



Corso di Laurea in Matematica
Dipartimento di Matematica e Fisica

Sistemi per l'elaborazione delle informazioni

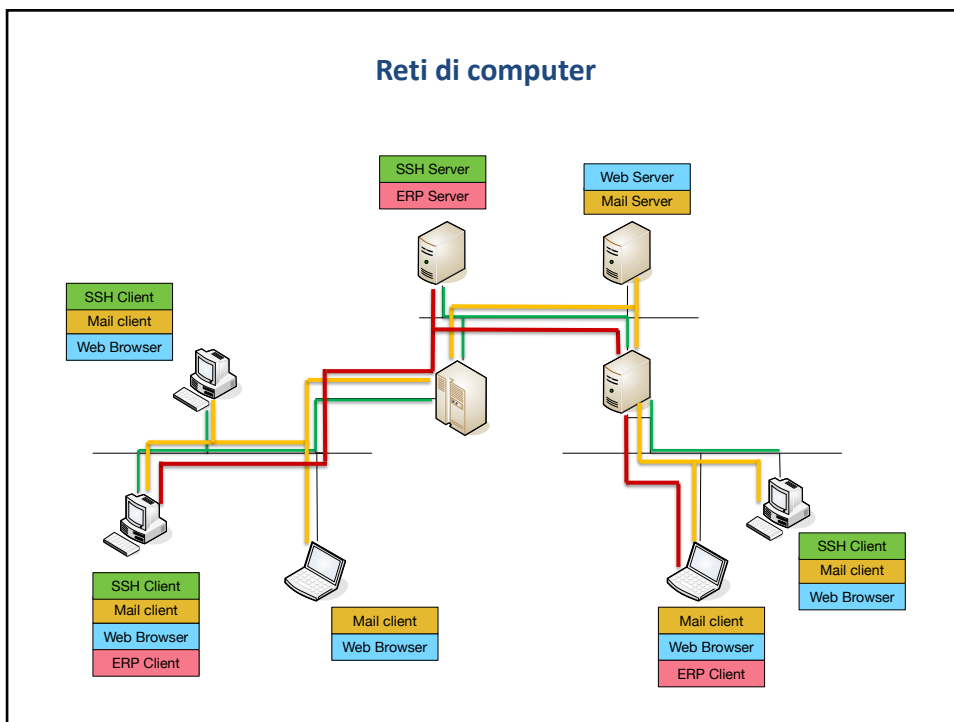
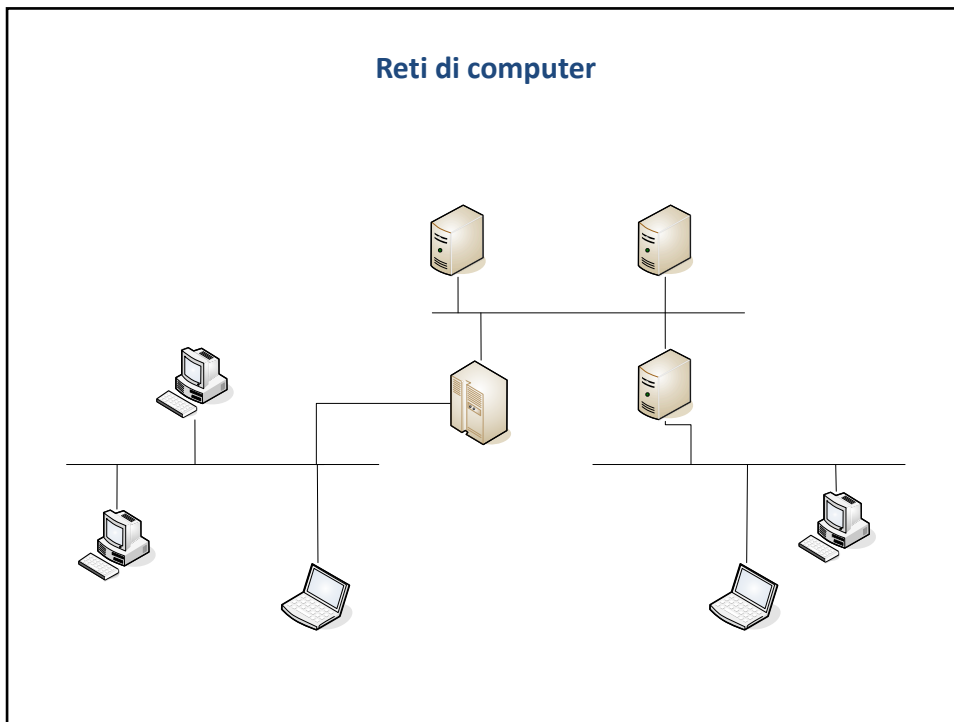
4. Reti di computer

Dispense del corso IN530 a.a. 2019/2020

prof. Marco Liverani

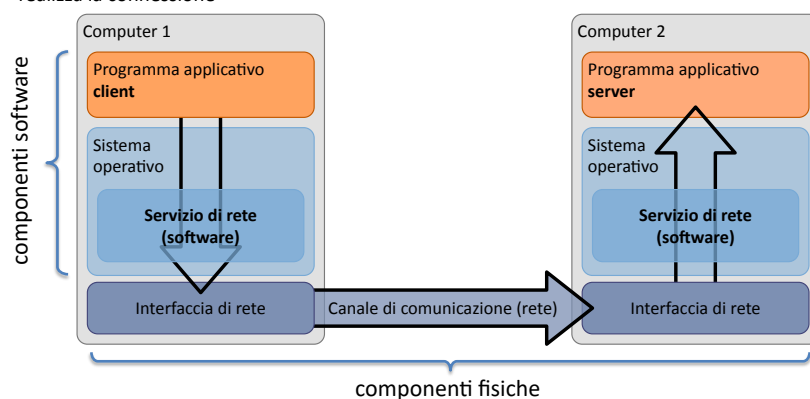
Reti di computer

- Le reti sono un componente fondamentale per la realizzazione di un sistema informativo
- Senza la rete i computer non possono comunicare fra loro e le uniche modalità di decentralizzazione dell'elaborazione dell'informazione sono costituite da:
 - collegamento di terminali ad un computer centrale: l'elaborazione e la gestione dei dati sono concentrate sul computer, ma l'interazione tra computer e utenti è decentralizzata sui terminali (a poca distanza fisica dal computer)
 - trasferimento di file da un computer all'altro, mediante connessioni punto-punto prive di un effettivo meccanismo di comunicazione bidirezionale
- Si parla di **reti di computer** quando effettivamente **più sistemi di elaborazione autonomi (computer)** possono **dialogare fra loro scambiandosi informazioni** attraverso un mezzo fisico di connessione e quando tale **meccanismo di comunicazione e di scambio è replicabile** su più computer, senza dipendere dalla specificità delle due macchine che comunicano fra loro
- È molto importante la possibilità offerta dalle moderne tecnologie di rete di **interconnettere** fra loro reti diverse, anche mediante **mezzi trasmissivi eterogenei**
- Gli utilizzi che oggi facciamo delle reti di computer (utilizzate ormai, anche inconsapevolmente, prevalentemente da persone prive di competenze tecniche specifiche) sono numerosissimi: scambio di informazioni (mail, messaggistica, ...), accesso a banche dati, accesso a servizi on-line, acquisti e transazioni finanziarie, giochi e divertimento, ecc.



Come si realizza una rete di computer

- Una rete di computer si realizza attraverso le seguenti componenti
 - **componenti fisiche:** i computer e i nodi della rete (router, gateway, switch, ...), ma anche i mezzi trasmissivi (cavi elettrici, cavi in fibra ottica, onde radio, ...)
 - **componenti software:** i programmi (in parte realizzati come componenti dei sistemi operativi) che consentono ai computer e ai nodi della rete di effettuare lo scambio di informazioni, con particolare attenzione all'efficienza, all'affidabilità e (in parte) alla indipendenza dal media fisico con cui si realizza la connessione



Packet switching

- Nelle reti di trasmissione dati **l'informazione da inviare dalla sorgente al destinatario viene suddivisa in pacchetti** trasmessi separatamente
- Questa tecnica si definisce **packet switching (commutazione di pacchetto)** e permette di utilizzare il medesimo canale di comunicazione per alternare la trasmissione di dati da parte di più nodi di una stessa rete che condividono il canale
 - Come nella tecnica del *multitasking*, dove viene ripartito il tempo della CPU dedicato all'elaborazione dei diversi processi, nel *packet switching* il canale di comunicazione viene utilizzato in tempi diversi da diversi nodi della rete, alternando rapidamente le operazioni di trasmissione da parte dei diversi nodi in modo da dare l'impressione che tutti i nodi della rete stiano usando il canale contemporaneamente
- Siccome il canale di comunicazione viene usato contemporaneamente da più nodi, i pacchetti possono andare in **conflitto** fra di loro, **corrompendosi**: in tal caso è necessario trasmettere nuovamente i soli pacchetti corrotti
 - La suddivisione in pacchetti permette anche di rendere più efficiente la **correzione degli errori**: nel caso in cui si verifichi un errore nella trasmissione di un pacchetto non sarà necessario ripetere l'intera trasmissione, ma il mittente potrà limitarsi a trasmettere nuovamente solo il pacchetto che si è corrotto
- I pacchetti vengono trasmessi sequenzialmente, ma, anche a causa delle ritrasmissioni, possono arrivare in un ordine diverso da quello con cui sono stati spediti: sarà il destinatario ad occuparsi di **ricostruire la sequenza corretta** dei pacchetti per restituire l'integrità all'informazione trasmessa

Reti broadcast e punto-punto

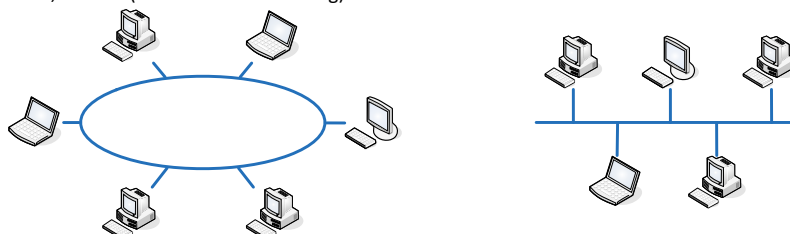
- Dal punto di vista fisico una rete può essere **broadcast** o **punto-punto**
 - **Reti broadcast**
 - Si utilizza lo **stesso canale fisico di trasmissione**, suddividendo la comunicazione in pacchetti
 - Ogni pacchetto contiene l'**identificativo del destinatario** che riceve e legge il pacchetto di dati, gli altri nodi della rete lo scartano
 - È possibile inviare pacchetti destinati a **tutti** i nodi della rete (**broadcast**)
 - È possibile inviare pacchetti destinati ad **alcuni** nodi della rete (**multicast**)
 - **Reti punto-punto**
 - Il canale trasmissivo è riservato alla comunicazione tra due nodi della rete
 - Per far giungere a destinazione un dato il pacchetto deve passare attraverso più nodi connessi fra loro in modalità "punto-punto": viene utilizzato un algoritmo di **routing** per instradare il pacchetto verso il nodo di destinazione
- Tipicamente, ma con molte eccezioni, le reti poco estese (reti locali) sono di tipo broadcast, quelle molto estese (reti geografiche) sono punto punto

