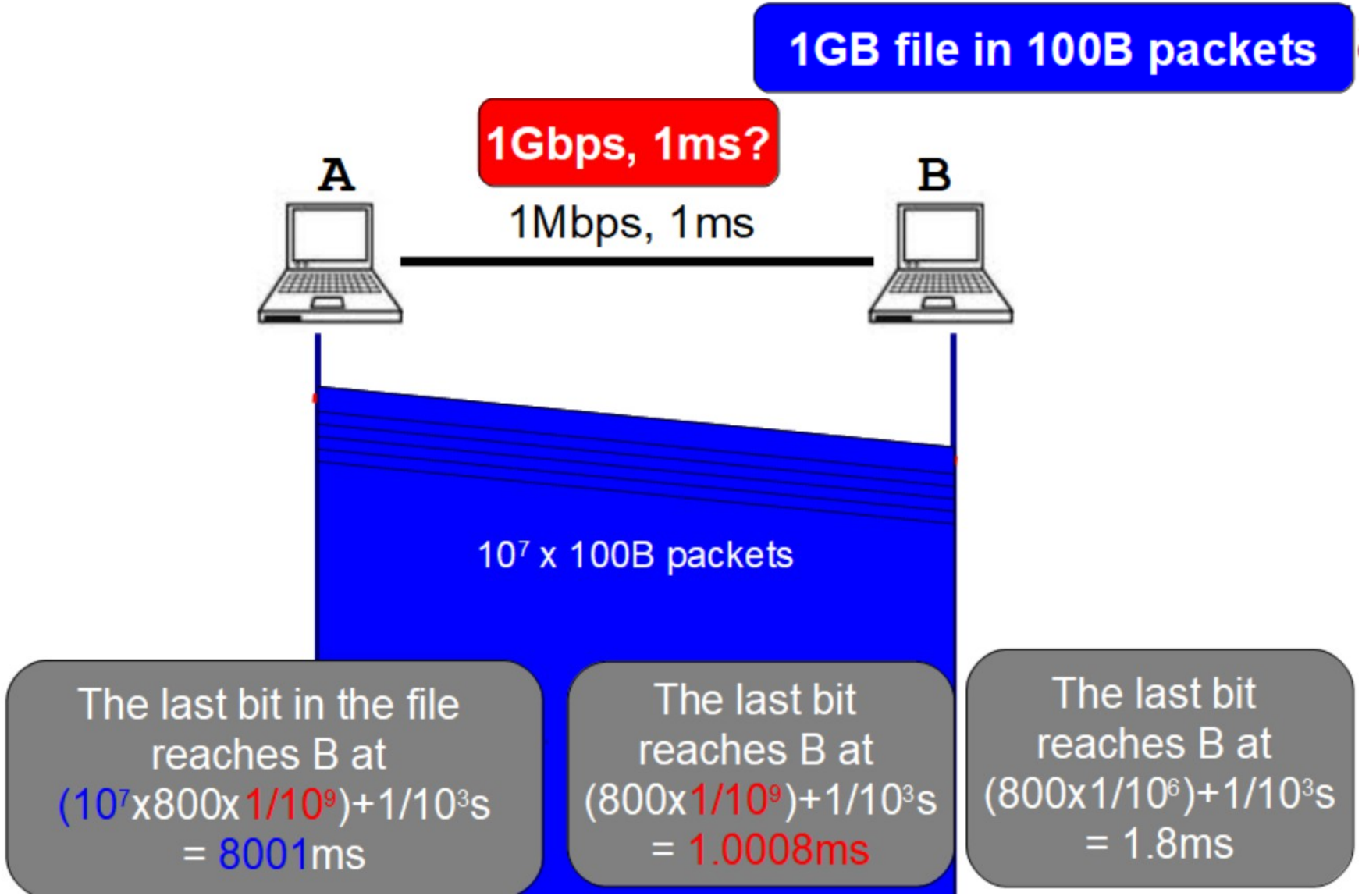
Quanto tempo per spedire un pacchetto di 100Bytes *da A a B*?



Quanto tempo per spedire 1GB file?



Quanto tempo per spedire 1GB file?



