

PARTE 3

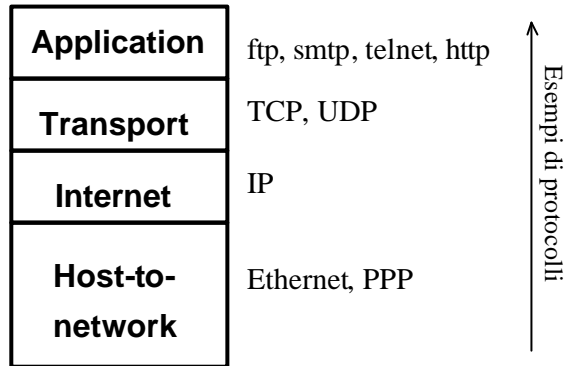
SUITE PROTOCOLLI TCP/IP ("I protocolli di Internet")

Parte 3

Modulo 1: Stack TCP/IP

TCP/IP Protocol Stack (*standard de facto*)

- Basato su 5 livelli invece che sui 7 dello stack ISO/OSI



Livello 1-2 (*host-to-network*)

- I primi due livelli (*fisico* e *data link*) non sono separati, nel senso che **connessione fisica** e protocollo **data link** sono interdipendenti
- Pertanto, è più corretto parlare di un livello **host-to-network (h2n)** che comprende i primi due livelli
- Esempi di protocolli h2n:
 - **Ethernet** (protocollo per LAN)
 - **PPP** (protocollo per connessioni via modem)

Livello 3 (*network*): Protocollo IP

- Protocollo per la consegna dei pacchetti da un host mittente ad un host destinatario
- **Servizi aggiuntivi rispetto a h2n**
 - *identificativo univoco di ciascun host (indirizzo IP)*
 - *comunicazione logica tra host*
- **Ma**
 - **privo di connessione**: ogni pacchetto è trattato in modo indipendente da tutti gli altri
 - **non affidabile**: la consegna non è garantita (i pacchetti possono essere persi, duplicati, ritardati, o consegnati senza l'ordine di invio)
 - **consegna con impegno**: tentativo di consegnare ogni pacchetto (l'inaffidabilità deriva dalle possibili congestioni della rete o guasti dei nodi/router)

Livello 4 (*transport*)

- Il livello transport estende il servizio di consegna con impegno proprio del protocollo IP tra due host terminali ad un servizio di consegna a due processi applicativi in esecuzione sugli host
- **Servizi aggiuntivi rispetto a IP**
 - *multiplazione e demultiplazione messaggi tra processi*
 - *rilevamento dell'errore (mediante checksum)*
- Esempi di protocolli transport
 - **UDP (User Datagram Protocol)**
 - **TCP (Transmission Control Protocol)**: offre servizi aggiuntivi rispetto a UDP

Livello 4 (*transport*)

[UDP]

- Protocollo che fornisce un livello di trasporto dell'informazione connectionless
- Specifica in [RFC 768]

Livello 4 (*transport*)

[TCP]

- Protocollo che fornisce un livello di trasporto affidabile e orientato alla connessione
- *Servizi aggiuntivi rispetto a UDP*
 - **orientato alla connessione**: comprende l'instaurazione, l'utilizzo e la chiusura della connessione
 - **orientato al flusso di dati**: considera il flusso di dati dall'host mittente fino al destinatario (→ considera sia rete sia host terminali)
 - **trasferimento con buffer**: i dati sono memorizzati in un buffer e poi inseriti in un pacchetto quando il buffer è pieno
 - **connessione full duplex** (bi-direzionale): una volta instaurata una connessione, è possibile il trasferimento contemporaneo in entrambe le direzioni della connessione

Livello 5 (*application*)

- Il livello application utilizza il livello di trasporto dell'informazione tra processi in esecuzione su host terminali per realizzare applicazioni di rete
- Esempi protocolli applicativi
 - ftp
 - telnet
 - http
 - smtp
 - irc
 - ...