Dimensione della rete

- Le reti possono essere classificate in base alla loro estensione fisica
 - **Reti locali**: collegano fra loro nodi poco distanti (es.: entro 1 Km); tipicamente collegano computer di uno stesso ufficio o edificio
 - **Reti metropolitane**: collegano edifici distanti solo pochi Km l'uno dall'altro (es.: la rete di una città o delle sedi di una stessa azienda nell'ambito di una città)
 - **Reti geografiche**: collegano computer (o intere reti locali o metropolitane) distanti fra loro centinaia o migliaia di Km
- La distanza dei computer da collegare è un dato importante, perché per distanze diverse si utilizzano tecnologie trasmissive diverse, più adatte a coprire in modo efficiente brevi o lunghe distanze

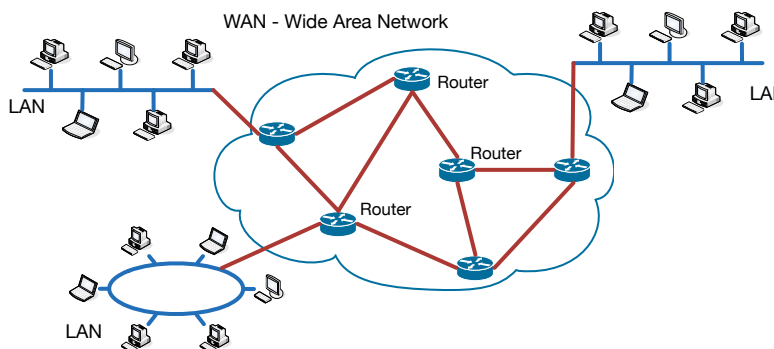
Reti Locali (LAN – Local Area Network)

- Sono di proprietà di una stessa organizzazione (tutta la rete è di proprietà e sotto il controllo della medesima azienda)
- Si estendono nell'ambito del medesimo edificio o di un "campus" (insieme di edifici in un'area privata): in generale non si può utilizzare il suolo pubblico per stendere cavi di telecomunicazione
- Velocità di trasmissione: molto elevata, da **10Mbps** fino a **1Gbps**
- **Topologia della rete** (topologia del grafo con cui può essere rappresentata la rete):
 - **bus**: tutti i nodi si agganciano ad uno stesso canale trasmissivo; se viene usato contemporaneamente da più nodi, si generano delle "collisioni" che obbligano a ritrasmettere i pacchetti di informazioni, causando un rallentamento nel traffico sulla rete (es.: reti Ethernet)
 - **ring**: la rete ha la forma di un anello su cui si agganciano tutti i nodi; i nodi trasmettono uno per volta, a turno (es.: reti IBM Token Ring)



Reti geografiche (WAN – Wide Area Network)

- Si estendono per diversi Km, anche sull'intero pianeta (Internet)
- Sono costituite da
 - **end-point**: i computer utilizzati dagli utenti
 - **canali trasmissivi**: mezzi eterogenei impiegati per la trasmissione dei dati
 - **punti intermedi**: *router*, nodi di commutazione, necessari per veicolare i dati trasmessi da un end-point ad un altro attraverso la rete
- Tipicamente una rete WAN è utilizzata per collegare fra loro più reti LAN



Reti geografiche (WAN – Wide Area Network)

- I **router** sono gli apparati fondamentali per la realizzazione di una WAN
 - ricevono i pacchetti di dati
 - li memorizzano temporaneamente
 - li inviano verso la destinazione finale ad un nodo vicino appena il canale di trasmissione è disponibile
 - i **gateway** sono router che si fanno anche carico di problematiche di interconnessione tra reti eterogenee (per protocollo o media fisico trasmissivo)
- La tecnologia trasmissiva per le reti WAN è eterogenea (anche nell’ambito della stessa rete):
 - linee dedicate per collegare due nodi (CDN – Circuito Diretto Numerico)
 - linee telefoniche (DSL – Digitale Subscriber Line; ISDN – Integrated Services Digital Network)
 - ponti radio
 - canali satellitari

Il software per le reti

- Nella comunicazione tra computer sono due programmi a dialogare tra di loro
- Nell’ambito del dialogo tra i due programmi spesso si distinguono due ruoli:
 - **client**: il software che inizia il dialogo collegandosi ad un server per accedere ad un servizio o ad una risorsa;
 - **server**: il software che è in attesa della connessione da parte di uno o più client e che fornisce il servizio o la risorsa richiesta.
- Il modello delle reti **peer-to-peer** (letteralmente: “da pari a pari”) ciascun nodo (software) presente nella rete svolge simultaneamente il ruolo di server (offrendo servizi ad altri nodi) o di client (richiedendo servizi ad altri nodi)

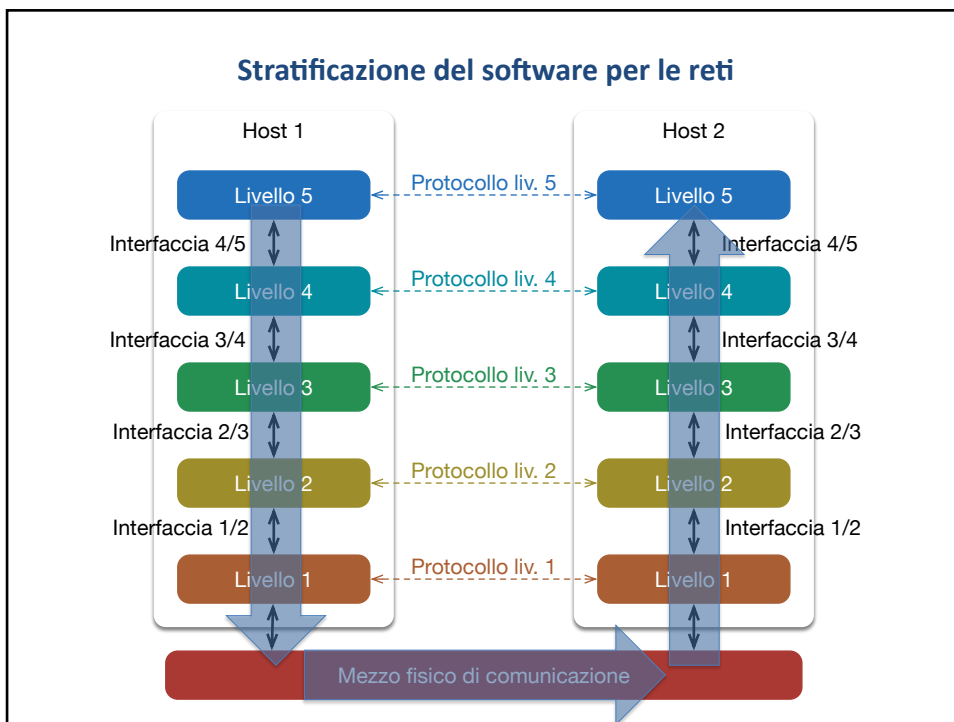
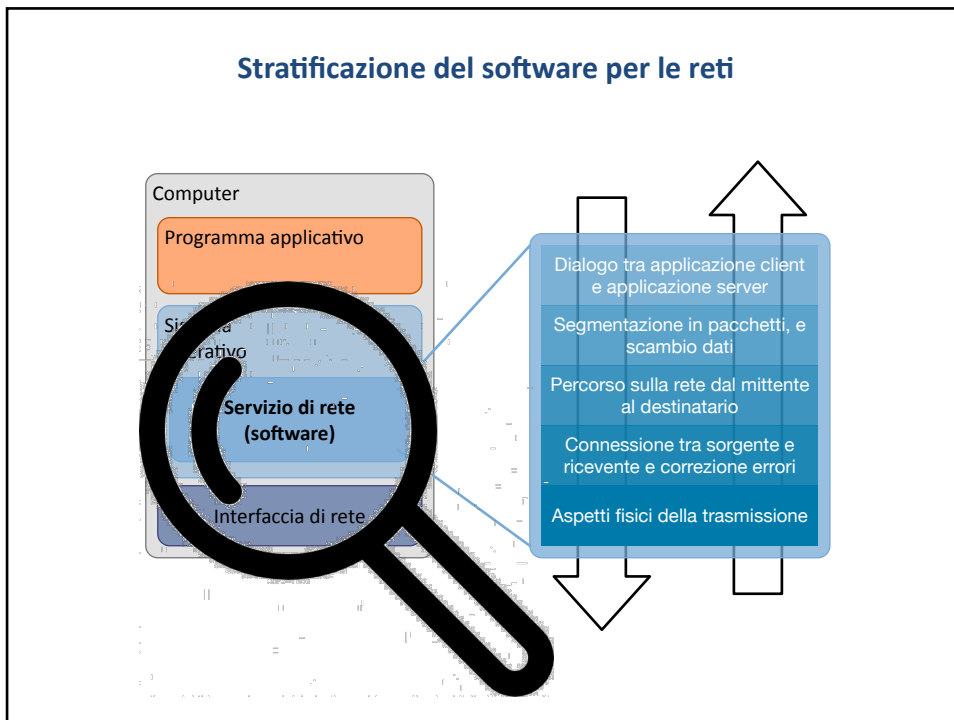
Il software per le reti

- Il software per la gestione delle comunicazioni in rete è estremamente **strutturato** e si compone di moduli **suddivisi per livelli**: ogni livello comunica solo con il livello soprastante e con quello sottostante
 - **Il livello più basso è quello fisico**, relativo alla gestione della trasmissione su un particolare media
 - **Il livello più alto è quello più astratto**, relativo alla interazione tra due applicazioni che operano su due *end-point* (due computer) distinti collegati in rete fra loro
- Una distinzione tra i diversi software/protocolli di gestione delle reti si effettua in base al numero di livelli di cui è composto e alla funzione assegnata a ciascun livello
- Lo scopo di ciascun livello è quello di **offrire dei servizi agli strati sovrastanti**, nascondendo la complessità relativa all'implementazione del servizio stesso
 - In questo modo è come se due livelli paritari, entrambi di livello n sui due nodi della rete, parlassero tra loro mediante un protocollo di livello n
 - In realtà i due componenti di livello n comunicano fra loro attraverso i servizi offerti dai componenti di livello $n-1$
 - Il dato viene costruito dal livello n del primo host
- Tra ogni coppia di livelli adiacenti è definita un'**interfaccia**, che definisce:
 - le **operazioni primitive** che possono essere richieste al livello sottostante
 - i **servizi** che possono essere offerti al livello sovrastante

Stratificazione del software per le reti

Il software per le reti è suddiviso in strati: ogni layer si occupa di una specifica problematica:

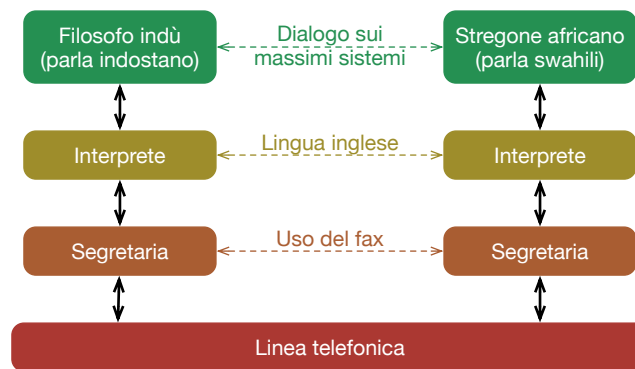
5. **Come fanno due programmi a comunicare tra loro?** Chi definisce le regole con cui avviene la comunicazione e quali dati devono essere trasmessi? Quale significato hanno?
4. Nella **comunicazione tra due programmi** come si svolge la "**sessione di comunicazione**"? Come si **segmenta** il dato da comunicare e come lo si **riasmonta** quando lo si riceve? **Chi apre la comunicazione, chi la tiene attiva e chi la chiude al termine della comunicazione?** Come si fa a tenere attive sessioni di comunicazione diverse con lo stesso computer? Come si distinguono tipi di comunicazione differenti (es.: pagine web, posta elettronica, chat, ecc.)?
3. Su una **rete**, non su un singolo segmento di collegamento tra due computer, come si fa a far arrivare i dati trasmessi dal mittente fino al giusto destinatario? **Quale percorso deve compiere la comunicazione?** Quale **cammino sul grafo** deve essere scelto per far comunicare due nodi? Come **identifico i nodi** della rete?
2. Come si **regola il flusso di dati** sul canale di comunicazione? Quanti dati devono essere inviati (o ricevuti) sul canale? Come ci si accorge di un **errore** di trasmissione e **come si rimedia ad un eventuale errore?**
1. Quale **mezzo di trasmissione** deve essere utilizzato? Come si **rappresentano i dati** sul canale di comunicazione analogico? Come si **attiva** e si **utilizza fisicamente** il canale?



