

Introduzione alle reti

Introduzione

Le reti di calcolatori sono insiemi di dispositivi di calcolo autonomi e interconnessi

A cosa servono?

- servizi di **comunicazione** tra utenti
- **condivisione** di risorse e dispositivi
- accesso a informazioni e risorse remote → ubiquità
- calcolo distribuito → calcoli complessi eseguiti da più macchine che collaborano
- sistemi scalabili → cloud

Tipi di reti

1. RETI SU CHIP
2. BAN → body area network → dispositivi indossabili interconnessi (es. smartwatch, sensori per il diabete...)
3. PAN → usb, bluetooth...
 - WPAN è una PAN wireless
4. LAN → local area nework (prefisso W sempre per wirelss, in caso), reti che coprono un ufficio, laboratorio, edificio...
5. Aggregando più LAN si ottiene una MAN, metropolitan area network (area urbana, rete civica... ALMAWIFI è una MAN)
6. Reti geografiche, WAN (wide area network) → connessioni molto ampie nazionali e internazionali
7. **Internet** → la rete globale composta dall'unione di reti di vario tipo, connesse tra loro e conformi a determinati protocolli

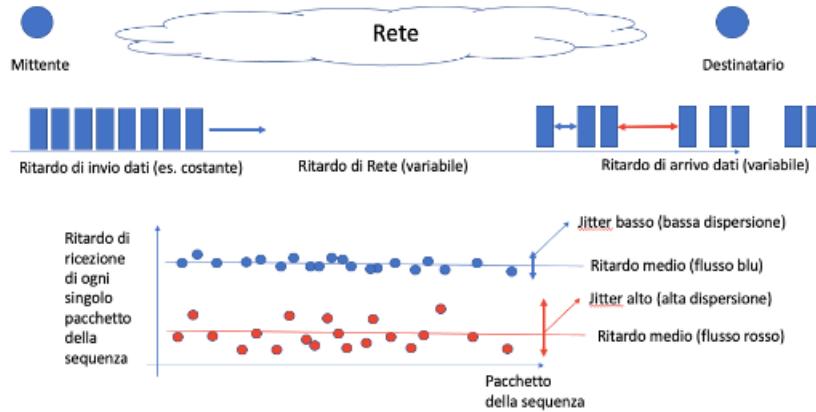
Prestazioni di rete

capacità di trasmissione → numero di bit o byte trasmessi o ricevuti in un secondo

ritardo di collegamento → tempo richiesto ai dati per transitare da mittente a destinatario

- ritardo dovuto all'elaborazione del segnale nei vari nodi di rete → più il segnale va mandato lontano più si accumulano ritardi (deve attraversare più nodi della rete)
- i tempi di gestione dovuti alle leggi dei processi di comunicazione (**protocolli**)

jitter → variazione del ritardo di ricezione dei pacchetti (dispersione attorno al valore medio)



buffer → sistema per ridurre i tempi di latenza, ad esempio, in un'applicazione di streaming. Prima della visualizzazione dell'output si aspetta di ricevere un certo numero di pacchetti per compensare eventuali ritardi una volta che il servizio è avviato (ad esempio per evitare, quanto possibile, interruzioni nel bel mezzo di un film su Netflix)

Componenti di rete

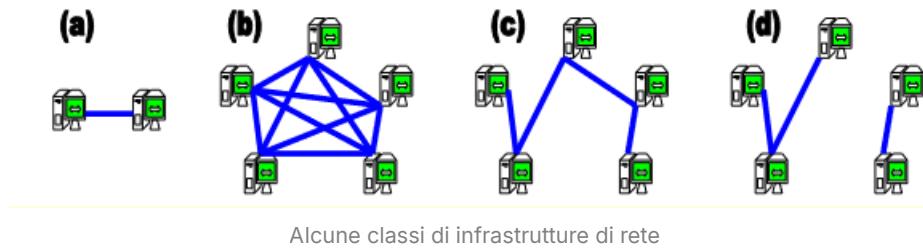
La connessione di un calcolatore a una rete di calcolatori richiede un insieme minimo di componenti, sia hardware che software, in aggiunta al calcolatore elettronico di base:

- **schede di rete**: sono dispositivi hardware per codificare, trasmettere, ricevere e decodificare dati dal calcolatore alla rete, e viceversa → gestite da componenti software del sistema operativo
- **mezzo di trasmissione**: supporto fisico alla propagazione e trasmissione di segnali: es. cavi elettrici, fibre ottiche → collegamento tra mezzo e scheda di rete mediante connettore di rete (componente hardware)
- **protocolli di rete**: insieme di regole implementate sotto forma di software del calcolatore, univocamente definite per garantire compatibilità, e corretta gestione della comunicazione. I protocolli devono essere:
 - conosciuti da tutti
 - non ambigui

Infrastrutture di rete e topologie

Una connessione tra dispositivi avviene tramite la condivisione di un mezzo di trasmissione, che supporta fisicamente la trasmissione del segnale.

Il modo in cui si articola la struttura della connessione (infrastruttura di rete) determina la **topologia** della rete



- connesioni di rete punto a punto → appresentano il caso più semplice di infrastruttura di rete, tra due soli dispositivi
- infrastruttura completamente connessa → è molto ridondante, infatti esistono molti cammini per connettere ogni coppia di nodi, molto complessa e costosa
- infrastruttura connessa → connette più dispositivi minimizzando il costo ma espone a un **single point of failure** e conseguentemente a eventuali formazioni di partizioni di rete
- partizione di rete → grado di comunicazione limitato a classi comunicanti tra loro ma isolate dal resto dei dispositivi, data da cause fisiche o errori protocollari

Topologie di rete

- **Anello**

- distanza massima tra due nodi n/2
- se salta un collegamento una delle due "strade" (senso orario / antiorario) è ancora percorribile
- parallelismo (comunicazione contemporanea tra più dispositivi nella rete) possibile solo parzialmente
- pacchetto speciale chiamato token
 - quando la rete viene avviata viene assegnato il token a un dispositivo, che ha diritto di trasmissione. Dopo che il tempo è scaduto, il token viene passato

- il token è un single point of failure → se viene perso (o attaccato) non è più possibile comunicare → dal punto di vista di sicurezza vince la rete a stella

- **Stella**

- molto scalabile (basta un collegamento in più per introdurre un dispositivo alla rete, ma limitazione fisica del numero di porte del dispositivo centrale)
- presenza di un dispositivo centrale
 - single point of failure → in caso di scollegamento del dispositivo centrale la rete crolla
 - mezzo dedicato per la comunicazione tra dispositivo centrale e dispositivo collegato → ampiamente possibile la comunicazione in parallelo
 - buffering del nodo intermedio → il nodo intermedio gestisce i pacchetti che devono essere inviati tra due nodi periferici
- distanza tra due oggetti periferici costante = 2

- **Bus**

- in voga negli anni 80
- unico mezzo trasmissivo da cui passa un segnale alla volta
- non esiste praticamente più (nel contesto delle reti)

- **Albero**

- architettura del sistema di rete
- si aggiunge la caratteristica peculiare ad ogni dispositivo dell'altezza rispetto all'albero
- si attribuisce più importanza ai nodi più in alto poiché sono un punto di passaggio obbligato se i sottoalberi devono comunicare tra loro
- facile decidere dove indirizzare un pacchetto → il protocollo è un if then else molto semplice

Le reti più piccole si articolano rispettando solitamente le topologie citate, reti più grandi assumono topologie ibride, dette a maglia

Componenti hardware

Il **mezzo fisico** utilizzato per connettere dispositivi nella rete è strettamente correlato al tipo di segnale utilizzato per la trasmissione dei dati

- fili metallici → trasmettono segnali elettrici (corrente e variazioni di tensione)
 - metodo più utilizzato
- fibre ottiche → velocità di trasmissione più veloce per ragioni fisiche
 - capacità e resistenza del cavo che rallentano un segnale elettrico non esistono per un segnale ottico

- si riescono a inviare meno impulsi elettrici rispetto a quelli ottici per unità di tempo
- infrastruttura di rete più costosa
- wireless → radiazione elettromagnetica nello spazio
 - onde radio, onde infrarosse
 - permettono mobilità dei calcolatori nell'ambiente
 - collegamento poco affidabile a causa di interferenze del segnale e prestazioni limitate

Il **dispositivo o scheda di rete** è un componente dell'architettura del calcolatore dotato di un'interfaccia di collegamento al calcolatore e un connettore al mezzo fisico

- la sua funzione è quella di memorizzare temporaneamente, codificare, decodificare, trasmettere e ricevere i dati da e verso il mezzo di trasmissione (cioè la rete) o il calcolatore
- il tipo di scheda di rete dipende dal mezzo utilizzato
- ha un codice identificativo unico, l'**indirizzo MAC**

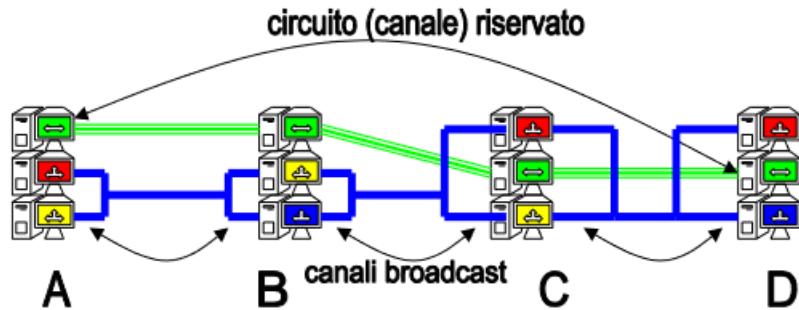
Canali di comunicazione

Un **canale di comunicazione** non è altro che una visione astratta del mezzo di trasmissione. Al di fuori da un canale punto a punto in cui la trasmissione è relativamente banale, in un canale ad accesso multiplo (**broadcast**) bisogna gestire una serie di problematiche:

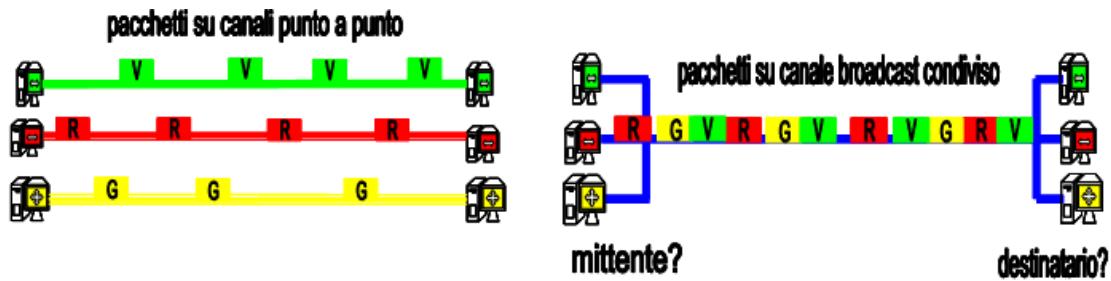
- arbitraggio del mezzo → chi trasmette? quando? Se avvengono **collisioni**, ovvero sovrapposizioni di trasmissioni sul canale, i dati si perdono
- indirizzamento → quale dispositivo è destinatario dei dati trasmessi? Utilizzo di indirizzi univoci per identificare un dispositivo nella rete

Una serie di canali può realizzare un servizio di trasferimento di informazioni tra due dispositivi anche molto distanti tramite due modalità:

- **commutazione di circuito** → viene riservato un circuito di canali di comunicazione punto a punto su ogni connessione lungo tutto il cammino dal mittente al destinatario
 - ritardo di comunicazione basso
 - si paga il tempo di connessione anche se la comunicazione non è in corso → spreco di risorse
 - utilizzato nella linea telefonica



- **commutazione di pacchetto** → utilizzo di un canale condiviso per la trasmissione dei dati
 - i dati vengono suddivisi in pacchetti indipendenti e trasmessi separatamente sul canale
 - ogni pacchetto contiene indirizzo di mittente e destinatario e vari flussi di pacchetti coesistono sul canale
 - i nodi ricevitori di un pacchetto, se non sono quelli di destinazione, inoltrano nuovamente il pacchetto sul broadcast
 - miglior utilizzo delle risorse di rete (canali e connessioni)
 - maggior ritardo della comunicazione
 - si paga per i dati trasmessi
 - utilizzato nella rete internet



Nelle reti a commutazione di pacchetto il servizio di trasmissione dei dati tra mittente e destinatario ha proprietà determinate dal ruolo dei protocolli di rete utilizzati:

- i **servizi connection-oriented** (es. telefono) garantiscono la consegna ordinata dei pacchetti ricevuti e la ritrasmissione dei pacchetti perduti
- nei **servizi connectionless** (come la posta ordinaria) i pacchetti possono seguire strade diverse, arrivare in ordine diverso dalla partenza o non arrivare proprio

Protocolli di rete

Un **protocollo** è:

- un insieme di regole e procedure di gestione dei processi di comunicazione
- una scelta di formati sintattici che definiscono scambi di messaggi non ambigui

Un protocollo determina uno standard che rende i dispositivi tra loro compatibili (fa in modo che possano "cavarsela")

I protocolli utilizzati nel contesto della comunicazione di rete sono molteplici e ognuno di essi cura un determinato aspetto della comunicazione, dal basso all'alto livello

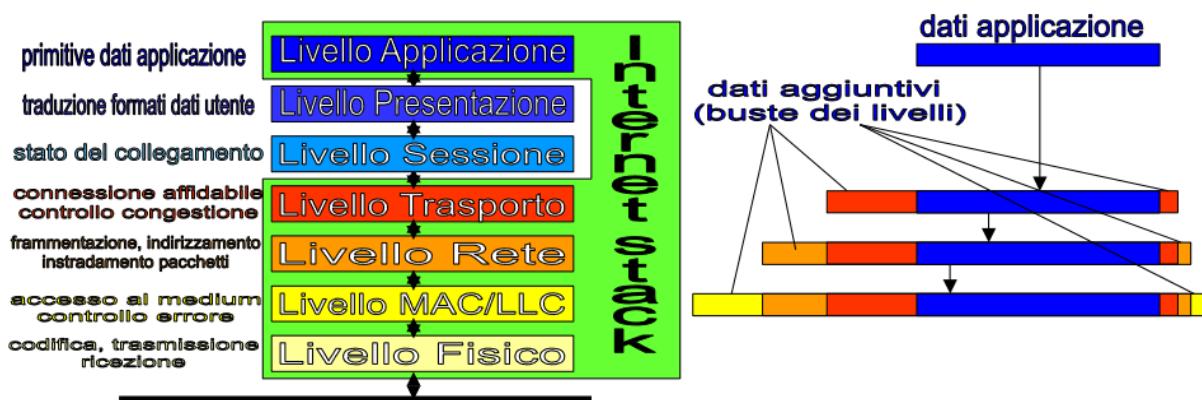
Per questo i protocolli di rete sono suddivisi in livelli di astrazione

- I livelli superiori effettuano richieste di servizio al loro livello inferiore
- I livelli inferiori forniscono servizi al loro livello superiore
- Ogni livello ha relazioni dirette solo con i livelli immediatamente superiore e inferiore, attraverso richieste e servizi concordati, detti interfaccia del livello

Architettura Standard di protocolli di rete

Lo **Standard ISO/OSI RM** è un riferimento standard per definire l'architettura dei protocolli delle reti di calcolatori

- Definisce un'architettura organizzata in 7 livelli, dove ogni livello gestisce una classe ristretta di problematiche della rete
- Dialogo tra livelli paritari avviene astraendo l'architettura sottostante





Approfondimento di tutti i livelli (copiato dalle slide)

Il livello applicazione (7) fornisce alle applicazioni in esecuzione sul calcolatore i servizi e le primitive di trasmissione e ricezione dei dati. Il livello presentazione (6) risolve eventuali eterogeneità del formato dei dati tra i nodi della rete. Il livello sessione (5) mantiene e gestisce lo stato attuale del collegamento tra due applicazioni remote. Il livello trasporto (4) si occupa di garantire i servizi di trasmissione dei pacchetti (orientati alla connessione e non) e del controllo della congestione della rete. Il livello rete (3) si occupa di frammentare i dati in pacchetti, scrivere gli indirizzi dei destinatari finali e instradare i pacchetti verso i destinatari intermedi del cammino. Il livello LLC/MAC (2) si occupa di garantire l'affidabilità del mezzo di trasmissione e la gestione dell'accesso al mezzo trasmissivo ad accesso multiplo (evitando le collisioni). Infine, il livello fisico (1) si occupa di definire le tecniche di codifica dei dati, la trasmissione e la ricezione dei dati sul mezzo fisico di trasmissione

L'architettura dei livelli di protocolli di Internet utilizza solo 5 dei 7 livelli ISO/OSI RM

