Riassunto Reti 2

# Il Livello Data Link

Il livello Data Link connette coppie di nodi adiacenti connessi da un mezzo fisico (in cui si diffondono le onde elettromagnetiche) e ha la responsabilità di trasferire tra essi pacchetti incapsulati in frame attraverso un link.

Servizi a livello Data link

* Rilevazione e correzione degli errori;
* Framing: permette l’incapsulamento dei pacchetti in frame;
* Accesso al collegamento: vi è un protocollo che controlla l’accesso al mezzo, specificando le regole con cui immettere i frame. Esso può essere punto-punto, ovvero vi sono solo due entità che utilizzano il mezzo fisico, oppure broadcast quando gli accessi a esso sono molti di più. In entrambi i casi i messaggi vengono inviati solamente quando il canale è libero, *onde evitare conflitti*. Alcuni esempi di canali broadcast sono Internet e le reti wireless, perché tutti condividono il mezzo e per evitare conflitti bisogna disciplinare l’accesso al mezzo.
* Indirizzamento a livello di link: si lascia l’indirizzo a livello link indipendente da quello IP e si creano dei meccanismi per associarli;
* Controllo di flusso: Serve per la sincronizzazione con i dispositivi troppo lenti inviando dati solo quando possibili attraverso un riscontro, *in questo modo si evitano delle trasmissioni inutili.*
* Recapito affidabile tra nodi adiacenti: viene utilizzato soprattutto nei Link ad alto tasso di errore, quindi nella LAN wireless dove questo succede, mentre viene raramente usato nella fibra ottica dove la frequenza di errori è molto più bassa.
* Half duplex e Full duplex: con half duplex i nodi possono trasmettere a turno, perchè il canale per ricezione e trasmissione è unico, mentre con full duplex possono trasmettere in contemporanea perchè ci sono due canali diversi.

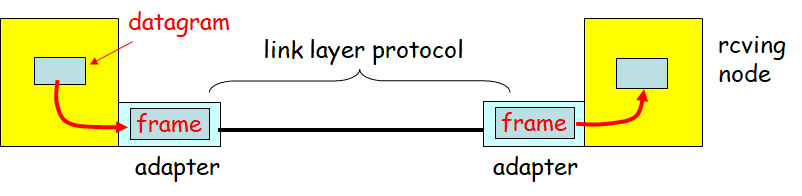
Questi concetti sono implementati con diversi metodi nelle reti di oggi.

Una rete è rappresentabile come un grafo in cui host e router sono i nodi e gli archi sono le linee di collegamento che li connettono, essi possono essere:

* Cablati;
* Wireless;

Una LAN può essere vista come una componente connessa che racchiude una serie di nodi e archi con una sottorete, in essa i frame passano attraverso i cavi o gli switch. I link sia cablati che wireless sono condivisi da più macchine, in questo contesto vengono fuori i problemi di condivisione e i protocolli di accesso multiplo, le macchine sono quindi adiacenti ovvero connesse direttamente a livello fisico e vengono gestite da una singola sottorete IP. Il frame è la struttura dati utilizzata a questo livello, esso presenta un preambolo, un campo dati e una coda indicante la fine, in essa vi è la parte relativa al controllo degli errori, *inserita lì per ragioni di sicurezza ed efficienza così da permettere la ricezione dei dati prima dei controlli, oltre al fatto che i dati arrivano sempre in ordine*. Nelle reti i collegamenti non devono tutti rispettare uno stesso servizio, spetta infatti a questo livello preoccuparsi di questo problema e quindi esporre al livello superiore un’unica interfaccia. Le reti di link possono essere differenti tipi di collegamenti con diversi tipi di servizi.

## Implementazione del livello

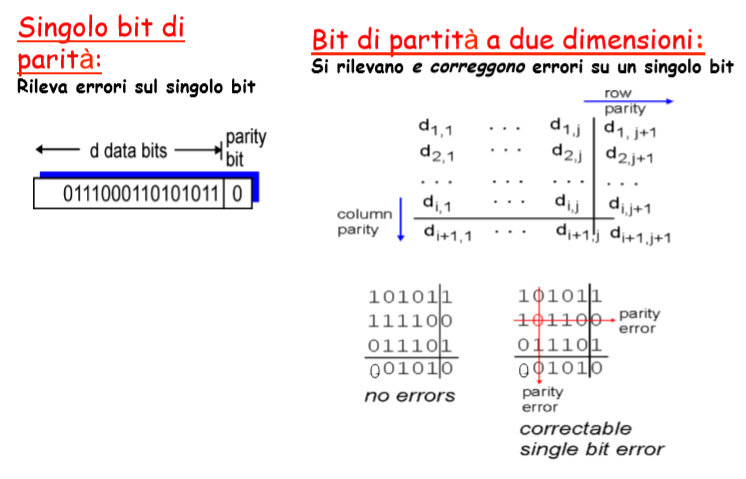
Il livello Data link è implementato in un chip detto adattatore di rete, la maggior parte delle funzionalità è quindi implementata a livello hardware. In trasmissione il pacchetto viene incapsulato in un frame riempiendone i vari campi e, seguendo le regole del protocollo d’accesso, lo trasmette al destinatario. In ricezione si controlla il frame per verificare che non ci siano errori,in tal caso viene richiesta la ritrasmissione oppure si inviano degli avvisi al livello superiore. L’adattatore gestisce tutti i segnali per la trasmissione/ricezione, il controllo del canale e tutti gli accessi relativi a esso se condiviso. *Le operazioni non devono essere troppo complesse a livello logico dato che il tempo di elaborazione influisce molto a questo livello.* 

## Rilevamento degli errori

La rilevazione degli errori viene implementata aggiungendo dei bit detti EDC (Error Detection and Correction), *essi servono per proteggere i dati all’interno del frame* dato che vi sono indirizzi MAC e altre informazioni sensibili.

EDC viene calcolato con un algoritmo real time, inserito nel frame e trasmesso, in ricezione questi bit vengono ricalcolati e si confrontano con quelli presenti all’interno. *Se non ci sono coincidenze significa che ci sono 1 o più errori*, tuttavia è rarissimo averne uno su un mezzo fisico (l’unico caso è il wi-fi). Questo controllo è presente anche a questo livello *dato che a livello trasporto i dati possono essere corrotti o persi*, in più nel caso del TCP vi sono rallentamenti causati dalla congestione, quindi questa “ripetizione” permette *maggior affidabilità.*

### Controllo di parità

Il controllo di parità è la forma più semplice di rilevamento degli errori e consiste nell’inserimento di 1 o più bit di parità, essi permettono un controllo facile da implementare ma allo stesso tempo più fragile, *infatti se due (o multipli) bit a 1 diventano 0, l’errore non viene rilevato, in più non è possibile capire quali bit sono corrotti*. Un modo per rendere questo controllo più efficiente è organizzare le stringhe di bit in una matrice e calcolare la parità su righe e colonne, in questo modo è possibile rilevare l’errore e quindi correggerlo. Questo schema permette di rilevare ogni combinazione di due errori e di correggere quelle che ne hanno soltanto uno, questa capacità è detta Forward Error Correction.

### Checksum di Internet

Il checksum di internet permette di rilevare gli errori trattando la stringa di bit come una sequenza di numeri interi di 16 bit, essi verranno poi sommati e verrà effettuato il complemento a 1, il risultato verrà inserito in un frame e inviato al ricevente. Quest’ultimo eseguirà lo stesso processo sui dati ricevuti (checksum incluso) e controlla se i bit sono tutti a 1, se non è così viene segnalato l’errore.

### Controllo a ridondanza ciclica

Dal momento che gli errori sono spesso annidati tutti insieme, si è deciso di utilizzare algoritmi che ragionano in questo modo. Il CRC funziona utilizzando una finestra grande L+1 bit, gli errori all’interno di essa verranno tutti rilevati, dato D grande d bit rappresentante i dati, esso viene allungato di r bit, col numero ottenuto viene deciso R di r bit e con esso si fa la divisione con D\*. Il resto di D\*/R verrà messo negli r bit finali di D\*, *rendendolo quindi divisibile per R*. In ricezione si controlla la presenza di eventuali errori dividendo il frame con R, se non è presente resto, allora il frame è corretto e i dati vengono passati al livello superiore, altrimenti viene scartato. Con questo sistema è possibile rilevare tutti gli errori presenti nella finestra, la correzione non ha importanza *dal momento che il frame corrotto viene scartato e si chiede la ritrasmissione.*

## Protocolli di accesso multiplo

In una rete possono esserci due tipi di connessione:

* Punto-punto: permette il collegamento tra due dispositivi di rete;
* Broadcast: più dispositivi vengono connessi a un altro utilizzando un unico mezzo.

In quest’ultimo caso, dato che il mezzo è condiviso, le onde possono andare in conflitto tra loro, quindi si deve adottare una delle seguenti soluzioni:

* Assemblea: vi è un moderatore che mette in lista gli host quando hanno bisogno di inviare i dati;
* Libero: Dal momento che non ci sono moderatori, gli host devono accordarsi per l’invio/ricezione dei dati, questo modello è più pratico e facile da gestire.

A livello logico, il mezzo fisico è rappresentato come un bus che connette tutti gli host a un unico dispositivo. Il protocollo di accesso gestisce l’accesso multiplo al mezzo ed è diverso da quello utilizzato nella trasmissione, esso infatti determina come i nodi condividono il canale e attua tutti i controlli di cui si ha bisogno. Il canale stesso serve a gestire la condivisione, tutte le operazioni eseguite avverranno solo su esso e non esistono canali “fuori banda”. *Per fare in modo che ogni informazione del canale venga ricevuta da tutti, occorre che tutti gli host inviino dati uno alla volta*, di conseguenza la banda è sempre utilizzata al massimo, tuttavia non è mai così: il tasso medio è quindi suddiviso equamente tra tutti gli host (Rate/#host). Il broadcast è decentralizzato e non ha moderatori, *dal momento che la loro presenza potrebbe causare problemi in caso di rottura*, in più non vi sono sincronizzazioni e quindi bisogna cercare algoritmi che non utilizzino il tempo in modo assoluto. La semplicità è importante dato che permette di ottenere una velocità maggiore.

### Protocolli MAC (Media Access Control)

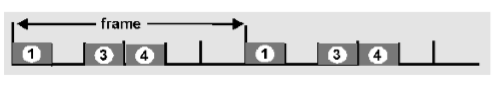
I protocolli MAC sono protocolli che permettono di identificare i dispositivi attivi e quindi coordinarne le trasmissione, possono essere di 3 tipi:

* A partizionamento di canale: si divide il canale in parti più piccole e ogni host ne riceve una porzione di esse a uso esclusivo;
* Ad accesso casuale: il canale non è suddiviso e per evitare collisioni si utilizza la “buona educazione”, ovvero si rimedia quando avvengono;
* A turno: gli host decidono un’ordine di invio/ricezione e utilizzano il canale a turno.

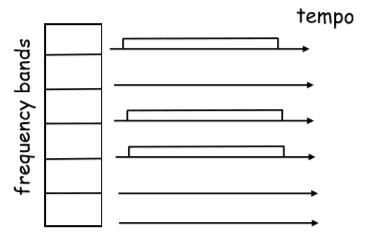
#### Protocolli a partizionamento di canale

##### TDMA (Time Division Multiple Access)

TDMA è un protocollo che sfrutta la tecnica TDM definendo una temporizzazione in frame composta da slot di lunghezza fissa, ogni slot viene assegnato a ogni host connesso e permette la trasmissione di un singolo pacchetto. Questa tecnica permette di evitare le collisioni e utilizza una politica imparziale: in ogni slot vi è un tasso trasmissivo pari a R/N. Questo protocollo però presenta due inconvenienti: il tasso trasmissivo è vincolato a R/N anche se vi sono host che non fanno niente, determinando anche uno spreco di slot, in più se si vuole trasmettere bisogna attendere il proprio turno.



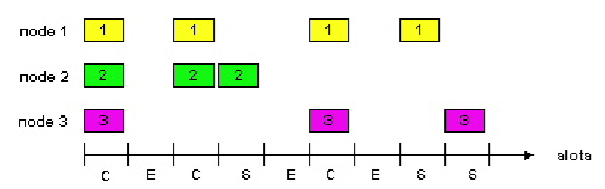
##### FDMA (Frequency Division Multiple Access)

Il FDMA è un protocollo che sfrutta la tecnica FDM per dividere il canale in sottocanali con rate R/N, ognuno assegnato a un host. L’unico problema in questo caso è la fisica, infatti i segnali nel mezzo trasmissivo vengono mischiati e separati in ricezione attraverso dei filtri. FDMA ha anche il problema di avere una banda limitata a R/N bps anche se c’è solo un host che vuole trasmettere, quindi tutta quella non utilizzata viene sprecata.

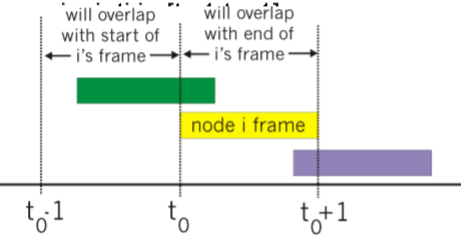
Entrambi i protocolli hanno le stesse prestazioni e non presentano penalità tra loro.

### Protocolli ad accesso casuale

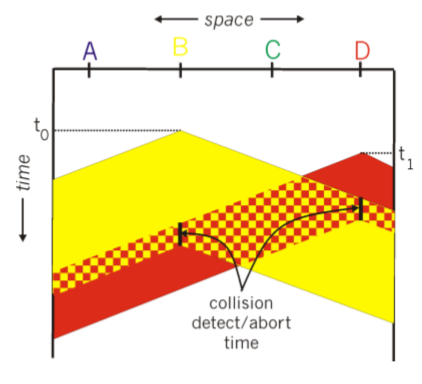
##### Slotted ALOHA

Slotted ALOHA è un protocollo in cui tutti i frame sono della stessa dimensione (L), essa utilizza TDM dividendo il tempo in slot lunghi L/R secondi. Quando un host vuole trasmettere, può farlo all’inizio di ogni slot, in questo modo *le collisioni saranno concentrate solamente in quella parte*. Nel caso avvenga una collisione, essa viene percepita da tutti gli host e il frame viene trasmesso negli slot successivi con una probabilità p finchè non riesce ad inviarlo correttamente. *Per diminuire il numero di collisioni, gli host vengono sincronizzati all’inizio di ogni slot*. L’efficienza di Slotted ALOHA può essere calcolata come la frazione di slot in cui è avvenuta la trasmissione, nel caso ideale non vi sono collisioni e quindi si ha la massima efficienza, tuttavia non è così, per calcolarla si utilizzano delle variabili binomiali indipendenti: data p come probabilità di avvenuta trasmissione, la probabilità che tutti i nodi trasmettano con successo è , con n che tende a infinito la formula tende a 1/e, quindi il canale è utilizzato il 37% delle volte con delle trasmissioni utili nel caso migliore.

##### ALOHA Puro

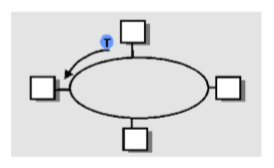
Questa è la prima versione di ALOHA in cui non sono presenti nè slot nè sincronizzazione, di conseguenza le collisioni avvengono anche quando il frame è in trasmissione nello stesso istante di un altro anche se quest’ultimo ha cominciato prima, questo lo rende poco efficiente rispetto alla versione Slotted. Dal momento che la probabilità di avere collisioni raddoppia a causa dei motivi citati prima, l’efficienza si calcola nel seguente modo: , quindi con n tendente a infinito si ha la formula tende a 1/(2\*e), corrispondente al 18% del tempo in cui la trasmissione ha successo nel caso migliore.

##### CSMA (Carrier Sense Multiple Access)

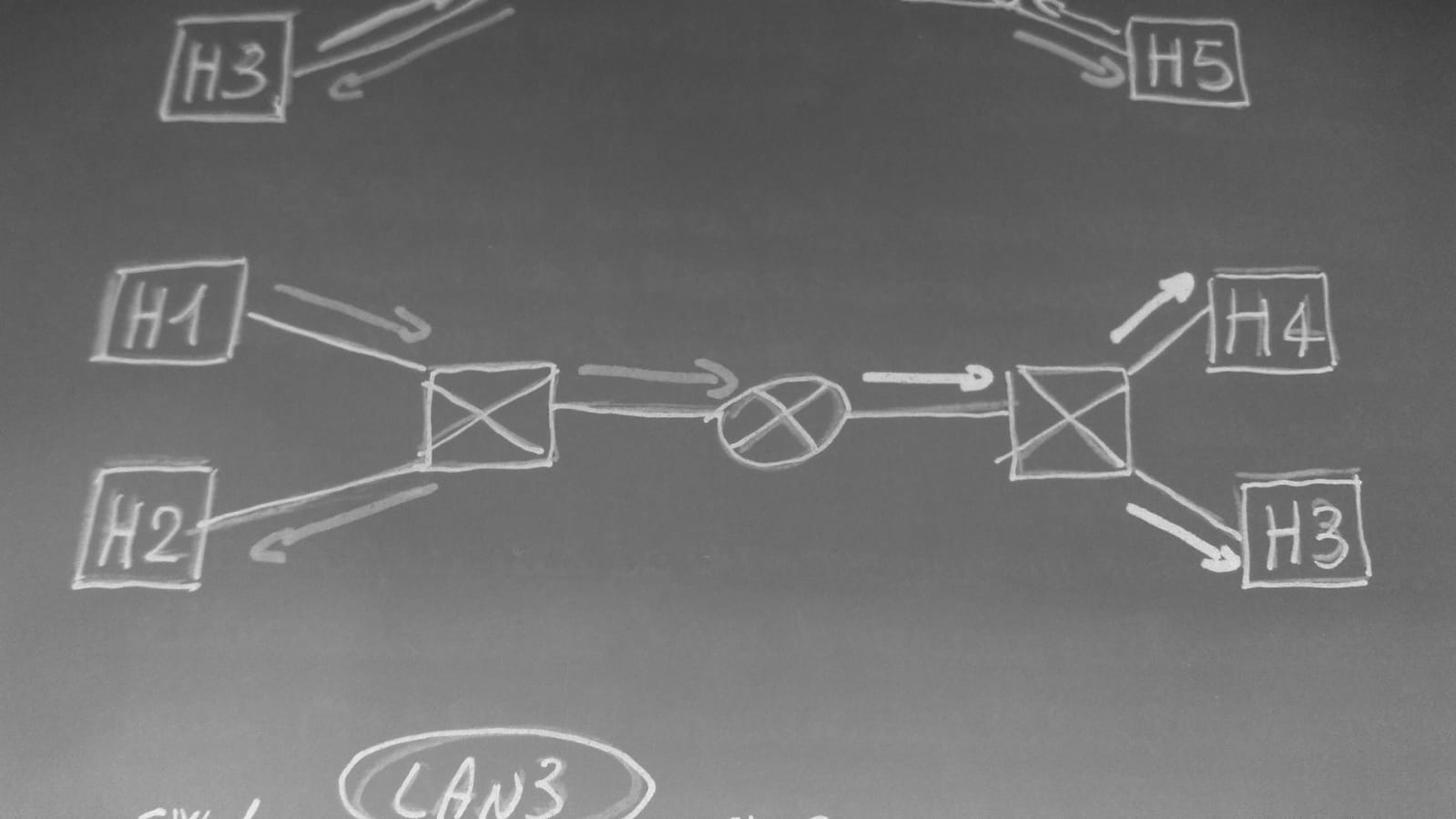
CSMA è un protocollo che controlla se il canale è libero prima della trasmissione, in tal caso trasmette, altrimenti no. Le collisioni rimangono tuttavia ancora presenti, questo perché quando un host sta ascoltando il canale e un altro sta già trasmettendo, c’è un lasso di tempo in cui il canale risulta ancora libero da entrambe le parti *a causa del tempo di propagazione* e, di conseguenza, inizia la trasmissione creando una collisione. Per evitare ciò, o si diminuiscono le distanze tra host oppure si rilevano le collisioni.

Per rilevare le collisioni, l’idea è ascoltare il canale mentre si trasmette confrontando quello che si sta trasmettendo con quello che si sente sul canale, in questo modo è possibile rilevare se altri nodi stanno trasmettendo e quindi interrompere la trasmissione facendo tornare pulito il canale, in seguito si ritrasmette. Questo ragionamento è utilizzato da CSMA/CD.

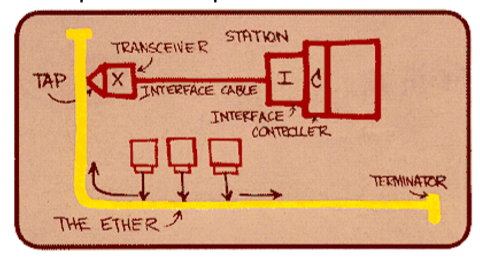
#### Protocolli ad avvicendamento

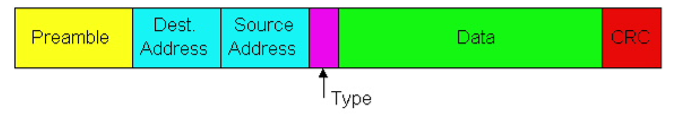
I protocolli ad avvicendamento sono protocolli che vogliono evitare le collisioni ma anche mantenere un allocazione dinamica della banda, essi vanno bene quando vi sono degli utilizzatori sempre attivi. Nell’avvicendamento basato sul polling vi è un master che invita gli slave a trasmettere fino a un dato numero di frame, questo però può causare overhead, latenza e single point of failure sul master. Un’altra soluzione è l’avvicendamento basato su token, in questo caso vi è un token di controllo che viene passato da un host all’altro come in una staffetta, questo viene trattenuto quando un host deve trasmettere, tuttavia l’invio del token occupa banda e, se viene perduto, bisogna rigenerarlo. Una rete a token utilizza una topologia ad anello in cui esso gira, tuttavia è preferibile una rete a bus in cui il token “gira logicamente” per evitare problemi riguardanti la topologia.

## Indirizzamento

L’indirizzamento a questo livello è dato dagli indirizzi MAC, essi non si riferiscono all’host stesso ma alla sua interfaccia di rete, quindi se ne possiede molteplici, possederà anche altrettanti indirizzi. Gli indirizzi MAC sono formati da 48 bit e sono solitamente univoci, per sapere a quale indirizzo MAC si riferiscono viene utilizzata la tavola di ARP (Address Resolution Protocol), una tabella in cui vi sono tutte le corrispondenze IP-MAC. Ogni riga della tabella possiede tre campi, ovvero IP, MAC e TTL, quest’ultimo serve per sapere se l’host corrispondente è ancora presente in rete e, al suo scadere, la riga viene cancellata, in caso contrario il campo viene rinnovato ogni volta che arriva un frame per quella corrispondenza. Inizialmente la tabella ARP è vuota, quindi si invia un frame ARP utilizzando l’indirizzo di broadcast (corrispondente a FF-FF-FF-FF-FF-FF, utilizzato quando non vengono trovate corrispondenze nella tabella), il suo scopo è quello di comunicare con tutti gli altri nodi della LAN per individuare l’indirizzo MAC corrispondente a un dato IP e quindi compilare la tabella. ARP è possibile anche quando un nodo non è nella propria sottorete, l’operazione in tal caso è più complessa: prima di tutto viene fatto un ARP col router della sottorete da interrogare, esso farà da ponte per effettuare un secondo ciclo dell’algoritmo in cui vengono interrogati gli host. ARP è necessario con la versione 4 di IP, *dal momento che quest’ultimo non può contenere il MAC per via della lunghezza minore*, questo metodo è invece possibile con la versione 6 del protocollo, *tuttavia non viene mai utilizzata dal momento che rende possibile la tracciabilità degli host a livello di rete*. Come soluzione si utilizza la Neighbour Discovery: si genera un numero IP a caso e si verifica attraverso un messaggio se esiste già nella rete.

### Ethernet

Ethernet è tutt’ora la tecnologia dominante per le LAN cablate, dato che è più semplice ed economica delle LAN a token e di ATM, essa era formata da un adattatore a basso livello connesso al cavo coassiale, agli estremi di quest’ultimo vi sono 2 terminatori e in mezzo ulteriori adattatori, in seguito si è passati a una versione più distribuita formata da quattro doppini intrecciati, utilizzando hub o switch per connettere più host tra loro. Ethernet si è però diffusa maggiormente grazie alla topologia a stella, una rete in cui un device centrale inoltra i dati su tutte le vie, essa può essere uno switch oppure un hub. Un frame ethernet è formato da vari campi:

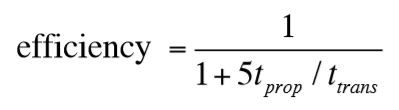
* Preambolo: 7 byte con pattern 10101010 seguita da 10101011, essi servono per la sincronizzazione con il device ricevente;
* Indirizzi MAC di destinazione e partenza, entrambi formati da 6 byte;
* Tipo di frame: indica il protocollo di rete che utilizza i dati;
* Dati: parte in cui vi sono i dati veri e propri, solitamente è un pacchetto IP;
* CRC: codice che serve per il controllo degli errori da parte del ricevente, se essi avvengono, il frame viene scartato ma non viene avvisato il mittente, questo infatti è compito del livello di rete, *per questo motivo è un servizio inaffidabile*.

la lunghezza di una frame è variabile e, per identificarla, si lascia un minimo di silenzio tra un frame e l’altro per assumerne la fine. La lunghezza minima di un frame è data dal fatto che è possibile rilevare collisioni con l’host più lontano dalla rete, quella massima dipende invece dal protocollo di destinazione utilizzato. Inizialmente ethernet venne utilizzato nelle reti a stella ad hub, una topologia in cui vi è un device centrale che propaga il segnale su tutte le altre vie.

