

INDICE

LE RETI.....	2
1. Un po' di storia.	2
2. Definizione e classificazione delle reti su estensione geografica.	2
3. Come funziona la rete.	4
3.1 I mezzi trasmissivi.	4
3.2 La Topologia.	6
3.3 I segnali ed i codici.	9
3.4 Codifica di canale ed errori.	12
3.5 Codici a rilevazione e correzione di errore.	13
3.6 I Protocolli: il modello ISO/OSI e TCP/IP.	17
I PROTOCOLLI DI RETE	18
4.1 ISO/OSI.....	18
4.2 TCP/IP.....	21
4.3 Protocol Data Unit (PDU).....	22
4.4 Le collisioni.....	23
4.5 Tipologie di trasmissione.	24
4.6 La tecnologia Ethernet	24
4.7 Gli indirizzi IP.....	27
4.7.1 Subnetting.	30
Laboratorio: Come modificare le impostazioni della scheda di rete.	35

LE RETI

1. Un po' di storia.

I primi computer erano *stand-alone*, questo termine indica computer non connessi con altri computer. Quando era necessario trasferire dati e/o programmi da un computer ad un altro, questo trasferimento doveva avvenire "fisicamente", caricando i dati su un supporto di memorizzazione (nastri, dischi, cartucce) da un computer, recarsi presso l'altro computer scaricare i dati su quest'ultimo.

Agli inizi degli anni '60 il ministero della difesa USA commissionò uno studio a diverse università americane, per realizzare un sistema di trasferimento di dati. Questo sistema prese il nome di **ARPANET** (**A**dvance **R**esearch **P**roject **A**gency **N**ET). Il sistema era, ovviamente, segreto e solo anni più tardi si diffuse dando il via alle attuali reti di computer.

2. Definizione e classificazione delle reti su estensione geografica.

Una rete è una connessione tra due e più dispositivi informatici per scambiare dati e risorse (hardware e software).

La prima classificazione delle reti la facciamo su base geografica ossia in base alla grandezza della rete. Le reti vengono classificate in LAN, MAN, WAN e GAN.

LAN (Local Area Network) è una rete che ha un'estensione di al più qualche chilometro, privata, non può attraversare spazi pubblici.

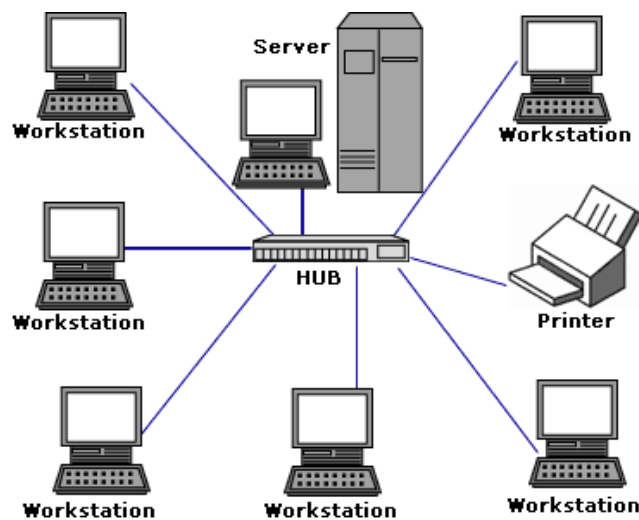


Fig. 1

MAN (Metropolitan Area Network) è una rete 'metropolitana' che si estende nell'ambito di un comprensorio (come ad esempio un comune).

WAN (Wide Area Network) è una rete molto estesa come, ad esempio un intero continente.

GAN (Global Area Network) la rete collega tutti i computer dislocati in tutto il mondo (internet).



Fig. 2

3. Come funziona la rete.

3.1 I mezzi trasmissivi.

Innanzitutto cominciamo con il dire che affinché due o più computer possano comunicare tra di loro, devono essere in qualche modo collegati. Questo collegamento avviene tramite i cosiddetti *mezzi trasmissivi*.

I mezzi trasmissivi si dividono in tre gruppi principali:

- **mezzi elettrici**, cavi che sfruttano la conducibilità elettrica dei metalli per trasmettere le informazioni sotto forma di impulso elettrico;
- **onde radio** (wireless) sfruttano le onde elettromagnetiche per trasferire le informazioni;
- **mezzi ottici**, le informazioni vengono trasmesse tramite la luce.

I principali *mezzi elettrici* sono il *cavo coassiale* ed il *doppino telefonico* (twisted pair).

Il cavo coassiale ha al suo centro un singolo conduttore di rame. Uno strato di plastica (dielettrico) garantisce l'isolamento tra il centro del conduttore ed uno schermo di metallo intrecciato (in aggiunta, possiamo trovare uno schermo laminato). Lo schermo di metallo aiuta a bloccare qualsiasi interferenza esterna.



Fig. 3

Il tutto ricoperto da una guaina esterna.

Il doppino telefonico è in cavo che troviamo nella rete telefonica.



Fig. 4

È costituito da coppie di fili in rame intrecciati. I fili vengono intrecciati per diminuire le interferenze ed in base alle schermature che possono essere presenti o meno su ogni coppia intrecciata, i doppini vengono classificati in **STP** (shielded twisted pair) in cui ogni coppia di fili è schermata (come in fig. 4), **FTP** (Foiled twisted pair) in cui c'è solo una schermatura che ricopre tutte le coppie di fili e **UTP** (Unshielded Twisted Pair) privo di schermatura. Ovviamente i cavi migliori (e più costosi) sono gli STP, quindi gli FTP in ultimo gli UTP.

Le *onde radio* fanno uso, per trasportare le informazioni, delle onde elettromagnetiche. La televisione, radio, i cellulari ad esempio utilizzano le onde elettromagnetiche per ricevere/trasmettere le informazioni. Le onde elettromagnetiche sono composte dal campo elettromagnetico generato dalle cariche elettriche e dal loro movimento (corrente elettrica). L'oscillazione delle cariche elettriche in un conduttore (un'antenna, un filo di rame) produce due campi: elettrico e magnetico che si propagano sotto forma di onda elettromagnetica nello spazio (anche nel vuoto). Le onde elettromagnetiche, generate da una sorgente, vengono intercettate da un dispositivo, ricevitore, che le codifica in segnali elettrici.



Fig. 5

I *mezzi ottici* utilizzano la luce per trasmettere le informazioni. Anche la propagazione della luce può avvenire tramite cavo o wireless. I cavi per trasmettere la luce è la fibra ottica. Il cavo in fibra ottica per trasportare i dati utilizza segnali luminosi trasmessi sopra una sottile fibra in vetro. Il cavo in fibra ottica consiste infatti di una parte centrale in vetro (core) nella quale viaggia la luce, siccome la luce può rimbalzare all'interno del core, questo è rivestito da una sorta di materiale simile ad uno specchio (cladding) che consente alla luce di essere riflessa. Il core ed il cladding sono circondati da parecchi strati di materiali protettivi. Questo cavo trasmette luce anziché segnali elettrici, eliminando così il problema dell'interferenza elettrica; questo lo rende il mezzo trasmissivo ideale in ambienti che hanno un'elevata interferenza elettrica. Il cavo in fibra ottica ha la capacità di trasmettere segnali su distanze maggiori rispetto al cavo coassiale e al twisted pair, ed inoltre consente di trasferire l'informazione a velocità più elevate.

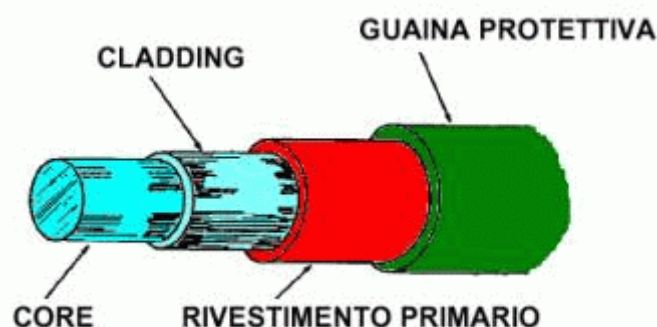


Fig. 6

I wireless ottici sono principalmente di due tipi: infrarossi, luce visibile, laser, raggi ultravioletti.

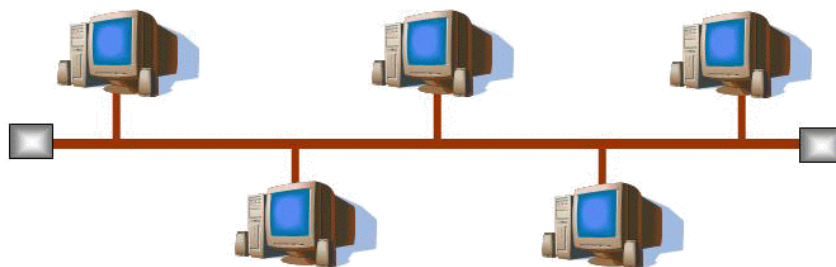
3.2 La Topologia.

Abbiamo classificato le reti in base alla loro estensione geografica. Ora le classificheremo in base alla disposizione fisica dei dispositivi del loro collegamento.

Abbiamo le seguenti *topologie*: topologia a **bus**, a **stella**, ad **anello**, ad **albero**, a **maglia**.

Topologia a bus.

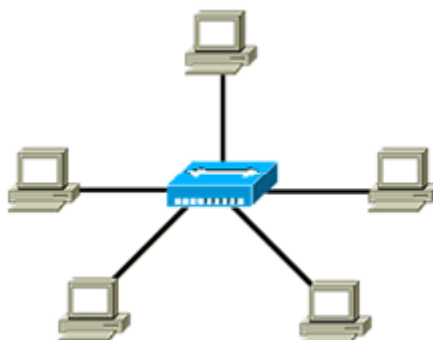
Una **topologia a bus** è l'organizzazione di rete più semplice. In effetti, in una topologia in bus tutti i computer sono collegati ad una stessa linea di trasmissione tramite un cavo, generalmente coassiale, alla fine del cavo sono inseriti dei terminatori che "assorbono" i messaggi presenti sulla linea che, altrimenti rimbalzerebbero, provocando disturbi. La parola «bus» designa la linea fisica che collega i terminali di rete:



Questa topologia ha il vantaggio della facile realizzazione e di avere un funzionamento semplice. Tuttavia, è estremamente vulnerabile dato che se una delle connessioni è difettosa, l'insieme della rete è compromesso.

Topologia ad stella.

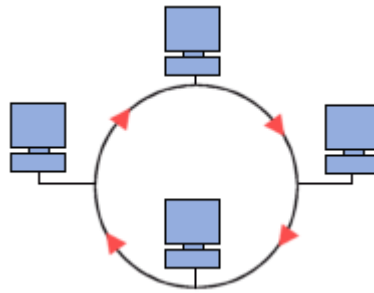
In una topologia a stella, i computer della rete sono collegati ad un dispositivo hardware centrale (es. Hub). Questi dispositivi hanno un certo numero di collegamenti nei quali è possibile inserire i cavi di rete dei computer ed hanno come ruolo quello di assicurare la comunicazione tra i diversi collegamenti:



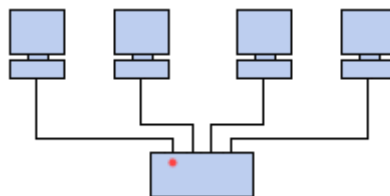
Contrariamente alle reti costruite su una topologia in bus, le reti con una topologia a stella sono molto meno vulnerabili dato che una delle connessioni può essere scollegata senza paralizzare il resto della rete.

Topologia ad anello.

In una rete con una topologia ad anello, i computer sono posti su un cerchio e comunicano ciascuno al loro turno:



In realtà, in una topologia ad anello, i computer non sono collegati in cerchio, ma ad un MAU (per Multistation Access Unit) che gestisce la comunicazione tra i computer a lui collegati impartendo ad ognuno di essi un intervallo di tempo:



Le due principali topologie logiche che usano questa topologia fisica sono Token Ring e FDDI.