In trasmissione, ogni livello riceve dati dai livelli superiori e li inserisce (incapsula) in "**buste**" con dati aggiuntivi, utili ad istruire il corrispondente livello del dispositivo ricevente

In ricezione, ogni livello legge i dati della busta, agisce di conseguenza e passa le informazioni restanti al livello sopra (decapsulamento)

- Il livello trasporto spezza i dati dell'applicazione in frammenti e li imbusta, aggiungendo informazioni utili all'ordinamento e al ri-assemblaggio dei dati ricevuti, oltre che al controllo della congestione della rete
- Il livello rete frammenta ulteriormente i dati in pacchetti (se sono troppo lunghi), scrive l'indirizzo del destinatario sulla busta, e decide il cammino sul quale inviare il pacchetto a seconda dell'indirizzo di rete del destinatario.
- Il livello MAC/LLC esegue la consegna finale dei dati a dispositivi di una rete locale

Astrazione nei livelli ISO/OSI

- livello fisico: la rete è solo un segmento = un mezzo di trasmissione condiviso tra dispositivi
 - regole per codificare e trasmettere dati, un'unica tecnologia in comune (data dal mezzo, che sia cavo in rame, fibra ecc...)
- livello MAC/LLC: la rete è locale, può integrare mezzi trasmissivi e tecnologie diverse
 - regole per gli indirizzi dei dispositivi, i tempi di accesso al mezzo, e gestione errori di trasmissione
- livello rete: la rete è una collezione di reti, e assume struttura gerarchica (reti di reti, sottoreti)

- regole per indirizzi di rete che nascondano i dettagli locali (indirizzi che identificano una rete intera, gerarchie tra reti)
- si definiscono nuovi dispositivi (router) che smistano i pacchetti di dati tra rete e rete
- livello trasporto: la rete è una collezione di reti organizzate gerarchicamente → si assume implementato il problema di indirizzamento
 - regole per la spedizione affidabile di pacchetti, e controllo della congestione della rete
- livello applicazione: la rete esiste, funziona, ed è utilizzabile dalle applicazioni dell'utente

Codifica dei dati nel livello fisico

La trasmissione dati sul canale di comunicazione richiede la codifica e decodifica dei dati, uno di seguito all'altro sul mezzo trasmissivo, da parte della scheda di rete

Codifica e decodifica avviene da segnaletica elettromagnetica a bit e viceversa

Tutti i segnali elettromagnetici viaggiano a velocità della luce, la differenza tra una rete veloce e una lenta fa la frequenza di bit trasmessi al secondo sul canale (capacità del canale)

In alcuni mezzi trasmissivi è possibile codificare il segnale più velocemente che in altri, vedi ad esempio il rame contro la fibra ottica

Affidabilità del canale

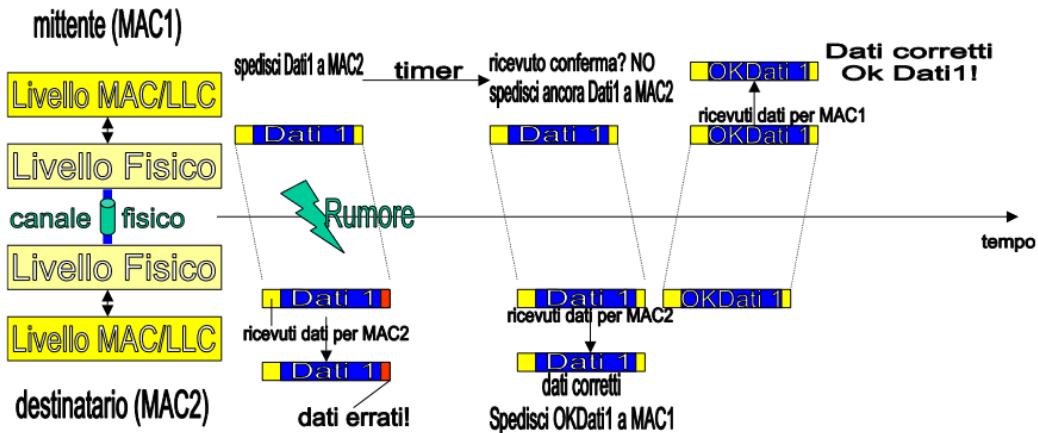
Un **canale affidabile** è un canale che evita errori di trasmissione dovuti a collisioni o interferenza

Come rendiamo una rete locale (quindi consideriamo solo il livello MAC) affidabile?

Il pacchetto (frame) trasmesso sul mezzo trasmissivo è stato ricevuto correttamente?

- Si! Il destinatario invia un frame di conferma della corretta ricezione al mittente (detto **acknowledgement**)
- No! il destinatario non fa nulla
- se il mittente non riceve conferma entro un tempo fissato, trasmette di nuovo il frame

Ai livelli dei protocolli superiori al livello due tutto ciò viene nascosto, e appare solamente la trasmissione corretta (o meno) del frame sul segmento di rete



In un contesto di rete non locale:

- Il livello di trasporto si occupa dell'affidabilità end-to-end, cioè dall'origine fino alla destinazione finale. Questo implica che se un segmento (un'unità di dati del livello di trasporto) non arriva correttamente a destinazione, il livello di trasporto si occuperà della ritrasmissione. Tuttavia, questa operazione può essere lenta perché richiede l'interazione tra i dispositivi finali (end-to-end).
- Il livello di rete decide il percorso che i pacchetti devono seguire attraverso la rete → non si occupa dell'affidabilità, ma solo dell'instradamento.
- Il livello MAC garantisce l'affidabilità della ritrasmissione del pacchetto tra dispositivi (ad es. router) intermedi → i vari destinatari intermedi da cui transita il pacchetto inviano un ack al dispositivo da cui l'hanno ricevuto → questo passaparola continua fino alla destinazione finale del pacchetto

Tecnologie per schede di rete

Esempi dei protocolli di livello 2 più usati:

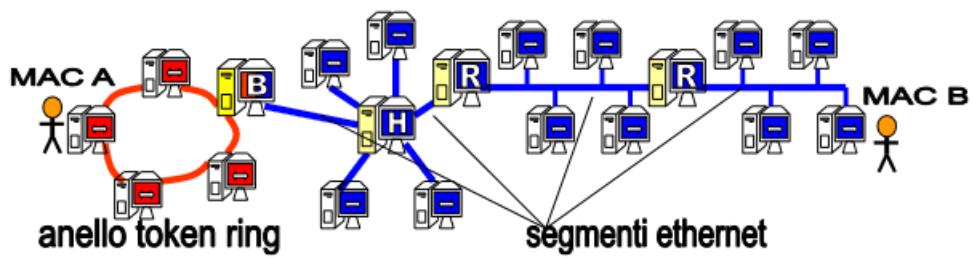
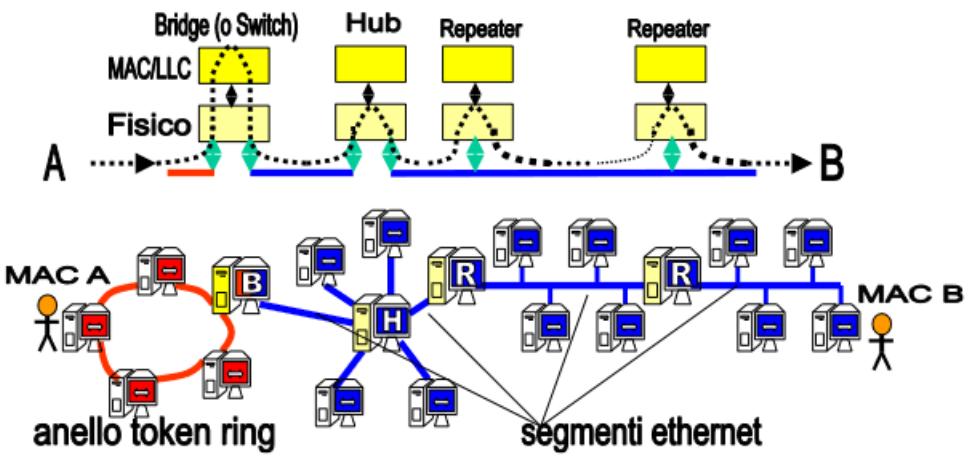
- **Ethernet**
 - Molto usato in reti locali cablato
 - La scheda di rete ascolta il canale e trasmette solo se nessuno sta già trasmettendo
 - Se viene rilevata una collisione la trasmissione è interrotta per riprovare più tardi (attesa di tempo casuale)
- **IEEE 802.11 (Wi-Fi)**
 - È alla base delle reti locali senza fili
 - La scheda di rete ascolta il canale e trasmette solo se nessuno sta già trasmettendo
 - si cerca di prevenire le collisioni (collision avoidance) dilazionando le trasmissioni nel tempo
- **Token ring**

- usata se i dispositivi sono collocati su topologia ad anello cablato
- Esiste un frame detto token che viene passato come un "testimone" tra i dispositivi
- Solo chi detiene il token ha diritto di trasmettere, poi deve passare il token dopo una prefissata unità di tempo

Comporre reti locali

Dispositivi utili a comporre reti locali di calcolatori, agendo al livello fisico (1) e MAC/LLC (2):

- **Repeater** (ripetitore): livello fisico (1)
 - I segnali trasmessi sul mezzo fisico degradano con la distanza → un repeater è un dispositivo che amplifica e rigenera il segnale ricevuto
 - collega due o più segmenti di rete (con stessa tecnologia MAC)
 - estende la lunghezza dei segmenti di rete locale
- **Hub** (perno di una ruota a raggi): livello fisico (1) (detto anche repeater multiporta), è il dispositivo centrale della topologia a stella
- **Bridge** (ponte): livello MAC/LLC (2)
 - connette segmenti di una rete locale con tecnologie MAC diverse (Es. connette un segmento Ethernet a un segmento Token Ring)
 - Traduce i frame ricevuti da un segmento nel formato frame dell'altro segmento
 - Ri-trasmette il frame tradotto usando il protocollo MAC opportuno
- **Switch** (commutatore): livello MAC/LLC (2)
 - Analogico al bridge, ma permette di connettere molti segmenti
 - Ha capacità di filtrare e inviare i frame sul segmento giusto, leggendo l'indirizzo MAC del destinatario del frame



Internetworking (TCP/IP)

Introduzione

Se i milioni di calcolatori oggi connessi a Internet fossero tutti organizzati secondo i protocolli e gli schemi visti finora per le reti locali, la comunicazione tra due calcolatori su Internet richiederebbe di passare per migliaia di calcolatori intermedi, switch, bridge, segmenti di rete, ognuno dei quali aggiungerebbe ritardi di gestione, complessità, rischi di errore

Il problema dell'instradamento dei frame (**routing**) come lo risolviamo allora?

- teniamo una lista di tutti gli indirizzi MAC dei dispositivi nel mondo, con a fianco l'indicazione della direzione di inoltro? → Complessissimo e limitazione critica alla scalabilità di Internet
- possibile soluzione: si elegge un rappresentante, che chiamiamo **router**, che è un nodo di rete che ha il compito di ricevere tutti i pacchetti indirizzati a dispositivi della sua rete locale, per poi recapitarli all'interno della rete, come se si trattasse di un frame a livello MAC/LLC destinato all'indirizzo MAC del destinatario
 - Allo stesso modo, ogni router dovrebbe farsi carico di inoltrare tutti i pacchetti uscenti dalla propria rete locale, verso i router delle reti di destinazione

Il livello che si occupa dell'instradamento dei pacchetti nello standard ISO/OSI è il livello di rete. I router comunicano tra loro attraverso collegamenti dati molto veloci, dette **backbone** (dorsali di rete)

Ogni router deve ricordare in una tabella di instradamento (**forwarding table**) solo quale sia il primo router intermedio per raggiungere ogni altro router (tendenzialmente si hanno i figli, ovvero la sottorete, il gateway, ovvero il router padre, ed eventuali fratelli)

Protocollo IP

Il protocollo IP introduce nel livello rete un nuovo tipo di indirizzamento globale e gerarchico, implementando un tipo di connessione connectionless

ATTENZIONE: IP è connectionless, ma collabora con protocolli connection-oriented nel contesto dello standard ISO/OSI

I router, amministratori del livello di rete, contengono tabelle di instradamento che illustrano la topologia della rete nascondendo dettagli interni delle LAN

Nella busta di ogni pacchetto da inviare che transita a livello rete viene aggiunto **indirizzo IP** di mittente e destinatario

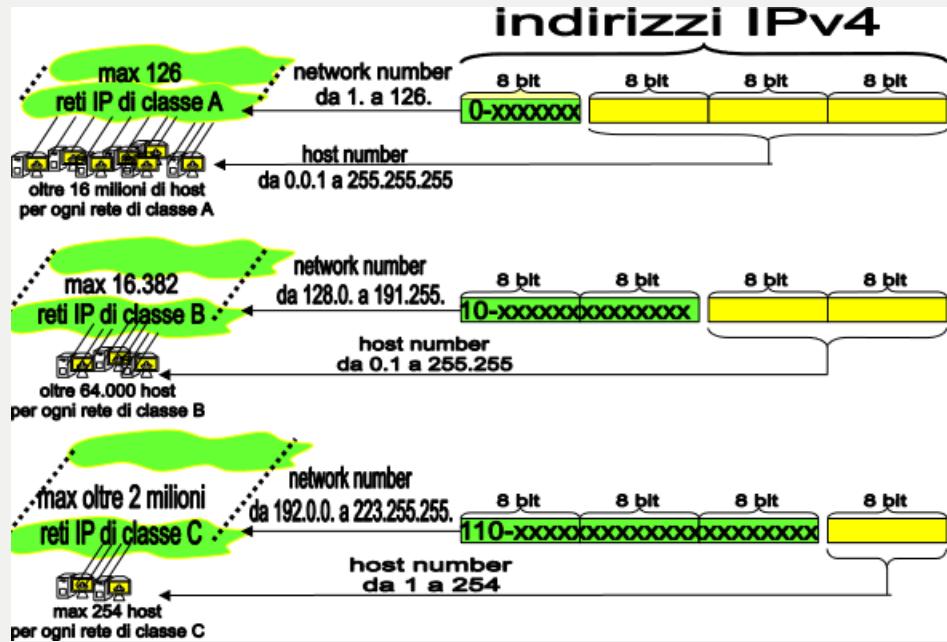
Indirizzamento IPv4

Gli indirizzi IP sono un nuovo tipo di indirizzo

- in ogni dispositivo vi è un indirizzo IP per ogni scheda di rete → associazione univoca tra indirizzo MAC e indirizzo IP
- l'IP può essere statico o dinamico
- Un **indirizzo IPv4** (quarta versione del protocollo IP) è fatto in questo modo:
 - 32 bit (4 Byte) = sequenza di 4 valori decimali separati da punto
 - ogni valore decimale può essere compreso tra i valori 0 e 255
 - è suddiviso in due parti, il **network number** che si riferisce a numero della rete a cui la scheda appartiene, e l'**host number** che identifica il dispositivo all'interno della rete



Classi IPv4



- Per le reti di **classe A**, il byte di indirizzo più significativo (a sinistra) ha sempre il primo bit uguale a zero, e può assumere i valori da 1 a 126 (network number) rispetto ai 128 valori possibili. I tre byte rimanenti possono assumere oltre 16 milioni di combinazioni, ognuna associabile a un host della rete
- Le reti di **classe B** sono al massimo 16.382 e ognuna può contenere fino a oltre 64.000 host. Per le reti di classe B, il network number è dato dai due byte di indirizzo più significativi (a sinistra), che hanno sempre i primi due bit uguali alla coppia 10-. I network number di classe B possono assumere i valori da 128.0. a 191.255. I due byte rimanenti (host number) possono assumere oltre 64.000 combinazioni, ognuna associabile a un host della rete
- Le reti di **classe C** sono oltre 2 milioni, e ognuna può contenere fino a 254 host. Per le reti di classe C, i tre byte di indirizzo più significativi (a sinistra) rappresentano il network number, e hanno sempre i primi tre bit uguali alla terna 110-. I network number di classe C possono assumere i valori da 192.0.0 a 223.255.255. Il byte rimanente (host number) può assumere 254 combinazioni utili, su 256 possibili, ognuna associabile a un host della rete

Sottoreti

Introduciamo ora il concetto di **sottorete**, vedendo come si comporta IPv4 nel caso in cui abbiamo dei router come host di altri router

Per implementare il concetto di sottorete in IPv4 ci servono nell'indirizzo dei bit in più relativi alla rete rispetto al network number che abbiamo già spiegato

Viene utilizzato uno strumento chiamato **subnet mask** (o netmask, maschera di rete/sottorete) che è una sequenza di bit così composta:

- bit uguali a uno nella subnet mask → indirizzo di sottorete (subnetwork)
- bit uguali a zero nella subnet mask → host number dell'host appartenente alla sottorete

I bit che identificano la sottorete sono i bit che non appartengono al network number ma che vengono indicati come bit di sottorete dalla subnet mask

Il router che amministra una determinata sottorete viene detto **default router**



Esempio: data la rete di classe B 130.136. per semplicità decidiamo di considerare possibili 256 sottoreti: netmask 255.255.255.0. Il numero della sottorete è quindi fornito dai primi tre byte dell'indirizzo IP, es. 130.136.1. è la sottorete 1, 130.136.2. è la sottorete 2, mentre ad esempio 130.136.1.22 è l'host 22 della sottorete 1, 130.136.3.48 è l'host 48 della sottorete 3, e così via

Esempio esplicativo sul forwarding di pacchetti

Host 140.217.2.10 spedisce pacchetto IP a host 130.136.2.33

- raccolto dal default router della sottorete Ry2: 140.217.2.254
 - destinatario non appartiene alla sottorete: inoltrato verso il default router di livello superiore: 140.217.0.254
- raccolto dal default router 140.217.0.254
 - destinatario 130.136.-.: tabella di forwarding indica di inviare a 190.89.0.254: inoltrato
- raccolto da 190.89.0.254
 - dest. 130.136.-.: tabella di forwarding indica di inviare a 130.136.0.254: inoltrato
- raccolto da 130.136.0.254
 - dest. 130.136.-.: la tabella di forwarding indica che appartiene a questa rete (k)
 - inoltrato al router 130.136.2.254
 - raccolto dal router 130.136.2.254
 - il dest. appartiene alla sottorete Rk2: 130.136.2. : inoltrato
 - raccolto dall'host destinatario finale 130.136.2.33: OK

Esercizio svolto - classi di reti, super-reti e sottoreti

indirizzo IPv4	IPv4 valido?	host o rete?	Classe? A,B,C?	numero sottoreti?
a) 99.99.99.99/7	sì	host	A	super-rete (99 e 98)
b) 11.111.1.11/9	sì	host	A	2 subnet
c) 123.123.123.321/8	no	–	–	–
d) 222.222.22.192/26	sì	rete	C	4 subnet
e) 101.0.0.101/16	sì	host	A	256 subnet
f) 210.210.210.120/29	sì	rete	C	32 subnet
g) 1.1.1.1/1	sì	(host)	A	super-rete (0-127)
h) 130.136.256.254/18	no	–	–	–
i) 192.0.1.0/16	sì	host	C	super-rete(256)
j) 191.0.0.0/16	sì	rete	B	–

Tabelle di forwarding - problemi

Come aggiorni le tabelle di forwarding dei router se un nodo si collega o scollega dalla rete (o per guasti di mezzi trasmissivi, interruzione delle linee, guasti di router, nuove politiche e accordi per lo scambio dei dati tra gestori di dorsali e sistemi autonomi.....)

- le modifiche dei cammini rendono sbagliate le tabelle di forwarding → rischio di perdita pacchetti
- protocolli di routing
 - invio di richiesta → si chiede ai nodi vicini se sanno come raggiungere un certo nodo nella rete
 - aggiornamento della tabella

Protocollo ICMP

Internet Control Message Protocol (ICMP) è un protocollo di messaggi di controllo su Internet

- Uno standard per definire la comunicazione di informazioni utili alla gestione di Internet
- ICMP è usato da host, router e gateway per scambiare informazioni di livello rete, usando dei pacchetti

Esempio di errori notificabili tramite ICMP:

- Rete di destinazione non raggiungibile (possibile interruzione di rete?)
- Rete di destinazione sconosciuta (indirizzo di rete male specificato?)
- Host destinazione non raggiungibile (host spento o scollegato?)
- Host destinazione sconosciuto (indirizzo di host male specificato?)
- Protocollo richiesto non disponibile (servizi non previsti)
- Ricerca di un cammino alternativo per la destinazione (se esiste)

es. di applicazioni basate su ICMP: ping e traceroute

ARP e RARP

Quando un router riceve un pacchetto destinato a un indirizzo IP della propria sottorete, deve produrre un frame di livello MAC/LLC che riporti specificato l'indirizzo MAC del destinatario (e non l'indirizzo IP), per poterlo passare al livello MAC/LLC per la trasmissione sulla rete locale

→ Protocollo Address Resolution Protocol (**ARP**)

Usato se il router non conosce l'indirizzo MAC corrispondente all'indirizzo IP

- Il router genera un frame spedito a tutti i dispositivi della rete locale con l'indirizzo IP di cui vuole sapere il MAC
- Se tale dispositivo esiste, riceve il frame e risponde con un frame di livello MAC indirizzato al router, nel quale viene evidenziato l'indirizzo MAC richiesto

Protocollo Reverse-ARP (**RARP**) → È la versione opposta del protocollo ARP, ma funziona in modo analogo

Ovviamente le associazioni trovate sono cachate dentro al router, ARP e RARP sono utilizzati sporadicamente

DHCP

L'assegnazione dei numeri di rete (classi A, B e C) viene effettuata da enti internazionali verso altri enti, organizzazioni ed aziende

L'assegnazione dei numeri di host in una rete invece avvengono manualmente tramite un amministratore di rete (associazione statica IP - MAC) o automaticamente da un **server DHCP** (Dynamic Host Configuration Protocol)

- l'assegnamento è dinamico
- usato in reti wireless, reti locali e connessioni domestiche
- il server DHCP in sé per sé è un host che implementa il servizio di assegnazione dell'indirizzo IP agli host che ne fanno richiesta

- dispone di un blocco di host number liberi per la sua rete con cui fare l'associazione

IPv6

IPv6 è una versione più avanzata (ma non largamente utilizzata quanto IPv4) del protocollo IP:

- indirizzi IP estesi a 128 bit (16 Byte) anziché i 32 bit (4 Byte) di IPv4
- ridefiniti i campi che costituiscono la busta dei pacchetti di livello IPv4, aggiungendo ad esempio parametri per la gestione di flussi di pacchetti IP con diversi livelli di priorità
- tecnica del **tunnelling** dei pacchetti IPv6 in IPv4 → si spediscono i pacchetti IPv6 in buste IPv4 perché siano compatibili con i router IPv4

Livello trasporto

Il livello di trasporto deve garantire:

- integrità dei dati → alcune applicazioni necessitano di un'affidabilità totale sui dati trasmessi (es. per transazioni web), altri possono tollerare delle perdite di dati
- timing → alcune applicazioni necessitano più di altre di un delay limitato per funzionare correttamente (es. gaming)
- throughput → necessario un throughput minimo per il funzionamento di determinati servizi (es. contenuti multimediali in streaming)
- sicurezza → necessaria la crittografia dei dati, la loro integrità, ecc...

I protocolli di Internet di quarto livello (trasporto) sono il Transmission Control Protocol (**TCP**) e lo User Data Protocol (**UDP**)

application	data loss	throughput	time sensitive
file transfer	no loss	elastic	no
e-mail	no loss	elastic	no
Web documents	no loss	elastic	no
real-time audio/video	loss-tolerant	audio: 5kbps-1Mbps video:10kbps-5Mbps	yes, 100's msec
stored audio/video	loss-tolerant	same as above	
interactive games	loss-tolerant	few kbps up	yes, few secs
text messaging	no loss	elastic	yes, 100's msec yes and no

- Protocollo TCP:
 - servizio affidabile → connection oriented (completa IP rendendo la rete affidabile)
 - configurazione e utilizzo di un **numero di porta**
 - numerazione sequenziale dei pacchetti, riordino (i pacchetti sono numerati), eliminazione duplicati
 - conferma di ricezione mediante pacchetto speciale di **acknowledgement**
 - rinvio di pacchetti non ricevuti dal destinatario
 - gestione della congestione e del flusso
- Protocollo UDP:
 - servizio non affidabile → connectionless → vengono inviati i pacchetti senza stabilire una connessione (se nessuno dei pacchetti arriva perché l'IP destinatario è errato, ad esempio, il mittente non lo saprà mai)
 - latenza nettamente inferiore in quanto è assente la procedura di apertura della connessione, l'acknowledgment e controlli di flusso e congestione

application	application layer protocol	underlying transport protocol
e-mail	SMTP [RFC 2821]	TCP
remote terminal access	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
file transfer	FTP [RFC 959]	TCP
streaming multimedia	HTTP (e.g., YouTube), RTP [RFC 1889]	TCP or UDP
Internet telephony	SIP, RTP, proprietary (e.g., Skype)	TCP or UDP

TCP

TCP si basa sul principio di connessione punto a punto tra punti virtuali detti **socket** (IP + numero di porta) e consente lo smistamento dei vari pacchetti verso le rispettive applicazioni in ascolto sulle porte

Esempio di **handshake** del protocollo TCP (inizializzazione della connessione)

- PC1 (client) invia richiesta TCP di connessione sul socket del PC2 (server)
- Se il socket esiste e non è occupato, TCP di PC2 risponde ok!
- Se riceve l'ok da PC2, TCP di PC1 può inviare i dati di configurazione
- Ora avviene lo scambio dati veri e propri a livello TCP
- Quando la connessione termina si liberano le porte utilizzate

NOTA: tendenzialmente le porte dei server sono standardizzate, ci sono delle well-known-port su cui vengono inviate le richieste di connessione TCP di modo che arrivino già sulla porta relativa all'applicazione corretta

Controllo di flusso e congestione di rete

Come gestisco la quantità di dati nel tempo da inviare a un host per evitare sia di impiegare troppo tempo che di sovraccaricare la rete (router intermedi, il mezzo di base non si sovraccarica mai) o saturare il destinatario?

- **controllo di flusso:** inviare pacchetti al massimo ritmo sostenibile dal destinatario finale

- **controllo di congestione:** inviare pacchetti al massimo ritmo sostenibile dal router più lento della rete dal mittente al destinatario finale

I meccanismi di controllo sono implementati da una **finestra scorrevole** (sliding window, SW) che è un valore intero che rappresenta il numero massimo di pacchetti che il mittente può spedire di seguito in attesa di conferma

Obiettivo: spedire al massimo ritmo possibile

- controllo di flusso: non spedisce più di SW pacchetti oltre l'ultimo non confermato
 - se un pacchetto non viene confermato lo rispedisce, prima di spedire i successivi SW → se il pacchetto 80 non arriva ma ho una SW di 20 pacchetti, non me la gioco coi 19 "posti" restanti per arrivare al pacchetto 200 MA me ne preoccupo prima e interrompo l'invio di pacchetti per tempo finché quel pacchetto non viene inviato correttamente
- controllo della congestione: spedisce un numero SW variabile di pacchetti
 - se SW pacchetti spediti sono tutti confermati, aumenta SW
 - appena un pacchetto degli SW inviati non viene confermato (scade il timeout) lo rinvia e riparte da SW minimo

NOTA: la sliding window è la medesima, che varia a seconda di flusso e congestione

Rendere TCP sicuro

TCP e UDP non implementano di default alcun modo per criptare i messaggi

SSL è un protocollo che fornisce strumenti per criptare una connessione TCP, garantire integrità dei dati (saper distinguere se un dato è stato alterato prima di raggiungere il destinatario) e autenticazione del destinatario

Consiste in uno strato di protezione del socket basato su un handshake tra due dispositivi che concordano su come proteggere i dati

Si trova nel livello applicazione, non nel livello di trasporto, di modo che i pacchetti vi arrivino già criptati

Configurazione di rete e sicurezza

Come configuriamo un host domestico?

- connesso attraverso il servizio di connessione fornito da un Internet Service Provider (ISP)
- il punto di accesso alla rete è un **modem** → è un dispositivo di rete che trasmette i bit a un altro modem attraverso la linea telefonica

- Per il collegamento dell'host occorre quindi: installare il modem e applicare al modem i protocolli PPP per il livello LLC (per stabilire una connessione del dispositivo con gli altri nodi della rete, e il protocollo IP per la rete)
- si configurano quindi indirizzo IP dell'host (automaticamente tramite DHCP o manualmente), maschera di rete, Indirizzo IP del default router, e indirizzo IP del DNS server
- Possono poi essere inseriti gli indirizzi IP di servizi applicativi come posta elettronica (server SMTP, POP3 o IMAP)
- Al rilascio della comunicazione, automaticamente l'host viene cancellato dalla sottorete dell'ISP e l'indirizzo IP eventualmente riciclato per altri dispositivi

Sicurezza

Per garantire la sicurezza di una rete bisogna prestare attenzione ai seguenti aspetti:

- Dotarsi degli strumenti per prevenire e contrastare la diffusione di programmi dannosi che attraverso le reti possono diffondersi (all'interno di posta elettronica, documenti scaricati...)
- Prevenzione e contrasto dell'accesso indiscriminato ai sistemi di rete privati → occorre bloccare l'accesso ai dati e alle applicazioni di intrusi
 - vengono filtrati i pacchetti di dati a livello rete, attraverso dei router speciali detti **firewall**.
 - i firewall sono posti di solito come il primo router che i pacchetti incontrano dall'esterno entrando nella rete privata (ma possono essere anche semplicemente dei filtri software)
 - Per decidere chi può attraversare il firewall vengono create delle liste di persone autorizzate e registrate o si utilizza una verifica dell'identità basata su autenticazione
- Un **application gateway** è un server (processo server o dispositivo fisico) che verifica tutte le applicazioni rischiose e filtra il traffico basato su di esse, consentendone l'uso solo da parte delle persone autorizzate
- L'ultimo aspetto da considerare è la segretezza (privacy) dei dati trasmessi in rete (che tutti potrebbero intercettare) → le soluzioni in uso si basano su tecniche di crittografia e cifratura dei dati

Livello applicazione

Introduzione

Il livello applicazione implementa primitive e protocolli per spedire e ricevere dati delle applicazioni, appoggiandosi fortemente al protocollo TCP mediante un'adeguata interfaccia (socket)

Inoltre:

- deve garantire il funzionamento degli applicativi su tutti i tipi di dispositivi
- deve essere costruito in modo da non obbligare il programmatore a scrivere un'applicazione network-core perché possa funzionare correttamente in rete, ma permettergli di sfruttare interfacce intuitive per concentrarsi sullo sviluppo dell'applicazione

Architetture di livello applicazione

Le applicazioni e i servizi su Internet possono essere realizzati secondo due modalità architetturali

- **Architettura Client/Server** → i client spediscono richieste di servizio, i server sono host sui quali sono in esecuzione i servizi che soddisfano le richieste (es. servizio DNS, servizio World Wide Web, servizio posta elettronica)
 - i server hanno solitamente IP permanenti e sono degli host always-on per garantire la continuità del servizio
 - i client hanno solitamente IP dinamici e si collegano in maniera intermittente ai server, nel momento in cui hanno bisogno di un determinato servizio
 - salvo architetture peer to peer (vedi sotto) in un'architettura client server la comunicazione tra due client è mediata da un server
- **Architettura Peer to Peer** (P2P) → tutti gli host sono contemporaneamente sia client che server
 - ogni host cerca di soddisfare le richieste ricevute e agisce da client quando spedisce ad altri host delle richieste (es. alcuni servizi di file-sharing)
 - non ci sono dispositivi always-on, i peers si collegano in maniera intermittente alla rete così come i client, cambiando indirizzo IP → gestione complessa
 - scalabilità → nuovi peers aumentano sia la capacità del servizio che la sua domanda

I processi a livello applicazione si distinguono in **processi client**, che inizializzano la comunicazione, e **processi server**, che aspettano di essere contattati

- tali processi comunicano scambiandosi messaggi
- in P2P abbiamo entrambi i tipi di processi in uno stesso host

I processi mandano o ricevono messaggi dai propri socket (numero di porta utilizzato dal processo, IP del dispositivo)

Protocolli di livello applicazione

I protocolli di livello applicazione gestiscono l'invio e la ricezione di richieste e risposte, definendo i campi dei vari messaggi e il significato delle informazioni che è necessario includere

Inoltre, viene definito quando e come i messaggi devono essere inviati per coordinare la comunicazione tra i dispositivi

I protocolli possono essere:

- **aperti** → standardizzati, documentati e pubblicamente disponibili, permettono interoperabilità tra diverse implementazioni e sistemi (es. HTTP e SMTP)
- **proprietari** → sviluppati e controllati da specifiche aziende o entità e non standardizzati ma relegati a uno specifico servizio proprietario

Cenni di HTTP

Una pagina web è composta da un insieme di oggetti (file HTML, immagini JPEG, audio...) → file HTML di base che include riferimenti a una moltitudine di oggetti, ognuno indirizzato da un URL