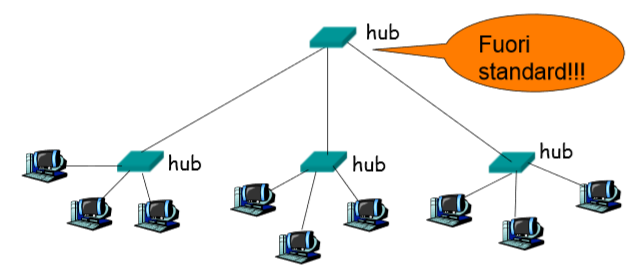
#### Accesso al mezzo e CSMA/CD

L’accesso al mezzo trasmissivo avviene tramite il protocollo CSMA/CD, esso ascolta il mezzo e, dopo un intervallo di tempo casuale *per evitare le collisioni*, comincia a trasmettere, in caso di collisione si genera un segnale di jam che prende il posto del CRC, il suo compito è evitare che il frame corrotto sia preso in considerazione, in seguito aspetta un altro intervallo di tempo casuale (questa volta causato dalla collisione stessa) e ricomincia a trasmettere quando esso finisce. Il ritardo è dato dall’algoritmo di back-off esponenziale: con n collisioni nella rete viene scelto un numero compreso tra 1 e , ad ogni collisione l’intervallo di tempo per il nodo trasmettitore aumenta esponenzialmente (dopo 10 tentativi si riporta l’errore al livello di rete, *onde evitare attese troppo lunghe*). Il numero scelto viene moltiplicato per 512, *il multiplo di tempo necessario per trasmettere un bit*, esso dipende anche dalla velocità della rete. A causa dell’enorme traffico che causa collisioni nelle reti a stella, si è deciso di modificarle cambiando l’hub con uno switch *dato che il funzionamento di quest’ultimo non permette collisioni.*

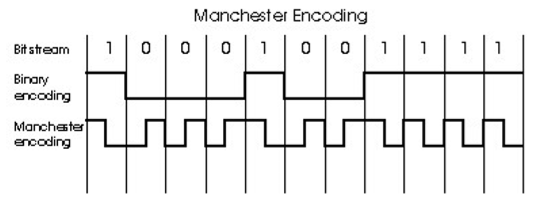
L’efficienza di CSMA/CD è inversamente proporzionale al rapporto tra tempo di propagazione e tempo di trasmissione, infatti se il primo tende a 0, l’efficienza tende a 1, essa tenderà sempre a 1 anche quando il tempo di trasmissione tende a infinito dato che il canale rimane occupato per molto tempo (*la rete rende meglio in presenza di frame lunghi*), di conseguenza vi sono meno collisioni.



#### Rete a stella con hub

Questo tipo di rete è molto simile a quella a bus dato che il funzionamento è praticamente lo stesso, l’unica differenza è la dimensione del cavo Ethernet, la sua dimensione massima è 100 metri. Ethernet è formata da 4 doppini intrecciati per evitare interferenze, la versione più comune è detta UTP perchè non presenta schermature, altre versioni sono FTP e STP i quali presentano una schermatura in alluminio (più accurata nel secondo). Ethernet utilizza 2 doppini per la trasmissione/ricezione, gli altri possono essere utilizzati per il PoE o altro. Ethernet è inaffidabile e senza connessione, non è presente handshaking tra adattatori e non vi sono conferme per l’avvenuta ricezione. Le collisioni avvengono ancora sull’host, in tal caso però la soluzione è più trasparente dato che *la struttura è rilevabile solo a livello fisico e non data link, visto che essa cambia nel primo livello.* Considerando infatti le macchine più lontane nella rete, si ha un tempo t dato dalla trasmissione da parte di una delle due macchine e, se l’altra trasmette in questo intervallo, provoca un RTT e una collisione: se il primo host che ha iniziato a trasmettere finisce prima di rilevarla(e quindi prima di RTT), non si accorge che il frame è corrotto, per questo motivo la durata minima della trasmissione di un frame è di un RTT. Un problema comune degli hub è che, ripetendo il segnale su tutte le linee, lo rallentano, di conseguenza più hub in cascata provocano rallentamenti a catena e l’impossibilità di rilevare le collisioni in tempo, *dato che si ha una sorta di “allungamento” della rete*. Per evitare questi problemi, la lunghezza dei cavi UTP deve essere limitata a 100 m al massimo e non vi possono essere più di 2 hub in cascata, questo *perché il numero di collisioni è direttamente proporzionale al tempo di propagazione.* L’hub è stato costruito in modo da rimpiazzare il bus e i suoi difetti, esso ne emula il comportamento dal momento che è un’installazione fisica tra tutte le porte, permettendo anche di riformare il segnale come l’originale al costo di un piccolo ritardo. Questo device ha segnato il passaggio dalla topologia a bus a quella a stella e tutto ha funzionato fino a quando il numero di accessi non è vertiginosamente aumentato, inchiodando tutto di conseguenza.

#### Codifica di Manchester

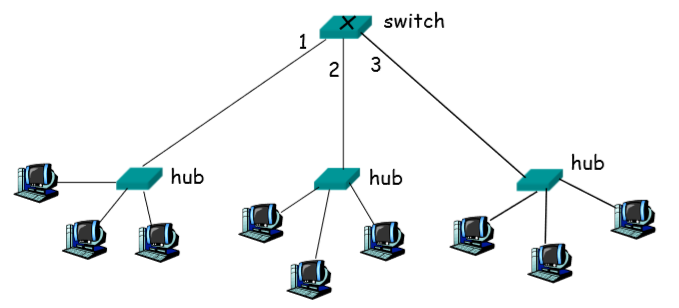
La codifica di Manchester permette di codificare a livello fisico i segnali sui doppini di Ethernet, essa utilizza le transizioni da 0 a 1 e viceversa per rappresentare rispettivamento 0 e 1 *per essere comprensibile anche senza sincronizzazioni*. Essa ha però un problema: se vengono trasmessi 2 valori uguali, il primo viene riconosciuto mentre il secondo no, per ovviare ciò si riposiziona il segnale alla fine di ogni slot. Per riuscire a catturare la transizione, i dispositivi devono essere sincronizzati abbastanza in modo da poterla catturare, questo è possibile grazie al preambolo all’inizio di ogni frame, esso infatti contiene un’onda quadra che permette ciò, la trasmissione del resto del frame viene segnalata grazie all’ultimo ottetto (10101011 invece di 10101010).

#### Rete a stella con switch

Per i motivi citati prima, l’hub nelle reti a stella è stato sostituito da uno switch, esso permette di memorizzare il frame e inoltrarlo direttamente al destinatario senza causare collisioni, la destinazione infatti viene scelta in un modo simile al funzionamento del router.

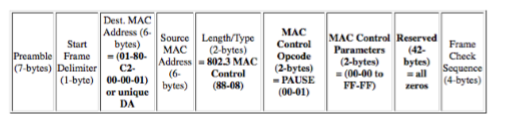
Gli switch introducono un tempo di ritardo causato dallo store & forward, per risolvere ciò vi sono due possibili soluzioni:

* cut through: se la porta in cui il frame deve essere trasferito è libera, esso viene trasferito subito senza memorizzarlo, *dal momento che si deve solo attendere preambolo e indirizzi*.
* dare priorità ad alcuni frame in caso di accodamenti.

Questo componente esegue l’instradamento attraverso una tabella in cui vi sono le corrispondenze tra la porta e l’indirizzo MAC dell’host connesso, la compilazione avviene tramite l’autoapprendimento: all’arrivo di un frame da una porta, si compila la tabella inserendo il MAC del trasmettitore e la relativa porta d’ingresso. Se nella tabella non è presente la corrispondenza relativa a un frame, quest’ultimo viene inviato su tutte le altre porte in broadcast, *l’host ricevente sarà l’unico a rispondere e quindi la tabella viene aggiornata col suo MAC e la relativa porta*. Per ogni riga della tabella vi è un TTL, come per ARP la riga viene cancellata al suo scadere e viene ripristinato quando arriva un frame da quella coppia. Lo switch è trasparente agli altri host presenti in rete, esso non possiede neanche un indirizzo MAC, quindi un host che vuole trasmettere indica direttamente l’indirizzo IP e MAC di destinazione lasciando il lavoro sporco al componente in questione, *questo perché un host deve vedere gli altri device come direttamente connessi*. Una particolare categoria di switch è quella degli switch managed, essi permettono di mettere dei filtri e altre cose a una data sottorete, in più posseggono un indirizzo MAC a differenza di quelli normali utilizzato per la gestione. Come per l’hub, più switch in cascata rallentano i frame *a causa dello store & forward*, lo standard attuale è dai 7 ai 9 switch in cascata. Una cosa molto importante è evitare i loop, *se così fosse i frame potrebbero corrompere la tabella e non arrivare mai a destinazione.*

#### Controllo di flusso per switch

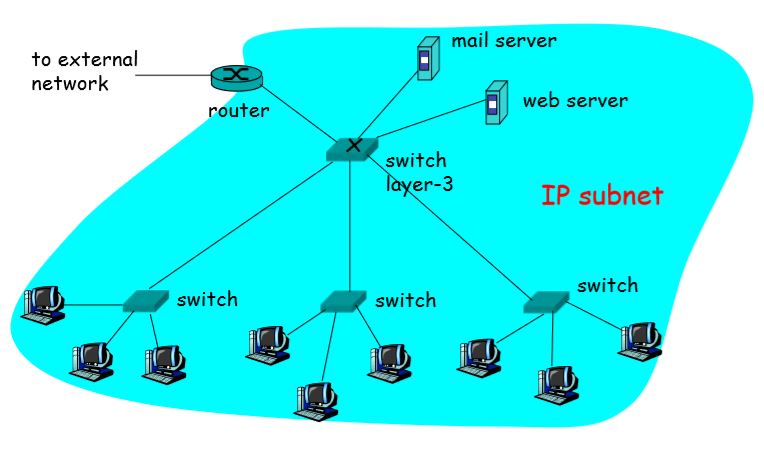
Una caratteristica dello switch moderno è il fatto di essere full-duplex, di conseguenza bastano due doppini del cavo UTP per la trasmissione/ricezione per annullare le collisioni, dato che il cavo comprende una porta TX e una RX nello stesso connettore. Tutt’oggi le porte si auto configurano adattandosi al cavo, tuttavia con più comunicazioni in parallelo interviene il controllo di flusso *per evitare la saturazione*: in poche parole si frena il traffico in entrata per smaltire quello presente, per farlo si utilizzano due approcci:

* Si emulano delle collisioni inviando dei JAM agli host, bloccandoli per un dato tempo;
* Si utilizza un apposito frame per indicare a chi lo riceve di rallentare o addirittura fermarsi per un dato tempo.

#### Caratteristiche degli switch

Gli switch sono full duplex, quindi possono gestire il traffico entrante/uscente contemporaneamente utilizzando 4 linee (2 doppini), hanno una combinazione di interfacce condivise/dedicate, permettendo la negoziazione della banda e la gestione di differenti flussi su differenti porte. Ultima ma non meno importante, gli switch supportano le VLAN quando sono managed.

#### Switch Managed (switch di livello 3)

Gli switch managed sono componenti che combinano le funzionalità dello switch con quelle del router, dal punto di vista logico sono separate mentre da quello tecnico si hanno delle ottimizzazioni nei costi. Questi tipi di componenti si comportano come switch quando hanno a che fare con delle sottoreti mentre fa da router nel momento in cui bisogna interfacciarsi verso l’esterno, inoltre la presenza di un firewall interno permette di effettuare eventuali controlli prima dell’inoltro. *Con l’avvento degli switch*, l’utilità del CSMA/CD viene meno, tuttavia è ancora utilizzato per la retrocompatibilità coi vecchi sistemi. 

#### Switch e router a confronto

Come già detto in precedenza, entrambi i componenti utilizzano lo store & forward per l’inoltro dei frame/pacchetti, mantengono delle tabelle e gestiscono degli indirizzi, rispettivamente gli indirizzi MAC per gli switch e quelli IP per i router. L’unica differenza tra switch e router sono che i primi compilano le tabelle con l’apprendimento, i secondi attraverso appositi algoritmi. All’esecuzione di ARP, lo switch riceve il frame e, dato che il MAC non è in tabella, lo invia in broadcast, presentando una vulnerabilità: se c’è qualcuno che fa sniffing, esso può sapere tutte le macchine presenti in quella LAN e quindi comunicare con esse e fare danni, infatti le app di scan della LAN effettuano la connessione su porte note per verificare se sono attive o meno.

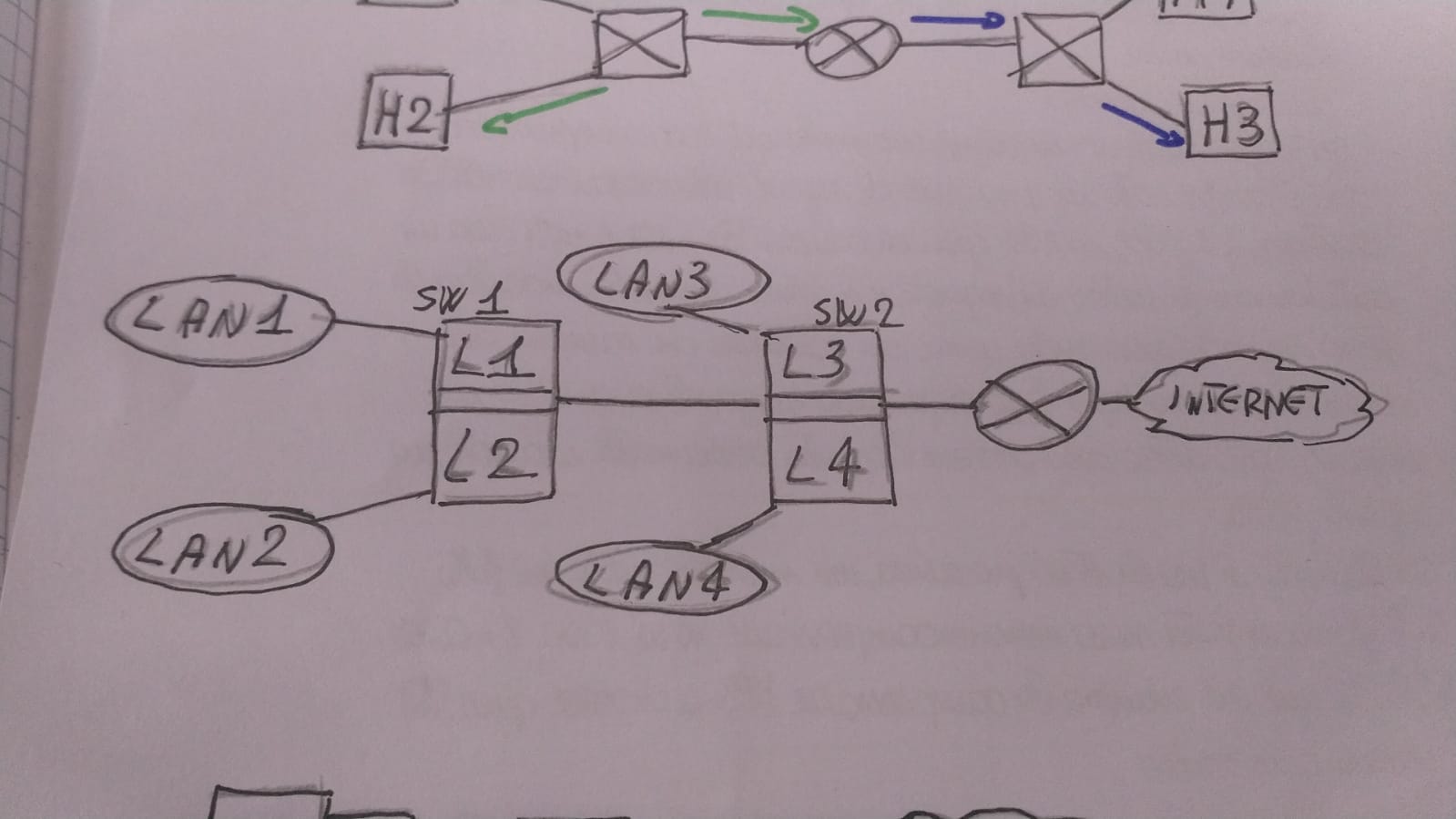
#### VLAN (Virtual Local Area Network)

Una VLAN è una particolare rete locale che permette di multiplexare più LAN nello stesso circuito , l’idea è nata all’inizio degli anni 2000 e permette di correggere i seguenti problemi:

* Permette l’isolamento del traffico: Il traffico di ogni LAN è limitato solo a essa in caso di broadcast e lo è anche per la sicurezza e la riservatezza dei dati;
* Permette di risolvere l’uso inefficiente degli switch;
* Gestione degli utenti: un utente può ora connettersi tra più gruppi senza cambiare la posatura della rete.

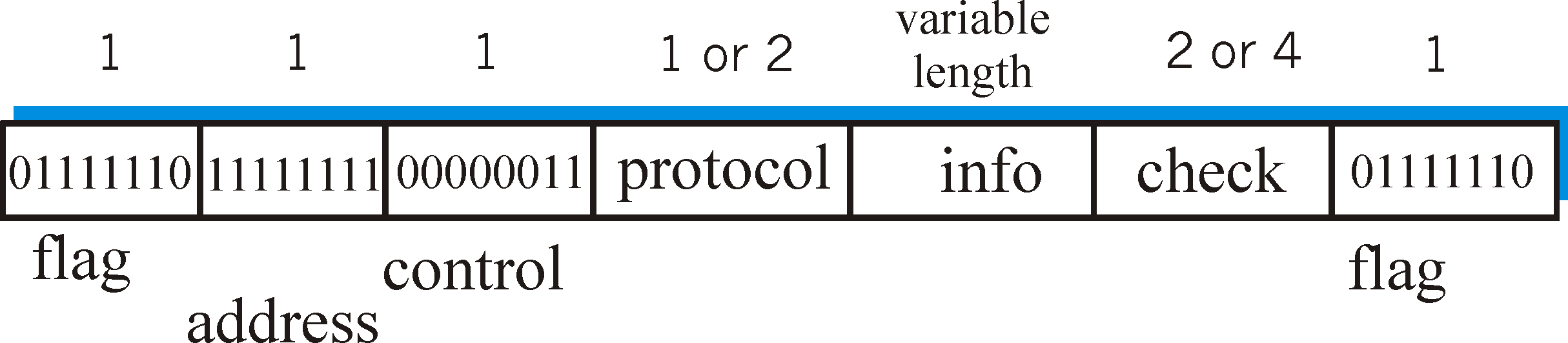
Gli utenti all’interno di una VLAN comunicano tra loro come se connessi allo stesso switch, le porte infatti vengono divise dal gestore della rete in gruppi, ognuno di essi costituisce una VLAN.

Esempio: supponiamo di avere 2 piani di un edificio, ognuno con 2 laboratori, si vogliono connettere questi a due a due verso un router centrale avendo a disposizione un solo cavo. Grazie alle VLAN, è possibile connettere 2 laboratori con uno switch managed tra loro, infine connettiamo quest’ultimo al secondo switch e al router, il risultato è rappresentato dalla seguente immagine:



Per mantenere il traffico separato, si dividono logicamente gli switch assegnando le porte: una parte sarà riservata ad un laboratorio, l’altra all’altro e una porta è riservata alla connessione col secondo switch (la cosiddetta porta di trunking, essa permette di interconnettere due VLAN). Per riconoscere la VLAN a cui appartiene l’host destinatario, si utilizza un formato esteso del frame Ethernet, esso aggiunge il campo VLAN di 4 byte che trasporta la marca della LAN, essa viene aggiunta in trasmissione e tolta in ricezione. Questo campo è diviso in TPID (Tag Protocol Identifier) e TCI (Tag Control Information) contenenti rispettivamente il campo di identificazione della VLAN e il campo di priorità. A livello di protocollo non è presente crittografia, di conseguenza non è robusta ad attacchi di livello fisico. La limitazione delle collisioni ha permesso l’utilizzo di altri mezzi trasmissivi (quali le fibre ottiche) oltre a Ethernet, permettendo grosse possibilità nella progettazione. Consideriamo ora la rete qua a fianco, in questo caso la configurazione permette una configurazione a due a due tra gli host: H1 può infatti comunicare solo con H2 e H3 fa lo stesso con H4 dato che hanno il terzo numero dell’indirizzo IP in comune. In questo modo però un host può cambiare il proprio indirizzo IP e quindi camuffarsi come host dell’altra rete, in più il traffico broadcast è lo stesso e non viene isolato tra le due “sottoreti”. Tutto questo serve a far capire che questa soluzione banale non è niente in confronto alle VLAN vere e proprie, le quali risolvono tutti i problemi trattati in precedenza.

#### Protocollo PPP (Point to Point Protocol)

PPP è un protocollo che permette la comunicazione tra due nodi della rete, la differenza principale col broadcast è che quest’ultimo permette la condivisione del canale, infatti a livello di dorsale i mezzi non sono condivisi, permettendo di fornire un’unica interfaccia a livello di rete. PPP offre un meccanismo di framing ma non di controllo di flusso e di rilevamento errori, questo perchè basta un minimo di disallineamento per causare errori (specie nel cuore della rete), per questo motivo i router rimangono sempre in ascolto. Nel frame vi sono 2 ottetti di bit che delimitano i bit di controllo, il protocollo di rete utilizzato, informative e dei bit di check che contengono un secondo delimitatore indicante la fine. Nel campo informazioni il pacchetto non può essere direttamente copiato dato che il suo payload può causare ambiguità nel frame confondendolo con la fine, tuttavia vi è un trucco per impedirlo: si ricodifica l’informazione oppure si utilizza il byte stuffing, un’imbottitura extra di byte ripetuta due volte nel campo informazioni, essa viene tolta in ricezione e il pacchetto viene passato al livello superiore (quando l’imbottitura non è presente, si tratta di un frame di controllo). *In questo modo è possibile dividere un frame dall’altro senza l’utilizzo di segnali speciali.* Il protocollo PPP è gestito da un automa a stati finiti che, a seconda dello stato in cui si trova, permette l’invio/ricezione dei frame:

* Si stabilisce il collegamento dei nodi, da essa è possibile passare alla configurazione a livello di rete (con o senza autenticazione);
* La configurazione di rete non è necessaria dato che è possibile farla in altri livelli con altri meccanismi, la connessione diventa così aperta;
* La terminazione avviene quando non vi sono dati da trasmettere e di conseguenza muore.

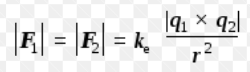
##### 

# Livello Fisico

#### Teoria dei segnali

Il segnale del codice di Manchester visto in precedenza è un’onda quadra in cui il fronte di salita indica lo 0 e 1 quello di discesa, tuttavia un segnale del genere ha poca vita su un mezzo trasmissivo ed è per questo motivo che si utilizzano le onde sinusoidali. Per introdurre i segnali, serve il concetto di campo magnetico, esso è un campo in cui scorrono e si propagano cariche elettriche con le seguenti grandezze fisiche:

* Campo elettrico Statico: le cariche elettriche opposte si attraggono per via della legge di Coulomb;



* Intensità: numero di cariche per unità di tempo che passa per un conduttore.

#### Serie di Fourier

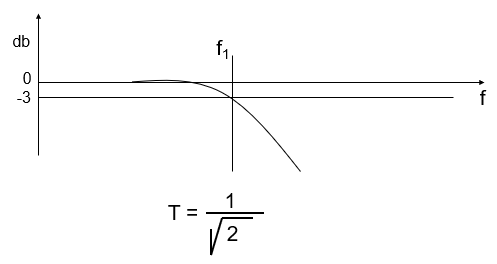
Il segnale sinusoidale può essere visto come un segnale “naturale” dato che si ricava estendendo un cerchio sull’asse del tempo, esso si ricava con la seguente formula:

in cui indica la velocità circolare in rad/s mentre indica l’angolo di sfasamento iniziale. Fourier ha dimostrato che una qualunque curva periodica è rappresentabile come una somma pesata di sinusoidi, il risultato matematico permette così di comprendere le proprietà della sinusoide quando si propaga. Per essere perfettamente quadro, un segnale avrebbe bisogno di infinite armoniche di frequenza, tuttavia gli ostacoli tagliano facilmente le frequenze alte fungendo da filtro, in più per trasmetterle a banda piena servirebbe il doppio del rate richiesto. Il guadagno permette l’amplificazione del segnale, esso può essere in potenza () oppure in tensione (), entrambi si misurano in decibel (db) e ne indicano la quantità. 