CARATTERISTICHE DI ALCUNI COLLEGAMENTI

Table 1.1 Sample Delay × Bandwidth Products

Link type	Bandwidth (typical)	One-way distance (typical)	Round-trip delay	RTT × Bandwidth
Dial-up	56 kbps	10 km	87 μs	5 bits
Wireless LAN	54 Mbps	50 m	0.33 μs	18 bits
Satellite	45 Mbps	35,000 km	230 ms	10 Mb
Cross-country fiber	10 Gbps	4,000 km	40 ms	400 Mb

APPLICAZIONI LATENCY BOUND

- L'importanza relativa di latenza e banda dipende dalla applicazione che si sta eseguendo sulla rete

- Esempio di applicazione latency bound:
  - un client invia al server un messaggio di 1 byte e riceve risposta di 1 byte, supponiamo che Il tempo per preparare la risposta sia trascurabile
    - canale transcontinentale con un RTT di 100 ms
      - propagation delay: 8 µs
    - trasmissione all'interno di uno stesso edificio con RTT di 1 ms.
      - propagation delay 0.08 µs
    - una banda 1 Mbps o 100 Mbps impatta poco sulla latenza totale. in entrambe i casi iltempo di trasmissione è trascurabile rispetto al tempo di propagazione.

- Esempi:
  - Trasmissione voce: richiede un numero contenuto di bytes, mentre è importante non perdere l'interattività della conversazione.
    - se domando qualcosa non voglio aspettare troppo per sentire la risposta

APPLICAZIONI BANDWIDTH BOUND

- l'applicazione richiede il download di una immagine di grosse dimensioni (25Mb)
- in questo caso la banda domina la latenza
- consideriamo una banda di 10 Mbps, il tempo di trasmissione è

  $25 \times 10^6 \times 8$ bits $\div 10 \times 10^6$ Mbps = 20 seconds

  sono necessari 20 secondi per la trasmissione dell'immagine, ovvero "immettere tutti i byte dell'immagine" sul canale di comunicazione
- propagation delay:
  - 1ms: tempo di download: 20.001 secondi
  - 100 ms: tempo di download 20.1 secondi
- la latenza risulta irrilevante per questo tipo di applicazione
- non è importante che il singolo pacchetto che compone l'immagine sia ricevuto in un tempo breve, quanto che tutta l'immagine sia ricevuta in tempo ragionevole.

APPLICAZIONI BANDWIDTH E LATENCY BOUND

- Per alcune applicazioni è importante sia la latenza che la band
- Video Conferenza:
  - Propagation delay: Il frame video (o audio) che un partecipante vede (o sente) non dovrebbe essere troppo in ritardo rispetto al video(o all'audio) delle altre persone che partecipano alla conferenza
  - Trasmission Delay: per avere un video di buona qualità, occorre la ossibilità di inviare/ricevere molti byte per secondo