Il **protocollo HTTP** è un protocollo applicazione che si basa sul modello client/server

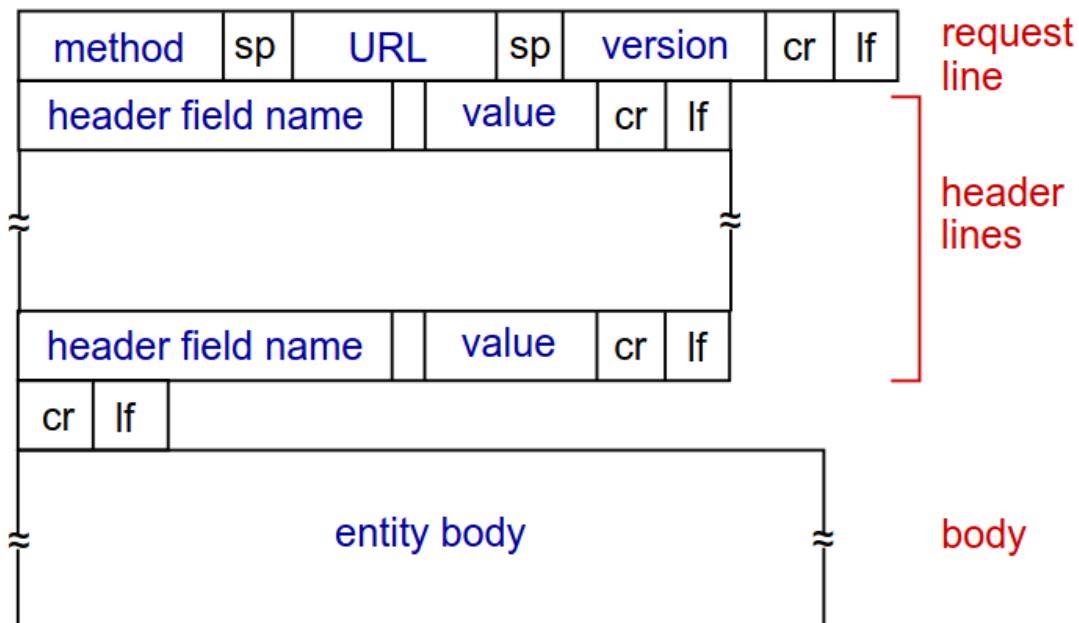
- il client è rappresentato da un **browser**, che richiede, riceve e mostra gli oggetti web
- il **server web** invia gli oggetti in risposta alla richiesta

Il protocollo HTTP sfrutta TCP a livello supporto, inviando richieste e risposte da socket a socket

HTTP è stateless → il server non mantiene nel tempo alcuna informazione riguardo alle richieste del client

Le connessioni HTTP possono essere:

- **non-persistenti:** massimo un oggetto inviato per connessione TCP, poi la sessione viene chiusa → scaricare oggetti multipli richiede connessioni multiple
 - tempo di risposta 2RTT (round trip time, tempo di viaggio di un pacchetto tra client e server e ritorno) + tempo di trasferimento del file
 - carico ingente del sistema operativo che deve aprire e chiudere una moltitudine di connessioni → situazione migliorata se si lavora in parallelo
- **persistenti:** possono essere inviati molteplici oggetti su una stessa connessione TCP
 - le connessioni rimangono aperte dopo la risposta del server
 - se il client non chiude la connessione dopo la ricezione (chiusa comunque entro un timeout o per limite di risorse) la connessione rimane pendente → problema di carico
 - tempo 1 RTT per ogni risorsa trasferita dopo la connessione iniziale (1 RTT)



Formato di una richiesta HTTP

Il **metodo** di una richiesta HTTP è sostanzialmente il tipo della richiesta che si va ad inviare (es. GET, POST...)

Il formato di una response è leggermente diverso e mostra uno **status code** che contiene l'esito della richiesta, ad esempio 404 (Not Found)

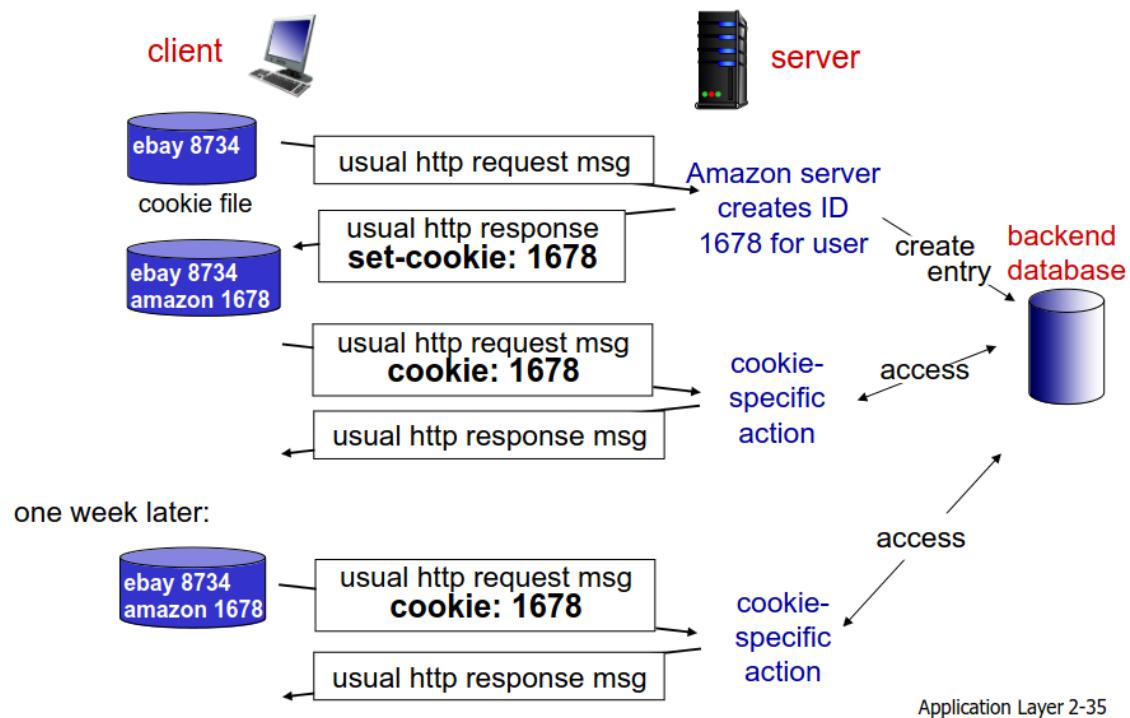
Cookies

I **cookies** sono dei pacchetti speciali utilizzati dal server per riconoscere un client per salvare delle informazioni riguardo ad esso o agire in un certo modo specifico a una sua richiesta → è un modo per rendere HTTP stateful nonostante di base sia stateless

I cookies vengono salvati a lato utente nel browser

- autorizzazioni
- carrelli
- stato della sessione dell'utente
- ...

I cookies permettono di fatto di fare profilazione → occhio alla privacy



Proxy server

I **server proxy** come una cache a lato utente per non interfacciarsi all'origin server a ogni richiesta

- il browser manda tutte le richieste al proxy
 - se la richiesta può essere soddisfatta con i dati già contenuti nel proxy (e sono aggiornati), risponde direttamente il proxy tagliando notevolmente i tempi di richiesta e risposta

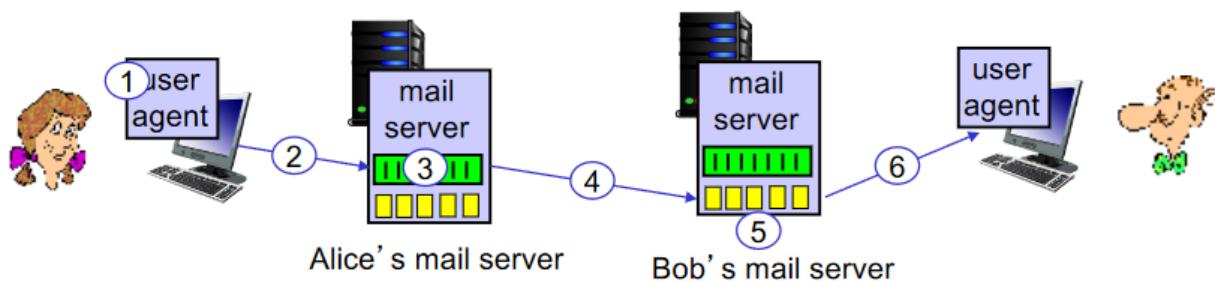
- se i dati sono mancanti o non aggiornati, il proxy richiede i dati all'origin server, li salva e manda la risposta al client
- il proxy solitamente è locale e utilizzato in contesti specifici (es. università, aziende...)
- oltre ad accorciare i tempi, il proxy riduce il traffico di rete all'access point dell'ente che lo utilizza

Vedere slide per esempio di utilizzo del proxy (omesso perché abbastanza intuitivo)

Protocolli per la posta elettronica

Un servizio di posta elettronica si basa su tre elementi cardine:

- **user agent** → l'applicativo che offre il servizio di posta elettronica a lato utente
 - permette di editare e leggere mail, nonché di riceverle
- **mail server** → il server che contiene fisicamente le mail come dati
 - contiene la mailbox e la lista di messaggi in uscita
- **SMTP (Simple Mail Transfer Protocol)** → il protocollo che gestisce il tutto
 - sfrutta TCP per l'affidabilità → prevedere un handshake tra i due server, il trasferimento e la chiusura della comunicazione
 - implementa un trasferimento diretto dal server del mittente al server del destinatario
 - implementa una comunicazione composta da comandi e risposte (come HTTP) che permette un invio di messaggi multiparti, ovvero composti da oggetti multipli
 - richiede che i messaggi siano in 7-bit ASCII



```

S: 220 hamburger.edu
C: HELO crepes.fr
S: 250 Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection

```

SMTP, si noti la prima parte di handshake, il client che tramite dei comandi definisce mittente, destinatario e corpo e le varie risposte del server

Altri protocolli:

- **POP** → implementa anche un sistema di autenticazione e di download delle mail sul client, stateless tra le sessioni
- **IMAP** → gestione più capillare dei dati sul server (es. cartelle), mantiene lo stato dell'utente tra le sessioni

DNS

Risulta più vantaggioso in rete riferirsi a indirizzi IP con nomi mnemonici e gerarchici piuttosto che con indirizzi IP, ma i protocolli necessitano di indirizzi IP per operare

La soluzione è il servizio Domain Name System (**DNS**)

- Basato su una catena di server chiamati **DNS server** organizzati gerarchicamente
 - Ogni host in rete deve conoscere almeno un DNS server (il suo IP)
 - Ogni server DNS conosce almeno un DNS server superiore (tranne i DNS radice)

I DNS server forniscono al client l'indirizzo IP relativo al nome richiesto → se un server DNS interrogato non conosce la risposta, demanda la richiesta a un server superiore

I **root DNS** conoscono tutte le associazioni IP-nome, sono pochi al mondo e molto costosi e sono in cima alla gerarchia dei DNS server

Sotto vi sono i **TLD servers** (Top Level Domain) che sono responsabili dei domini com, org, net, it....

Ancora al di sotto vi sono gli **authoritative DNS servers**, di proprietà di varie organizzazioni

Infine, i **local DNS name server**, che non appartengono direttamente alla gerarchia, sono dei server di riferimento per gli host

- quando un host fa una richiesta essa è inoltrata al local DNS, che funziona come una sorta di proxy interfacciandosi alla gerarchia e utilizzando una cache per velocizzare la risoluzione delle richieste

Inoltre, DNS permette di:

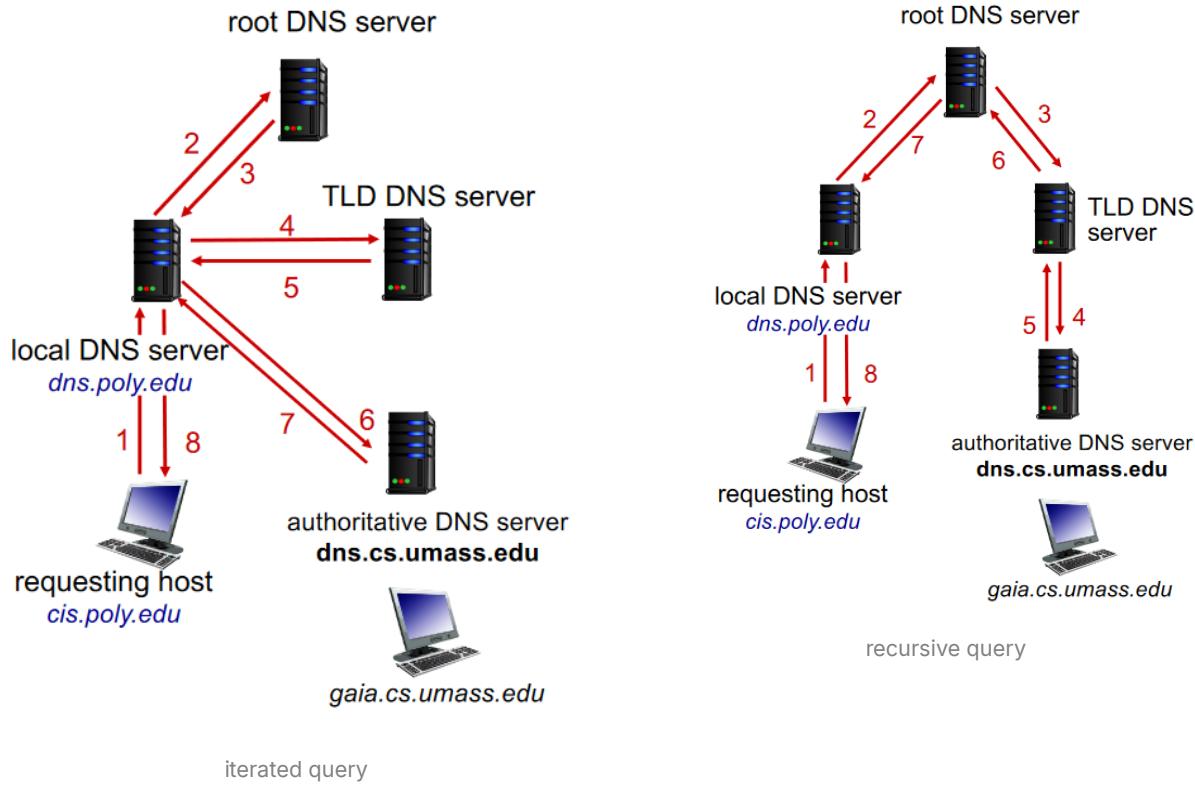
- riferirsi a un solo identificatore per un database o un servizio distribuito su più macchine fisiche → **load distribution**
- utilizzare diversi nomi per riferirsi allo stesso indirizzo IP → **host aliasing**

NOTA: i nomi non sono duplicabili all'interno del dominio stesso

NOTA2: DNS sfrutta principalmente UDP per le richieste, in caso di necessità (pacchetti di risposta troppo grandi, fallimento della richiesta) si può ricorrere a TCP

Risoluzione dei DNS name

- **iterated query** → il server contattato risponde col nome del server da contattare successivamente (passando dal local DNS)
- **recursive query** → demanda al server contattato il proseguimento della risoluzione



Ogni server (non solo local) sfrutta un sistema di cache per ricordare quali dispositivi stanno al di sotto di lui → aggiornato durante le risoluzioni, i dati relativi a una entry (un'istanza IP + name) vengono cancellati dopo un certo TTL (time to leave)

- non è garantito che i dati su un server DNS siano aggiornati → se un host cambia indirizzo IP, potrebbe non essere conosciuto il nuovo indirizzo associato al nome finché non scadono tutti i TTL

Record DNS e messaggi

De facto il DNS è un database distribuito → le entry si chiamano **record di risorse** (RR), ve ne sono varie tipologie

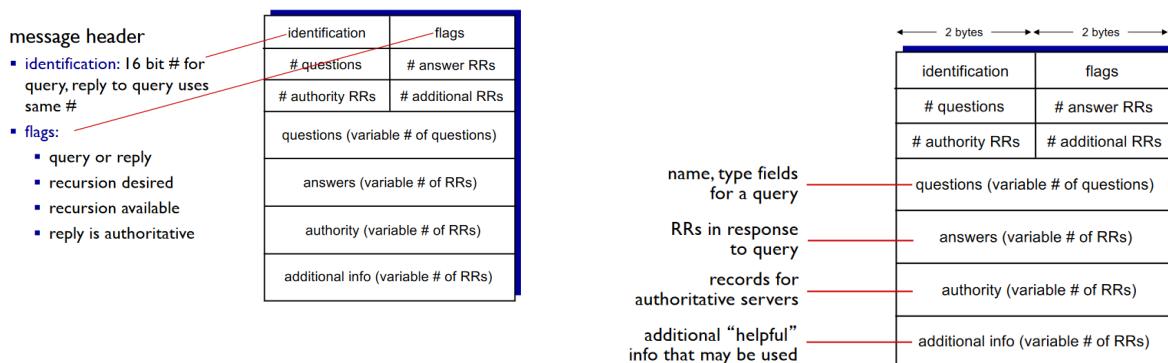
RR format (struttura base di un RR generico):

- **name:** Nome dell'host o dominio.
- **value:** Valore associato al nome.