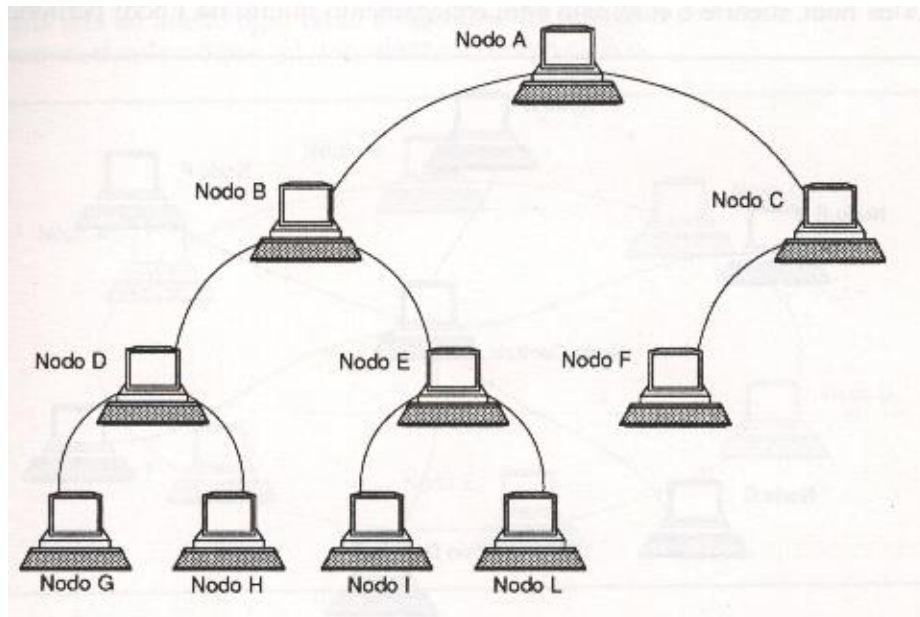
Il token (gettone) viene trasferito da un computer al successivo finché non raggiunge quello su cui sono disponibili dati da trasmettere. Il token viene modificato dal computer trasmittente che aggiunge al dato l'indirizzo del destinatario e quello del mittente e lo rinvia lungo l'anello.

I dati passano attraverso ciascun computer finché raggiungono quello il cui indirizzo corrisponde a quello indicato sui dati. Questo computer restituisce un messaggio di conferma al computer trasmittente il quale crea un nuovo token e lo immette nella rete.

FDDI (Fiber Distributed Data Interface) è una rete Token ring in cui il mezzo trasmissivo è la fibra ottica.

Topologia ad albero

Questo tipo di collegamento è di natura gerarchica. Ogni nodo (dispositivo) può rivolgersi ad un nodo superiore per connettersi agli altri. Qualora il collegamento desiderato non sia raggiungibile attraverso di esso, si ricorrerà al nodo ancora superiore e così via.



Se un nodo non è raggiungibile nemmeno attraverso la radice, si trova fuori dalla rete. E' una tecnica indicata per le reti geograficamente estese, e per collegare più reti fra loro. Non però offre cammini alternativi in caso di interruzione di una linea o di guasto di un nodo.

Topologia a maglia.

Una rete a maglia è un particolare in cui i nodi sono collegati tra loro più di una volta, così da creare canali di comunicazioni alternativi in caso che la rete sia congestionata, che la linea non funzioni bene o che uno dei nodi abbia malfunzionamenti.

Una rete a maglia può essere definita completa (fig. 7) se tutti i nodi sono connessi tramite percorsi alternativi in maniera diretta a tutti gli altri o parziale (fig. 8) se vengono implementati soltanto alcuni percorsi alternativi tra i nodi.

Ed è proprio la rete a maglia parziale ad essere particolarmente utilizzata in scenari reali, poiché il costo è relativamente contenuto rispetto a quanto impiegherebbe realizzare in toto una maglia totale – si sceglie, per la realizzazione, un buon compromesso tra i collegamenti richiesti ed il costo per implementarli.

Le reti a maglia, seppur costose, riducono sensibilmente il pericolo di malfunzionamento generale rispetto alle reti ad anello e alle reti a bus.

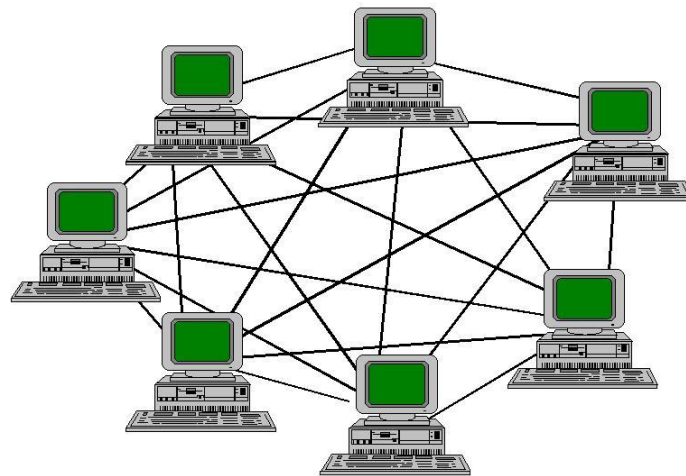


Fig. 7 - Maglia completa

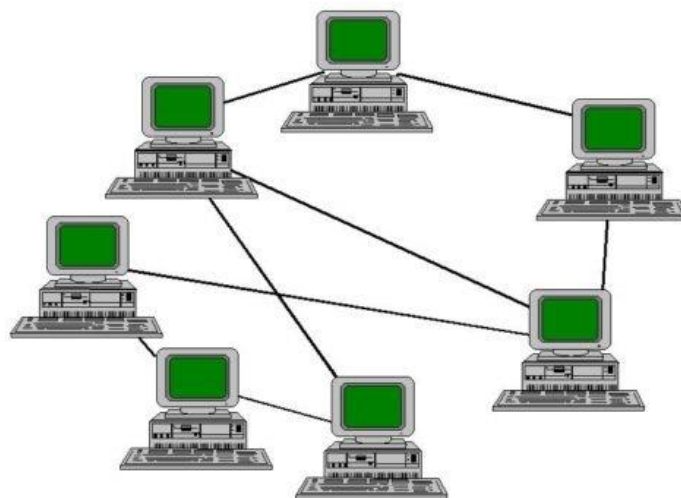


Fig. 8 - Maglia parziale.



Ci sono reti che sono un "miscuglio" delle topologie indicate. Queste reti prendono il nome di topologia mista.
Ogni dispositivo della rete viene definito ***punto di connessione***.

3.3 I segnali ed i codici.

Sul mezzo trasmissivo, per la trasmissione delle informazioni, viaggiano i *segnali*. I segnali si suddividono in segnali elettrici, ossia quelli che attraversano i mezzi trasmissivi

elettrici, i segnali ottici, quelli che attraversano mezzi ottici. I segnali si distinguono, inoltre, in segnali analogici e segnali numerici o digitali. I segnali analogici sono i segnali che assumono con continuità i valori compresi in un determinato intervallo. I segnali che assumono valori discreti (ad esempio $(0,1)$) si dicono numerici o digitali.

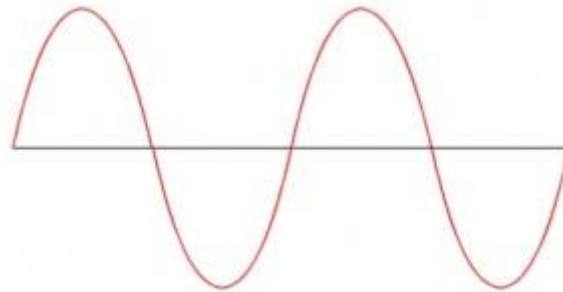


Fig. 9 - segnale analogico

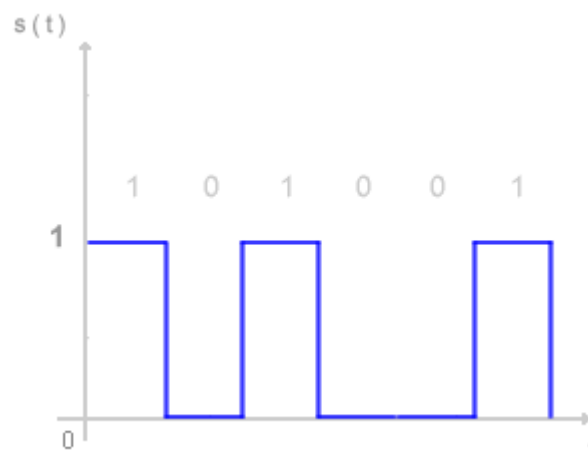


Fig. 10 - segnale numerico o digitale

I segnali durante il viaggio dal mittente al destinatario, possono essere trasformati da analogici a digitali e viceversa. Per effettuare tali conversioni si utilizzano due circuiti: l'**ADC** - converte i segnali da analogico a digitale e il **DAC** che converte i segnali da digitali ad analogici. I dispositivi che effettuano le due conversioni, si chiamano **modem** (modulatori-demodulatori). Questi dispositivi, convertono i segnali elettrici che viaggiano su rete telefonica di tipo analogico in segnali digitali (quando entrano nel computer collegato alla rete) e viceversa.

I messaggi che transitano sulla rete sono quindi formati da un insieme di segnali che viaggiano sui mezzi trasmissivi. Le reti possono essere:

- a **commutazione di pacchetto**: il messaggio viene diviso in pacchetti, ognuno contenente l'indirizzo del mittente e del destinatario ed altre informazioni per l'assemblaggio del messaggio una volta che tutti i pacchetti sono arrivati a destinazione. Ogni pacchetto, nel percorso dal mittente al destinatario, può seguire una strada, sulla rete, diversa.

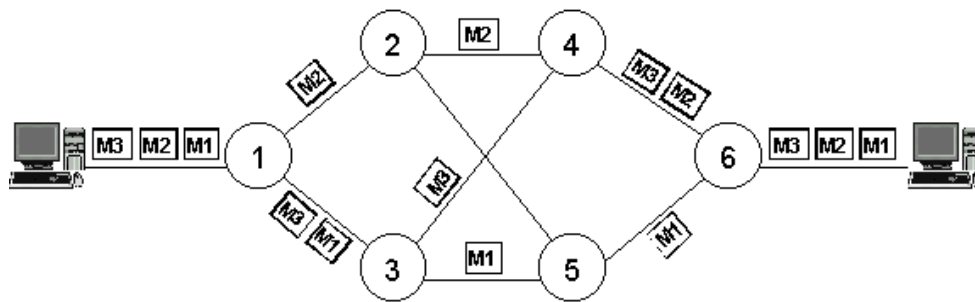


fig. 11

- a **commutazione di circuito**: prima di effettuare una trasmissione si deve stabilire a priori il percorso fisico tra sorgente e destinatario; questo tipo di trasmissione è utilizzata nella trasmissione telefonica tradizionale.

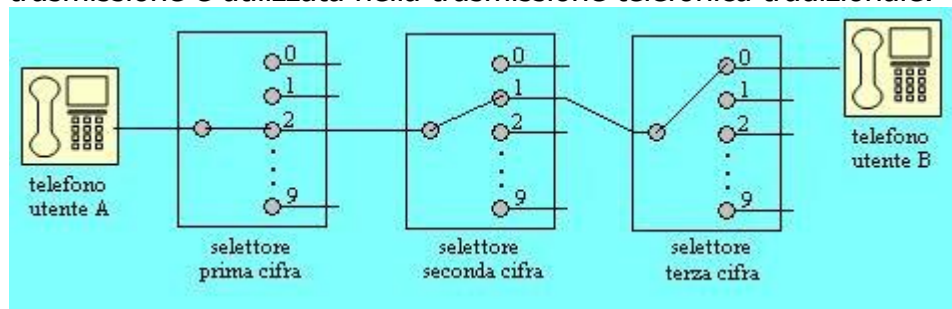


fig. 12

I mezzi trasmissivi vengono definiti anche con il termine *canali trasmissivi*; la caratteristica principale del canale è la **capacità o larghezza di banda (broadband)** definita come la *quantità di informazione che può essere trasmessa dal quel canale nell'unità di tempo*. La quantità effettiva di dati trasmessi sul canale si definisce

Throughput: possiamo definire broadband il throughput massimo. Sia il broadband che il throughput vengono misurati in bit per secondo (bps) o nei multipli kbps (kilo bit per sec.) Mbps (Mega bit per sec.), Gbps (Giga bit per sec.). Il canale può essere affetto da disturbi, chiamati rumore, che fanno sì che il ricevente riceva un segnale diverso da quello trasmesso, pertanto il mezzo trasmissivo (canale) scelto deve rendere minima l'alterazione del messaggio trasmesso, tramite alcune tecniche (tecniche di ridondanza). L'entità dei danni subiti a causa dei disturbi viene misurata tramite il rapporto segnale-rumore (S/R) che rappresenta un indice di qualità del collegamento.