#### Filtri

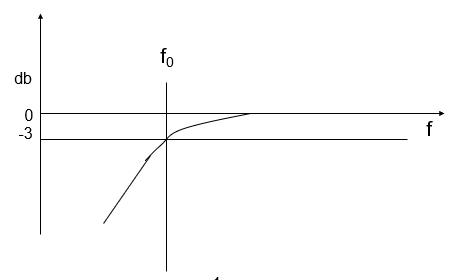
I filtri permettono di modificare il segnale tagliando le frequenze che non si vogliono considerare, un cavo può essere considerato come un filtro di questo tipo dato che ha una resistenza interna che frena gli elettroni e una capacità che indica che il mezzo “si riempie” prima di trasmettere. Il punto in cui il filtro inizia a tagliare le frequenze è detto frequenza di taglio, tutte le frequenze superiori (Passa-Basso) o inferiori (Passa-Alto) vengono tagliate, in quel punto in segnale ha già ricevuto un’attenuazione di 3 db, quindi 1000 volte meno rispetto all’originale.

##### Filtro Passa-Basso

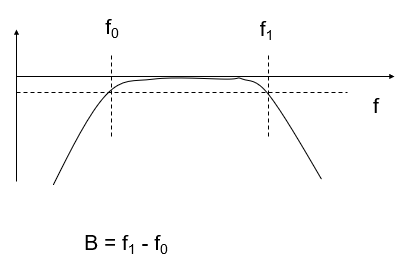
Il filtro passa-basso permette di tagliare tutte le frequenze al di sopra della frequenza di taglio, quando la frequenza è alta la velocità angolare aumenta e il tasso di trasmissione diminuisce, con quelle basse avviene il contrario:

)

##### Filtro Passa-Alto

Il filtro passa-alto è il complementare del passa-basso: tutte le frequenze più basse della frequenza di taglio vengono tagliate, in questo caso le frequenze basse fanno tendere a 0 il al numeratore, quest’ultimo fa tendere a 0 anche T *perchè è di ordine superiore rispetto a quello del denominatore*:

#### Filtro Passa-banda

Unendo un filtro passa-basso e uno passa-alto, si ottiene un passa-banda (o escludi-banda), esso taglia tutte le frequenze più alte della frequenza del passa-basso e più basse di quella del passa-alto. La banda di frequenza passante è quindi data dalla sottrazione delle due frequenze di taglio:

In questo modo è possibile dividere i segnali in bande di frequenza e trasmetterle con lo stesso mezzo utilizzando il multiplexing.

Come già detto prima, un cavo è implicitamente un filtro passa-basso a causa degli attriti interni e alla propria lunghezza/diametro, quindi c’è bisogno di un’amplificazione ogni tanto. In più questo fenomeno dipende anche dalla qualità del cavo, infatti le possibili discontinuità su esso riflettono una parte del segnale, non è quindi consigliato eseguire giunzioni a caso.

#### Frequency Division Multiplexing (FDM)

FDM è una tecnica che permette di far passare più segnali su un unico cavo senza mescolarli, ognuno di essi ha una banda di frequenza assegnata, quindi per ricavare il segnale originale in ricezione occorre filtrare il tutto. Un particolarità del cavo è che, quando passa della corrente all’interno, genera un campo magnetico, è quindi possibile sfruttarla utilizzando le antenne, esse infatti si comportano esattamente come i cavi, tuttavia il canale wireless è condiviso e quindi si utilizza la TDM per scambiarselo. Tra la banda trasmissiva e quella di frequenza vi è un legame: per la trasmissione di un bit su un canale bisogna generare un’onda quadra con una data frequenza, essa deve essere almeno il doppio del bitrate per una trasmissione ottimale, questo perchè un segnale per essere riconoscibile deve essere formato da almeno due forme d’onda, una avente una data frequenza e l’altra con frequenza doppia.

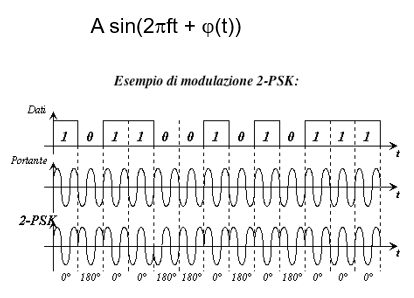
Dato che il codice di Manchester trasmette a banda piene, FDM non funzionerebbe con esso e quindi si utilizzano altre tecniche. Secondo il teorema di Nyquist, la capacità che può trasmettere un canale è due volte la banda di frequenza per il logaritmo del numero di livelli del segnale:

La banda indica che le frequenze che passano nel mezzo devono limitare la frequenza di modifica dei livelli, infatti una troppo alta e con banda piccola non permette il riconoscimento esatto del segnale trasmesso, quindi si può utilizzare un limite superiore a cui attenersi. Un cosa da considerare è anche il rumore di fondo dovuto all’agitazione termica degli atomi, nella pratica si utilizza quindi il teorema di Shannon, esso permette di calcolare la capacità di una canale: l’ampiezza del segnale è data dalla banda per il logaritmo di SNR+1, dove SNR è il rapporto in db tra la potenza del segnale e quella del rumore, *necessario per risolvere i problemi di Nyquist relativi al rumore di fondo*:

#### Modulazione di ampiezza (AM)

AM modula il segnale distorcendone l’ampiezza utilizzando un ulteriore segnale detto portante, avente frequenza maggiore, il risultato ottenuto è un segnale con la frequenza della portante e l’ampiezza della modulante. Rispetto a Manchester però non vi è molta differenza, aumentando i livello però si ottengono risultati migliori.

#### Modulazione di frequenza (FM)

FM modula il segnale variando la frequenza della portante in base all’ampiezza della modulante, il risultato è un segnale con l’ampiezza della portante con sinusoidi annidate a intervalli regolari, essa permette di rimanere stabile anche a lunghe distanze.

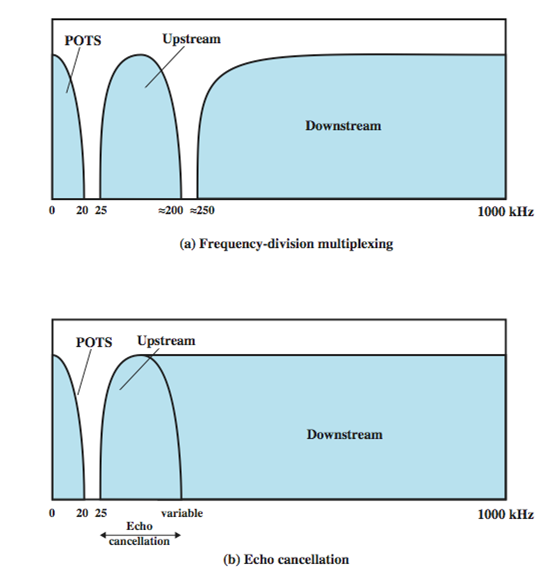
#### Modulazione di fase (PM)

PM modula il segnale sfasando in modo riconoscibile la portante, essa è tuttavia sconveniente quando vi sono troppi angoli.

#### Modulazione multipla

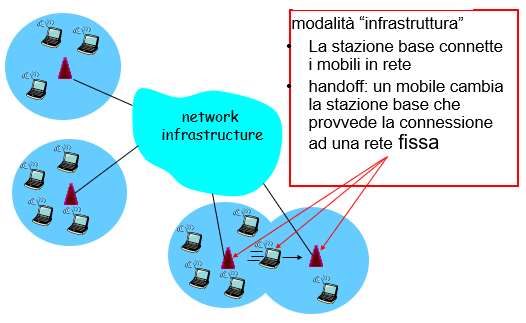
La modulazione multipla utilizza una PM a 4 o 8 angoli di sfasamento e infine modula il segnale in AM o FM, essa permette l’invio di più bit alla volta ed è utilizzata nella connessione DSL aumentandone la banda trasmissiva.

### ADSL

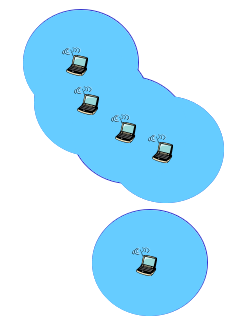
ADSL permette l’accesso a Internet attraverso la rete telefonica e utilizzando una banda di frequenza tra 0 e 1 MHz. Con questo sistema il segnale viene trasmesso pulito perchè, con molto rumore, la trasmissione risulterebbe nulla. ADSL è suddivisa in tre bande: la prima è piccola e dedicata alla voce (il cosiddetto POTS), una più grande per l’upload dei dati e il resto per il download, *per sfruttarle al meglio si suddividono in ulteriori sottobande da 8kHz*, ognuna codificata in PSK, l’infittimento delle sottobande permette infatti di aumentare la banda trasmissiva. L’utilizzo di canali separati nell’ADSL permette di gestire il bit rate a seconda di dove si trova la sottobanda interessata, i dati infatti vengono distribuiti tra le bande e ricomposti in ricezione. L’ADSL ha permesso la distribuzione di dati fino all’arrivo della fibra ottica, essa infatti può gestire bande enormi attraverso FDM o TDM ed è possibile fare giunzioni in ogni punto permettendo così una rapida diffusione, l’unico problema sono gli scavi (anche se si tende a utilizzare quelli già presenti). Oltre alla fibra ottica, vi sono i ponti radio terrestri e satellitari, i primi si sono evoluti diventando reti 3G e 4G, i secondi invece possono essere:

* geostazionari: seguono la rotazione terrestre, essi funzionano ovunque ma sono molto lenti;
* a orbita bassa: orbitano in modo indipendente e distribuiscono la connessione in date zone in un dato momento;

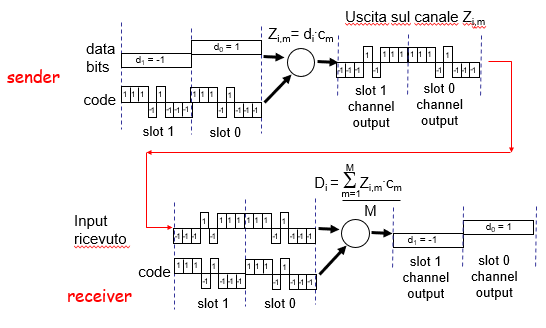
### Reti Wireless

Le reti wireless esistono grazie al protocollo 802.11, ovvero il Wi-fi, esso si è evoluto molto velocemente ai giorni nostri e permette lo scambio di dati senza l’ausilio di cavi. La banda di frequenza del Wi-fi è a 2,4 GHz, tuttavia i suoi limiti ne permettono il funzionamento solo negli spazi chiusi, *per questo motivo si è passati ai 5 GH*z che, oltre ad avere un corretto funzionamento negli spazi aperti, incrementa anche la capacità trasmissiva *dal momento che le bande sono più fitte*. 2,4 e 5, essendo due bande molto separate, possono essere gestite in contemporanea e senza disturbi. Un segnale wireless si attenua molto più velocemente rispetto alla sua controparte su cavo, l’intensità infatti diminuisce quadraticamente all’aumentare della distanza dalla fonte. Le reti wireless possono essere di due tipi basati su tecnologie differenti:

* La rete cellulare, che permette lo spostamento nel territorio senza limiti fisici;
* La rete locale, basata su 802.11.

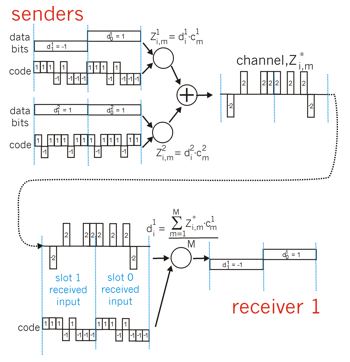
Entrambe devono gestire l’utente mobile e la sua mobilità, può capitare infatti che esso si sposti da una rete all’altra cambiando di conseguenza l’IP e tutto il resto. Se il cambiamento è a livello Data link non vi è alcun problema, dato che l’indirizzo IP non cambia. Tuttavia in questo tipo di reti è difficile rilevare le collisioni, l’access point in più adatta la sua frequenza alle condizioni della rete e questo ne penalizza le performance, si può considerare come un hub senza fili. L’access point è un componente di rete il cui compito è quello di coprire una data zona o una parte di essa permettendo l’accesso a tutti i dispositivi all’interno e trasportando i dati in rete utilizzando i cavi. Una rete only-wireless è possibile ma non permette il collegamento a Internet, quindi è utilizzata in contesti ristretti. Queste tecnologie possono essere utilizzate per connettere a Internet più zone attraverso i ponti radio, per quanto riguarda le campagne si utilizza la tecnologia WiMax, essa divide la banda in sottobande modulate in differenti modi. Per risolvere lo spostamento degli utenti da un punto di accesso ad un altro, si può chiedere l’accesso trasparente a tutti i punti, tuttavia l’approccio utilizzato è molto differente dal momento che è necessario gestire i socket anche quando l’indirizzo IP cambia. La modalità ad-hoc è la comunicazione diretta tra i nodi presenti nella propria area di copertura, essa però è molto instabile a causa delle interferenze e alla geografia del territorio, in particolare quando sono presenti materiali metallici oppure cose ricche d’acqua.

#### Caratteristiche

I protocolli di comunicazioni wireless sono fatti in un certo modo perchè devono ovviare a varie problematiche: a differenza di quelli cablati, infatti, i collegamenti senza fili hanno l’intensità del segnale minore e che va a perdersi lungo il cammino più rapidamente, specie quando sono presenti ostacoli, in più un segnale radio può percorrere differenti percorsi a causa delle riflessioni. 

Esempio: supponiamo che vi siano tre host A, B e C, l’host B può quindi sentire A e C ma questi ultimi non possono comunicare a causa della distanza, infatti se inviano entrambi dei dati a B, essi non si accorgono della collisione e quindi continuano a trasmettere, *per questo motivo non è possibile utilizzare CSMA/CD*. Per risolvere questo problema si utilizza un protocollo che non rileva le collisioni e conferma la trasmissione con degli ACK.

#### Code Division Multiple Access (CDMA)

CDMA è un protocollo che permette la condivisione di un segnale senza l’utilizzo di FDM o TDM. Tutti i segnali utilizzano la stessa banda e, per essere riconosciuti, vengono utilizzati i codici di chipping, dei codici univoci per ogni utente che permettono il riconoscimento del segnale *senza mescolarsi con gli altri e con la minima differenza*. La codifica del segnale avviene effettuando l’AND logico con il codice di chipping, la sua decodifica attraverso la somma dei prodotti tra il bit del segnale e quello del codice. Il vantaggio è che la comunicazione, dato che due host generano due differenti forme di onda per il riconoscimento, anche in caso di interferenza tra essi è possibile ricostruire il segnale originale, tuttavia più sono le comunicazioni simultanee, più i codici di chipping sono “vicini” e quindi è più difficile ricostruire il segnale a causa delle maggiori interferenze, serve quindi un certo arbitraggio, in più per utilizzarlo la banda deve essere almeno il doppio del codice di chipping. 

#### Codifica dei segnali wireless a livello fisico

Il Wi-fi a livello fisico codifica i segnali utilizzando appositi canali ricavati dalle bande di frequenza (il numero di canali è 11 o 14 a seconda della versione), esse sono sovrapposte *in modo che la scelta dei canali da utilizzare si disgiunta*, se si sceglie il canale 1, ad esempio, *gli altri access point possono scegliere i canali 3 e 5 per evitare di interferire tra loro*. Dopo che l’access point ha scelto il canale, gli host che vogliono connettersi devono adeguarsi a esso. L’access point sceglie solitamente il canale assegnato dall’amministratore oppure quello con meno interferenze, tuttavia possono verificarsi a causa di un altro access point che ha scelto lo stesso. La codifica utilizzata è OFDM, in cui:

* il canale scelto viene diviso in sottobande;
* si convertono i bit in sequenze di sottobit utilizzando i codici di chipping e si trasmettono in sottobande differenti modulandoli in fase e ampiezza.

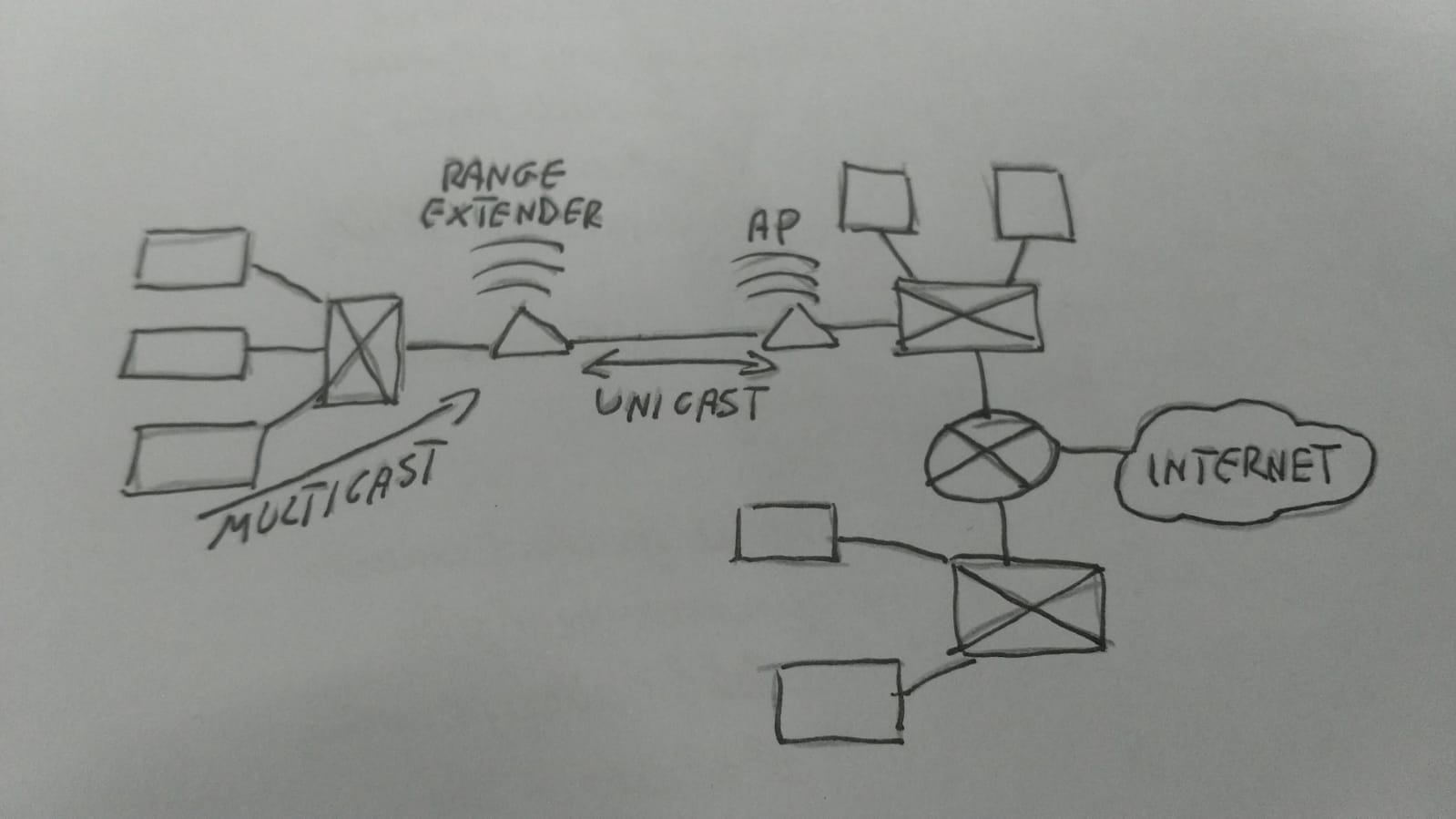
Il codice di chipping viene comunicato in modo semplice: dal momento che si trasmettono solo 0 e 1, si guarda la prima parte della trasmissione del bit, qui viene capito e utilizzato in ricezione.

La frequenza dell’access point viene solitamente assegnata dall’amministratore di rete oppure autoassegnandosi quella con meno interferenze, in quest’ultimo caso può capitare che due access point vicini scelgano la stessa creando così interferenze. Quando un host vuole associarsi a una rete, esso scandisce i canali ascoltando il beacon frame di ognuno, esso contiene l’SSID e l’indirizzo MAC dell’access point associato, ne sceglie uno e da qui inizia l’associazione per procurarsi un indirizzo IP della relativa sottorete. Per evitare malintenzionati, l’SSID viene nascosto. Per l’accesso multiplo si utilizza CSMA per evitare le collisioni, esso però funziona in modo differente da Ethernet, infatti:

* Non vengono rilevate le collisioni dato che *non è sempre possibile a causa della distanza degli host*, i frame vengono quindi inviati senza interruzioni fino al completamento anche se corrotto;
* Vengono utilizzati degli ACK perchè, *dato che si è senza Collision Detection, non si può sapere se avviene una collisione o no*, essi vengono inviati quando ricevuti correttamente.

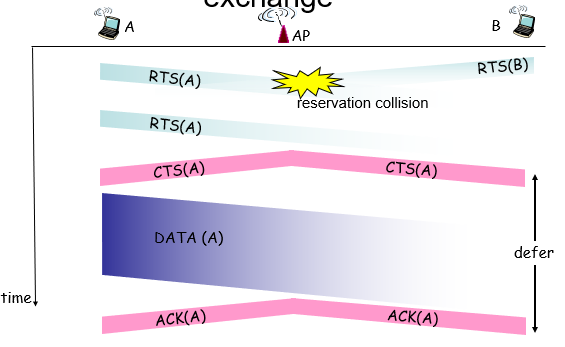
Il meccanismo utilizzato da CSMA e il Collision Avoidance (CA) e permette di evitare le collisioni in modo probabilistico.

#### CSMA/CA

L’idea di CSMA/CA è evitare le collisioni sfruttando gli intervalli tra un frame e l’altro, l’host infatti rimane in ascolto sul canale e, se è libero, trasmette e attende l’ACK, altrimenti sceglie un tempo di attesa casuale, fa partire un timer e trasmette al suo scadere. In questo modo però si rischia la “serie sfortunata”, infatti ritardi troppo lunghi potrebbero impattare ai livelli superiori, il meccanismo è stato quindi modificato per garantire una certa qualità del servizio. I ritardi tra un frame e l’altro sono DIFS e SIFS, il primo crea un intervallo minimo in cui un host in modo che i frame siano chiaramente separati, il secondo ha lo stesso scopo del primo ma è più breve, *permettendo così all’ACK di essere trasmesso prima che un altro frame inizi la trasmissione senza evitare collisioni*. Nel broadcasting non vengono utilizzati gli ACK dato che si utilizza l’indirizzo del broadcast per farlo, quindi non si riuscirebbe a capire chi ha ricevuto correttamente il frame e chi no, oltre al fatto che un client che non risponde penalizzerebbe l’intera rete *per via dell’adattamento del wireless alle condizioni della rete stessa*. Gli ACK non vengono neanche utilizzati nel multicast dato che il Wi-fi non risolve i suoi problemi, tuttavia è ancora possibile utilizzarli se si ha una lista di device a cui si è inviato il frame. In più gli host che non rispondono penalizzerebbero la rete, questo insieme ad altri malfunzionamenti ha portato a preferire una serie di unicast dall’access point all’host, è comunque possibile separare i frame dagli altri inserendo 01 nell’ottetto più a sinistra dell’indirizzo MAC. 

Problema: consideriamo la sottorete a sinistra, in questo caso gli host ricevono due volte lo stesso ACK dato che il frame multicast passa sia per il range extender, sia dall’access point, il problema della conversione multicast-unicast non completamente risolto proprio per questo motivo.

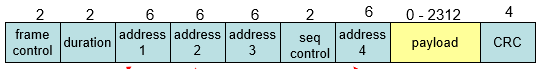
### Request/Clear To Send (RTS e CTS)

Un modo per evitare le collisioni è l’utilizzo dello schema RTS/CTS, l’idea alla base è riservare il canale piuttosto che fare l’accesso casuale per i frame lunghi, esso funziona nel seguente modo:

* Quando un host ha dati da inviare, incomincia inviando un frame corto detto RTS contenente un codice, essi possono collidere ma la loro lunghezza rende quasi inevitabile ciò;
* Il ricevente, ricevuto RTS, invia in broadcast il frame CTS, esso permette all’host trasmettitore di trasmettere i dati sul canale riservato mentre tutti gli altri lo cambiano e non interferiscono, *in questo modo si evitano tutte le collisioni nei dati.*

L’access point è trasparente agli host, esso bufferizza e ripete i dati come uno switch e, come un hub, non può mantenere più connessioni distinte. Inoltre l’access point non tiene tutti i messaggi ricevuti in attesa di ACK, *se così fosse sarebbe un guaio dato che potrebbe riempire tutta la memoria, il compito di inviare l’ACK spetta quindi al ricevente.*

### Indirizzamento

Il frame 802.11 è molto differente da quello Ethernet, infatti vi sono quattro indirizzi MAC, ovvero sorgente, destinazione, access point e gestione delle reti ad-hoc (quando utilizzate), quest’ultimo è utile quando sono presenti più access point nella stessa rete e si vuole essere sicuri di prendere quello giusto, *senza infatti si rischia di inviare due volte lo stesso messaggio*. Quando il frame deve andare in Internet o in una parte cablata della rete, l’access point lo riceve, lo converte in formato 802.3 e lo inoltra, in ricezione invece avviene il contrario. Può capitare che, passando da access point all’altro, lo switch non aggiorni la sua tabella e quindi continua a inviare frame a quello vecchio, questo problema è però stato risolto negli switch più moderni. Gli altri campi del frame 802.11 servono per i dati o la frammentazione. L’access point è quindi un componente di rete trasparente al router e allo switch dato che si preoccupa di convertire il frame da 802.3 a 802.11 e viceversa senza però cambiare l’indirizzo MAC, *si ha quindi l’illusione di un invio diretto del frame all’interno di una stessa sottorete*. Per quanto riguarda l’associazione, l’access point invia periodicamente dei beacon frame contenente le informazioni per associarsi, l’host richiedente esegue l’autenticazione e prende un indirizzo IP attraverso il DHCP, entrando quindi nella LAN.