- **type:** Tipo di record.
- **TTL:** Time to Live, ovvero il tempo per cui il record è valido nella cache dei resolver DNS.

1. **type = A (Address Record)** → utilizzato per mappare un nome di dominio a un indirizzo IP (classico)
 - name: Nome dell'host
 - value: Indirizzo IP dell'host
2. **type = NS (Name Server Record)** → utilizzato per specificare i server DNS autoritativi per un dominio → sfruttato durante la risoluzione
 - name: Nome del dominio (es. foo.com)
 - value: Nome dell'host del server autoritativo per il dominio
3. **type = CNAME (Canonical Name Record)** → utilizzato per mappare un nome di dominio a un altro nome di dominio (ad esempio, www.ibm.com è realmente servereast.backup2.ibm.com).
 - name: Nome alias per un nome "canonico" (il nome reale)
 - value: Nome canonico
4. **type = MX (Mail Exchange Record)** → utilizzato per specificare i server di posta elettronica per un dominio (es. outlook.com)
 - name: Nome del dominio
 - value: Nome del server di posta associato al dominio

Struttura di query e reply DNS:



Esempio di inserimento record

Il **registrar DNS** (organizzazione che gestisce la registrazione dei nomi di dominio) inserisce due RR nel server del TLD .com per il dominio *networkuptopia.com*:

1. Un record NS (Name Server) che collega il dominio *networkuptopia.com* al nome del server DNS autoritativo (nel RR: networkuptopia.com, dns1.networkuptopia.com, NS)
2. Un record A (Address) che collega il nome del server DNS autoritativo all'indirizzo IP corrispondente (nel RR: dns1.networkuptopia.com, 212.212.212.1, A)

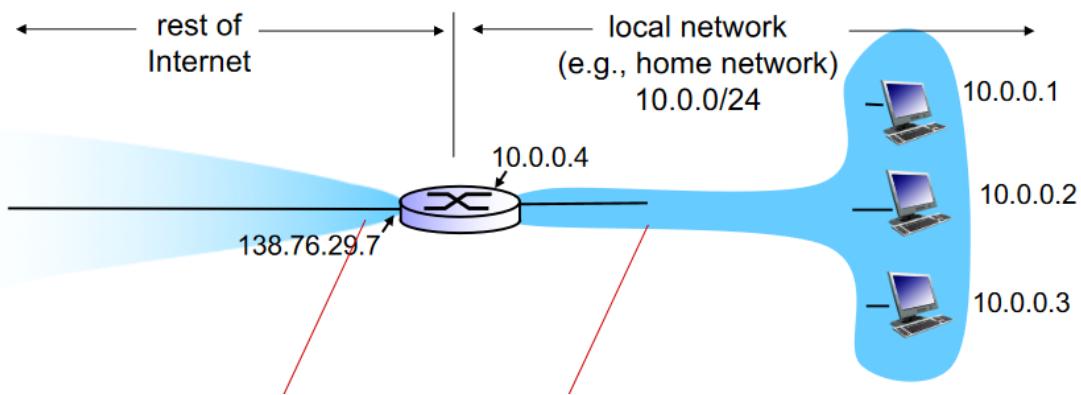
Il registrar DNS configura quindi i propri server DNS autoritativi, creando quindi un record di tipo A in uno di tali server che mappa www.networkuptopia.com a un indirizzo IP specifico

Infine, crea anche un record di tipo MX (Mail Exchange) per networkuptopia.com, che specifica il server di posta associato al dominio, sempre sullo stesso server autoritativo

Attacchi DNS

- **DDoS (Distributed Denial of Service)** → si tratta di bombardare un server DNS di richieste inutili da multipli dispositivi per rendere inaccessibile il server al resto del traffico in rete
 - conseguente inaccessibilità al sito in quanto la risoluzione non è possibile e ad eventuali servizi di posta elettronica o altre risorse
 - rischio di esposizione ad altri attacchi
 - a causa del sistema di caching che vuole evitare accessi troppo frequenti ai livelli alti della gerarchia, sono più efficienti attacchi a TLD server o authoritative DNS piuttosto che ai root
- **Redirect attacks**
 - [man-in-the-middle](#) → intercettazione di richieste
 - [DNS poisoning](#) (o cache poisoning) → sostituzione in un RR dell'effettivo IP relativo a un dominio con uno a discrezione dell'attaccante
- **DDoS tramite amplificazione DNS** → invio di richieste DNS con un indirizzo IP sorgente falsificato (spoofed)
 - facendo sembrare che le richieste provengano dall'IP segnalato, le risposte DNS che sono spesso molto più grandi delle richieste inviate, generano un traffico ingente che colpisce la vittima

NAT



Il **NAT (Network Address Translation)** è un protocollo che permette di utilizzare solo un indirizzo IP per identificare una sola rete locale, e disambiguare i vari host della rete mediante la rispettiva porta sul router

- a un provider basta fornire un solo indirizzo IP
- gli indirizzi IP nella rete locale possono essere riassegnati all'interno della rete senza che per il resto di Internet cambi qualcosa
- è possibile cambiare provider senza dover cambiare gli indirizzi dei dispositivi nella rete
- maggior sicurezza perché i dispositivi nella rete non sono indirizzabili direttamente

Implementazione

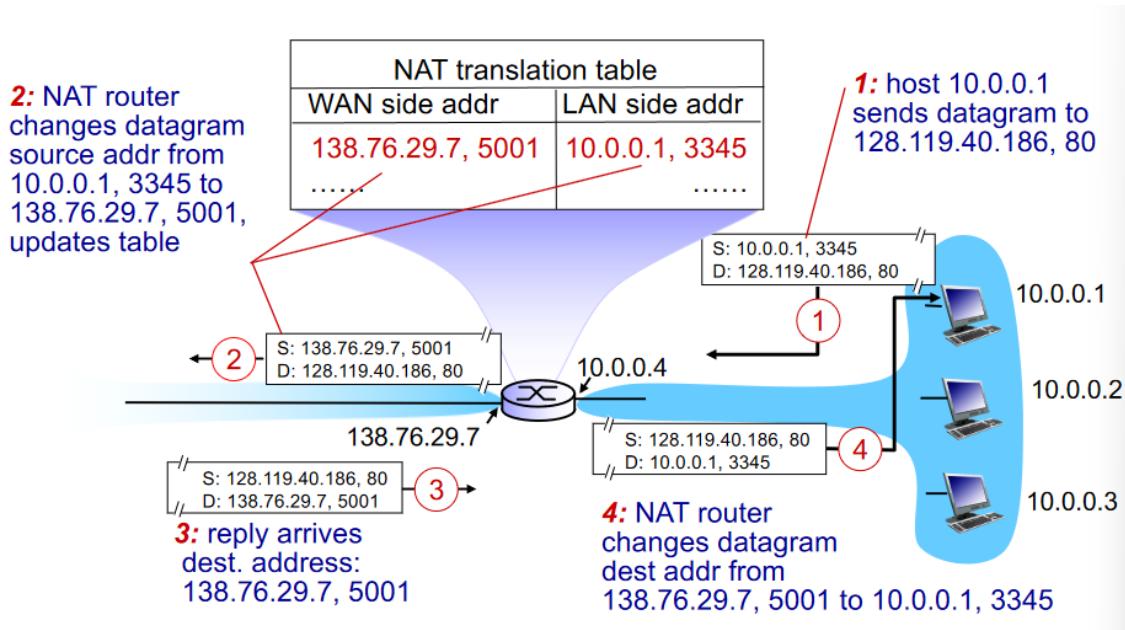
Il router NAT contiene una **NAT translation table** in cui ad ogni entry corrisponde un socket (numero di porta + IP) di un host della rete

NAT router IP, numero di porta del router	indirizzo dell'host, porta dell'host
---	--------------------------------------

All'esterno della rete si usa la prima colonna, all'interno la seconda

→ Il port number è espresso in un campo a 16-bit → 60000 connessioni simultanee possibili

NOTA: il numero di porta del router è di fatto un numero a caso, serve solamente per disambiguare tra i vari IP con relative porte presenti nella rete locale



Esempio

Controversie

1. I router manipolano i dati uscendo di fatto dall'astrazione ISO/OSI → il livello di rete dovrebbe occuparsi solo di routing
2. IPv6 è sotto vari punti di vista una soluzione migliore per la scarsità di indirizzi IPv4 rispetto al NAT
3. Viola il concetto di comunicazione end-to-end, poiché vi sono funzionalità di gestione del pacchetto mentre è in transito
4. Quando un client desidera connettersi a un server situato dietro un dispositivo NAT è necessario utilizzare tecniche specifiche, così come nel P2P, poiché il NAT crea una sorta di barriera che oscura tutto ciò che è oltre al NAT router, ed è quindi designato soprattutto per indirizzi non routabili su internet

Network security

Introduzione

Vedremo quali sono i requisiti di sicurezza, i protocolli e i principi alla base di una comunicazione sicura

Algoritmi di sicurezza

- Conversione da un formato comprensibile (audio, immagine, testuale) a una sequenza di bit pseudocasuale
- Basati su aspetti matematici: applicazione "agevole" in una direzione, applicazione "praticamente impossibile" nel senso opposto → **asimmetria**
- Algoritmo non segreto, deve essere fruibile → cosa garantisce quindi la sicurezza? Una **chiave**

Brute force attack → tentare tutte le combinazioni possibili della chiave per bypassare la sicurezza

- Una chiave più lunga risolve il problema? I calcolatori sono comunque estremamente efficienti e lavorando in parallelo i tempi possono drasticamente diminuire
- La sicurezza non esiste → chi crede che la sicurezza sia ottenibile in modo assoluto sbaglia → studieremo la probabilità che un algoritmo di sicurezza possa essere craccato

A livello pratico, la maggior parte degli attacchi con successo sono sul fattore umano, ovvero inducono una persona dietro al sistema a commettere l'errore

Principi della network security

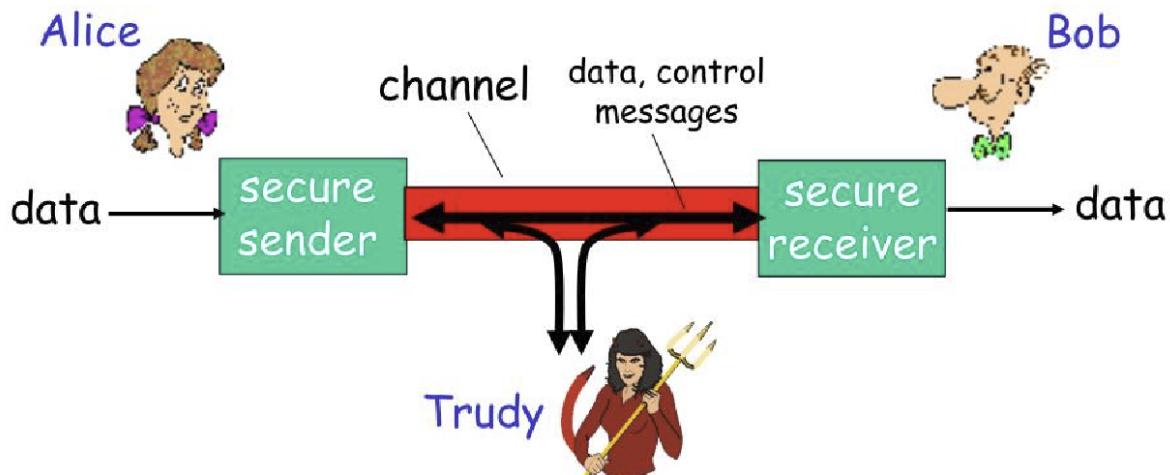
- **crittografia**
 - il mittente critpa il messaggio
 - il ricevente lo decripta
 - problema di comunicazione della chiave
- **autenticazione** (es. identità digitale, face id, password...)
 - "sipario di internet" → necessario assicurare l'identità dell'interlocutore

- **message integrity**
 - impedire la modifica dei dati in transito (o nel caso accorgersene)
- **accesso e continuità** del servizio di rete

IMPORTANTE: gli attacchi possono essere finalizzati anche al danneggiamento dei dati, all'interruzione delle comunicazione, ecc... non per vantaggi dell'attaccante ma fini a sé stessi → i sistemi di sicurezza devono sempre tenere in considerazione questo aspetto



SSL (Secure Socket Layer) → cifratura di un messaggio da socket a socket



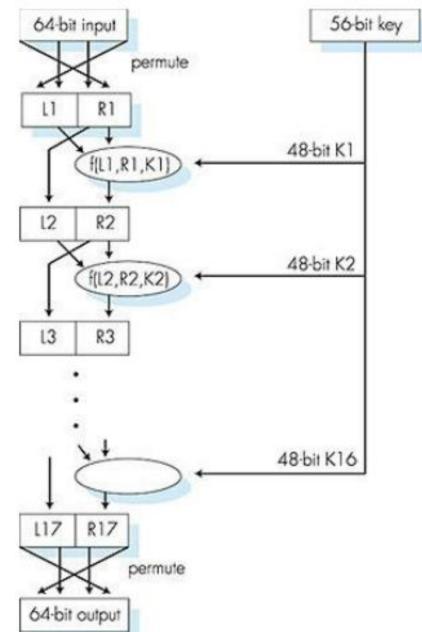
Crittografia

Per rompere un sistema di crittazione, a partire dal testo cifrato:

- brute force attack: prova di tutte le chiavi possibili
- **analisi statistica**
 - known-plain text attack (Turing ed Enigma, "heil Hitler") → Trudy ha il testo originale corrispondente a una parte del testo cifrato, e può determinare delle associazioni tra caratteri in chiaro e caratteri cifrati
 - chosen-plaintext attack → Trudy può cifrare messaggi a sua scelta

Crittografia a chiave simmetrica

- Alice e Bob condividono la stessa chiave (**chiave simmetrica**). Dato il messaggio m e la chiave K_S :
 - Alice calcola $K_S(m)$ e lo invia sul canale di comunicazione
 - Bob decifra il codice usando la medesima chiave $\rightarrow K_S(K_S(m)) = m$
- es. cifrario di Cesare, cifrario di Vigenere
- Esiste qualche protocollo che sfrutta la crittografia a chiave simmetrica? \rightarrow DES, 1993
 - chiave simmetrica a 56 bit, dati suddivisi in blocchi da 64 bit
 - oggi si buca tramite brute force attack in meno di un giorno
 - Funziona utilizzando una permutazione iniziale e finale dei dati e 16 applicazioni della chiave utilizzando 48 dei 56 bit, ogni volta diversi



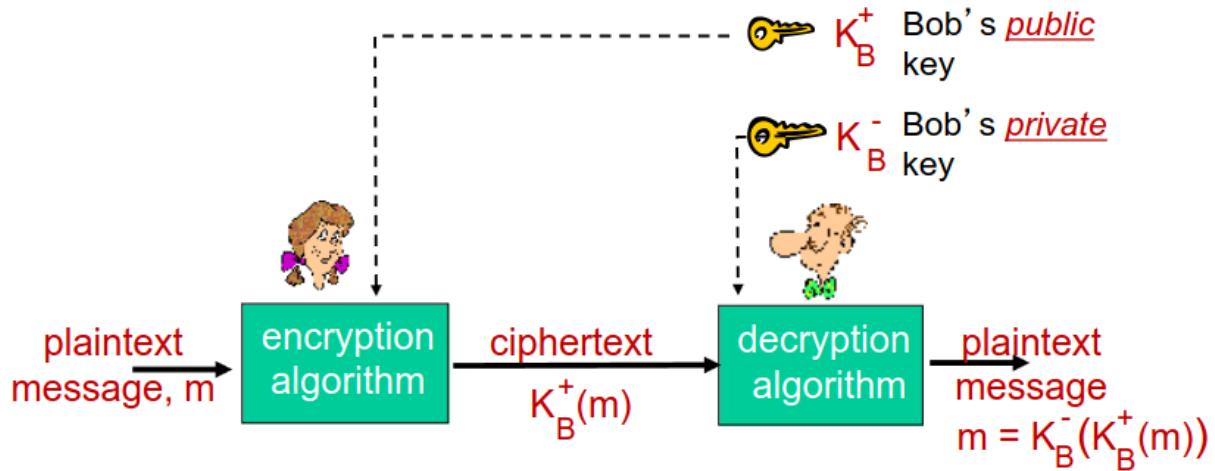
AES - Advanced Encryption Standard

NIST, 2001

- blocchi di dati di 128 bit
- chiavi da 128, 192 o 256 bit
- dato un calcolatore che buca DES in 1 secondo, ci metterebbe 149 trilioni di anni per AES