Stratificazione del software per le reti

Metaforicamente è come se il software per le reti funzionasse come nella seguente figura in cui due filosofi stranieri devono comunicare tra di loro (tratto da G. Bongiovanni):

- Il filosofo sfrutta il servizio di traduzione dell'interprete che traduce e scrive un documento in inglese
- L'interprete sfrutta il servizio della segretaria per inviare il testo via fax
- La segretaria utilizza il fax come canale fisico di trasmissione



Stratificazione del software per le reti

- La comunicazione è **segmentata in pacchetti** in modo che
 - il canale trasmissivo non sia impegnato continuamente per tempi molto lunghi
 - in caso di errore di trasmissione debba essere rispedito solo un pacchetto e non l'intero flusso di dati
- In fase di trasmissione, ogni livello aggiunge al pacchetto ricevuto dallo strato sovrastante un **header**, formando così un pacchetto più grande (il pacchetto viene "imbustato")
- L'header contiene **informazioni di servizio** utili per lo scambio del pacchetto di dati
 - lunghezza del pacchetto
 - numero progressivo del pacchetto nell'ambito dell'intero flusso di comunicazione
 - data ed ora in cui il pacchetto è stato prodotto (*timestamp*)
 - ...
- In fase di ricezione ogni livello riceve un pacchetto "imbustato" dal livello inferiore: il livello n scorpora l'**header** del pacchetto dal segmento contenente i dati veri e propri (il pacchetto viene estratto dalla busta di livello n)
 - il livello n esegue anche dei **controlli di integrità del pacchetto** sulla base delle informazioni contenute nell'**header**
 - il ricevitore **ricostruisce il flusso di comunicazione originario concatenando i pacchetti ricevuti** non in base all'ordine di ricezione, ma in base al numero progressivo presente nell'**header** del pacchetto

Modelli e architetture delle reti

- Esistono *modelli teorici* per la definizione di un'architettura di una rete e *architetture effettive* che di fatto costituiscono uno standard nel mondo delle reti
- **Modello OSI** (*Open Systems Interconnection*)
 - definisce il numero, le caratteristiche e le relazioni funzionali tra i livelli di un software di rete, ma non ne definisce i protocolli effettivi
 - consiste in **7 livelli**, forse un po' troppo astratti e di difficile implementazione (ricalca IBM SNA)
- **Architettura TCP/IP** (*Transmission Control Protocol / Internet Protocol*)
 - definisce, livello per livello, anche i protocolli effettivi
 - consiste in **4 livelli** (semplificando il modello OSI)

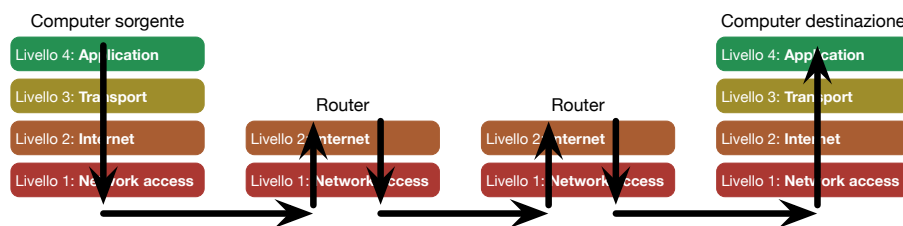
Modello OSI	TCP/IP
Livello 7: Application	Livello 4: Application
Livello 6: Presentation	
Livello 5: Session	
Livello 4: Transport	Livello 3: Transport
Livello 3: Network	Livello 2: Internet
Livello 2: Data Link	
Livello 1: Fisico	Livello 1: Network access

Lo "stack" TCP/IP

- L'architettura di rete TCP/IP è quindi un insieme di protocolli che operano coordinati fra di loro a vari livelli di uno *stack* costituito da quattro livelli
- Livello 1: **Network access**
 - protocolli fisici per reti LAN e WAN (es.: Ethernet, DSL, ISDN, PPP, ecc.); include il livello Data Link
- Livello 2: **Internet**
 - protocollo **IP** (*Internet Protocol*) per l'indirizzamento e il routing dei pacchetti sul canale fisico verso la destinazione finale
- Livello 3: **Transport**
 - protocollo **TCP** (*Transmission Control Protocol*): frammenta la comunicazione in pacchetti in uscita e riassembla i pacchetti in ingresso per ricostruire il flusso di dati originari; garantisce la consegna del pacchetto
 - protocollo **UDP** (*User Datagram Protocol*): frammenta in pacchetti, ma non garantisce l'ordine né la consegna del pacchetto
- Livello 4: **Application**
 - protocolli applicativi per la comunicazione tra un programma client ed un programma server: **FTP** (*File Transfer Protocol*), **Telnet** (emulazione di terminale per connessioni remote), **SMTP** (*Simple Mail Transfer Protocol*), **POP3** (*Post Office Protocol v. 3*), **DNS** (*Domain Name Service*), **HTTP** (*HyperText Transfer Protocol*), **SOAP** (*Simple Object Access Protocol*), ecc.

Lo "stack" TCP/IP

- Lo stack TCP/IP viene implementato per intero nel sistema operativo di un computer, su cui gira anche l'applicazione "end-user" che è l'elemento sorgente o di destinazione della comunicazione
- Su apparati di rete intermedi (router, gateway, ...) il protocollo viene implementato parzialmente, senza spingersi fino al livello Application, che sarebbe inutile per tali apparati



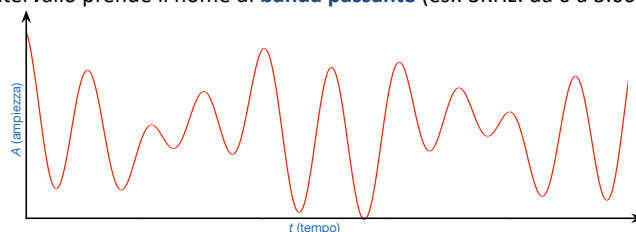
Lo stack OSI semplificato

- Per descrivere i diversi aspetti del protocollo di rete adottiamo il **modello OSI semplificato**, ridotto a soli cinque livelli:
 1. Livello Fisico
 2. Livello Data Link
 3. Livello Network
 4. Livello Transport
 5. Livello Application
- Nel TCP/IP il livello **Network Access** include sia il livello **Fisico** che il **Data Link** del modello OSI semplificato

Livello 1: Fisico

Livello 1

- Il canale di comunicazione fisico per la realizzazione di reti di computer è un canale **analogico**, in cui i valori variano in modo continuo (energia elettrica, campo elettromagnetico, luce, ecc.) all'interno di un determinato intervallo detto **ampiezza**
- Il segnale trasmesso su un canale analogico può essere rappresentato mediante una **funzione sinusoidale non periodica** che, con gli strumenti dell'analisi di Fourier, può essere interpretata come **somma di funzioni trigonometriche** con ampiezza e periodo (e frequenza) diverse
- Un segnale è quindi caratterizzato da un intervallo di frequenze delle funzioni sinusoidali che lo compongono: tale intervallo si chiama **banda di frequenza del segnale**
- Anche il mezzo fisico di trasporto del segnale è caratterizzato da un intervallo di frequenze delle funzioni che compongono il segnale che può essere trasmesso da tale mezzo fisico; questo intervallo prende il nome di **banda passante** (es.: 3KHz: da 0 a 3.000 HZ)



Banda passante e velocità di trasmissione

Livello 1

- **Harry Nyquist** (1889-1976) nel 1927 dimostrò che un segnale analogico di banda h (da 0 a h Hz) può essere ricostruito mediante una campionatura eseguita $2h$ volte al secondo
- Un segnale di banda h consente di rappresentare per ogni secondo una quantità di informazione con un numero di bit pari a $2h \log_2 V$, dove V è il numero di possibili valori del segnale (possibili valori discreti del segnale)
- Quindi con un segnale binario ($V = 2$) con una banda di 3 KHz, il numero di bit trasmessi al secondo è pari a $2 \times 3000 \times 1 = 6.000$ bit per secondo (6 Kbps)
- Per trasmettere ad una velocità superiore a 6 Kbps si escogitano delle codifiche del segnale che sfruttano un numero di valori del segnale $V > 2$ (nella pratica V è un valore piuttosto elevato)
- **Claude Shannon** (1916-2001), padre della *Teoria dell'Informazione*, introdusse il concetto di **rumore** su un canale di comunicazione e formulò un celebre teorema (*Secondo Teorema di Shannon*, 1948) in cui affermò che la massima quantità di informazione trasferibile in modo affidabile su un canale affetto da rumore è limitato da una certa soglia che prende il nome di **capacità di canale**
- Indicando con S/N il rapporto tra segnale e rumore, secondo il Teorema di Shannon la massima velocità trasmissiva del canale con banda passante di h Hz è dato da $h \log_2 (1+S/N)$

Livello 2: Data Link

Livello 2

- Il livello **Data Link** offre una **comunicazione affidabile ed efficiente tra due nodi direttamente connessi** da un canale fisico (un cavo Ethernet, una linea telefonica, ecc.)
 - offre servizi al livello Network (sovrastante)
 - determina la **suddivisione in frame** dei bit a livello fisico
 - gestisce gli **errori** di trasmissione (verifica integrità dei propri pacchetti spediti, altrimenti li ritrasmette, ad esempio tramite l'algoritmo backoff)
 - regola il flusso di dati tra sorgente e destinatario
- Questo livello dello *stack* è tipicamente implementato nel firmware (o nei driver) di una scheda di rete Ethernet o di un'interfaccia wi-fi; i protocolli **PPP** (*Point-to-Point Protocol*) e **SLIP** (*Serial Line Internet Protocol*) sono protocolli del livello Data Link implementati via software
- Il livello Data Link in trasmissione:
 - riceve dal livello Network un pacchetto di dati (formato binario)
 - frammenta la sequenza di bit in **frame** dipendenti dal particolare protocollo fisico di trasmissione
 - calcola un **checksum** di ciascun frame
 - passa il pacchetto formato dal **frame** e dal **checksum** al livello Fisico sottostante, che lo spedisce come sequenza di bit
- Il livello Data Link in ricezione:
 - riceve una sequenza di bit dal livello fisico
 - ricostruisce i frame uno dopo l'altro
 - ricalcola il checksum e verifica che sia uguale a quello contenuto nel frame

