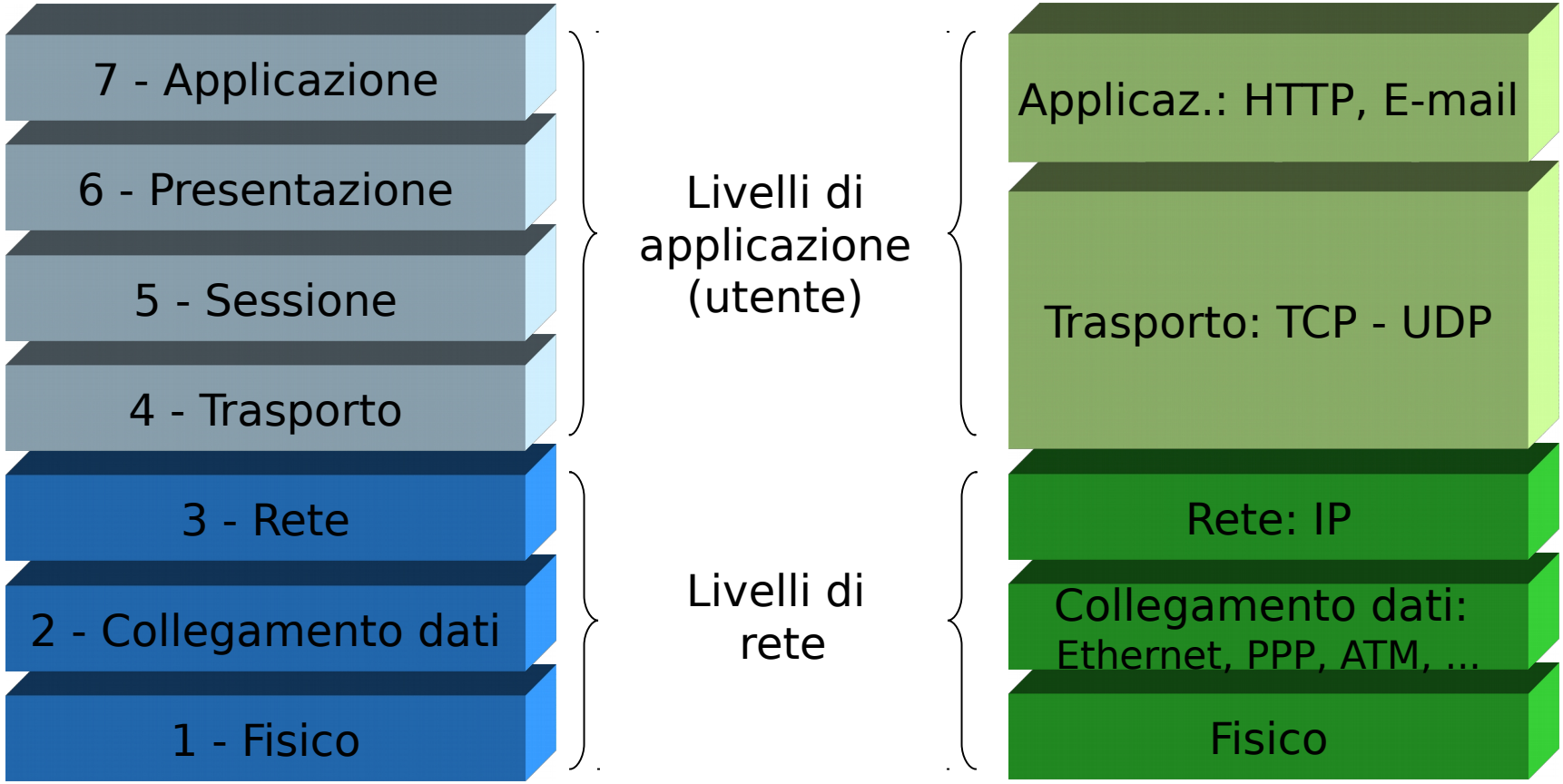
Concetti fondamentali delle reti

Damiano Carra, Davide Quaglia

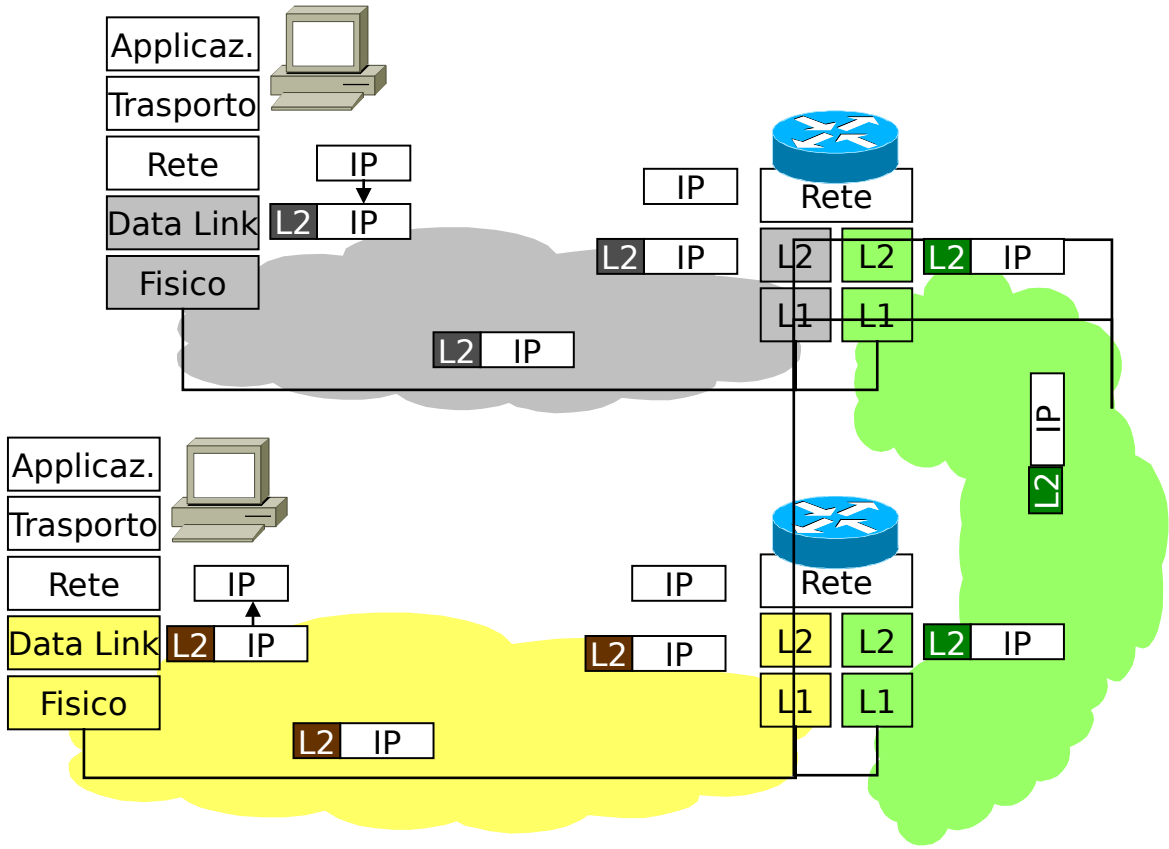
Università degli Studi di Verona
Dipartimento di Informatica