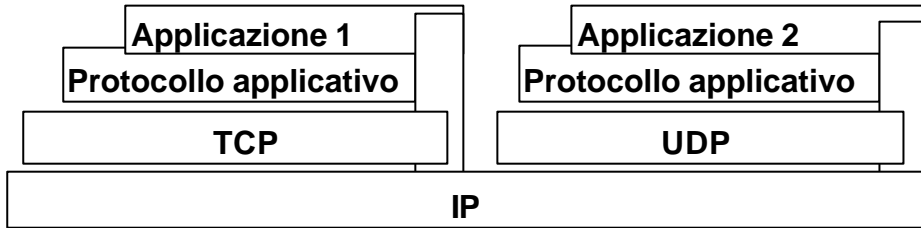
NOTA: Applicazioni di rete¹ protocolli applicativi

Applicazioni di rete e protocolli applicativi

- Tipicamente, ciascuna applicazione di rete definisce un nuovo protocollo di livello applicativo ed un'interfaccia utente
- Esempi
 - *Trasferimento file*: usa protocollo applicativo **ftp**
 - *Collegamento a terminale remoto*: usa protocollo **telnet**
 - *World Wide Web*: usa protocollo **http**
 - *Posta elettronica*: usa protocollo **smtp**
 - *Chat*: usa protocollo **irc**
 - ...

“Lo stack visto dall’alto”

- Ciascun protocollo di livello applicativo deve utilizzare il protocollo IP, ma può utilizzare diversi protocolli di livello trasporto (TCP o UDP)
- Vista concettuale dello stack TCP/IP da parte dello sviluppatore di applicativi di rete



Protocolli di trasporto per applicazioni di rete

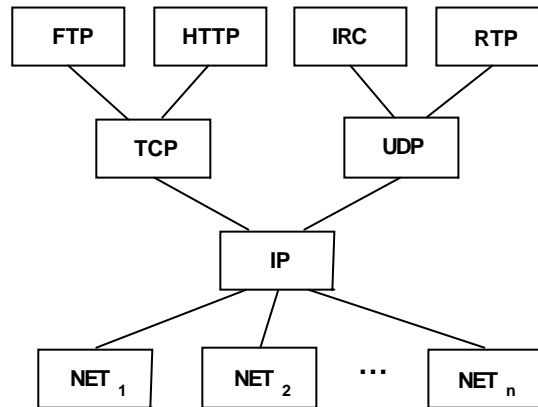
Applicazioni che usano TCP

- World Wide Web
- Trasferimento file
- Collegamento a terminale remoto
- Posta elettronica

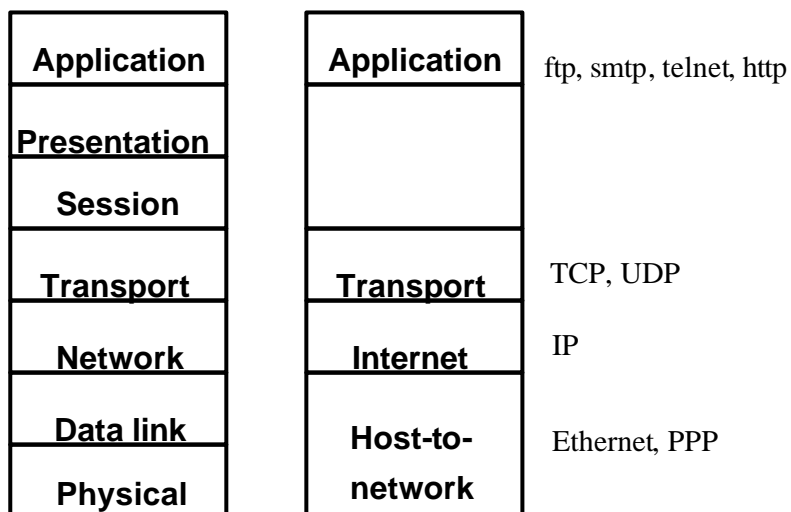
Applicazioni che usano UDP

- Streaming di dati multimediali
- Teleconferenze
- Telefonia su Internet

Progetto Internet “a clessidra”