Livello 3: Network / Internet

Livello 3

- Il livello Network si occupa di **trasferire il pacchetto di dati dalla sorgente alla destinazione passando attraverso una serie di nodi intermedi** (routing, instradamento dei pacchetti)
- Il livello Network si occupa di
 - conoscere la topologia della rete
 - scegliere di volta in volta il cammino migliore per "instradare" i pacchetti (*routing*)
 - gestire il flusso dei dati e la congestione della rete
 - gestire problematiche relative alla interconnessione di reti diverse
- Il livello Network è particolarmente sofisticato sugli apparati router, proprio per l'implementazione di un algoritmo di routing efficiente
- Gli algoritmi di routing sono spesso basati sui seguenti principi
 - **algoritmi di cammino minimo**: il pacchetto viene instradato verso la via che richiede un numero inferiore di *hop* (passaggi per nodi intermedi) per giungere al router di destinazione
 - **algoritmi di flooding**: il pacchetto viene inviato su tutti i cammini uscenti dal router, tranne quello attraverso cui il pacchetto è arrivato al router
 - **algoritmi flow-based**: viene calcolato il traffico sulle diverse linee uscenti dal router, tenendo conto della loro capacità trasmissiva e, in base a questi dati, si stabilisce su quale linea inviare il pacchetto

Protocollo TCP/IP e indirizzi IP

Livello 3

- Il livello 3 "Network / Internet" nel protocollo TCP/IP è realizzato mediante il **protocollo IP** (*Internet Protocol*) che si occupa di:
 - Traffico in uscita:
 - ricevere i dati dal livello 4 di Trasporto e incapsularli in pacchetti IP (max 64 KByte, tipicamente 1.500 Byte)
 - instradare i pacchetti sulla rete mediante il livello Data Link
 - Traffico in ingresso
 - riassemblare i pacchetti ricevuti dai livelli sottostanti (possono essere frammentati)
 - spacchettare il pacchetto IP, verificarlo ed estrarre i dati del livello Trasporto
 - consegnare i dati al livello trasporto nell'ordine in cui questi sono arrivati (che può essere un ordine diverso da quello con cui sono stati spediti)
- Un **pacchetto IP** è una sequenza di byte costituita da un *header* e da un segmento di dati
- La parte *header* contiene:
 - la *versione* del protocollo
 - la *lunghezza dell'header* e la lunghezza totale del pacchetto
 - un *identificativo del pacchetto* per ricostruirlo a partire da frammenti più piccoli e un progressivo del frammento nel caso il pacchetto IP debba essere ricostruito assemblando più frammenti
 - il *time to live*, ossia un contatore che viene decrementato di uno ad ogni passaggio per un nodo intermedio nella rete: quando arriva a zero il pacchetto viene scartato
 - indirizzo del *mittente* e del *destinatario*
 - altre informazioni opzionali

Indirizzi IP

Livello 3

- Un indirizzo IP è composto da 32 bit, suddivisi in 4 gruppi da 8 bit (4 byte)
- L'indirizzo viene assegnato ad ogni interfaccia di un nodo della rete per identificarlo **univocamente**; contiene due informazioni distinte:
 - l'**identificativo della rete** in cui si trova l'host
 - l'**identificativo dell'host** all'interno della rete
- Uno stesso host può essere identificato da più indirizzi IP
- Gli indirizzi che identificano le reti connesse direttamente ad Internet (reti che sono una porzione della rete Internet) vengono **assegnati** alle organizzazioni che ne fanno richiesta (aziende, enti pubblici, associazioni, ecc.) da organismi nazionali detti **NIC** (*Network Information Center*)
- A livello mondiale il registro di assegnazione degli indirizzi IP viene gestito da **ICANN** (*Internet Corporation for Assigned Names and Numbers*) e dalla sua organizzazione controllata denominata **IANA** (*Internet Assigned Numbers Authority*)
- In Europa il registro degli indirizzi IP è gestito da **RIPE NCC** (*Réseaux IP Européens – Network Coordination Centre*); in Italia il registro è gestito da **NIC-IT**
- Sono disponibili **identificativi per reti IP private**, non connesse **direttamente** ad Internet: per l'uso di identificativi riservati alle reti private non è necessario ottenere alcuna autorizzazione da enti di gestione

Livello 3

Indirizzi IP

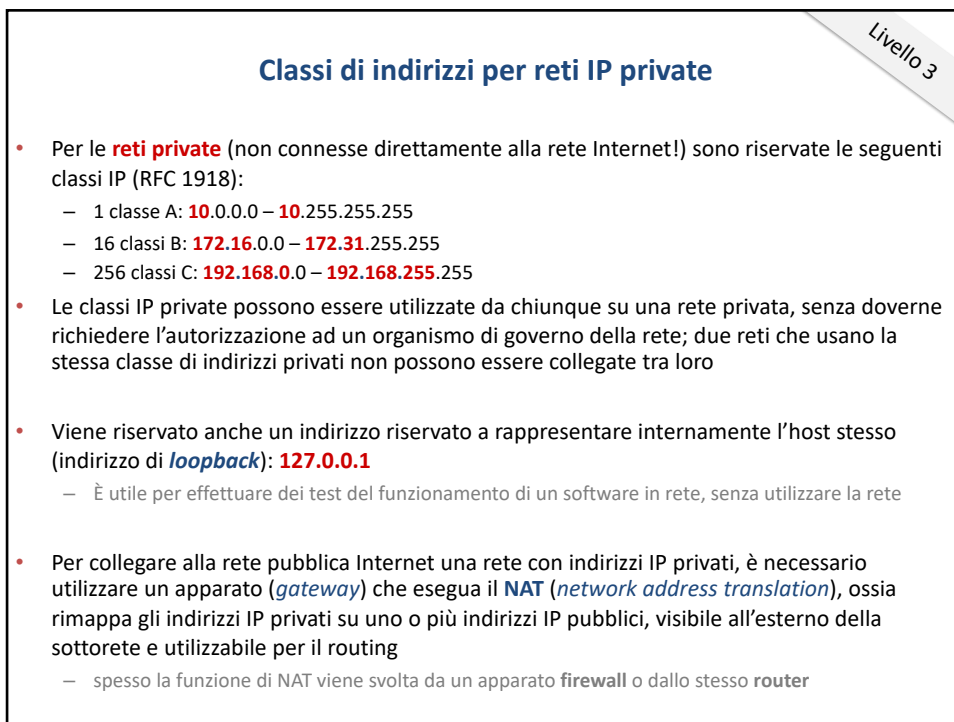
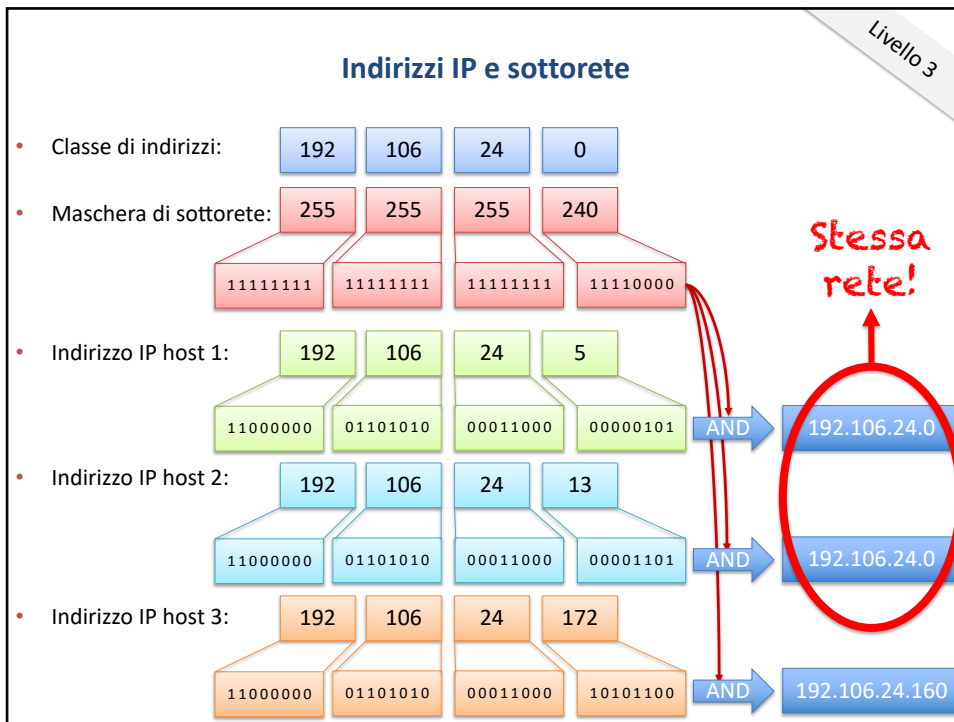
- L'insieme dei **32 bit** che costituiscono l'indirizzo IP viene ripartito tra bit riservati all'**identificazione della rete** e bit riservati all'**identificazione dell'host** in base al numero di host che devono essere indirizzati in una stessa rete
 - se devo assegnare indirizzi a molti host, allora riserverò più bit all'identificazione degli host e meno bit all'identificazione della rete
 - in questo modo si hanno poche reti distinte contenenti ciascuna moltissimi indirizzi IP e molte reti che invece contengono un numero piuttosto limitato di indirizzi
- Gli indirizzi di rete sono distinti in **tre diverse classi**, a seconda del numero di bit riservati all'identificazione della rete o degli host:
 - 126 reti di **classe A**, che possono contenere 16 milioni di host ciascuna
 - 16.382 reti di **classe B**, con circa 64.000 host ciascuna
 - 2 milioni di reti di **classe C**, con 254 host ciascuna