Crittografia a chiave asimmetrica

Detta anche crittografia a **chiave pubblica** \rightarrow ci sono due chiavi, quella pubblica che tutti possono conoscere, e quella privata, che non viene mai condivisa e che conosce solo chi la genera



Il requisito per questo sistema di crittografia è che conoscendo la chiave pubblica sia impossibile conoscere anche la privata

RSA

Algoritmo di chiave pubblica che sfrutta l'aritmetica modulare



In una comunicazione di rete i messaggi sono sempre codificati, qualsiasi cosa siano, e sono di conseguenza rappresentabili da un numero → criptare un messaggio equivale quindi a criptare un numero

Creare una coppia di chiavi (pubblica e privata):

1. si scelgono due numeri primi molto grandi p e q (es. 1024 bit l'uno)
2. $n = pq \quad z = (p - 1)(q - 1)$
3. si sceglie un tale $e < n$ tale che e e z sono coprimi
4. si sceglie d tale che $ed - 1$ è esattamente divisibile per z
5. $K_N^+ = (n, e) \quad K_B^- = (n, d)$

Criptare e decriptare:

$$1. c = m^e \text{ mod } n$$

$$2. m = c^d \text{ mod } n$$

Criptare con la chiave pubblica e decriptare con quella privata è equivalente a criptare con la privata e decriptare con la pubblica

RSA è sicuro perché data la chiave pubblica (n, e) è molto difficile trovare la privata → bisognerebbe scomporre n in fattori primi senza conoscere p e q , ma n è un numero molto grande

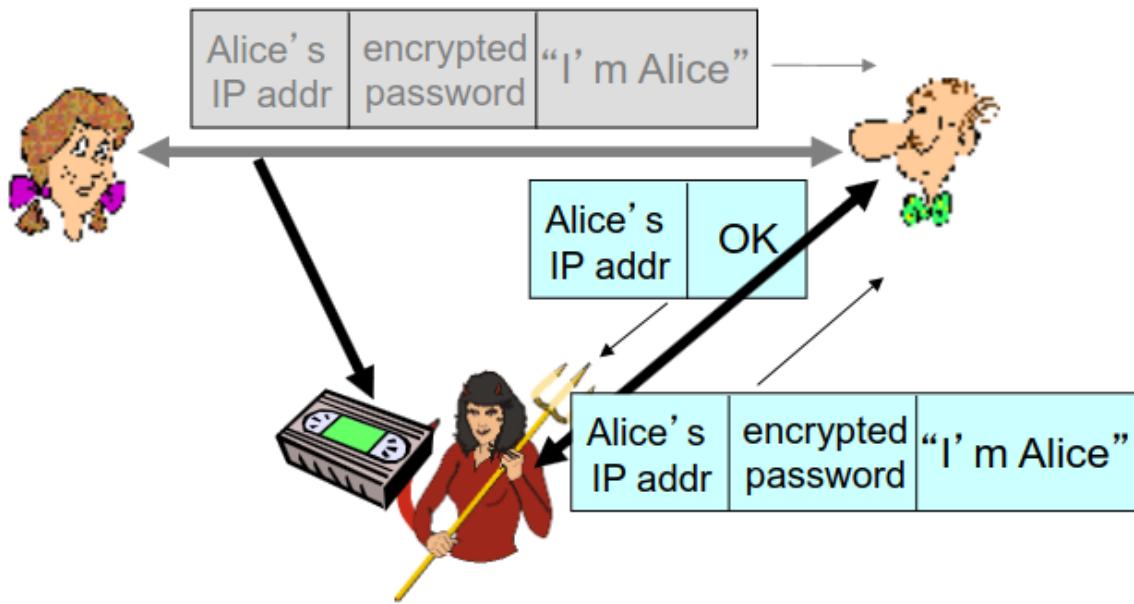
Approfondimenti di calcolo sulle slide

Criptare e decriptare un messaggio con RSA è dispendioso → si utilizza molto nel contesto dello scambio di chiavi simmetriche, e si usano poi quelle per criptare e decriptare i messaggi

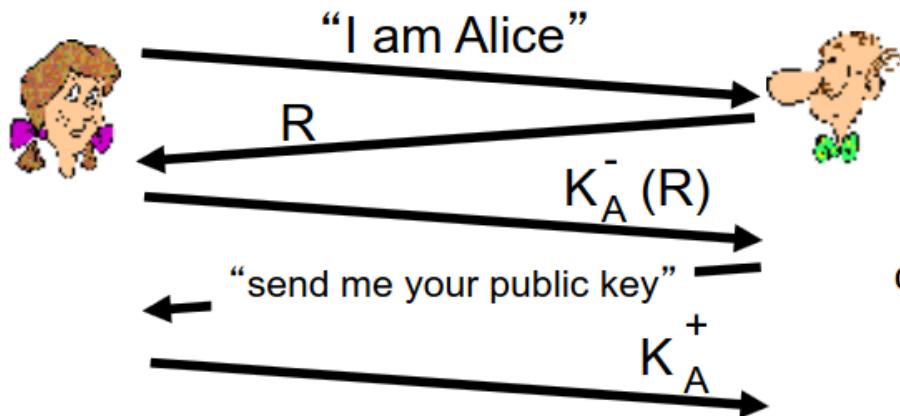
Autenticazione

Il problema dell'autenticazione è che non si conosce la vera natura del mittente (non lo vediamo) → Trudy può dichiarare di essere Alice

- Se Alice dovesse mandare un pacchetto col suo IP per autenticarsi (Bob conosce l'IP di Alice quindi può verificare la veridicità), sarebbe sufficiente?
 - Trudy può creare un pacchetto modificando l'IP, se conosce l'IP di Alice
- Aggiungiamo anche una password segreta per autenticare la connessione che conosce solo Bob
 - Se Trudy "ascolta" il canale, intercetta il pacchetto di Alice e invia a Bob il pacchetto al posto di Alice, riesce ad eludere questo sistema, che la password sia criptata o meno

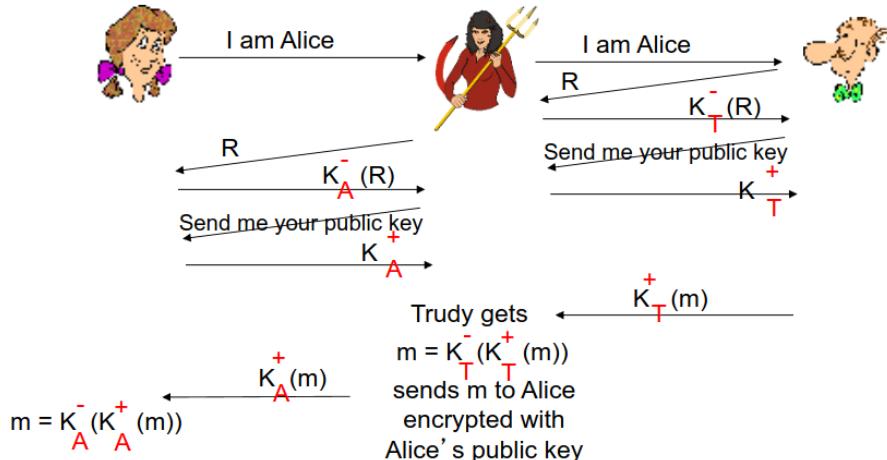


- Dobbiamo evitare questo tipo di intercettazione e rinvio, detto **playback attack**
- Usiamo una sorta di OTP (**NONCE**), ovvero un numero che usiamo solo per autenticare questa comunicazione, che chiamiamo R → Bob manda R ad Alice, che deve criptare (chiave simmetrica o asimmetrica) R e rispedirlo a Bob → se Bob decrittando il messaggio riconosce il numero che lui ha spedito all'inizio, sta davvero parlando con Alice
- Il NONCE può anche essere generato direttamente da Alice → quel NONCE deve essere ricevuto da Bob una volta sola



Versione con chiave asimmetrica

- vulnerabile a un attacco di tipo **man in the middle** → si risolve tramite certification authority, senza certificati questo attacco è sempre possibile



Firma digitale

Il sender di un messaggio (o di un documento) stabilisce di esserne il proprietario/creatore

L'integrità di un messaggio deve essere

- verificabile → il ricevente può provare che Bob ha firmato il documento
- non forgiabile → non deve essere eludibile il sistema di verifica

Un semplice esempio di firma digitale si ottiene utilizzando la crittografia a chiave asimmetrica

- Bob critta il messaggio con la sua chiave privata e invia la chiave pubblica associata
- Alice verifica che il messaggio è stato inviato da Bob decriptando il messaggio con la chiave pubblica → se funziona è dimostrato che il messaggio è stato firmato con la chiave privata di Bob

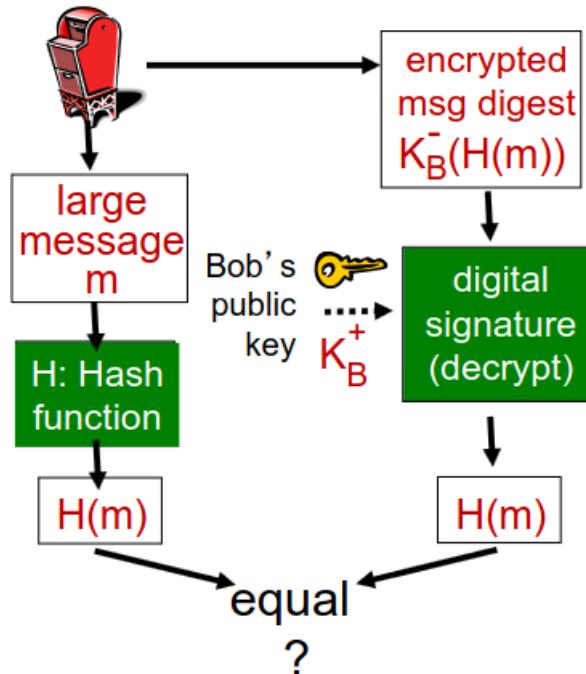
Message digest

Come garantiamo l'integrità del messaggio?

Utilizziamo delle hash function che siano facilmente calcolabili in una direzione, e quasi impossibili nell'altra, tali che $H(m)$ è di lunghezza fissata → tali funzioni si dicono **message digest**

- many-to-one

Per capire se il messaggio è integro confrontiamo $H(m)$ ricevuto con $H(m)$ calcolato localmente



Esempi:

- MD5
- SHA-1

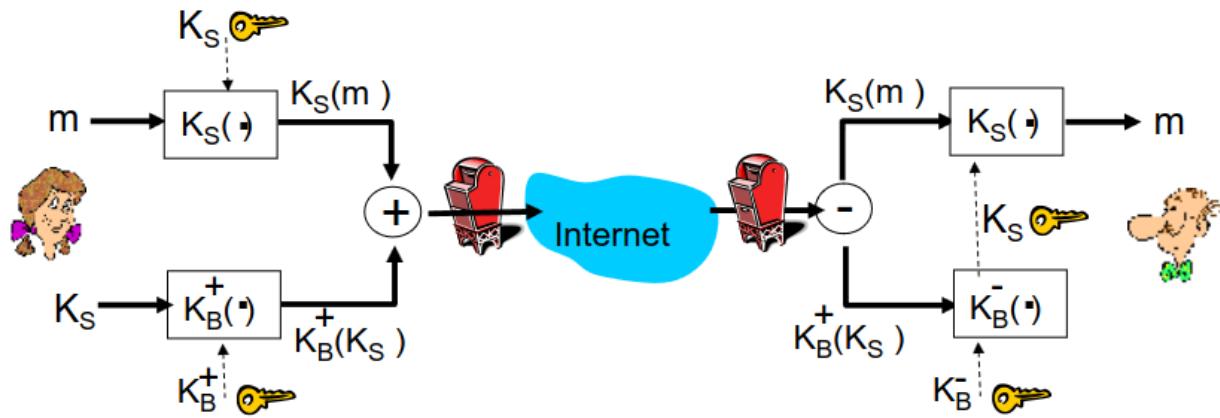
Certification authorities

Una **certification authority (CA)** lega una chiave pubblica a una particolare entità E

- E (router, persona...) registra la sua chiave pubblica con una certification authority
- CA certifica il legame tra E e la sua chiave pubblica criptando la chiave pubblica di E con la sua chiave privata → la chiave pubblica di CA è divulgata → chiunque può verificare che la chiave di E è stata firmata dalla CA e che quindi il legame è attendibile
- Le CA sono degli enti fidati, che non nascono dal niente → sono organizzate a piramide e si certificano vicendevolmente
- I certificati di sicurezza possono scadere perché tenere la stessa chiave per troppo tempo espone a brute force (per quanto la loro riuscita sarebbe decisamente fortuita)

Applicazione: secure e-mail

Confidenzialità

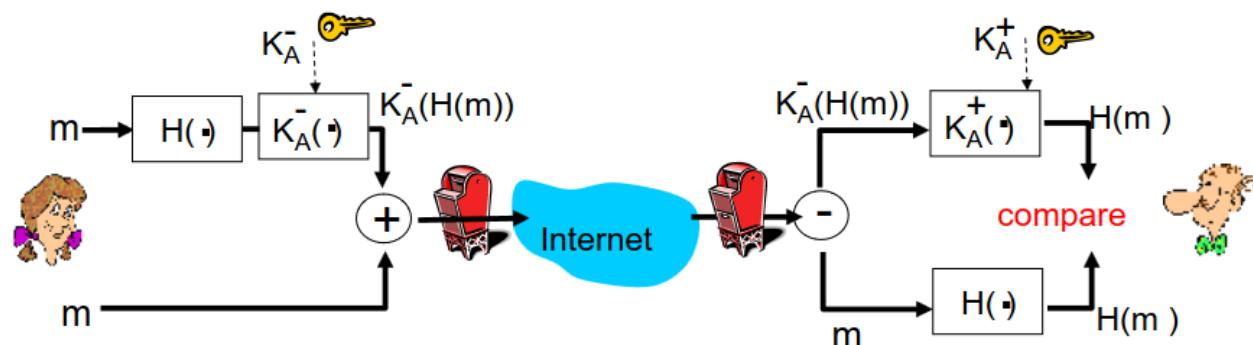


1. Alice ottiene da una CA la chiave pubblica di Bob
 - critta la chiave simmetrica
 - la invia a Bob che la decripta con la sua chiave privata
2. Alice critta il messaggio con la chiave simmetrica
 - invia il messaggio
 - il messaggio viene decriptato da Bob che a questo punto ha la chiave

Le due procedure possono avvenire contemporaneamente

Così è garantita la confidenzialità ma non l'integrità e la garanzia del mittente

Integrità e autenticazione

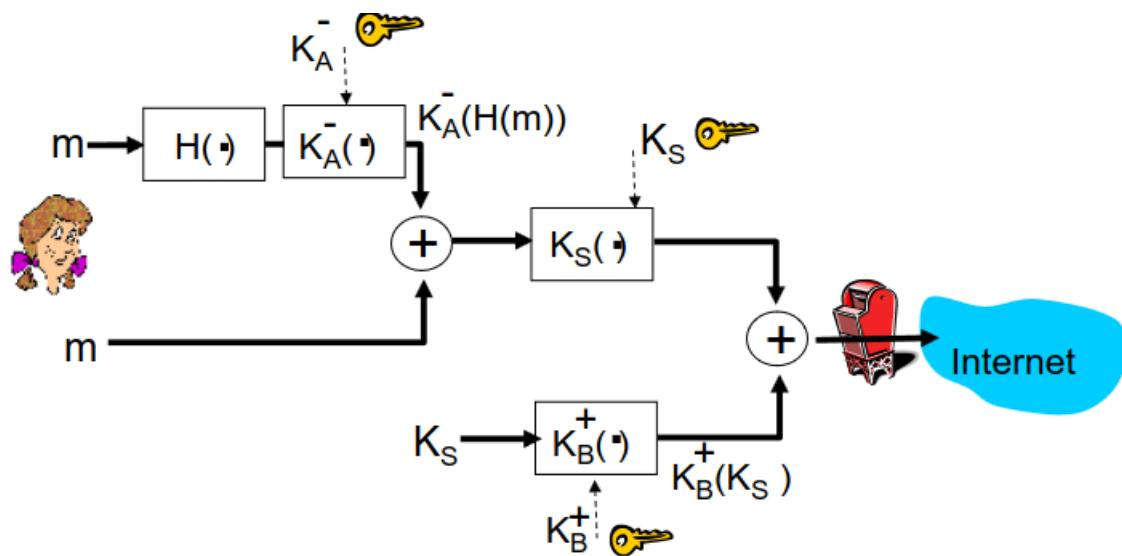


Alice firma digitalmente il messaggio con la funzione hash e critta il messaggio con la sua chiave privata per firmarlo, e invia sia questa versione che l'originale