Stack ISO/OSI e TCP/IP



Critiche

Al modello ISO/OSI

- **Cattiva tempistica**
- **Cattiva tecnologia**
 - Influenzato dal modello IBM-SNA
 - Ridondanze
- **Cattiva implementazione**
 - Complessità
 - Eccessivi 7 livelli per la tecnologia del tempo
- **Pessima politica**
 - Modello imposto contro il libero TCP/IP legato a Unix

Al modello TCP/IP

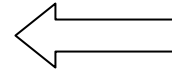
- **Poco generale**
- **Meno concettuale e più orientato al funzionamento**
- **Livelli host-to-network confusi e interdipendenti**
- **Protocolli sviluppati ad hoc**

Parte 3

Modulo 2: Principi di funzionamento

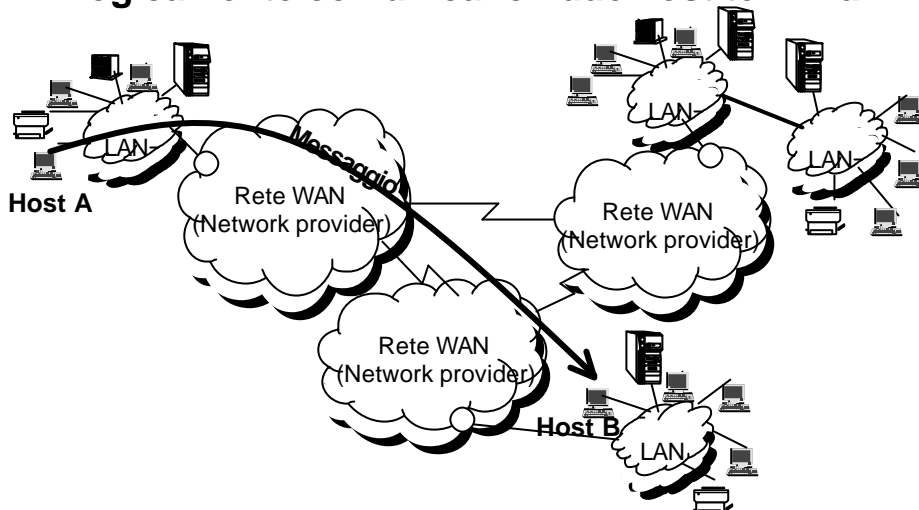
Indice

- **Comunicazioni in Internet**
 - Transito dell'informazione
 - PDU dello stack TCP/IP
 - Uso della rete: *Circuit-switching* vs. *Packet switching*
- **Tipologie di servizio di trasporto**
 - Affidabili e non affidabili
 - *Connection-oriented* e *connectionless*



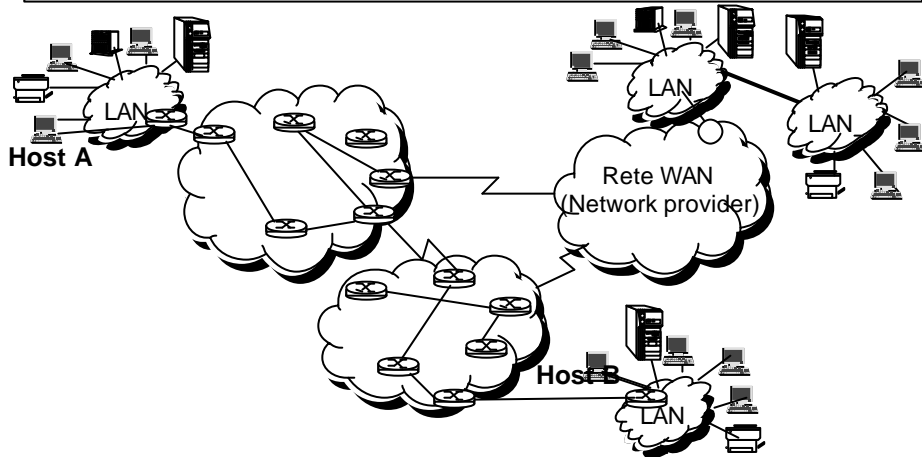
Comunicazione in Internet [vista 1]