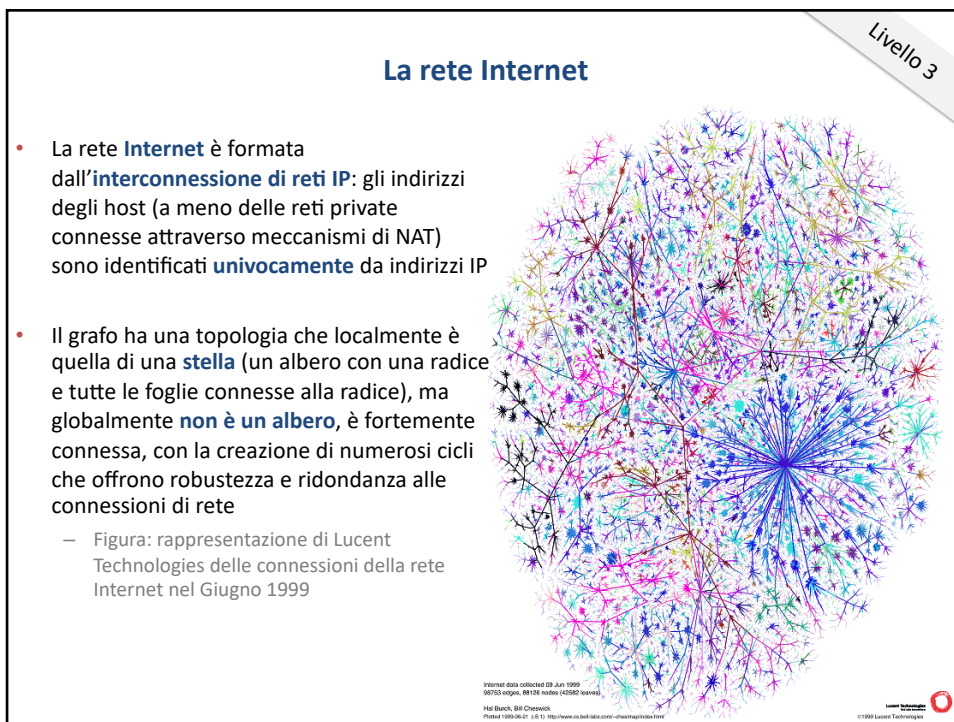
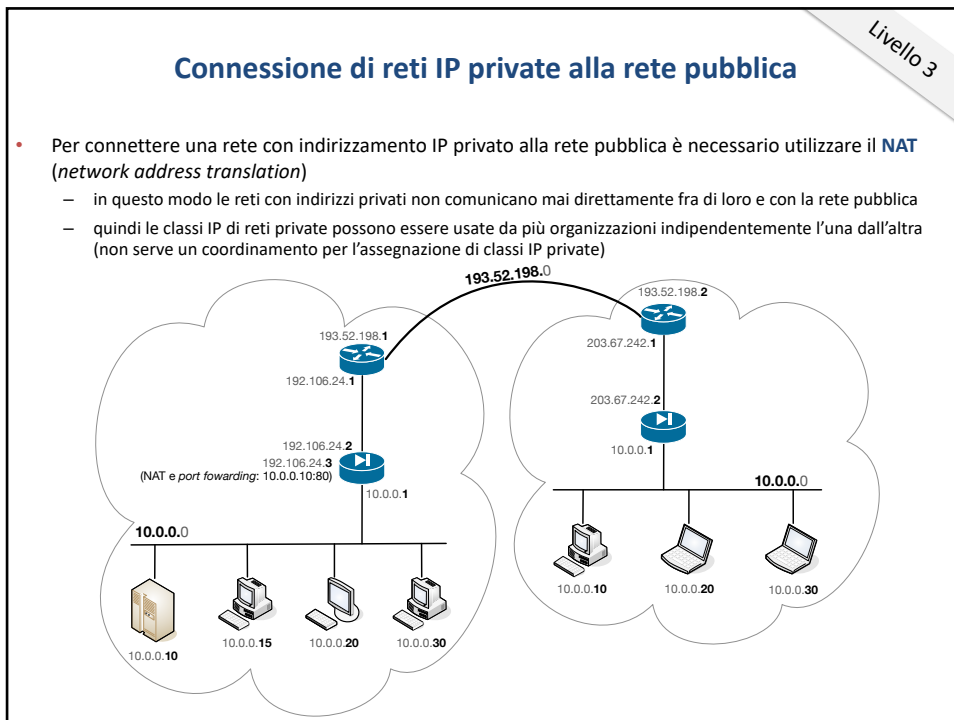
		8 bit	8 bit	8 bit	8 bit	
Classe A	0	rete (7 bit)	host (24 bit)	host (8 bit)	host (8 bit)	Es.: 87.23.134.216 = 01010111.00010111.10000110.11011000
Classe B	1 0	rete (14 bit)	host (16 bit)	host (8 bit)	host (8 bit)	Es.: 157.76.134.216 = 10011101.01001100.10000110.11011000
Classe C	1 1 0	rete (21 bit)	host (8 bit)	host (8 bit)	host (8 bit)	Es.: 215.138.97.216 = 11010111.10001010.01100001.11011000

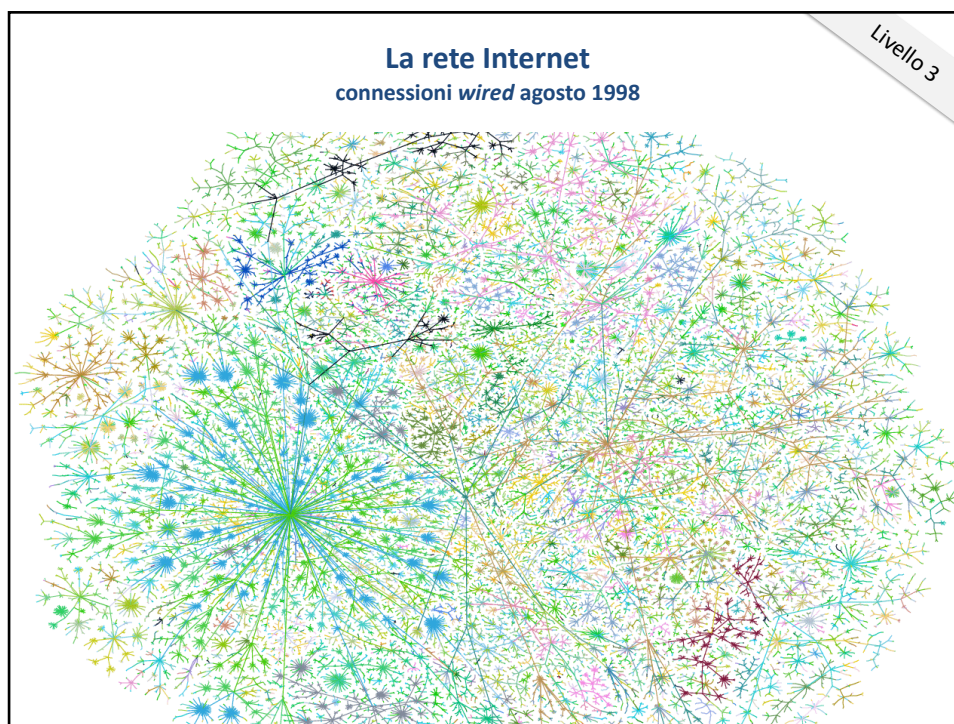
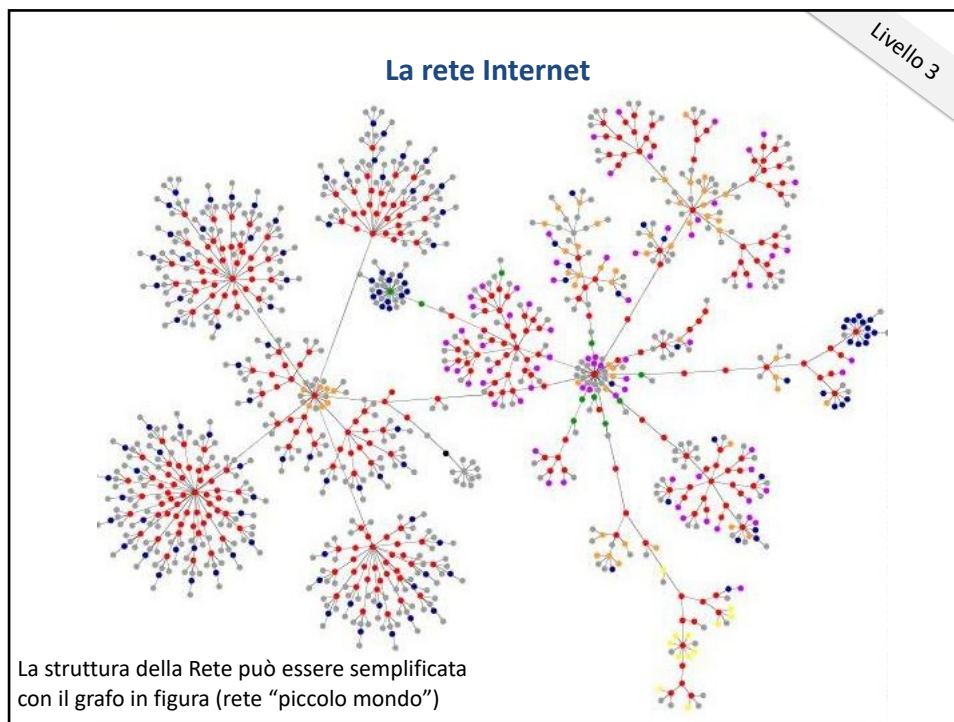
Livello 3

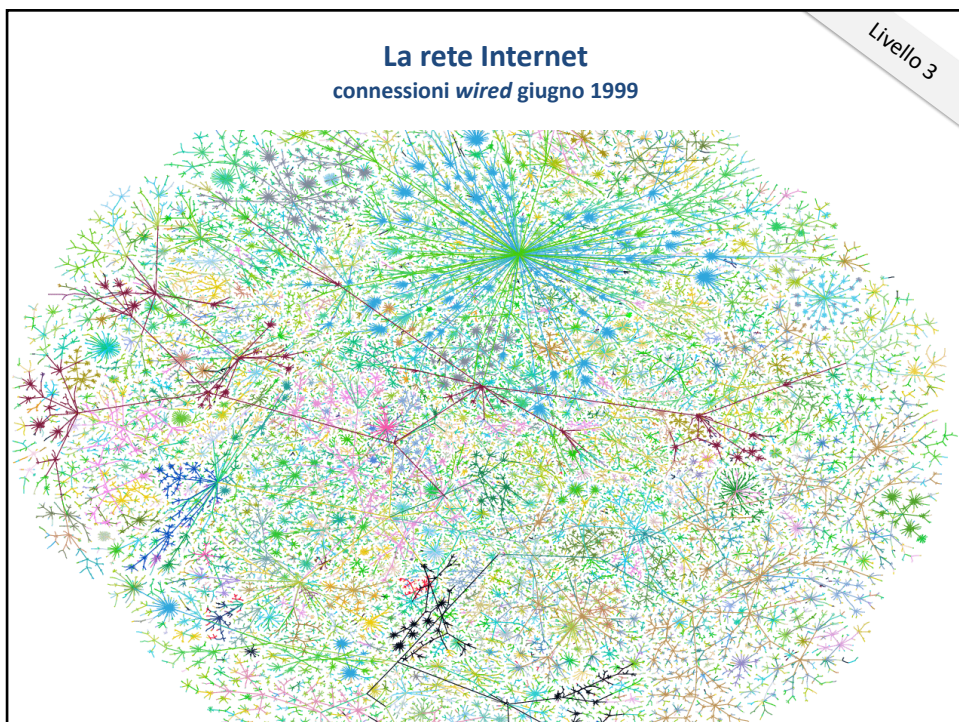
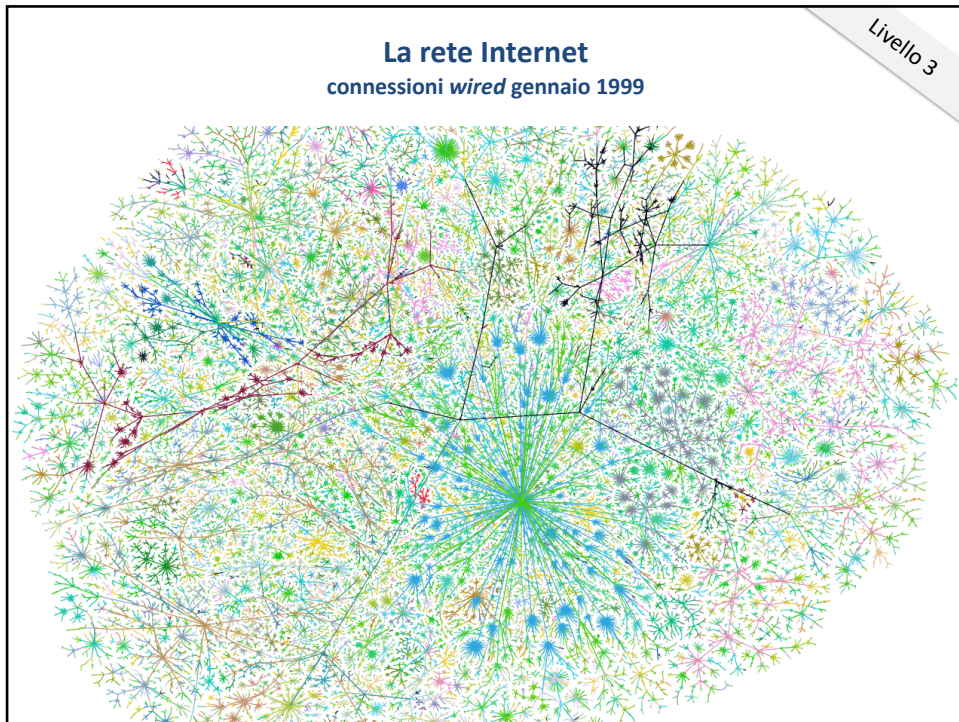
Indirizzi IP e sottorete