In definitiva nella trasmissioni delle informazioni abbiamo la **catena di Shannon**:

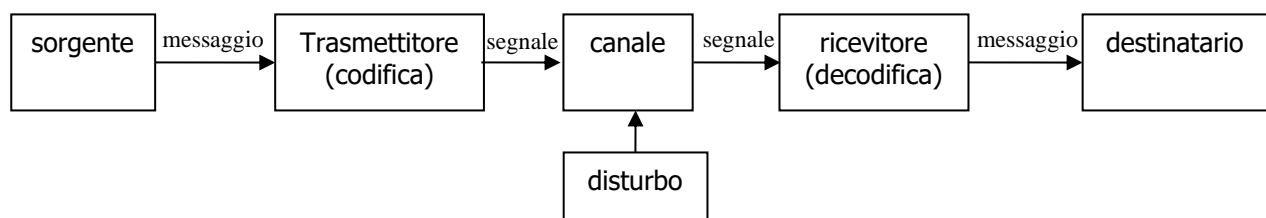


Fig.13

È ovvio che affinché la sorgente e la destinazione si "capiscano" quando si inviano messaggi, devono usare lo stesso **codice** (due dispositivi che vogliono scambiare delle informazioni devono usare gli stessi simboli la cui codifica rappresenta lo stesso carattere;

ad esempio, una sequenza di 1 e 0 disposti in un certo modo e di una certa determinata lunghezza rappresentano un carattere dell'alfabeto latino come la codifica ASCII). Durante la trasmissione di un messaggio, i vari dispositivi della rete possono utilizzare diverse codifiche:

codifica di sorgente: la codifica utilizzata per codificare il messaggio emesso dalla sorgente;

codifica di canale: sul canale, come detto, i segnali possono subire dei disturbi, per limitare i danni di questi rumori si utilizzano codici particolari, detti ridondanti, che possono:

- far capire al destinatario che il messaggio è arrivato errato,
- che consentono la correzione automatica del messaggio che ha subito errori durante la trasmissione sul canale;

codifica di linea: il processo di trasformazione del messaggio in segnali che transitano sul mezzo trasmissivo (canale).

Per tutte e tre le codifiche si dovrà prevedere un *decodificatore*.

In definitiva, per costruire un messaggio da immettere sulla rete sono necessari un **alfabeto** (esempio l'alfabeto latino o i 10 numeri interi) i cui caratteri in sequenza rappresentano una **parola**; i caratteri che compongono la parola vengono, quindi, codificati con un **codice**.

3.4 Codifica di canale ed errori.

Quando due persone parlano, le onde sonore si propagano nell'aria e se siamo in un ambiente in cui ci sono dei rumori, ad esempio un martello pneumatico a lavoro, la comunicazione tra le due persone risulta difficile ad alcune volte, in parte, incomprensibile. La stessa cosa può accadere ad un segnale che attraversa un mezzo trasmissivo, già la propagazione del segnale stesso nel mezzo può, da solo, generare un disturbo. Quindi nella trasmissione il segnale è accompagnato da un altro segnale del tutto causale che prende il nome di **rumore** o **noise**.

I disturbi si dividono in:

rumore bianco: è il rumore che si genera dal mezzo trasmissivo stesso e dai circuiti elettrici, generato dall'attraversamento, ad esempio, degli elettroni nel conduttore, difficilissimo da eliminare;

disturbo impulsivo: è il disturbo che proviene dall'esterno del mezzo trasmissivo ma che lo influenza, come ad esempio la scarica elettrica generata da un fulmine che influenza le onde radio, o un campo magnetico che può influenzare una trasmissione su un cavo non adeguatamente schermato.

Il problema principale quando si genera un disturbo in una trasmissione, è che il destinatario si accorga che il messaggio originario è arrivato errato e se possibile correggere l'errore o chiedere alla sorgente di ripetere la trasmissione. Per ottenere ciò i codici di canale sono **ridondanti**.

Vediamo di capire cosa significa ridondanza.

Sappiamo che per codificare un alfabeto di S caratteri in binario, abbiamo bisogno di n cifre binarie in modo tale che $2^n = S$.

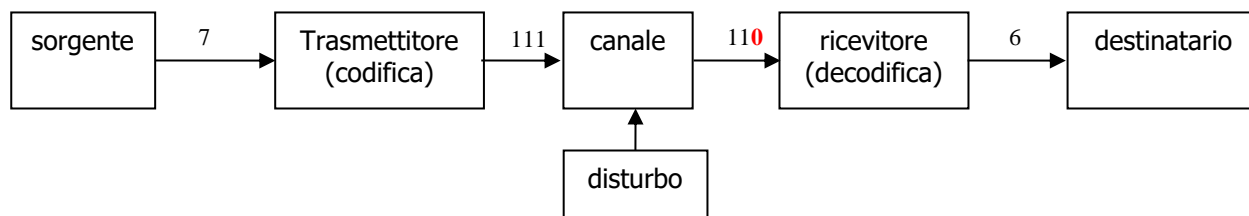
Ad esempio per codificare un carattere ottale (un numero da 0 a 7) abbiamo bisogno di 3 cifre binarie.

ottale	binario
0	000

1	001
2	010
3	011
4	100
5	101
6	110
7	111

La codifica binaria di sopra utilizza il minimo numero di cifre binarie necessarie per rappresentare le otto cifre del sistema ottale.

Supponiamo che la sorgente voglia inviare il numero 7 ad un destinatario ma sul canale si verifichi un errore dovuto ad un disturbo impulsivo che modifica uno o più bit della codifica binaria del numero:



Il destinatario riceve una cifra (errata) dell'alfabeto ottale e non può accorgersi che la trasmissione è stata inficiata da un errore.

Utilizzando quindi una codifica con il numero di bit esattamente necessari per rappresentare un alfabeto e si verifica un errore durante la trasmissione, il destinatario non riesce ad accorgersi che si è verificato un errore. Per evitare questa situazione si utilizza una codifica con un maggiore numero di cifre. Questi codici si dicono ridondanti e sono formati da un certo numero (n) di cifre per rappresentare il dato, **bit di dato**, e da un numero (k) di bit (ridondanti) per controllare se si è verificato un errore, **bit di controllo**.

3.5 Codici a rilevazione e correzione di errore.

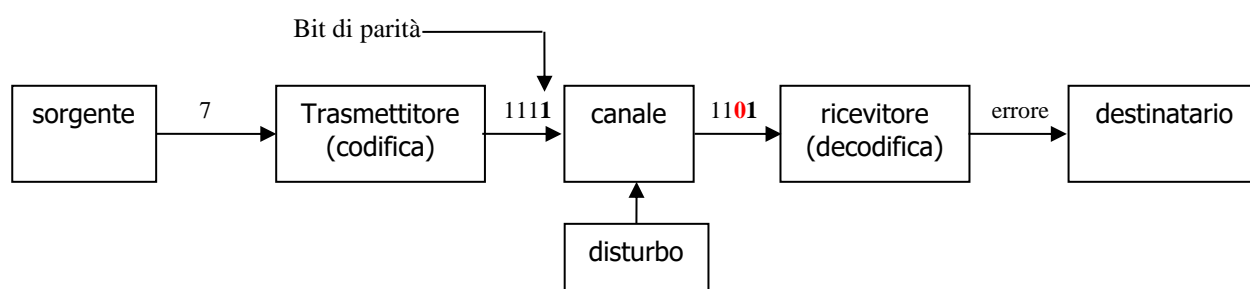
Controllo di parità.

Questo codice permette di individuare se solo su un bit del messaggio si verifica un errore. Il codice in questione ha un solo *bit di controllo* ($k=1$) chiamato **bit di parità** e vale 1 se il numero di uno dei *bit di dato* sono dispari, 0 altrimenti. Ad esempio se volessimo codificare i numeri ottali, la codifica a *controllo di parità* sarebbe formata da 4 bit: 3 bit di dato ed un bit di controllo:

	Controllo parità	
ottale	Bit dato	Bit parità
0	000	0
1	001	1

2	010	1
3	011	0
4	100	1
5	101	0
6	110	0
7	111	1

Il destinatario controllando il bit di parità può accorgersi se durante la trasmissione si è verificato un errore su un bit e, quindi, richiedere, ad esempio, la ritrasmissione del messaggio alla sorgente:



Come si può notare, la sequenza (errata) 1101 non è presente nella tabella di codifica sopra indicata, quindi la fase di decodifica si accorge dell'errore.

Controllo a ridondanza ciclica

Il **CRC** (**C**yclic **R**edundancy **C**heck, controllo a ridondanza ciclica) è un metodo a controllo polinomiale in cui i bit che formano un messaggio vengono visti come i coefficienti di un polinomio $T(x)$ di grado $n-1$ in x , dove n è il numero di bit del messaggio. Ad esempio se il messaggio da trasmettere è **10011101** il polinomio $T(x)$ associato è:

$$T(x) = 1 \cdot x^7 + 0 \cdot x^6 + 0 \cdot x^5 + 1 \cdot x^4 + 1 \cdot x^3 + 1 \cdot x^2 + 0 \cdot x + 1 = \mathbf{x^7 + x^4 + x^3 + x^2 + 1}$$

Questo polinomio viene diviso per un altro polinomio, $G(x)$, detto **polinomio generatore**, di grado minore rispetto al polinomio $T(x)$ e con il termine noto uguale ad 1.

La divisione tra $T(x)$ e $G(x)$ produce un polinomio resto $R(x)$.

Ora sia la sorgente che la destinazione posseggono il polinomio generatore $G(x)$. La sorgente trasforma il messaggio, prima d'inviarlo, in modo tale che sia divisibile per $G(x)$:

1. aggiunge al messaggio originale tanti 0 quanto è il grado di $G(x)$ chiamiamo $T'(x)$ questo messaggio
2. divide $T'(x)$ per $G(x)$ ottenendo il resto $R(x)$
3. sottrae $R(x)$ a $T'(x)$: sia $T''(x)$ il polinomio risultante dalla divisione, tale polinomio è divisibile per $G(x)$.

$T''(x)$ in realtà è uguale a $T(x)$ con l'aggiunta, alla fine, di alcuni bit detti crc che rendono il polinomio divisibile per $G(x)$.

La sorgente invia, quindi, $T''(x)$ quando il destinatario $T''(x)$, lo divide per $G(x)$. Se il resto della divisione è 0 il messaggio è arrivato in modo corretto altrimenti si è verificato un errore durante la trasmissione.

Controllo trasversale e longitudinale.

Questo codice è simile al bit di parità solo che si applica su insiemi di 8 byte (ogni byte sappiamo che è formato da 8 bit) di caratteri, che disposti l'uno sotto l'altro formano una matrice 8x8, in cui si controlla sia in orizzontale che in verticale in numero di 1 presenti.

Supponiamo ad esempio che sono stati inviati i caratteri C1, C2, C3 e C4 (per semplicità ogni carattere è di 4 bit ossia un semi byte) che formano la matrice sottostante:

	CONTROLLO LONGITUDINALE				
C1	1	0	1	0	0
C2	1	1	1	0	1
C3	1	1	0	0	0
C4	1	0	0	0	1
CONTROLLO TRASVERSALE	0	0	0	0	

Se durante la trasmissione di questi quattro caratteri si verifica *un* errore il codice ci permette anche di individuare dove si è verificato:

	CONTROLLO LONGITUDINALE				
C1	1	0	1	0	0
C2	1	1	0	0	0
C3	1	1	0	0	0
C4	1	0	0	0	1
CONTROLLO TRASVERSALE	0	0	1	0	

L'incrocio tra i bit di controllo diversi (segnati in rosso) rispetto a quelli inviati, ci dicono dove si è verificato l'errore ed effettuare l'opportuna correzione.

Distanza di Hamming e rilevazione degli errori .

Diamo alcune definizioni:

Peso: il numero di uno presenti in una parola, ove per parola indichiamo un byte es. 10011100 ha peso 4.

- ✓ **Distanza tra due parole:** è il numero di posizioni in cui due parole hanno valori diversi nei corrispondenti bit es. P1 = 1001001 e P2 = 1000101 ha distanza 2, infatti i bit indicati in rosso in P1 e P2 hanno valori differenti.
- ✓ **Molteplicità dell'errore:** è la distanza tra la parola trasmessa e quella ricevuta. Ad esempio se P1 è il messaggio inviato e P2 quello ricevuto, la molteplicità dell'errore è 2.
- ✓ **Distanza minima di Hamming (h):** se M è un messaggio con N parole, h è la distanza minima tra due qualsiasi parole del messaggio se h è uguale ad 1 il codice è efficiente se h è maggiore di 1 è ridondante.

Codici in cui $h > 1$ permettono di rilevare la presenza di errori.

Codice di Hamming.

Il codice di Hamming è un codice a correzione di errore e si ottiene inserendo dei bit di parità in posizioni specifiche all'interno del messaggio che si vuole trasmettere.

Per determinare quanti bit di controllo (R) si devono usare, deve essere verificata la seguente relazione:

$2^R \geq M + R + 1$ ove M è la lunghezza del messaggio ed R sono i bit ridondanti.

Ad esempio se abbiamo un messaggio con 4 bit da trasmettere si deve trovare il più piccolo R in modo che sia soddisfatta la disequaglianza di sopra:

$2^R \geq 4 + R + 1$ è facile verificare $R = 3$ soddisfa la disequaglianza.

Vediamo adesso dove devono essere posti questi bit di controllo all'interno del messaggio da trasmettere.

Questi bit devono essere posti nelle posizioni delle potenze di due che vanno da $2^0, 2^1, \dots, 2^{R-1}$.