Logicamente comunicano i due host terminali



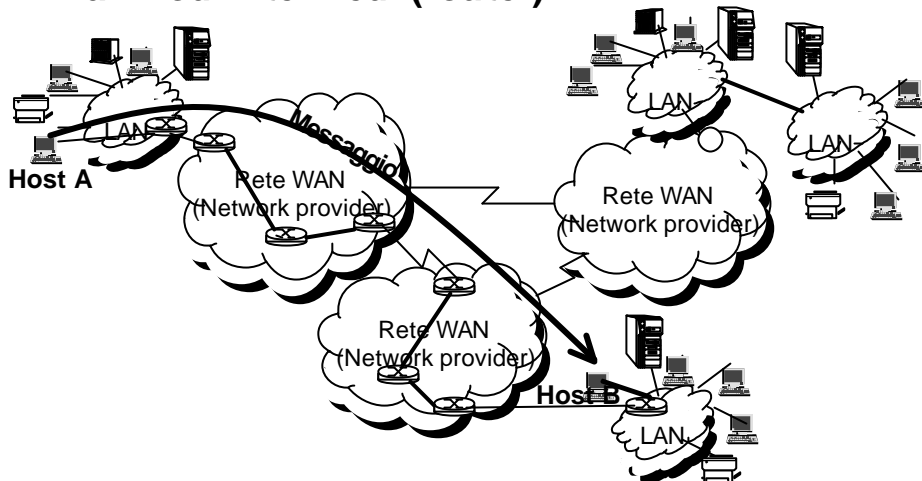
Architettura Internet

In realtà, Internet consiste in milioni di **host** (computer, PDA, TV,...), dispositivi che instradano i messaggi (**router**) e **link** di comunicazione (cavi, fibra ottica, satellitari,...)



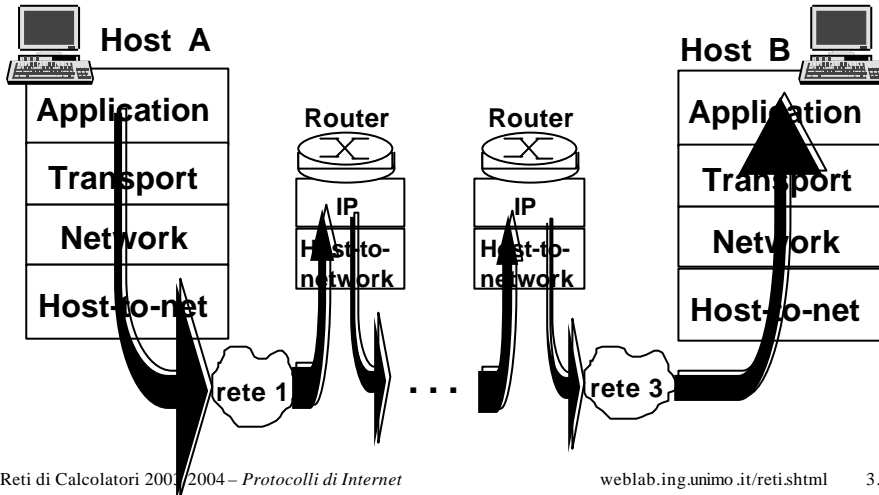
Comunicazione in Internet [vista 2]

Quindi, in realtà il messaggio deve attraversare **vari nodi intermedi (router)**



Comunicazioni in Internet [vista 3]

In ciascun nodo, l'informazione attraversa tutti i livelli necessari (5 per host, 3 per router)