Stack OSI... ...e Stack TCP/IP

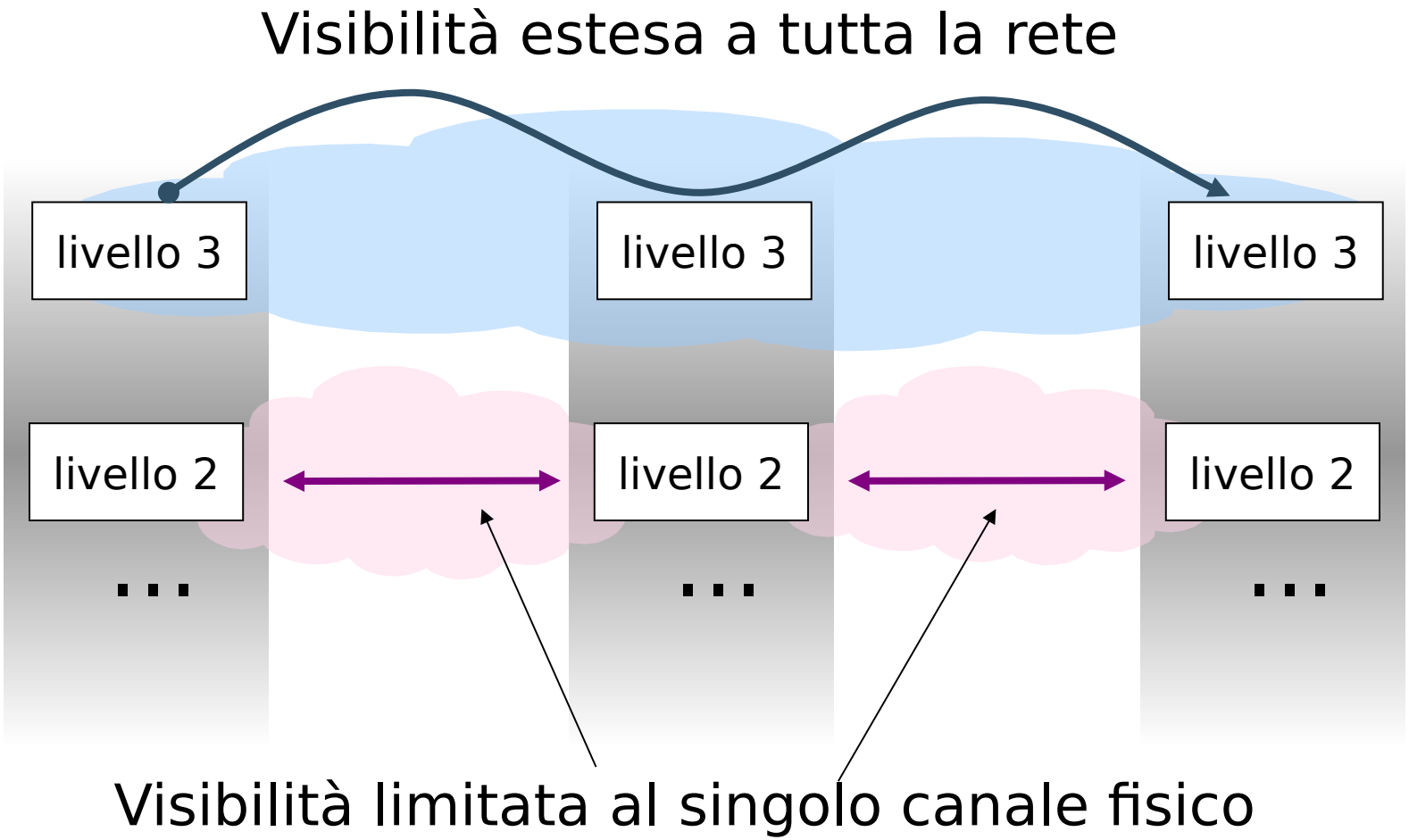


Visione generale

- Trasporto dei segmenti dall'host sorgente all'host destinazione
- Lato sorgente, i segmenti vengono incapsulati in ***datagrammi***
- Lato destinazione, i datagrammi vengono consegnati al livello di trasporto
- Il livello di rete e' presente in ogni end-host e ogni apparato intermedio (router)
- I router esaminano i campi dell'header presenti nei datagrammi IP



Visibilità della rete del livello 2 e 3



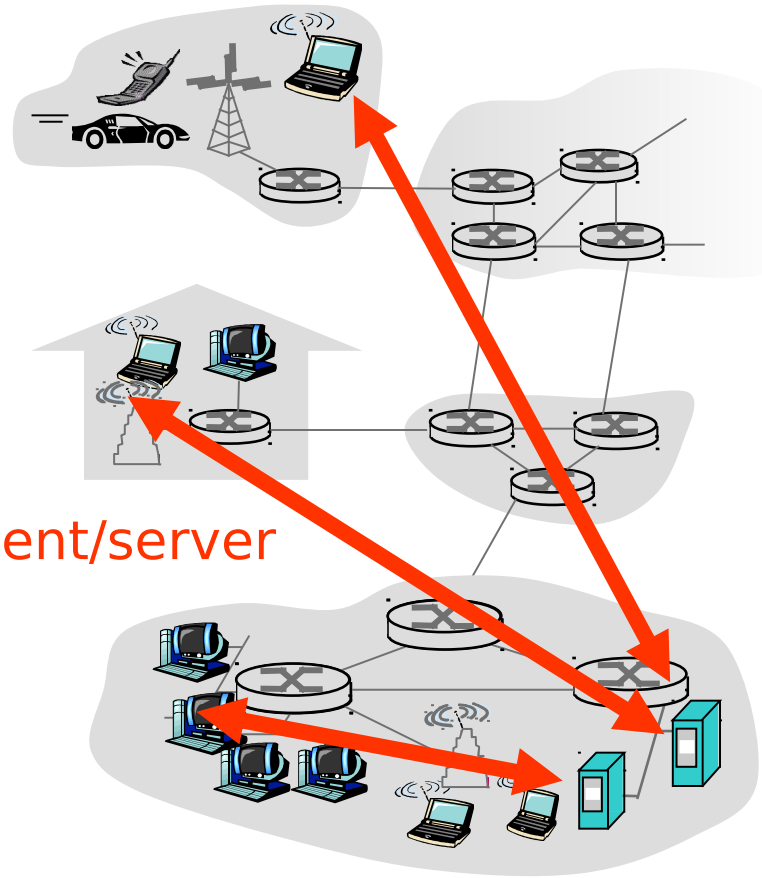
Alcune diffuse applicazioni di rete

- Posta elettronica
- Web
- Messaggistica istantanea
- Autenticazione in un calcolatore remoto
- Condivisione di file P2P
- Giochi multiutente via rete
- Streaming di video-clip memorizzati
- Telefonia via Internet
- Videoconferenza in tempo reale
- Grid computing

Architetture delle applicazioni di rete

- Client-server
- Peer-to-peer (P2P)
- Architetture ibride (client-server e P2P)

Architettura client-server



server:

- host sempre attivo
- indirizzo IP fisso
- server farm per creare un potente server virtuale

client:

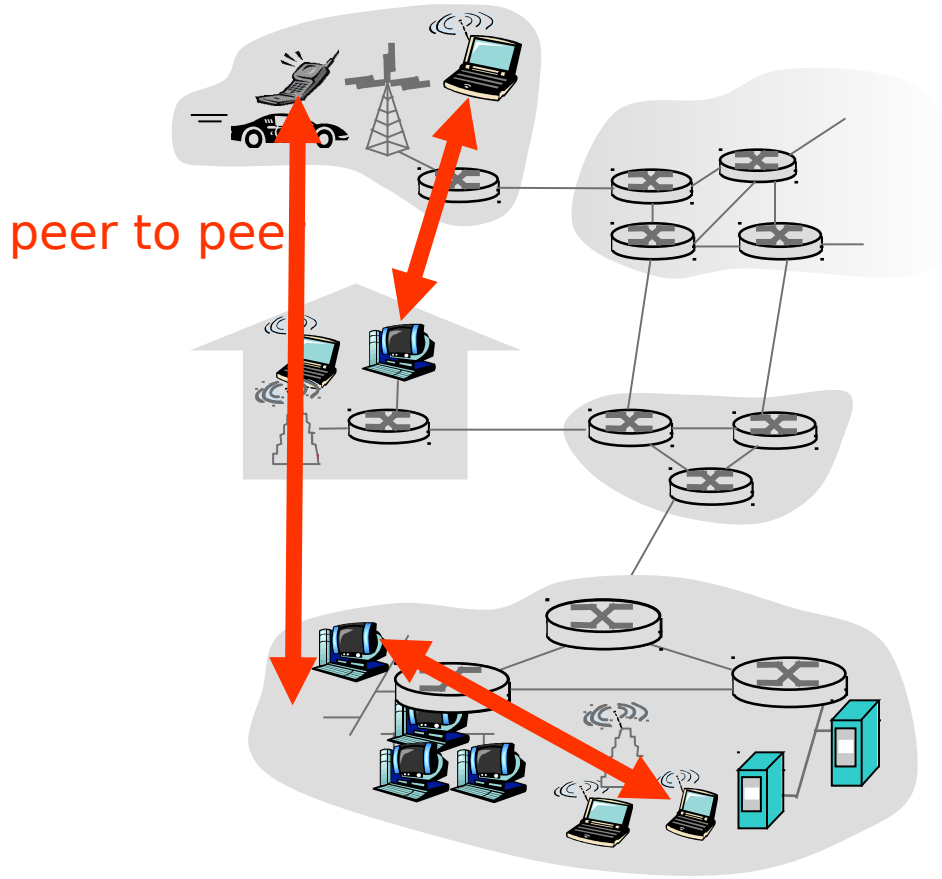
- comunica con il server
- può contattare il server in qualunque momento
- può avere indirizzi IP dinamici
- non comunica direttamente con gli altri client

Architettura P2P pura

- non c’è un server sempre attivo
- coppie arbitrarie di host (peer) comunicano direttamente tra loro
- i peer non devono necessariamente essere sempre attivi, e cambiano indirizzo IP

Facilmente scalabile

Difficile da gestire



Ibridi (client-server e P2P)

Skype

- Applicazione P2P di Voice over IP
- Server centralizzato: ricerca indirizzi della parte remota
- Connessione client-client: diretta (non attraverso il server)

Messaggistica istantanea

- La chat tra due utenti è del tipo P2P
- Individuazione della presenza/location centralizzata:
  - l'utente registra il suo indirizzo IP sul server centrale quando è disponibile online
  - l'utente contatta il server centrale per conoscere gli indirizzi IP dei suoi amici

Protocollo a livello di applicazione

- Tipi di messaggi scambiati, ad esempio messaggi di richiesta e di risposta
- Sintassi dei tipi di messaggio: quali sono i campi nel messaggio e come sono descritti
- Semantica dei campi, ovvero significato delle informazioni nei campi
- Regole per determinare quando e come un processo invia e risponde ai messaggi

Protocolli di pubblico dominio:

- Definiti nelle RFC
- Consentono l'interoperabilità
- Ad esempio, HTTP, SMTP

Protocolli proprietari:

- Ad esempio, Skype

Quale servizio di trasporto richiede un'applicazione?