Bob decripta il messaggio con la chiave pubblica di Alice fornita da CA, fa l'hashing del messaggio originale che arriva e guarda se coincidono i due codici

Garantita integrità e autenticazione ma non la confidenzialità → non ripudiabilità garantita

Completo



Unione delle due procedure: inviati $K_S(K_A^-(H(m)))$, $K_S(m)$, $K_B^+(K_S)$

La PEC è quello che abbiamo visto ora, con in aggiunta una **marca temporale** → stringa che dice in UTC il momento in cui il server di invio ha ricevuto quel messaggio per inviarlo via posta

Anche la marca temporale è fornita da una certification authority

Reti wireless

Introduzione

L'energia elettromagnetica è generata da corrente alternata ad alta frequenza nelle antenne. In particolare, la variazione del voltaggio (energia potenziale elettrica) agli estremi dell'antenna porta a un movimento degli elettroni che genera di conseguenza un campo elettromagnetico variabile

- ampiezza (dB) → l'ampiezza di un'onda elettromagnetica è direttamente proporzionale alla potenza (Watt) del segnale
- frequenza (Hz) → numero di oscillazioni complete (periodi) nel tempo, dividendo la velocità della luce per la frequenza si ottiene la lunghezza d'onda (m)



Legame tra antenne e lunghezza d'onda

esiste una proprietà fisica che dice che le antenne, per essere più efficienti, devono avere una lunghezza pari a 1, $\frac{1}{2}$, $\frac{1}{4}$ (o comunque, devono essere un sottomultiplo binario) della lunghezza d'onda del segnale che stiamo utilizzando. Infatti, le antenne dei router WiFi ad esempio, hanno le antenne di lunghezza pari a 12,5 cm

Propagazione delle onde elettromagnetiche

La propagazione delle onde elettromagnetiche è un fattore cruciale nella trasmissione del segnale → di base, l'ampiezza del segnale si disperde allontanandosi dalla sorgente fino a diventare quasi nullo (in modo esponenziale)

Si determina quindi una quantità chiamata **Signal Detection Limit**, che è la minima energia necessaria per capire informazioni (bit) dall'onda radio → se un client è al di fuori di questa soglia, o si avvicina alla sorgente, o bisogna aumentare la potenza trasmittiva del trasmettitore

Range di propagazione del segnale

- transmission range → il segnale è sufficientemente potente perché possa avvenire la comunicazione

- detection range → il segnale è rilevato ma non è possibile comunicare
- interference range → il segnale si somma al rumore e potrebbe non essere individuato

Gli ostacoli possono assorbire, riflettere le onde o essere oltrepassati → tendenzialmente, le onde ad alta energia vengono riflesse, quelle più a bassa energia (come le onde radio) vengono assorbite o oltrepassano gli oggetti

Fase e interferenza

La **fase** è lo spostamento sull'asse del tempo della sinusoide dell'onda radio rispetto ad un segnale di riferimento → la fase è positiva se rispetto al segnale di riferimento ha uno spostamento a sinistra (segna in anticipo) e negativa altrimenti

Le onde che arrivano al ricevente possono farlo prendendo strade differenti, rimbalzando contro oggetti ecc...

Ecco che nasce il problema dello sfasamento → andando a sommare vettorialmente onde su fasi diverse si potrebbero avere guadagni o perdite in ampiezza (o addirittura annullarsi a vicenda in caso di opposizione di fase)

Problema dello sfasamento → somma vettoriale tra i segnali

Soluzione → array di antenne → se un'antenna si trova in un punto in cui avviene interferenza che annulla il segnale, l'altra sta ricevendo qualcosa

Polarizzazione

Le onde elettromagnetiche sono composte da un campo elettrico e un campo magnetico perpendicolari di cui varia l'intensità → di conseguenza le onde hanno un orientamento rispetto al "livello del suolo" → problema della polarizzazione

Soltamente nelle WLAN viene utilizzata polarizzazione verticale, quindi si ha campo elettrico verticale e campo magnetico orizzontale rispetto al suolo

Il massimo grado di trasferimento di ha quando l'antenna del ricevente è polarizzata come l'antenna del trasmettitore

In ambiente chiuso la polarizzazione delle antenne è poco importante, poiché l'onda radio rimbalza sui muri → ci sarà sempre un'onda che rimbalzando arriva alla stessa polarizzazione dell'antenna del ricevente



Filtro passabanda → circuito elettronico che lascia transitare i segnali che hanno una frequenza in un determinato intervallo

Non si può fare sovrapposizione di onde a stessa frequenza generate da entità diverse → marmellata di segnale, si perde la comprensione di entrambe le comunicazioni → il passabanda lascerebbe passare entrambe

Bisogna che tutti i ripetitori e le sorgenti di una data area comunichino su frequenze differenti

Guadagno e perdita

L'ampiezza del segnale si misura in dB → ci sono vari modi per perdere o guadagnare segnale

- guadagno attivo: quando il segnale guadagna energia mediante un amplificatore
- guadagno passivo: quando si induce l'interferenza costruttiva, ad esempio con una parabola che focalizza il segnale sul punto del ricevitore
- perdita intenzionale: attraverso delle resistenze l'ampiezza viene ridotta trasformando l'energia in calore
- perdita non voluta: dovuta ad oggetti che assorbono l'energia della trasmissione prima che arrivi al ricevente o deviano il percorso del segnale

La perdita è influenzata da:

- **shadowing**: impedisce che l'onda prosegua
- **riflessione**: l'onda viene riflessa
- **rifrazione**: causata dal cambio di dielettrico (ad esempio aria → acqua) cambia leggermente la direzione del segnale
- **diffrazione ai bordi**: una deviazione, dovuta a bordi a curvatura molto alta (ad esempio i bordi di una montagna)
- **scattering**: quasi come un rimbalzo, quando l'onda colpisce uno spigolo vivo

VSWR

Impedenza nominale → somma delle resistenze che una determinata corrente attraversa in un circuito

Se nel percorso trasmettitore-antenna si hanno impedanze diverse, si possono avere problemi di dissipazione di energia (perdita di energia della comunicazione)

Il **Voltage Standing Wave Ratio** (VSWR) misura la portata di questo fenomeno → per evitare problemi dobbiamo fare attenzione che il rapporto di impedenza sia sempre 1 a 1

- Quando abbiamo un sistema di comunicazione, tutto il blocco (esclusa l'antenna) che invia il segnale si chiama **Intentional Radiator** (Radiatore intenzionale, IR)
- Quando misuriamo l'energia di un sistema di trasmissione radio l'energia a cui facciamo riferimento è l'energia che arriva all'antenna → questa energia viene calcolata nel sistema come **Intentional Radiator Power Output** (ovvero la potenza di uscita dell'Intentional Radiator), che non è l'energia complessiva ma quella che arriva all'antenna, escludendo quindi tutte le perdite dovute ai collegamenti, alla trasmissione, ecc...

EIRP

Un **radiatore isotropico** è un'antenna "perfetta" (non realizzabile), che emette energia nello spazio a 360 gradi equivalentemente → un'antenna direzionale, che emette energia in almeno una direzione preferenziale, avrà una quantità di energia emessa non uniforme nello spazio

Il valore EIRP (**Equivalent Isotropically Radiated Power**) è il valore dell'energia di Intentional Radiator Power Output equivalente a quella necessaria ad un radiatore isotropico per generare lo stesso effetto di onda elettromagnetica nella direzione preferenziale



Esempio dalle dispense

Se do 16mW direttamente all'antenna isotropica, l'energia irradiata in tutte le direzioni è 16mW.
Se do 16mW ad un'antenna che focalizza 10 volte l'energia in una direzione preferenziale (sottraendola alle altre direzioni ovviamente) il valore EIRP di questa nuova antenna è il valore che dovrei dare ad un isotropico poiché esso possa irradiare 160mW, e quindi in questo caso il valore EIRP è pari a 160mW.

Quando l'antenna non è isotropica, EIRP mi dice quale sarebbe l'energia dell'isotropica equivalente.

Misurazione dell'energia

Watt

Nell'ambito dell'elettromagnetismo (una delle definizioni di potenza):

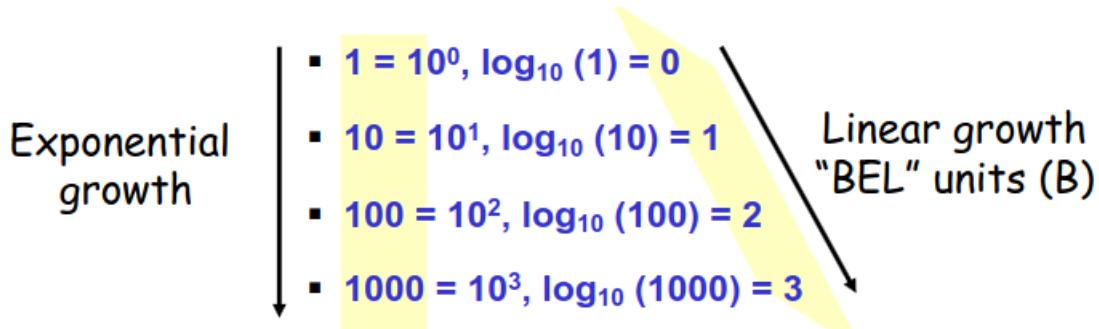
$$1 \text{ Watt} = 1 \text{ Ampere} \cdot 1 \text{ Volt}$$

- l'Ampere è l'unità di misura dell'intensità di corrente elettrica, ovvero è indicatrice sulla quantità di elettroni che attraversano, ad esempio, un cavo
- il Volt è l'unità di misura della differenza di energia potenziale elettrica (o tensione), ovvero la "pressione" applicata al flusso di elettroni (tanto maggiore quanto la quantità di carica si differenzia ad esempio tra gli estremi di un condensatore)
- La potenza elettrica, espressa in **Watt**, è quindi l'energia necessaria nell'unità di tempo per applicare una determinata tensione a una quantità di carica

Decibel

Il **Decibel** (dB) è l'unità di misura della potenza al segnale (alternativa al Watt) che serve per esprimere perdite e guadagni di potenza → nasce da una scala logaritmica che mappa i mW alle rispettive grandezze in dB

NOTA: il dB non è una quantità di energia pura ma rappresenta un rapporto di valori di energia tra due riferimenti → un valore in dB positivo denota un guadagno di potenza mentre un valore negativo è associato a una perdita



Esempio

- E.g.: A signal transmitted at [TX] 100 mW is received at [RX] 0.000005 mW
 - Power Difference (dB) = $10 * \log([RX] / [TX]) = 10 * \log(0.000005\text{mW}/100\text{mW}) = -73$
 - A signal transmitted at 100 mW is received with gain (loss) -73 dB

NOTA: un guadagno di 3 dB equivale circa al raddoppiare della potenza del segnale, un guadagno di 10 dB equivale a un 10x (essendo la scala logaritmica, il guadagno o la perdita non sono lineari all'aumentare dei dB)

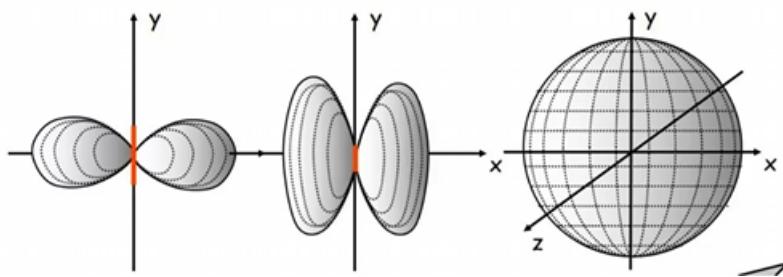
dBm → unità di misura utilizzata per la misurazione della potenza del segnale → 1 mW = 0 dBm (misura di riferimento)

Dipolo, dBi e dBd

In un'antenna, il **dipolo** è un segmento conduttore con due poli, uno positivo e uno negativo, verso i quali la corrente oscilla in variazione al potenziale applicato ai poli. Questa variazione di corrente sinusoidale genera l'onda eletromagnetica

La forma del dipolo influenza come l'energia è irradiata nello spazio attorno all'antenna

- le antenne ad alto guadagno presentano un dipolo più allungato ed irradiano più energia orizzontalmente ma penalizzando l'asse verticale
- le antenne a basso guadagno hanno un dipolo più schiacciato e tendono a irradiare l'energia maggiormente in verticale perdendo sull'asse orizzontale
- l'antenna ideale (isotropica) si ottiene con un dipolo nullo → l'energia è irradiata uniformemente in tutte le direzioni



dBi → dB-isotropico, misura normalizzata del guadagno passivo di un'antenna

In sostanza è un'unità di misura che si riferisce al guadagno o alla perdita di potenza dell'antenna in una determinata direzione rispetto alla potenza irradiata

dBd → dB-dipolo, misura normalizzata del guadagno passivo di un'antenna rispetto a un dipolo di riferimento e non a un'antenna isotropica (il dipolo di riferimento è un dipolo a mezza lunghezza d'onda che ha un guadagno di 2.15 dBi → per ottenere dBd a dBi basta fare +2.15)

Monitorare la potenza del segnale

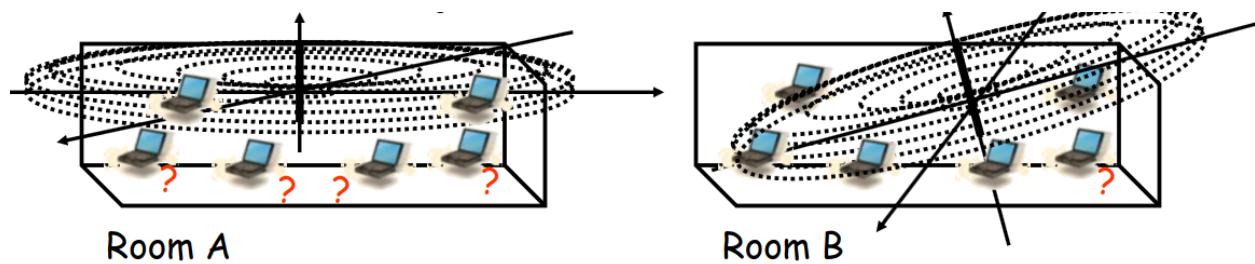
In IEEE 802.11 (protocollo WiFi) monitorare la potenza del segnale è necessario per far lavorare correttamente i driver → esiste un indicatore chiamato **RSSI** (Received Signal Strength Indicator) che è sostanzialmente una funzione device-dipendente (scala diversamente a seconda del device) che prendi in input i dBm o mW ricevuti e ritorna un numero puro al driver → in linea di massima, più è alto il valore, più è potente il segnale ricevuto

Antenne

Antenne omnidirezionali

Sono le antenne che assomigliano di più all'isotropico, e le antenne maggiormente utilizzate nella vita quotidiana → il dipolo ha lunghezza variabile, distinguendo tra antenna a basso o ad alto guadagno (vedi sopra)

Di base l'antenna è orientata con il dipolo verticale e trasmette a "ciambella", quindi sopra e sotto il dipolo il segnale è scarso → si può inclinare l'antenna ottenendo un'**antenna tilt**

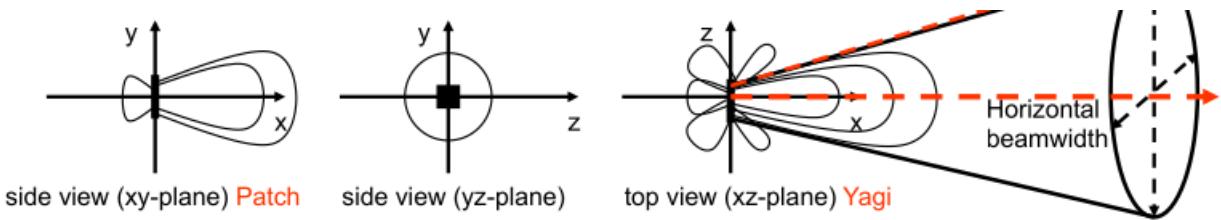


Antenne semidirezionali

Le antenne semidirezionali trasmettono di più in una direzione ma NON SOLO in una direzione, un po' come delle torce elettriche

Possono essere di vario tipo:

- patch/panel → antenne piatte montate solitamente a muro, copertura relativamente ristretta
- yagi → antennone "classico" di metallo, con quelle specie di spine che escono, copertura più ampia