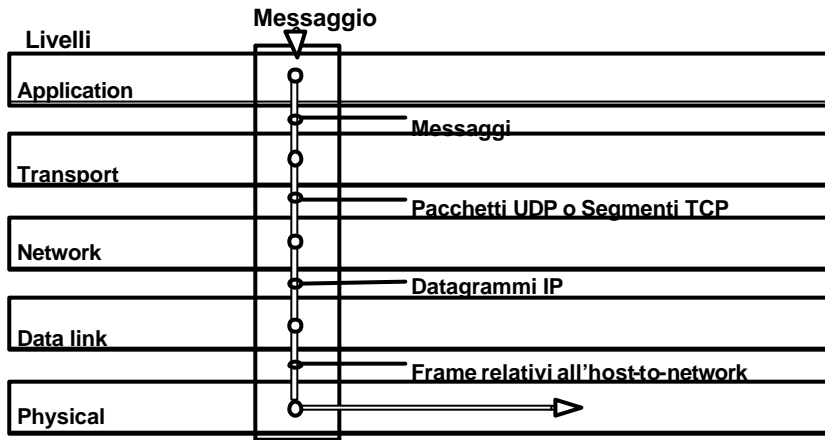


PDU dello Stack TCP/IP

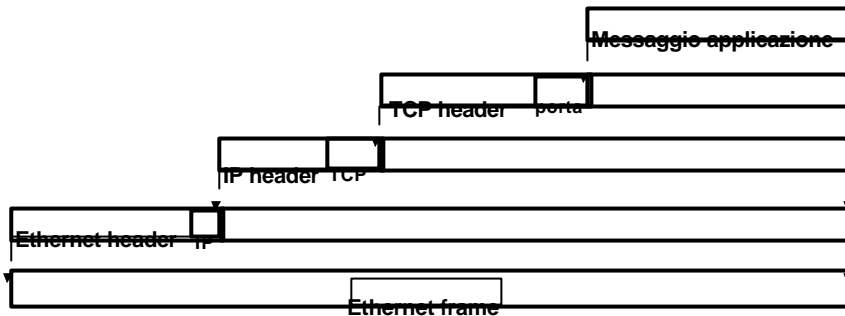
PDU: Protocol Data Unit

TCP/UDP IP	Application	messaggio
	Transport	segmento TCP
	Network	datagramma IP
	Data link	frame
	Physical	

Livelli del modello TCP/IP

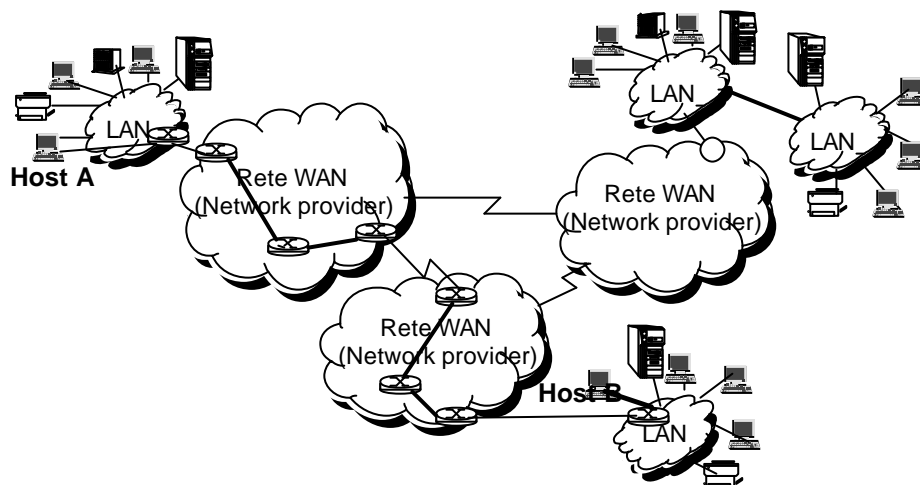


Incapsulamento di un messaggio trasmesso mediante TCP su rete Ethernet



Riprendiamo [vista 2] di Internet

Internamente, Internet è una rete di router interconnessi mediante una topologia non regolare



Reti di Calcolatori 2003/2004 – *Protocolli di Internet*

weblab.ing.unimo.it/reti.shtml

3.25

Come è possibile trasferire dati?

- **Circuit-switching**

- Un circuito virtuale dedicato per ogni comunicazione
→ **L'idea alla base del sistema telefonico**

- **Packet-switching**

- I dati sono suddivisi in “parti” ed inviati attraverso la rete
→ **L'idea alla base di Internet**