UN GRAFICO RIASSUNTIVO

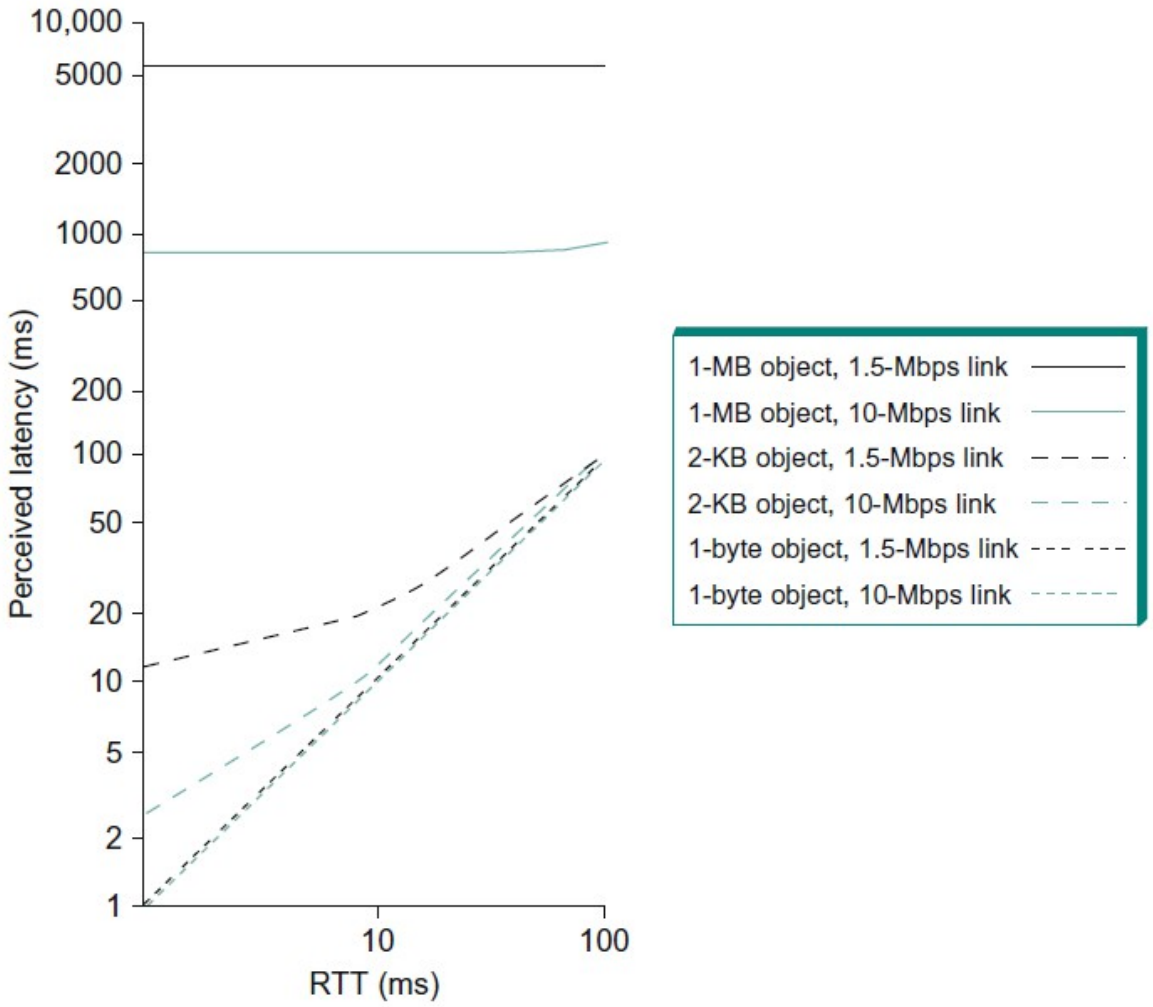


Grafico in scala logaritmica

PERFORMANCE LAYERS

- in precedenza abbiamo considerato la banda a livello di link
- Analogo concetto a livelli diversi, ma si parla di throughput
- link con capacità effettiva di trasmissione di 10Gbit/s
- data link layer:
  - aggiunge un MAC header al messaggio (18 bytes), se spedisco un pacchetto di un byte, il throughput diventa $\frac{1}{1+18}$ della banda del link fisico
- TCP layer: i bytes devono essere spediti al livello applicazione in modo ordinato:
  - banda fisica di 1000 bytes al secondo,
  - il secondo ed il terzo byte arrivano fuori ordine
  - Il terzo byte viene bufferizzato I attesa del secondo
  - Il throughput a livello TCP, inteso come numero di bytes ricevuti al secondo, si riduce rispetto alla banda fisica

PACKET E CIRCUIT SWITCHING: PERFORMANCE

- Circuit switching: banda e latenza fissate al momento della definizione della connessione

- Packet switching:
  - La performance dipende da quante applicazioni stanno condividendo le risorse della rete, quando trasmettono dati, quali cammini vengono scelti per i dati
  - performance migliori nel caso medio, peggiori nel caso pessimo
  - applicazione che spedisce dati a 9 Mbit/s su un link da 10 Mbit/s link, lascia solamente una capacità di 1 Mbit/s per le altre applicazioni
    - "perfomance determinism" non garantito, solo best effort
  - offre una migliore utilizzazione delle risorse