Affidabilità

- alcune applicazioni (ad esempio, audio) possono tollerare qualche perdita di pacchetto
- altre applicazioni (ad esempio, trasferimento di file, telnet) richiedono un trasferimento dati affidabile al 100%

Ritardi

- alcune applicazioni (ad esempio, telefonia Internet, giochi interattivi) per essere "realistiche" richiedono bassi ritardi o bassa variazione del ritardo

Throughput

- alcune applicazioni (ad esempio, streaming video) per essere "efficaci" richiedono almeno una certa capacità del canale
- altre applicazioni (ad es. trasferimento file) utilizzano la capacità del canale disponibile

Sicurezza

- Cifratura, integrità dei dati, autenticazione

Requisiti del servizio di trasporto di alcune applicazioni comuni

Applicazione	Sensibilità alla perdita di dati	Throughput	Sensibilità al ritardo
Trasferimento file	Sì	Variabile	No
Posta elettronica	Sì	Variabile	No
Documenti Web	Sì	Variabile	No
Audio/video in tempo reale	No	Audio: da 5 Kbps a 1 Mbps Video: da 10 Kbps a 5 Mbps	Sì, centinaia di ms
Audio/video memorizzati	No	Come sopra	Sì, pochi secondi
Giochi interattivi	No	Fino a pochi Kbps	Sì, centinaia di ms
Messaggistica istantanea	Sì	Variabile	Intermedia

Servizi dei protocolli di trasporto Internet

Servizio di TCP:

- *orientato alla connessione:* è richiesto un setup fra i processi client e server
- *trasporto affidabile* fra i processi d'invio e di ricezione
- *controllo di flusso:* il mittente non vuole sovraccaricare il destinatario
- *controllo della congestione:* limita il processo d'invio quando le rete è sovraccaricata
- *non offre:* controllo dei ritardi, garanzie su una capacità del canale minima, sicurezza

Servizio di UDP:

- trasferimento dati inaffidabile fra i processi d'invio e di ricezione
- non offre: setup della connessione, affidabilità, controllo di flusso, controllo della congestione, controllo del ritardo, capacità del canale minima, sicurezza

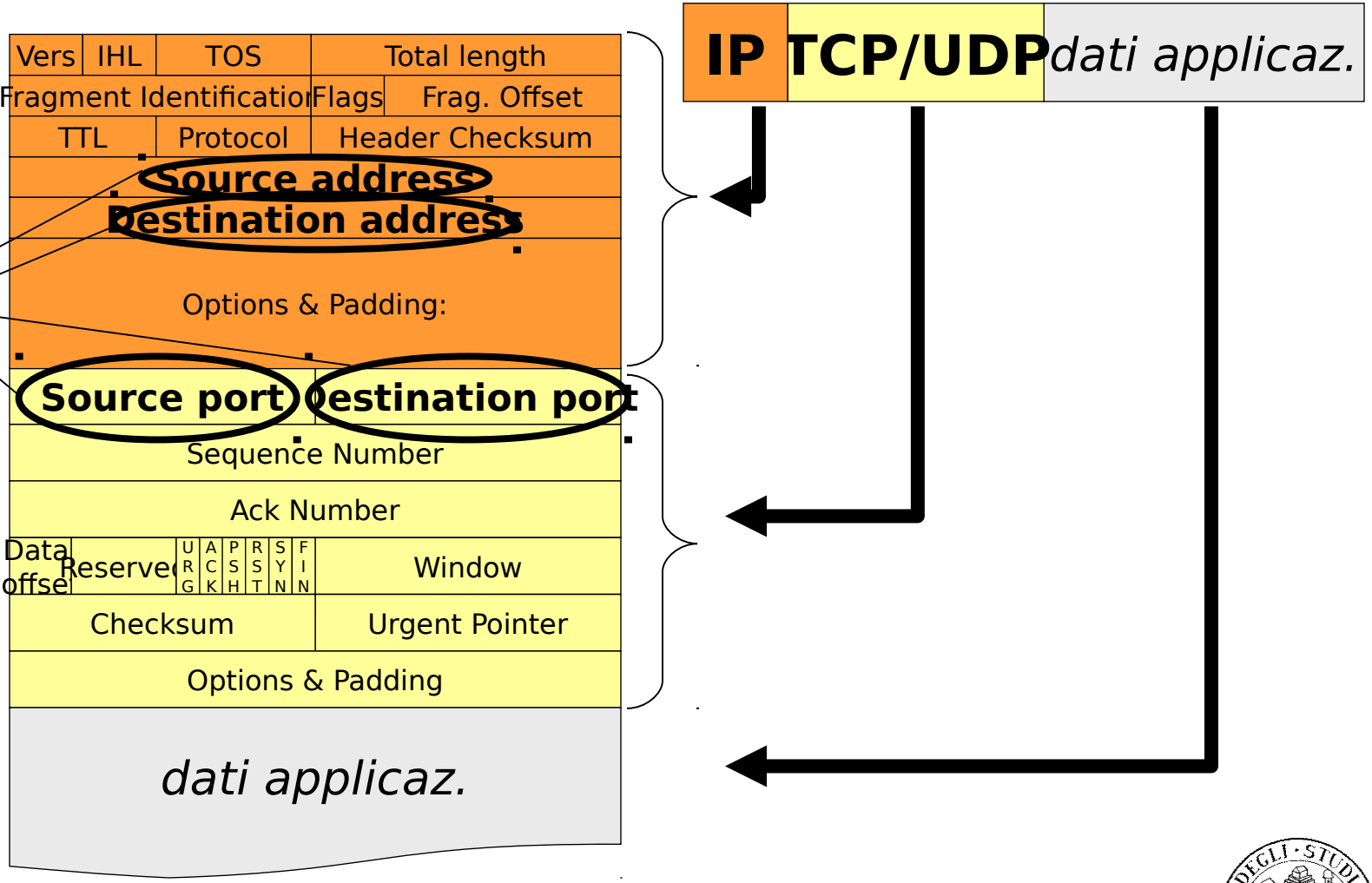
D: perché esiste UDP?

Applicazioni Internet: protocollo a livello applicazione e protocollo di trasporto

Applicazione	Protocollo a livello applicazione	Protocollo di trasporto sottostante
Posta elettronica	SMTP [RFC 2821]	TCP
Accesso a terminali remoti	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
Trasferimento file	FTP [RFC 959]	TCP
Multimedia in streaming	HTTP (es. YouTube) RTP [RFC 1889]	TCP o UDP
Telefonia Internet	SIP, RTP, proprietario (es. Skype)	Tipicamente UDP

Il concetto di Socket

SOCKET
tupla che identifica il flusso tra due applicazioni su host anche molto distanti tra loro

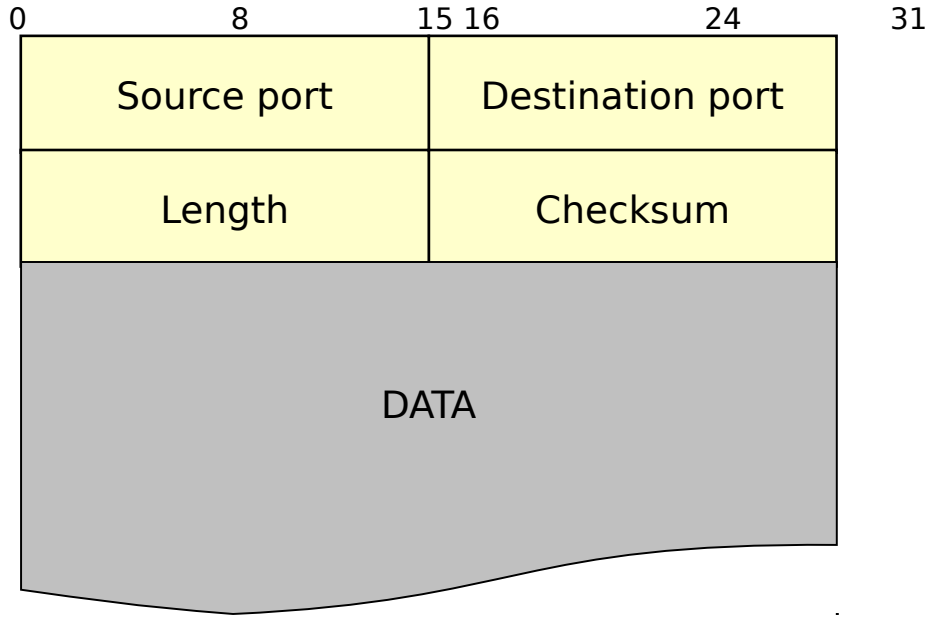


User Datagram Protocol

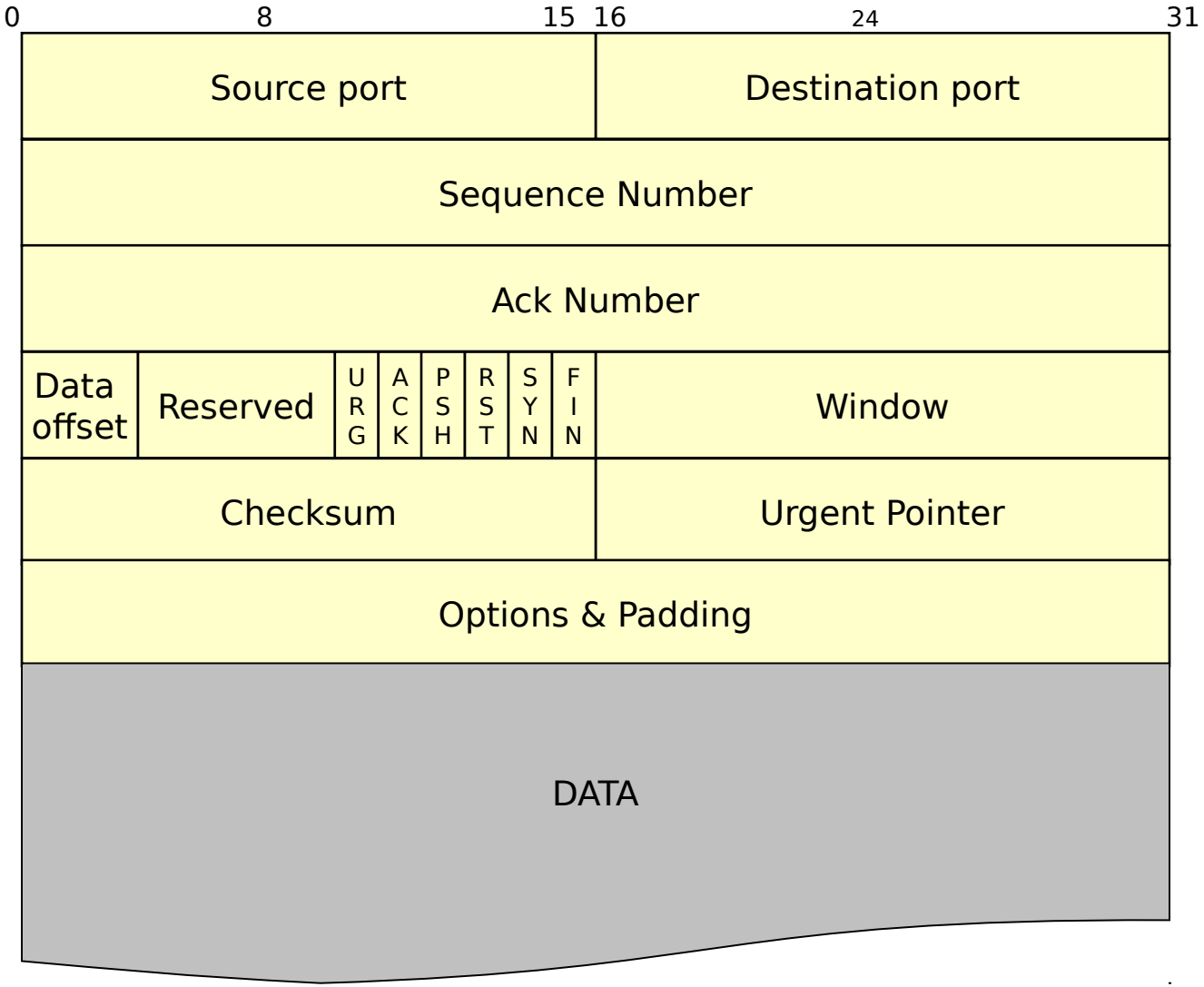
- E’ un protocollo di trasporto connectionless non affidabile
- Svolge solo funzione di indirizzamento delle applicazioni (porte)
- NON gestisce:
  - connessioni
  - controllo di flusso
  - recupero degli errori (solo rilevamento)
  - controllo della congestione
  - riordino dei pacchetti
- E’ compito dei livelli superiori (applicazioni) la verifica della perdita dei messaggi, consegna in ordine, controllo di flusso, ...
- E’ utilizzato per il supporto di transazioni semplici tra applicativi e per le applicazioni multimediali