Reti di Calcolatori 2003/2004 – *Protocolli di Internet*

weblab.ing.unimo.it/reti.shtml

3.26

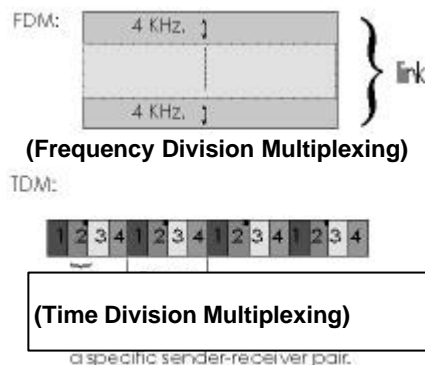
Circuit-switching

- **Necessità di riservare tutte risorse end-to-end prima di trasmettere**
- Banda del link, capacità degli switch
- **Risorse dedicate:** non c'è possibilità di condivisione
- Necessaria una fase di setup per ogni chiamata
- Prestazioni garantite dalla tipologia di risorse riservate

Circuit Switching (*cont.*)

In realtà le risorse di rete (per es., la banda) sono divise in “parti”

- Le parti sono assegnate alle chiamate
- Le parti di risorse, riservate per una chiamata, non sono utilizzabili da altre anche se non sono utilizzabili dalla chiamata che le possiede (non c'è possibilità di condivisione)
- Vi sono due approcci per la suddivisione di banda
 - basata sulla frequenza
 - basata sul tempo



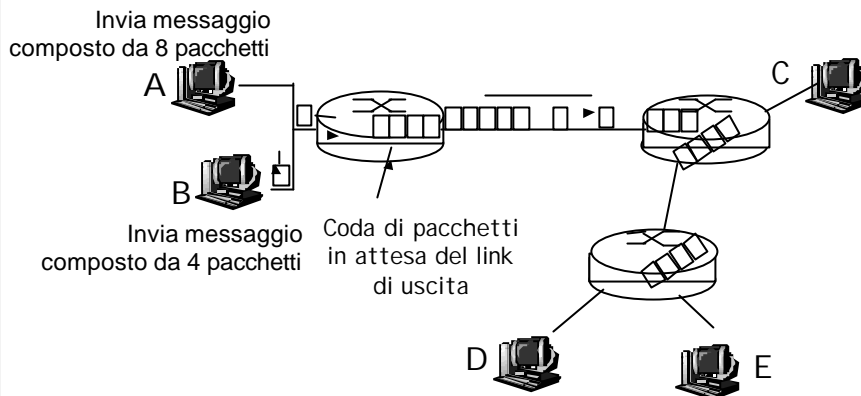
Packet Switching

Ogni comunicazione è suddivisa in pacchetti

- I pacchetti *condividono* le risorse della rete
- Ogni pacchetto utilizza tutta la capacità trasmissiva di un link
- **Le risorse utilizzate sulla base della necessità e non della prenotazione**

Prenotazione delle risorse
Assegnamento esclusivo
Divisione della banda in parti

Packet Switching (cont.)

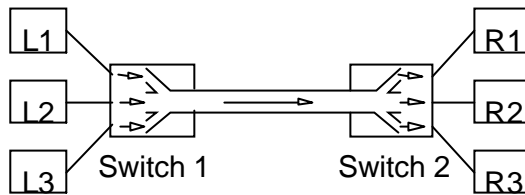


“Analogia del ristorante”

- **Circuit switching** = con prenotazione del tavolo
- **Packet switching** = senza prenotazione

Multiplexing

- **Multiplexing (*condivisione*) di risorse**

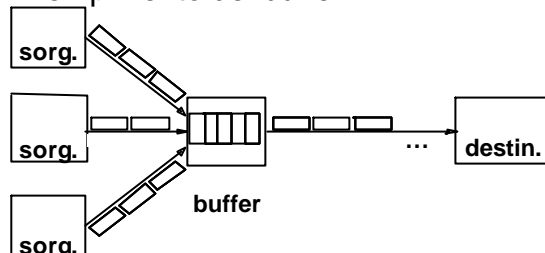


Deterministico:

- Time-Division Multiplexing (TDM)
- Frequency-Division Multiplexing (FDM)

Multiplexing statistico

- **Suddivisione di tempo, ma su richiesta invece che ad intervalli prefissati**
- Pacchetti da diverse sorgenti “mescolate” sullo stesso link
- Si bufferizzano i pacchetti in conflitto per lo stesso link
- La coda dei pacchetti può essere processata in ordine FIFO, ma non necessariamente
- **Congestione**= riempimento del buffer