Nel nostro esempio siccome $R = 3$, le posizioni ove porre i bit di controllo sono in

$2^0 = 1$ posizione, $2^1 = 2$ posizione, $2^2 = 4$ posizione.

Ad esempio se il messaggio è formato dai bit dati $D_1 D_2 D_3 D_4$ e se indichiamo con P_0, P_1, P_2 i bit di parità, il messaggio da trasmettere assumerà la forma:

messaggio	P_0	P_1	D_1	P_2	D_2	D_3	D_4
posizione	1	2	3	4	5	6	7

I bit del messaggio originale ($D_1 D_2 D_3 D_4$) vengono posti nelle posizioni "libere" 3, 5, 6, 7.

Vediamo ora come assegnare un valore 0 o 1 ai bit di parità supponiamo che il messaggio che vogliamo trasmettere sia $D_1 D_2 D_3 D_4 = 0101$

messaggio	P_0	P_1	0	P_2	1	0	1
posizione	1	2	3	4	5	6	7

Consideriamo le posizioni dove sono presenti degli 1. Le posizioni in cui sono presenti degli uno sono la posizione 7 e la posizione 5. trasformiamo in binario le posizioni coinvolte: $7_{10} = 111_2$ e $5_{10} = 101_2$

Facciamo XOR tra i due numeri (se fossero state coinvolte più posizioni si faceva XOR a due a due, prima tra le prime due posizioni il risultato con la terza e così via). Otteniamo: $111 \text{ xor } 101 = 010$. Assegniamo, dalla posizione meno significativa, i bit del risultato dell'xor ai bit di parità: $P_0 = 0, P_1 = 1, P_2 = 0$. Il messaggio diventa, in definitiva:

messaggio	0	1	0	0	1	0	1
posizione	1	2	3	4	5	6	7

Una volta trasmesso, il destinatario fa la medesima operazione. Gli 1 ora si trovano in posizione 2, 5 e 7; analogamente $2 = 010, 5 = 101$ e $7 = 111$. Facciamo gli xor:

$010 \text{ xor } 101 = 111 \Rightarrow 111 \text{ xor } 111 = 000$. Siccome il risultato è 0 significa che la trasmissione è avvenuta correttamente. Supponiamo che il destinatario riceva invece un messaggio con errori:

messaggio	0	1	0	0	0	0	1
posizione	1	2	3	4	5	6	7

(in posizione 5 si è verificato un errore).

Gli 1, ora, sono in posizione 2 ed in posizione 7: $010 \text{ xor } 111 = 101$.

Ma $101 = 5$ ossia la posizione del bit errato che viene corretto.

3.6 I Protocolli: il modello ISO/OSI e TCP/IP.

Cominciamo con il definire **sistemi aperti**, sistemi elaborativi che pur avendo caratteristiche (sistemi operativi, architettura Hardware, ...) differenti, riescono a comunicare tra di loro tramite standard predefiniti: l'adozione di questi standard da parte di un sistema elaborativo, consente a questo sistema di diventare un sistema aperto.

Per intenderci, i sistemi elaborativi (computer) ed i dispositivi informatici, collegati tramite i mezzi trasmissivi, possano scambiare delle informazioni, è necessario che questi sistemi devono avere un insieme di regole tali che il flusso di informazioni che vengono inviate da una "sorgente" (ossia il sistema elaborativo da cui partono le informazioni) siano "comprese dalla destinazione" (dal sistema elaborativo cui arrivano le informazioni).

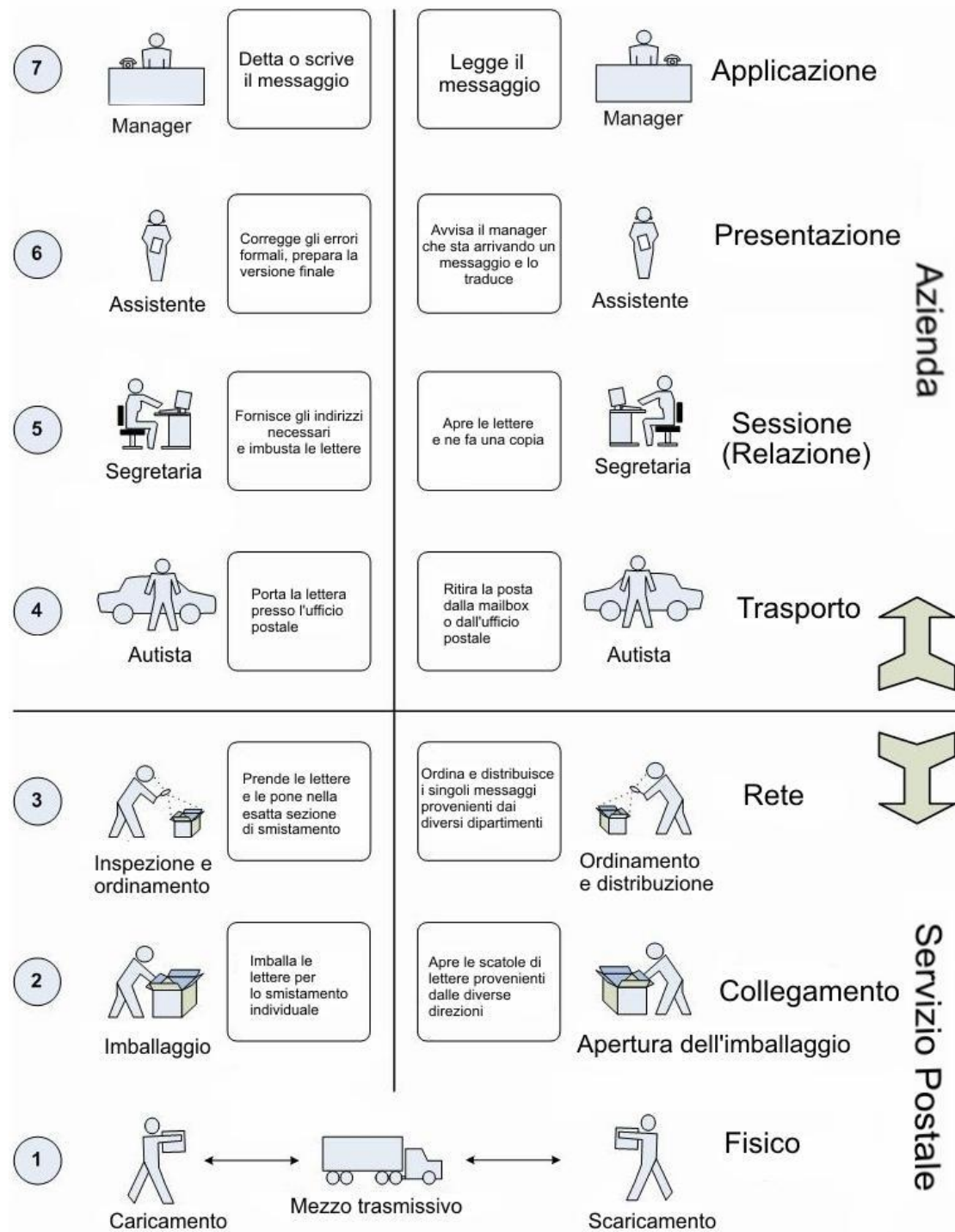
Questo insieme di regole che servono per poter espletare i *servizi di rete* (indichiamo con questo termine il tipo di trasmissione di dati - posta elettronica, flussi di dati, pagine web, ecc.) si chiamano **protocolli**.

I protocolli sono suddivisi in strati chiamati *livelli*. A sua volta ogni strato ha dei propri protocolli per gestire la comunicazione. Un messaggio (ossia l'informazione che deve essere inviata dalla sorgente e ricevuta da un destinatario) deve essere trasformato da qualcosa che è comprensibile ad un utente a qualcosa che possa viaggiare sul mezzo trasmissivo. Quando questo messaggio, così trasformato, arriva a destinazione deve subire il procedimento inverso, ossia da segnale che viaggia sul mezzo trasmissivo in qualcosa di comprensibile dall'utente destinatario. Questa trasformazione avviene facendo elaborare il messaggio dai vari livelli che compongono il protocollo.

I PROTOCOLLI DI RETE

4.1 ISO/OSI.

L'ISO/OSI (International for Standardization Organization/Open System Interconnection) è uno standard (teorico) formato da una pila di 7 livelli di protocolli di cui possiamo sintetizzare il funzionamento facendo un parallelo tra l'invio di una lettera di un manager di un'azienda ad un altro manager di un'altra azienda. Vediamo come lavorano i 7 livelli.



Parallelo tra una lettera e modello ISO - OSI

Livello 1 (Livello Fisico – Physical Layer)

Il livello fisico si occupa di *codificare* i messaggi che gli arrivano dal livello superiore in segnali compatibili ad essere inviati tramite il mezzo trasmissivo e *decodificare* i segnali che gli arrivano dal mezzo trasmissivo in messaggi per il livello superiore.

Livello 2 (Collegamento – Data Link Layer)

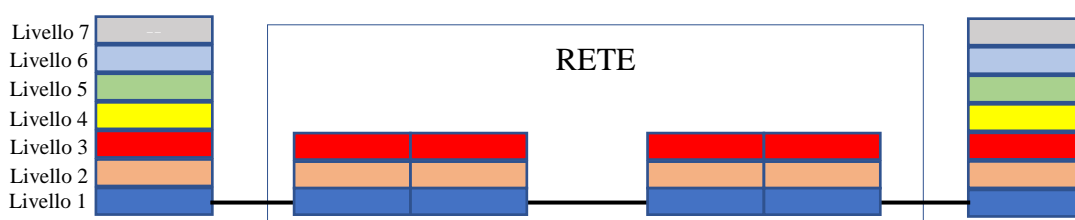
Si occupa di trasferire le informazioni sulla stessa rete (individuazione del dispositivo fisico che deve ricevere il messaggio tramite l'individuazione del nic (network interface card) ossia della sk di rete che contiene un indirizzo fisico univoco - mac address – che la identifica). Il livello inoltre esegue il controllo dell'invio/ricezione corrette del messaggio (tramite il checksum).

Livello 3 (Rete – Network Layer)

Aggiunge al messaggio proveniente dal livello superiore, gli indirizzi del destinatario e del mittente. Nella ricezione, controlla se il messaggio è destinato all'host proprietario ed in caso contrario, tramite tecniche di instradamento e commutazione invia il messaggio su un canale di transito in modo che il messaggio possa arrivare al destinatario. In definitiva il livello di rete stabilisce il percorso, tra i tanti percorsi possibili che collegano due host, che un messaggio deve compiere dall'host mittente all'host destinatario (routing), ossia tra reti diverse. Nel caso il messaggio è destinato all'host proprietario lo invia al livello superiore.



N.B. questi tre livelli appartengono, oltre agli host (sistemi elaborativi), ai dispositivi di rete che si occupano dell'istradamento o ritrasmissione dei messaggi quali, ad esempio, Hub (livello 1), Switch (livelli 1 e 2), Router (livelli 1, 2, 3).



Livello 4 (Trasporto – Transport Layer)

Il messaggio proveniente dal livello superiore viene scomposto in *segmenti* numerati ordinati per essere inoltrati sulla rete. Non è detto che questi segmenti seguano tutti lo stesso percorso per giungere all'host destinatario. Compito di questo livello in fase di ricezione è quello di riordinare questi segmenti (che possono essere giunti in modo disordinato) nella sequenza originaria ri assemblando il messaggio: nel caso questo non sia possibile per qualche malfunzionamento della rete che ha provocato la perdita di qualche

segmento, richiede la ritrasmissione del messaggio. Inoltre effettua la gestione del canale logico (uso protocolli tcp o udp).

Livello 5 (Sessione – Session Layer)

Alcune applicazioni necessitano di “regolare” la comunicazione tra diversi host sia come *sincronizzazione* sia come *intervallo di tempo* in cui la comunicazione per lo scambio di messaggi è attiva (ossia è permesso lo scambio di messaggi tra i due host). Ad esempio, un’applicazione che consente di gestire un Data Base condiviso sulla rete a più host, deve fare in modo che se un host richiede un aggiornamento di dati su questo Data Base, gli altri host non devono avere la possibilità di accedere al data base finché l’host che sta effettuando la modifica non abbia terminato; altre applicazioni che permettono ad un utente di autenticarsi effettuando il *login* (tramite l’inserimento di una user-id e password ad esempio) stabiliscono una comunicazione tra l’host usato dall’utente (client) e host ove risiede l’applicazione (server) a login effettuato. La comunicazione deve essere attiva finché l’utente non si disconnette o dopo un certo intervallo di tempo. Questi intervalli in cui è possibile effettuare la comunicazione tra i diversi host vengono chiamati “sessioni di lavoro” e gestite da questo livello. In definitiva gestisce la comunicazione tra due dispositivi effettuando l’apertura/chiusura della connessione logica tra di loro.