UDP: Formato dei pacchetti

- Source Port e Destination Port [16 bit]: identificano i processi sorgente e destinazione dei dati
- Length [16 bit]: lunghezza totale (espressa in byte) del datagramma, compreso l'header UDP
- Checksum [16 bit]: campo di controllo che serve per sapere se il datagramma corrente contiene errori nel campo dati



Header TCP


0
8
15
16
24
31
Source port
Destination port
Sequence Number
Ack Number
Data offset
Reserved
URG
ACK
PSH
RST
SYN
FIN
Window
Checksum
Urgent Pointer
Options & Padding
DATA

Significato dei campi

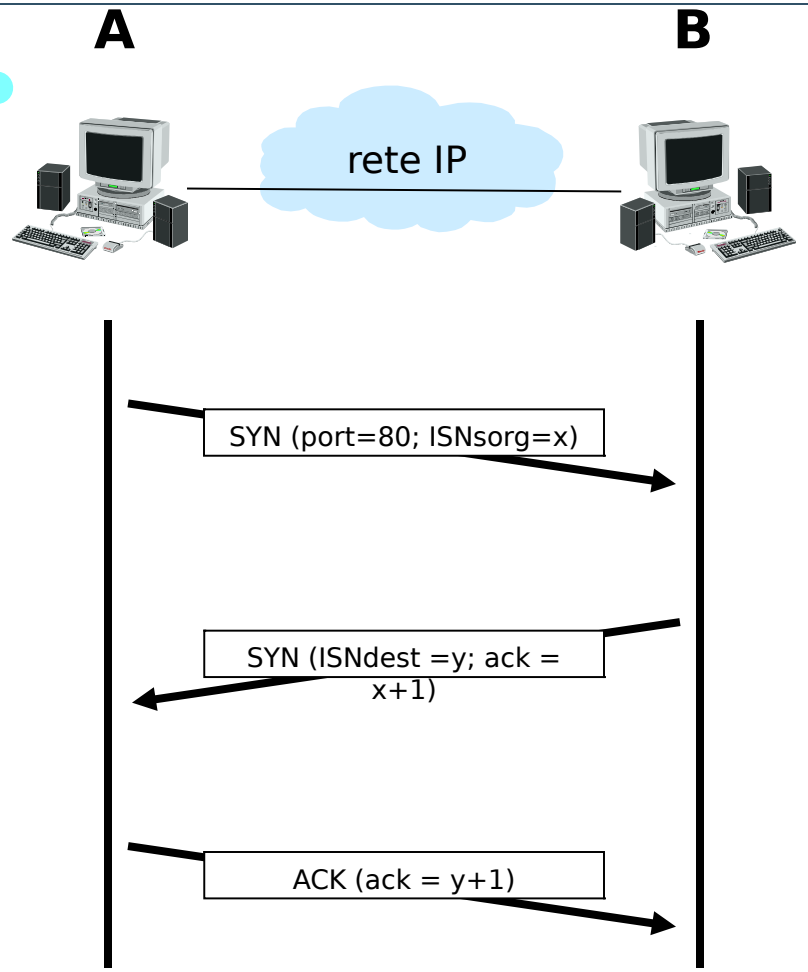
- *Source port – Destination port* [16 bit]: indirizzi della porta sorgente e della porta destinazione
- *Sequence Number* [32 bit]: numero di sequenza del primo byte del payload
- *Acknowledge Number* [32 bit]: numero di sequenza del prossimo byte che si intende ricevere (ha validità se il segmento è un ACK)
- *Offset* [4 bit]: lunghezza dell'header TCP, in multipli di 32 bit
- *Reserved* [6 bit]: riservato per usi futuri
- *Window* [16 bit]: ampiezza della finestra di ricezione (comunicato dalla destinazione alla sorgente)
- *Checksum* [16 bit]: risultato di un calcolo che serve per sapere se il segmento corrente contiene errori nel campo dati
- *Urgent pointer* [16 bit]: indica che il ricevente deve iniziare a leggere il campo dati a partire dal numero di byte specificato. Viene usato se si inviano comandi che danno inizio ad eventi asincroni "urgenti"

Significato dei campi

- *Flag* [ogni flag è lunga 1 bit]:
  - URG: vale uno se vi sono dati urgenti; in questo caso il campo urgent pointer ha senso
  - ACK: vale uno se il segmento è un ACK valido; in questo caso l'acknowledge number contiene un numero valido
  - PSH: vale uno quando il trasmettitore intende usare il comando di PUSH;
  - RST: reset; resetta la connessione senza un tear down esplicito
  - SYN: synchronize; usato durante il setup per comunicare i numeri di sequenza iniziale
  - FIN: usato per la chiusura esplicita di una connessione
- *Options and Padding* [lunghezza variabile]: riempimento (fino a multipli di 32 bit) e campi opzionali come ad esempio durante il setup per comunicare il MSS
- *Data*: i dati provenienti dall'applicazione

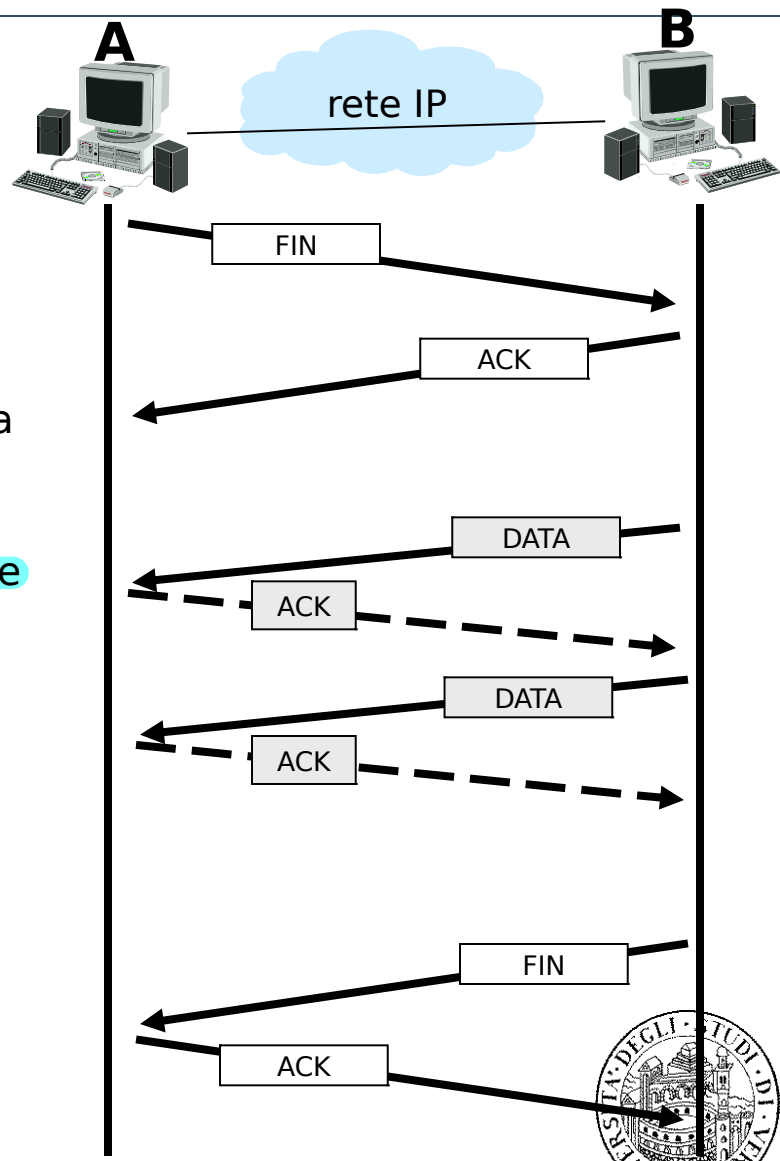
Instaurazione della connessione

- Il TCP è un protocollo ***connection oriented***
  - prima di iniziare a trasferire i dati ci deve essere una connessione tra i due end-system
- L'instaurazione della connessione avviene secondo la procedura detta di "three-way handshake"
  - la stazione che richiede la connessione (A) invia un segmento di SYN
    - parametri specificati: numero di porta dell'applicazione cui si intende accedere e Initial Sequence Number (ISNA)
  - la stazione che riceve la richiesta (B) risponde con un segmento SYN
    - parametri specificati: ISNB e riscontro (ACK) ISNA
  - la stazione A riscontra il segmento SYN della stazione B (invia un ACK alla stazione B)