### Adattamento del bit rate

*Per evitare che la comunicazioni peggiori a causa di un basso rapporto segnale/rumore*, bisogna andare a calare il bit rate per tutti gli host connessi a un dato access point, tuttavia per questo motivo la rete wireless può degradare andando pianissimo. Questa abilità *non è sempre un vantaggio dato che incide su tutti gli host*.

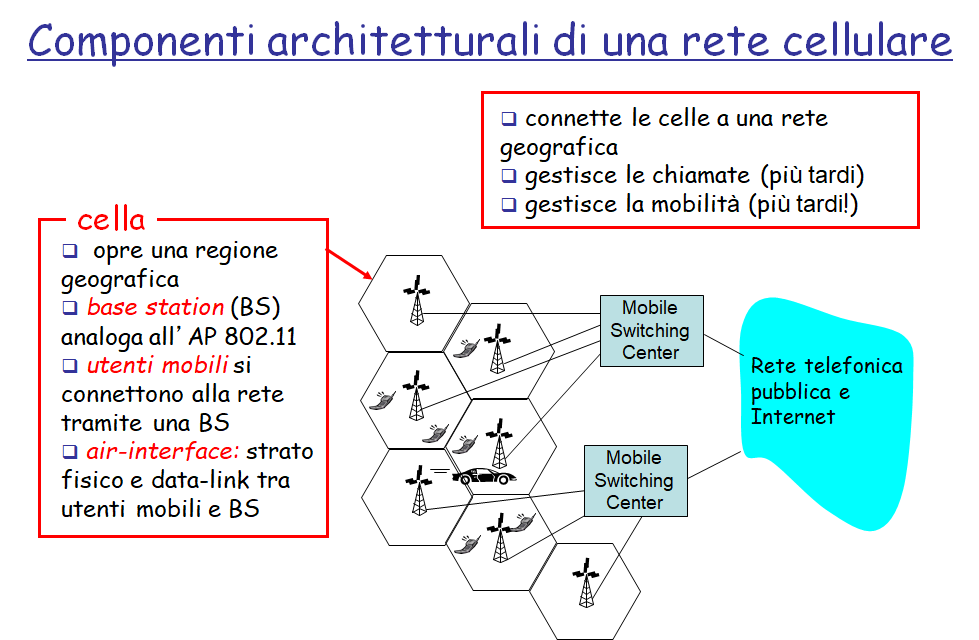
### Risparmio energetico

Dal momento che il Wi-fi è sempre attivo, esso consuma corrente, quindi si cerca di avere sistemi a basso consumo da applicare insieme ad appositi protocolli per il risparmio energetico: un host si mette a dormire fino allo scadere di un timer, fino a quel momento non riceverà frame, una strategia è quindi svegliarsi prima dell’invio del prossimo beacon frame e di conseguenza connettersi. Tutto questo avviene grazie a un frame inviato dall’host all’access point, al risveglio parlerà con quest’ultimo per verificare se c’è qualcosa che gli interessa e, se è così, rimane sveglio, altrimenti torna a dormire.

### Rete Cellulare

La telefonia si è evoluta in modo indipendente a Internet, si è deciso così di utilizzarla prima per la comunicazione su cavo tra computer, poi tra persone e infine si è passati all’utilizzo del wi-fi. I primi telefoni sfruttavano la frequenza di 4kHz (56 kb gestibili), l’equivalente della voce umana per la codifica del segnale, per un suono udibile infatti si taglia il segnale intorno a questa banda. A causa della conversione analogico-digitale, si è passati ai 60 kHz, essa permette di mantenere una certa qualità nella voce e passando alla gestione di 64kb di dati. Grazie all’accorpamento di più canali, la telefonia si è evoluta aumentano il bit rate fino a quando la visione della comunicazione non si è rovesciata: prima si crea qualcosa in grado di far passare i dati, poi si pensa a far passare la voce. Nonostante ATM garantisca una qualità del servizio migliore rispetto ad IP, *l’esigenza era espandere Internet in modo omogeneo e quindi venne favorita quest’ultima*, dato che meno costosa e più facile da implementare. La telefonia wi-fi è nata coi cordless TAP analogici che funzionavano attraverso onde radio, il suo successore, GSM, implementa la conversione A/D nella telefonia *permettendo così anche il passaggio dei dati*.

#### Architettura delle rete cellulare

L’architettura della telefonia cellulare è gestita da molteplici celle con topologia esagonale, *la conformazione del terreno però non lo permette e quindi ci si accontenta*, esse rappresentano il punto di accesso al servizio. Tutte le celle confluiscono in un Mobile Switching Center (MSC), essi gestiscono le chiamate mantenendo continuità durante gli spostamenti dedicando al telefono un canale, esso cambierà spostandosi di cella in cella. Dal momento che la telefonata è nella rete pubblica, il telefono deve esse quindi assegnato a una cella che, finchè il destinatario si trova nello stesso campo, va bene, altrimenti bisogna cercare di connettersi con esso. Al cambiare di una MSC, viene effettuato del routing tra quella vecchia e quella nuova mantenendo un canale aperto tra esse, tutt’oggi però si tende a mantenere tutto in una singola MSC *per evitare troppi carichi*. La cella permette l’apertura di una regione geografica ed è formata da una base station (simile a un access point), dagli utenti connessi e dall’air-interface, uno strato a metà strada tra il livello fisico e data link che connette le prime due parti. *Quando la base station di una cella sente che il segnale sta diminuendo*, fa partire il cosiddetto hand over, esso consiste nell’invio di una lista di possibili celle a cui il cellulare dell’utente può connettersi, MSC in seguito decide a quale attaccarlo, lo avverte dell’assegnazione e infine chiude il collegamento con la vecchia base station.

#### Gestione della comunicazione

La comunicazione telefonica viene gestita combinando FDM e TDM, questo permette l’utilizzo di più slot differenti per una data banda di frequenza, per quanto riguarda l’accesso multiplo si utilizza invece CDMA. A partire dai sistemi 2G ai 2.5G si è passati a CDMA-2000, col 3G si utilizza insieme a UMTS e infine a LTE (Long Term Evolution) col 4G, quest’ultimo è nato col 3.5G e permette di ritrattare gli strati alti mantenendo invece quelli bassi, *permettendo così l’adattamento dell’architettura al progresso*.

#### LTE: Principii base

La tecnologia LTE è basata su tre protocolli:

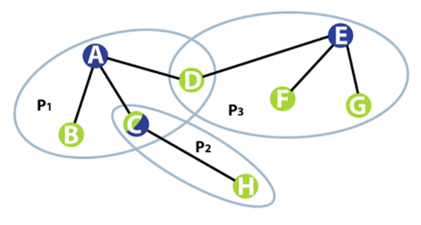
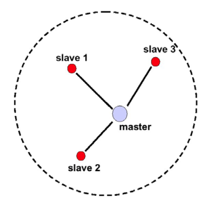
* OFDM: permette di dividere il canale in sottocanali, codificare il segnale coi codici di chipping e inviare ogni bit in un sottocanale differente, permettendo di evitare riflessione/interferenze e un alto grado di resilienza.
* MIMO: vengono sfruttate le riflessioni multiple aumentando il bit rate e usandolo come ridondanza;
* SAE: l’architettura è aperta e permette la sua evoluzione.

LTE permette anche la gestione fisica delle risorse, l’allocazione del canale infatti avviene dinamicamente dividendo slot e sottofrequenze, l’idea è dare all’utente un gruppo di queste per avere qualità del servizio, più è grande questo insieme, più può trasmettere, il resto viene lasciato agli altri. *In questo modo è possibile sfruttare al meglio la banda a disposizione.*

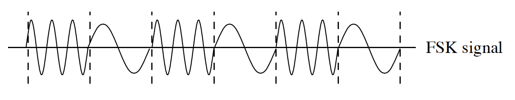
### Bluetooth (802.15.1)

Bluetooth è uno standard di comunicazione a basso consumo sviluppato da Ericcson nel 1996, esso è utilizzato nella ricetrasmissione dei dati a corto raggio e con transfer rate bassi, lo scopo infatti è ottenere un trasferimento semplice, facile da implementare e di tipo master-slave, quest’ultima qualità permette di utilizzarlo in reti piccole e semplici.

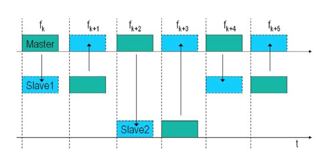
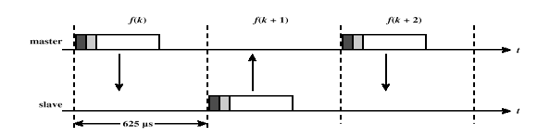
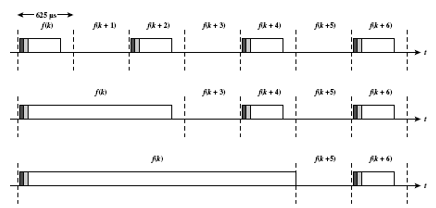
#### Piconet

Una rete di dispositivi Bluetooth è detta piconet, essa è una WPAN (Wireless Personal Area Network) in cui si utilizza l’approccio master-slave *dato che, essendo una rete ad-hoc, non è necessaria un’infrastruttura per interconnettere i device*. Il dispositivo master ha il compito di regolare la piconet, tutti gli altri sono dispositivi slave, essi possono essere al massimo 7 non possono comunicare direttamente tra loro, per farlo devono utilizzare il master come intermediario, oltre a essi vi sono un massimo di 255 dispositivi in modalità Park: dispositivi sincronizzati al master ma che non partecipano attivamente alla comunicazione, possono soltanto ricevere messaggi broadcast. L’entrata e l’uscita di un dispositivo da una piconet avvengono in modo dinamico, questo *perché i dispositivi non sono vincolati da cavi fisici e quindi si possono (s)connettere quando si avvicinano o allontanano dal master.* L’insieme di più piconet forma una scatternet, in essa alcuni dispositivi master sono slave di altre piconet oppure un dispositivo slave è tale in più di esse. Una scatternet permette di aumentare il numero di dispositivi attivi in una connessione, andando oltre i soliti 8 imposti dalla piconet standard.

#### Comunicazione

La comunicazione avviene a 2.45 GHz utilizzando la modulazione 2-FSK, ovvero una FM che utilizza due frequenze una per 1 e una per lo 0, *per evitare interferenze* si utilizza il Frequency Hopping, esso ha il compito di ripartire nel tempo le trasmissioni sullo spettro di frequenza. La banda viene divisa in 79 canali monitorati da tutti, ogni dispositivo che trasmette ne sceglie una a caso e inizia la comunicazione, *in questo modo è possibile limitare il numero di collisioni dato che vi è una probabilità molto bassa che due dispositivi scelgano lo stesso* canale. Tuttavia un dispositivo cambia spesso il canale (lo fa ogni volta che finisce lo slot, questo cambiamento è infatti segnato da un tempo di silenzio) con cui trasmette e qui entra in gioco il Frequency hopping citato prima:all’inizio la frequenza utilizzata è la stessa per tutti i componenti di una piconet e cambia a seconda di uno schema pseudo-casuale definito dal master (generazione di numeri casuali a partire da uno stesso seme), *lo scopo infatti è minimizzare la probabilità di creare interferenze tra componenti.*

#### Accesso al mezzo

L’accesso al mezzo avviene tramite la tecnica TDM: Il master manda uno slot di interrogazione definendo un timing di 625 us per ogni slot, esso comunica con gli slave attraverso gli slot pari, questi ultimi invece utilizzano quelli dispari e comunicano uno alla volta “per alzata di mano”, *in questo modo si evitano collisioni dato che il master non è in grado di gestire due slave nello stesso istante*. La frequenza utilizzata è gestita dal master e di conseguenza anche il frequency hopping, ogni slave che vuole connettersi dovrà quindi adattare la propria frequenza. I ruoli di ogni dispositivo sono intercambiabili, infatti se un master decade, uno slave lo diventa per sostituirlo, un master può essere slave di un’altra piconet oppure può diventare slave per far diventare master un altro dispositivo. Il multislot è deciso dal master alla trasmissione e permette, come dice il nome, l’utilizzo di più slot per la comunicazione, in tal caso la frequenza di trasmissione non cambia quando essa è iniziata. Il timing è anch’esso regolato dal master in base al clock di ogni device (625/2 us), alla trasmissione di un frame, il master dà in modo implicito la sincronizzazione, gli slave quindi sintonizzano i loro clock sull’inizio del frame stesso.

#### Indirizzamento

L’indirizzamento del Bluetooth è a 48 bit, 24 identificano il costruttore del dispositivo, i restanti invece il dispositivo stesso, l’intero indirizzo è quindi rappresentabile con 12 cifre esadecimali. 

#### Collegamento

Quando un dispositivo vuole connettersi a una rete bluetooth, effettua in ordine le seguenti fasi:

* Inquiry: esplora l’ambiente circostante per verificare se è possibile connettersi a una rete oppure costruirne una (quest’ultima azione è fatta solitamente dal master);
* Paging: una volta scoperti i componenti, si attiva una connessione stabile con essi, essa può essere:
  + Attiva;
  + Sniff: dorme a intervalli regolari per poi svegliarsi e ascoltare il master;
  + Hold: dorme per un periodo dato dal master e poi si sveglia;
  + Park: si addormenta fino a quando il master non decide di svegliarlo, ogni tanto si risincronizza con gli slave e ascolta i messaggi.

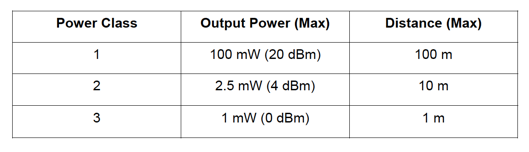
Queste modalità servono per evitare un consumo eccessivo della batteria.

#### Sincronizzazione durante l’enquiring

*Dal momento che le frequenze di una piconet cambiano continuamente*, il master viene incontro ai device (i quali possono ascoltare su una sola frequenza alla volta) inviando la richiesta su più frequenze in sequenza, *in questo modo gli slave prima o poi riceveranno il messaggio e verrà avviata la connessione.*

#### Stack Bluetooth

La pila bluetooth ha più livelli rispetto a ISO-OSI, fino al terzo vi è un livello simile a rete/data link mentre in cima si hanno quelli simili ad applicazione e trasporto.Per aumentare le prestazione il bluetooth si è evoluto, si possono infatti distinguere tre classi:

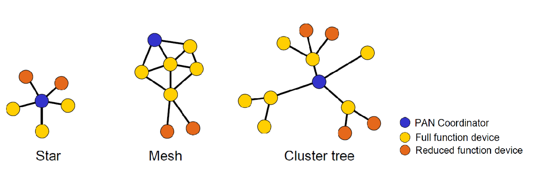


Oltre alle classi, Bluetooth è disponibile in varie versioni:

* 1.2: versione base con rate da 1Mbps fino a 10m di distanza;
* 2.1 + Enanced Data Rate: il range è diminuito a 3m ma vi è più sicurezza;
* 3.0: permette il routing tra piconet e ha un rate di 24 Mbps;
* 4.0: al costo di una distanza (50m) e un bit-rate minore (0,27 Mbps), vengono ottimizzati i consumi.

### ZigBee (802.15.4)

ZigBee è uno standard di comunicazione aperto e a basso consumo molto supportato dalle aziende, tuttavia non è diffuso quanto il bluetooth o altri sistemi. Questo standard utilizza una topologia nata per ottenere connessioni a basso costo, consumo e infrastruttura, utili quindi per l’IoT e la domotica instradando il traffico con lo scopo di fornire servizi. Le tre possibili topologie sono a stella, a cluster e a mesh, quest’ultima ha il vantaggio di essere ridondante nei cammini, scalabile (*basta che un nodo ne incontri un altro che fa da router per poter instradare*) e può coprire aree vaste comunicando a bassa potenza. I nodi della rete possono essere:

* coordinatori: il loro compito è quello di coordinare la PAN;
* nodi a piena funzione (full-function): sono analoghi ai master nel bluetooth, essi possono offrire servizi e fare da router per l’instradamento;
* nodi a funzione ridotta (reduced-function): sono nodi analoghi agli slave che operano sotto il controllo di un full-function, anch’essi possono offrire servizi ma non possono fare da router.

#### Architettura

L’architettura di ZigBee è definita da IEEE ai livelli più bassi, l’applicazione è definita dall’utente mentre i livelli in mezzo sono opera della ZigBee Alliance, il livello trasporto non è presente *dato che il suo funzionamento lo renderebbe superfluo perchè non si pensa a instradare il traffico e non c’è un forte scambio di dati.*

L’indirizzamento prevede un id da 16 bit e uno da 64 bit, il primo identifica un servizio in modo simile alle porte di rete e viene assegnato dinamicamente quando si associa, il secondo invece identifica il dispositivo in modo assoluto. I canali vengono scelti *in modo da minimizzare le interferenze*, è possibile ma non necessario cambiarlo quando la situazione della rete cambia.

#### Accesso al mezzo

L’accesso al mezzo avviene solitamente utilizzando il CSMA/CA cambiando i parametri in modo opportuno, il coordinatore però *in presenza di fasi critiche utilizza una versione slotted dello stesso protocollo*: vengono utilizzati dei time frame più corti (*perchè devono trasportare meno dati*) che costituiscono il superframe(gruppo di trasmissioni delimitate da beacon frame), si disciplina l’immissione di quest’ultimo con slot fissi e, come in slotted ALOHA, si invia il frame competendo con gli altri, vi è comunque un gruppo di slot senza contesa assegnabili *per evitare collisioni*. Il beacon frame permette di dividere il superframe in due parti: un periodo di attività in cui è possibile trasmettere e un altro in cui tutti i dispositivi, controller compreso, possono risparmiare energia andando a riposo. ZigBee utilizza quattro possibili formati di frame:

* Il beacon frame è utilizzato dal coordinatore per “invitare” i device a connettersi;
* Il data frame è utilizzato nel normale trasferimento dati;
* ACK è utilizzato per confermare che un frame inviato è stato ricevuto;
* MAC Command Frame, utilizzato per la gestione delle interfacce MAC.

#### Routing

Dal momento che alcuni nodi della rete possono fare da router, l’instradamento avviene gestendo tutte le possibili strade e la caduta di una o più di esse, per farlo si utilizza il Distance-Vector insieme a un meccanismo ad-hoc che permette di scoprire tutte le possibili strade, inoltre è anche possibile la gestione sia in unicast che in multicast. Un device che vuole scoprire o riparare una strada esegue il Discovery, ovvero che manda in broadcast un messaggio per raggiungere un dato nodo e, se viene ricevuto, si viene a conoscenza della rotta che ha percorso il messaggio per arrivare dato che viene utilizzata per spedire l’ACK.

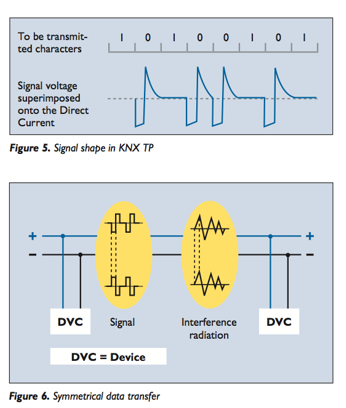
### Bus e reti per la domotica

Le reti utilizzate nell’ambito domotico devono rispettare determinati requisiti: i componenti interni devono essere a basso costo e consumo, il traffico deve occupare la minor banda possibile *(in modo da utilizzare bitrate ridotti*), deve essere semplice da installare e, quando possibile, utilizzare cablaggi già presenti. Un particolare bus concepito appositamente per questo ambito è KNX, esso è uno standard europeo accettato da OSI e supportato da molti produttori, utilizza come mezzo fisico standard il doppino attorcigliato, tuttavia è possibile utilizzare anche il cavo ethernet con IP, il Wi-fi e l’impianto elettrico casalingo. In quest’ultimo caso è possibile connettere moltissimi oggetti, un host può infatti essere un interruttore, un elettrodomestico, eccetera. Il volume dei dati da trasportare è molto piccolo (dell'ordine dei byte) e la velocità deve soddisfare le tempistiche umane, questo ha permesso l’utilizzo di tecnologie a basso costo con bitrate bassi.

#### Architettura

L’architettura utilizzata è solitamente una rete a bus, l’idea infatti e connettere tutti gli host sullo stesso cavo *in modo che con esso possano sia inviare/ricevere dati, sia alimentars*i. L’organizzazione della distribuzione può essere gerarchica, ovvero che vi è una dorsale comune da cui partono quelle secondarie e, da esse, quelle locali che connettono i singoli host.

#### Comunicazione con doppino

La comunicazione tramite doppino è quella più utilizzata quando possibile, essa infatti permette sia l’alimentazione del dispositivo, sia il trasporto dei dati, l’unico problema è la caduta di tensione causata da tensioni molto basse, quindi si preferiscono differenze di potenziale più alte (12, 24 o 48 volt, quest’ultima è utilizzata nel PoE), KNX in questo caso adopera i 24 volt di alimentazione. Sopra la tensione di alimentazione continua, si sovrappone una forma d’onda, il segnale è nella stessa densità su entrambe le linee *onde evitare collisioni*, esse sono infatti dette bilanciate. La codifica avviene in banda base e permette un bit rate di 9600 bps. Le linee bilanciate sono fatte in modo tale da trasportare il segnale in modo simmetrico tra le linee così che i disturbi che percorrono le linee lo fanno nello stesso verso e se si sommano, viene 0 e quindi azzero il disturbo. I bit del segnale vengono ricavati guardando entrambe le linee, precisamente dalla loro differenza di potenziale.

#### Trasmissione

Il formato con cui vengono trasmessi i dati è il telegramma, esso comprende un byte di controllo, 5 di indirizzo, da 1 a 16 di dati e infine uno di checksum, l’accesso è gestito come in Ethernet con la differenza che le collisioni si evitano in modo particolare: il meccanismo di controllo delle linee è fatto in modo tale che chi vi inserisce 0 comanda, permettendo a chi ha inserito 1 di ritirarsi in tempo senza rovinare il pacchetto, tutto questo è possibile grazie alle trasmissioni in bassa frequenza che permettono di vedere bene la transizione. La trasmissione del telegramma parte dal controllo e arriva al checksum, questo rende impossibile la scrittura dello stesso dato dal momento che gli indirizzi sono tutti differenti. Questo meccanismo è trasparente e l’host che continua a trasmettere non si accorge di nulla. Per evitare la starvation causata da uno stesso componente che inserisce sempre 0, quest’ultimo si può ritirare per un dato tempo’ e lascia trasmettere agli altri.

#### Comunicazione con impianto elettrico (Power Line)

KNX permette la comunicazione attraverso la linea elettrica casalinga attraverso l’ausilio di un adattatore, rispetto al doppino si cambia tipo di codifica *dato che la frequenza è differente*, su essa viene poi sommato il segnale modulato in frequenza a 110kHz. La power line funziona come un tunnel: viene incapsulato un frame in un altro frame specifico per queste linee: si hanno infatti 4 bit di sequenza di training (permette la comprensione della frequenza degli 0/1), 2 byte di preambolo da 9 ai 23 per i dati (in cui è contenuto il frame) e l’ultimo *per identificare il componente che lo ha inviato*. Il bit rate è di 1200 bps mentre l’accesso al mezzo è simile a CSMA/CA: si ascolta il mezzo e, se è libero si trasmette, altrimenti, se rileva il preambolo di un altro device, aspetta un intervallo di tempo casuale (scelto tra 7 possibili valori) e dopo ritenta.

#### Comunicazione con radio frequenza e IP

Con la radio frequenza, il segnale viene modulato in frequenza con una portante di 868.3 MHz, arrivando a un rate di 16,384 kbps. Questo tipo di comunicazione implementa il protocollo wireless m-bus, *permettendo in questo modo l’utilizzo di componenti a batteria a basso consumo*, la trasmissione permette l'inserimento di più blocchi di dati nello stesso frame. KNX su IP utilizza invece il protocollo UDP per la trasmissione dei pacchetti, la comunicazione avviene utilizzando un tunnel interfacciandosi col gateway, quest’ultimo riceve i comandi e converte il pacchetto in un telegramma da mandare nella rete domotica.