Livello 6 (Presentazione – Presentation Layer)

Affinché un messaggio prodotto dal livello superiore possa viaggiare sulla rete, deve rispettare delle regole sintattiche e semantiche che vengono controllate da questo livello. In passato, siccome gli host erano estremamente diversi tra di loro, questo livello si occupava principalmente di uniformare i messaggi prodotti da un host in un formato standardizzato uguale per tutti (*Abstract Syntax Notation*). Oggi i messaggi possono avere nel loro interno diversi formati (pensiamo ad un messaggio multimediale che ha nel suo interno caratteri, immagini, filmati...) compito di questo livello è “spiegare” come è formato il messaggio tramite una struttura dati che precede il messaggio. Il presentation Layer si occupa della crittografia/decrittografia di un messaggio.

Livello 7 (Applicazione – Application Layer)

Questo livello è a contatto con l’utente che manda in esecuzione un’applicazione (browser, posta elettronica...) per inviare/ricevere messaggi sulla/dalla rete.

4.2 TCP/IP.

Come accennato il protocollo ISO/OSI è un protocollo teorico nel senso che stabilisce delle regole standardizzate per la realizzazione di sistemi aperti.

Partendo dallo standard ISO/OSI è stato realizzato il protocollo TCP/IP (**T**ransmission **C**ontrol **P**rotocol/**I**nternet **P**rotocol) che ha avvicinato l'aspetto teorico al pratico.

CONFRONTO TRA I LIVELLI ISO/OSI E TCP/IP

Applicazione	Applicativo
Presentazione	
Sessione	
Trasporto	Trasporto
Rete	Internet
Collegamento	Network e fisico
Fisico	

Il protocollo TCP/IP è composto di quattro livelli: Application, Transport, Internet, Network Physical:

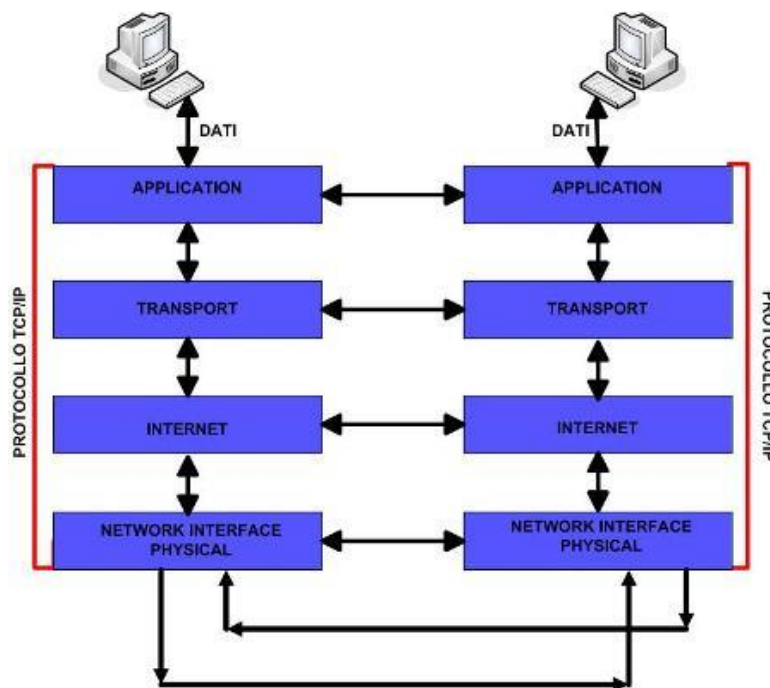


FIGURA 11

Livello Network e Physical: descrive l'interfaccia fisica fra il nodo della rete ed il mezzo trasmissivo: definisce le caratteristiche del mezzo stesso, i livelli del segnale, i tassi di trasmissione, lo schema di codifica dei dati e altri dettagli relativi alla trasmissione e (network) descrive lo scambio di dati tra un nodo e la rete e definisce le modalità di individuazione del destinatario e il tipo di servizio.

Livello internet: simile al livello di rete di ISO/OSI, descrive la trasmissione dati tra due nodi della rete (commutazione di pacchetto o trasmissione non affidabile) e definisce il

formato dei pacchetti, il sistema di indirizzamento globale, il meccanismo di instradamento dei pacchetti. In questo livello viene utilizzato il protocollo IP, che fornisce un servizio di comunicazione non affidabile, cioè non garantisce la consegna dei pacchetti.

Livello di trasporto: crea una connessione logica tra sorgente e destinazione indipendentemente dalla rete utilizzata, segmenta (nella fase di trasmissione) e assembla (nella fase di ricezione) i dati che provengono dai livelli ad esso adiacenti (applicazione/internet). Ad ogni segmento viene assegnato un numero in modo da garantire l'affidabilità della ricezione corretta dei segmenti da parte del corrispondente livello del destinatario. In questo livello vengono utilizzati principalmente due protocolli: il protocollo **TCP** e **l'UDP**.


-**TCP**: Tcp fornisce un livello di trasporto affidabile e orientato alla connessione. Per **affidabile** s'intende che prima di inviare i dati il server esegue l'handshaking, cioè chiede al client se è pronto a riceverli. Per **orientato alla connessione** s'intende che una volta stabilita la connessione i pacchetti vengano spediti in modo ordinato e che il client li riceva TUTTI in modo ordinato. Viene utilizzato per la sua sicurezza.

-**UDP**: Udp fornisce un servizio di trasporto datagram-oriented (non affidabile) e non orientato alla connessione. I pacchetti quindi vengono inviati senza chiedere al client se è pronto e non viene controllato se il client ha ricevuto tutti i pacchetti. Viene utilizzato per la sua velocità.

Livello delle applicazioni: gestisce le applicazioni che usiamo per comunicare sulla rete (Browser: per il web, posta elettronica: Outlook...), ognuno di queste applicazioni utilizza uno specifico protocollo (**http** per il web, **Ftp** per trasferire file, **smtp** e **pop3** per la posta elettronica...).

4.3 Protocol Data Unit (PDU)

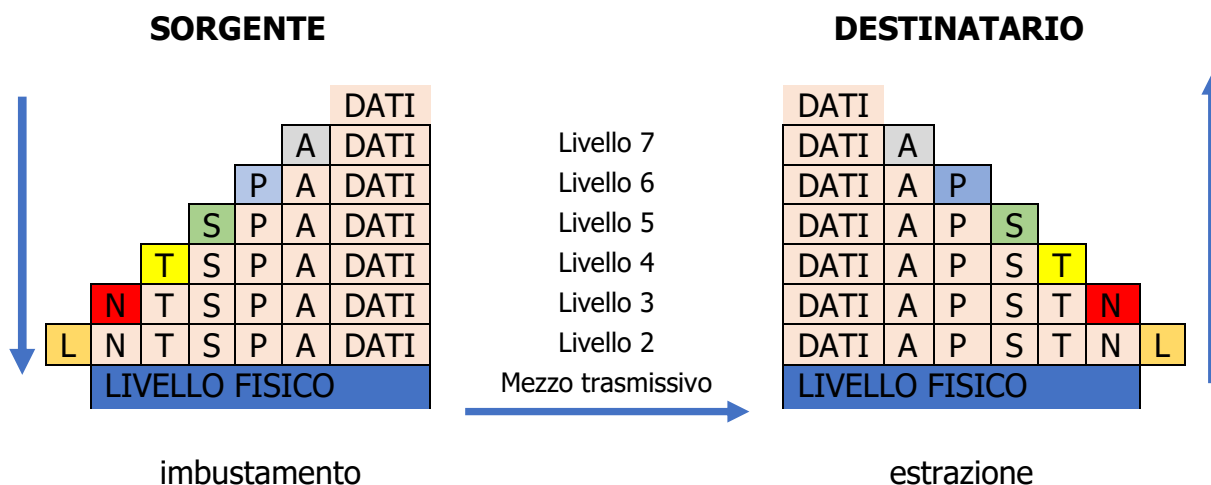
Un messaggio quando passa da un livello ad un altro per, alla fine, essere immesso sulla rete viene "arricchito", dai livelli che attraversa, da ulteriori informazioni che servono ai corrispondenti livelli dell'host destinatario per "decifrare e controllare" la parte di messaggio relativa ad ogni livello.

 *Per fare un parallelo "figurato", possiamo pensare all'invio di oggetto. Supponiamo che questo oggetto sia creato e imballato dal mittente e disimballato ed usato dal destinatario tramite un insieme di robot (ogni robot è un livello della pila ISO/OSI). Il robot a livello 7 produce l'oggetto e gli attacca un cartellino con le istruzioni per il suo uso, una volta che il robot a livello 6 ha controllato l'oggetto, lo avvolge in un involucro sul quale stampa delle istruzioni per il disimballaggio dall'involucro e su come va trattato l'oggetto; quindi l'oggetto passa ad un terzo robot (di livello 5) che lo inserisce in una scatola sulla quale appone altre istruzioni per il disimballaggio dalla scatola, e così via... Una volta arrivato a destinazione i robot di pari livello, seguendo le istruzioni stampate su ogni imballaggio, disimballano correttamente l'oggetto sino ad arrivare all'oggetto senza alcun imballaggio e usato correttamente.*

Il messaggio così trattato ad ogni livello prende il nome di PDU e le operazioni che compiono i vari livelli che arricchiscono con ulteriori informazioni (di livello) il messaggio, vengono dette di "imbustamento".

In definitiva l'imbustamento di un messaggio in un livello serve affinché il corrispondente livello del dispositivo che riceve il messaggio lo tratti allo stesso modo.

Per "imbustare" un messaggio, ogni livello aggiunge al messaggio un'intestazione (header):



N.B. anche il protocollo TCP/IP prevede, ovviamente, nei suoi livelli, l'imbustamento dei dati.

4.4 Le collisioni

Quando due o più dispositivi trasmettono informazioni sullo stesso mezzo trasmissivo, può accadere che le informazioni si "scontrino" ed i messaggi originali inviati dalle sorgenti diventano irriconoscibili. A questo punto al destinatario non rimane altro che richiedere il reinvio del messaggio alla sorgente con il conseguente aumento del tempo di trasmissione. Diverse sono le tecniche per far diminuire o eliminare le collisioni sulla rete. Vediamo quelle che più interessano al nostro studio.

L'utilizzo del token.

Questa tecnologia viene usata principalmente su reti a bus e ad anello e consiste nel far circolare sulla rete un insieme di bit chiamati token (gettone). Un host che vuole inviare un messaggio deve "impadronirsi" del token e, quando ci riesce, può procedere all'invio del messaggio verso un host destinatario. Una volta che la trasmissione è terminata l'host mittente reimmette il token sulla rete.

Un'altra tecnica consiste nel far "ascoltare" al dispositivo che vuole trasmettere un messaggio, se il mezzo trasmissivo sul quale vuole effettuare la trasmissione è "libero" o "occupato": se il mezzo trasmissivo (canale) è libero procede all'invio del messaggio, altrimenti aspetta un intervallo di tempo riascolta il canale e ripete l'operazione. Questa tecnica, però, non assicura che una collisione non possa ugualmente avvenire: se due dispositivi ascoltano il canale nello stesso istante entrambi lo possono trovare libero ed entrambi possono, di conseguenza, procedere all'invio dei messaggi che collideranno.

CSMA/CD

Una tecnica più sofisticata è **Carrier Sense Multiple Access with Collision Detection (CSMA/CD)**, possiamo sintetizzare questa tecnica con: "Ascolta prima di trasmettere e

mentre trasmetti. Se mentre trasmetti rilevi collisioni, fermati, segnala a tutte le altre stazioni la collisione e riprova più tardi secondo modalità di ritrasmissione stabilite". Il dispositivo che deve effettuare una trasmissione ascolta il canale sia prima di trasmettere sia durante la trasmissione e se, durante la trasmissione, avviene una collisione, ferma la trasmissione, aspetta un intervallo di tempo e ritenta la trasmissione dopo aver ascoltato il canale.

Definiamo **dominio di collisione** un insieme di nodi di una rete (host, dispositivi,...) che trasmettono sullo stesso mezzo trasmissivo e che concorrono per accedervi senza che l'accesso sia regolato in modo determinato.

4.5 Tipologie di trasmissione.

Le trasmissioni di un dispositivo sulla rete può essere di tipo:

broadcast – i messaggi vengono spediti verso tutti i dispositivi della rete;

unicast – i messaggi vengono spediti ad un unico dispositivo sulla rete;

multicast -i messaggi vengono spediti ad un certo numero di dispositivi sulla rete.

Inoltre la comunicazione può essere:

- **orientata alla connessione** (connection oriented) è una comunicazione in cui chi trasmette si assicura che il destinatario possa ricevere i messaggi (es. una telefonata) viene definita anche trasmissione sicura;
- **non orientata alla connessione** (connectionless) è una trasmissione in cui chi trasmette non si assicura se il messaggio è stato ricevuto dal destinatario (es. l'invio di una lettera). I frame trasmessi in modo connectionless prendono anche il nome di datagramma.