- Il sistema operativo di ogni host è configurato con l'indicazione di
 - indirizzo IP** dell'host
 - maschera di sottorete**, per ricavare l'identificativo della rete
 - l'indirizzo del **gateway di default**, a cui inviare i pacchetti IP quando il destinatario non appartiene alla stessa rete dell'host
- Maschera di sottorete, esempio: 255.255.255.0
 - eseguendo un'operazione di **AND** sui bit dell'indirizzo IP del mittente e del destinatario si ottiene la stessa stringa binaria solo se i due host appartengono alla stessa rete









Livello 3

La rete Internet: crescita inarrestabile

- La rete (il numero di nodi connessi) **continua a crescere in modo inarrestabile** e ben presto, con la "Internet delle Cose" (IoT – *Internet of Things*) saranno connessi in rete singoli oggetti di uso comune (orologi, elettrodomestici, il collare del cane, ecc.)
- Lo "spazio di indirizzamento" di IP v.4 è troppo limitato, si dovrà passare ad un nuovo "spazio di indirizzamento", più esteso, denominato **IP v.6**
 - indirizzi di 16 byte: 2^{128} indirizzi IP (sono tantissimi: la superficie della Terra è di 510.072.000 Km², quindi avremo a disposizione in media $6,7 \times 10^{23}$ indirizzi IP per ogni m²)
 - semplificazione dell'header: 7 campi invece di 13
 - funzioni di autenticazione e privacy dei dati, basate sulla crittografia

Year	Global population (billions)	Internet users (billions)
2000	~6.1	~0.2
2010	~6.9	~1.6
2020	~7.5	~5.0

Livello 4

Livello 4: Transport

- Il livello Trasporto è un'astrazione del livello Network e consente di **stabilire connessioni tra programmi (client e server) e attuare lo scambio di dati**, senza curarsi degli aspetti fisici o di routing gestiti dai livelli sottostanti
- Nell'ambito dell'architettura TCP/IP sono definiti due protocolli del livello 4:
 - **TCP**, *Transmission Control Protocol*
 - **UDP**, *User Datagram Protocol*
- UDP è un protocollo semplice e "**non affidabile**" (non garantisce la consegna del pacchetto di informazione)
- TCP offre un servizio "**affidabile**", ossia garantisce la consegna del dato (mascherando la complessità e la reiterazione di operazioni che questa affidabilità comporta) anche su una rete non affidabile
- Il protocollo TCP si occupa di
 - trasmettere dati in uscita:
 - accettare dati dal livello Application
 - spezzarli in *segmenti* (dimensione massima 64 KByte, tipicamente circa 1.500 Byte)
 - consegnarli al livello Network, eventualmente ritrasmettendoli
 - ricevere dati in ingresso:
 - ricevere segmenti dal livello Network
 - rimetterli in ordine, eliminando buchi e doppioni
 - consegnare i dati, in ordine, al livello Application

Il protocollo TCP e i socket

Livello 4

- La connessione creata dal protocollo TCP al Livello 4 dello stack, è definita identificando i due punti di accesso detti **socket**
- Ogni **socket** ha un **socket number** che consiste nella coppia "IP address : port number" (es.: 193.204.165.101:80)
- Il **port number** è un numero da 16 bit (0-65.535); i valori tra 0 e 255 identificano i **port number** di servizi standard (*well-known TCP port*)
- Il sistema operativo offre delle interfacce API per le chiamate di sistema per l'apertura di connessioni verso specifici **socket** (indirizzi IP e porte TCP)
- Per garantire l'affidabilità della connessione, il protocollo si occupa della fase di connessione con un *handshaking* tra i due nodi, sia in fase di connessione che di chiusura della connessione

Uso dei socket per comunicazione tra client e server

Livello 4

Il sistema operativo offre una API (*application programming interface*) per l'utilizzo dei **socket** da programma con una serie di *system call* dedicate all'esecuzione delle diverse operazioni

in questa fase l'applicazione server è dedicata alla comunicazione con una sola applicazione client

se il server deve gestire più connessioni contemporanee, allora per ogni connessione da parte di un client deve generare un processo figlio (*fork*)

Livello 5: Application

Livello 5

- Il livello **Application** è quello in cui effettivamente due applicazioni (un *client* ed un *server*) che operano su due nodi della rete (o anche sullo stesso nodo) dialogano inviandosi dei dati, sfruttando tutti i livelli sottostanti per rendere effettivo il processo di comunicazione
- A questo livello sono definiti numerosi protocolli applicativi:
 - **protocolli di servizio**, utilizzati per consentire o semplificare il corretto funzionamento della comunicazione in rete (es.: DNS, SNMP, ...)
 - **protocolli standard** per servizi comuni (es.: emulazione di terminale, posta elettronica, web, ...)
 - **protocolli proprietari**, definiti da chi ha progettato un determinato software, per far dialogare la componente client con quella server (es.: un programma di gestione contabile, ...)
- Un protocollo applicativo è costituito da un insieme di **comandi** e da una **sintassi** che definisce la modalità con cui i comandi possono essere scambiati tra client e server e il formato che devono rispettare i dati e i parametri forniti ai comandi

DNS – Domain Name System (TCP port 53)

Livello 5

- L'utilizzo degli indirizzi IP da parte degli utenti di una rete TCP/IP è piuttosto scomodo, per questo motivo è stato definito un sistema dei nomi di dominio, per identificare più facilmente i nodi della rete
- Il sistema DNS è costituito da
 - uno **schema gerarchico di definizione dei nomi**, basato sul concetto di **dominio**
 - un **database distribuito** che memorizza e rende disponibile l'insieme dei nomi
 - un **protocollo di comunicazione** per l'interrogazione del database e l'aggiornamento dei nomi associati ai nodi della rete (che sono comunque identificati da un indirizzo IP)
- I nomi associati agli host sono utilizzati dagli utenti e dalle applicazioni, ma non possono sostituire l'indirizzo IP su cui si basa il Livello Network (o Internet); per passare dal nome all'indirizzo IP si seguono questi passi:
 1. l'applicazione che deve eseguire la connessione verso un host di cui conosce il nome (es.: www.google.com) utilizza la funzione API per la risoluzione dei nomi che invia una richiesta di traduzione ad un DNS server locale
 2. il DNS server locale se ha nel proprio database l'indirizzo IP associato a quel nome lo restituisce al client, altrimenti chiede ad un DNS server di livello superiore
 3. l'applicazione riceve come risposta l'indirizzo IP associato al nome e lo usa per aprire la connessione TCP verso tale host

DNS – Domain Name System (TCP port 53)

Livello 5

- I nomi host associati ad un indirizzo IP hanno la seguente struttura:
 $hostname.subdomain.domain.top-level-domain$
 come ad esempio:
 $ciop.mat.uniroma3.it$
- Nell'esempio si riconoscono le componenti del nome separate da punti (quindi le componenti del nome non devono contenere il carattere "."):
 - hostname:** "ciop", è il nome che identifica univocamente il nodo di rete in un determinato sottodominio
 - subdomain:** "mat", è il sottodominio in cui è inserito l'host; nell'esempio è il dominio assegnato al Dipartimento di Matematica dell'Università Roma Tre
 - domain:** "uniroma3", è il dominio a cui appartiene l'host; nell'esempio è il dominio assegnato all'Università degli Studi Roma Tre
 - top-level-domain:** "it", è il dominio di primo livello che è assegnato all'autorità italiana di gestione dei nomi di dominio Internet
- Un nome di host che comprenda tutte le componenti dal nome host fino al top-level-domain è un **FQDN**: *fully qualified domain name*
- Ad uno stesso FQDN può essere associato un solo indirizzo IP, mentre uno stesso indirizzo IP può essere associato a più FQDN: la traduzione è univocamente determinata solo nel passaggio da nome di dominio a indirizzo IP (non il viceversa)

DNS – Domain Name System (TCP port 53)

Livello 5

- Il database del DNS è un database "gerarchico": ICANN e IANA gestiscono la radice dell'albero e demandano ad altre organizzazioni la gestione dei top-level-domain:
 - Country Code TLD** (ccTLD): ".it", ".fr", ".uk", ".es", ".de", ".tv", ... Sono TLD assegnati alle autorità nazionali di gestione della rete Internet (NIC-IT in Italia)
 - Generic TLD** (gTLD): ".com", ".net", ".org", ".edu", ".gov", ".mil", ... Sono TLD generici, non corrispondente ad uno specifico Paese
- Singole organizzazioni (enti, aziende, associazioni, singoli cittadini) possono richiedere l'assegnazione di un **nome di dominio di secondo o terzo livello** ad un Ente di Registrazione accreditato presso l'autorità che gestisce il TLD
- Per gestire un dominio (di primo, secondo, terzo livello) è necessario disporre di un **DNS server**, configurato per gestire il database degli indirizzi IP associati ai nomi host del dominio di propria competenza
- La definizione e l'assegnazione dei nomi host alle macchine del proprio dominio, viene definita autonomamente dall'organizzazione intestataria del dominio stesso, configurando opportunamente i propri server DNS

Logo dell'autorità italiana
di registrazione dei domini
sotto al ccTLD ".it"
www.nic.it

Registroit

Livello 5

DNS – Domain Name System (TCP port 53)