### Network Address Translation (NAT)

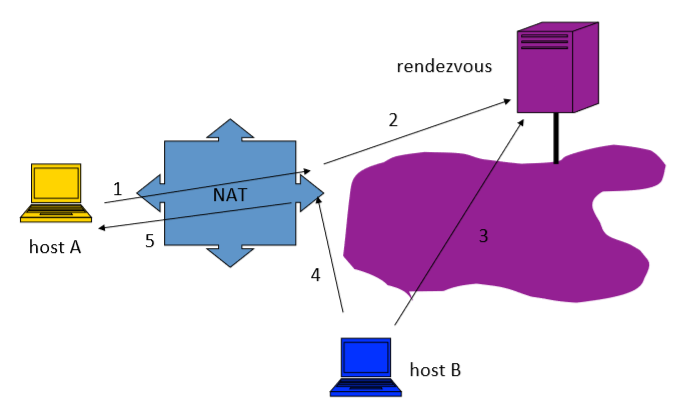
Per far comunicare due oggetti posti in due differenti reti private, si utilizza la NAT, un meccanismo che permette la “conversione” di un IP privato in uno pubblico, essa viene effettuata dal router inserendo nel campo IP sorgente del pacchetto l’indirizzo identificativo della rete (in ricezione, invece, avviene il contrario). La NAT può essere:

* statica: vi sono indirizzi IP associati permanentemente al dispositivo attraverso una tabella che mantiene tutte le corrispondenze IP privato-pubblico, tutto il traffico che arriva dalla coppia esterna viene dirottato su quella interna, l’operazione è però onerosa e quindi viene solamente effettuata negli host che forniscono servizi *dato che un grosso numero di IP NATtati impatterebbe sulle prestazioni*, un’altra soluzione è utilizzare un unico IP pubblico per tutti i device di una sottorete;
* dinamica: la corrispondenza IP privato-pubblico vale soltanto per una sessione, vi sono tuttavia problemi nel caso in cui si deve fornire un servizio, quindi c’è bisogno di meccanismi che informino l’utente per comunicare che l’indirizzo IP è cambiato.

Tutt’oggi la NAT permette, oltre alla mappatura degli indirizzi IP, di mappare anche le porte, *permettendo così di ottenere molte combinazioni utilizzando un singolo IP*. *Per motivi di sicurezza*, la NAT viene utilizzata con delle restrizioni, l’idea infatti è fare quello che si vuole quando si è nella propria rete, all’esterno invece no. Vi sono infatti delle politiche che permettono la comunicazione di host privati sulla rete pubblica, esse sono:

* Cono pieno;
* Cono ristretto;
* Cono ristretto con porta invitata;
* Simmetrico.

#### NAT a cono pieno

La NAT a cono pieno è la politica più libera in cui tutte le richieste da parte di un IP interno vengono mappate sulla stessa coppia IP-porta. Una volta creata la corrispondenza, tutto il traffico ricevuto dall’IP pubblico viene dirottato su quello privato senza controllare l’IP dell’host remoto, rendendolo visibile dall’esterno. L’incontro avviene nel seguente modo: L’host invia una richiesta a un server relay, passando per una NAT, facendo pendere una richiesta, il secondo host contatta lo stesso server, riceve l’IP pubblico e con esso contatta il primo host.

#### NAT a cono ristretto

La NAT a cono ristretto funziona come quella a cono pieno, con la differenza che l’host remoto può inviare pacchetti a quelli interni solo se questi ultimi hanno precedentemente inviato un pacchetto al suo indirizzo. L’incontro infatti è un po’ differente: quando entrambi gli host hanno inviato la richiesta al relay server, quest’ultimo invia l’indirizzo del secondo host al primo, il quale lo contatterà e attenderà una conferma. *Nessun’altro al di fuori degli “invitati” può contattato l’host che ha inviato gli inviti,* tutto questo è possibile perché la NAT svolge una funzione di firewall, infatti se un macchina viene contattata, si controlla se quest’ultima ha già contattato la macchina esterna e, se è così, accetta e recapita il pacchetto verso l’interno, altrimenti no*.*

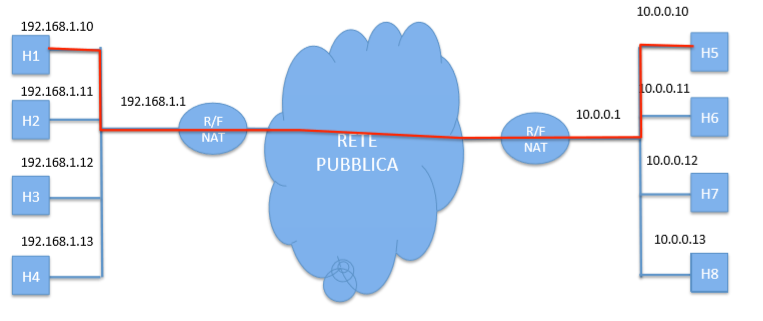
#### NAT a cono ristretto con porta invitata

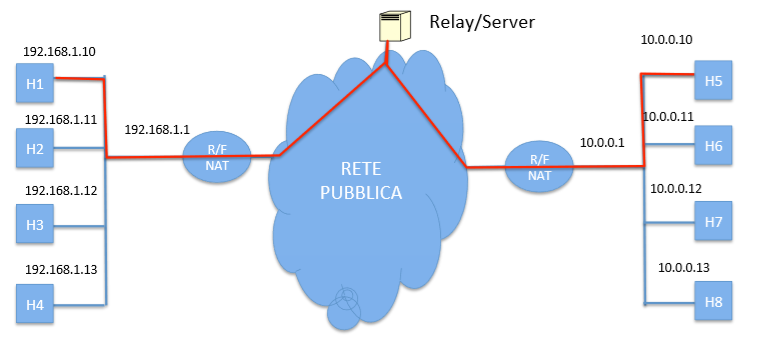
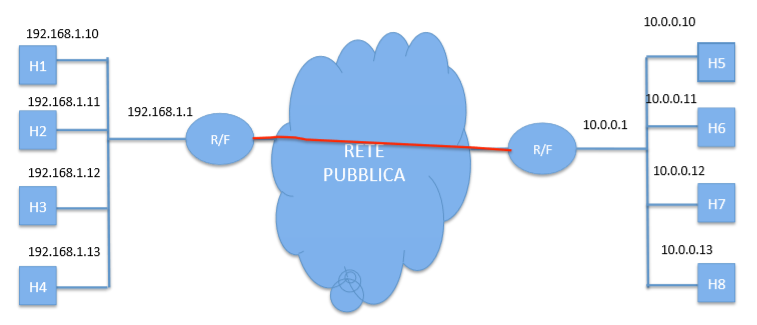
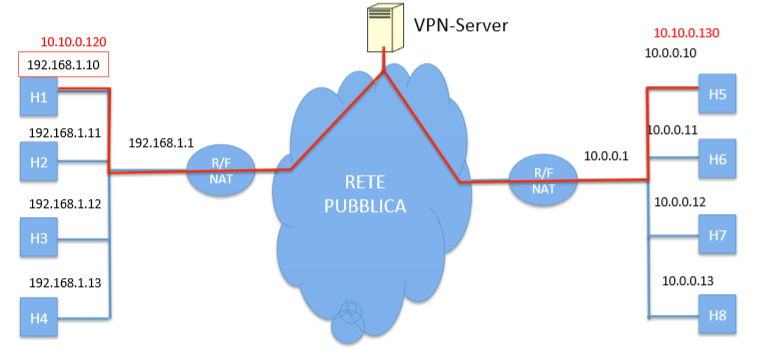
Questo tipo di NAT funziona come la precedente, la restrizione interessa invece le porte: l’host remoto può infatti inviare pacchetti a quelli interni solo a date porte specificate degli host interni, l’obiettivo infatti è quello di *fare entrare solamente gli “invitati” e nessun’altro.* Questi tre meccanismi sfruttano il cosiddetto Punching Hole: all’apertura di una porta viene creato un “buco” di ingresso nella barriera del firewall, permettendo l’aggancio di servizi che stanno nelle reti private.

#### NAT simmetrica

Nella NAT simmetrica, tutte le richieste da una coppia IP-porta interni verso una data destinazione vengono mappate sulla stessa coppia IP-porta esterni, con una destinazione differente verrà utilizzata una seconda coppia IP-porta esterna. Solamente l’host esterno che riceve il pacchetto può inviare agli host interni tramite UDP. Con questa NAT non è possibile la comunicazione diretta, l’unico modo è quindi passare attraverso un server relay, questo *perchè. i firewall sono configurati in modo da non accettare ogni comunicazione che arriva dall’esterno, permettendo solo quelle che partono dall’interno e quindi quelle coi relay server.*

## Comunicazione tra host esterno e interno

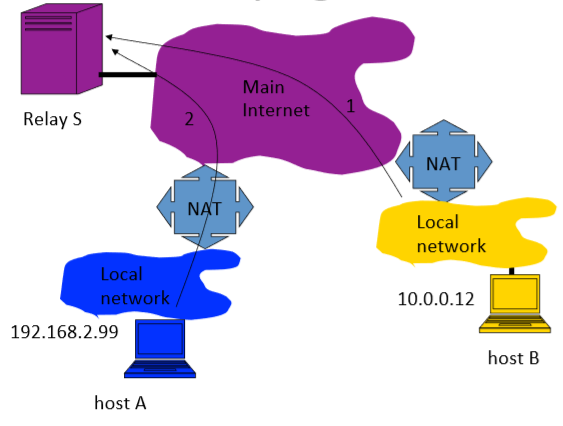
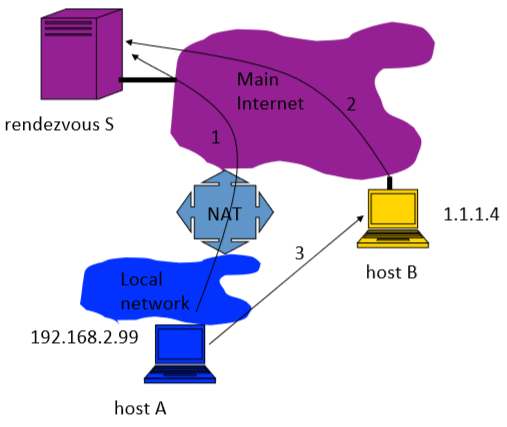
La comunicazione tra l’host esterno e quello interno può avvenire nei seguenti modi:

* Comunicazione diretta: si predispongono i router esponendo le proprie porte all’esterno, è la più comoda ed efficace ma non è possibile quando entrambe le NAT sono simmetriche, in tutti gli altri casi è possibile il collegamento diretto o tramite un server intermedio utilizzando il protocollo STUN come supporto. La comunicazione diretta è possibile con tutte le politiche di NAT tranne quella simmetrica, utilizzando come supporto il protocollo STUN;
* Comunicazione tramite relay: due host si connettono a un server che si occupa dell’instradamento dei dati, tuttavia esso stesso può fare da collo di bottiglia e quindi si utilizza la comunicazione diretta quando possibile. Questo tipo di comunicazione è supportata dal protocollo TURN e si può utilizzare con qualsiasi tipo di NAT, è invece l’ultima spiaggia quando quest’ultima è simmetrica da entrambe le parti;
* Comunicazione site-to-site: viene creato un tunnel diretto tra i due router in cui passano i pacchetti, IPSEC è utilizzato come supporto;
* Comunicazione VPN a stella: si utilizza la rete pubblica per creare un tunnel criptato che interconnette due router, le due sottoreti funzionano come se fossero nello stesso edificio, infatti è abbastanza comune a livello aziendale.

#### NAT multiplo

Oltre al NAT semplice, è anche possibile eseguire quello multiplo, in tal caso ogni host è connesso a più NAT ed è solitamente utilizzata per creare grossi reti LAN, dato che si è in carenza di indirizzi IP. L’idea è utilizzare una NAT principale per uscire e tante NAT private il cui compito è dividere la prima dalle sottoreti, queste ultime sono configurabili per consentire il port-forwarding, quella pubblica invece non è possibile farlo. In questi casi può capitare una situazione in cui una NAT privata e una pubblica non collaborino, non permettendo così l’attraversamento, quindi bisogna trovare delle soluzione utilizzando specifiche tecniche di attraversamento.

#### Tecniche di attraversamento con NAT che non collabora:

* Relaying: viene permette di stabilire una connessione diretta, o il cosiddetto rendezvous, ovvero che vi è un’entità che fa da punto di incontro tra due client quando vogliono comunicare. L’entità in questione è il relay server e ha il compito di:
  + Predisporre i client alla comunicazione diretta stabilendo i contatti;
  + Effettuare la gestione del traffico tra essi.
* Connection reversal: vi è un rendezvous server in cui gli host che vogliono comunicare si incontrano, la connessione avviene utilizzando la politica a cono pieno/ristretto. Uno degli host rimane in attesa su una porta, inviando un pacchetto contenente la coppia IP-porta esterna al server, il secondo invece chiede a quest’ultimo dove sta passando il primo e, ricevuta la coppia, invia un pacchetto e inizia la connessione (questo per la NAT a cono pieno), in quella a cono ristretto invece il server deve inviare l’invito all’host.

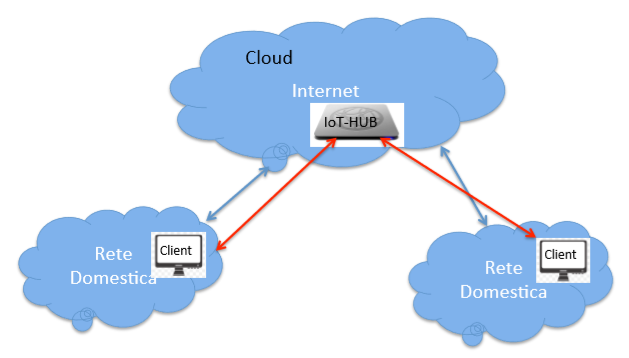
#### Protocollo STUN

STUN è un protocollo che permette l’attivazione di una comunicazione tra due macchine in una rete con NAT non simmetrici, per farlo il protocollo deve scoprire la presenza di una NAT, il suo tipo e i firewall presenti tra rete interna e pubblica. La comunicazione è limitata all’utilizzo di UDP, tuttavia vi è una versione più avanzata che utilizza TCP.

#### Protocollo TURN

TURN è un protocollo che supporta il relaying dietro una NAT sia su TCP che su UDP, insieme a un relay serve a mettere in contatto due applicazioni attraverso i loro socket. Il relay server ha il compito di gestire la comunicazione tra gli host recapitando i dati, esso funge quindi da interfaccia dato che la comunicazione deve sembrare diretta dal punto di vista degli host stessi. Questo protocollo si può considerare come l’ultima spiaggia quando le NAT degli host sono entrambe simmetriche.

### IoT Hub e broker

Dal protocollo STUN si è arrivati alla seguente idea: si mette un sistema rigido a basso livello e se ne propone uno più flessibile ad alto livello. Da questa idea sono nati i message broker e gli IoT Hub, l’applicazione IoT è infatti a basso traffico e i dati sono caratterizzati in tipologie, offrendo quindi qualcosa adatto alla situazione e che offra servizi aggiuntivi. L’IoT Hub è quindi paragonabile a un insieme di caselle postali con cui è possibile interfacciarsi col server, la comunicazione è vista come una comunicazione a memoria comune con la differenza che lo scambio di dati non avviene all’interno di un sistema operativo ma in rete (rimane tuttavia un sistema molto lento rispetto al primo anche con l’avvento delle reti ad alta velocità), il traffico viene gestito da un sistema distribuito di server dato che deve gestire grosse quantità di dati. Quando viene attivato un processo, esso si associa a una delle caselle postali e vengono avvisati tutti quelli che hanno a che fare con lui, questi ultimi invieranno dei messaggi che verranno letti dal processo stesso, tutto questo viene implementato attraverso delle chiamate send() e receive(). Vi è però un problema: in presenza di un sistema avente migliaia di processi che comunicano tra loro, la (dis)assegnazione di una casella è obsoleta e difficile da realizzare. Per questo motivo sono stati introdotti i message broker: una struttura simile a una bacheca in cui tutte le novità vengono segnalate a tutti i device dell’applicazione IoT che, se interessati, possono visualizzare. Il paradigma utilizzato è di tipo publish-subscribe, il messaggio viene quindi pubblicato in un topic della bacheca e tutti quelli registrati a esso riceveranno una notifica. A differenza dell’IoT Hub (molto simile alla comunicazione tra processi), il message broker è più simile alla memoria condivisa, *quindi vengono gestiti tutti gli host allo stesso modo con dovute precauzioni*, esso è preferito al primo perchè la gestione di milioni di utenti durante lo scambio di messaggi è più gestibile (anche se più naturale), tuttavia non è adatto allo streaming *dato che la sua velocità non lo permette*. 

#### Principali classi di applicazione:

* Smart Home: gestione della casa a livello di luci, elettrodomestici e allarme, rendendola intelligente;
* Dispositivi indossabili;
* Connected cars: vetture e trasporti connessi;
* Industrial Internet: monitoraggio e controllo degli apparati industriali;
* Smart Cities: come smart home ma a livello cittadino;
* Agriculture: monitoraggio e controllo delle colture;
* Smart Retail: supporto alla vendita al dettaglio;
* Energy Engagement: gestione della rete di distribuzione;
* Healthcare: monitoraggio e cura dei pazienti distribuiti nel territorio;
* Allevamento di animali.

#### Implementazione di un broker

I message broker vengono implementati utilizzando MQTT (Message Broker Telemetry Transport) (Message Broker Telemetry Transport), un protocollo proposto a IBM tutt’oggi adottato in molte piattaforme tra cui Azure, la sua natura leggera permette infatti di orientarsi facilmente ai message broker, basandosi sul modello publish-subscribe: i processi comunicano tra loro tramite il loro argomento senza esporre loro stessi direttamente, infatti la comunicazione non è diretta ma avviene modificando dati comuni organizzati in una certa gerarchia.

#### Funzionamento

L’architettura di un broker è ad albero le cui foglie rappresentano i dati, essi sono descritti dai nodi ai livelli superiori, *permettendo così di prenderli gestendo l’intero albero oppure dei sottoalberi*.I processi client possono essere:

* Publisher: dopo la connessione al broker, comunicano con un thread locale a esso che gestisce la memoria condivisa, tramite esso è possibile pubblicare i dati;
* Subscriber: alla loro connessione al broker (mantenuta dal subscriber stesso attraverso un token) allocano un thread e, attraverso quest’ultimo, si colloquia con un thread consumatore locale al broker, il suo compito è quello di recapitare i dati richiesti.

La comunicazione col broker avviene utilizzando i socket, i processi client attendono su essi fino alla fine dell’operazione, la connessione viene effettuata utilizzando TCP dal momento che UDP non è reliable. Per capire meglio il funzionamento, facciamo il seguente esempio: l’host A è un subscriber che si connette al broker e attende, l’host B è invece un publisher, quando quest’ultimo scrive un messaggio, esso viene ricevuto da tutti quelli connessi in quel momento (in questo caso A), per disconnettersi basta togliersi dal broker.

#### Formato del messaggio

Il messaggio utilizzato per la comunicazione tramite message broker è binario, l’unica parte non binaria è quella dei nomi e dei path-name, i quali utilizzano lo standard UTF-8, *permettendo così messaggi di dimensioni ridotte*. Un messaggio è composto dai seguenti campi:

* Header di controllo: permette di capire il tipo di messaggio inviato, in tutto sono 16;
* Lunghezza;
* Variabili della lunghezza: blocco di dati variabile che dipende dal tipo di messaggio;
* Payload: dati effettivi.

Tra i tipi di messaggi vi sono pingReq e pingResp, necessari per far circolare il token e quindi mantenere viva la connessione.

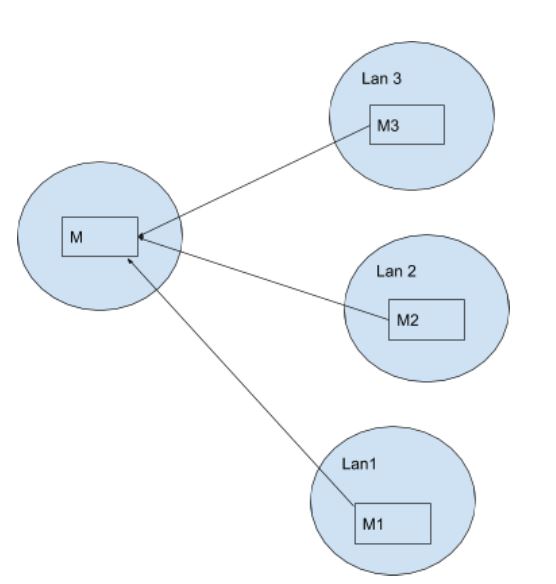
#### Qualità del servizio

La qualità del servizio del broker riguarda il recapito dei messaggi e non la velocità, ve ne sono di tre tipi:

* At Most One (Best-effort): fa un tentativo per consegnare il messaggio al broker, se funziona bene, altrimenti si accontenta. In questo tipo di QoS l’ACK non viene mandato *dato che non si preoccupa della ricezione dei dati;*
* At Least One: Il messaggio deve essere ricevuto almeno una volta, tutte le altre non alterano il comportamento per via dell’idempotenza. La conferma di ricezione avviene attraverso un ACK, se questo viene perduto, il mittente attende un intervallo e dopo ristrasmette il messaggio;
* Exactly One: il messaggio deve essere ricevuto una e una sola volta. Questo meccanismo è il più complesso, infatti per ricevere un messaggio una sola volta si procede nel seguente modo: viene inviato un messaggio al broker e quest’ultimo risponde con un PUBREC, se quest’ultimo non viene ricevuto, si ritrasmette il messaggio con un flag DUP. Vedendo che c’è il flag DUP, il broker scarta il messaggio e invia un PUBREC, quando ricevuto il mittente invia un PUBREL al broker e quest’ultimo risponderà con un PUBCOMP. Si fa quindi una sorta di three way handshake in cui *è possibile capire dove è possibile riprendere da dove ci si è fermati in caso di interruzioni*.

Il tipo di qualità del servizio viene scelto alla registrazione e ce n’è uno per ogni topic. Il problema sorge quando publisher e thread dispongono di QoS differenti, bisogna quindi fare in modo che la qualità del servizio del thread sia maggiore o uguale a quella di chi si è registrato *per evitare incidenti durante la trasmissione*.

#### Mosquitto

Mosquitto è un protocollo leggero del progetto Eclipse e sponsorizzato da IBM fornendo apposite librerie, l’organizzazione dei subject è ad albero e con esso è possibile fare Bridging, ovvero che si possono creare reti di broker con infrastruttura locale e nel cloud. I Subject possono essere cercati attraverso + e #, il primo indica un nome di un qualunque subject mentre il secondo indica un intero sottoalbero. Per fare un esempio, data l’immagine a lato, un subscribe B/a/x riceve solo da sè stesso, Un subscribe B/+/x può ricevere da B/a/x e da B/b/x mentre B/# può riceve da B/a/x, B/a/y, B/b/x e B/b/y. Ritornando a parlare del bridging, consideriamo il seguente esempio: m1, m2 e m3 fungono da publisher e subscriber di dati, l’interazione avviene col Mosquitto locale e possono comunicare sia con M, sia col cloud. M vede m1, m2 e m3 come dei client, questi ultimi possono pubblicare su M e, se un mn di un’altra LAN è iscritto al topic appena aggiornato, quest’ultimo riceverà una notifica, questo permette di creare sistemi che vanno oltre le reti locali. MQTT a differenza di Mosquitto non presenta alcuna forma di sicurezza, è comunque possibile implementarne una utilizzando la crittografia, Mosquitto invece li cripta in automatico solamente quando vi è Bridging.

#### Problematica concettuale di MQTT

MQTT utilizza un protocollo guidato dagli eventi, gestendoli quando essi si attivano. Un evento però può causare a sua volta altri eventi, andando contro la mentalità classica della programmazione funzionale, per risolvere ciò si astrae quindi questo meccanismo attraverso l’implementazione di chiamate a procedure in rete, anche dette chiamate remote.

### Representational State Transfer (REST)

Con l’evoluzione di Internet, si è passati alla sola visualizzazione di pagine web al web semantico, con essa è nata l’interfaccia REST: un metodo per modellare interazioni attraverso lo scambio di messaggi HTTP tra processi remoti, permettendo l’ottenimento di GUI in tempi ristretti (dato che sviluppata ad alto livello, quindi implementabile anche dai meno esperti) e altri vantaggi. Per fare ciò, REST utilizza quattro dei metodi predefiniti da HTTP:

* GET: richiesta di una data informazione;
* POST:creazione di una risorsa;
* PUT: aggiornamento di una risorsa;
* DELETE: cancella una risorsa.

L’approccio utilizzato è l’invio di un messaggio in modo formale in cui viene descritto cosa si vuole fare, di tale chiamata remota non è presente una tipatura forte, permettendo l’utilizzo di linguaggi di descrizione gestibili anche da chi non è esperto.

HTTP è utilizzato per inviare i messaggi, URI per organizzare la memoria e i linguaggi di markup servono a rappresentare i dati, tutto questo definisce il payload di un pacchetto REST, l’intestazione contiene invece la codifica, il comando, l’indirizzo (in formato URI) e la porta utilizzati, la versione di HTTP, eccetera. Al contrario di quel che si può pensare, REST non è uno standard, bensì è un modo per utilizzare gli standard in modo differente, ovvero la comunicazione tra processi, sfruttando l’utilizzo di tecnologie già esistenti senza crearne di nuove. Il sistema è definito da chiamate <URI, oggetto> che restituiscono l’oggetto associato a un dato URI. Quando si programma un sistema di questo tipo si definisce un proprio standard che però non è interoperabile, l’unica cosa che è possibile fare è quindi documentarlo. Questo approccio è potente perchè permette elasticità, tuttavia dato che non è standardizzato potrebbe risentirne la diffusione. Questo modus operandi ha portato all’approccio “Serverless”, ovvero si scarica l’applicativo che girerà su browser e comunica con REST col server solo quando necessario, ciò permette una facile manutenzione andando a prendere in considerazione solamente il lato server dell’applicativo. L’idea infatti è quella di importare la rappresentazione in stringa di un oggetto invece dell’oggetto stesso, in questo modo è possibile analizzarla e prendere ciò che è necessario.