4.6 La tecnologia Ethernet

Intorno gli anni 70 si cominciò a sviluppare un progetto denominato 802 che suddivide il livello 2 (comunicazione – data link) ISO/OSI in due sottolivelli: **LLC** (Logical Link Control) e **MAC** (Media Access Control). Questa suddivisione del livello 2 permette un collegamento più semplice tra i diversi dispositivi che si connettono fisicamente al mezzo trasmissivo con il livello di rete. Precisamente il sottolivello LLC è un'interfaccia tra il livello di rete e il sottolivello MAC, che ospita il *Mac Address*, e si interfaccia a sua volta con il livello fisico che gestisce i diversi mezzi trasmissivi su cui inviare i segnali.

Livello di rete				
LLC 802.2				
Mac 802.3	Mac 802.4	Mac 802.5	Mac 802.6	FDDI
Livello fisico				

802.X sono gli standard utilizzati. Quindi a prescindere il mezzo trasmissivo e la modalità in cui vengono inviate/ricevute le informazioni (standard 802.3, 802.4, 802.5, 802.6, 802.8, 802.11), LLC (802.2) maschera al livello di rete la comunicazione del livello fisico.

Oltre l'802.2 (LLC) abbiamo:

802.3 CSMA/CD

802.4 Token bus

802.5 token ring

802.6 Metropolitan Area Network -DQDB (Distributed Queue, Dual Bus)

802.8 Fiber-optic technical advisory group

802.11 Wireless network

La tecnologia ethernet identifica tutti i dispositivi conformi alle specifiche 802.3.

Il successo di ethernet si ebbe però grazie alla realizzazione di una scheda elettronica che svolge tutte le funzionalità logiche di elaborazione necessarie a consentire la connessione di un dispositivo ad una rete e che implementa le specifiche 802.3. la scheda di rete prende anche il nome di NIC (Network Interface Controller). Ogni scheda di rete è individuata in modo univoco da un indirizzo detto indirizzo MAC (o fisico – MAC address) formato da 6 bytes: i primi tre bytes individuano il costruttore, i secondi tre un codice univoco all'interno dell'azienda produttrice.

Ethernet è, a livello di rete, un servizio connectionless, di tipo broadcast e la gestione delle collisioni è gestita da CSMA/CD.

La velocità di trasmissione era inizialmente di 10 Mbps oggi di 100 Mbps (sino ad arrivare alle 1 - 10 Gbps: le Gigabit Ethernet)

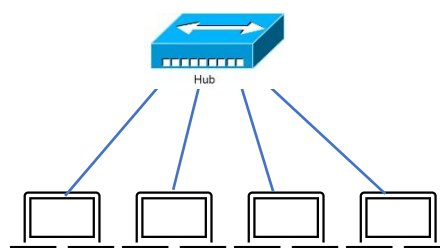
Le notazioni 10 base 2, 10 base 5, ... , 100 base T... indicano che la velocità della rete è a 10 o 100 Mbps come banda base (larghezza di banda) su un segmento (2, 5) di un mezzo trasmissivo di 200 metri, 500 metri, o da un tipo di conduttore di tipo T (cavo UTP), ecc.

REPEATER.

Siccome i mezzi trasmissivi hanno una lunghezza massima su cui il segnale può transitare (segmento), per estendere una rete si usano dispositivi che ripetono il segnale (**ripetitori** o **repeater**). Questi dispositivi lavorano a livello 1 della pila ISO/OSI, prendono il segnale che gli arriva da una porta d'ingresso ove è collegato un segmento e lo ritrasmettono da una porta di uscita su un altro segmento.

HUB.

Gli **HUB** altro non sono che ripetitori che ritrasmettono i bit in ingresso su una porta, su tutte le altre porte. Tutti i dispositivi collegati tramite Hub condividono lo stesso dominio di collisione.



BRIDGE.

Questi dispositivi consentono di 'spezzare' i domini di collisione. Quando un bridge riceve un frame, memorizza l'host di destinazione e la porta su cui spedisce il frame, su una tabella detta filtering table:

MAC address	Porta	Timestamp

Filtering table

Quindi se un messaggio proviene dalla stessa porta dell'indirizzo destinatario (vedi figura A1) non lo inoltra (l'hub che gestisce la rete collegata alla porta del bridge ha già inoltrato il messaggio a tutti i dispositivi ad esso collegati, quindi anche al dispositivo destinatario). Ogni riga nella filtering table viene cancellata allo scadere del timestamp. Siccome i bridge memorizzano il MAC address 'lavorano' a livello 2 della pila ISO/OSI. Operando in questo modo, i bridge suddividono i domini di collisione poiché i messaggi che vengono prodotti nei dispositivi collegati ad una porta del bridge non vengono inoltrati ai dispositivi collegati alle altre porte.

SWITCH.

Di solito i bridge hanno da due ad un massimo di 4 porte, gli switch hanno un funzionamento simile ai bridge ma hanno un numero molto maggiore di porte. Gli switch, negli ultimi anni hanno sostituito l'uso sia degli hub che dei bridge.

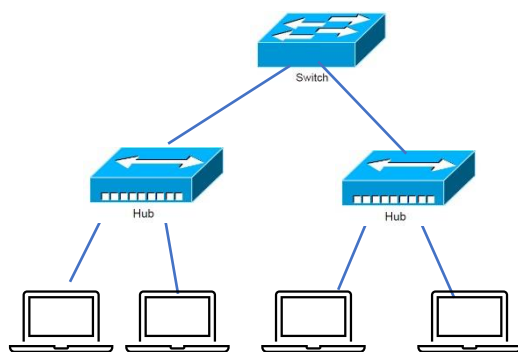


Fig. A1

ROUTER

I router sono dei veri e propri PC progettati per assolvere a dei compiti specifici (internetworking – instradamento dei dati) con un loro Sistema Operativo: **IOS** (Internetwork Operating System). Usati come interfacce per il collegamento di reti diverse (di tipo eterogeneo) o meno (leggi sotto), operano a livello 3 della pila ISO/OSI (utilizzano quindi gli indirizzi IP, invece degli indirizzi MAC, per individuare i dispositivi sulla rete) suddividendo il dominio di broadcast. Per **dominio di broadcast** si intende l'insieme di tutti i dispositivi di una rete che scambiano informazioni a livello 2 (collegamento - data link). In poche parole, tutti i dispositivi collegati ad uno stesso bridge o switch. Come per il

dominio di collisione, il dominio di broadcast provoca un aumento del traffico su tutti i dispositivi collegati a quella rete (ricordiamoci che *broadcast* significa che un messaggio viene inviato a tutti i dispositivi della rete) per far diminuire questo traffico, una rete con lo stesso dominio di broadcast, può essere suddivisa, come vedremo successivamente, in una rete con diversi domini di broadcast, utilizzando, appunto, i router. Tale suddivisione effettuata dai router, provoca anche la suddivisione della rete in altrettanti domini di collisione.

I router 'instradano' i pacchetti di reti diverse che vogliono comunicare tra loro. Spesso i router, infatti, vengono definiti **Gateway** poiché, in un router, spesso è presente anche questo dispositivo (Hardware e/o Software). I Gateway, in realtà, servono per l'interconnessione tra reti che usano protocolli diversi permettendo la comunicazione tra loro.

Un gateway, in parole povere, funge da passaggio tra due reti che operano su diversi protocolli di rete. Agendo come una sorta di "zona di passaggio" tra esse, garantisce che i dati inviati da una rete possano essere compresi ed elaborati dall'altra. Il gateway di rete traduce essenzialmente il linguaggio di una rete in quello dell'altra, facilitando lo scambio di informazioni.

Il funzionamento di un gateway di rete è intricato ma affascinante. La sua funzione principale è quella di convertire i protocolli, garantendo la compatibilità tra le reti. Quando i pacchetti di dati arrivano al gateway, vengono ispezionati alla ricerca dell'indirizzo IP di destinazione. Se la destinazione si trova all'interno della stessa rete, i dati vengono inviati direttamente. Se invece la destinazione si trova su una rete diversa, il gateway interviene convertendo i dati in un formato che può essere decifrato dalla rete ricevente.

4.7 Gli indirizzi IP.

L'indirizzo IP è un indirizzo 'logico' e serve per individuare un nodo di una rete sul quale è presente un dispositivo. Mentre l'indirizzo MAC individua 'fisicamente' ed univocamente la scheda di rete (e quindi il dispositivo ove è installata); se, ad esempio, si sostituisce il dispositivo (PC, stampante, ...) di un nodo di una rete con un altro dispositivo, a questo nuovo dispositivo può essere assegnato lo stesso indirizzo IP che era stato assegnato al vecchio dispositivo (così come se spostato il dispositivo su un'altra rete cambierà l'indirizzo IP che lo individua); al contrario, gli indirizzi MAC delle schede di rete dei due dispositivi (quello che è stato sostituito e il sostituto) sono diversi.

Gli indirizzi IP sono formati da 4 gruppi di 8 bit (ottetto) suddivisi da punti (32 bit in totale): siccome una sequenza di 8 bit viene definita byte, definiamo un indirizzo IP come 4 bytes suddivisi da punti. Ora siccome un byte può rappresentare 256 disposizioni di 1 e 0 (00000000 – 11111111) la conseguente codifica degli 'ottetti' binari in numeri decimali va da 000 a 255. In definitiva, utilizzando la codifica decimale (dotted decimal notation), gli indirizzi IP sono compresi quindi tra:

000.000.000.000 e 255.255.255.255

In binario:

00000000.00000000.00000000.00000000 e 11111111.11111111.11111111.11111111

Ogni sistema, che usa il protocollo TCP/IP per essere collegato ad una rete, deve avere assegnato un indirizzo IP: in questo modo il sistema viene individuato sulla rete.

Con 32 bit il numero massimo di indirizzi IP è:

$$2^{32} = 4.294.967.294$$

Per quanto possa sembrar grande questo numero, pensare che nel mondo siano connessi 'solo' 4 miliardi di dispositivi contemporaneamente sta diventando... riduttivo (tant'è che l'attuale protocollo IP (Ipv4) a 32 bit sarà sostituito dal protocollo Ipv6 a 128 bit...).

Per 'aumentare' il numero di dispositivi sulla rete, l'indirizzo IP è stato suddiviso in due parti 'variabili', una parte individua la rete (Net-ID) una parte il nodo (Host-ID):

Net-ID	Host-ID
--------	---------

Questa 'variabilità' tra Net-ID e Host-ID ha portato ad una classificazione degli indirizzi IP.

La seguente tabella definisce le 5 classi in cui sono suddivisi gli indirizzi IP, individuabili anche dai primi bit del Net-ID:

	Net-ID	Host-ID
<i>classe A</i>	0 7 bit	
<i>ottetto</i>	1	2 3 4

	Net-ID	Host-ID
<i>classe B</i>	1 0 14 bit	
<i>ottetto</i>	1	2 3 4

	Net-ID	Host-ID
<i>classe C</i>	1 1 0 21 bit	
<i>ottetto</i>	1	2 3 4

	Host-ID (dedicata al multicasting)
<i>classe D</i>	1 1 1 0 28 bit
<i>ottetto</i>	1 2 3 4

	Classe per usi futuri
<i>classe E</i>	1 1 1 1 1
<i>ottetto</i>	1 2 3 4

N.B. i bit più significativi (in rosso) del IP, per ogni classe, assumono i valori indicati in tabella: a questi vanno aggiunti i restanti bit per completare l'ottetto (gli ottetti).

In definitiva i range di valori che possono assumere le classi sono:

	da:	a:
Classe A	0.0.0.0	127.255.255.255
Classe B	128.0.0.0	191.255.255.255
Classe C	192.0.0.0	223.255.255.255
Classe D	224.0.0.0	239.255.255.255
Classe E	240.0.0.0	255.255.255.255

E' semplice intuire che quando aumentano i bit a disposizione di un ID diminuiscono quelli a disposizione dell'altro. Così la rete internet che collega i (tantissimi) dispositivi Host del pianeta sarà di classe A, mentre le classi B e C sono utilizzate dalle LAN.

Gli indirizzi IP si suddividono in indirizzi pubblici e privati. Gli indirizzi pubblici sono gli indirizzi che servono ad individuare gli Host su internet a questi indirizzi sono, inoltre, associati nomi mnemonici, detti domini (DNS – Domain Name System). Gli indirizzi pubblici vengono rilasciati da un ente internazionale **ICANN** (Internet Corporation for Assigned Names and Numbers).

Per definizione Host-ID formato da tutti 0 non viene assegnato a nessun dispositivo perché individua la rete (come vedremo nel paragrafo seguente), mentre se un host vuole mandare messaggi in broadcast (a tutti gli host collegati alla rete) l'Host-ID (detto in questo caso di broadcast) assume tutti 1.