Terminazione di una connessione

- Poiché la connessione è bidirezionale, la terminazione deve avvenire in entrambe le direzioni
- Procedura di terminazione
  - la stazione che non ha più dati da trasmettere e decide di chiudere la connessione invia un segmento FIN (segmento con il campo FIN posto a 1 e il campo dati vuoto)
  - la stazione che riceve il segmento FIN invia un ACK e indica all'applicazione che la comunicazione è stata chiusa nella direzione entrante
- Se questa procedura avviene solo in una direzione (half close), nell'altra il trasferimento dati può continuare (gli ACK non sono considerati come traffico originato, ma come risposta al traffico)
  - per chiudere completamente la connessione, la procedura di half close deve avvenire anche nell'altra direzione



Stima del Round Trip Time

$$SRTT_{attuale} = (\alpha * SRTT_{precedente}) + ((1-\alpha) * RTT_{istantaneo})$$

$RTT_{istantaneo}$ → misura di RTT sull'ultimo segmento

$SRTT_{precedente}$ → stima precedente del valore medio di RTT

$SRTT_{attuale}$ → stima attuale del valore medio di RTT

α → coefficiente di peso $(0 < \alpha < 1)$

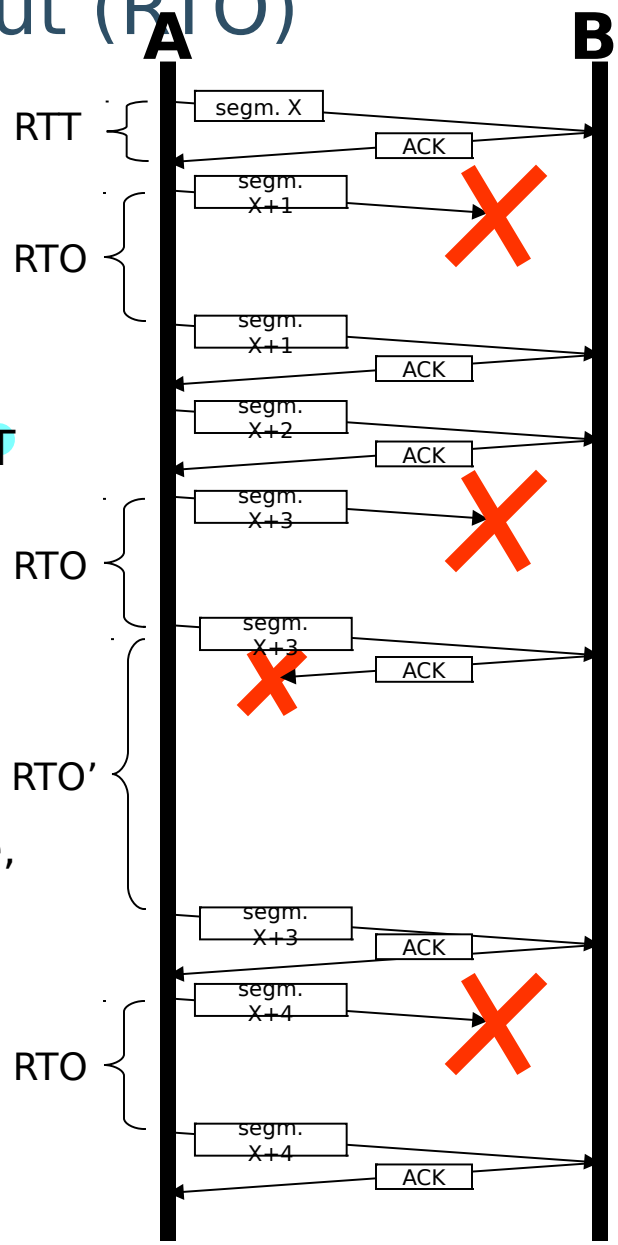
- Poiché RTT può variare anche molto in base alle condizioni della rete, il valore di RTT (SRTT, Smoothed RTT) utilizzato per il calcolo di RTO risulta una stima del valor medio di RTT sperimentato dai diversi segmenti
- Il parametro α è regolabile e, a seconda dei valori assunti, rende il peso della misura di RTT istantaneo più o meno incisivo
  - se α → 1 il SRTT stimato risulta abbastanza stabile e non viene influenzato da singoli segmenti che sperimentano RTT molto diversi
  - se α → 0 il SRTT stimato dipende fortemente dalla misura puntuale dei singoli RTT istantanei
  - tipicamente α = 0.9

Stima del Retransmission Time Out (RTO)

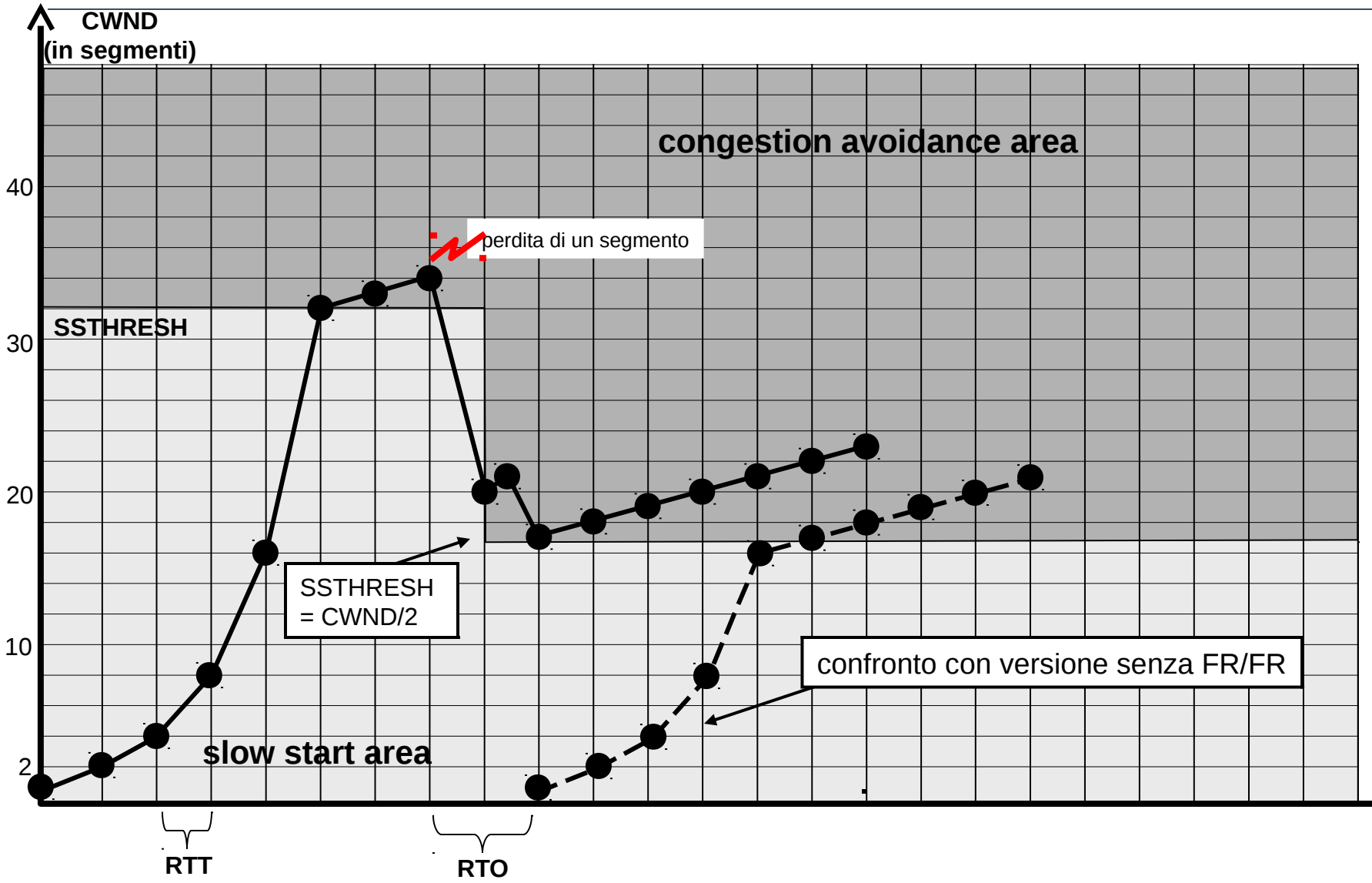
$$RTO = \beta * SRTT$$

$\beta \rightarrow$ delay variance factor $(\text{tipicamente} = 2)$

- La sorgente, dunque, attende fino a 2 volte il RTT medio (SRTT) prima di considerare il segmento perso e ritrasmetterlo
- In caso di ritrasmissione, il RTO per quel segmento viene ricalcolato in base ad un processo di exponential backoff
  - se è scaduto il RTO, probabilmente c'è congestione, quindi meglio aumentare il RTO per quel segmento
  - RTOretransmission = 2*RTO



Fast Retransmit – Fast Recovery: esempio



Funzioni chiave del livello di rete

- Routing
  - Determina il percorso che i pacchetti devono seguire dalla sorgente alla destinazione