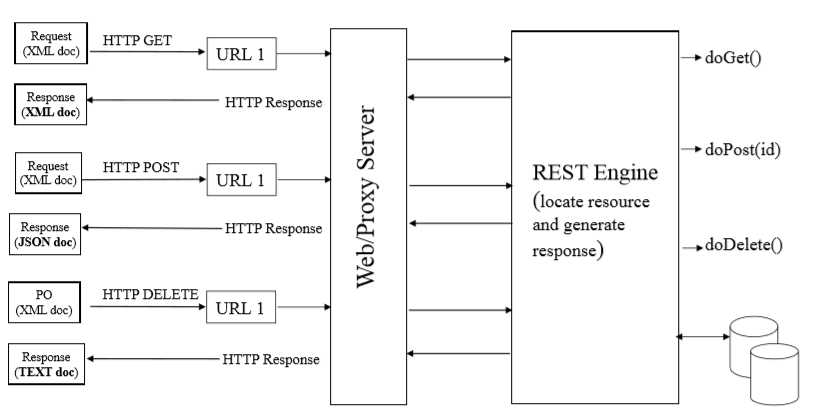
#### Concetti principali

REST si basa su tre concetti:

* Nomi: identificano le risorse utilizzando lo standard URI;
* Verbi: indicano i metodi HTTP utilizzati per le chiamate remote;
* Rappresentazioni: descrivono gli oggetti e li rappresentano attraverso i linguaggi di markup.

Le risorse sono l’astrazione di REST e sono associazioni di URI a insiemi di entità: ogni informazione denominabile è infatti una risorsa identificabile con un id globale (URI in questo caso). Il nome della risorsa, ovvero URI, non è altro che un path-name che permette di navigare attraverso i dati partendo da quelli più generici e andando verso lo specifico. La rappresentazione dei dati avviene principalmente in JSON o in XML, tuttavia è comune e comodo avere più rappresentazioni dello stesso dato. Una chiamata REST è il trasferimento di un oggetto il cui risultato è una risposta OK,ERROR oppure l’oggetto associato a un URI.

#### Architettura

Tutti i tronconi delle applicazioni REST devono includere un web server, esso offre servizi di memorizzazione o di implementazione di funzioni, adoperando macchine virtuali che eseguono l’informazione/chiamata data restituendo il risultato al mittente. Il web server si occupa anche di convertire tutte i metodi HTTP in funzioni in modo da essere facilmente comprensibili dalle macchine virtuali, *la cosa* *più importante infatti è che il server deve esser provvisto di classi che gestiscono HTTP proprio per questo motivo,* un’alternativa è l’implementazione di un programma che interpreta il tutto, prendendo un URI come parametro e restituendo un oggetto come risultato. Utilizzando l’approccio MQTT, è possibile scaricare una pagina web emulando HTTP col modello Publish-subscribe, rendendo così non necessaria l’aggiunta di un web server.

### Websocket

I websocket sono un’interfaccia per la comunicazione bidirezionale tra browser e server, essi sono utilizzati soprattutto per ottenere applicazioni reattive. Il primo approccio utilizzato dai websocket è quello client-server, *permettendo così il download di pagine web e altre funzioni*, tuttavia per aggiornare lo stato bisogna scaricare nuovamente la pagina (risolvibile con un reload automatico). Un’altra soluzione è l’utilizzo di un’applicazione nativa, essa supera il modello client-server *permettendo l’interattività e implementando il P2P, in questo modo è possibile sapere quando lo stato del server è aggiornato o meno*. Nell’approccio REST non è possibile utilizzare una PUT dal browser per l’aggiornamento di un dato, quindi è possibile utilizzare i websocket implementando il P2P e attivando una connessione tra due processi remoti. I websocket permettono cose possibili nelle applicazioni native ma non sul browser, in più viaggia sotto un protocollo perchè deve avere particolari caratteristiche: deve infatti avere robustezza e deve poter attraversare il proxy, per quest’ultimo problema si aggiunge uno strato software che emula un socket TCP, l’interfaccia MQTT e il relativo server viaggiano così attraverso il suddetto websocket.

#### Attivazione di un websocket

Per quando riguarda l’attivazione, si adopera un messaggio in stile HTTP che attiva il meccanismo del browser, tutto questo avviene codificando i dati utilizzando la chiave nel messaggio. Una volta attivato il websocket, si stabilisce un canale virtuale in cui viaggiano i pacchetti aventi la seguente struttura:



Il pacchetto permette l’attraversamento del socket in modo indisturbato, la maschera serve a non confonderlo con una pagina web, infatti è formata da pezzi di dati random non riconoscibili dal proxy. In Java esiste una classe WebSocket che implementa un meccanismo di callback: vi è un meccanismo simile al gestore degli interrupt che, all’arrivo di un segnale, attiva l’apposita procedura, esso *permette di fare quel che si vuole ai livelli superiori dal momento che è poco strutturato.*

### Zero-conf

La zero-conf è un tipo di architettura di rete autoconfiguabile utilizzando appositi protocolli e una filosofia che scende fino al livello IP, l*o scopo è quello di ottenere una rete funzionante che non dipende da infrastrutture o specifiche conoscenze, semplificando l’integrazione tra componenti*. Quando un host si connette a una rete già configurata, non vi è alcun problema, essi sorgono quando appunto la rete non è così: il modello infatti parte dal fatto di non avere nulla a livello di comunicazione, che la rete non sia strutturata e che, per comunicare, bisogna:

* scoprire chi fornisce gli indirizzi per farsene dare uno;
* cercare un DNS per prendersi un nome (chiedendolo in multicast);
* localizzare i servizi presenti nella rete.

#### Assegnamento degli indirizzi

Per quanto riguarda l’assegnamento di un IP, la versione 6 del protocollo ha il cosiddetto Link Local, un indirizzo di 8 byte sui 16 dell’indirizzo (8 rappresentano la rete e i restanti l’utente) che in base alle configurazioni possibili ne viene scelto uno libero, nella versione 4 non è presente però è stato introdotto a posteriori un approccio simile: vengono scelti dei numero casuali utilizzando l’indirizzo MAC dell’host come seme e si controlla che esso sia nello spazio di indirizzi e la sua disponibilità utilizzando ARP, *infatti se l’indirizzo è già assegnato il meccanismo viene portato a termine con successo, altrimenti no e quindi riprova.* Per l’ottenimento dei nomi, se nella rete è presente un DNS dinamico, si comunica con esso, assumendo però di non averlo si è costretti all’utilizzo del multicast per cercarli. Vi è però un problema: una macchina rischia di essere tracciabile dato che nell’IPv6 viene utilizzato l’indirizzo MAC, in più vi possono essere conflitti con esso *dato che non si può considerare univoco*, per risolvere ciò si genera casualmente un indirizzo e si prova a dialogare con esso, se nessuno risponde viene assegnato. Un host connesso a una rete con più macchine virtuali all’interno può comunicare con una parte o con l’altra in base alla sua configurazione, esse infatti sono virtualmente separate e si comportano come due host distinti, di conseguenza avranno due IP differenti.

#### Multicast DNS

Nel multicast DNS viene inviata una richiesta quasi standard la cui destinazione è un gruppo di multicast, ottenendo un risultato simile a una richiesta del DNS con la differenza che vi sono cambiamenti nell’implementazione: viene inviata un’unica risposta quando si chiede la corrispondenza nome-indirizzo per evitare ambiguità, per quanto riguarda i servizi vi possono essere più risposte *dato che più host possono fornire la stessa tipologia di servizio, le risposte vengono date in multicast in modo che tutti i clienti possano condividere l’informazione.* Una richiesta m-DNS viene fatta su un indirizzo IP per ottenere i suoi dati come risposta.

Lo stesso meccanismo permette il cosiddetto load-balancing: le richieste vengono distribuite equamente in base a una priorità, la scelta è casuale e si basa su quest’ultima (solitamente equivale alla quantità di job che riesce a smaltire). Tutti i servizi che un host vuole offrire vengono registrati in un daemon locale, per sapere dove trovarli, si esegue un multicast e si sceglie casualmente un host tra tutti quelli che lo erogano. Un problema è che la gestione degli host nel dominio .local non consente una struttura gerarchica degli stessi e, per proteggersi dagli attacchi esterni, il TTL dei pacchetti di risposta deve essere a 255 per prevenire attacchi dall’esterno, dato che un pacchetto con un TTL del genere non verrà mai propagato dal router.

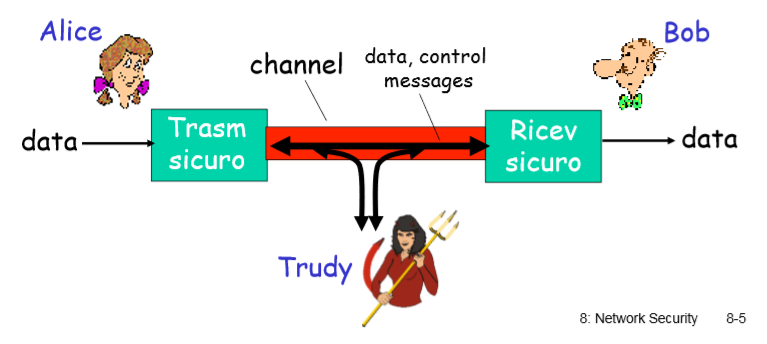
#### Assegnamento dei nomi

Per quanto riguarda l’assegnamento dei nomi, i singoli host creano i nomi che intendono utilizzare, tuttavia vi è la possibilità di creare conflitti, quindi si crea un nome finchè esso non è univoco, per farlo si utilizza il multicast. Nel caso in cui vi siano host testardi che non vogliono cambiare il proprio nome, invece di farlo fare a loro si cambia quello degli altri, lo stesso discorso vale anche per IPv4.

Un’alternativa è UP&P, un protocollo di Microsoft avente un meccanismo che permette l’autoconfigurazione, permettendo agli host di dialogare col router (che supporta il protocollo) per fare Port Forwarding, rendendosi visibili all’esterno, essa deve essere fatta adeguatamente per garantire la sicurezza.

### Sicurezza

Il discorso della sicurezza è equiparabile a quello della guerra: un cannone che distrugge un carro armato spinge a fare carri armati più robusti che a sua volta spinge a fare cannoni più potenti, allo stesso modo se un sistema viene violato viene reso più sicuro e così via. Un primo metodo consiste nell’utilizzo di Firewall per il controllo dell’accesso alle macchine attraverso le porte, un secondo metodo è invece quello di proteggere i dati con la crittografia, quest’ultima tecnica è più robusta (se fatta bene) *dato che con la prima è facile far entrare dei cavalli di Troia.* Con sicurezza, si intende:

* Confidenzialità: solo mittenti e riceventi possono sapere cosa si dicono, essa è gestita con la crittografia cifrando il messaggio in trasmissione e decifrandolo in ricezione;
* Autenticazione: garantisce che mittente e ricevente riconoscano l’identità dell’altro in modo reciproco;
* Integrità: il messaggio non deve essere modificato da terzi;
* Accesso e disponibilità: i servizi devono essere accessibili e disponibili agli utenti.

Facciamo il seguente esempio: supponiamo che vi siano un mittente e un ricevente che debbano trasferire dei dati, durante la comunicazione dei possibili intrusi possono inserire messaggi, fingersi uno dei due protagonisti (forgiando falsi IP sorgente) per ingannare l’altro, dirottare dei dati sostituendosi a uno dei due, sbirciare oppure bloccare il servizio.

#### Crittografia

La crittografia permette di soddisfare confidenzialità, autenticazione e integrità, essa interviene in trasmissione e in ricezione, essa può essere:

* Simmetrica: viene utilizzata la stessa chiave (generata da un algoritmo e condivisa al ricevente in qualche modo) per la cifratura e la decifratura di un messaggio;
* Asimmetrica: l’algoritmo genera una coppia di chiavi, quella pubblica permette di cifrare il messaggio ed è disponibile per tutti mentre quella privata è invece l’unica che può decifrarlo. Questo meccanismo permette di aggirare il problema di condivisione delle chiave *dato che il messaggio non è decifrabile con la chiave pubblica*.

La forma più banale di crittografia simmetrica è quella per sostituzione (detta anche cifrario di Cesare, il suo creatore), in essa i caratteri vengono sostituiti con altrettanti caratteri in base a un offset, la struttura del messaggio non cambia. Tutt’oggi il cifrario di Cesare non è più sicuro dato che è risolvibile con la forza bruta, quindi si è deciso di seguire la seguente idea: cercare di far diventare intrattabile lo spazio occupato e il tempo trascorso per un PC moderno, in poche parole si deve cercare un metodo efficiente per la cifratura e inefficiente per la decifratura, da questa filosofia sono nati DES e AES. Un’applicazione della crittografia si trova nelle VPN, i dati infatti vengono cifrati in trasmissione, fatti passare per un canale pubblico e decifrati in ricezione.

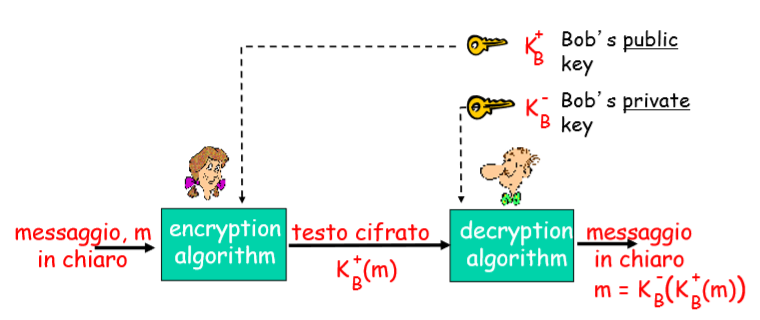
##### DES

Questo algoritmo di cifratura utilizza una chiave di 56 bit e un testo in chiaro di 64 bit, esso è stato violato in 4 mesi con la forza bruta dopo anni di resistenza, tuttavia esistono anche soluzioni senza backdoor. Per rendere sicuro il DES, si possono utilizzare tre chiavi in sequenza su ogni dato (3-DES) oppure si utilizzando la concatenazione di più blocchi di cifratura. DES cifra il messaggio eseguendo inizialmente una permutazione iniziale, in seguito eseguendo 16 cicli in cui viene applicata la funzione (utilizzando 48 bit differenti di chiave) e termina con una permutazione finale, restituendo il messaggio cifrato.

#### AES

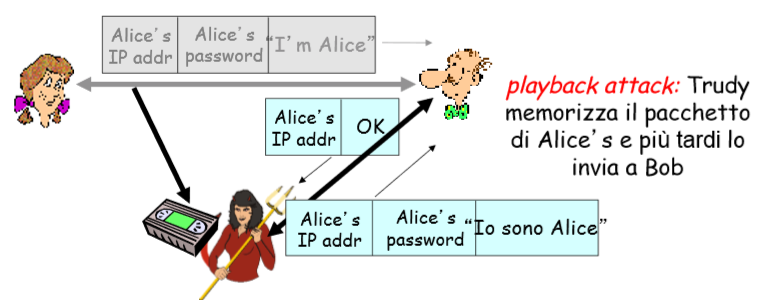
AES a differenza di DES permette l’utilizzo di chiavi più lunghe (la più sicura è quella a 256 bit) e con le attuali tecnologie non è violabile, infatti un normale PC ci mettere 149 trillioni di anni per violarlo con la forza bruta. Una possibile scappatoia sono i computer quantistici, essi sono in grado di prendere lo stato in un dato momento, permettendo così di conoscere il risultato della decifratura e quindi decifrare AES in tempo reale.

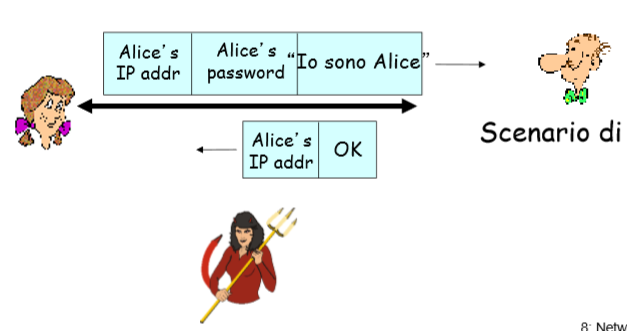
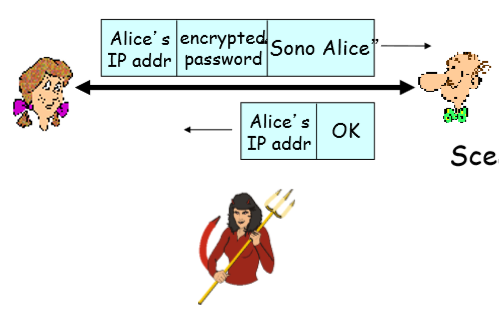
#### Crittografia asimmetrica

*Il problema principale degli algoritmi simmetrici sta nella condivisione della chiave*, non vi è alcun problema se viene fatto offline ma ciò non è sempre possibile, quindi per questo entra in gioco la crittografia asimmetrica: la coppia di chiavi permette infatti la cifratura con quella pubblica e la decifratura con quella privata, questo può garantire che un dato proveniente da un determinato mittente non sia stato modificato. L’approccio tutt’oggi utilizzato è quello di Diffie-Hellman, mittente e ricevente non condividono alcuna chiave segreta, al contrario la chiave pubblica è nota a tutti mentre quella privata è conosciuta solo dal secondo. Un problema di questo tipo di crittografia è la pesantezza e di conseguenza non è adatta per le applicazioni real-time, in compenso non vi è la necessità di scambiare le chiavi (solo se sono generate col metodo di Diffie-Hellman). Un particolare requisito per il funzionamento della crittografia asimmetrica è che, dato un messaggio m e c=publicKey(m), privateKey(c) deve essere uguale a m, in poche parole la decifratura di una cifratura di un testo in chiaro deve essere uguale al testo stesso, quindi le due chiavi devono essere scelte secondo questa regola:

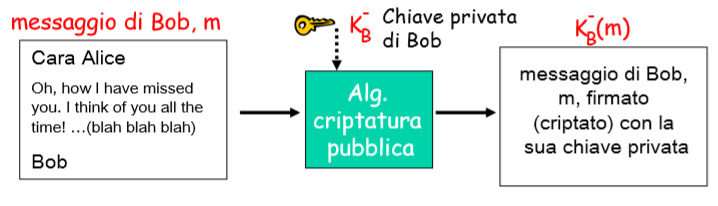
privateKey(publicKey(m))=m

Il secondo requisito è che, data una chiave pubblica, deve essere difficile a livello computazionale calcolare la chiave privata. Per la generazione delle due chiavi si utilizza RSA, esso funziona nel seguente modo:

* vengono scelti due numeri p e q molto grandi e primi tra loro;
* si calcola n=p\*q e z=(p-1)\*(q-1);
* viene scelto un numero e minore di n e non divisibile per z;
* si sceglie un numero d in modo che e\*d-1 sia divisibile per z;
* la chiave pubblica sarà quindi ottenuta dalla coppia (n,e) mentre quella privata da (n,d).

Una particolarità delle chiavi generate con RSA è che esse sono intercambiabili: le chiave pubblica può anche essere privata e viceversa. Per garantire più sicurezza, è possibile criptare il messaggio prima con la chiave pubblica del ricevente e in seguito con la propria chiave privata, il ricevente decripterà il messaggio prima con la chiave pubblica del mittente e in seguito con la sua privata. Vi è però un problema, possibili intrusi possono falsificare la chiave privata, per risolvere ciò si utilizza l’autenticazione: dal momento che in una rete un host può dichiararsi di essere un altro, *bisogna fare in modo che mittente e ricevente si riconoscano*. Un primo approccio sta nell’utilizzo dell’indirizzo IP, utile nei casi in cui vi è un accesso esclusivo a rete, in questo modo però gli intrusi possono confondersi con esso cambiando il proprio o falsificando il pacchetto, in più in Internet è più facile confondersi rispetto a una rete privata. Un secondo approccio è l’inserimento di una password segreta nel pacchetto e conosciuta solo dal destinatario, il discorso tuttavia rimane identico al precedente dato che *l’intruso può memorizzare il pacchetto e inviarlo al ricevente (confondendosi così per il mittente)*. Criptare la password non servirebbe a nulla *dal momento che il ricevente non sa che l’intruso si sta spacciando per il mittente*, ritornando così al caso precedente, questo attacco è detto playback. Una possibile soluzione è rendere la password diversa a ogni sessione, ciò avviene generando numeri pseudo-casuali da entrambi i lati, *in questo modo non possono fare niente anche se riescono a prenderla*. Per evitare il cosiddetto playback, il mittente chiede al destinatario di inviargli il cosiddetto Number Once, *un numero generato casualmente e utilizzato una sola volta*, esso viene criptato con la chiave condivisa e restituito al ricevente, quest’ultimo proverà che il mittente è chi dice di essere dal momento che il messaggio è criptato con tale chiave. La chiave utilizzata è simmetrica ed è stata decisa in precedenza, per condividerla si può utilizzare lo stesso ragionamento della crittografia asimmetrica: si chiede la chiave pubblica al ricevente, si cripta la chiave condivisa con essa e si invia, il problema è che terzi possono prendere il pacchetto e decodificare il contenuto, dal momento che “si mette in mezzo” spacciandosi al mittente per il ricevente e viceversa, *in questo modo riesce a ottenere sia la chiave pubblica che quella privata del mittente e di conseguenza dialogare con le entità.*

### Firma digitale

La firma digitale è un metodo che permette di dimostrare che un documento appartiene a un determinato mittente, chiunque riceve il documento firmato può dimostrare che è solo il mittente ad averlo fatto e nessun’altro, incluso il ricevente stesso. Un primo approccio è che, dato un messaggio m, esso viene criptato con la chiave privata e inviato. Si può dimostrare che il mittente ha firmato il documento *dal momento che è l’unico a possedere la chiave privata.* Per verificare che il documento sia firmato dal mittente, il ricevente applica la chiave pubblica del mittente su esso verificando che publicKey(privateKey(m)) sia uguale a m (la regola vista in precedenza), *se è così chi ha firmato ha utilizzato la chiave privata del mittente*, di conseguenza il ricevente verifica che il mittente ha firmato il documento, che quest’ultimo non sia stato firmato da nessun’altro e che egli non abbia firmato altri documenti, un’ulteriore verifica si può dimostrare portando il messaggio e la sua firma in tribunale. Questo metodo è tuttavia molto pesante da applicare, quindi si preferisce utilizzare una funzione hash aventi le seguenti proprietà:

* Dato un testo di grandezza arbitraria, la funzione hash deve restituire un testo di grandezza fissa, se l’hash è lungo k, ad esempio, vi possono essere al massimo possibili combinazioni prima di trovare il testo in chiaro;
* i valori di ingresso molto simili devono generare uscite diverse distribuite in modo pseudo-casuale;
* trovare un altro testo con la stessa funzione hash deve essere infattibile, anche se vi sono infinite combinazioni.

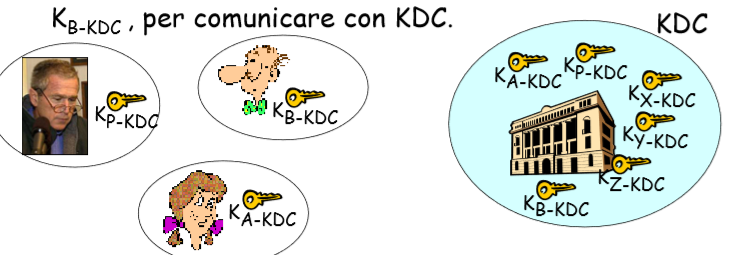
*Per evitare alterazioni*, la funzione hash viene criptato con la chiave privata.

Il processo avviene nel seguente modo:

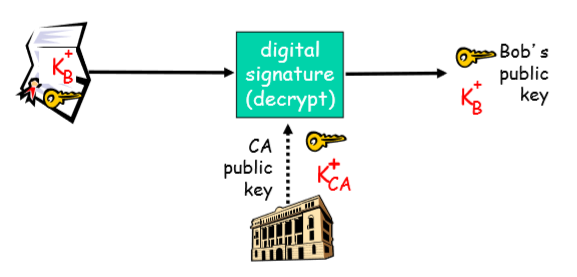
* Dato un messaggio m, si ottiene il messaggio c utilizzando la funzione hash;
* il messaggio c viene criptato con la chiave privata, ottenendo d;
* si lega m a d e si invia;
* in ricezione si calcola l’hash di m, si decripta d e si confrontano i risultati, la firma (e quindi l’integrità del messaggio) viene dimostrata dall’uguaglianza dei due hash.

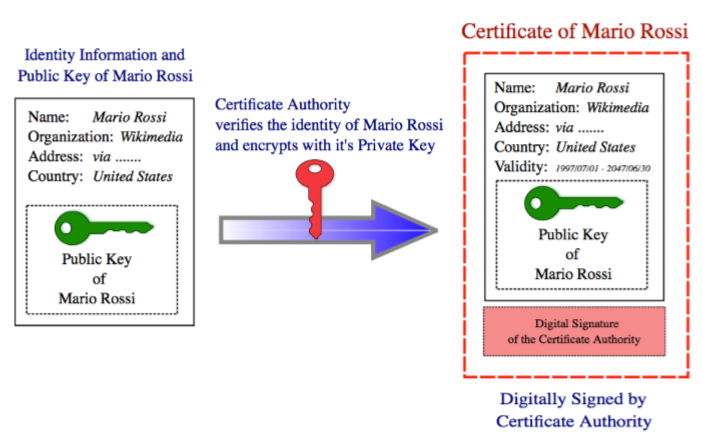
Eventuali modifiche a m hanno molta probabilità di corrispondere a un altro hash, di conseguenza *non vi è corrispondenza in ricezione*, l’unica cosa da garantire è che la chiave pubblica del mittente sia effettivamente la sua. Gli standard attualmente utilizzati sono MD5 e SHA1, il primo calcola un estratto di 128 bit in un processo a passi (è quindi difficile trovarne una uguale), il secondo invece calcola un condensato di 160 bit, quest’ultimo è tuttavia stato violato nel 2018 e quindi si è passati a SHA-256 in grado di generare condensati di 256 o più bit. Per distinguere un algoritmo dall’altro si inserisce il suo nome nel pacchetto, tuttavia eventuali alterazione rovinerebbero tutto.

### Distribuzione e certificazione

Per scambiare una chiave simmetrica tramite Internet in modo sicuro, si utilizza un intermediario, ovvero il centro di distribuzione delle chiavi (KDC), esso ha il compito di condividere chiavi segrete differenti per ogni utente registrato. Quando due client vogliono comunicare, il primo contatta il KDC con un messaggio criptato con la propria chiave privata chiedendone una per il secondo, esso risponde con una chiave e una copia criptata da inviare al secondo (decifrabile solo da lui), il tutto viene inviato utilizzando la chiave pubblica del primo. Una volta inviata la chiave criptata, il secondo la decifra e quindi è possibile iniziare la comunicazione col primo client in modo sicuro. In questo metodo di condivisione vi è però un problema: il primo client possiede sia la sua chiave, sia la sua copia criptata, di conseguenza potrebbe in qualche modo ricavare la chiave privata del secondo, tenersela e chiederne altre al centro, aumentando le probabilità di decifrare un messaggio, il primo client può inoltre fingersi il secondo una volta trovata la chiave.

### Autorità di certificazione (CA)

Una CA è un ente che permette il rilascio di certificati sulle chiavi pubbliche, garantendo che esse appartengano a una data entità, ogni volta che un client crea un coppia di chiavi pubblica-privata, l’autorità controlla che il client sia effettivamente lui e, se è così, la firma e l’autentica, permettendone così di inviarla in giro. Per evitare eventuali falsificazioni, la CA è spesso sigillata fisicamente, farsi rubare la firma digitale è infatti compromettente e per questo bisogna fare denuncia quando succede. La firma può essere reperita in rete e/o memorizzata nel browser, i sistemi di firma tuttavia tendono a non memorizzare la chiave privata nel PC per evitare che venga rubata. Un certificato contiene:

* Un numero di serie, unico per l’emettitore;
* Informazioni sul proprietario, sull’algoritmo e sul valore delle chiavi, quest’ultima parte non viene mai mostrata;
* Informazioni sull’emettitore;
* Inte di validità;
* La firma elettronica dell’autorità (la chiave pubblica).

Un altro sistema utilizzato è il PGP, in esso una chiave pubblica viene certificata da altri utenti a loro volta certificati, la macchina che la riceve ha una lista in cui vi sono le chiavi pubbliche, quindi controlla che il candidato sia nella comunità per vedere se accettarlo o meno.

#### OpenSSL

OpenSSL è una libreria opensource dai molteplici utilizzi, essa utilizza la codifica in base 64 (formata dalle lettere dell’alfabeto maiuscole e minuscole più i numeri e i simboli “+” e “/”), *permettendo il trasporto dati sicuro e senza corruzione* attraverso un raggruppamento delle stringhe in gruppi da 6 (con un codice assegnato a ognuno). Una particolarità di questo protocollo è la generazione di chiavi e password senza scambiarle con un KDC, per farlo si seguono i seguenti punti:

* si cripta la chiave simmetrica con quella pubblica del secondo;
* si calcola l’hash corrispondente;
* si cripta l’hash ottenuto con la chiave pubblica del secondo e si invia insieme alla chiave criptata.

In ricezione, si decripta l’hash e si calcola quello della chiave criptata per confermare che il mittente sia chi dice di essere, se è così si decripta la chiave criptata con quella privata del secondo e si inizia la comunicazione.

#### Nella realtà

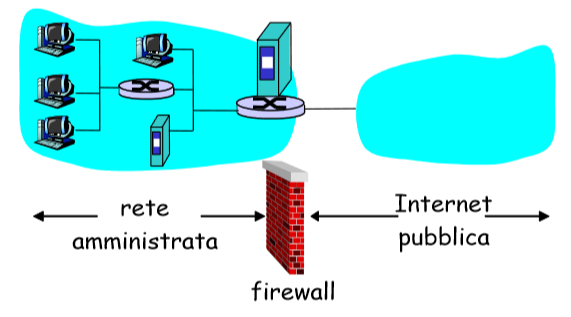
Per evitare di violare la privacy delle persone, i dati vengono criptati: supponiamo di avere un sistema di videosorveglianza di un negozio formato da tre entità: la sorveglianza, il rappresentante del personale e il proprietario del negozio, ognuno di essi dispone di una chiave privata e quindi, per rendere inutilizzabili i dati presi dalle videocamere, essi vengono criptati tre volte con le chiave pubbliche di ognuno. Le chiavi private vengono criptate con le rispettive chiavi pubbliche e consegnate a ogni entità tramite PEC, mantenendo così segreta la chiave.

#### Firewall

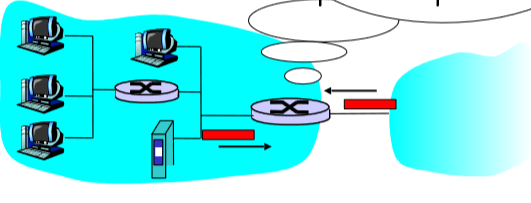
I firewall sono componenti di rete che permettono la protezione locale di una sottorete in cui sono presenti dati sensibili, consentendo solo a dati pacchetti di passare. Inoltre, i firewall consentono la creazione delle cosiddette zone demilitarizzate (DMZ), un’area della rete meno sicura e quindi fatta apposta *per far girare alcuni tipi di servizi pubblici*. Questo componente di rete aggiunge un livello in più alla protezione, *in questo modo gli intrusi ci metteranno più tempo a entrare e di conseguenza rischiare di essere scoperti*, tutto questo è possibile configurando il firewall in modo da adottare le politiche citate sopra. Un firewall può essere:

* Stealth: non è visibile a livello IP, funziona come un bridge e consente la comunicazione con certe porte e non con altre;
* Router: sono router con funzione di filtraggio dei pacchetti, la maggior parte implementa un kernel Linux;
* Applicativo: sono firewall che funzionano quasi come dei proxy, controllando tutto a livello applicazione.

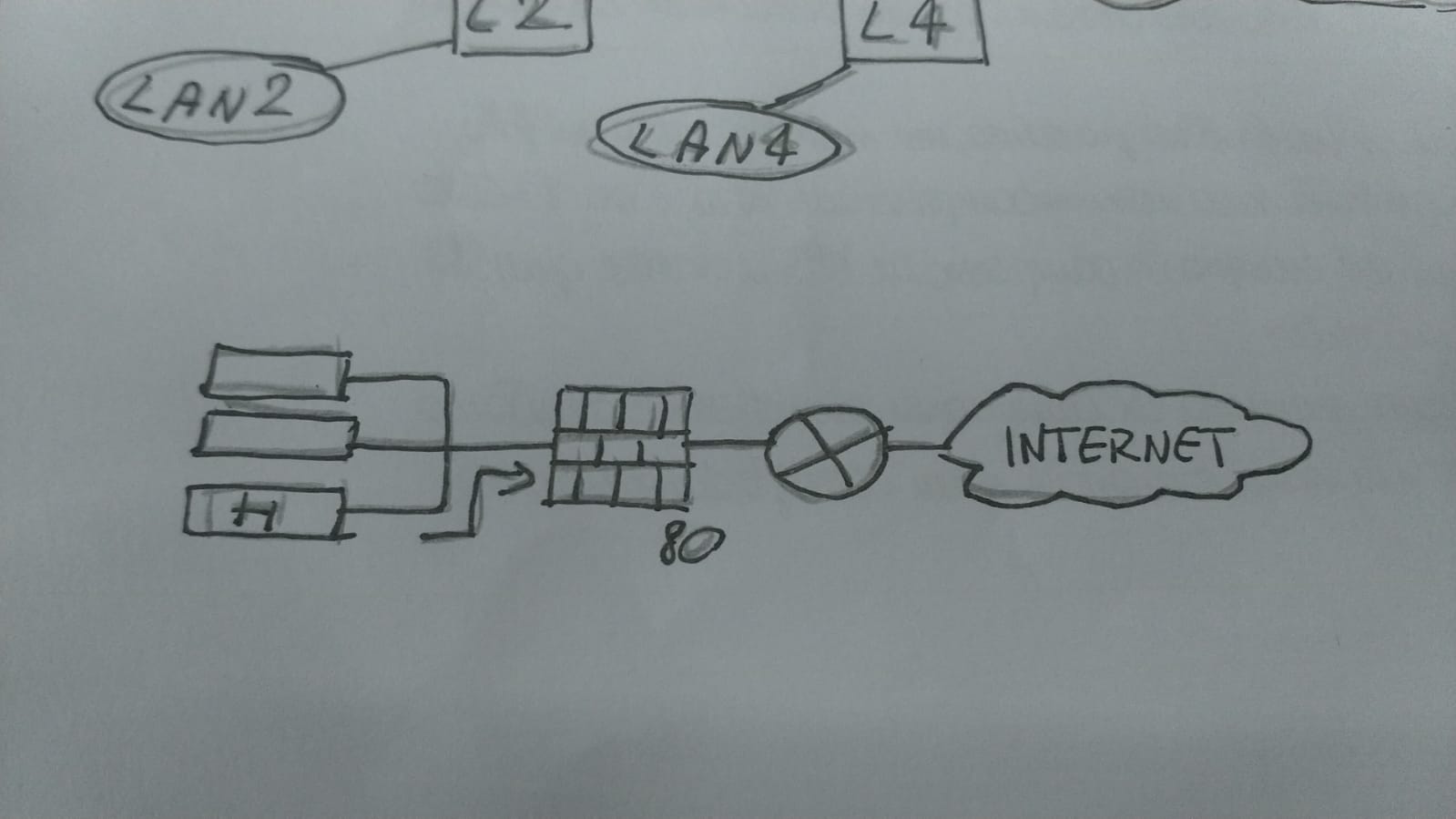
L’elaborazione del firewall è molto costosa e a seconda del caso d’uso può arrivare anche a migliaia di euro per l’analisi del traffico in tempo reale. La tendenza è l’utilizzo di un firewall insieme a sistemi di monitoraggio della rete, *essi andranno in allarme in caso di situazioni anomale*, tuttavia si deve avere un certo grado di tolleranza agli attacchi dato che possono essere presenti dei falsi positivi. Il compito del firewall è quello di:

* prevenire gli attacchi DoS evitando il SYN flooding, ovvero un numero spropositato di connessioni TCP finte che tolgono risorse a quelle vere, causando un time out, per farlo non vengono più accettati i segmenti di un dato IP dopo un’ di SYN consecutivi. Un altro problema riguarda la frammentazione, *dal momento che il pacchetto viene ricomposto in ricezione*, essa tiene occupato un processo e parte della memoria per gestire frammenti (e non) che continuano ad arrivare senza fine. 
* prevenire modifiche e accessi illegali ai dati interni, il loro accesso è infatti autorizzato solamento a chi si trova all’interno.

#### Filtraggio dei pacchetti

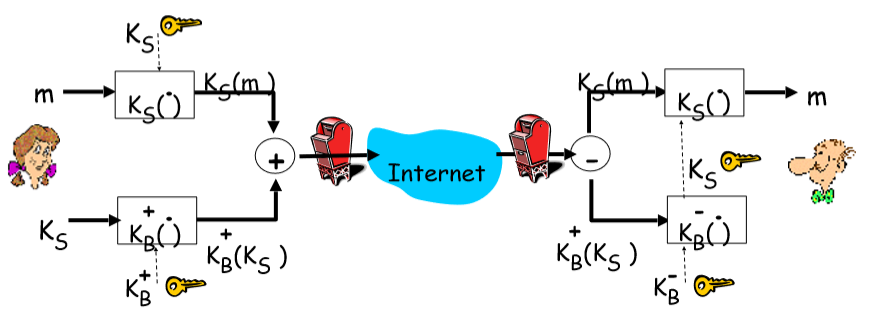
Il firewall che permette questa funzione è connesso alla rete e permette l’uscita verso internet, essi possono controllare gli indirizzi IP, le porte (e di conseguenza i servizi), i tipi di messaggi ICMP e i bit SYN e ACK a livello trasporto. Inoltre è possibile regolare l’accesso a dati e servizi, per esempio è possibile ammettere/bloccare gli accessi ai pacchetti che utilizzando un determinato protocollo del TCP/IP e/o una data porta di sorgente e destinazione, oppure bloccare i segmenti che arrivano dall’interno/esterno con certi valori nei bit. Nel caso di FTP, nel funzionamento passivo il client invia una richiesta al server per caricare o scaricare dei dati, in quello attivo invece è il server a connettersi col client inviando dei dati. Il firewall in questo caso impedisce a FTP di essere attivo, la modalità passiva permette quindi la comunicazione col client evitando attacchi dall’esterno, per questo motivo i server FTP si trovano nella DMZ, si fa ciò per impedire che i dati vengano modificati dall’esterno. Il firewall controlla il tipo di messaggi dato che possono essere utilizzati da intrusi per capire com’è fatta la rete e quindi agire di conseguenza, lo stesso discorso vale anche per i bit SYN e ACK, essi infatti indicano da quale parte arriva il pacchetto. 

#### Firewall Applicativi

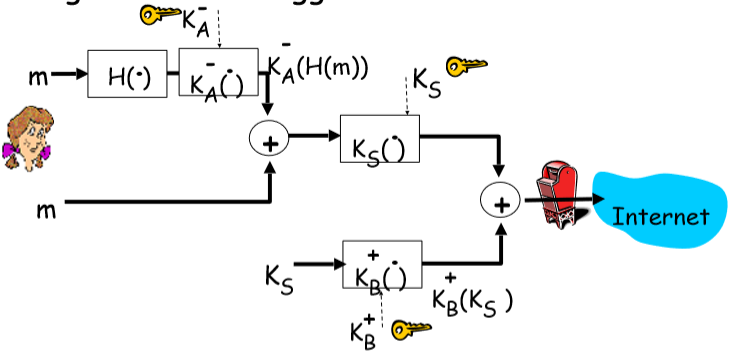
I firewall applicativi si comportano come dei proxy, essi permettono a una data macchina di accedere a Internet o a un suo sottoinsieme solo se è autorizzato a farlo andando a controllare le regole imposte. Quello che si fa solitamente è utilizzare un firewall che guarda i tipi di dati, le porte, eccetera in cui si impone l’uscita solamente attraverso un proxy, di conseguenza tutti gli host devono connettersi a esso per uscire. 

#### Limiti dei firewall

I firewall hanno però dei limiti:

* IP Spoofing: vengono mandati pacchetti con IP uguali a quello del proxy, di conseguenza è possibile uscire e quindi comportarsi di conseguenza nel caso vi siano altri livelli di protezione. Il router però non è solito guardare il campo dell’IP sorgente del pacchetto, quindi per limitarlo si deve forzare e indicare se gli IP in uscita si riferiscono a una data interfaccia, di conseguenza ogni applicazione deve avere un suo gateway per poter uscire in modo controllato;
* A livello software vi sono i cosiddetti cavalli di Troia, essi sono file eseguibili che, oltre a svolgere la loro funzione, attivano delle backdoor che permettono l’invio di pacchetti verso l’esterno o causando danni all’interno;
* I servizi non sono esenti da punti deboli, bisogna quindi trovare un modo per coprirli o evitare che non vengano sfruttati;
* Lo stack overflow consiste nel riempire lo stack della macchina con tanti livelli causati da chiamate di funzioni fino a quando non è pieno e quindi riuscire a prenderne il controllo.

#### E-mail sicure

La posta confidenziale coinvolge una serie di operazione che permettono al mittente di inviare o ricevere messaggi in modo sicuro: innanzitutto il messaggio viene criptato con una chiave simmetrica *per motivi di efficienza*, quest’ultima viene poi criptata con la chiave pubblica del ricevente e inviata insieme al messaggio criptato. In ricezione il destinatario decripta con la sua chiave privata quella criptata e con essa il messaggio. Per quanto riguarda l’autenticazione, il mittente firma il messaggio con una funzione hash, lo cripta con la sua chiave privata e lo invia insieme al messaggio stesso, in ricezione il destinatario decripta con la chiave pubblica del mittente e calcola un secondo hash dal testo in chiaro per confrontarlo col primo, *in questo modo è possibile garantire al destinatario l’integrità del messaggio e che il mittente sia appunto chi dice di essere*. Per garantire sia la segretezza del messaggio, sia autenticazione e integrità, il passaggio descritto in precedenza deve essere criptato con la chiave simmetrica prima dell’invio, il mittente utilizza quindi tre chiavi: la sua chiave privata, quella pubblica del ricevente e quella simmetrica. Una soluzione efficiente per mantenere segreta la chiave simmetrica è l’utilizzo di una SmartCard che ne genera una nuova ogni volta, tenerne una fissa nel PC non è infatti sicuro d*ato che malintenzionati potrebbero riuscire a prenderla facilmente.*

#### Sicurezza a livello di canale (SSL)

In una rete vi sono tre livelli di protezione:

* A livello Presentazione vi è la crittografia, esso è gestibile dal programmatore;
* A livello Trasporto/Rete vi è IPSEC, esso è gestibile dall’amministratore ed è quello che va da un host all’altro;
* A livello Rete/Data Link con WEP, WPA, eccetera, esso prende il collegamento tra due entità vicine nella rete.

SSL è un protocollo che permette la comunicazione sicura a livello presentazione criptando i socket, la crittografia avviene anche prima di una lettura/scrittura dopo che viene eseguita una connessione, per farlo però bisogna autenticarsi e generare una chiave simmetrica, per farlo si sceglie uno dei seguenti approcci:

* Soft: il server si autentica per indicare che è chi dice di essere e apre un canale criptato col client, quest’utimo può utilizzare username e password;
* Strong: entrambi si autenticano e scambiano informazioni attraverso un canale criptato.

Per quanto riguarda l’autenticazione, si effettua un handshake in cui vengono utilizzate le stringhe casuali, *esso sono una forma di identificazione a livello di rete, inoltre permettono una protezione dal packet sniffing, infatti non è possibile utilizzare i pacchetti vecchi dato che viene ogni volta generata una stringa nuova,* anche lo spoofing viene meno dal momento che, nel caso non abbia la chiave privata, non può decriptare i messaggi.Questo meccanismo serve per generare le chiavi segrete necessarie per la comunicazione, funziona nel seguente modo:

* Il client chiama il server inviando un messaggio contenente una stringa casuale, la quale identifica la connessione, e i suoi algoritmi di crittografia;
* Il server risponde con un messaggio contenente la tecnica da utilizzare e una stringa casuale, in più manda anche il certificato con la chiave pubblica firmata dalla CA. Nel caso il server chieda al client un certificato per l’autenticazione di quest’ultimo, esso invia anche una richiesta contenente tutti quelli supportati e i nomi delle CA accettabili;
* Dopo aver verificato il certificato, il client invia una master-key, generata casualmente, con cui è possibile generare la chiave segreta, essa è criptata con la chiave pubblica del server. In più, in caso di richiesta di certificato, il client invia una stringa casuale criptata con la sua chiave privata insieme al certificato o un avviso (quest’ultimo dà esito negativo nel caso in cui l’autenticazione sia obbligatoria);
* Il server verifica il certificato del client;
* il client finisce l’handshake inviando un messaggio di fine criptato con la chiave segreta. Dopo aver ricevuto il messaggio, il server farà lo stesso;

Oltre alla stringa casuale, SSL genera una chiave KH, la concatena al messaggio, ne calcola l’hash e lo si invia, *l'hash infatti può essere ricostruito solamente se si posseggono entrambe le componenti*, permettendo in questo modo di evitare di firmare il messaggio quando esse mancano.

#### Sicurezza con SSH

SSH è un protocollo nato in parallelo a SSL, esso consente la comunicazione sicura a livello di riga di comando, sostituendo Telnet da quale ne eredita il funzionamento. La comunicazione avviene nel seguente modo:

* Il client si connette all’SSH daemon presente nel server, qui avviene un prima fase di autenticazione tramite user-password oppure utilizzando le chiavi pubbliche, in quest’ultimo caso si utilizzano i number once (si riceve questo numero dal server, si cripta con la propria chiave privata e lo si spedisce a quest’ultimo);
* Dopo l’autenticazione, SSH daemon stabilisce un canale criptato tra il client e il server e creerà in quest’ultimo un processo su cui è possibile inviare comandi, si ha quindi l’illusione di essere davanti alla macchina a cui ci si è connessi.

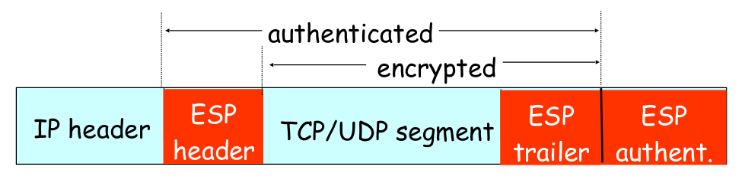
La differenza principale tra SSH e SSL è che quest’ultimo è utilizzato nelle reti di grandi dimensioni (avvalendosi delle CA), il primo è invece gestito da un gruppo ristretto e dal system manager. Il canale criptato citato in precedenza è detto tunnel SSH, esso viene concordato col daemon permettendo l’aggancio di una data porta del client con una del server o di una macchina remota. Tutto quello che arriva sulla porta della macchina remota viene intercettato dal daemon, criptato e spedito al client attraverso il socket. VPN si può considerare come una versione generalizzata di SSH, la differenza è la comunicazione sicura può essere implementata su più terminali rispetto a quest’ultimo che ne può connettere solamente due, è comunque possibile emularla aprendo più connessioni SSH verso differenti host remoti.

#### Protezione nativa di MOSQUITTO:

* Login/Password: si richiede username e password al client per l’accesso al broker, quest’ultimo può richiedere la QoS e il relativo meccanismo oppure la persistenza, ovvero che i messaggi non ricevuti vengono consegnati alla riconnessione. Ogni client può essere associato a una lista di topic in cui vi le parti accessibili a esso;
* Utilizzo di un canale SSL criptato per evitare che i dati vengano letti da terzi facendo connessioni e scambiandosi il certificato. Se i broker sono tutti di uno stesso proprietario è possibile creare un sistema chiuso che non permette accessi dall’esterno, *tuttavia questo è un problema dato che dovrebbero essere liberi;*
* Introduzione di componenti che fanno da imbottitura per evitare che i messaggi vengano riconosciuti dalla loro dimensione (infatti sono deducibili dalla regolarità dei dati), *senza essi è infatti più facile decifrare il messaggio.*

#### IPSEC

IPSEC è uno standard collocato tra il livello Rete e Trasporto il cui scopo è quello di rendere sicuro il protocollo IP creando strutture dati che implementano VPN. Un pacchetto IPSEC comprende un authentication header che permette di evitare lo spoofing e man in the middle senza criptare i dati. Per quanto riguarda la crittografia, è possibile implementarla anche in presenza di tunnel. L’autenticazione viene effettuata tramite l’authentication header, esso ha un meccanismo che permette di inserire un IPSEC header tra il payload e l’header del pacchetto e contiene tutto quello che deve esse essere invariante attraverso la rete, infatti non può includere gli indirizzi IP dato che potrebbero cambiare a causa di una NAT, la sua struttura è aperta e variabile e *permette di inserire più informazioni a catena fino al payload come in IPv6*. Il protocollo ESP è utilizzato per garantire autenticazione e integrità del messaggio, il suo compito è criptare il segmento e le informazioni di autenticazione per decifrarle in ricezione, questo metodo ha tuttavia lo svantaggio di essere pesante a livello di rete, *per questo motivo si preferiscono i metodi a livello presentazione per un’applicazione più snella.*



#### Sicurezza nelle reti wireless

La sicurezza è opportuna anche nella connessione senza fili dal momento che è visibile a tutti e quindi sniffabile, per farlo si attivano dei meccanismi posti tra il livello Rete e Data Link:

* WEP: questo meccanismo utilizza una password comune e una crittografia banale, su ogni pacchetto infatti vengono generate chiavi a catena utilizzate per cifrarlo, di conseguenza se queste chiavi vengono registrate da terzi, è possibile decifrare i pacchetti facilmente anche perchè le fasi di autenticazione sono standard. L’autenticazione avviene utilizzando i Number Once: si invia questo numero a un host, esso lo decripta con la chiave condivisa e lo invia all’access point il quale ne controllerà la correttezza;
* 802.11i: é uno schema di protezione, implementabile a piacere, in cui vi è un meccanismo di distribuzione delle chiavi, ogni host infatti può averne una differente. Per l’autenticazione è ottenuto tramite una chiave WPA o WPA2 utilizzando un server dedicato oppure direttamente l’access point (quest’ultima alternativa non è molto consigliata dato che è più facile prendere la chiave), il meccanismo è composto da quattro operazioni: in presenza di un client che vuole accedere, l’access point chiede al server, quest’ultimo genera una chiave master (utilizzata per generare chiavi di durata limitata, essa impedisce decifrature su basi statistiche) e la invia alle altre entità, queste ultime generano infine la chiave di sessione e può iniziare la comunicazione, essa avviene in modo simile a SSL;
* EAP: vi sono server di autenticazione, detti radius, aventi un proprio protocollo di autenticazione, la comunicazione si appoggia a SSL, per farla si eseguono i seguenti passaggi:
  + si stabilisce una connessione criptata col server e si negozia l’entrata in rete, in questa fase vengono ricevute le chiavi master e tutto il necessario;
  + si passa su un canale in chiaro perchè la connessione criptata non è ancora attiva a livello wireless;
  + si attiva la crittografia tra client e access point; Questo meccanismo diventa trasparente e automatico dopo l’ultima fase, in più permette una buona autenticazione anche se si ha bisogno di buona attrezzatura fin da subito.

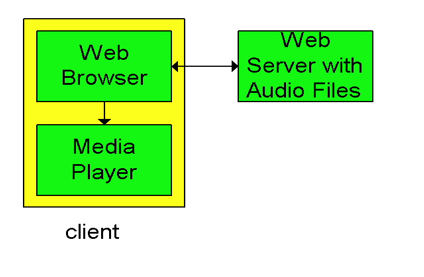
La tendenza odierna è l’utilizzo di reti libere in cui le applicazioni criptano i dati per conto loro.

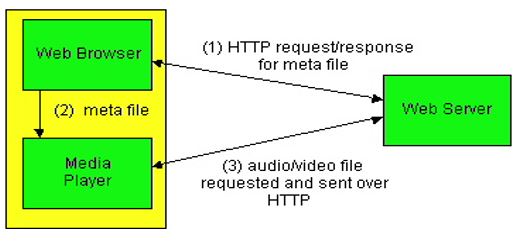
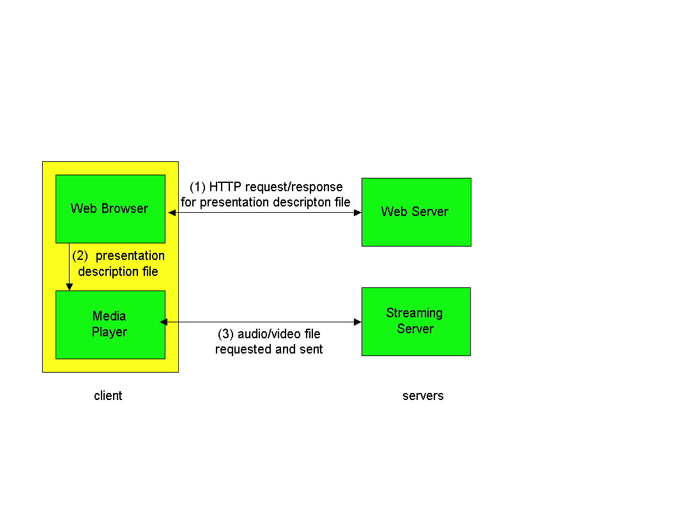
### Reti multimediali

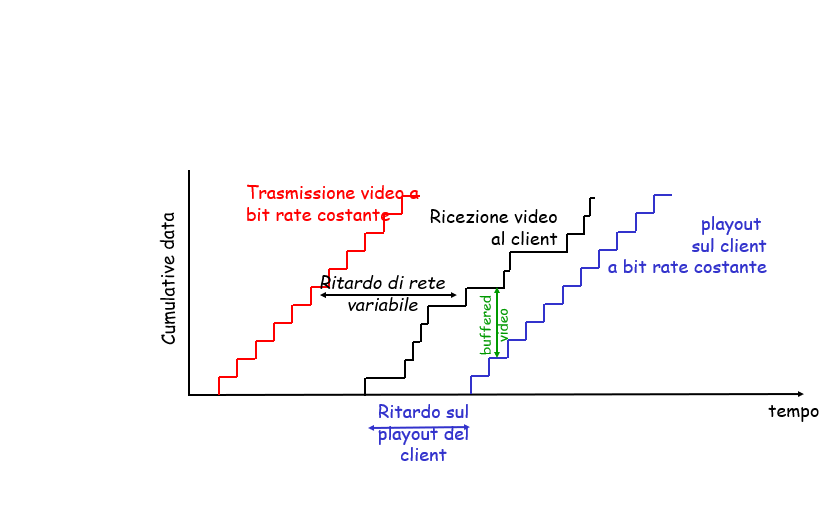
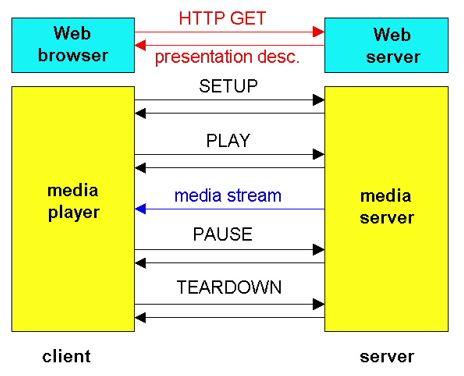
Le reti multimediali sono particolari tipi di reti in cui vi è lo scambio di contenuti multimediali, a differenza del calcolo i dati devono arrivare integri e col minor numero di errori possibile *dato che riguardano i nostri sensi e la nostra psiche*. L’essere umano è infatti tollerante in termini di qualità: un’immagine anche a bassa risoluzione deve essere comprensibile, lo stesso discorso vale anche per la mancanza di voce causata dal rumore e così via. E’ quindi possibile sfruttare queste informazioni per contenere i ritardi, di conseguenza *non è possibile utilizzare i protocolli che ne introducono anche a fin di bene*, come nel caso di TCP, quindi si tende all’utilizzo di UDP *perchè permette anche il multicast*. Per ridurre i flussi, si comprimono i dati e si elimina la ridondanza mantenendone l’integrità, un’alternativa è invece l’approssimazione, ovvero si mantengono i dati dal punto di vista percettivo. Per una migliore comprensione da parte del calcolatore, un segnale analogico presente in natura viene campionato per convertirlo in una matrice di numeri in modo tale da essere ricostruibile, a ogni valore viene poi data una sensibilità in byte, nel caso delle immagini si dà un byte per ogni colore. Nel suono i segnali vengono campionati utilizzando un convertitore A/D, i campioni ottenuti vengono quantizzati arrotondati a valori discreti e in seguito compressi utilizzando tecniche che controllano la differenza tra i vari campioni, infatti se essa è minima vengono raggruppati. In più, *dato che il cervello è in grado ricostruire i dati a livello percettivo*, è possibile eliminarne alcuni durante la compressione.

Le reti multimediali sono classificabili in tre tipi:

* Distribuzione preregistrata in streaming: tipo di rete in cui il file non viene scaricato ma viene visto direttamente in rete, rendendo anche più difficile la pirateria;
* Streaming in diretta;
* Videoconferenza-chiamata: tipo di rete in cui vi sono due o più entità che interagiscono, la sincronizzazione è difficile da mantenere

La fluttuazione del ritardo dei pacchetti in una stream è il cosiddetto jitter, per diminuirlo si utilizzano dei buffer a livello di client, il contenuto continua ad andare finchè il buffer è pieno e regolarmente viene riempito dal server con nuovi frame, vi sono differenti modi per farlo:

* il modo più semplice è non fare streaming, quindi il browser scarica tutto e infine lo passerà al media player, tuttavia non è presente pipelining e quindi vi sono lunghi ritardi prima della riproduzione vera e propria;
* Una seconda soluzione è effettuare lo streaming tramite un web server in cui browser e media player si appoggiano a esso rispettivamente per ottenere un metafile e uno stream, tuttavia vi possono essere ritardi dato che utilizza HTTP (il quale si appoggia a TCP);
* La terza e migliore soluzione è limitare il web server alla sola interfaccia e utilizzare un ulteriore server che utilizza UDP per lo streaming (tranne nei casi in cui il firewall non lo permette), come nel punto precedente il browser prende il metafile e lo passa al media player, in questo modo può sapere dove andare per la riproduzione del video.

Il buffer non deve essere eccessivamente grande, se lo è infatti si introduce un ritardo troppo grande (dovuto al buffering) mentre se è troppo piccolo rischia di svuotarsi presto, la soluzione migliore è adattarlo in base all’andamento della trasmissione.

La comunicazione tra client e server emula un videoregistratore attraverso il protocollo RTSP: esso funziona utilizzando una coppia di porte in cui il media player manda i comandi allo streamer e un’altra su cui passano le stream.

Con RTSP è possibile utilizzare il multicast dato che si appoggia a UDP, tuttavia non ha molto senso dato che è difficile da sincronizzare perchè gli utenti dovrebbero tutti vedere lo stesso film allo stesso momento, questa soluzione si può però adattare agli eventi in diretta. La più difficile però riguarda le applicazioni interattive: l’interattività infatti influisce sui meccanismi percettivi umani e quindi la risposta di un utente può arrivare più tardi rispetto alle nostre tempistiche (massimo 400 ms), bisogna quindi trovare un modo che lo permetta. La soluzione migliore sarebbe una rete con qualità del servizio, tuttavia la rete odierna non lo permette a causa delle congestioni e di altri problemi, di conseguenza il software deve essere pensato in modo da limitare queste fluttuazioni, per farlo si utilizza un ragionamento duale a quello di TCP: invece di introdurre ritardi per mantenere l’integrità, viene ridotta la qualità del contenuto senza rallentamenti per far fronte alle congestioni, cercando quindi di evitare che le collisioni non superino le tempistiche umane. In caso di perdita di un pacchetto, invece, si può ignorare oppure si sfrutta la ridondanza per coprirlo.

#### Internet phone

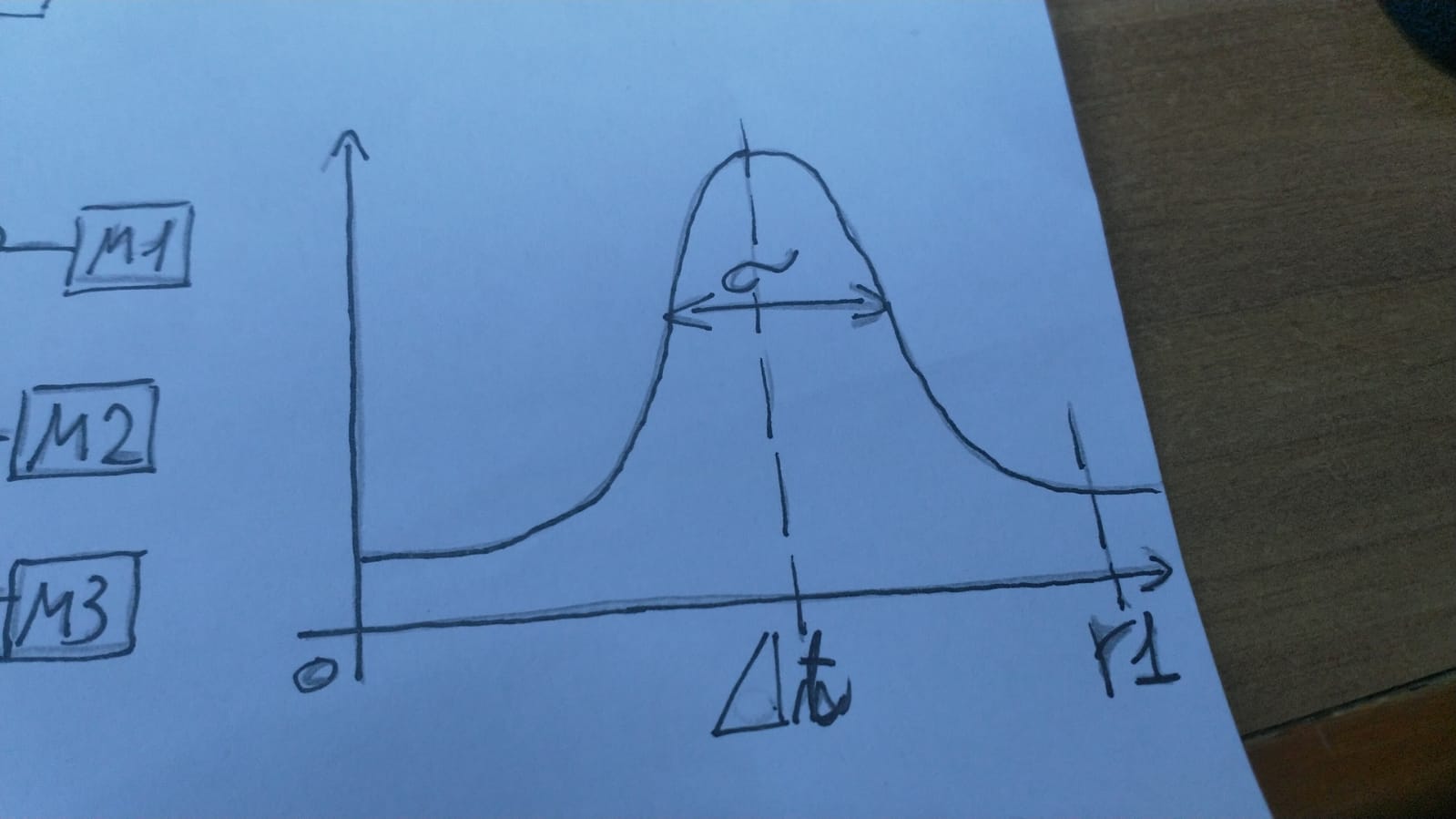
La telefonia su rete funziona nel seguente modo: un microfono genera un segnale elettrico a partire dalla voce e, con un convertitore A/D, viene generato un segnale in cui ogni campionamento corrisponde a un numero tipicamente di 8 bit, dopo si riempe il frame con questi dati e si spedisce in rete. In ricezione avviene il contrario: vengono estratti i dati dal frame e mandati verso un convertitore D/A che ricostruisce il segnale elettrico a partire dai campioni, ora è possibile sentirlo attraverso un altoparlante. La qualità del segnale varia a seconda dell’utilizzo: la musica avrà bisogno di una qualità maggiore rispetto al telefono, *dal momento che quest’ultima ha il solo compito di comunicare a distanza*. Per diminuire il volume dei dati è possibile inviare solamente la differenza tra un campione e l’altro al posto di quello intero. Il protocollo di compressione viene invece deciso dai client prima della comunicazione accordandosi sul CODEC. La telefonia in Internet non è tuttavia esente da ritardi, se questi sfiorano il jitter si rischia di avere un ritardo nella comunicazione, in tal caso vi sono tecniche che permettono il recupero, si cerca comunque di lavorare sul buffer in modo che il jitter non crei problemi, rendendolo il più piccolo possibile per non superare le tempistiche umane. Non vi è nessun problema finchè la rete introduce un ritardo piccolo, essi però si presentano quando si ha un jitter alto, per risolvere ciò si utilizzano delle tecniche adattative: quando un frame arriva fuori tempo viene scartato e si continua a prendere i successivi in ordine. E’ possibile ridimensionare il buffer nei periodi di silenzio, permettendo così di variare il ritardo in base alla situazione della comunicazione e/o della rete.

##### Soluzione a ritardo fisso

Quando il ritardo nella riproduzione è fisso, dato un ritardo q, il ricevitore tenta la riproduzione di ogni spezzone a q ms dopo che esso viene generato. Ogni spezzone ha un’etichetta temporale t che indica quando viene generato (il suo timestamp), quindi viene riprodotto esattamente a t+q ms, tuttavia se lo spezzone arriva in un tempo maggiore, i dati arrivano troppo tardi e quindi sono considerati come persi. Il ritardo q deve quindi essere un compromesso tra un minor numero di perdite di pacchetti e più ritardi (q grande) e una miglior esperienza interattiva (q piccolo).

##### Soluzione a ritardo adattativo

La soluzione al ritardo adattativo ha bisogno del timestamp, l’obiettivo infatti è minimizzare il ritardo mantenendo un tasso di perdita basso. L’approccio utilizzato è presente anche a livello trasporto per il calcolo del ritardo in TCP: esso è un aggiustamento adattativo del ritardo in cui a ogni istante t(i) viene ricalcolata la media secondo la seguente formula:



dato un pacchetto i:

* di è il ritardo medio;
* ri è il suo tempo di ricezione;
* pi è il tempo di riproduzione;
* ri-ti indica il ritardo della rete (jitter);
* ti indica il timestamp;
* u è un numero compreso tra 0 e 1 che permette di pesare la media.

Il ritardo attuale è quindi una media pesata tra quello attuale e la media vecchia, quest’ultima decade esponenzialmente man mano che si va avanti, soprattutto in presenza di valori molto grandi di u. Questo ritardo cambia solamente nei momenti di pausa, ovvero quando si regola il buffer, onde evitare troppe distorsioni nella voce o nel segnale. Per stimare il rischio di un ritardo, ovvero di quanto può variare, serve la sua varianza: mettendo la soglia in r1, vi è molta probabilità di non perdere i pacchetti, cosa che non succede in dove vi è il 50% di possibilità che succeda. Per far fronte a ciò,si può calcolare la deviazione media del ritardo vi:

Le due stime vengono calcolate per ogni pacchetto anche se sono solamente utilizzate all’inizio di ogni fase parlata. Per il primo pacchetto il ritardo è:

in cui K è una costante positiva.

Quando la rete diventa regolare, tutto tende a K e quindi tende a 0, quest’ultimo valore non deve superare le tempistiche umane e, se ci si avvicina a questo limite, si cerca di aumentare la compressione cambiando tecnica.

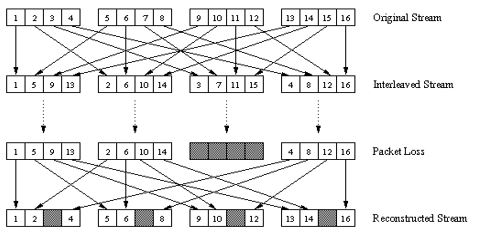
Per determinare se ci si trova in un periodo di silenzio o meno, il trasmettitore misura la potenza del segnale in quel livello e, se sotto un certo limite, indica che lo è e quindi non trasmette, al termine del periodo comincia la trasmissione utilizzando il nuovo ritardo. Il problema viene quando ci sono perdite nei pacchetti, bisogna quindi richiedere la doppia informazione: il tempo in cui il pacchetto è stato trasmesso e il numero di buchi, il primo indica se vi sono delle pause o meno mentre il secondo le eventuali perdite. Quando manca un pacchetto, il delay aumenta e ci sono buchi tra i sequence number, indicando delle perdite, se invece è una pausa, il delay aumenterà ma non saranno presenti . In caso di perdita di pacchetti è possibile ignorare il problema se è piccola, altrimenti bisogna utilizzare tecniche di recupero che aumentano il contenuto dei dati trasmessi.

##### Ridondanza old school (RAID) La ridondanza è una tecnica adattativa simile al RAID, vi sono infatti dei record ridondanti che permettono la ricostruzione dei blocchi perduti, ragionando nel seguente modo: il primo byte è lo XOR di tutti i blocchi, se ne viene perso uno è possibile ricavarlo utilizzando lo XOR con quelli che restano. Gli svantaggi sono l’aumento del volume col rischio di perdere blocchi e un peggioramento dell’intasamento, in più bisogna introdurre un ulteriore ritardo per via del pezzo in più, può supportare al massimo una sola perdita.

##### Doppia stream

Questa tecnica adattativa consiste nella trasmissione di due stream rispettivamente di qualità alta e bassa che si alternano in base alle condizioni della comunicazione e della rete. Le due stream viaggiano all’interno dello stesso frame e nel caso la prima venga perduta, si può coprire la perdita utilizzando la seconda. L’aggravio sulla banda dipende dal livello di qualità della rete.

##### Interleaving

La tecnica adattativa dell’interleaving utilizza la seguente idea: dato un flusso di dati, se si perde dell’informazione su un tratto molto piccolo, poco ci importa perché è possibile ricostruirlo, per farlo i dati vengono campionati, ridistribuiti su delle stream differenti e inviate. In caso di perdita di un pacchetto, è possibile ricostruire il segnale utilizzando tutti gli altri pezzi attraverso tecniche di interpolazione senza aggiungere altra informazione, tuttavia non tutti i segnali possono essere trattati in questo modo e la ricostruzione è possibile solo se sono arrivati tutti i blocchi, aggiungendo quindi un ulteriore ritardo. Le immagini si adattano bene a questo tipo di tecnica.

#### Real-Time Protocol (RTP)

RTP è un protocollo standardizzato che permette di veicolare i dati multimediali utilizzando un’apposita struttura a livello di pacchetto per trasportarli, essi si appoggiano a UDP per evitare rallentamenti se è presente una connessione ottimale, altrimenti è anche possibile utilizzare TCP in presenza di firewall che bloccano il primo. Questo protocollo possiede le informazioni necessarie per lavorare in real-time, tuttavia non fornisce alcun tipo di qualità del servizio, in più non impacchetta i dati utilizzando formati differenti, infatti fornisce solamente un supporto standardizzato per la trasmissione di questo tipo di dati in modo da offrire interoperabilità tra componenti. Insieme a questo protocollo ve n’è un altro che viaggia insieme, esso è RTCP e permette di verificare la qualità della rete, rendendo così possibili cambiamenti al volo nella codifica delle informazioni. Come già detto in precedenza, RTP non controlla la qualità del servizio dato che non fornisce meccanismi che assicurino il recapito, in più non viene imposto alcun tipo di codifica. L’header RTP presenta i seguenti campi:

* Tipo di payload: indica il tipo di codifica utilizzata in quel momento, in caso di cambiamenti è possibile notificarlo agli altri con questo campo;
* Sequence number: indica il numero di sequenza del pacchetto, esso è utilizzato dal protocollo per scoprire se vi sono state perdite di pacchetti e per ricostruire le sequenza corretta;
* Timestamp: indica l’istante di campionamento del primo byte del pacchetto;
* Id della sorgente: serve a sincronizzarsi con la sorgente dei dati, esse infatti possono essere molteplic, esse possono avere timing differenti.

#### Real-Time Control Protocol (RTCP)

RTCP è un protocollo che funziona in multicast e ha il compito di raccogliere e recapitare le informazioni sul funzionamento dell’applicazione a livello di rete, in base a essi il trasmettitore può adattare la sua trasmissione. Nelle videoconferenze ogni stazione è allo stesso tempo sender e receiver e comunicano normalmente a turno, tuttavia è possibile farlo simultaneamente quando si è in pochi.

Esempio: caso in cui sono presenti un trasmettitore e più ricevitori:

In questo caso RTCP è gestito da entrambe le entità, i trasmettitori possono indicare le sorgenti e il relativo timing per la sincronizzazione, il numero di pacchetti inviati e altro, i ricevitori indicano invece il numero di pacchetti ricevuti, il ritardo medio, eccetera. Con questi dati è possibile capire se una rete è intasata o meno guardando il numero di pacchetti persi, in tal caso si cambia codifica dei dati, l’operazione è analoga anche in caso di miglioramenti delle condizioni della rete.

##### Scaling

Per evitare che i pacchetti RTCP intasino la rete, il protocollo sta attento a non superare un certo consumo, cercando di non superare il 5% della banda disponibile. Ritornando al precedente esempio, al sender e ai receiver vengono assegnati da RTCP rispettivamente il 25% e il 75% di questo 5%, permettendo di indicare il ritmo con cui i rapporti vengono inviati. Inoltre, essendo in multicast, ogni stazione sa quante stazioni ci sono e quindi quanta banda spetta a ognuno, nel caso in cui l’applicazione dovesse cambiare in meglio o in peggio questo numero, allora anche la banda di RTCP viene cambiata, in questo modo si evita che il protocollo intasi la rete.

#### Session Initiation Protocol (SIP)

SIP è un protocollo di segnalazione il cui compito è quello di raggruppare degli utenti intorno a un’applicazione, quindi si possono creare dinamicamente dei gruppi su richiesta. Ogni utente è indirizzato da nomi o da indirizzi e-mail ed è raggiungibile in qualsiasi punto della rete (anche se roaming) e da qualunque dispositivo utilizzato. I servizi supportati da questo protocollo sono:

* Attivazione della chiamata: vengono forniti tutti i meccanismi per mettere in contatto il chiamante e il chiamato, inizialmente il primo deve contattare il secondo (sa dove trovarlo) e, se accetta, si inviano dei CODEC a esso per scegliere qual è il migliore per entrambi, in seguito inizia la chiamata. Lo stesso meccanismo interviene anche quando la chiamata è finita;
* Reperimento del chiamato: vengono utilizzate reti di supporto in cui sono presenti register e proxy, i primi sono server in cui vengono registrati gli utenti e la loro posizione nella rete mentre i secondi sono un’interfaccia per questi ultimi dato che possono trovarsi anche in reti private. All’avvio della comunicazione si cerca il relativo proxy attraverso il suo IP, esso contatta il register in cui è presente il chiamato e dopo si avvia la comunicazione;
* Gestione della chiamata: si occupa di tutta la parte gestionale, come l’aggiunta di un nuovo utente o media alla conversazione, al cambio di codifica o al trasferimento di chiamate in sospeso.