4.7.1 Subnetting.

Per subnetting si intende una suddivisione di una rete LAN in più sottoreti per economizzare ed ottimizzare l'uso degli indirizzi IP all'interno della rete. Per suddividere la rete in sottoreti, vengono presi in prestito alcuni bit del Host-ID: l'operazione è quindi di fatto invisibile all'esterno della LAN (il Net-ID rimane invariato); questa suddivisione avviene all'interno della LAN tramite l'utilizzo della Subnet-Mask.

Questa maschera (subnet-mask) formata anch'essa da 4 ottetti binari (4 Bytes), ha tutti i bit relativi al Net-ID, della classe cui appartiene l'IP, ad 1 e i bit relativi all'Host-ID... che ci dicono se questo contiene una sottorete o meno. Per determinare se una rete ha sottoreti o meno, basta fare l'AND tra i bit della subnet-mask e i corrispondenti bit dell'indirizzo IP



*L'operatore logico AND tra due bit restituisce **1** se i due bit sono entrambi **1**; **0** altrimenti.*

Facciamo un esempio: supponiamo di avere il seguente indirizzo IP di classe C e la sua codifica binaria:

192.168.2.130 -> 11000000.10101000.00000010.10000010

11000000	10101000	00000010	10000010
192	168	2	130

Devo capire se 130 è l'effettivo Host-ID di un dispositivo del nodo della rete di indirizzo 192.168.2 o contiene ulteriori informazioni, quali appunto, l'indirizzo/i di sottorete della rete e l'indirizzo di un host di tale sottorete. Per accorgermene prendo la subnet -mask associata alla rete e faccio l'AND con l'indirizzo IP.

Supponiamo che la subnet-mask della rete sia 255.255.255.0.

	192	168	2	130
IP	11000000	10101000	00000010	10000010
Subnet.mask	11111111	11111111	11111111	00000000
AND (tra IP e Subnet-Mask)	11000000	10101000	00000010	00000000
	192	168	2	0

Il risultato dell'AND tra l'IP e il subnet-mask restituisce 192.168.2.0 ossia il Net-ID dell'IP: questo significa che non ci sono sottoreti e che 130 è l'indirizzo del nodo all'interno della rete. Se il risultato aveva l'ottetto (ottetti) relativi all'Host-ID diverso da zero, eravamo in presenza di una sottorete. (più semplicemente se gli ottetti relativi all'host-id della subnet-mask sono 0, la rete non è suddivisa in sottoreti, altrimenti otteniamo il Net-ID dell'IP della sottorete di appartenenza).

Vediamo ora come si fa a suddividere una rete in sottoreti. Questa operazione si compie agendo sia sulla subnet-mask, o meglio, sui bit più significativi dell'ottetto/degli ottetti relativo/i al corrispondente Host-ID dell'IP sia sugli ottetti corrispondenti all'Host-ID dell'IP. Innanzitutto bisogna decidere in quante sottoreti vogliamo suddividere la nostra rete: questo ci serve per determinare quanti bit dobbiamo usare degli ottetti dell'Host-ID dell'IP e settare ad 1 nella subnet-Mask.

Supponiamo, ad esempio, di voler suddividere la rete 192.168.2.0 in due sottoreti. Ora per dividere in due una rete abbiamo bisogno di un solo bit (assumiamo i **due** valori 0 e 1), quindi il bit più significativo dell'ottetto relativo all'Host-ID assumerà il valore 0 per individuare la prima sottorete, 1 per individuare la seconda sottorete:

prima sottorete:

11000000	10101000	00000010	00000000
192	168	2	0

Seconda sottorete:

11000000	10101000	00000010	10000000
192	168	2	128

La subnet-mask deve ampliare in numero di uno che indicano l'indirizzo di rete con, nel nostro caso, un 1 in più, in quanto sono state aggiunte 2 sottoreti:

subnet-mask (uguale per tutte le sottoreti):

11111111.11111111.11111111.10000000

11111111	11111111	11111111	10000000
255	255	255	128

Ora tutti gli indirizzi host che possiamo assegnare alle due sottoreti possono utilizzare solo 7 degli otto bit (come abbiamo visto il primo bit è stato utilizzato per dividere in due sottoreti la rete).

Collegiamo, quindi, a Hub o Switch i dispositivi che devono far parte di una sottorete e dell'altra. Le due sottoreti però, siccome comunicano a livello 2, non comunicano tra loro a meno che non vengano collegate con un router. Se le vogliamo far comunicare, colleghiamo, quindi, gli Hub (Switch) al router, invece che ad un altro Hub (Switch). Il router necessita che sia assegnato, ad ogni sua porta, un indirizzo IP: dedichiamo un indirizzo IP di ogni sottorete da assegnare alle porte del router (gateway): per convenzione l'indirizzo del gateway è il numero precedente all'indirizzo di broadcast della rete/sottorete.



Ad esempio, nella rete 200.10.10.0 l'indirizzo host di broadcast è 11111111 ossia 255 (200.10.10.255) l'indirizzo del gateway è 200.10.10.254. Gli indirizzi così formati per il gateway vanno dichiarati alle sottoreti.

Per indicare la subnet-mask viene utilizzata la notazione **XXX.XXX.XXX.XXX/24** ove **/24** indica i numeri di 1 che formano la subnet-mask in questo caso è la subnet-mask 255.255.255.0 -> 11111111.11111111.11111111.00000000 (ventiquattro 1). Nell'esempio di sopra delle 2 sottoreti, avremmo indicato 192.168.2.0/25.

I numeri di dispositivi collegabili in una sottorete sono determinati dal numero di bit dell'Host-ID che non vengono usati per la sottorete; quindi se N è il numero di bit che formano l'Host-ID (nel nostro esempio 8) e K è in numero di bit usati per la sottorete, il numero di bit per l'Host-ID è **H = N - K** da cui consegue che il numero di dispositivi

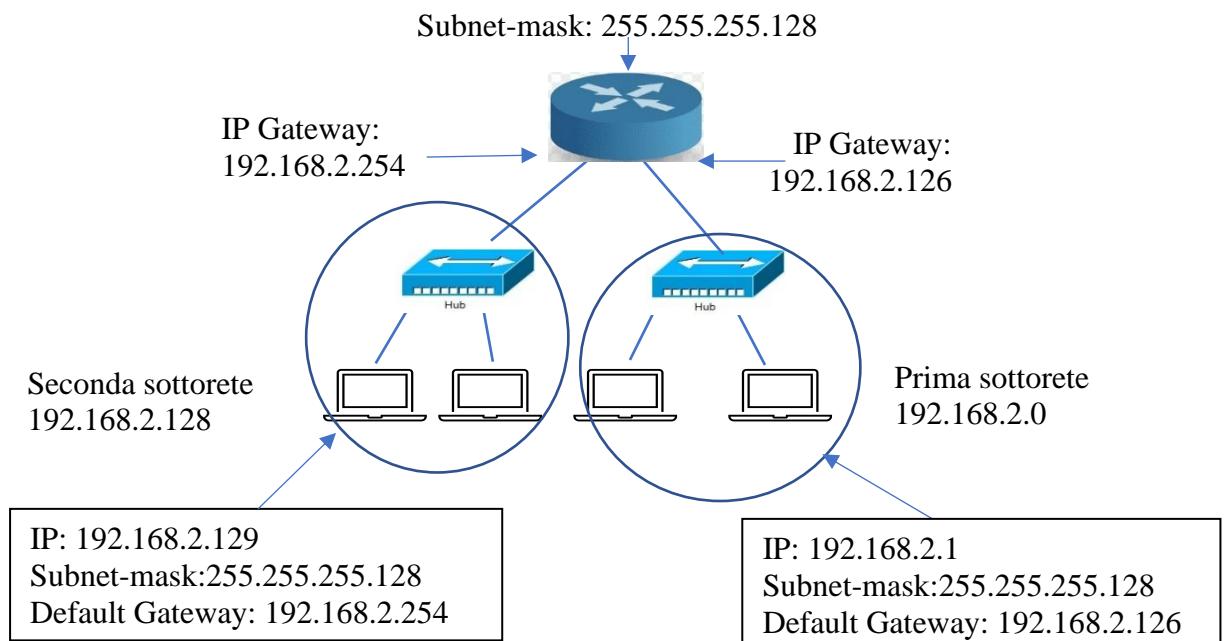
collegabili è: $2^H - 3$ (due combinazioni che non possono essere usate sono quelle formate da 0 che indica l'indirizzo di rete e da tutti 1 per indicare il broadcast + il gateway IP).
Per completare il nostro esempio abbiamo quindi che $H = 8 - 1 = 7 \rightarrow 2^7 - 3 = 128 - 3 = 125$ dispositivi collegabili per ogni rete. In definitiva otteniamo:

Prima sottorete:

NETWORK	192.168.2.0 11000000.10101000.00000010.00000000
BROADCAST	192.168.2.127 11000000.10101000.00000010.01111111
GATEWAY	192.168.2.126 11000000.10101000.00000010.01111110
INDIRIZZI IP ASSEGNABILI	192.168.2.1 – 192.168.2.125 11000000.10101000.00000010.00000001 - 11000000.10101000.00000010.01111101
SUBNET-MASK	255.255.255.128

Seconda sottorete:

NETWORK	192.168.2.128 11000000.10101000.00000010.10000000
BROADCAST	192.168.2.255 11000000.10101000.00000010.11111111
GATEWAY	192.168.2.254 11000000.10101000.00000010.11111110
INDIRIZZI IP ASSEGNABILI	192.168.2.129 – 192.168.2.253 11000000.10101000.00000010.10000001 - 11000000.10101000.00000010.11111101
SUBNET-MASK	255.255.255.128



Esempio

Un'azienda vuole strutturare la sua LAN di indirizzo 200.10.10.0 (di classe C) in tre sottoreti con almeno 20 dispositivi per sottorete per le sue tre aree: commerciale, magazzino, produzione.

In questo caso un solo bit dell'Host-ID non ci basta! Con un bit possiamo suddividere la rete in due sottoreti; quindi, per suddividerla in tre sottoreti abbiamo bisogno di 2 bit (con due bit possiamo dividere la rete in $2^2 = 4$ sottoreti). La subnet-mask sarà quindi:

11111111.11111111.11111111.11000000 -> 255.255.255.192

Il primo indirizzo della sottorete sarà:

11001000.00001010.00001010.00000000 -> 200.10.10.0

Il secondo indirizzo:

11001000.00001010.00001010.10000000 -> 200.10.10.128

Il terzo lo poniamo:

11001000.00001010.00001010.01000000 -> 200.10.10.64

(N.B. avremmo potuto scegliere la combinazione 11001000.00001010.00001010.11000000 -> 200.10.10.192)

In definitiva avremo:

Prima sottorete (commerciale):

NETWORK	200.10.10.0 11001000.00001010.00001010.00000000
BROADCAST	200.10.10.63 11001000.00001010.00001010.00111111
GATEWAY	200.10.10.62 11001000.00001010.00001010.00111110
INDIRIZZI IP ASSEGNABILI	200.10.10.1 – 200.10.10.61 11001000.00001010.00001010.00000001 - 11001000.00001010.00001010.00111101
SUBNET-MASK	255.255.255.192

Seconda sottorete (magazzino):

NETWORK	200.10.10.128 11001000.00001010.00001010.10000000
BROADCAST	200.10.10.191 11001000.00001010.00001010.10111111
GATEWAY	200.10.10.190 11001000.00001010.00001010.10111110
INDIRIZZI IP ASSEGNABILI	200.10.10.129 – 200.10.10.189 11001000.00001010.00001010.10000001 - 11001000.00001010.00001010.10111101
SUBNET-MASK	255.255.255.192