Trasmissioni e conflitti nel packet switching

Comunicazione store and forward

(i pacchetti si muovono di un hop alla volta)

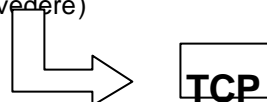
1. trasmessi su un link, arrivano ad un router
2. aspettano (presso il router), il loro turno per poter essere trasmessi sul successivo

Conflitto di risorse

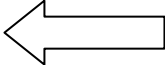
- La domanda aggregata di risorse può eccedere la quantità disponibile
- Non essendoci prenotazione, si possono creare **congestioni** (*impreviste*):
 - i pacchetti rimangono accodati (se c'è spazio) in attesa di poter utilizzare il link
 - Se la coda è piena, il pacchetto viene perduto
- Possibilità di utilizzare un link differente a seconda dello stato della rete

Packet-switching

- C'è condivisione di risorse
- Non c'è la necessità di prenotare risorse end-to-end
- Il packet switching è ottimo per dati che arrivano in gruppi
- **Rischi di congestione:**
 - Ritardo e perdita di pacchetti
 - E' necessario un protocollo che garantisca almeno le seguenti due proprietà:
 - **Trasferimento dei dati affidabile** (in grado di capire se c'è perdita di pacchetti e in grado di provvedere)
 - **Controllo della congestione**



Indice

- **Comunicazioni in Internet**
 - Transito dell'informazione
 - PDU dello stack TCP/IP
 - Uso della rete: *Circuit-switching* vs. *Packet switching*
- **Tipologie di servizio di trasporto** 
 - Affidabili e non affidabili
 - *Connection-oriented* e *connectionless*

Modalità di servizi di trasporto

- **Orientati alla connessione (*connection oriented*)**

Esempio: Modello telefonico

 - Si apre la connessione
 - Si scambiano i messaggi
 - Si chiude la connessione
- **Non orientati alla connessione (*connectionless*)**

Esempio: Modello postale

 - Il messaggio viene affidato al servizio di trasporto

Servizi di trasporto a confronto

- **Connectionless**
 - Senza setup iniziale
 - Indirizzo di destinazione nei pacchetti
 - Ordine dei pacchetti non garantito
 - Senza controllo del flusso
- **Connection oriented**
 - Con setup iniziale
 - Indirizzo di destinazione nel setup e/o nei pacchetti
 - Ordine dei pacchetti garantito
 - Con controllo del flusso

Servizi affidabili e non affidabili

- **Servizi affidabili**
 - Utilizzano le primitive response/confirm per gestire la ritrasmissione dei messaggi in caso di errori
 - Garantiscono la ricezione completa e corretta di tutti i messaggi
 - Sono più lenti
- **Servizi non affidabili**
 - Non garantiscono la corretta ricezione dei messaggi
 - Utilizzano soltanto le primitive request/indication

Esempio 1: Caratteristiche TCP

Il trasferimento dati tra host end-to-end avviene in 3 fasi:

1. **Handshaking**: fase di setup in cui ci si prepara per il trasferimento dei dati
 - Si deve arrivare ad uno stato di setup in entrambi gli host che devono comunicare
2. **Trasmissione**: fase di trasmissione di uno o più pacchetti
3. **Chiusura connessione**

Servizi del TCP [RFC 793]

- **affidabilità**, trasferimento ordinato di uno stream di dati
 - *acknowledgement*
 - perdita gestita con ritrasmissione
- **congestion control**:
 - L'host mittente deve diminuire il tasso di trasmissione dei pacchetti quando la rete è congestionata
- **flow control**:
 - L'host mittente non deve sovraccaricare il ricevente

Esempio 2: Caratteristiche UDP

- Similitudini con TCP
 - Lo scopo è sempre quello di trasferire dati da un host all'altro
- Differenze con TCP
 - Trasferimento dei dati non affidabile
 - Non c'è flow control
 - Non c'è congestion control

Dove ci troviamo?

- Abbiamo introdotto i termini fondamentali del corso
- Abbiamo accennato ai concetti principali che verranno trattati in profondità nell'ambito del corso

DA FARE:

