

- Programmazione di reti
 - Comunicazione
 - Telecomunicazione
 - Canale
 - Rete
 - Componenti reti
 - Topologia di rete
 - Esempi di tipologie reti
 - Maglia completa
 - A stella
 - Anello
 - Bus
 - Informazioni segnali e digitalizzazione
 - Esempio sinusoida
 - Larghezza di banda
 - Come fare passare più canali nello stesso mezzo trasmissivo?
 - CDM code division multiplexing
 - FDM Multiplazione a divisione di frequenza (frequency division multiplexing)
 - TDM time division multiplexing
 - TDM slotted
 - TDM unslotted
 - Assegnazione della banda
 - Statica
 - Dinamica
 - PCM pulse code modulation
 - funzioni di rete e commutazione
 - funzioni di rete
 - Tecniche di commutazione
 - Fasi della comunicazione a Commutazione di circuito
 - Reti a pacchetto
 - ISO-OSI (open system interconnection)
 - Terminologia
 - Modalità di servizio
 - Modalità di dialogo
 - Segmentazione e riassemblamento
 - Multiplazione e splitting
 - Uso dei SAP
 - Internet

- Arpanet
- Timeline
- Enti di gestione di internet
- RFC
- InterNIC (Network Information Center)
- IANA Internet Assigned Number Authority
- Indirizzamento
- Indirizzo globale e locale
- URL
 - COOKIES
 - Rete Locale
 - MAC address
 - Indirizzo IP
 - Numero di porta
- Implementazione dei servizi e modelli di interazione
 - Nomenclatura
 - Vari modelli
 - Uno a molti
 - Peer to peer
- Ricerca destinazione
- Domain Name System
 - Flags di DNS
 - PDU DNS
 - Formato domanda
 - Formato risposta
 - PDU
 - SDU
 - PCI
- Protocolli applicativi
 - architettura generale protocolli applicativi
 - HTTP
 - Formato messaggi
 - FTP file transport protocol
 - Funzionamento
 - Posta elettronica:
 - SMTP simple mail transform protocol
 - 3 fasi:
 - POP3 Post Office Protocol

- IMAP
- HTTPS
 - Fattori di identificazione:
 - Tipologie attacchi
 - Strategie di sicurezza
 - Crittografia
 - Algoritmi di cifratura
 - Cifratura simmetrica
 - Cifratura asimmetrica
 - Firma digitale
- SSH
- Socket
 - Esempio di stream socket : comunicazione affidabile
 - Esempio di datagram socket: comunicazione non affidabile
 - Server iterativo
 - Server concorrente
- VoIP voice over ip
 - SIP session initiation protocol
 - Richieste
 - Risposta
 - Scenari applicativi del sip
- Controllo di canale
 - Controllo errore
 - Codice a blocco sistematico
 - Bit di parità
 - Internet checksum
 - Codici polinomiali
 - INVIO
 - RICEZIONE
 - Rivelazione dell'errore
 - Capacità del codice e scelta di $Gr(x)$
 - Errore singolo
 - Numero dispari di errori
 - Due errori
 - Errori a burst (a raffica)
 - Possibili scenari:
 - ARQ automatic repeated request
 - Finestra scorrevole

- Cos'è la finestra?
- Approfondimento
- Selective repeat ARQ
- Dimensione Time out
- Prestazioni delle reti e protocolli
 - Prestazioni stop and wait
 - Protocolli a finestra scorevole
 - Analisi con errori
 - Finestra e go back N
- Protocolli di trasporto
 - UDP
 - TCP
 - Three ways handshake

Programmazione di reti

Comunicazione

Condividere un'informazione, l'informazione è una risposta a una domanda.

La comunicazione ha dei limiti:

- temporale
 - e spaziale
- dato che la comunicazione nasce per condividere informazioni rapidamente e a persone vicine.

Telecomunicazione

Si pone l'obiettivo di trasmettere informazioni lontano

Problema tecnico:

- trasmettere lontano nello spazio

Canale

Il canale è l'entità che trasporta il flusso informativo tra i vari utenti. Le telecomunicazioni utilizzano *canali di comunicazione*, è il mezzo che trasporta l'informazione da punto A a punto B, e può avere delle caratteristiche:

Monodirezionale:

L'informazione può essere trasferita in una sola direzione

Bidirezionale:

L'informazione può essere trasferita in entrambe le direzioni

Punto-punto:

Un nodo è collegato con un singolo nodo

Punto-multipunto:

Un nodo può comunicare con tanti altri

Broadcast:

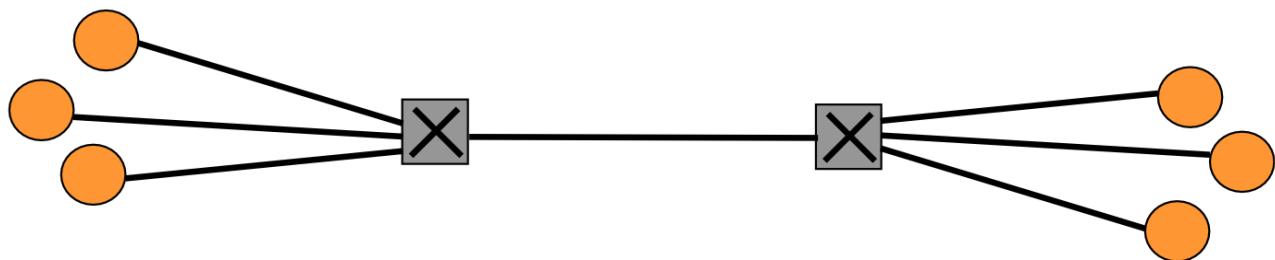
un nodo trasmette allo stesso tempo a tutti i nodi della rete

Multicast:

un nodo trasmette allo stesso tempo ad un sottoinsieme dei nodi

Rete

È il sistema che permette ad una grande popolazione di condividere un insieme di canali per comunicare a richiesta.



Componenti reti

Terminali → Fungono da interfaccia con l'utente finale, Codificano l'informazione in modo consono ad essere trasferita in rete

Collegamenti → Permettono il trasporto dell'informazione, realizzati con il rispettivo **Mezzo trasmisivo**, che è il canale fisico (fibra, rame, satelliti)

Nodi di commutazione → Utilizzano i mezzi trasmisivi al fine di creare canali di comunicazione sulla base delle richieste degli utenti

Topologia di rete

È la descrizione geometrica di una rete, composta da:

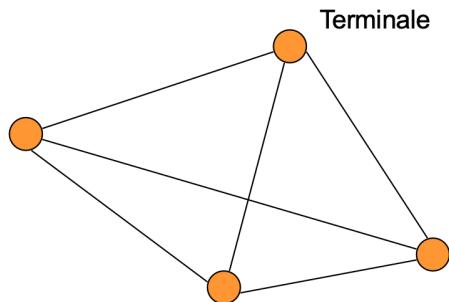
- Rami (archi)
- Nodi, punti agli estremi dei collegamenti

La rete è descrivibile attraverso un **grafo**

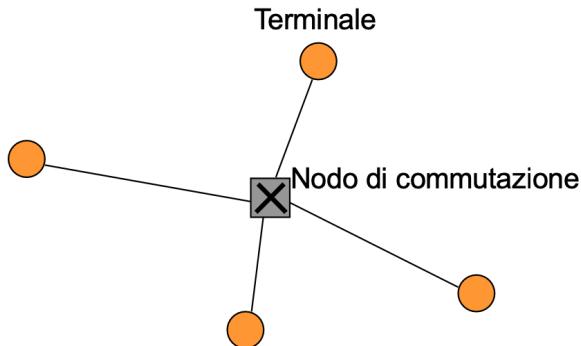
Esempi di tipologie reti

Maglia completa

- Un collegamento per ogni coppia di nodi
- N nodi implicano $N(N-1)/2$ collegamenti
 - Grande resistenza ai guasti
 - Complessità e costo



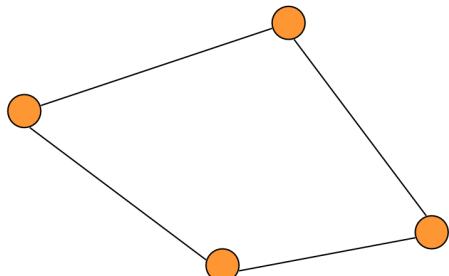
A stella



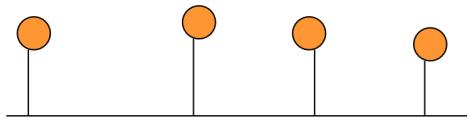
- N collegamenti
- Centro stella (attivo o passivo) deve smistare le informazioni
 - Minor costo
 - Minore resistenza ai guasti

Anello

- Anello
 - Anelli monodirezionali
 - Se un collegamento si interrompe la rete si guasta
 - Anelli bidirezionali
 - Maggiore complessità per maggiore resistenza ai guasti



Bus



Bus bidirezionale

- Il mezzo di trasmissione è condiviso
 - È necessario definire un opportuno protocollo di accesso (MAC)

QUESTIONE

Le reti si suddividono anche in :

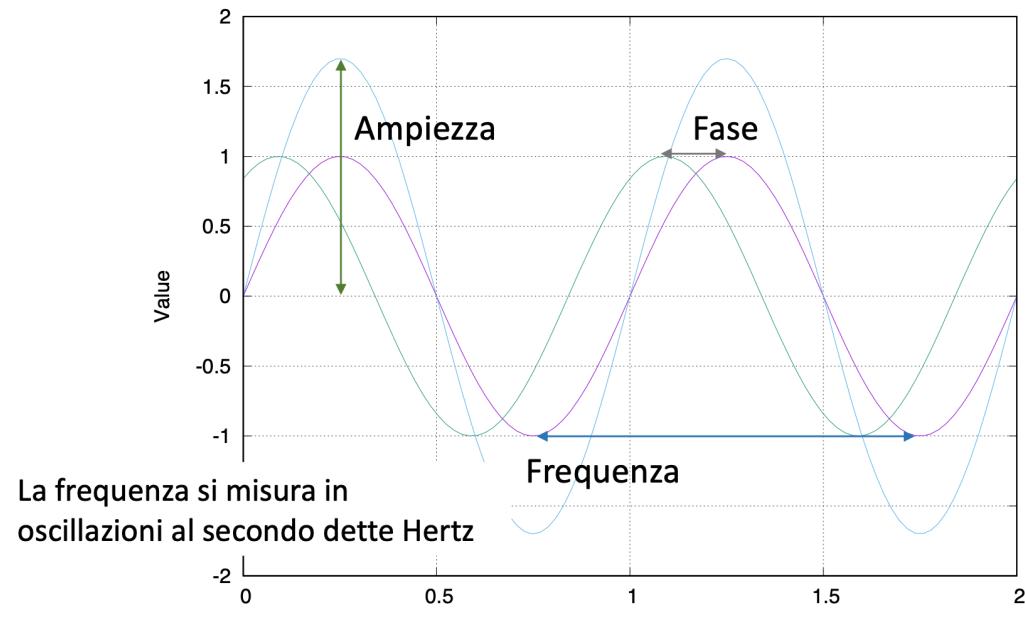
- **Reti di accesso**
- **Rete di transito o trasporto (backbone)**

Informazioni segnali e digitalizzazione

Nel mondo reale i segnali sono analogici, anche se per trasmetterli tramite un canale talvolta è necessario ridurli in formato digitale.

Esempio sinusoidale

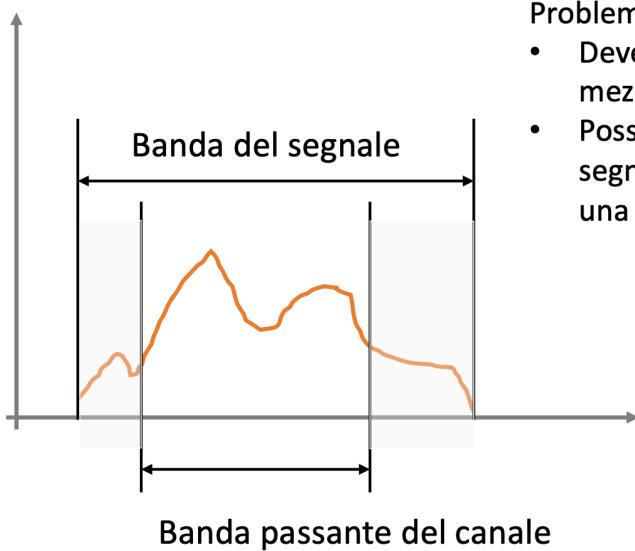
$$A \sin(2\pi ft + \phi)$$



Le tre grandezze fondamentali sono sufficienti per descrivere questo segnale, e focalizzandosi solo su queste tre è facile semplificare il segnale, anche in caso di segnali complessi.

Larghezza di banda

La larghezza di banda è definita come la differenza tra frequenza massima e frequenza minima di un segnale: $B = f_M - f_m$ in generale identifica la complessità, maggiore lo spettro maggiore complessità segnale.



Problema:

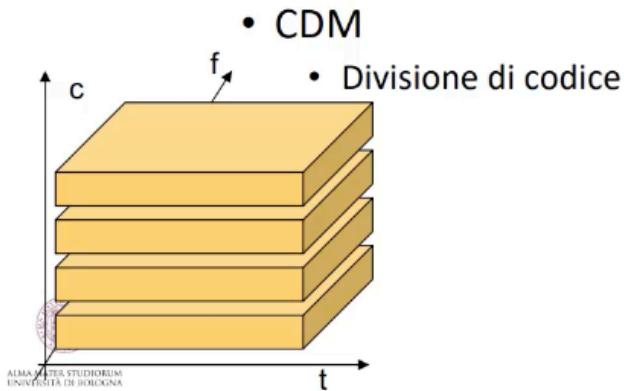
- Deve esserci una coerenza fra segnale e mezzo utilizzato per trasmetterlo
- Possiamo chiederci se l'usabilità del segnale possa essere compatibile con una parziale perdita di frequenze?



Nasce però un problema nel caso in cui la larghezza di banda del canale è minore del segnale, e in questo caso bisogna determinare se la perdita di segnale è accettabile o meno, ovviamente facendo scelte ponderate in base al contesto di utilizzo (nella linea telefonica non è importante che si senta tutto il suono, ma solo che arrivi il messaggio e che sia comprensibile).

Come fare passare più canali nello stesso mezzo trasmissivo?

Ci sono vari modi:

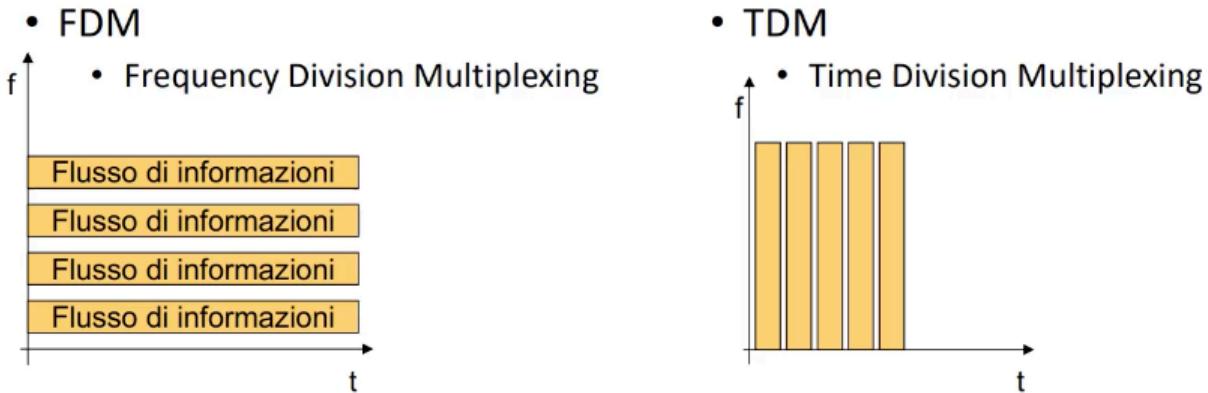


CDM code division multiplexing

Vengono divise in base ad operazioni difficili fatte su bit

La costante per ognuna di queste caratteristiche è il costo computazionale che serve per determinare di quale canale è l'informazione inviata.

Con lo sviluppo sempre maggiore delle CPU e dei computer in generale è diventato sempre più conveniente utilizzare TDM o CDM. In questo corso vedremo particolarmente TDM.



FDM Multiplazione a divisione di frequenza (frequency division multiplexing)

In cui tanti segnali coesistono contemporaneamente nello stesso mezzo trasmissivo ma su frequenze diverse, in modo tale da fare sì che si possano scomporre all'arrivo.

TDM time division multiplexing

Per un certo tempo una informazione di un flusso e per un altro tempo un altro flusso. Possibile grazie alle nuove tecnologie, usando anche buffer e altre tecnologie

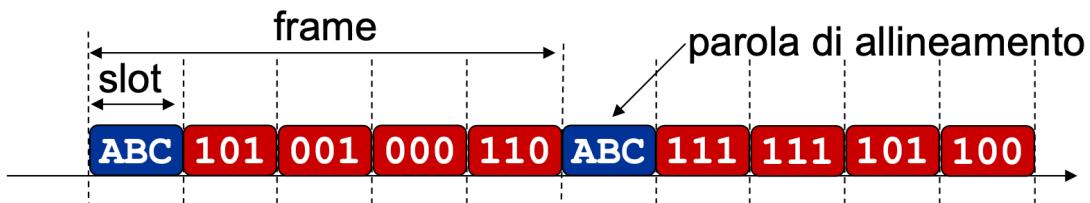
multiplazione a divisione di tempo	
slotted	unslotted
framed	unframed
assegnazione statica della banda	assegnazione dinamica della banda

TDM slotted

- L'asse dei tempi è suddiviso in intervalli di durata prefissata (slot)
- le unità informative hanno tutte la stessa lunghezza commisurata al singolo slot



Framed

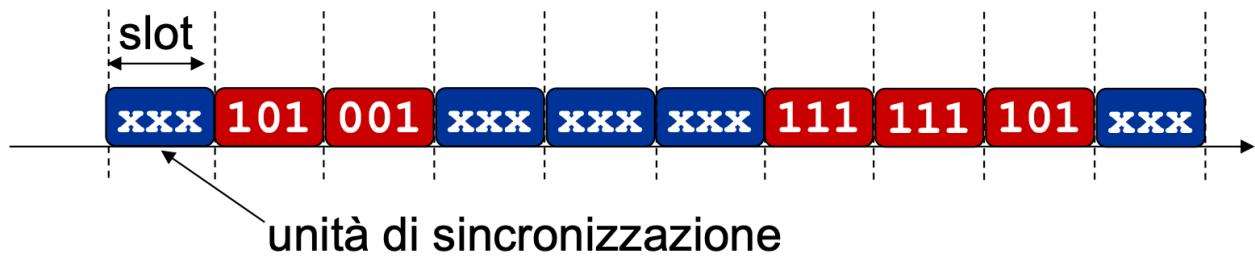


gli slot vengono strutturati in trame (frame)

Si sincronizza la trama

- non è necessaria la sincronizzazione a livello del singolo slot)

Unframed



In uno schema di multiplazione TDM slotted unframed

- gli slot si susseguono senza una struttura predefinita
- occorre un sistema di sincronizzazione di slot

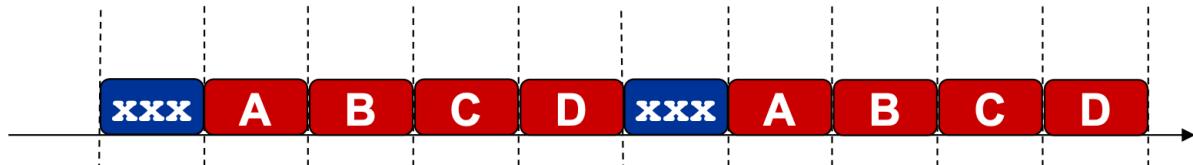
TDM unslotted

- l'asse dei tempi non è suddiviso a priori
- si possono adottare unità informative di lunghezza variabile
- è necessario un sistema esplicito di delimitazione delle unità informative



Assegnazione della banda

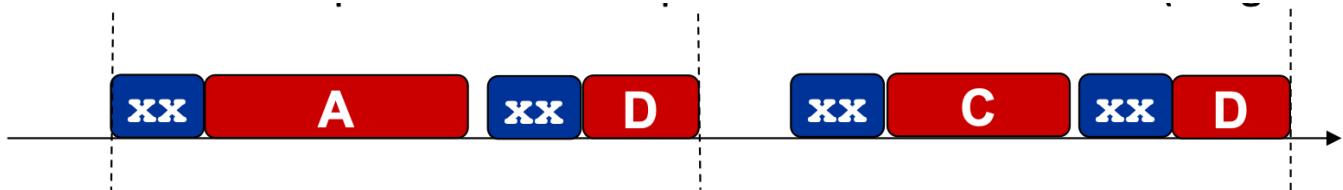
Statica



Flusso informativo = banda dedicata (bit/sec)

- La banda non può cambiare a comunicazione in corso
- La richiesta complessiva di banda è ben controllabile se si controlla il numero di flussi attivi

Dinamica



Molti flussi informativi condividono liberamente la banda in base alle necessità

- La banda può cambiare a comunicazione in corso
- La richiesta complessiva di banda può diventare intollerabile (congestione)
La direzione di ora è dinamica, dato che si vuole usare lo stesso collegamento per inviare più tipologie diverse di messaggi

PCM pulse code modulation

Tecnologia utilizzata nella rete telefonica

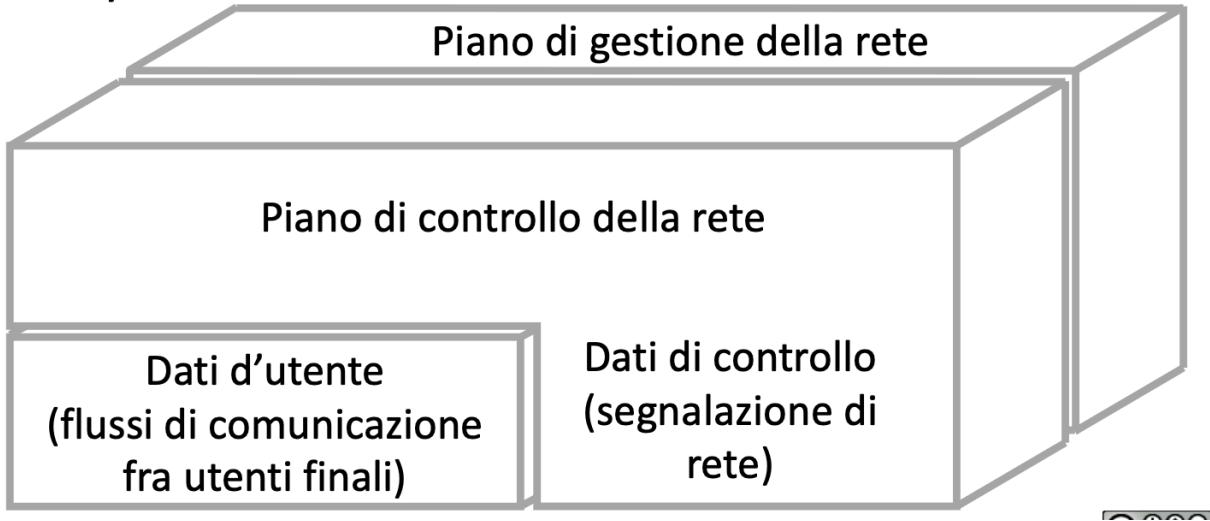
Un canale telefonico Banda 300 Hz - 4 KHz a campionamento, ogni 125 us (8000 campioni/sec) e quantizzazione a 8 bit

- Risultano $8 \times 8000 = 64000$ bit/s

funzioni di rete e commutazione

La rete ha l'obiettivo di consentire la comunicazione, tra un insieme di terminali e con un livello accettabile di QoS. La rete moderna è un **Ipercubo**, e l'utilizzo che ne fa l'utente finale è solo una faccia rispetto al totale del solido. Ci sono altre informazioni che servono alla rete stessa dato è un **sistema distribuito**.

Tra i vari compiti che devono essere svolti sono garantire il corretto comportamento della rete, gestire riconfigurazioni e malfunzionamenti, gestire gli aspetti economici legati alla rete (tariffazione ecc.).



funzioni di rete

Funzioni di rete

Trasmissione:

Trasferimento fisico del segnale da punto a punto o da un punto a molti punti

Commutazione

Instradamento delle informazioni all'interno della rete (nodi e collegamenti) al fine di permettere la comunicazioni fra punti terminali per soddisfare le richieste degli utenti.

Segnalazione

Scambio delle informazioni necessarie per la gestione della comunicazione e della rete stessa

- Segnalazione utente e rete
- Segnalazione interna alla rete

Gestione

Tutto ciò che concerne il mantenimento delle funzioni della rete; riconfigurazione di fronte ai guasti o cambiamenti strutturali, allacciamento di nuovi utenti ecc.

Multiplazione

È il meccanismo o tecnica di trasmissione per cui più canali trasmissivi in ingresso condividono la stessa **capacità trasmissiva** disponibile in uscita ovvero combinando più **segnali analogici** in un solo segnale (detto *multiplato*) in uscita su uno stesso collegamento fisico.

Tecniche di commutazione

Commutazione di circuito (rete telefonica)

Informazione analogica o digitale, La rete crea un canale di comunicazione dedicato fra due terminali che vogliono colloquiare, Il circuito è riservato ad uso esclusivo del chiamante e chiamato, esiste un ritardo iniziale per instaurare il circuito (call set-up-time). Dopodiché è garantita la *trasparenza temporale* (minimizzazione ed equalizzazione del ritardo).

Commutazione di messaggio (rete telegrafica) o di pacchetto (reti di calcolatori)

Fasi della comunicazione a Comutazione di circuito

Per riuscire a comunicare è necessario dividere il processo di comunicazione in 3 fasi:

Instaurazione del circuito

Prima di comunicare i due dispositivi creano circuito (circuito *end-to-end*), ovviamente richiede un'opportuna segnalazione.

Dialogo → Scambio di informazioni

Disconnessione del circuito → Il circuito deve essere liberato per altri

Pacchetto vs messaggio → Il messaggio è un'informazione completa, e il pacchetto è solo un componente

Contro

- se le sorgenti hanno un basso tasso di attività il circuito è sottoutilizzato,
- la capacità del canale è fissata dalla capacità del circuito e non si può variare.

Reti a pacchetto

Si utilizza il modello ISO-OSI

ISO-OSI (open system interconnection)

Le definizioni contenute nell'OSI coinvolgono tre livelli di astrazione:

Modello di riferimento:

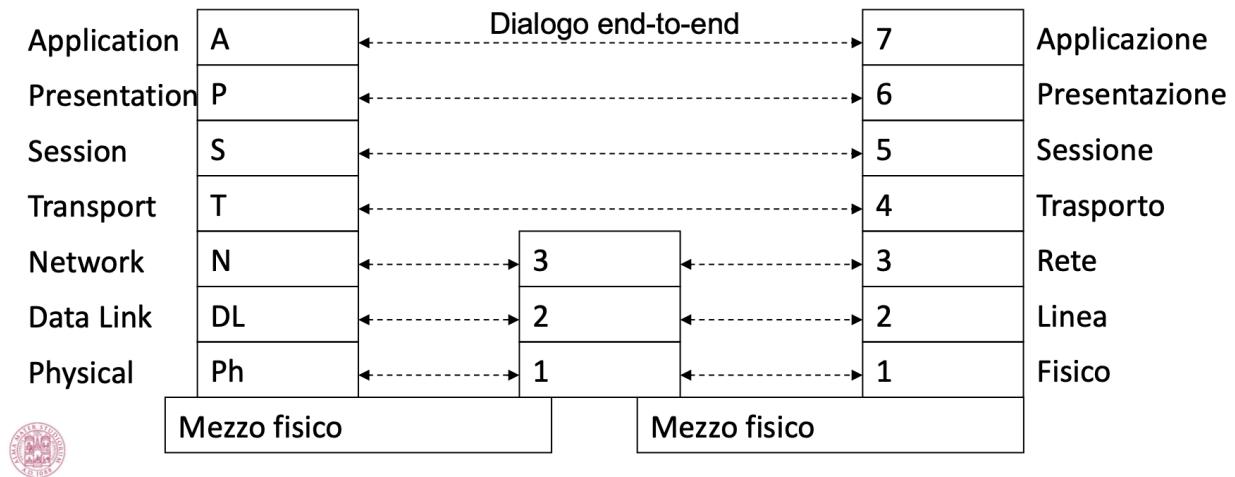
- schema concettuale
- numero degli strati coinvolti
- definizione generale delle funzioni degli strati e delle relazioni fra di essi.

Definizione dei servizi:

- definizione astratta di ciò che viene fornito da uno strato.

Specifiche di protocolli ed interfacce:

- descrizione di come viene fornito un servizio da uno strato.



1, 2, 3 sono detti lower o network oriented layers

5, 6, 7 sono detti upper o application oriented layers

4 funge da raccordo fra gli upper e lower layers, permette l'evoluzione indipendente dei livelli sopra e sotto

La differenza tra iso-osi e tcp/ip è che il tcp/ip è sostanzialmente l'implementazione del modello teorico iso-osi.

ISO-OSI	TCP/IP	ESEMPIO PROTOCOLLI
Application	Application	HTTP, TELNET, FTP, SMTP, POP, DNS, SNMP
Presentation	Application	
Session	Application	
Transport	transport	TCP , UDP
Network	Network	IP , ICMP, IGMP, ARP, RARP
Data Link	Link	
Physical	Link	ETHERNET, IEEE 802, HDLC, PPP

Spiegazione specifica dei vari livelli

• **Livello 1: Fisico**

- **Funzione:** Trasmissione dei bit grezzi sul mezzo fisico.
- **Esempi:** Cavi, connettori, segnali elettrici.

• **Livello 2: Collegamento Dati**

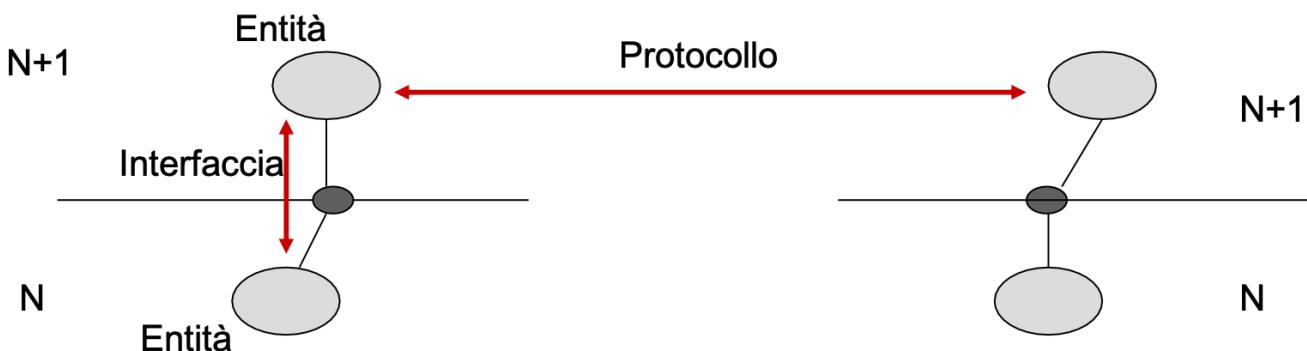
- **Funzione:** Trasferimento dati affidabile tra nodi adiacenti, framing e controllo errori/flusso locale.
- **Esempi:** Ethernet, indirizzi MAC, switch.
- **Livello 3: Rete**
 - **Funzione:** Indirizzamento logico e routing dei pacchetti tra reti diverse.
 - **Esempi:** IP (IPv4, IPv6), router, indirizzi IP.
- **Livello 4: Trasporto**
 - **Funzione:** Comunicazione end-to-end tra processi, segmentazione e controllo errori/flusso globale.
 - **Esempi:** TCP, UDP, numeri di porta.
- **Livello 5: Sessione**
 - **Funzione:** Gestione sessioni di comunicazione (inizio, mantenimento, fine).
 - **Esempi:** Spesso integrato nei livelli superiori, API di sessione.
- **Livello 6: Presentazione**
 - **Funzione:** Formattazione, compressione e crittografia/decrittografia dei dati per l'applicazione.
 - **Esempi:** Formati dati (JPEG, MPEG), crittografia (TLS).
- **Livello 7: Applicazione**
 - **Funzione:** Fornisce servizi di rete direttamente alle applicazioni utente.
 - **Esempi:** HTTP, FTP, SMTP, browser web, client email.

Vengono poi definiti 3 elementi:

Entità ogni elemento attivo in uno strato, identificata da un nome simbolico (title). Nello strato N-esimo possono essere attive una o più entità

Protocollo regole di dialogo fra entità dello stesso livello

Interfaccia regole di dialogo fra entità di livelli adiacenti



Terminologia

- **N-Protocol Data Unit (PDU)**: dati trasferiti fra entità di strato N
- **N-Service Data Unit (SDU)**: dati passati allo strato N dallo strato N+1
- **N-Service Access Point (SAP)**: indirizzo di identificazione del flusso dati fra N+1 ed N
- **N-Protocol Control Information (PCI)**: informazioni aggiuntive per il controllo del dialogo a livello N
- **Encapsulation**: N-PDU = N-PCI+ N-SDU

Modalità di servizio

Connection oriented → instaurare connessione prima di comunicare

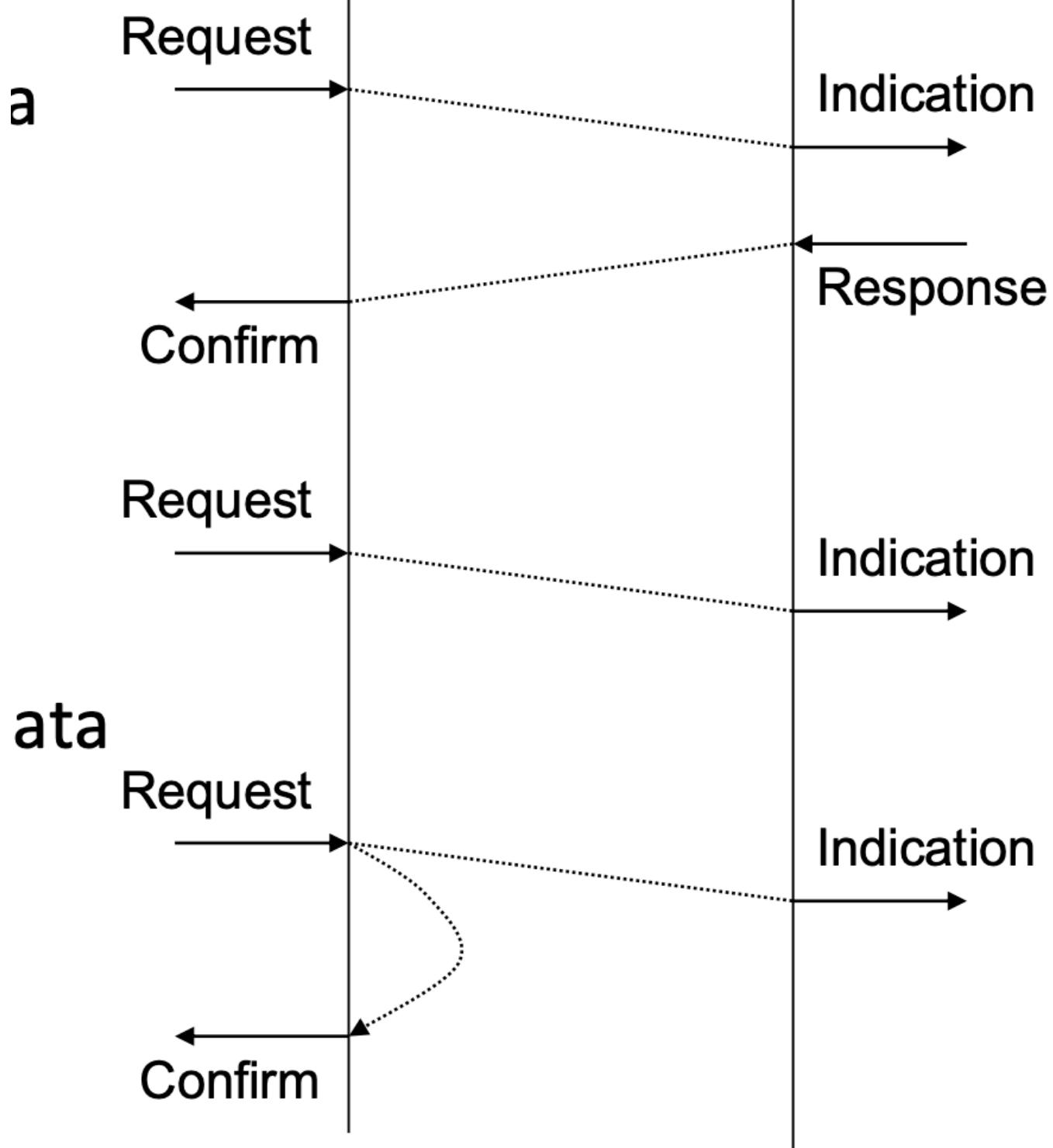
Connectionless → senza prima instaurare connessione

Modalità di dialogo

Confermato → Prevede esplicita conferma da parte del destinatario

Non confermato → Non prevede alcuna conferma

Parzialmente confermato → La richiesta viene confermata dal service-provider



Segmentazione e riassemblamento

Vengono effettuati per permettere il passaggio del messaggio se necessaria una conformazione.

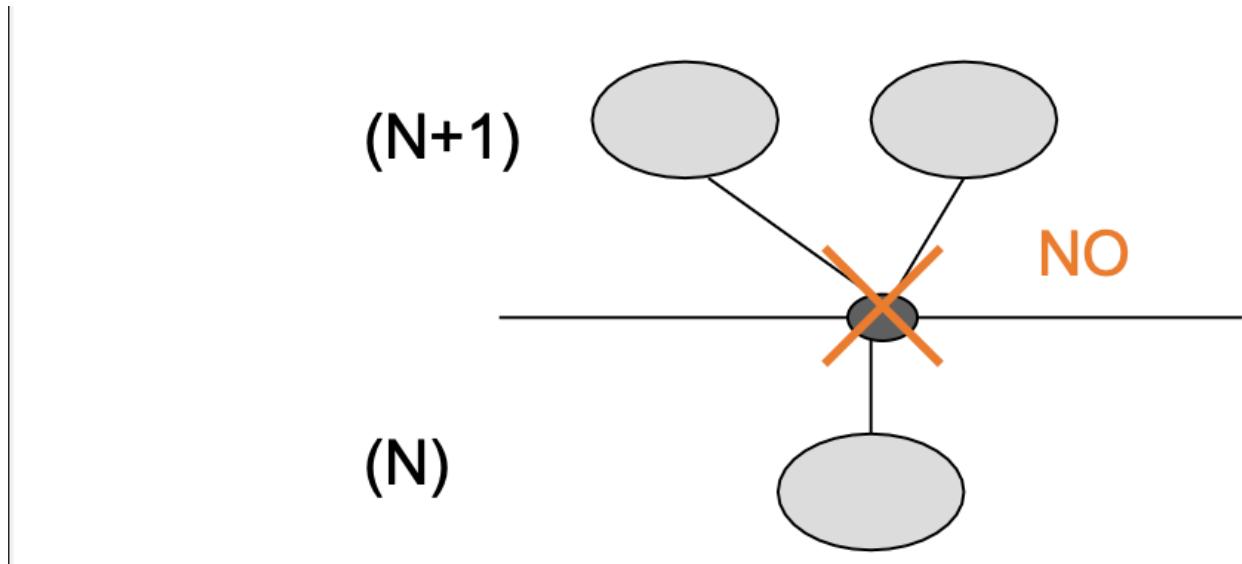
Multiplazione e splitting

multiplazione → più connessioni di livello N+1 o N-1 mappate in un livello N

splitting → spartire le connessioni in 2 livelli

Uso dei SAP

- Un'entità di strato N può servire più (N)-SAP contemporaneamente.
- Un utilizzatore di strato N può servirsi di più (N)-SAP contemporaneamente
- Non è permesso connettere più (N)-user allo stesso (N)-SAP
 - Si genererebbe ambiguità sulla provenienza/destinazione dei dati
 - Ad ogni indirizzo deve essere univocamente associato un nome



Internet

Principio originale → una rete assunta fin dal principio come inaffidabile e quindi non centralizzata in modo da funzionare anche in pezzi

Arpanet

1969 → ARPA finanzia la sperimentazione di rete di calcolatori mettendo in comunicazione:

- UCLA (University of California at Los Angeles)
- Stanford Research Center
- UCSB (University of California at Santa Barbara)
- Università dello Utah

Timeline

- 1973 prima connessione internazionale
- 1982 TCP/IP
- 1986 NSFNET
- 1990 ARPANET no more
- 1992 world wide web
- 1994 primi servizi commerciali

Enti di gestione di internet

Non c'è gestione di internet ma ci sono enti che fanno coordinamento.

Oggi Internet ha la Internet Advisory Board che è formata da due gruppi

- Internet engineering task force → coordina le attività di ingegnerizzazione
- Internet research task force → coordina attività di ricerca

RFC

Insieme di definizioni di protocolli, conservate in documenti di pubblico dominio

InterNIC (Network Information Center)

NSF (national science foundation) fonda internNIC → registrazione di nuovi domini, manutenzione di database, servizi informativi sulla rete

IANA Internet Assigned Number Authority

Mantiene i database dei numeri che hanno significati convenzionali nei protocolli di Internet

Indirizzamento

come fa il chiamante a specificare il chiamato? → diversi modi di indirizzare:

- esseri umani → nomi simbolici
- nodi di comunicazione → indirizzi
- sistemi di sicurezza → identità
- applicazioni → identificativi

In ambito di internet abbiamo:

- URN → identifica il nome
- URI → **identificativo** di una certa risorsa

- URL → indirizzo necessario per **localizzare** tale risorsa

Indirizzo globale e locale

- Globale → valido per tutta la rete, univoco, deve essere assegnato in modo che non sia doppio
- Locale → valido per una porzione della rete, non univoco (può essere) assegnato localmente con procedura non specifica.

URL

Il calcolatore in sé va identificato univocamente su internet, parte dell'indirizzo deve avere significato unico e universale: IP

un URL può contenere anche un nome utente e una password:

`http://admin:password@foobar.com/`

//possono essere inviati anche dati assieme alla richiesta:

`https://twitter.com/Twitter/status/nomevar1=valore1&nomevar2=valore2`

caratteri speciali

- # tutti i caratteri dopo sono ignorati
- & fine url inizio variabili

`http://foobar.com/?var=hello % world`

world viene ignorato ed è un problema se si vuole passare come parametro e si usa la **Codifica URL**
una codifica non dannosa

`http://foobar.com/?var=hello%26%23+mondo`
%codice esadecimale del carattere

COOKIES

`https` -> statefull, identificano il browser nei confronti di un server. Vengono impostati nel campo di risposta http "Set-Cookie", o anche impostati lato client con JS.

Sono composti da: nome, valore e alcune meta-informationi tipo: server che invia, data scadenza.

Rete Locale

LAN → un calcolatore si connette alla rete prima collegandosi ad una LAN (local area network) ed usano un canale di trasmissione ricezione condiviso tra tutti i calcolatori della LAN.

Il canale condiviso implica una comunicazione in broadcast, che rende le comunicazioni confusionarie ed incomprensibili. Per ovviare questo problema si utilizzano degli indirizzi di LAN chiamati :

MAC address

composti da 48 bit e cablati nella scheda di rete del calcolatore, essi sono univoci a livello mondiale e grazie a loro è possibile specificare

- singolo destinatario
- indirizzo di gruppo
- invio a tutte le stazioni

Indirizzo IP

lunghezza fissa 32 bit, sequenza di 4 numeri decimali da 0 a 255, L'indirizzo identifica i punti di interconnessione di un host con la rete

- Non identifica un host individuale, ma una delle sue interfacce di rete

Numero di porta

indirizzo a 16 bit, locale al singolo calcolatore e ripetute su tutti i calcolatori. condiviso tra tutti i protocolli di trasporto.

Implementazione dei servizi e modelli di interazione

Nomenclatura

Server -> *Rende disponibile* un servizio, mediante un'interfaccia standard (protocollo)

Client -> È in grado di *utilizzare* i servizi messi a disposizione da un server.

Apertura ->

- ricevere una connessione eseguendo una *apertura passiva* (attende richiesta).
- *Attiva* fatta da client

Vari modelli

Uno a molti

Molti (client) a uno (server) (many to one):

- Sincrono e bloccante (client attende server)
- **Binding** dinamico, ogni richiesta il client sceglie se collegarsi o no
- server può offrire diversi tipi di prestazione

Peer to peer

tutti gli host di rete sono equivalenti e fungono alternativamente sia da client che da server verso altri nodi. è Auto-scalabile, cioè nuovi peer server possono essere aggiunti.

Ricerca destinazione

il client deve conoscere l'indirizzo IP e numero di porta del server destinazione. Le informazioni relative a fare ciò sono contenute nell'URL.

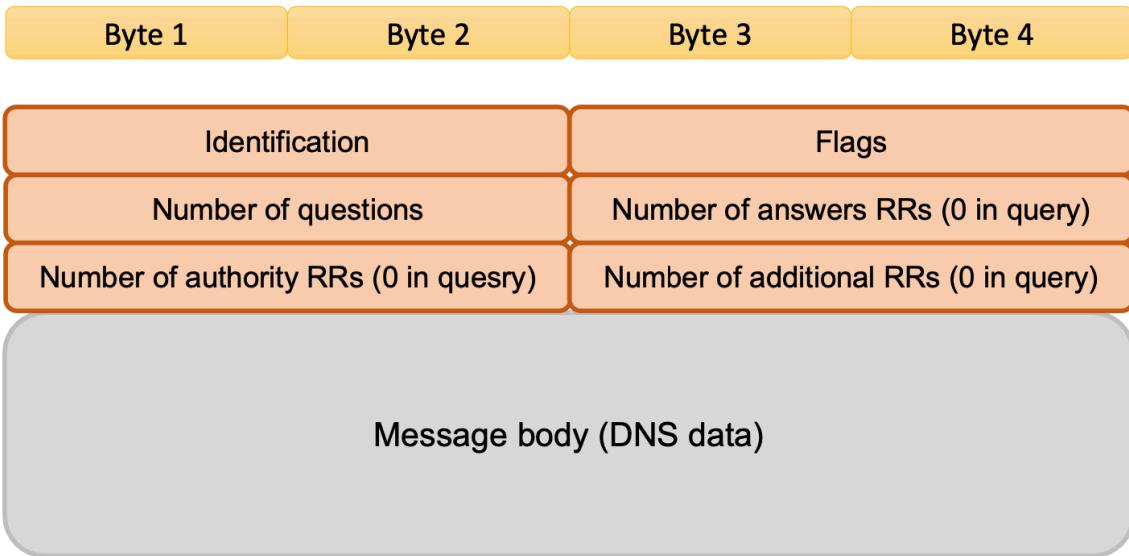
- protocollo applicativo
- eventuale numero di porta non standard
- il numero IP o nome del server

Domain Name System

- i calcolatori devono usare numeri IP da 32 bit nelle PCI delle PDU di livello 3
- deve esistere un metodo per tradurre un nome simbolico in un numero IP
- il metodo in questione si basa su una applicazione di rete chiamata Domain Name System (DNS)
Esistono vari metodi per soddisfare una richiesta a un dns:
ricorsiva: tu fai richiesta e il name resolver si occupa di chiedere ricorsivamente ai suoi sottodomini e poi ritorna la risposta(ovvero l'ip) esempio di richiesta a cs.edu.com edu.com di occupa di trovare cs.edu.com

Iterativa: iterativamente ti ritornano il successivo a cui chiedere fino a che non si arriva a trovare quello specifico

Nella realtà si usa un mix di entrambi i metodi.



PDU del dns, contiene un ID specifico che serve per matchare request e response

Flags di DNS

QR -> 0 query, 1 response

rCode -> 0 nessun errore, 1 errore di formato, 2 errore nel server ecc..

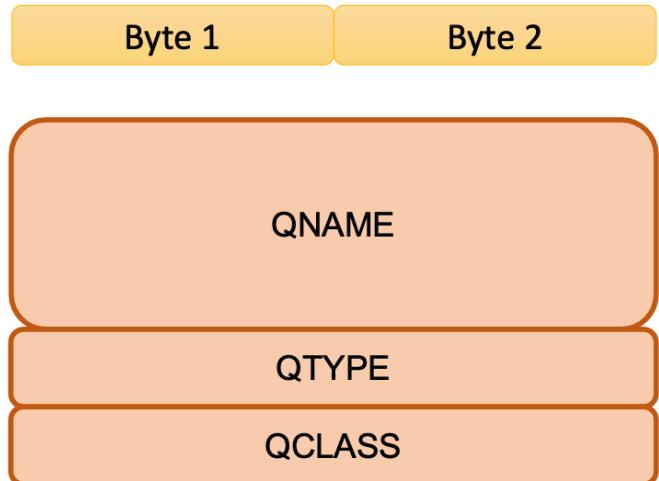
PDU DNS

il protocollo ha 2 tipi di PDU:

- query → due sezioni:
 - HEADER (PCI)
 - QUESTION (domande al server)
- response → suddiviso in 5 sezioni:
 - HEADER
 - QUESTION (copia delle domande della query)
 - record
 - answer records
 - authoritative records
 - additional records

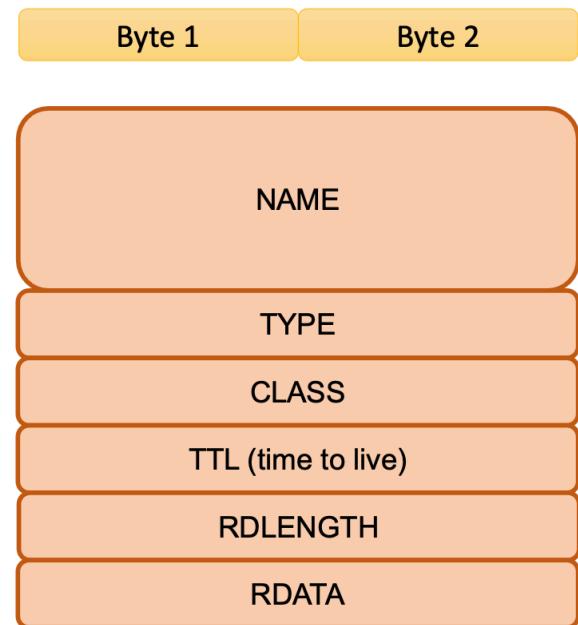
Formato domanda

- **QNAME**
 - Un nome di dominio per cui si effettua la richiesta
- **QTYPE**
 - Tipo della richiesta (codificato in due byte)
- **QCLASS**
 - Classe della domanda



Formato risposta

- **NAME**
 - Un nome di dominio per cui si effettua la richiesta
- **TYPE**
 - Tipo della risposta (significato del contenuto in RDATA)
- **TTL**
 - Durata in secondi del tempo per il quale la risposta può essere mantenuta in memoria



PDU

Protocol data unit

la PDU è l'unità di dati che viene scambiata tra due entità di pari livello in un modello di comunicazione, wrappa sdu

SDU

Service Data Unit

Il messaggio effettivo

PCI

Protocol control information (header o trailer)

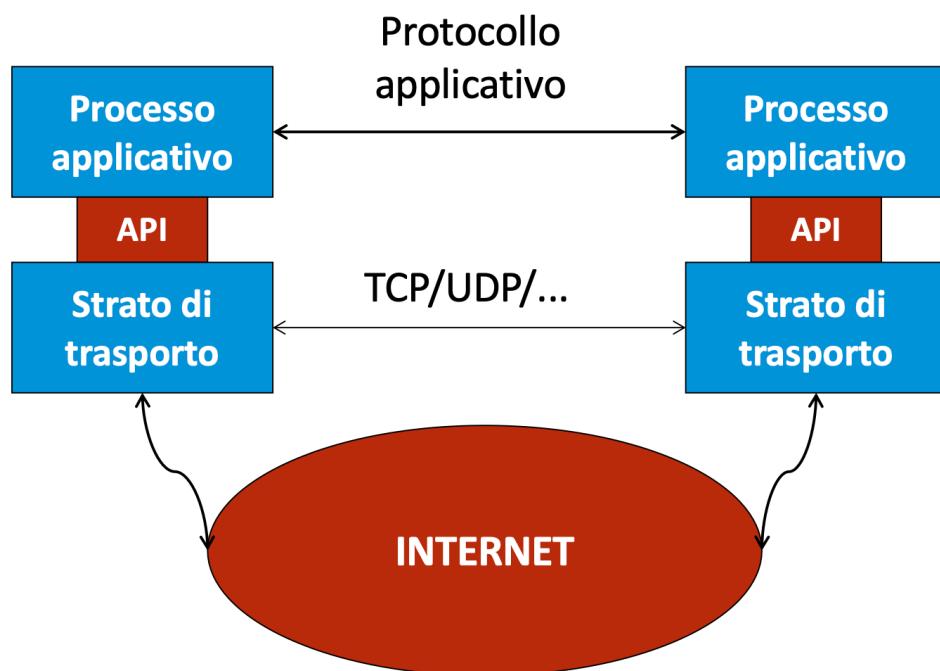
l'informazione di controllo aggiunta al pacchetto PDU da un livello per garantire la corretta trasmissione e ricezione dei dati.

Protocolli applicativi

Sono protocolli usati da applicazioni per scambiare informazioni e devono definire:

- Le tipologie di messaggi che vengono scambiati
- La sintassi dei vari tipi di messaggio
- Il significato (semantica) delle informazioni contenute nei messaggi
- Le regole che governano lo scambio dei messaggi

architettura generale protocolli applicativi



HTTP

HTTP è il protocollo usato dal World Wide Web, ha reso popolare Internet al di fuori degli ambienti accademici. Le sue versioni principali sono:

- **HTTP/1.0** (1996, obsoleto)
- **HTTP/1.1** (1997)
- **HTTP/2** (2015)
- **HTTP/3** (2022)

I **client HTTP** includono browser ma anche applicazioni che usano HTTP (es. API REST).

I **server HTTP** includono Apache, Nginx, IIS, Caddy, ecc.

Formato messaggi

- le richieste HTTP sono **stateless**, può diventare **stateful** tramite i **cookie**.
- Esistono due tipi di messaggi: **richiesta** e **risposta**, ognuno con due parti:
 - **Intestazione**, composta da righe di testo terminate da **CR+LF** (carriage return + line feed).
 - Le **richieste** iniziano con una *riga di richiesta* seguita da righe di intestazione.
 - Le **risposte** iniziano con una *riga di stato* seguita da righe di intestazione.
 - **Corpo**, contiene i dati (es. HTML, immagini, ecc.).

Esempio richiesta:

Riga di richiesta	Metodo	“sp”	URL	“sp”	Versione	“CR”	“LF”																		
	Nome intestazione	“.”	“sp”	Valore	“CR”	“LF”																			
...																									
Righe di intestazione	Nome intestazione	“.”	“sp”	Valore	“CR”	“LF”																			
Riga vuota	“CR”	“LF”																							
Corpo	<table border="1"><thead><tr><th>Metodo</th><th>Significato</th></tr></thead><tbody><tr><td>GET</td><td>Richiesta di una pagina web al server</td></tr><tr><td>HEAD</td><td>Richiesta di informazioni relative a una pagina web</td></tr><tr><td>PUT</td><td>Richiesta di memorizzazione di una pagina web</td></tr><tr><td>POST</td><td>Invio di informazioni relative a una pagina web</td></tr><tr><td>DELETE</td><td>Richiesta di cancellazione di una pagina web</td></tr><tr><td>TRACE</td><td>Richiesta di ricevere l'eco del messaggio inviato</td></tr><tr><td>CONNECT</td><td>Richiesta di connessione attraverso un proxy</td></tr><tr><td>OPTIONS</td><td>Richiesta di informazioni sulle opzioni disponibili per una pagina web</td></tr></tbody></table>							Metodo	Significato	GET	Richiesta di una pagina web al server	HEAD	Richiesta di informazioni relative a una pagina web	PUT	Richiesta di memorizzazione di una pagina web	POST	Invio di informazioni relative a una pagina web	DELETE	Richiesta di cancellazione di una pagina web	TRACE	Richiesta di ricevere l'eco del messaggio inviato	CONNECT	Richiesta di connessione attraverso un proxy	OPTIONS	Richiesta di informazioni sulle opzioni disponibili per una pagina web
Metodo	Significato																								
GET	Richiesta di una pagina web al server																								
HEAD	Richiesta di informazioni relative a una pagina web																								
PUT	Richiesta di memorizzazione di una pagina web																								
POST	Invio di informazioni relative a una pagina web																								
DELETE	Richiesta di cancellazione di una pagina web																								
TRACE	Richiesta di ricevere l'eco del messaggio inviato																								
CONNECT	Richiesta di connessione attraverso un proxy																								
OPTIONS	Richiesta di informazioni sulle opzioni disponibili per una pagina web																								

Esempio risposta:

Riga di stato	Versione	"sp"	Codice stato	"sp"	Frase	"CR"	"LF"																	
	Nome intestazione	:	"sp"		Valore	"CR"	"LF"																	
...																								
Righe di intestazione	Nome intestazione	:	"sp"		Valore	"CR"	"LF"																	
Riga vuota	"CR"	"LF"																						
Corpo	<table border="1"> <thead> <tr> <th>Codice</th> <th>Significato</th> <th>Esempio</th> </tr> </thead> <tbody> <tr> <td>1xx</td> <td>Messaggio informativo</td> <td>100 Continue</td> </tr> <tr> <td>2xx</td> <td>Richiesta del client accettata</td> <td>200 OK 202 Accepted</td> </tr> <tr> <td>3xx</td> <td>Reindirizzamento</td> <td>301 Moved Permanently 304 Not Modified</td> </tr> <tr> <td>4xx</td> <td>Errore del client</td> <td>401 Unauthorized 404 Not Found</td> </tr> <tr> <td>5xx</td> <td>Errore del server</td> <td>503 Service Unavailable</td> </tr> </tbody> </table>						Codice	Significato	Esempio	1xx	Messaggio informativo	100 Continue	2xx	Richiesta del client accettata	200 OK 202 Accepted	3xx	Reindirizzamento	301 Moved Permanently 304 Not Modified	4xx	Errore del client	401 Unauthorized 404 Not Found	5xx	Errore del server	503 Service Unavailable
Codice	Significato	Esempio																						
1xx	Messaggio informativo	100 Continue																						
2xx	Richiesta del client accettata	200 OK 202 Accepted																						
3xx	Reindirizzamento	301 Moved Permanently 304 Not Modified																						
4xx	Errore del client	401 Unauthorized 404 Not Found																						
5xx	Errore del server	503 Service Unavailable																						

La consultazione del web avviene tramite richieste di oggetti identificati da **URL** (Uniform Resource Locator).

FTP file transport protocol

Uno dei più vecchi protocolli di internet, e standard per condivisione di file prima di http.
La versione di base è intrinsecamente insicura.

Funzionamento

A livello applicativo due entità colloquiano utilizzando una sessione di dialogo. Una singola sessione può includere numerose connessioni di trasporto contemporanee

FTP usa due connessioni:

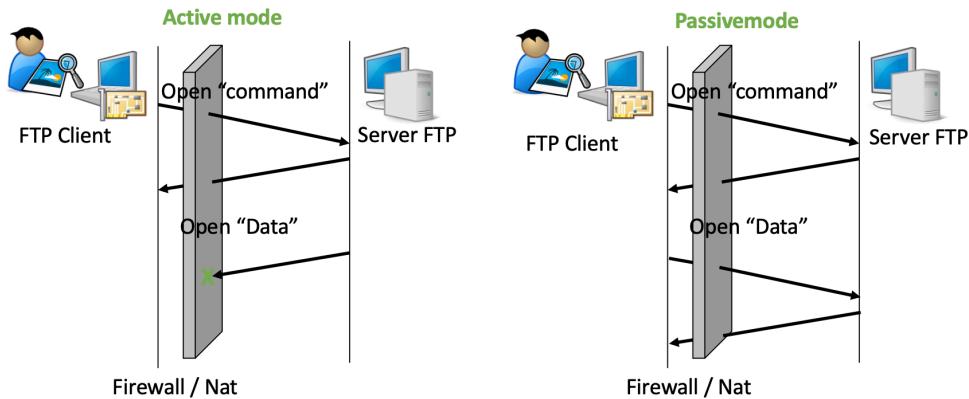
- Connessione per i comandi
- Connessione per i dati

Apertura connessione **Command**:

- **client** chiama il server sulla porta well known 21

Apertura connessione **Data**:

- **Attiva**: Server apre connessione
- **Passiva**: Client apre connessione, su porta comunicata da server



Active ha il problema che se c'è un nat il server non riesce a ricomunicare al client.

Posta elettronica:

SMTP simple mail transform protocol

Da client a server e tra server a server, e usa TCP per trasferire in maniera sicura i messaggi su porta 25.

Solitamente **trasferimento diretto**, il server contatta direttamente il server ricevente.

3 fasi:

- handshaking
- trasferimento dei messaggi
- chiusura

Vengono poi usati o POP o IMAP per leggere le mail.

POP3 Post Office Protocol

Modalità “scarica e cancella”

- I messaggi sono scaricati sull’agent locale e cancellati dal server
- Non possono essere riletti con un altro agent

Modalità: “scarica e mantieni”

- I messaggi vengono copiati sull’agent ma rimangono anche sul server
- È possibile copiare i messaggi su più client senza cancellarli dal server
- Cosa succede se cancello un messaggio su un client?
- La modifica NON viene propagata agli altri client per cui ciascun client avrà una sua visione della casella di posta non coerente e non sincronizzata con gli altri

IMAP

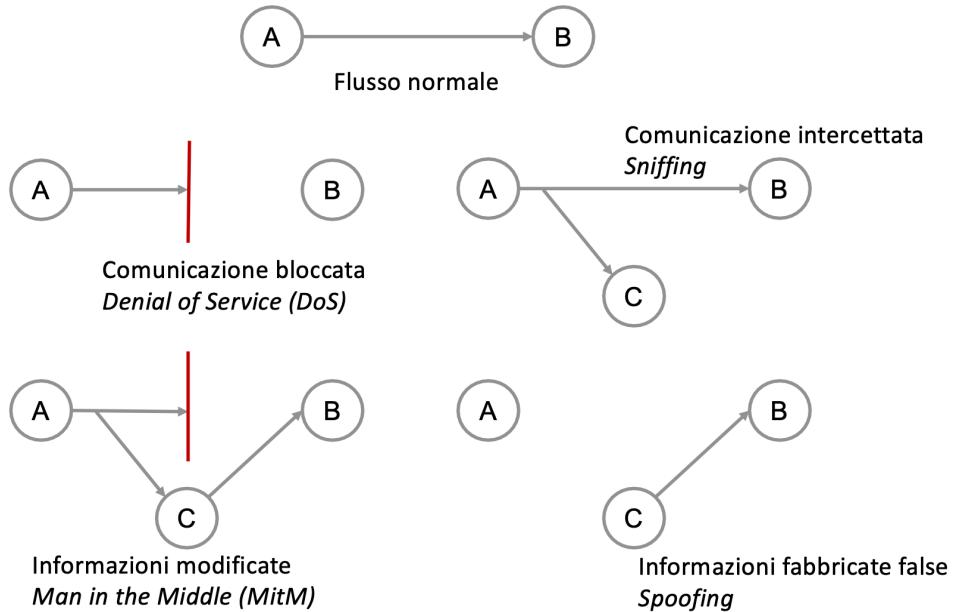
Mantiene tutti i messaggi solo nel server

- Organizzazione in **cartelle**
- **IMAP** conserva lo stato dell'utente tra le varie sessioni

Multipurpose Internet Mail Extensions (MIME): estensioni di messaggi di posta multimediali, RFC 2045, 2056 con alcune righe aggiuntive nell'intestazione per dichiarare il tipo di contenuto MIME.

HTTPS

Riassunto di tutti i problemi che possono accadere per colpa di terzi ad un flusso informativo:



Perciò per **difendere l'informazione** bisogna garantire:

- Integrità (integrity): impedire la modifica non autorizzata (accidentale o deliberata) delle informazioni
- Riservatezza (privacy): impedire l'accesso alle informazioni da parte di utenti non autorizzati
- Disponibilità (availability): garantire in qualunque momento la possibilità di usare le informazioni a chi è autorizzato
- Paternità (non-repudiability): impedire ad un utente di ripudiare un suo messaggio

Bisogna inoltre garantire **Autenticazione** (verifica identità) e in base a questa concedere determinate **autorizzazioni**.

Fattori di identificazione:

- **Qualcosa che sai** (password, PIN ecc.)

- **Qualcosa che hai** (carta d'identità, tocken, badge ecc.)
- **Qualcosa che sei** (impronta digitale, immagine del viso, timbro del parlato, ecc.)
- **Dove ti trovi** (localizzazione la posizione l'autenticazione può essere limitata al fatto di essere effettuata da una specifica o da specifiche posizioni nello spazio)
- **Quando:** limitata a specifiche finestre temporali

Tipologie attacchi

Attivi: es impersonificazione di altro utente

Passivi: es lettura non autorizzata di informazioni altrui

Strategie di sicurezza

principio di sicurezza minimo:

- proteggersi dagli attacchi passivi
- accorgersi degli attacchi attivi

Crittografia



- P: testo in chiaro (plain text), comprensibile a tutti
- C: testo cifrato (ciphertext), comprensibile solo al destinatario
- E: funzione di cifratura, capace di rendere il messaggio decifrabile solo dal destinatario
- D: funzione di decifrazione utilizzata dal destinatario per leggere il messaggio cifrato (solo il destinatario la conosce)

Algoritmi di cifratura

L'algoritmo di cifratura è la funzione matematica usata per cifrare e decifrare il messaggio

Algoritmi basati su carattere

- sostituzione:

ogni simbolo si trasforma in un altro simbolo dell'alfabeto
cambiano i simboli ma non il loro ordine nel testo

- trasposizione:

i simboli vengono permutati in base ad una permutazione stabilita
i simboli dell'alfabeto non cambiano ma cambia l'ordine in cui compaiono nel messaggio

Algoritmi basati su chiave

oltre a definire l'algoritmo, si usa una chiave per cifrare/decifrare in questo modo lo stesso algoritmo, con chiavi diverse, produce testi cifrati diversi a partire dallo stesso testo in chiaro.

Cifratura simmetrica

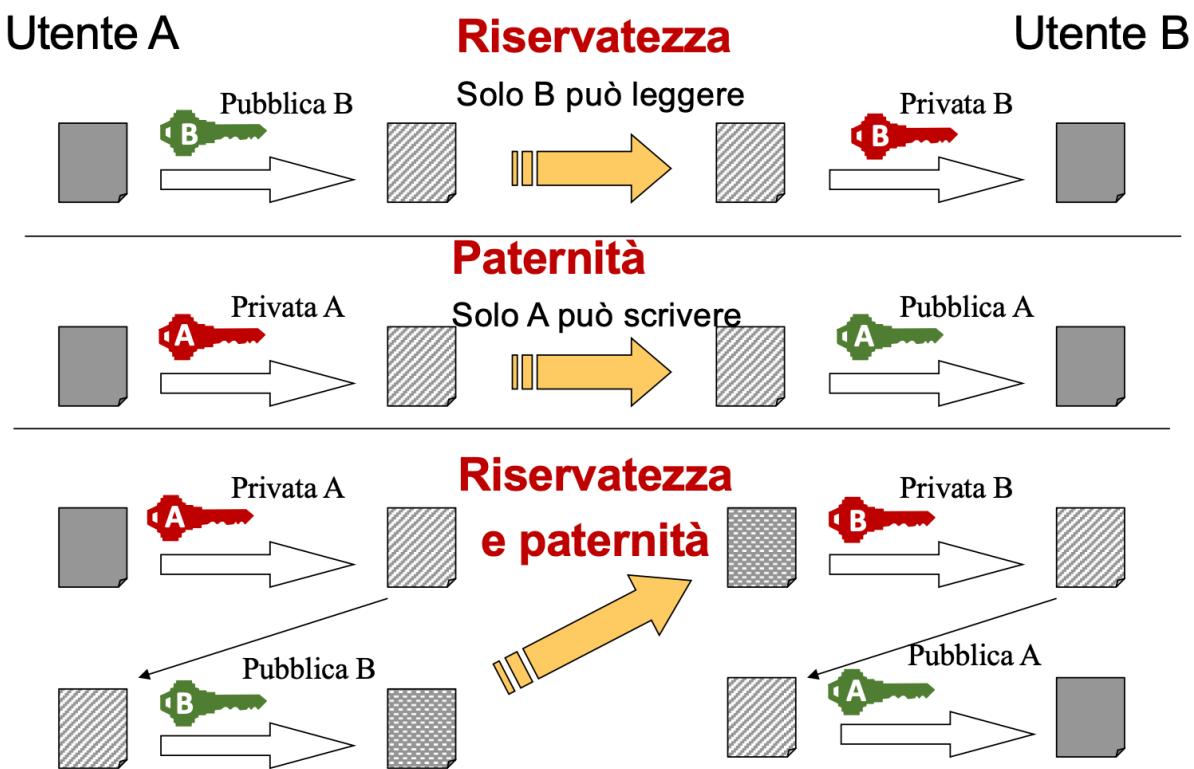
Viene usata la stessa chiave sia per criptare che per decifrare. Più efficiente ma meno sicuro

$$D(E(P, K), K) = P$$

$$E(D(C, K), K) = C$$

Cifratura asimmetrica

Viene usata K_1 per criptare e K_2 per decifrare. Una delle due è **pubblica** e l'altra **privata**.



Firma digitale

Per garantire la cifratura di M da parte di utente A devo fare:

1. A riassume tutto il documento con funzione di Hash
2. A critta con chiave privata
3. B riceve messaggio e decifra con chiave pubblica di A

4. E confronta Hash suo con Hash documento e se tutto uguale è garantita la firma di A

Vi è un grande problema, ed è il fatto che servono delle autorità che certificano la proprietà della chiave pubblica. Cioè che associno Persona A a chiave A. Perciò sono nate **Certification Authority**.

Altre volte non vi è una vera e propria certification Authority, vengono però istituite delle reti di persone fidate che si occupano di "verificare" nuovi utenti che si aggiungono alla rete.

SSH

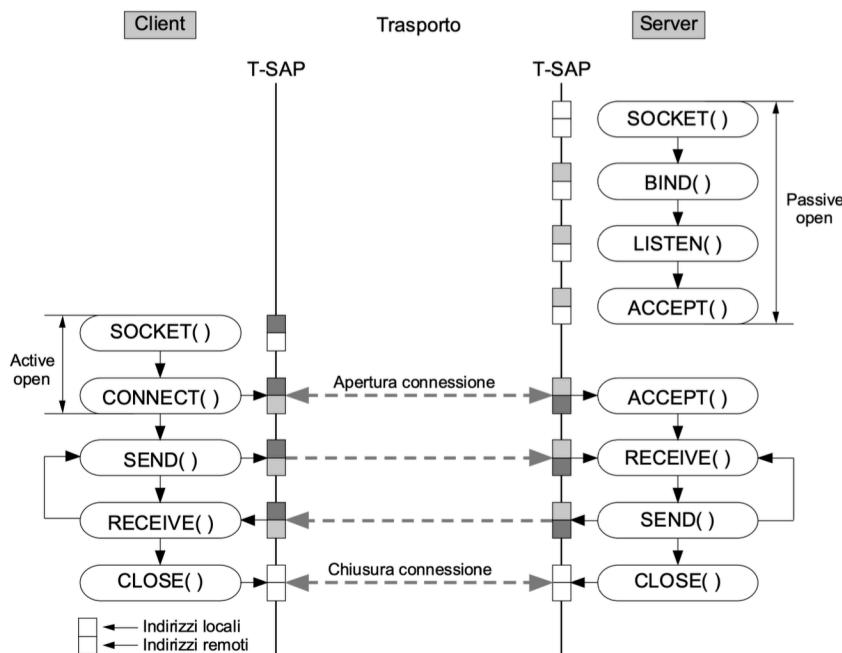
Per accesso remoto e sicuro a un calcolatore

Socket

Interfaccia che le applicazioni usano per interagire con i protocolli dello strato di trasporto, fornita sa OS tramite api e attivata da host, rappresenta l'implementazione software di **T-SAP** (transport service access point).

tramite queste api sono disponibili diverse funzioni:

Esempio di stream socket : comunicazione affidabile



server:

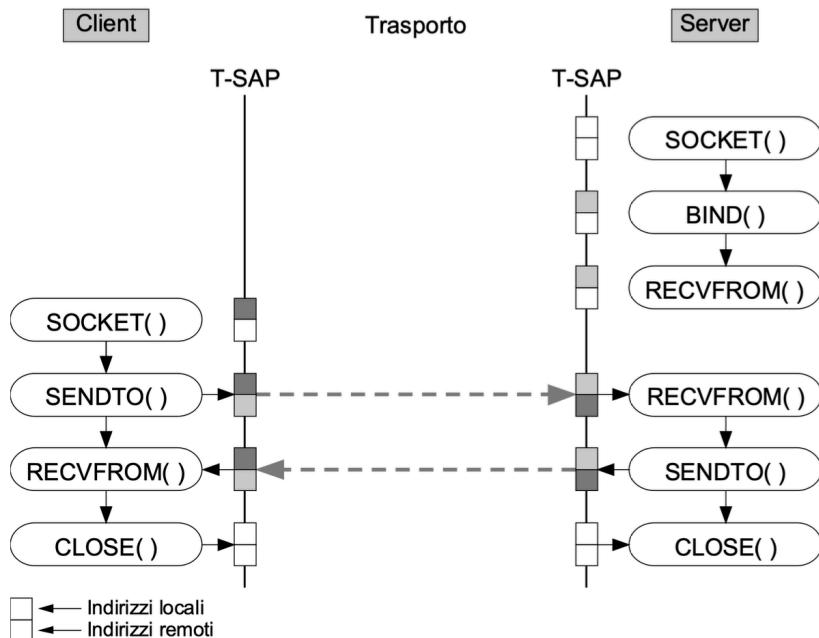
- socket: crea una nuova entità T-SAP (simile all'apertura di un file)
- bind: associa l'indirizzo (< indirizzo host : T-SAP >) alla socket creata
- listen: si mette in ascolto sulla socket creata

- accept: pone il server in attesa di accettare una richiesta da un client, a valle della quale crea un processo separato per gestirla (fork) e torna in ascolto sulla socket
- send/receive: trasmette/riceve dati sulla connessione stabilita
- close: chiude la connessione e rilascia l'indirizzo della socket

client:

- socket: crea una nuova entità T-SAP
- connect: blocca il processo client e tenta di aprire una connessione verso il server; sblocca il client a connessione instaurata
- send/receive: trasmette/riceve dati sulla connessione
- close: chiude la connessione e rilascia l'indirizzo della socket

Esempio di datagram socket: comunicazione non affidabile



Server

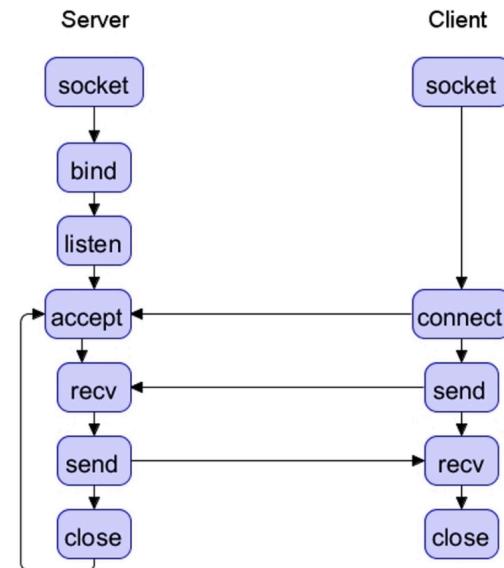
- socket: crea una nuova entità T-SAP (simile all'apertura di un file)
- bind: associa l'indirizzo (< indirizzo host : T-SAP >) alla socket creata
- sendto/recvfrom: trasmette/riceve dati a/da una socket remota specificata
- close: chiude la comunicazione e rilascia l'indirizzo della socket

Client

- socket: crea una nuova entità T-SAP
- sendto/recvfrom: trasmette/riceve dati a/da una socket remota specificata
- close: chiude la comunicazione e rilascia l'indirizzo della socket

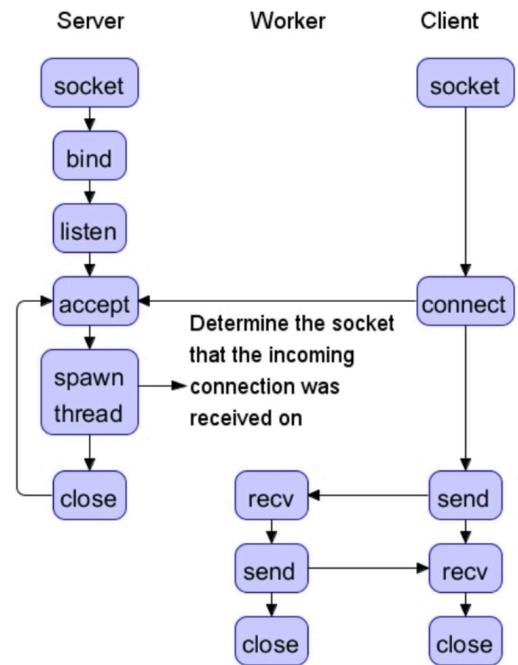
Server iterativo

- Un ciclo infinito permette al server di rispondere a più richieste di connessione successive, ma in sequenza
- Una nuova connessione non viene servita finché non termina il servizio di quella eventualmente in corso e di altre eventualmente già in attesa



Server concorrente

- Un ciclo infinito permette al server di rispondere a più richieste di connessione successive, stavolta in parallelo
- Si genera un processo o un thread separato che gestisce ogni nuova connessione, tornando immediatamente in ascolto di eventuali altre richieste



VoIP voice over ip

protocollo che serve per effettuare chiamate telefoniche tramite Internet. Maggiore è il bitrate e migliore è la qualità del segnale.

Vengono usate due tecnologie per implementarlo:

IP trunking → tecnologia IP su collegamenti della rete di trasporto, non ha impatto sulla rete di accesso dell'utente ma può avere impatto sulla tariffazione.

Telefonia IP → tecnologia IP per la telefonia ha impatto sulla rete di accesso e anche sulla rete di trasporto

Entrambe usano un PBX (Private Branch Exchange Voice over IP). Un centralino che si occupa di tradurre da IP a rete telefonica.

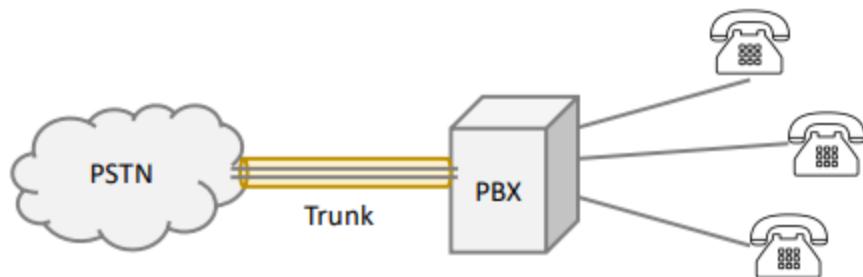
Telefonia tradizionale :

PSTN → public switched telephone network

PBX → private branch exchange

telephone line → collegamento alla rete di un terminale telefonico

trunk → collegamento che trasporta numerose linee telefoniche in parallelo.



SIP session initiation protocol

SIP gestisce la "gestione" della chiamata: avvia, mantiene e termina la sessione di comunicazione.

Gli URI SIP

- Definiscono nominalmente un utente (Naming):
`sip:user:password@host:port;uri-parameters?headers`
- Forniscono le informazioni per contattare un utente (nome host o l'indirizzo IP, numero di porta, protocollo di trasporto)
- Possono portare parametri addizionali
- Possono identificare servizi
- Possono richiedere una comunicazione sicura, URI 'sips'

Richieste

- request → richieste da UA utente a UA server

- response → risposte da server a client

Formato di un generico messaggio:

- start-line → Request-Line / Status-Line

- headers →
- CRLF → carattere di a capo
- body del messaggio → contiene il contenuto effettivo del messaggio

Risposta

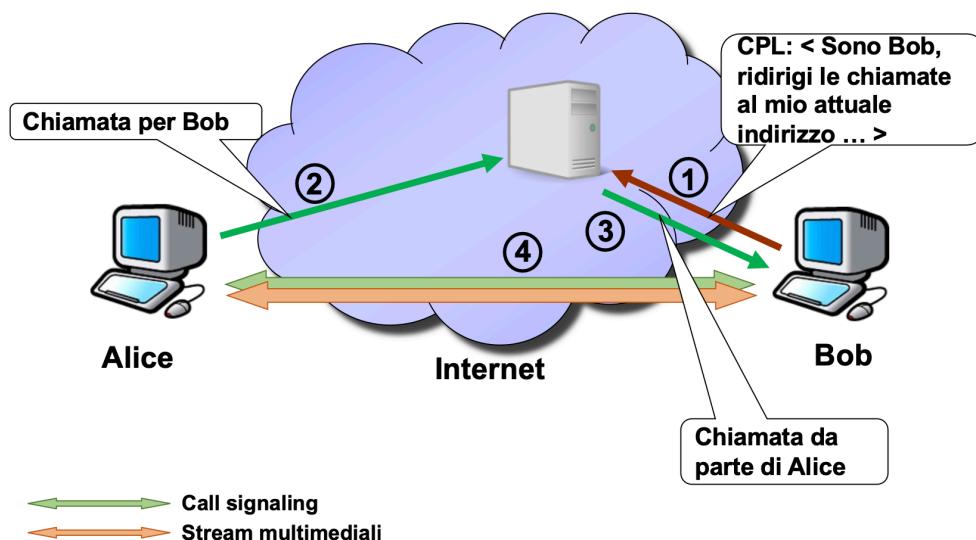
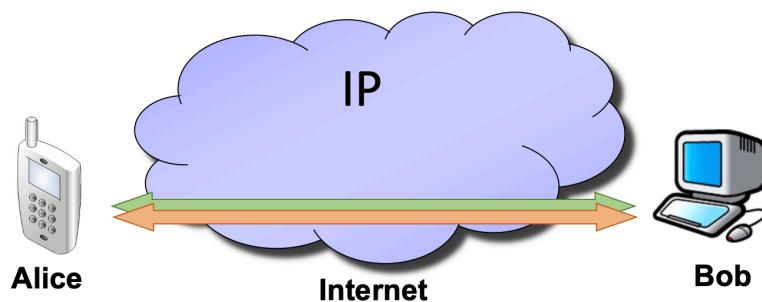
la prima cifra definisce la classe di risposta XX sono cifre numeriche proprie di una determinata azione. Classi:

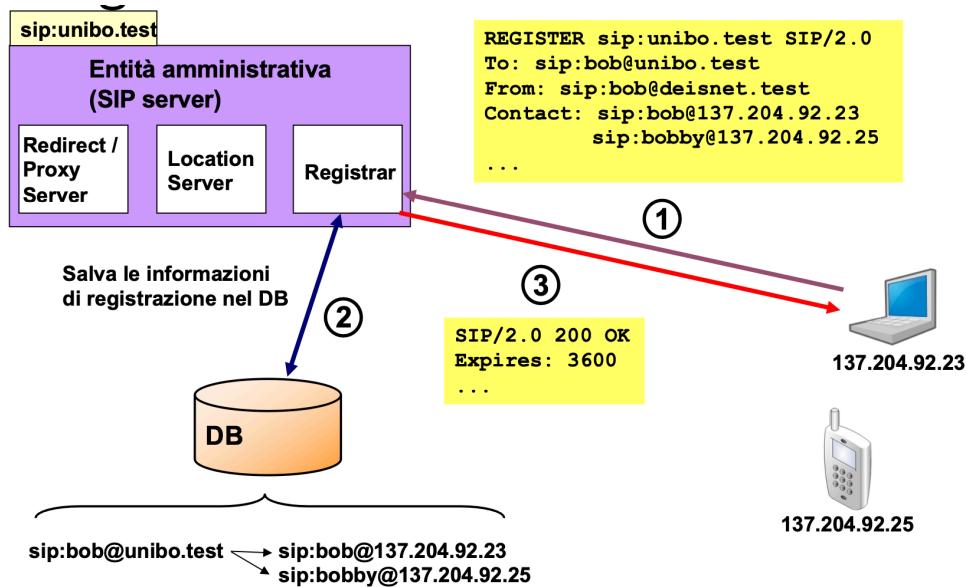
- 1xx → provisional: in ricerca
- 2xx → Success
- 3xx → redirection: forwarding
- 4xx → client error
- 5xx → server error
- 6xx → global failure

Scenari applicativi del sip

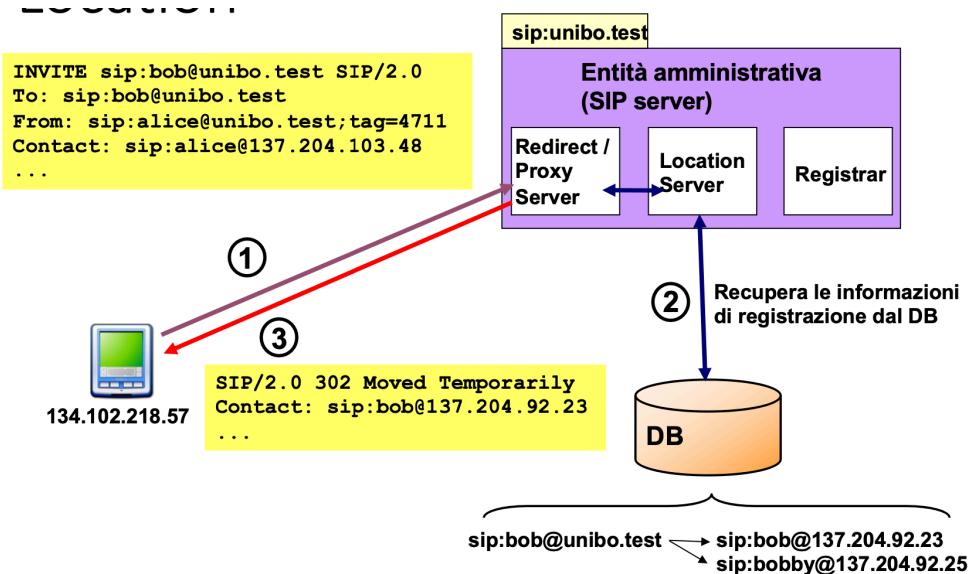
Chiamata Diretta:

aa UA a UA

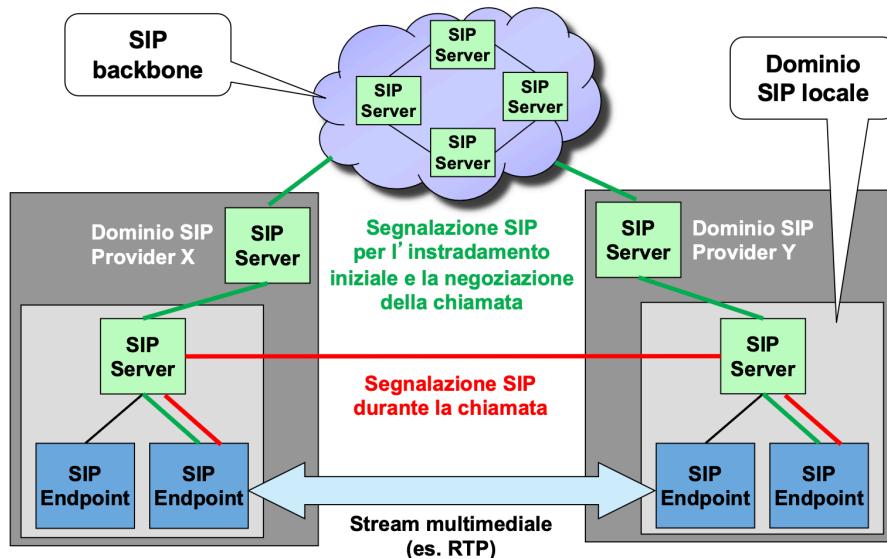




Location server: usato per accedere al db e chiedere lista di indirizzi da parte di SIP server



Si crea quindi una situazione multidominio in cui diversi provider devono dialogare per collegare vari utenti finali



Controllo di canale

Il canale non è ideale e introduce errori di propagazione, specialmente di carattere semantico.