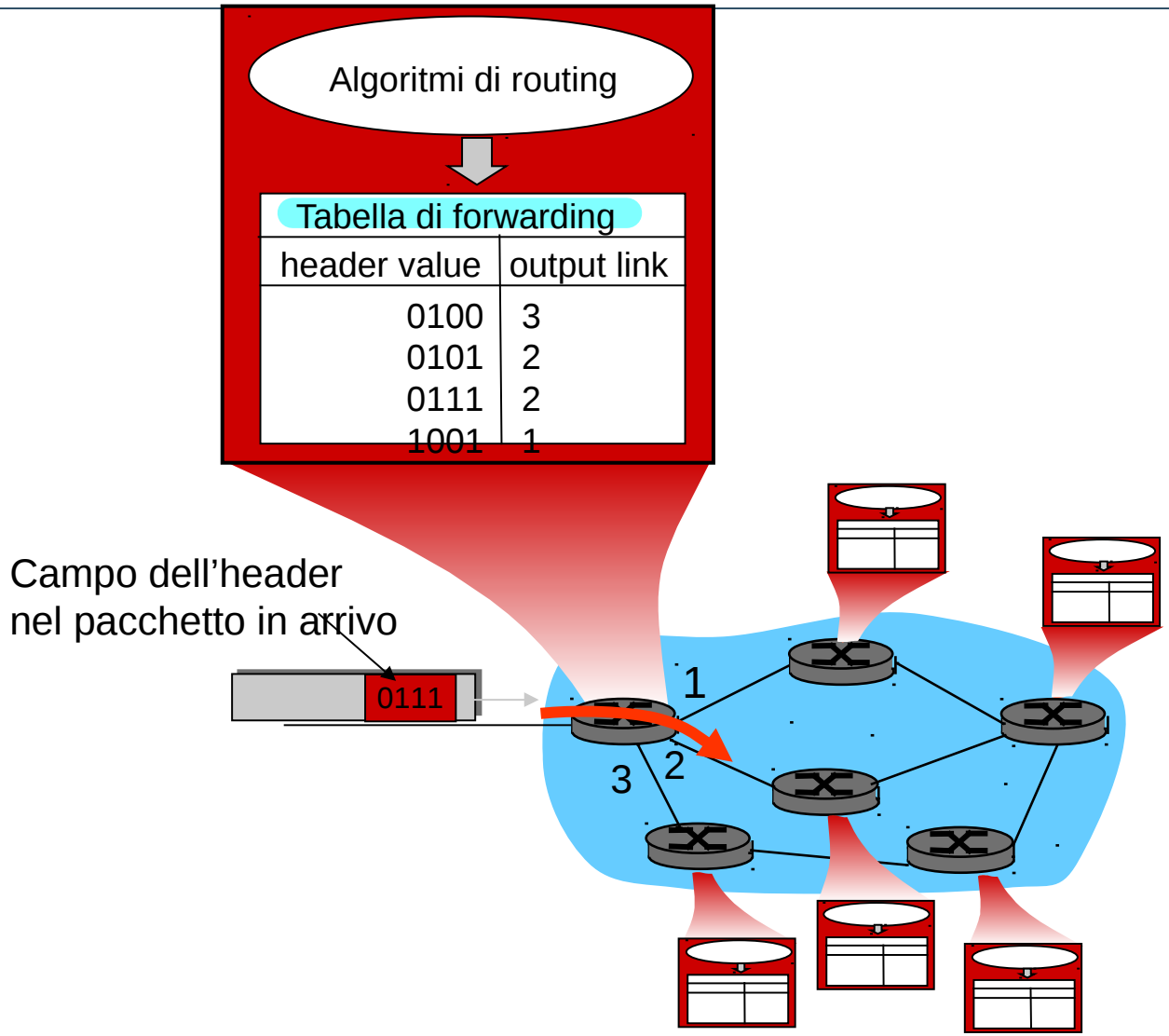
  ➔ Algoritmi di routing

    - Analogia: processo di pianificazione di un viaggio, dalla partenza all'arrivo
- Forwarding
  - Dato un router, trasferisce i pacchetti da una porta di input alla porta di out appropriata

    - Analogia: nel caso di un viaggio, passaggio attraverso un incrocio, in cui si deve scegliere quale strada prendere

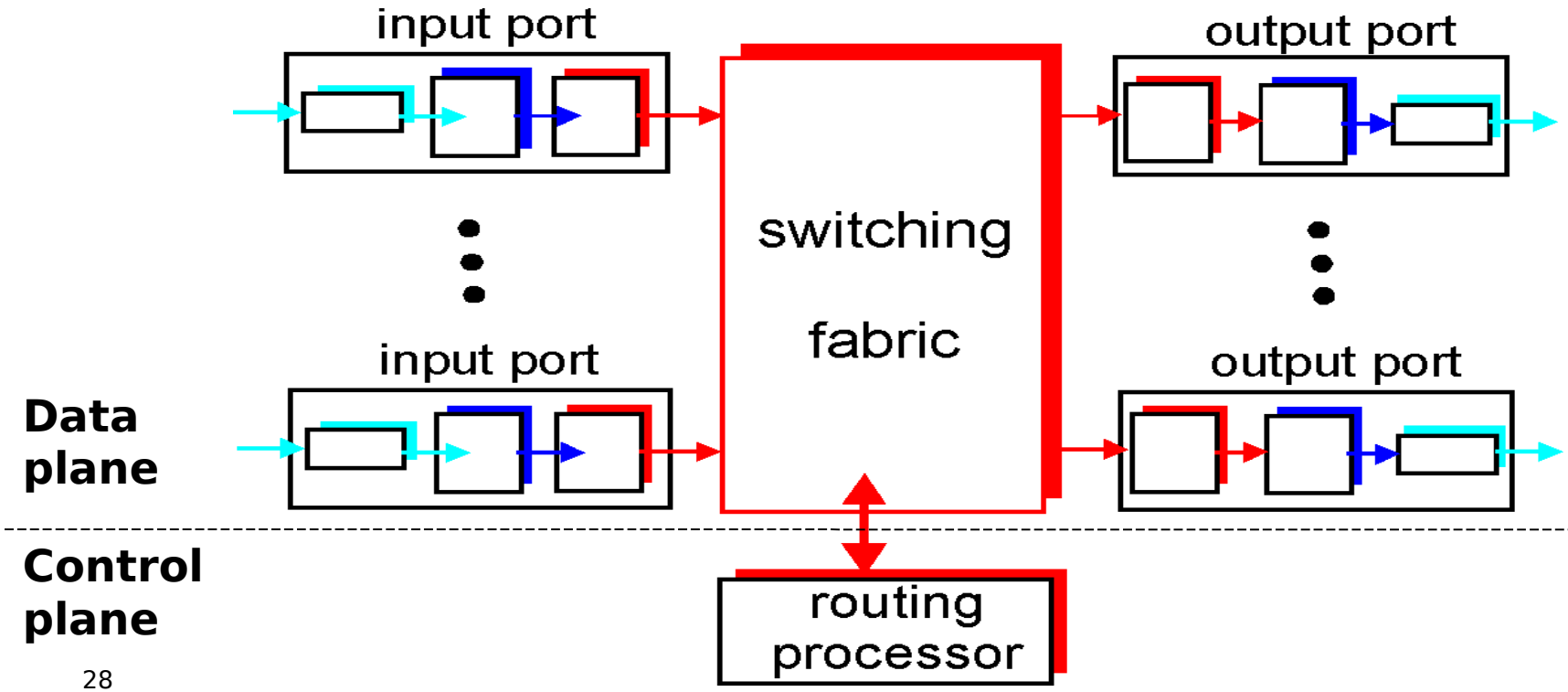
- Queste funzioni richiedono un componente fondamentale
  - l'indirizzamento (Addressing)

Interazione tra Routing e Forwarding

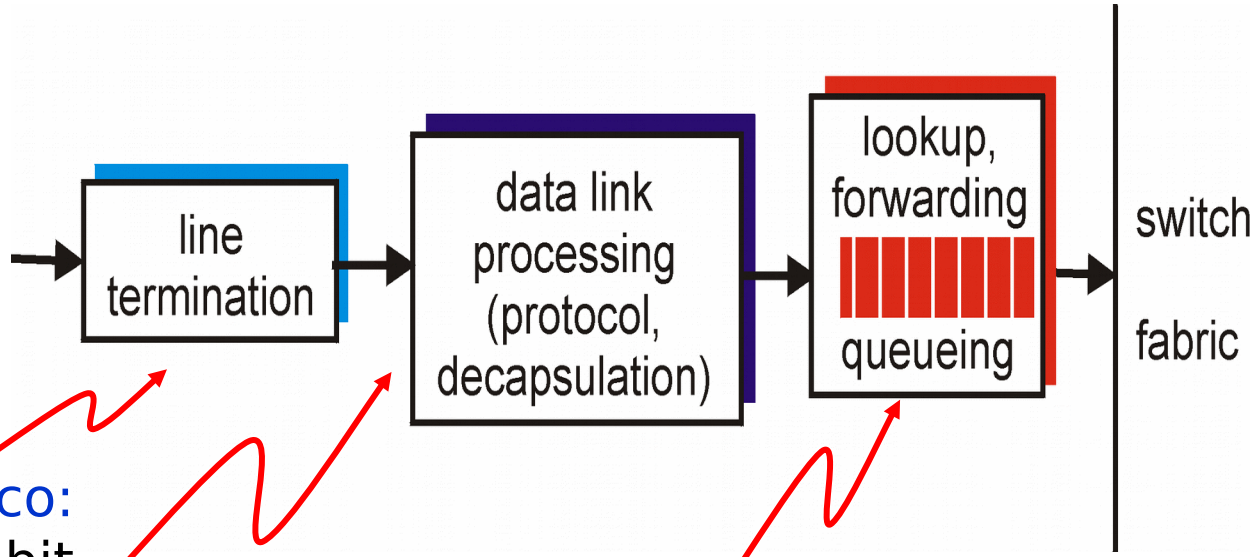


Architettura di un Router: Overview

- Funzionalità chiave:
  - Eseguire gli algoritmi e i protocolli di routing (RIP, OSPF, BGP)
  - Trasferire (commutare) datagrammi dalla porta di input alla porta di output



Funzioni delle porte di Input



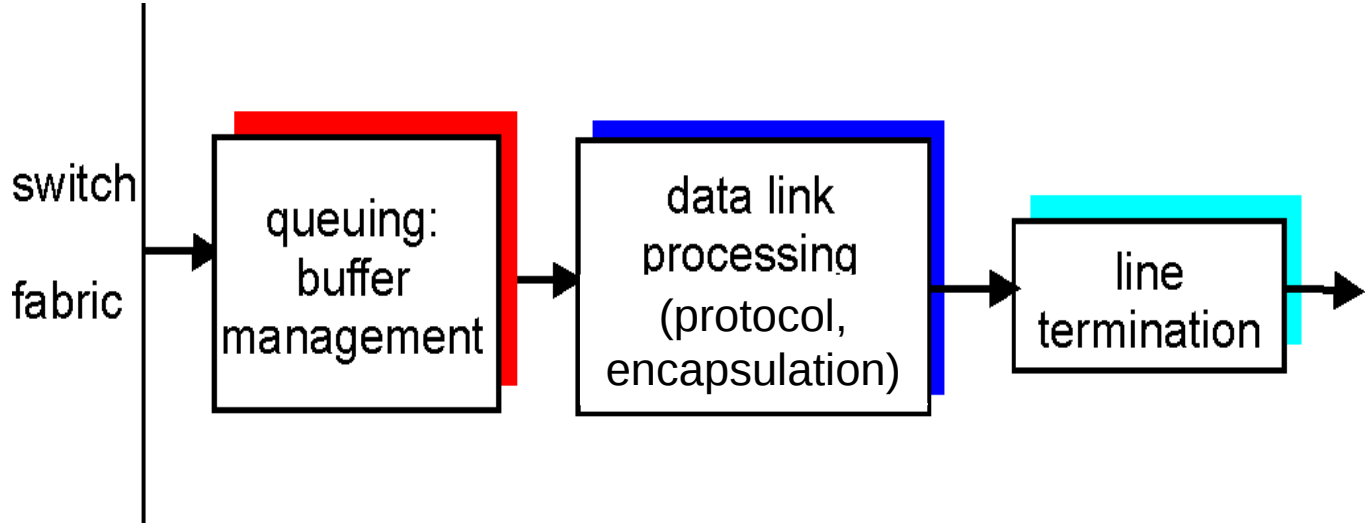
Livello fisico:
Ricezione dei bit

Livello Data link:
Ad es., Ethernet

Switching decentralizzato:

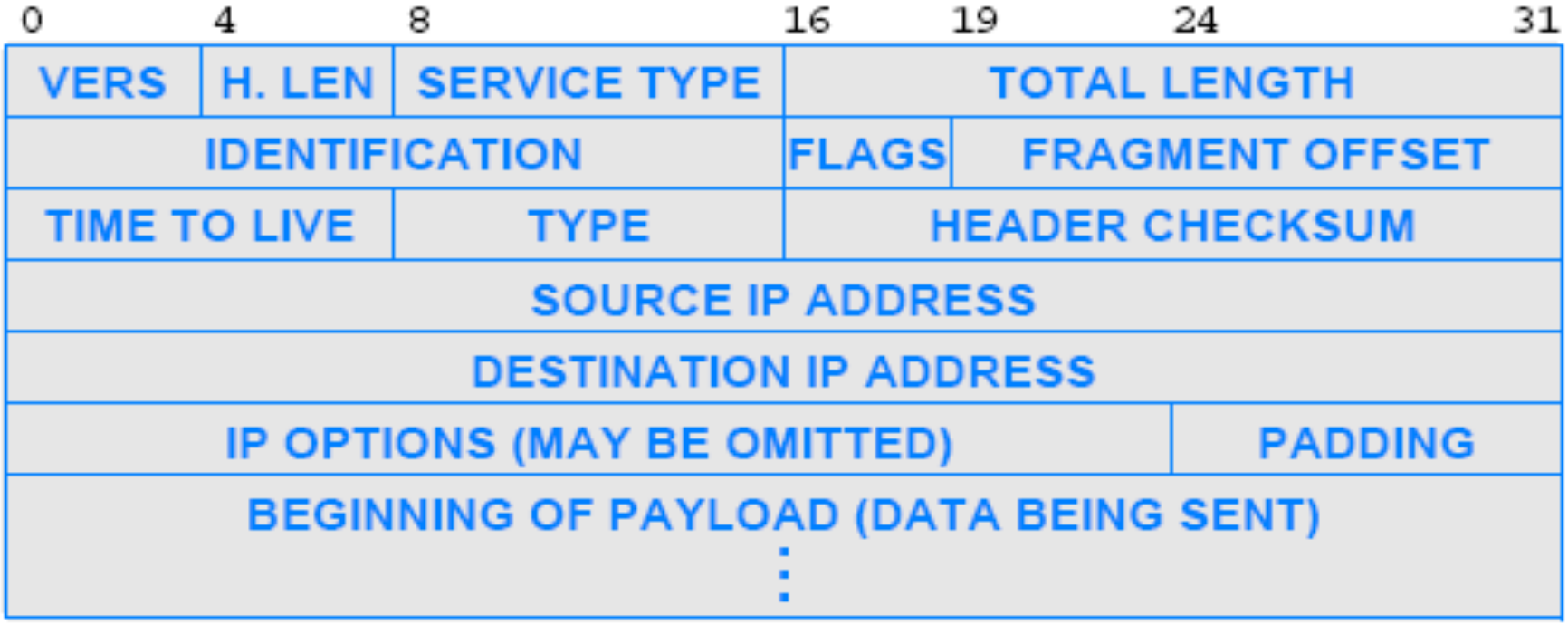
- data la destinazione, determinare la porta di output utilizzando le tabelle di forwarding in memoria locale
- scopo: completare l'elaborazione dei pacchetti a velocita' di linea
- queuing: gestire le code in caso i datagrammi arrivino con velocita' superiore alla velocita' di uscita (i canali in uscita sono occupati)

Porte di Output



- Queuing: necessario quando la velocità di arrivo dei datagrammi è superiore alla velocità di uscita
- Scheduling: utilizzato per determinare l'ordine di trasmissione dei datagrammi in coda nel buffer

Il formato dell'header del datagramma IP



Gerarchia degli indirizzi IP

- Gli indirizzo IP sono divisi concettualmente in due parti:
- Un prefisso ➔ identifica la rete fisica a cui l'host e' connesso (noto anche come NetID)
  - A ciascuna rete in Internet viene assegnato un prefisso di rete univoco
- Un suffisso ➔ identifica un host specifico all'interno di una rete (noto anche come HostID)
  - A ciascun host su una data rete viene assegnato un suffisso univoco mentre il prefisso è uguale per tutti
- Lo schema degli indirizzi IP garantisce due proprieta' :
  - A ciascun host viene assegnato un indirizzo unico
  - L'assegnazione dei numeri di rete (prefissi) viene coordinata globalmente
    - I suffissi vengono assegnati localmente, senza bisogno di coordinazione globale

Subnet e indirizzamento classless

- Prefissi di lunghezza fissa → indirizzamento classful (4 classi)
- Con l'espansione di Internet lo schema classful originale si e' rivelato limitante
- Tutti chiedevano indirizzi di classe A o B
  - In modo da poter avere un numero sufficiente di indirizzi per eventuali espansioni
    - conseguente sottoutilizzo di indirizzi all'interno di ogni classe
- Sono stati proposti due meccanismi correlati per risolvere tali limitazioni
  - Subnet
  - Indirizzamento Classless
- Invece di avere un insieme ristretto di lunghezze per i prefissi / suffissi, la scelta della lunghezza viene resa arbitraria

Notazione CIDR

- Classless Inter-Domain Routing (CIDR)
- L'utilizzo di maschere per specificare la dimensione del prefisso viene fatta per questioni di efficienza
  - operazioni di AND bit a bit molto veloci in hardware
- Tuttavia, per facilitare la gestione da parte degli utenti, si utilizza una notazione piu' semplice e diretta
  - viene specificata la dimensione del prefisso
- In particolare, la notazione CIDR prevede la seguente forma:

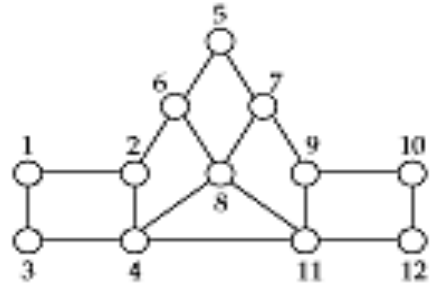
  ddd.ddd.ddd.ddd/m
  - ddd e' il valore decimale nella notazione decimale puntata
  - m è il numero di bit del prefisso
- Esempio di definizione di reti in CIDR:
  - NetA: 128.10.0.0/16
  - NetB: 64.10.2.0 /24

Routing: che cos'e'?

- Processo di scoperta del cammino da una sorgente ad ogni destinazione nella rete
- Si assuma che un utente si voglia collegare con un server nell'Antartide dal proprio desktop
  - quale cammino dovrebbe prendere?
  - esiste un cammino piu' corto? e piu' veloce?
  - cosa succede se un link sul cammino individuato si guasta?
- Il routing gestisce questi tipi di problemi

Nozioni di base

- Un protocollo di routing gestisce una tabella di routing nei router
  - la tabella indica, per ogni destinazione, qual è l'**interfaccia di uscita** su cui inviare il pacchetto
- I nodi fanno scelte locali basandosi su topologia globale
  - questo rappresenta il problema principale



ROUTING TABLE AT 1

Destination	Next hop	Destination	Next hop
1	—	7	2
2	2	8	2
3	3	9	2
4	3	10	2
5	2	11	3
6	2	12	3

Problema chiave

- Come effettuare decisioni locali corrette?
  - ciascun router deve conoscere qualcosa sullo stato globale
- Stato globale della rete
  - intrinsecamente grande
  - dinamico
  - informazioni di difficile reperibilità
- Un protocollo di routing deve saper riassumere le informazioni più importanti