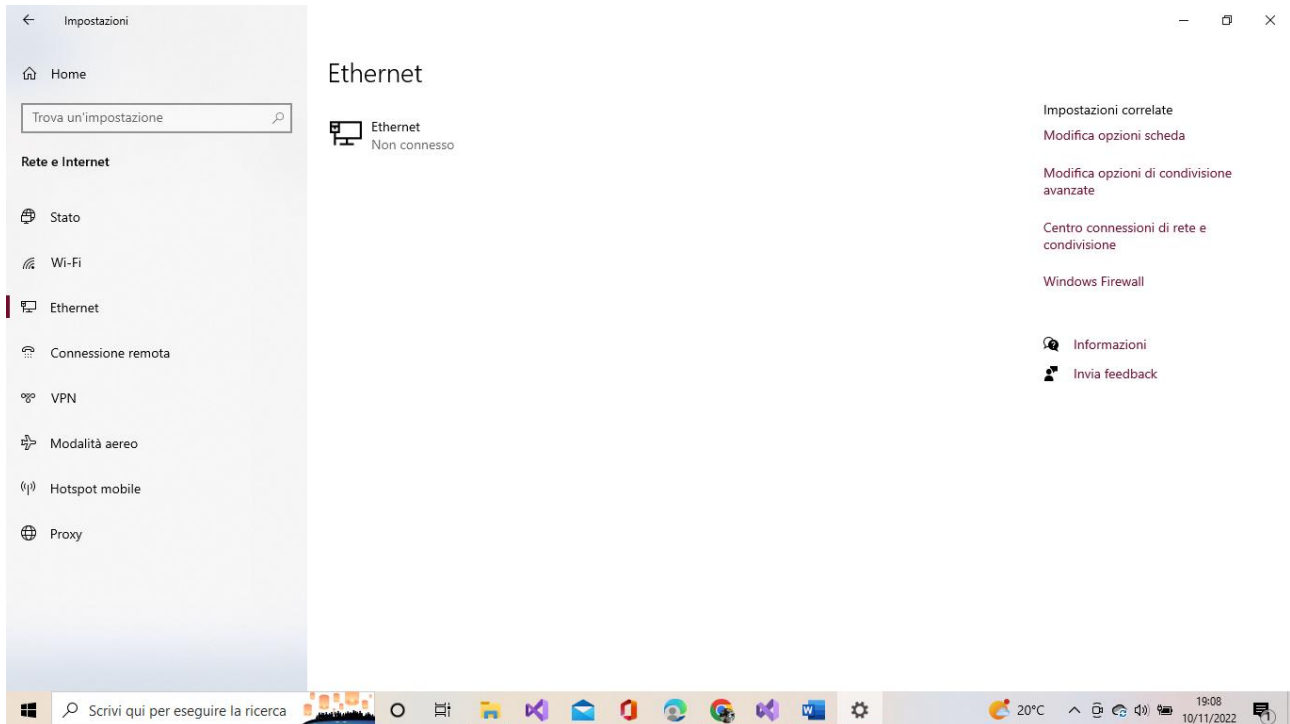
Terza sottorete (produzione):

NETWORK	200.10.10.64 11001000.00001010.00001010.01000000
BROADCAST	200.10.10.127 11001000.00001010.00001010.01111111
GATEWAY	200.10.10.126 11001000.00001010.00001010.01111110
INDIRIZZI IP ASSEGNABILI	200.10.10.65 – 200.10.10.125 11001000.00001010.00001010.01000001 - 11001000.00001010.00001010.01111101
SUBNET-MASK	255.255.255.192

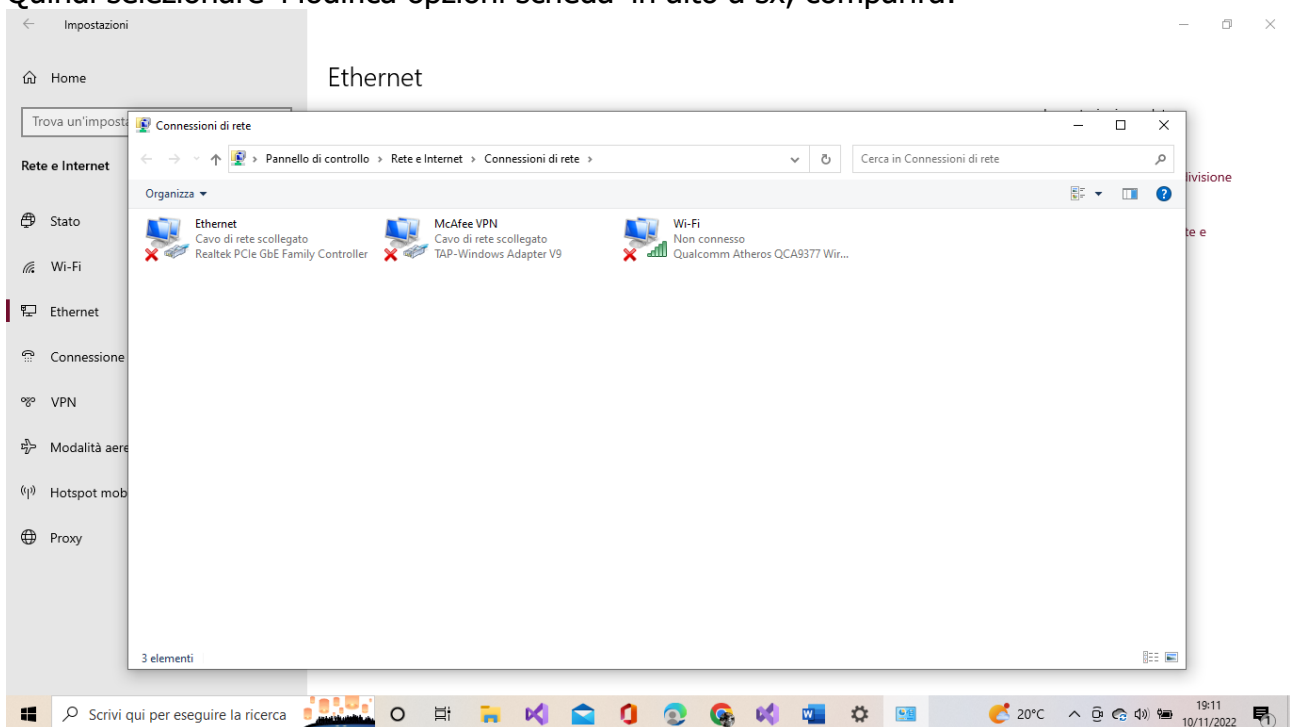
N.B. ogni sottorete può contenere 60 dispositivi quindi più che sufficienti per il problema dato.

Laboratorio: Come modificare le impostazioni della scheda di rete.

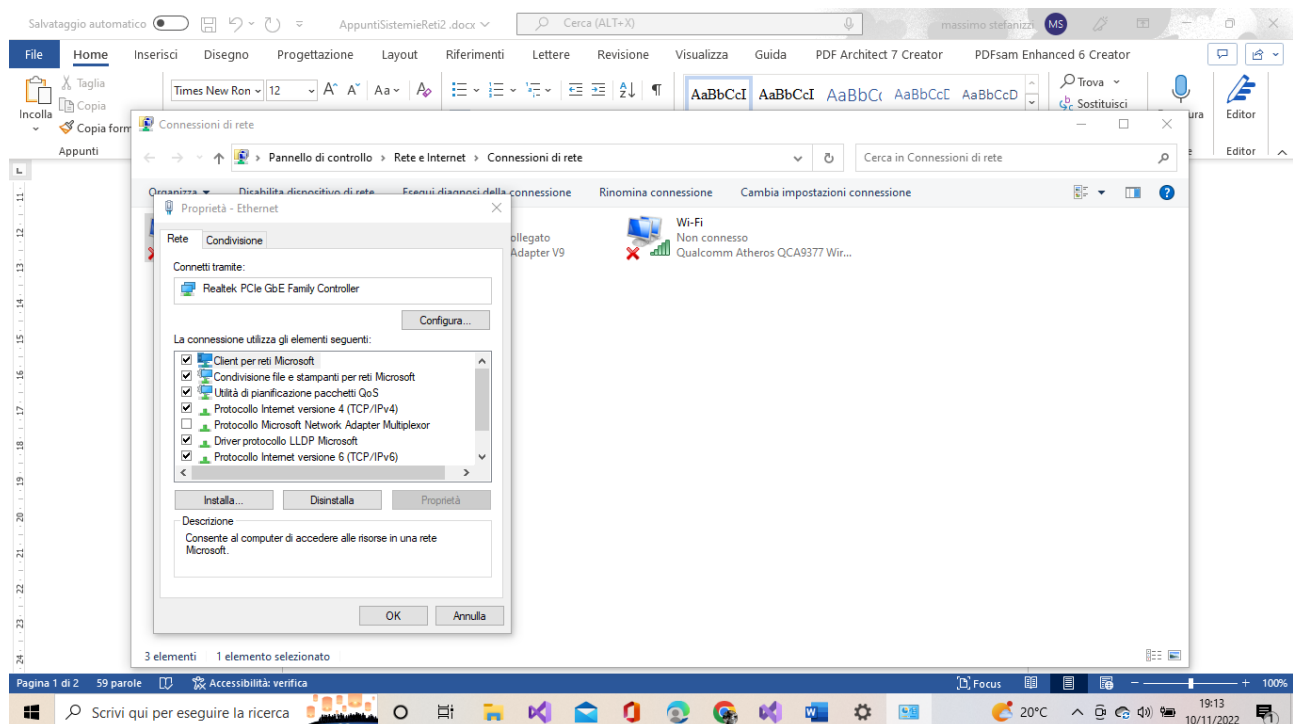
Con il S.O Windows 10 da 'Start' → 'Impostazioni' → selezionare 'Rete e Internet'.
Dalla schermata che vi compare, selezionare dal menù sulla dx del vostro schermo, 'Ethernet'



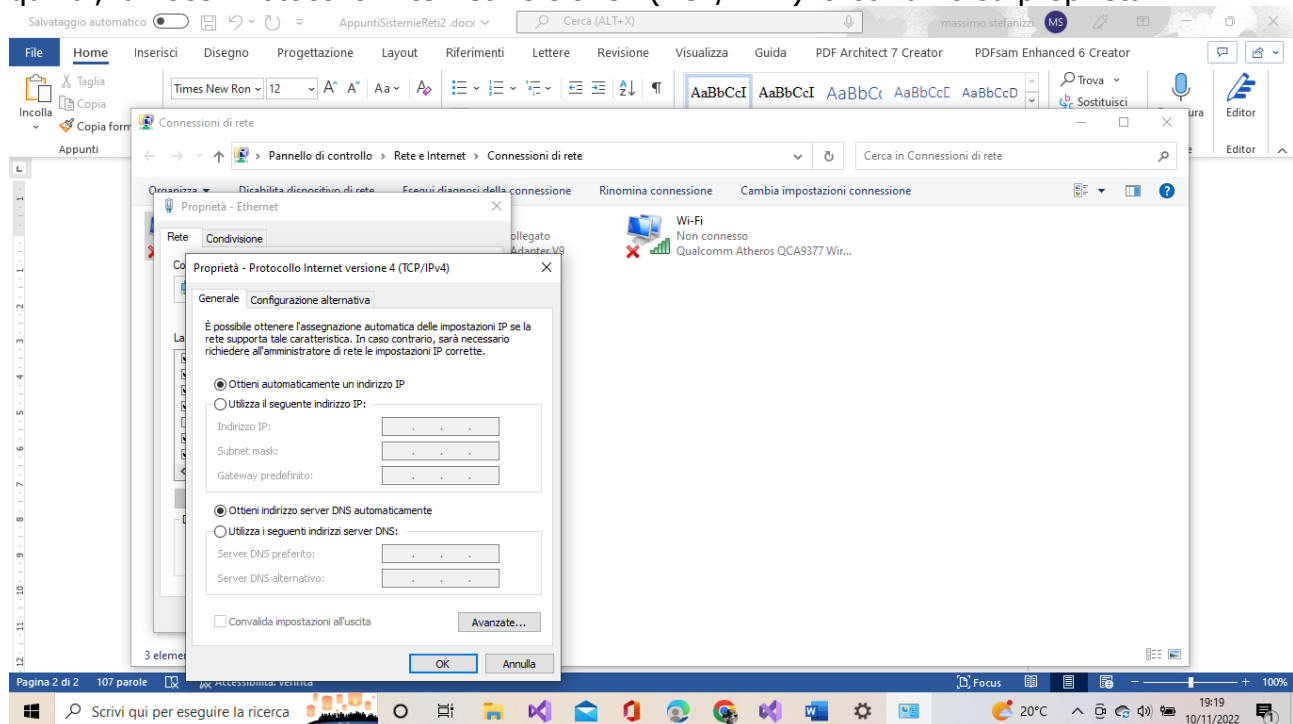
Quindi selezionare 'Modifica opzioni scheda' in alto a sx, comparirà:



Selezionate la rete che vi interessa (nel ns caso Ethernet') cliccandoci due volte sopra.



Tramite la maschera che vi compare si possono effettuare diverse operazioni. Per i nostri scopi si devono modificare/inserire l'indirizzo IP, la subnet mask per realizzare le sottoreti, l'eventuale indirizzo del Gateway predefinito e l'indirizzo dei server DNS. Selezioniamo, quindi, la voce 'Protocollo Internet versione 4(TCP/IPv4)' clicchiamo su proprietà:



La prima opzione 'utilizza automaticamente un indirizzo IP' facciamo assegnare in modo automatico (DHCP) un indirizzo IP al ns computer. Se dobbiamo assegnarlo manualmente clicchiamo su 'utilizza il seguente indirizzo IP:' quindi inseriamo i parametri che abbiamo definito nella progettazione della rete su tutti i sistemi che ne fanno parte.