Il controllo del livello 2 è più semplice del livello 3, perciò si parte a capire il livello 2 e poi si estende al 3.

I servizi di controllo del canale intendono

- rendere affidabile e sicuro il servizio di collegamento che lo strato 2 offre alle entità di strato 3

Le funzioni tipicamente svolte dallo **strato 2** per il controllo del canale:

- strutturazione del flusso di dati
- Le PDU di strato 2 sono dette trame o frame
- controllo e gestione degli errori di trasmissione
- controllo di flusso
- controllo di sequenza
- gestire il protocollo di accesso per un collegamento punto-multipunto

Ma non tutti i protocolli di strato 2 svolgono tutte queste funzioni, alcuni implementano solo dei sottoinsiemi

Controllo errore

Il controllo di errore si può fare sia a correzione che rilevazione.

I sistemi di codifica a correzione di errore sono molto più costosi di quelli di rilevazione.

Quello che conviene dipende dalla tipologia del canale.

Di seguito alcuni algoritmi di rilevazione.

Codice a blocco sistematico

Un'altra modalità simile è quella di creare una sorta di "riassunto" similmente alla firma digitale e poi viene aggiunto a fine messaggio, questo serve per rilevare l'errore, ovviamente mittente e destinatario devono usare la stessa funzione. Se il destinatario controlla e sono uguali allora messaggio giusto, altrimenti errato.

Bit di parità

Si aggiunge a fine stringa 0 o 1 per fare sì che il numero di 1 sia pari. È un metodo semplice ed efficace, ma se il numero di errori è pari non viene rilevato

Internet checksum

- Nei protocolli di Internet vengono solitamente utilizzati codici a blocchi sistematici
- Sono **estensioni del bit di parità**, volte ad estenderne le prestazioni
- Si applica su parole di 16 bit, indipendente dalla lunghezza complessiva del blocco dati, e **non dipendono dal big endian o little endian**.

Viene fatta la somma a complemento ad 1, che permette di ottenere sempre risultati con lo stesso numero di bit degli addendi di partenza.

somma complemento a 1

```
11110010 +
11110100
-----
111100110
1
-----
**11100111**
```

Questo protocollo è molto sicuro, ma bisogna usare per forza 16 bit, con 8 diventa molto meno sicuro e inoltre è difficile predire l'accuratezza in partenza. Perciò vengono inventati altri algoritmi tipo quelli polinomiali:

Codici polinomiali

I codici polinomiali sono una tecnica usata nel campo delle telecomunicazioni per rilevare e correggere errori che possono verificarsi durante la trasmissione dei dati. L'idea di base è rappresentare le sequenze di bit (cioè i dati) come coefficienti di un polinomio. Questo polinomio viene poi "codificato" combinandolo con un altro polinomio detto **polinomio generatore**, che serve a costruire una versione più lunga del messaggio, arricchita di bit ridondanti. Questi bit aggiuntivi non portano nuove informazioni, ma permettono al ricevitore di verificare se il messaggio ricevuto è integro.

Nel momento in cui il messaggio arriva al destinatario, il sistema può controllare se ci sono stati errori dividendo il polinomio ricevuto per lo stesso polinomio generatore usato in trasmissione. Se il resto della divisione è zero, il messaggio è probabilmente corretto; altrimenti si è verificato un errore. Questo approccio è tipico dei **codici CRC** (Cyclic Redundancy Check), molto usati in rete Ethernet, nei protocolli seriali e in tanti altri contesti.

In sintesi, i codici polinomiali permettono un controllo efficiente degli errori sfruttando l'algebra dei polinomi, offrendo un buon compromesso tra affidabilità e semplicità di implementazione, soprattutto

per il rilevamento di errori singoli o a raffica.

esempio di creazione di messaggio:

- Il **messaggio** da trasmettere è: **1101**
- Il **polinomio generatore** (prestabilito e condiviso da trasmittitore e ricevitore) è: **1011**
Questo corrisponde al polinomio $x^3 + x + 1$

INVIO

1. Si prende il messaggio **1101** e si **aggiungono 3 zeri** in fondo (perché il generatore è di grado 3):

$$1101 \rightarrow 1101000$$

2. Si **divide** questo nuovo numero binario per il generatore **1011** usando la divisione binaria modulo 2 (come XOR):

Il resto di questa divisione (diciamo r) sarà lungo al massimo 3 bit.

3. Supponiamo che il resto della divisione sia **011**.

Si **aggiunge il resto al messaggio originale**, ottenendo il messaggio codificato:

$$1101 + 011 \rightarrow \mathbf{1101011}$$

RICEZIONE

Il ricevitore riceve **1101011** e lo **divide per lo stesso polinomio generatore 1011**.

- Se il **resto è 0**, significa che non ci sono stati errori nella trasmissione.
- Se il **resto è diverso da 0**, significa che c'è stato un errore.

Rivelazione dell'errore

Il mittente ha preparato il messaggio $T(x)$ aggiungendo dei bit ridondanti in modo che **sia perfettamente divisibile** per il polinomio generatore $G_r(x)$ (cioè, senza resto).

Quando il ricevitore riceve un messaggio $T'(x)$, questo può essere stato corrotto da un errore rappresentato da un polinomio $E(x)$,

$$T'(x) = T(x) + E(x)$$

Si esegue la divisione:

$$T'_{n-1}(x)/G_r(x) = [T_{n-1}(x) + E(x)]/G_r(x) = T_{n-1}(x)/G_r(x) + E(x)/G_r(x)$$

- $T(x)/G_r(x) \rightarrow$ dà **resto 0**, perché il messaggio originale era stato costruito per essere divisibile per $G_r(x)$.
- Quindi, se il **resto finale ≠ 0**, è colpa di $E(x)/G_r(x)$, cioè c'è un errore rilevato.

Per **rilevare l'errore**, è sufficiente che:

$$E(x)/Gr(x) \rightarrow resto \neq 0$$

Errore non rilevato solo se:

$$E(x) \text{ è divisibile per } Gr(x)$$

Perciò **Gr(x)** va scelto in modo da **minimizzare la probabilità** che un errore dia resto nullo.

Capacità del codice e scelta di Gr(x)

Vediamo come Gr(x) si comporta con diversi tipi di errori:

Errore singolo

$$E(x) = x^i$$

È sufficiente che $Gr(x)$ abbia **almeno 2 bit a 1** per rilevarlo sempre.

Numero dispari di errori

- Se $Gr(x)$ è **multiplo di (1 + x)**, **non divide mai** un polinomio con un numero dispari di termini (cioè un numero dispari di errori).
- Se invece $Gr(x) = 1 + x$, si ottiene il classico **bit di parità** (1 bit di ridondanza).

Due errori

$$E(x) = x^i + x^j = x^j(x^h + 1)$$

Alcuni generatori **non dividono mai ($x^h + 1$)** → utile per rilevare 2 errori.

ITU propone: $G16(x) = x^{16} + x^{12} + x^5 + 1$

Questo polinomio rileva efficacemente molti casi di errore.

Errori a burst (a raffica)

Nelle telecomunicazioni, gli errori non sono sempre distribuiti casualmente, ma possono concentrarsi in blocchi (burst). Un errore a burst potrebbe essere dato da un evento casuale non prevedibile (una persona schiaccia un cavo e per un istante modifica le caratteristiche del filo).

Un **burst** di lunghezza k viene rappresentato con un polinomio di grado $k - 1$.

Possibili scenari:

Lunghezza del burst (k-1)	Risultato della divisione
$k - 1 < r$	Errore sempre rilevato
$k - 1 = r$	Errore rilevato con probabilità $1 - 1/2^{(r-1)}$
$k - 1 > r$	Errore rilevato con probabilità $1 - 1/2^r$ (cioè può sfuggire)

Dove r è il grado di $Gr(x)$, quindi il numero di **bit di ridondanza**.

ARQ automatic repeated request

Protocolli che servono per rendere affidabile la comunicazione in rete. Lavorano su due livelli: strato di linea (tra due nodi direttamente collegati) e strato di trasporto (tra due dispositivi finali, anche molto distanti tra loro).

Ogni messaggio trasmesso viene verificato tramite codici a rilevazione di errore (es. CRC). Se si rileva un errore, oppure se un messaggio non arriva entro un certo tempo, viene chiesta la ritrasmissione automatica. In questo modo si possono affrontare problemi come:

- errori nei bit trasmessi,
- perdita di informazioni,
- fuori sequenza dei pacchetti.

Finestra scorrevole

È una tecnica usata nei protocolli di comunicazione per gestire in modo efficiente e coordinato tre aspetti fondamentali:

- Controllo dell'errore – assicurarsi che i dati arrivino correttamente.
- Controllo di flusso – evitare di sovraccaricare il ricevitore con troppi dati troppo velocemente.
- Controllo di sequenza – mantenere l'ordine corretto dei pacchetti ricevuti.

Per fare questo, si usano insieme:

- Codici di rivelazione d'errore (es. CRC), per capire se un pacchetto è corrotto;
- Numerazione dei pacchetti, per tenerne traccia e ordinarli;
- Conferme di ricezione (ACK), per far sapere al mittente che certi pacchetti sono arrivati correttamente.

Cos'è la finestra?

Immagina una "finestra" che si muove lungo una sequenza di pacchetti numerati.

Il mittente può inviare solo i pacchetti che stanno all'interno della finestra, evitando che il mittente mandi troppi dati a un ricevitore lento.

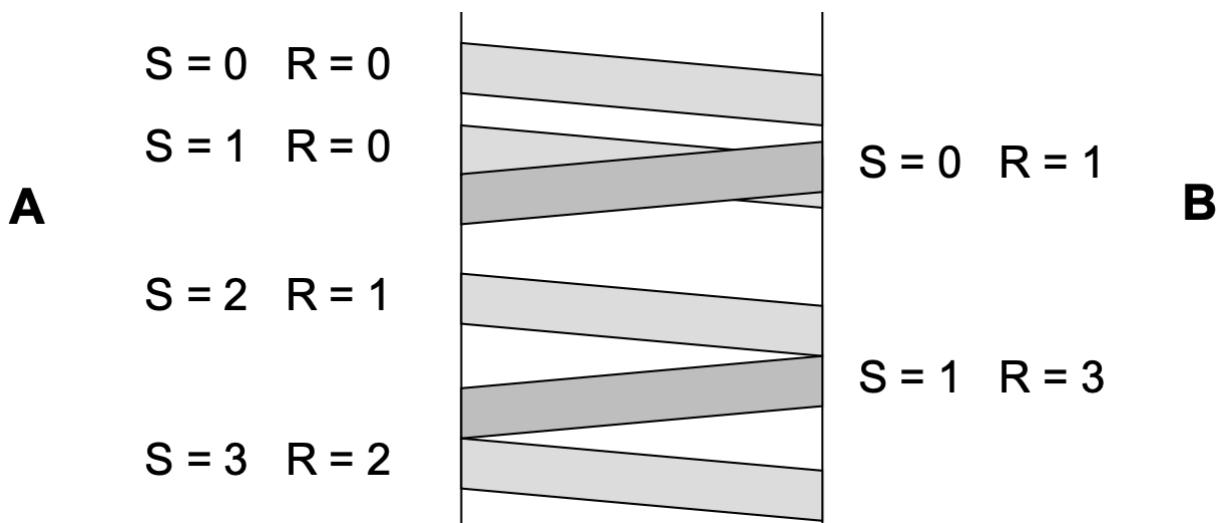
Quando riceve una conferma (ACK) per il primo pacchetto, la finestra scorre avanti e il mittente può inviare un nuovo pacchetto.

Permette di inviare più pacchetti senza aspettare ogni singola conferma, trasferimento più veloce e rende possibile la ritrasmissione selettiva dei soli pacchetti persi o errati.

Esempio

I protocolli ARQ numerano sequenzialmente le unità informative (UI) da consegnare ai protocolli superiori

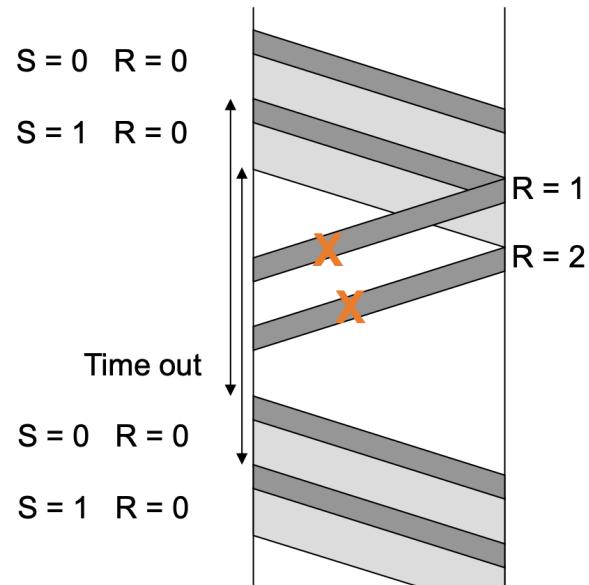
- S conta unità inviate **"posizionamento" nel flusso**
- R conta unità ricevute in modo corretto **confermare di ricezione**



Nel caso di errore o Time-out

Il trasmettitore invia nuovamente il pacchetto.

- Il protocollo può entrare in stallo (**deadlock**)
 - Se le trame informative sono perdute
 - Se gli ACK sono perduti
- È necessario un **time out** per riprendere il dialogo
 - Un orologio parte al termine della trasmissione di ciascuna trama
 - Se si raggiunge il time out senza avere conferma si ritrasmette la trama



ACK

ACK parte dal ricevitore inviando al trasmettitore il proprio valore di R.

- Le PDU ricevute in modo corretto fanno aumentare R
- Quando una PDU viene ricevuta in modo non corretto viene ignorata ed R non viene modificato

TIPOLOGIA ACK

- **Esplicita**: ogni PDU ricevuta correttamente genera una conferma
- **Implicita** (cumulativa): una PDU di conferma con $R = n$ conferma la ricezione fino a $n-1$
- In **piggybacking**: viaggia inserita (a “cavalluccio”) in una PDU contenente dati utili

Approfondimento

Gli acknowledge o ACK

- non portano dati di utente ma solo informazioni di controllo per il protocollo

Servono qualora

- Il protocollo ARQ non possa usare il piggybacking
- Il ricevitore non abbia dati da trasmettere

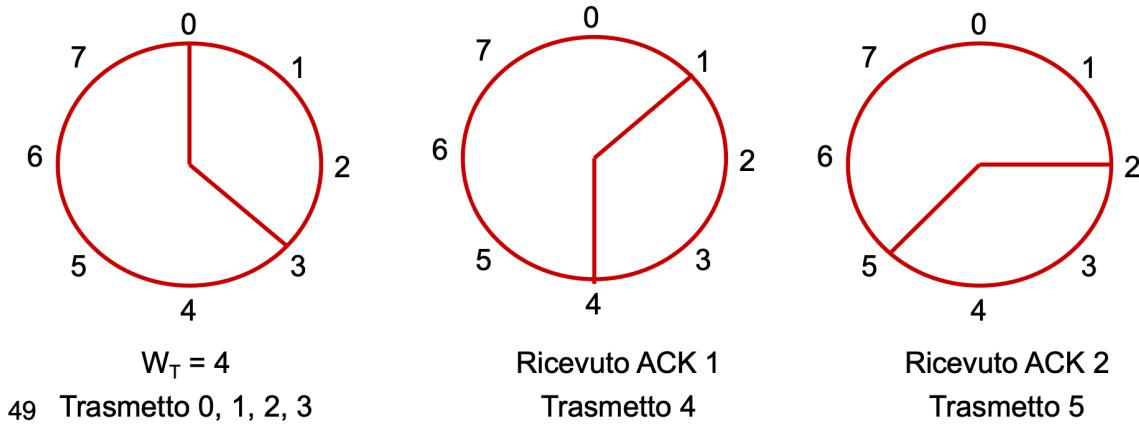
Non è necessario numerare gli ACK

- I protocolli ARQ tipicamente confermano la ricezione delle PDU che portano dati d’utente e non confermano la ricezione degli ACK (conferma della conferma)
- Non si ritiene necessario controllare la sequenza degli ACK

WT = numero massimo di trame che il trasmettitore può inviare senza ricevere alcuna conferma

La numerazione delle trame viene effettuata modulo M

- $M = 2n$ dove n è il numero di bit utilizzati per la numerazione
 - Si può procedere con la trasmissione di nuove trame solo al ricevimento delle conferme
- La numerazione delle trame trasmesse scorre nel tempo (sliding window)



Per quale motivo imporre W finito e sospendere la trasmissione delle trame?

Garantire unicità di numerazione delle trame

Lo spazio di numerazione dipende dal numero di bit dedicati alla numerazione nell'intestazione e ha dimensioni limitate.