Tipologie di algoritmi di routing

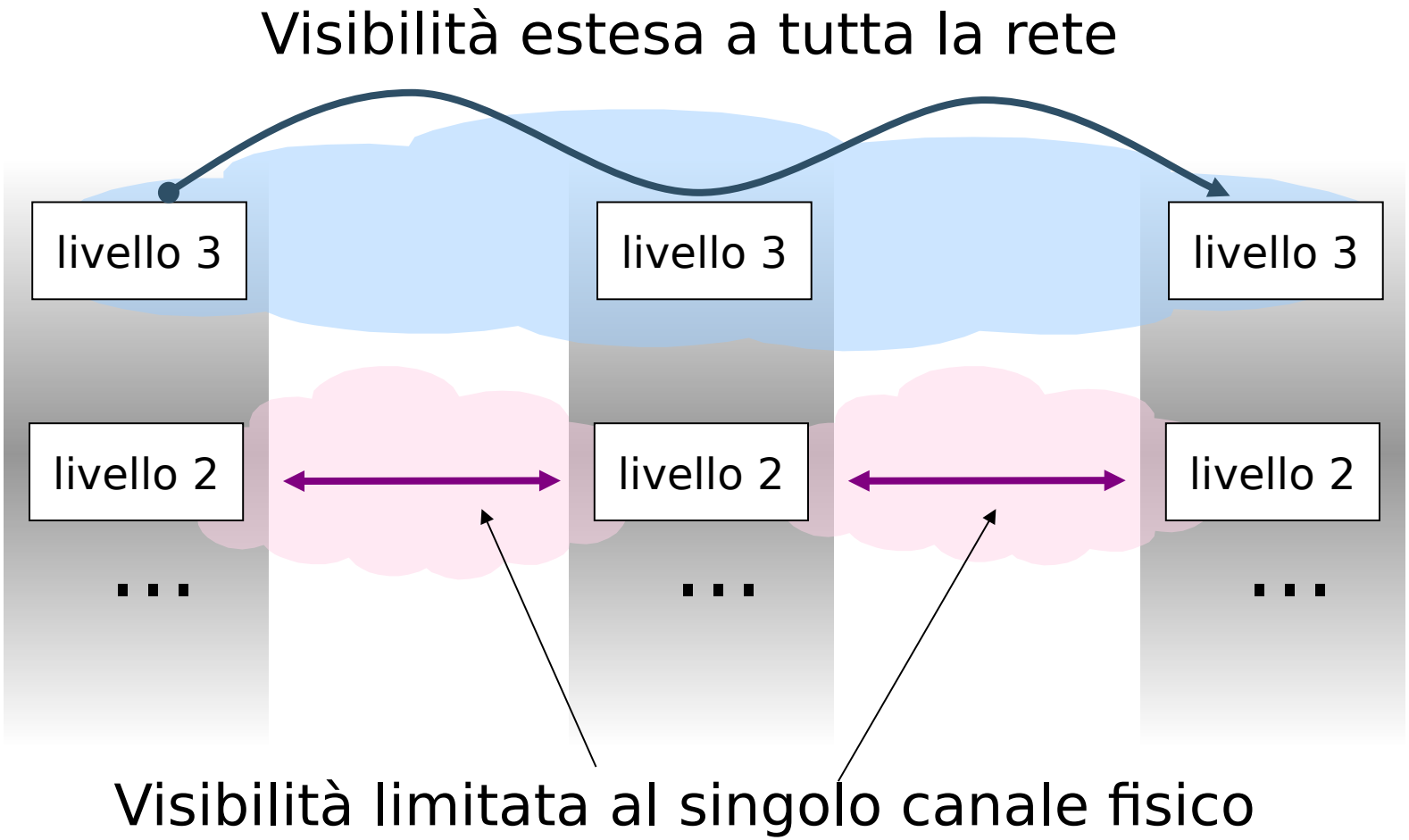
Distance vector (DV):

- Periodicamente ogni nodo **invia ai propri vicini** il vettore delle distanze (distance vector, DV) **verso tutte le sottoreti del mondo**

Link state:

- Periodicamente ogni nodo **invia a tutte le sottoreti del mondo** lo stato dei collegamenti (link state) **verso propri vicini**

Visibilità della rete del livello 2 e 3



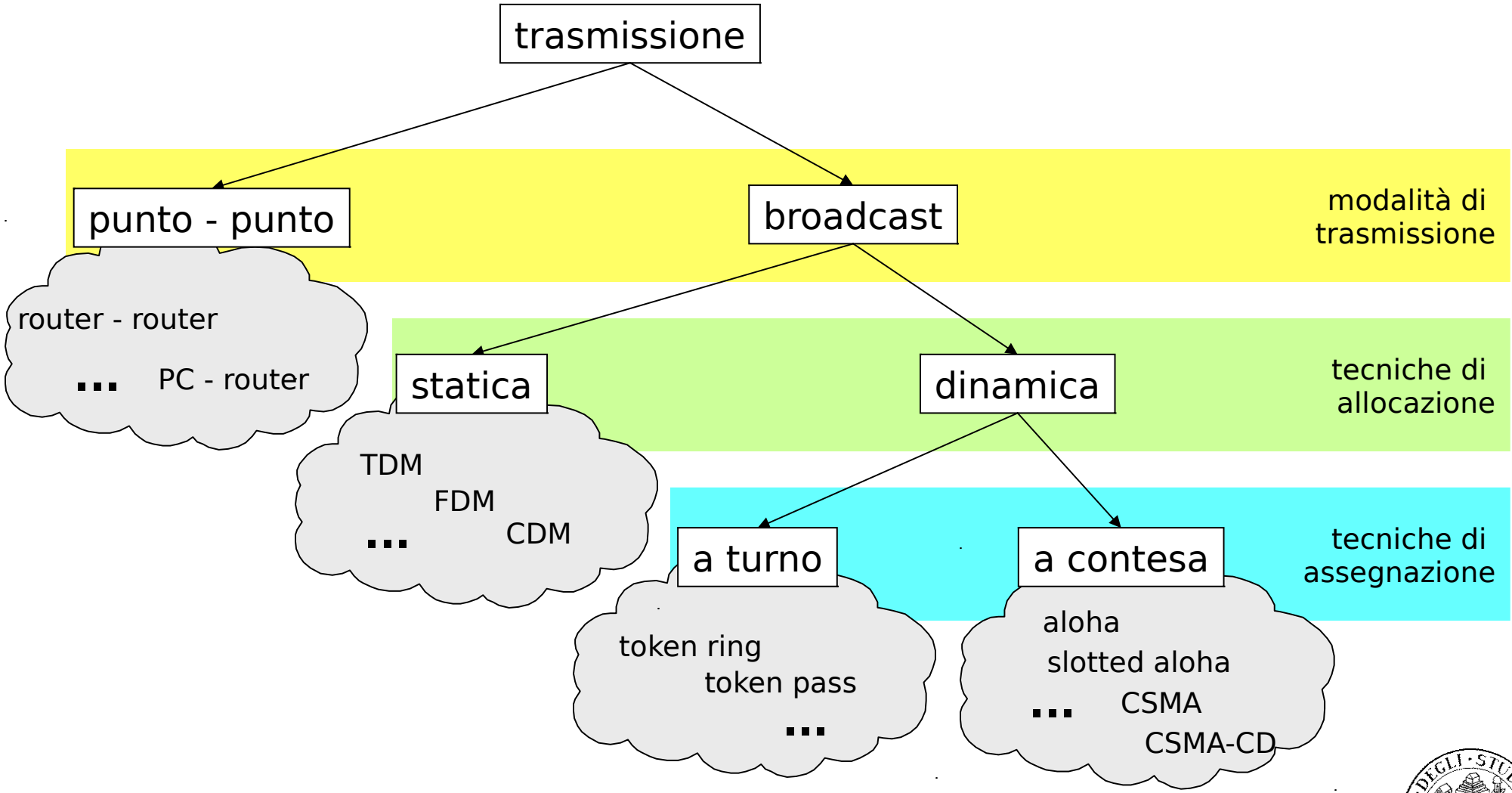
Funzioni di competenza del livello 2

- Le principali funzioni svolte dal livello 2 sono:
  - framing
    - delimitazione dei messaggi (frame) come gruppi di bit sul canale fisico
  - rilevazione/gestione errori
    - controlla se il frame contiene errori ed eventualmente gestisce il recupero
  - metodo di accesso al canale
  - controllo di flusso
    - gestisce la velocità di trasmissione

Framing

- Il livello 2 riceve dal livello superiore (rete) dei pacchetti
- Considerando che:
  - la lunghezza dei pacchetti (di livello 3) e delle corrispondenti trame (livello 2) è variabile
  - i sistemi non sono sincronizzati tra loro, ovvero non hanno un orologio comune che segna la stessa ora per tutti
  - il livello 1 tratta solo bit, e quindi non è in grado di distinguere se un bit appartiene ad una trama o a quella successiva
- ... nasce il problema della ***delimitazione dei frame***
- La funzionalità di *framing* è dunque di rendere distinguibile un frame dall'altro attraverso l'utilizzo di opportuni codici all'inizio e alla fine del frame stesso

Metodo di accesso al canale



Protocolli di accesso multiplo

- In letteratura sono disponibili molti algoritmi di accesso multiplo al mezzo condiviso con contesa
- Principali algoritmi (utilizzati dai protocolli):
  - ALOHA
    - Pure ALOHA
    - Slotted ALOHA
  - Carrier Sense Multiple Access Protocols
    - CSMA
    - CSMA/CD (con rilevazione della collisione – Collision Detection)
    - CSMA/CA (con tentativo di riduzione delle collisioni – Collision Avoidance)

Carrier Sense Multiple Access (CSMA)

- Ambito LAN: le stazioni possono monitorare lo stato del canale di trasmissione (ritardi bassi)
- Le stazioni sono in grado di "ascoltare" il canale *prima di iniziare a trasmettere* per verificare se c'è una trasmissione in corso
- Algoritmo
  - se il canale è libero, si trasmette
  - se è occupato, sono possibili diverse varianti
    - non-persistent
      - rimanda la trasmissione ad un nuovo istante, scelto in modo casuale
    - persistent
      - nel momento in cui si libera il canale, la stazione inizia a trasmettere
  - se c'è collisione, come in ALOHA, si attende un tempo casuale e poi si cerca di ritrasmettere