- Due sono i tool software più comuni per l'interrogazione dei database DNS e del registro dei nomi di dominio:
 - **nslookup**: è un software client per interrogare un DNS server e richiedere la risoluzione di un hostname nell'indirizzo IP corrispondente
 - **whois**: è un client per interrogare i database "whois" delle autorità di registrazione dei domini

```
marco@aquilante ~$ whois uniroma3.it
Domain:                uniroma3.it
Status:                ok
Created:               1996-01-29 00:00:00
Last Update:          2015-02-14 00:52:34
Expire Date:           2016-01-29

Registrant
  Organization:        Universita' "Roma Tre"
  Address:              Via Ostiense, 159
                      Roma - 00154 - RM - IT
  Created:              2007-03-01 10:54:28
  Last Update:         2014-10-28 16:54:06
  ...
Nameservers
  ns.uniroma3.it
  dns.uniroma3.it
```

```
marco@aquilante ~$ nslookup
> server ns.uniroma3.it
Default server: ns.uniroma3.it
Address: 193.204.167.146#53
> www.uniroma3.it.
Server:      ns.uniroma3.it
Address: 193.204.167.146#53

Name:      www.uniroma3.it
Address: 193.205.139.223
> exit
```

Livello 5

FTP – File Transfer Protocol (TCP port 21)

- Il protocollo **FTP** consente di scambiare file tra un host server ed un host client
- I file scambiati possono essere file di testo ASCII o file binari (programmi eseguibili, immagini, file in formato compresso, ecc.)
- Il server FTP mette a disposizione del client un filesystem remoto, strutturato ad albero con directory e sottodirectory
- Alcuni dei comandi implementati dal protocollo sono i seguenti:
 - **user**: autenticazione
 - **put**: upload file (da client a server)
 - **get**: download file (da server a client)
 - **list/dir**: elenco file in una directory
 - **cd/lcd**: cambio directory corrente (sul server/sul client)
 - **bin**: switch tra trasferimento file binari o testo ASCII
 - **hash**: switch per la visualizzazione della progressione del trasferimento dati

The diagram illustrates the FTP architecture. On the left, a 'Server FTP' box contains a 'Server Protocol Interpreter' and a 'Server Data Transfer Process'. On the right, a 'Client FTP' box contains a 'User Interface', a 'User Protocol Interpreter', and a 'User Data Transfer Process'. A 'File System' is connected to the 'Server Data Transfer Process'. Bidirectional arrows between the 'Server Protocol Interpreter' and 'User Interface' are labeled 'Comandi FTP' and 'Risposte FTP'. Bidirectional arrows between the 'Server Data Transfer Process' and 'User Data Transfer Process' are labeled 'Trasferimento Dati'. A 'File System' is also shown on the right side of the diagram.

FTP – File Transfer Protocol (TCP port 21)

Livello 5

Il protocollo FTP gestisce il concetto di **sessione**:

- nell'ambito di uno stesso collegamento da un client verso un server, l'utente può modificare lo stato della configurazione (es.: può modificare la directory corrente sul server FTP) e può dialogare con una successione di comandi che tengono conto dell'effetto del comando precedente

```
marco@aquilante ~$ ftp
ftp> open ftp.freebsd.org
Trying 193.162.146.4...
Connected to ftp.geo.freebsd.org.
Name (ftp.freebsd.org:marco): anonymous
331 Guest login ok, send your email address as password.
Password: *****
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> dir
150 Opening ASCII mode data connection for '/bin/ls'.
total 12
drwxr-xr-x  3 1006 1006  512 Nov 27  2012 pub
drwxr-xr-x  3 1006  0   512 Jul 21  2014 www
226 Transfer complete.
ftp> cd pub/FreeBSD/ports/i386/packages-9-stable/x11-clocks
250 CWD command successful.
ftp> bin
200 Type set to I.
ftp> hash
Hash mark printing on (1024 bytes/hash mark).
ftp> get xclock-1.0.7.1.tbz
local: xclock-1.0.7.1.tbz remote: xclock-1.0.7.1.tbz
229 Entering Extended Passive Mode (||65212|)
150 Opening BINARY mode data connection for 'xclock-1.0.7.1.
#####
226 Transfer complete.
29118 bytes received in 00:00 (160.06 KiB/s)
ftp> quit
221 Goodbye.
```

Telnet – Emulazione di terminale (TCP port 23)

Livello 5

- È un protocollo nato per scopi piuttosto generali:
 - <<... fornire un supporto per le comunicazioni sufficientemente generalizzato, bidirezionale ed orientato ai byte ...>>
- È utilizzato per stabilire connessioni via rete in **emulazione di terminale** da un host remoto verso un server (oggi è scoraggiato l'uso di Telnet, perché comunica "in chiaro")
- Per sua stessa natura, tuttavia, può essere utilizzato per connettersi alla porta TCP di un qualsiasi servizio ed utilizzare il client per scambiare messaggi su altri protocolli

```
marco@aquilante ~$ telnet gondrano
Trying 10.211.55.42...
Connected to 10.211.55.42.
Escape character is '^]'.

FreeBSD/i386 (gondrano.aquilante.net) (pts/0)

User: liverani
Password: *****
Last login: Sun Mar  8 21:51:03 from localhost
FreeBSD 10.0-RELEASE (GENERIC) #0 r260789: Fri Jan 17 01:46:25 UTC 2014

liverani@gondrano:~$ logout
Connection closed by foreign host.
marco@aquilante ~$
```

Livello 5

SMTP – Simple Mail Transfer Protocol (TCP port 25)

- È il protocollo utilizzato per **trasmettere messaggi di posta elettronica** da un **MTA (mail transfer agent)** ad un altro
- Per ogni dominio Internet deve essere definito (mediante uno o più **record di tipo MX** sul server DNS) almeno un **Mail eXchanger**, ossia un host che, mediante un software MTA che utilizza il protocollo SMTP, è in grado di ricevere posta elettronica per il dominio stesso
- L'architettura del servizio di posta elettronica può essere schematizzato come in figura

```

    graph LR
        MUA1[MUA (mail user agent)] -- SMTP --> MTA1[MTA (mail transfer agent)]
        MTA1 --- Cloud((SMTP))
        Cloud --- MTA2[MTA (mail transfer agent)]
        MTA2 --- Mailbox[(Mailbox)]
        MTA2 -- POP3 --> MUA2[MUA (mail user agent)]
        MUA1 --- Mittente[Mittente]
        MUA2 --- Destinatario[Destinatario]
    
```

Livello 5

SMTP – Simple Mail Transfer Protocol (TCP port 25)

- Il protocollo SMTP gestisce una **sessione** di lavoro tramite la quale possono essere inviati più messaggi di posta elettronica

```

marco@aquilante ~$ telnet mail.pippo.com 25
220 mail.pippo.com ESMTP Postfix
HELO aquilante.net
250 Hello aquilante.net, pleased to meet you
MAIL FROM: <marco@aquilante.net>
250 marco@aquilante.net ... Sender ok
RCPT TO: <marina@pippo.com>
250 marina@pippo.com ... Recipient Ok
DATA
354 End data with "." on a line by itself
Subject: messaggio di prova
From: marco@aquilante.net
To: marina@pippo.com

Ciao,
questa e' una prova.
.
250 Ok: queued as 12345
QUIT
221 Bye
    
```

Livello 5



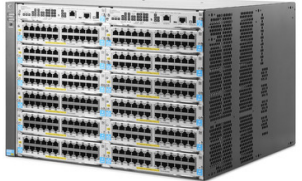
POP3 – Post Office Protocol v.3 (TCP port 110)

- È uno dei protocolli che consente ad un software client di posta elettronica (Microsoft Outlook, Apple Mail, Mozilla Thunderbolt, ecc.) di connettersi ad un mail server su cui risiedono le mailbox degli utenti, autenticare l'utente, e **scaricare i messaggi di posta elettronica** giacenti sul server per l'utente

```
marco@aquilante ~$ telnet mail.aquilante.net 110
+OK <22593.1129980067@aquilante.net>
USER pippo
+OK
PASS pluto
+OK
LIST
+OK
  1 817
  2 124
  .
RETR 1
+OK
Return-Path: <pippo@aquilante.net>
Delivered-To: pippo@aquilante.net
Date: Sat, 22 Oct 2005 13:24:54 +0200
From: Maria Rossi <maria@rossi.org>
Subject: Cena in pizzeria
Content-Type: text/plain; charset=ISO-8859-1
testo messaggio
  .
DELE 1
+OK
QUIT
+OK
```

Apparati di rete

- **Hub**
 - è un nodo di concentrazione e di smistamento sulla LAN dei pacchetti di rete: riceve un pacchetto da una porta e lo inoltra a tutte le altre porte (non esegue alcuna operazione di routing o di indirizzamento selettivo del pacchetto)
 - opera a livello "Fisico": il livello 1 della pila OSI semplificata o dello stack TCP/IP
- **Bridge**
 - consente il collegamento tra due LAN che operano con mezzi fisici differenti, trasportando a livello fisico i pacchetti da un media all'altro
 - svolge il ruolo di "commutatore" nello smistamento dei pacchetti che provengono da una porta del bridge e indirizzandoli (dopo averne analizzato l'header) verso la porta su cui è presente il nodo destinatario
 - opera a livello "Data Link": il Livello 2 della pila OSI semplificata o Livello 1 dello stack TCP/IP
- **Switch**
 - è un apparato analogo al bridge, ma opera generalmente su uno stesso media (es.: rete Ethernet) e, al contrario di un bridge, è dotato di numerosissime porte (decine o centinaia)
 - collega in maniera trasparente gli host su una LAN operando come un "commutatore" e riducendo le collisioni su una rete broadcast
 - opera a livello "Data Link": il Livello 2 della pila OSI semplificata o Livello 1 dello stack TCP/IP

Domini di collisione e di broadcast

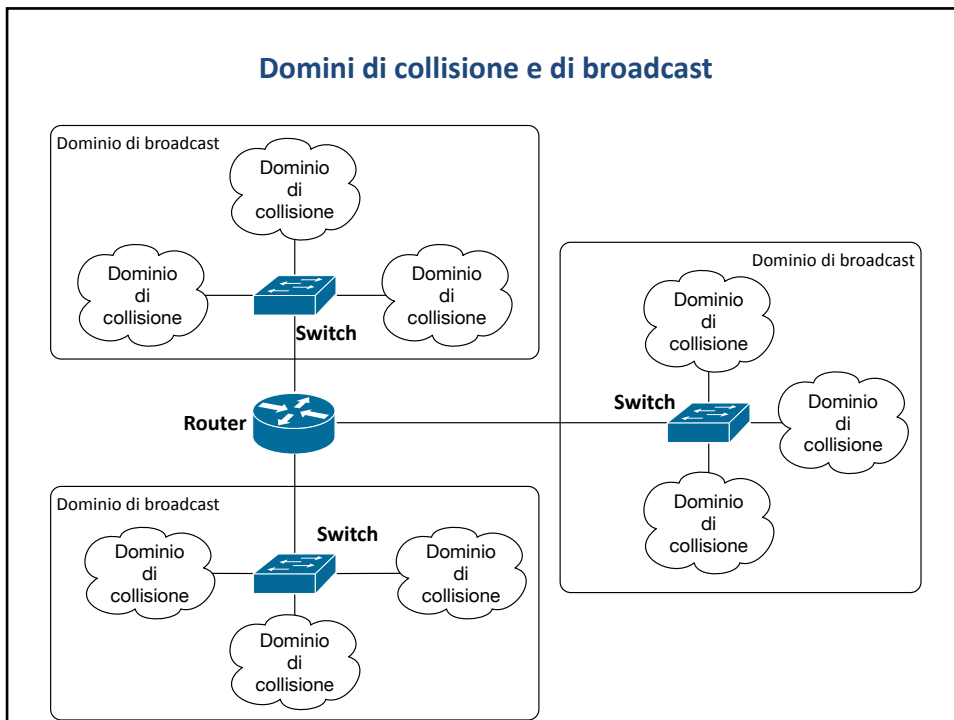
- **Dominio di collisione:** perimetro della rete entro cui i pacchetti possono andare in collisione fra di loro
- **Dominio di broadcast:** perimetro della rete entro cui i pacchetti vengono inviati in modalità broadcast a tutti i nodi del dominio
- **Hub:** consente la creazione di un dominio di collisione con più di due nodi
- **Switch:** chiude un dominio di collisione collegando fra loro domini di collisione diversi
- **Router:** chiude un dominio di broadcast collegando fra loro più domini di broadcast