Se si continuasse a trasmettere all'infinito non si avrebbe più una corrispondenza biunivoca trame-numero

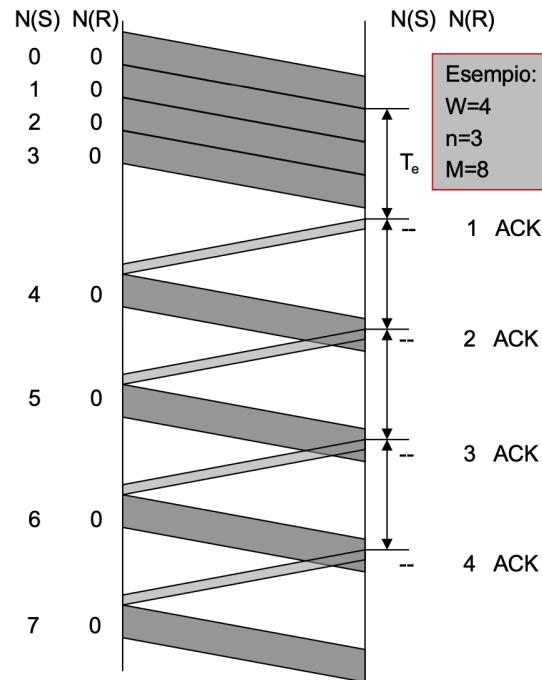
- Le trame con uguale numerazione sono indistinguibili

- Accorda la velocità del trasmettitore alla capacità del ricevitore (e della rete)

- Il ricevitore

- Deve essere in grado di gestire un' intera finestra
 - Memorizzazione ed elaborazione di W trame
- Accorda il flusso di trame in arrivo tramite le conferme
- A regime invia una nuova trama ogni T_e
 - T_e = tempo necessario per elaborare una trama

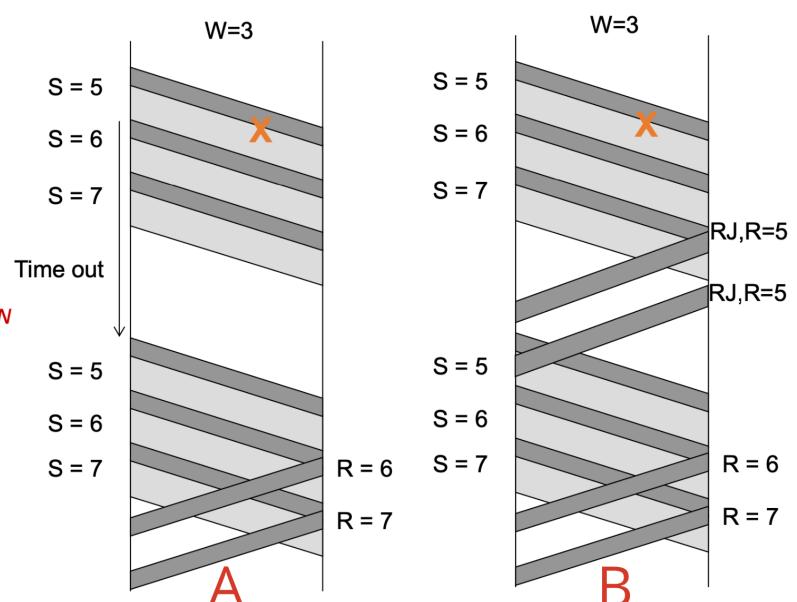
52



Due situazioni nel caso di errore:

- **A** Il mittente invia nuovamente tutte le trame partendo da quella errata dopo un time-out
 - È molto inefficiente se ci sono errori frequentemente
 - bisogna attendere timeout
- **B** Quando destinatario riceve 6 dopo 4 capisce che 5 non c'è e invia un ACK negativo dicendo che manca 5 e il mittente invia nuovamente tutto

- Viene persa la trama N
- Il ricevitore
 - **Scarta** tutte le trame successive a quella errata
 - A seconda dell' implementazione
 - **Segnale** al trasmettitore la mancata ricezione della trama N
 - **Rimane in silenzio** senza inviare alcuna trama di segnalazione
- Il trasmettitore
 - **Ritrasmette tutte la trame a partire dalla numero N**
- Vantaggi
 - Semplicità operativa
 - Ridotta complessità nel ricevitore
- Svantaggi
 - Inefficienza
 - Si ritrasmettono trame senza che questo sia strettamente necessario



La finestra deve essere strettamente minore dello spazio di numerazione, altrimenti non si riesce a capire in maniera precisa di quale messaggio si sta facendo riferimento

Selective repeat ARQ

Diversamente dagli altri il selective, nel caso di errore, invia esclusivamente il pacchetto errato.

Dimensione Time out

tropo breve: invio trame prima di ricevere ACK

tropo lungo: troppo tempo perso

Va commisurato in base al RTT (round trip time)

tempo necessario per effettuare un'andata e ritorno sul canale. Nello strato 2 è prevedibile ed è abbastanza standard.

Prestazioni delle reti e protocolli

I protocolli sono progettati per garantire:

- Funzionalità
- Prestazioni

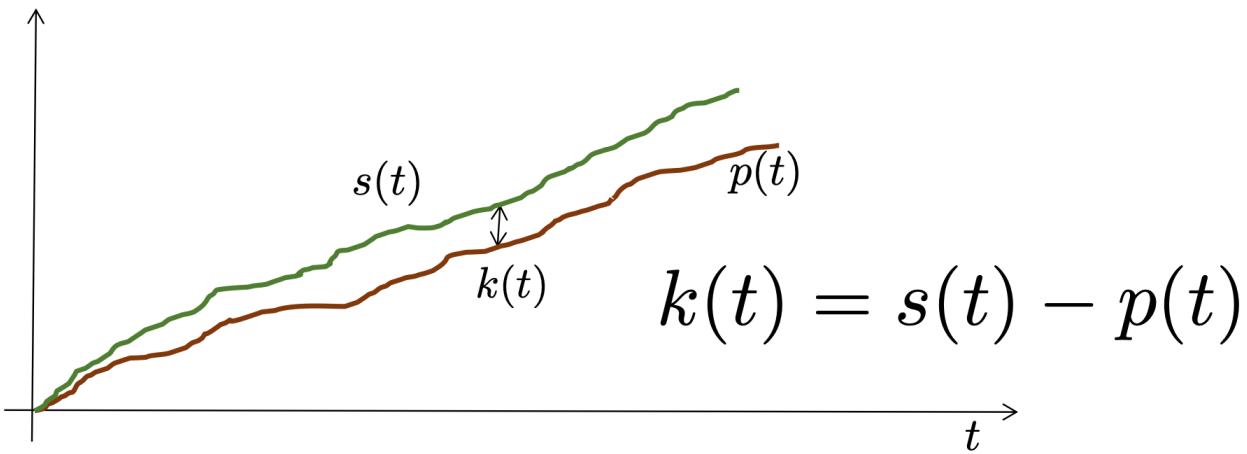
Un sistema deve smaltire del lavoro che gli viene offerto dall'esterno

• Esempio nel caso specifico delle reti di tlc

- Livello N+1 invia PDU al livello N tramite la relativa interfaccia (e un opportuno SAP)
- Livello N impiega un certo tempo per soddisfare la richiesta

Terminologia

- $k(t)$ Numero di richieste accettate in attesa di essere soddisfatte
- $a(t)$ Numero di richieste di servizio giunte al tempo t
- $s(t)$ Numero di richieste accettate al tempo t
- $p(t)$ Numero di partenze dal sistema al tempo t



Frequenza media delle richieste offerte

$$\lambda = \lim_{t \rightarrow \infty} \frac{a(t)}{t}$$

Frequenza media delle richieste smaltite

$$\lambda_s = \lim_{t \rightarrow \infty} \frac{p(t)}{t}$$

Se il sistema in oggetto non produce lavoro ma lo riceve solamente dall'esterno

$$\lambda_s \leq \lambda$$

questo implica che

$$r(t) = a(t) - s(t)$$

$r(t)$ = richieste non accettate e rifiutate o perdute

$$\lambda_p = \lim_{t \rightarrow \infty} \frac{r(t)}{t}$$

ovvero la frequenza di perdita e quindi

$$\lambda = \lambda_s + \lambda_p$$

Problema

Oltre ad aver definito dei valori standard bisogna scegliere un'unità di misura sempre uguale:

Il riferimento è il tempo di servizio dell'intero pacchetto
che solo se completato produce un risultato «utile» per
l'utente

può essere **aleatorio o deterministico**
identificato da θ

esempio deterministico

- L lunghezza del pacchetto in bit
- C capacità del canale in bit per secondo

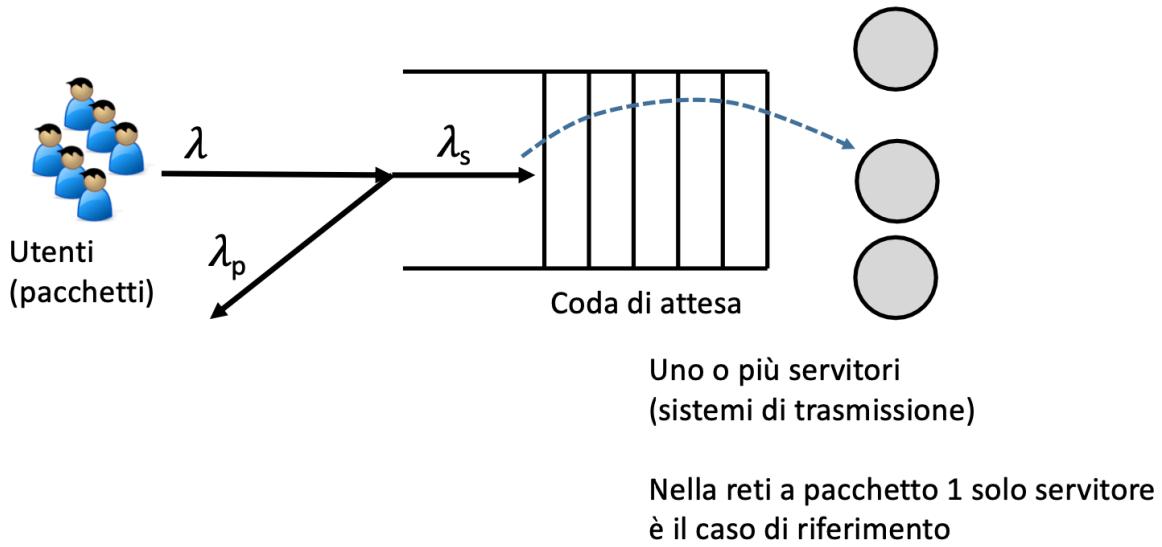
$$\bar{\theta} = \frac{L}{C}$$

l'inverso del tempo medio di servizio viene detto frequenza media di servizio

$$\mu = \frac{1}{\bar{\theta}}$$

esempio:

se $\bar{\theta} = 0.5s$ il servitore smaltisce al massimo $\mu = 2 \text{ pacchetti}/s.$



$\bar{\sigma}$ = tempo medio totale speso dal singolo utente nel sistema a coda che è composto da:

- $\bar{\theta}$ = tempo di servizio
- T_a = tempo speso in coda

$$\bar{\sigma} = \bar{\theta} + T_a$$

Si definisce **traffico** il numero medio di utenti presenti nel sistema

$$A = \lambda \bar{\delta}$$

il traffico ha l'unità di misura fittizia Erlang. In realtà vale in base alla tipologia di rete e comunicazione.

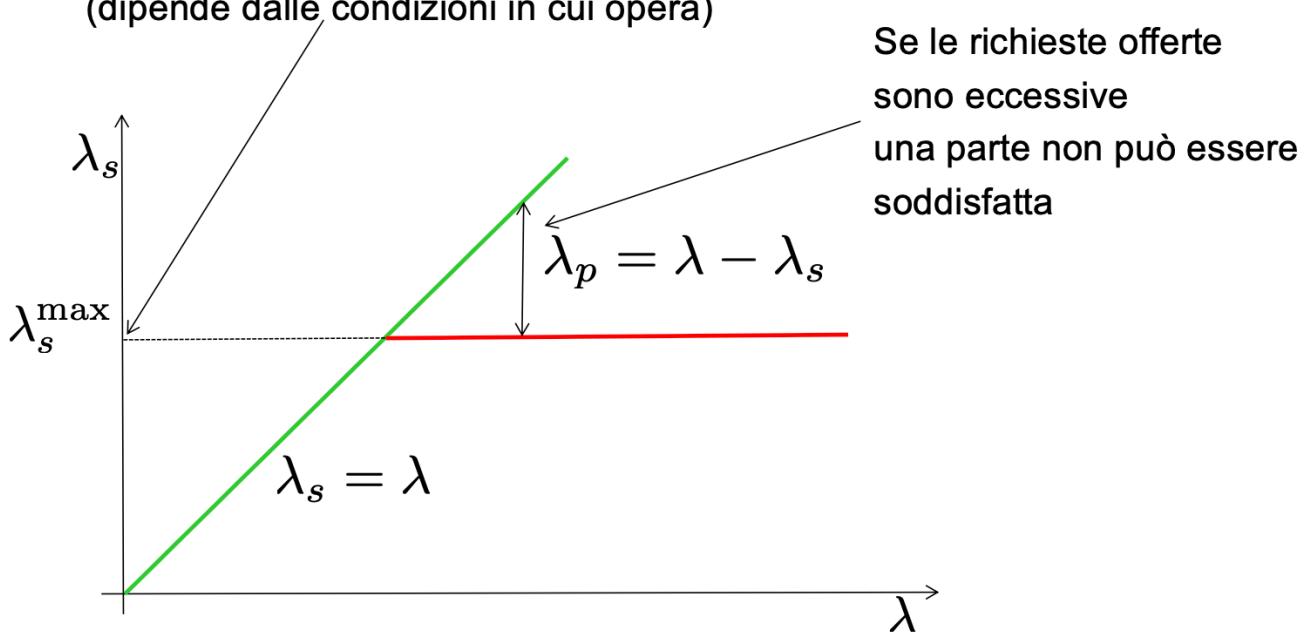
- $A_0 = \lambda \bar{\delta} \rightarrow$ Traffico offerto
- $A_s = \lambda_s \bar{\delta} \rightarrow$ Traffico smaltito o **throughput**
- $A_p = \lambda_p \bar{\delta} \rightarrow$ Traffico perduto

m è il numero di servitori:

$$0 \leq A_s \leq m$$

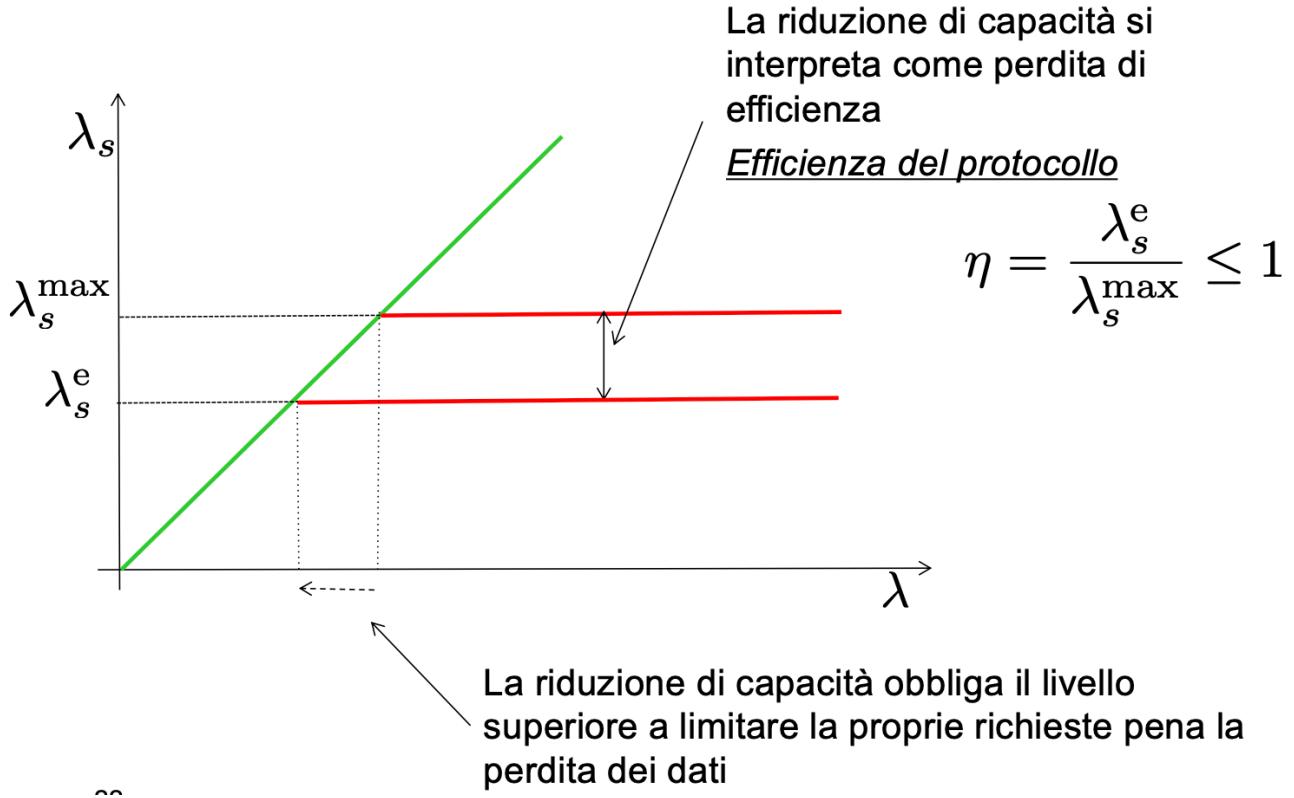
Richieste offerte in sistema ideale

Il sistema ha una capacità massima finita
di smaltire richieste
(dipende dalle condizioni in cui opera)



21

Richieste offerte in sistema reale



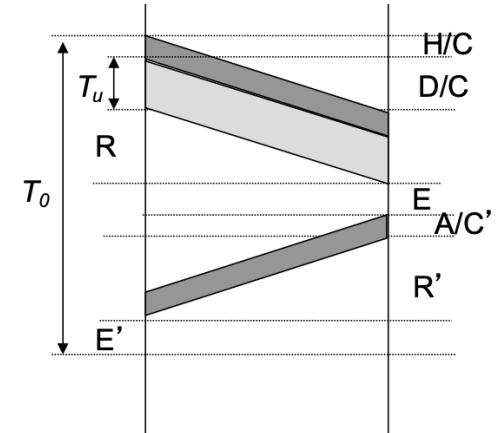
Prestazioni stop and wait

Altri valori per valutare la Capacità massima

- C Capacità max canale
- $\bar{\sigma} = \frac{L}{C} = \frac{1}{\mu}$ tempo medio di servizio minimo
- $\bar{\sigma}_e = \frac{L}{C_e} = \frac{1}{\mu}$ nel caso in cui il protocollo necessiti di maggior tempo per trasmettere la trama

Prestazioni Stop-and-Wait

- Qual è l'efficienza del protocollo ARQ stop and wait
 - Stop and wait equivale ad un protocollo a finestra scorrevole con finestra unitaria
- D : dimensione campo dati in bit
- H : dimensione dell' header (PCI) in bit,
- F=D+H : lunghezza totale del frame,
- A : lunghezza dell'ACK,
- E, E' : tempi di elaborazione per il controllo del frame in arrivo e per la preparazione del frame in partenza
- R : tempo di propagazione del segnale da un capo all'altro del collegamento,
- I = E+R ; I' = E' +R'
- C, C' : velocità dei canali di trasmissione
- In generale il canale di andata e di ritorno possono essere diversi



Dall'immagine sopra si evince che gran parte del tempo sprecato è il vuoto a sinistra, motivo per cui viene implementata la finestra scorrevole. Si possono inviare altri bit prima che si riceva ACK.

Di seguito vengono elencate alcune formule per ottenere efficienza:

Tempo intercorso fra l'invio di due frame successivi:

$$T_0 = \frac{F}{C} + I + \frac{A}{C'} + I'$$

Mentre il tempo necessario per trasmettere solo dati utente, senza nessun protocollo è sempre

$$T_d = \frac{D}{C}$$

Definiamo poi l'efficienza come:

- **Efficienza:**

$$\eta = T_d / T_0 = D / (C T_0) = D / (D+H+IC+I'C+AC/C')$$

- Per semplicità poniamo $I = I'$, $C = C'$, inoltre l'ACK è praticamente composto dalla sola PCI, e quindi $A \approx H$

$$\eta = D / (D+H+A+2IC) = D / (D+2H+2IC) = D / (D+O)$$

- **Overhead**

$$O = 2H + 2IC$$

IMPORTANTE

Ovviamente bisogna minimizzare overhead. Ovviamente come immaginabile Header deve essere minimo come I (distanza fisica tra terminali che non si può controllare). Controintuitivamente se C è troppo alto si rischia di avere una perdita di efficienza.

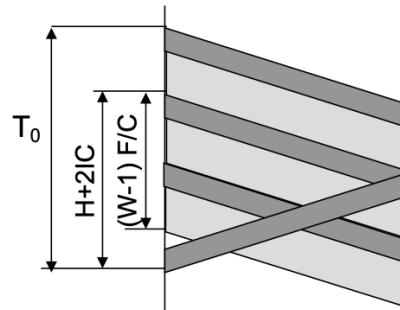
Protocolli a finestra scorevole

Come si sceglie la finestra?

- È necessario distinguere due casi

- $(W-1)F/C < H + 2IC$, cioè $WF < CT_0$
 - Si trasmettono W trame in un tempo pari a T_0 per cui

$$\eta = (WD) / (D+2H+2IC)$$

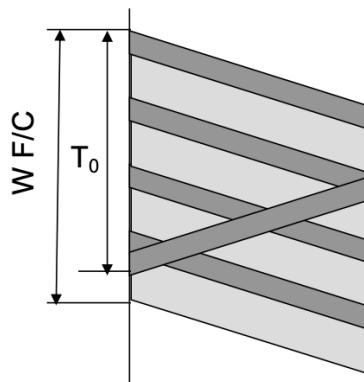


- $WF \geq CT_0$
 - il trasmettitore non interrompe mai la trasmissione delle trame per cui risulta:

$$\eta = D / (D+H) \text{ oppure}$$

$$\eta = D / (D+O), \text{ con } O = H$$

- Il secondo caso è il migliore per l'efficienza : in trasmissione non ci sono tempi morti



Per avere efficienza massima devo avere la finestra più lunga di T_0

Analisi con errori

- Prima di trasmettere correttamente una trama possono avvenire k errori

$$k \geq 0$$

- Il tempo necessario a trasmettere la trama, dati k errori, vale

$$T_k = k T_e + T_o$$

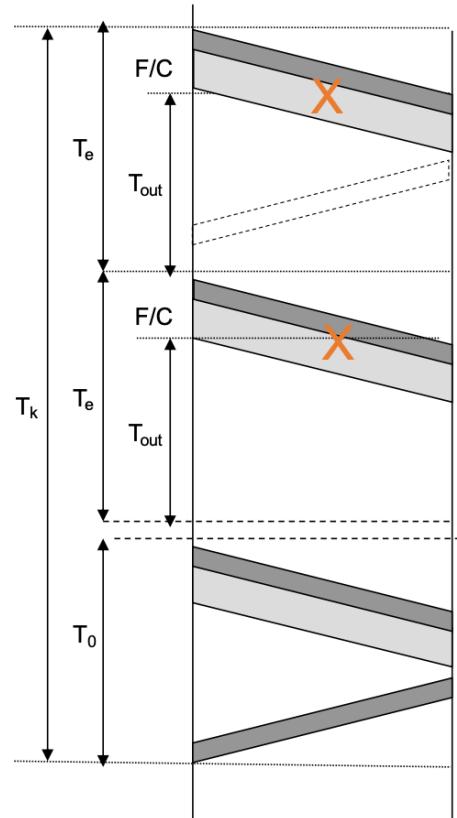
- Dove

$$T_o = (D + O)/C$$

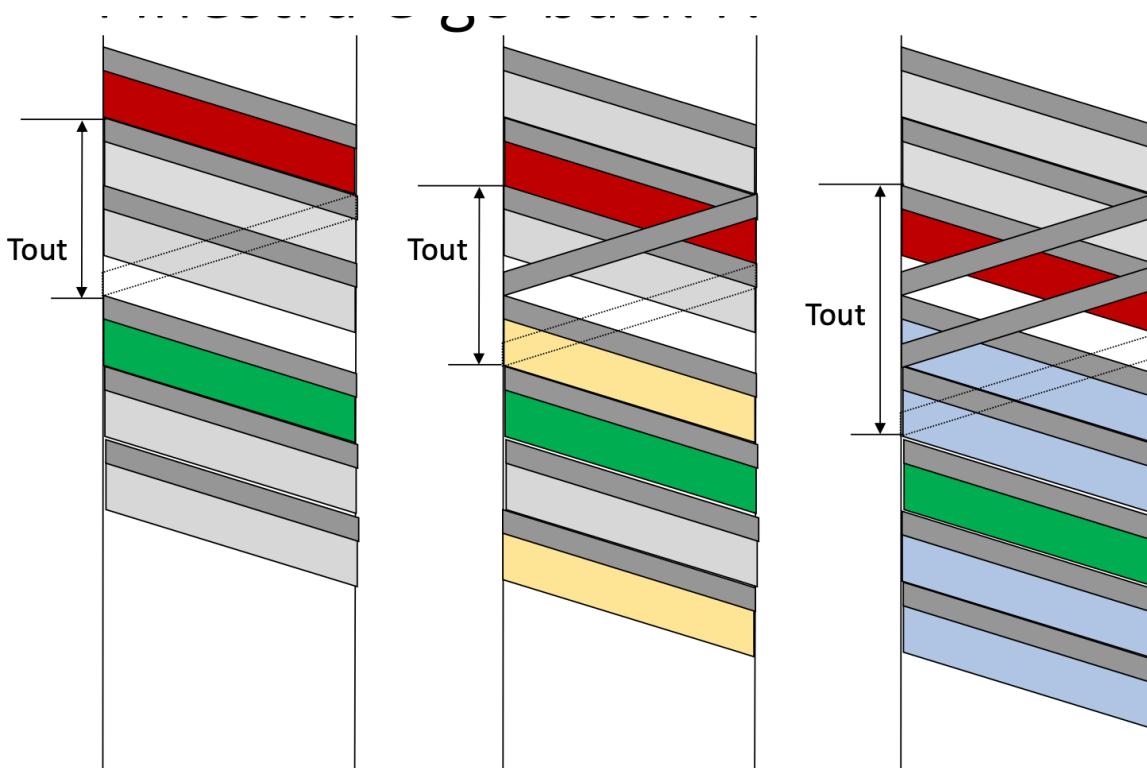
$$T_e = D/C + H/C + T_{out}$$

- Se P_k è la probabilità di avere k errori, il tempo medio per trasmettere una trama vale

$$E[T_k] = \sum_{k=0 \dots \infty} T_k P_k = \sum_{k=0 \dots \infty} (k T_e + T_o) P_k$$



Finestra e go back N



- In pratica se utilizzo go-back-N ARQ la trama di ritrasmettere è sempre la prima di una finestra

Protocolli di trasporto

In Internet, lo strato di **trasporto** ha il compito di **gestire la comunicazione** tra applicazioni che girano su dispositivi diversi. È tra IP e Applicazione.

I due protocolli principali sono:

- **TCP** connessione affidabile.
- **UDP** più veloce ma meno sicuro, non controlla errori.

Con l'arrivo delle applicazioni multimediali (come streaming e videochiamate), si sono rese necessarie soluzioni per il trasporto real-time, portando alla definizione di nuovi protocolli come RTP e RTCP.

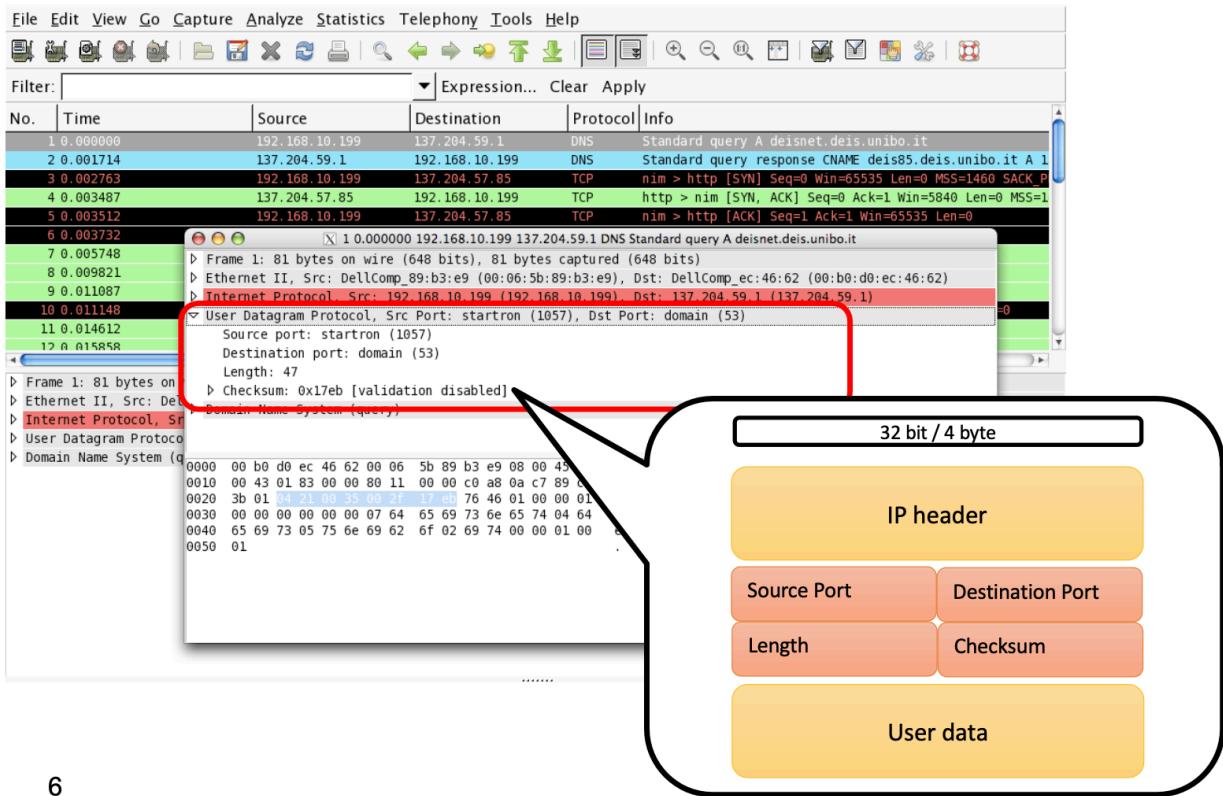
Una funzione chiave dello strato di trasporto è la **multiplazione**, cioè la possibilità di far comunicare più applicazioni contemporaneamente, distinguendo i dati grazie ai numeri di porta. Inoltre, questo strato cerca di assicurare che la **qualità della trasmissione sia adatta** alle esigenze dell'applicazione che lo utilizza.

UDP

È un protocollo **connectionless**. Ogni messaggio viaggia in modo indipendente, senza garanzie di ordine o consegna.

Per **piccoli blocchi** di dati, dove non è necessario un controllo accurato della qualità del trasporto. Viene spesso usato in applicazioni come **DNS** o **e-mail**, dove la perdita o il ritardo di un singolo pacchetto non compromette gravemente la comunicazione.

Formato messaggio



6

TCP

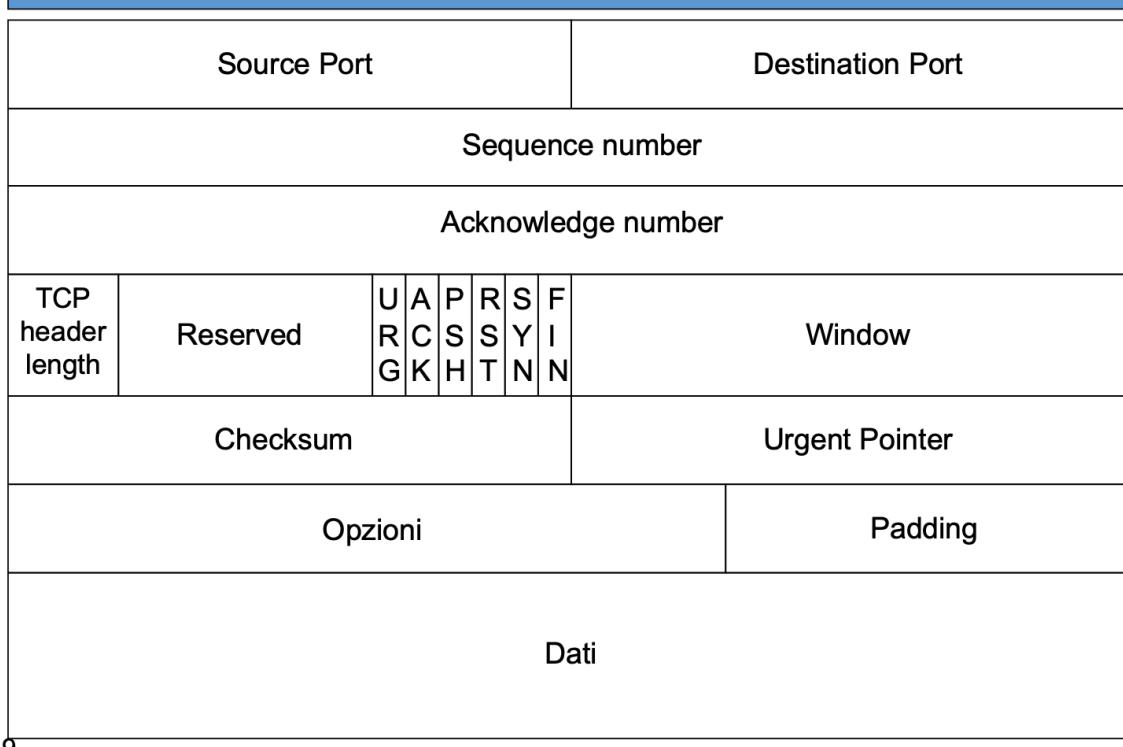
Nel protocollo TCP, i dati dell'applicazione vengono suddivisi in unità chiamate segmenti. Ogni segmento è composto da:

- **header standard** di 20 byte,
- **header opzionale** usato per opzioni avanzate,
- **payload** dati veri e propri dell'applicazione.

La dimensione massima dei dati che possono essere trasportati in un singolo segmento è chiamata MSS (Maximum Segment Size). L'MSS indica quindi la quantità massima di dati applicativi contenibili in un segmento, escludendo l'header.

Formato header TCP

32 bit



9

- **Source\Destination port:** numero della porta sorgente\destinazione
- **Sequence number:** numero di sequenza del primo byte del pacchetto; se è presente il bit SYN questo è il numero di sequenza iniziale su cui sincronizzarsi
- **Acknowledge number:** se il bit ACK è a 1 allora questo numero contiene il numero di sequenza del blocco di dati che il ricevitore si aspetta di ricevere
- **TCP Header Length (4 bit):** numero di parole di 32 bit dell'intestazione TCP; indica dove iniziano i dati
- **Reserved:** sei bit riservati per uso futuro
- **Control bit** sono 6 bit di controllo
 - URG posto a 1 se si deve considerare il campo Urgent Pointer
 - ACK posto a 1 se si deve considerare il campo Acknowledge
 - PSH posto a 1 serve per la funzione di push
 - RST posto a 1 per resettare la connessione
 - SYN posto a 1 per sincronizzare i numeri di sequenza
 - FIN posto a 1 per indicare la fine dei dati

- **Window:** finestra del ricevitore, cioè il numero di byte che il ricevitore è disposto a ricevere, partendo dal numero di sequenza di quello contenuto nel campo acknowledge
- **Checksum:** controllo dell'errore sul segmento
- **Urgent Pointer:** contiene puntatore a dati urgenti eventualmente presenti nel pacchetto (es. per abortire programma remoto in esecuzione), ha senso se il bit URG è posto ad 1
- **Options:** contiene opzioni per la connessione
- **Padding:** bit aggiuntivi per fare in modo che l'intestazione sia multipla di 32 bit

Three ways handshake

Procedura usata dal protocollo TCP per stabilire una connessione affidabile tra due dispositivi (tipicamente un client e un server).

Garantisce che entrambe le parti siano pronte a comunicare e che abbiano sincronizzato correttamente i numeri di sequenza, necessari per tenere traccia dei dati inviati e ricevuti.