The diagram illustrates the relationship between network devices and their domains. A Hub connects several nodes (laptops and desktops) within a single 'Dominio di collisione' (collision domain). A Switch connects multiple such collision domains, creating a larger 'Dominio di broadcast' (broadcast domain). A Router connects multiple broadcast domains, separating them into distinct broadcast domains. A cloud labeled 'Altri domini di broadcast' (Other broadcast domains) is connected to the Router, indicating its role in interconnecting these domains.

Apparati di rete



- **Router**
 - è un *appliance* che consente l'interconnessione tra due o più reti e si occupa di instradare i pacchetti da una rete verso la rete su cui è presente l'host di destinazione
 - opera al livello "Network": il Livello 3 della pila OSI semplificata o il Livello 2 dello *stack* TCP/IP
- **Firewall**
 - è un *server* o un *appliance* che si occupa di filtrare (interrompendo la trasmissione dei pacchetti o instradandola verso la destinazione) il traffico in entrata o in uscita rispetto ad una rete; è utilizzato per proteggere i nodi di una rete da intrusioni provenienti dall'esterno della rete stessa
 - può avere funzioni **IDS (Intrusion Detection System)** che rilevano possibili tentativi di attacco dall'esterno analizzando il traffico di rete
 - può avere funzioni **IPS (Intrusion Prevention System)** che attuano blocchi del traffico di rete da determinati host, reti esterne o protocolli, su indicazione del sottosistema IDS
 - opera ai livelli "Network" e "Transport": i livelli 3 e 4 della pila OSI semplificata o i livelli 2 e 3 dello *stack* TCP/IP

The Router image shows a rack-mountable device with multiple ports and a control panel. The Firewall image shows a similar rack-mountable device with a prominent front panel featuring a display screen and numerous ports.



Apparati di rete

- **Load Balancer**
 - è un *appliance* che consente il «bilanciamento del carico» su più apparati server per l'erogazione di un medesimo servizio; in questo modo garantisce la *scalabilità* e l'*alta affidabilità*
 - opera al livello "Application" (5) o a livello "Network" (3) della pila OSI semplificata (rispettivamente livelli 4 e 2 dello *stack* TCP/IP)
- **Proxy Server**
 - È un server, un *appliance* o un software che svolge un ruolo di intermediazione tra più client e uno o più server; in questo modo può:
 - Stabilire se i client possono eseguire o meno la richiesta verso i server;
 - Filtrare le risposte dei server prima di restituirle al client, entrando nel merito del contenuto della risposta;
 - Eseguire *caching* dei contenuti, per migliorare le performance
 - Nascondere dietro ad uno stesso indirizzo IP le richieste effettuate da più client (meccanismo utilizzato anche dai TOR server).
 - Il servizio offerto da un proxy server può essere «aperto» o «autenticato» (ossia soggetto all'autenticazione dell'utente che intende utilizzare il servizio) e, in quest'ultimo caso, può applicare regole di autorizzazione basate sul profilo dell'utente
 - Opera a livello «Application» della pila OSI semplificata (livello 4 dello *stack* TCP/IP)

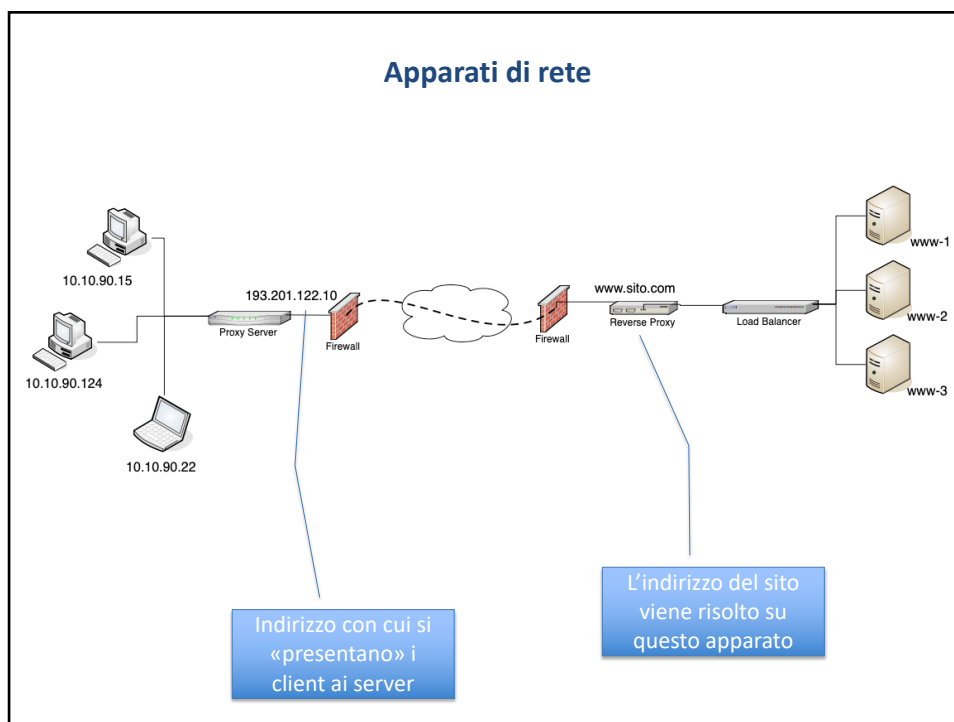
Apparati di rete

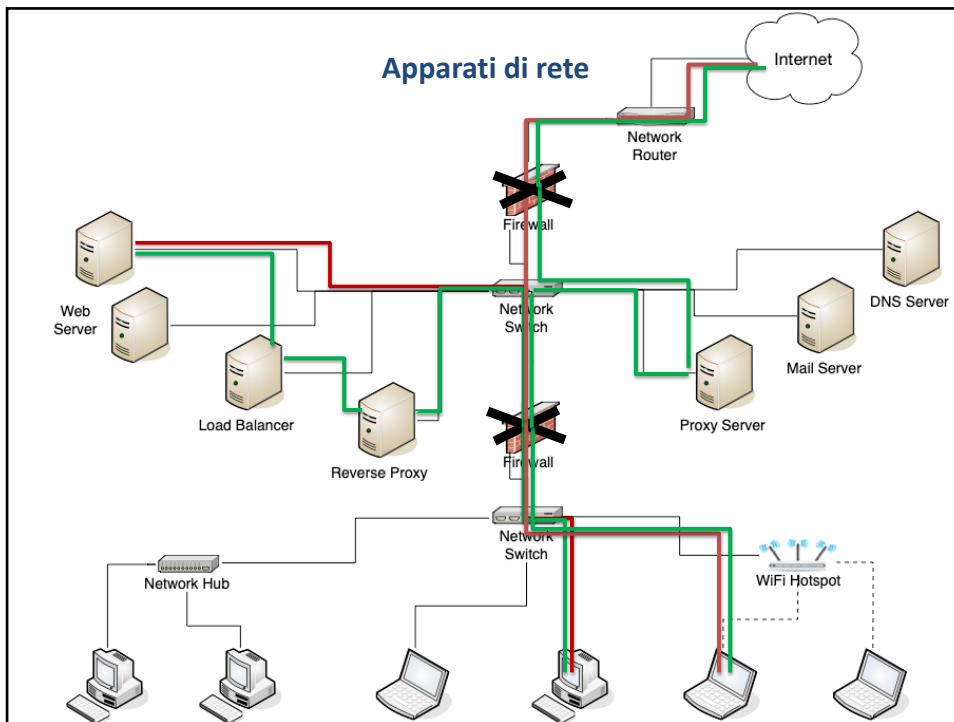
- **Reverse Proxy**

- è un server, un'applicazione, talvolta un appliance, che svolge il ruolo di proxy, ma dall'esterno della rete verso l'interno, in modo per certi versi analogo ad un load balancer, ma eseguendo controlli di autorizzazione sulle richieste indirizzate ai server e intervenendo sul contenuto delle richieste inviate dai client al server e delle risposte prodotte dal server verso i client
- nei sistemi di *access management* viene utilizzato per filtrare le richieste dei client verso i web server e attuando politiche di autenticazione e autorizzazione degli utenti
- opera al livello "Application" (5) della pila OSI semplificata (livello 4 dello *stack* TCP/IP)



Apparati di rete





Apparati di rete e Livelli dello stack

Pila OSI	Stack TCP/IP	Protocolli	Apparati di rete
Livello 1: Fisico	Livello 1: Network access	Blue Tooth, DSL, 802.11	Hub Modem
Livello 2: Data Link		Ethernet, PPP, wi-fi	Bridge Switch
Livello 3: Network	Livello 2: Internet	IP	Router Firewall Load Balancer
Livello 4: Transport	Livello 3: Transport	TCP	Firewall
Livello 5: Application	Livello 4: Application	FTP, SMTP, HTTP, POP3, SSH, Telnet, ...	Firewall Load Balancer Proxy server Reverse Proxy Applicazioni server e client

Bibliografia essenziale

- ① Craig Hunt, *TCP/IP Network Administration*, O'Reilly & Associates, 1994.
- ② Andrew S. Tanenbaum, David J. Wetherall, *Reti di calcolatori*, quinta edizione, Pearson, 2011.
- ③ Alvaro Retana, Don Slice, Russ White, *Advanced IP Network Design*, CISCO Press, 1999.
- ④ Robert Wright, *IP Routing Primer*, CISCO Press, 1998.

- ⑤ IETF – Internet Engineer Task Force, *RFC – Request For Comments*, <https://www.ietf.org/rfc.html>
- ⑥ IANA – Internet Assigned Number Authority, <http://www.iana.org>
- ⑦ NIC-IT – Registro .it, <http://www.nic.it>



Workstation grafica Sun 3/60

Sun Microsystems (oggi incorporata in Oracle) è stata un produttore di macchine UNIX leader di mercato, fra le prime aziende a credere che il futuro dell'informatica fosse strettamente legato alle reti, tanto da farne un famoso slogan aziendale: <<the network IS the computer>>