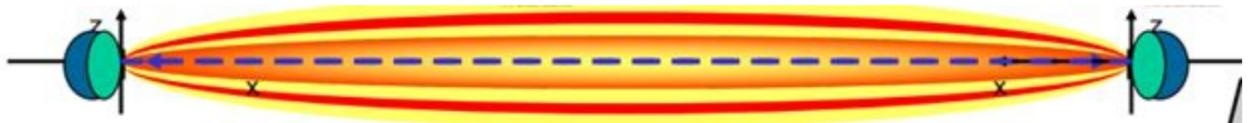
Forma del flusso di emissione dell'antenna semidirezionale, a seconda delle varianti e delle angolazioni

Antenne fortemente direzionali

Antenne in cui tutta l'energia emessa è concentrata (tramite passive gain) su un cono d'ampiezza molto ristretto

- Sono le comuni parabole o le antenne a griglia
- Uso comune → connessione punto a punto

Fondamentale aspetto è che l'antenna sia saldamente fissata e direzionata molto precisamente, altrimenti la connessione fallisce → può essere talvolta sensato rinunciare a del guadagno per aumentare l'ampiezza del cono



line of sight → linea blu in figura, una linea dritta che congiunge i due dispositivi

Quando la line of sight è libera il trasferimento di energia tra trasmettitore e ricevitore è ottimale

Quando la **Fresnel zone**, ovvero la zona attorno alla line of sight, è libera, si gode appieno degli effetti additivi del segnale → l'ideale è calcolarne la sezione e lasciarla libera almeno per il 60%

$$R_{60\%} = 43.3 \times \sqrt{(d/4f)}$$

$$R_{100\%} = 72.2 \times \sqrt{(d/4f)}$$

d è in miglia, f è in GHz

Il raggio della sezione è dipendente da una costante, dalla distanza tra trasmettitore e ricevitore, e dalla frequenza impiegata dalla comunicazione (risultato in piedi, distanza in miglia)

NOTA: l'ampiezza della FZ è indipendente dal tipo di antenna e dalla potenza del segnale, quindi aumentare la potenza erogata o cambiare antenna non risolve il problema se la FZ è ostruita

NOTA2: per antenne molto distanti si tiene conto della curvatura terrestre, sollevando trasmettitore e ricevitore da terra

Antenne settorizzate

Usando un insieme di antenne direzionali o semidirezionali si può ottenere qualcosa di simile a un'antenna omnidirezionale, collocandone un certo numero in maniera circolare di modo che le emissioni delle varie antenne non facciano conflitto → in questo modo c'è un riuso maggiore delle frequenze MA due settori adiacenti non usano mai lo stesso canale

Space multiplexing → gestione dello spazio per massimizzare il riuso di una frequenza

Utilizzo di una frequenza pilota che coordina la parte amministrativa della trasmissione (tutti i telefoni la utilizzano) e comunica ai dispositivi quali frequenze sono disponibili in quali punti per potersi agganciare → o vengono assegnate direttamente, o il dispositivo può decidere a quale appoggiarsi tra quelle libere

Hand Off o Hand Over (passaggio di mano) → passaggio di collegamento di un host da un ripetitore all'altro

Azimuth e Elevation charts (RECUPERARE)

Confronto tra antenne

Antenna type	H beamwidth	V beamwidth
Omni-dir.	360°	7°.. 80°
Patch/panel	30° .. 180°	6° .. 90°
Yagi	30° .. 78°	14° .. 64°
Parabolic dish	4° .. 25°	4° .. 21°

Diversity

Nei router moderni vi sono **array di antenne**, non un'antenna soltanto:

- Sono tutte a distanza reciproca di Lambda/2 → Perché?
 - In questo modo è impossibile avere opposizione di fase su tutte le antenne (avrò almeno un'antenna dove non vi è opposizione di fase)
- Per sfruttare ancora meglio le possibilità di avere un array di antenne, viene estratto il segnale da tutte le antenne: sommando i segnali si genera guadagno, così da catturare al meglio l'energia in arrivo anche in condizioni non ideali
- In caso di rumore, il massimo comun divisore di ciascun segnale in arrivo è il messaggio

Path loss

Come calcoliamo il **path loss**, ovvero qual'è la funzione che ci indica come si attenua il segnale all'aumentare della distanza? (La costante è 36.6 sulla terra e 32.4 nel vuoto)

$$Loss = 36.6 + (20 \times \log_{10}(F)) + (20 \times \log_{10}(D))$$

Loss (dB) F(Mhz) D(miles)

-6dB rule → al raddoppiare della distanza si ha una perdita di 6dB, ovvero il segnale si riduce a circa 1/4, quindi per raddoppiare la distanza percorsa dobbiamo quadruplicare l'energia del segnale

Link Budget

Con **link budget** intendiamo l'eccesso di segnale al destinatario della trasmissione rispetto alla signal detection limit

Si può misurare in maniera relativa (dB) o assoluta (dBm, mW)

Il sistema di comunicazione va progettato di modo che questa quantità sia positiva e con un certo margine, di modo che le interferenze non compromettano la trasmissione

$$Link\ Budget = received\ power - receiver\ sensitivity$$

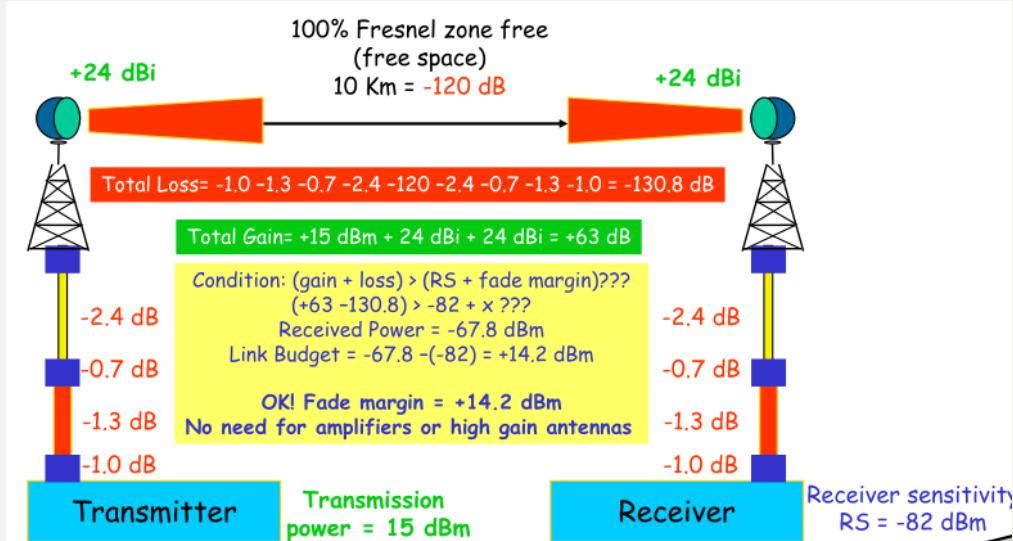
Tutto calcolato in dBm

Il **fade margin** è un margine da tenere perché la comunicazione regga varie condizioni di interferenza
→ tra 10 e 20 dB normalmente

Il link budget deve almeno rientrare nel fade margin

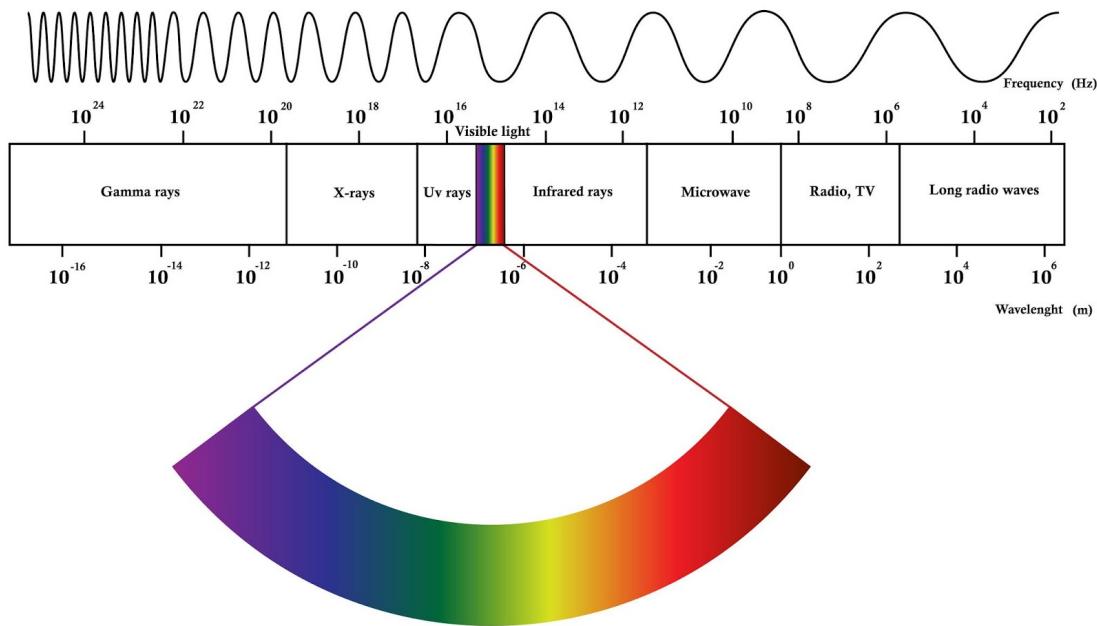


Esempio



Livello fisico

Introduzione



Lo spettro radio è compreso tra 100KHz e 100GHz → spettro molto ampio

Tra 1.8 e 2.4 GHz, abbiamo il dualband, tra 2.4 e 5 abbiamo Wifi e Bluetooth

La quasi totalità di frequenze non sono in uso esclusivo di una tecnologia, ma hanno più utilizzi in parallelo



Rete e confini di stato

I soldi che si danno a un provider sono legati alla licenza d'uso di una fetta di banda (da pagare allo stato) → problema ai confini tra stati per non "sconfinare" con le trasmissioni

Nel caso di trasmissioni internet vi è un capillare controllo sull'IP del dispositivo che accede a un servizio → un servizio con licenze vincolate a un contesto nazionale, se l'IP di destinazione è estero il contenuto non viene erogato

Nel caso di trasmissione dei bit tramite onda radio, cosa differenzia le varie tecnologie? → frequenza di trasmissione → comunicazione di più bit al secondo

Collegamenti in un canale radio

- link assente → i dispositivi non riescono a comunicare
- link unidirezionale → solo uno parla con l'altro
- link asimmetrico bidirezionale → la frequenza di trasmissione è asimmetrica, la connessione è bidirezionale
- link simmetrico bidirezionale

ADSL → download >> upload

Tipi di canale

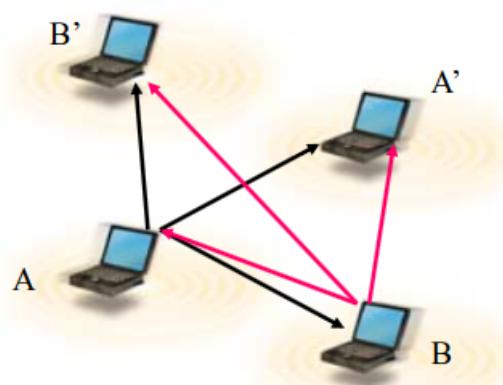
1. Canale **narrow band** → singola radiofrequenza concordata (o intervallo stretto) scelto per comunicare
2. Canale **spread spectrum** (a spettro sparso) → canale radio rappresentato da uno spettro di frequenze
 - **Frequency Hopping** → si fanno salti (hop) tra frequenze dell'intervallo tra una trasmissione e l'altra
 - **Direct Sequence** → informazione (bit) spalmata su tutte le frequenze dell'intervallo

Narrow band

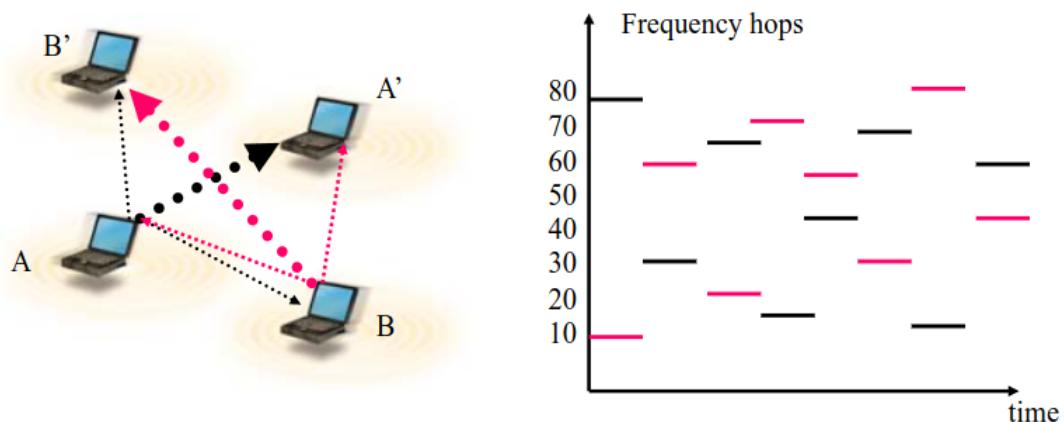
Se sono più coppie di dispositivi a comunicare bisogna che i segnali non si disturbino a vicenda → diversificazione dei canali → utilizzo di frequenze diverse per comunicare

Da una marmellata di segnale il filtro passabanda riesce a ottenere solo il segnale che è destinato al ricevente

Necessario un agreement su quali frequenze comunicare → canale separato utilizzato per stipularlo chiamato canale di servizio



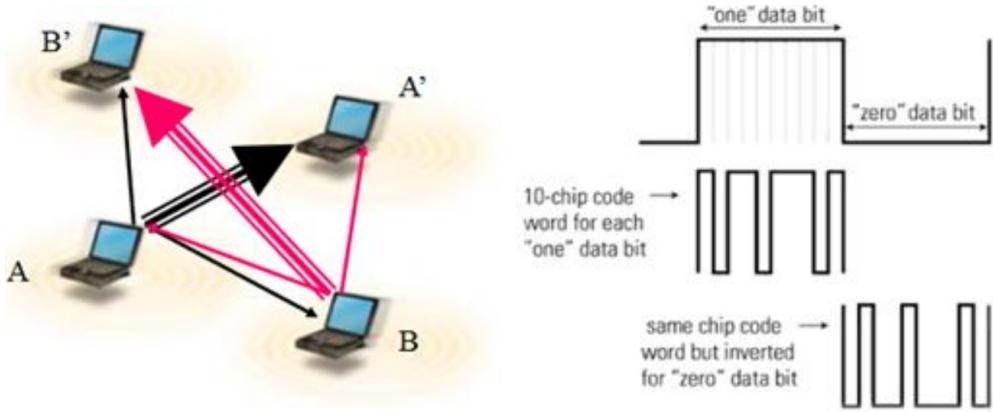
Frequency Hopping



Si cerca di evitare che la comunicazione tra link diversi avvenga su stessa frequenza → algoritmo pseudocasuale di salti sincroni

- Il pattern è conosciuto da ricevitore e trasmettitore (chiave pre-condivisa) → sicura la comunicazione, poiché chi non conosce il pattern non riesce a sentire la comunicazione completa
- se c'è interferenza sul canale, si salta alla frequenza successiva
- le collisioni sono inevitabili

Direct Sequence



Si divide il segnale in **"chip"**, ovvero in frammenti, distribuiti nell'intervallo di frequenza:

- Nel caso di codifica binaria, la ricostruzione dei chip significa il bit 1
- Prendendo l'inverso di quei chip si ottiene lo 0

→ per ricostruire il segnale si sommano i chip che arrivano dai vari canali

Per diversificare tra comunicazioni diverse, si utilizzano chip differenti per la rappresentazione dell'informazione tra ogni coppia di interlocutori

NOTA: i segnali se sommati direttamente assieme si possono annullare, tuttavia da una marmellata di segnale il ricevitore riesce tendenzialmente a correlare la sequenza all'informazione che deve arrivare a lui → i golden codes sono sequenze che garantiscono un'autocorrelazione ottimale e le utilizziamo in questo caso

NOTA2: la trasmissione del segnale non è più lenta in quanto i chip vengono mandati in parallelo sulle varie frequenze della banda

Tipi di reti wireless (per copertura)

- Wireless Wide Area Network (WWAN)
 - geographic coverage (e.g. satellite, cellular)
- Wireless Metropolitan Area Network (WMAN)
 - Metropolitan coverage (e.g. town, large campus)
- Wireless Local Area Network (WLAN)
 - local area coverage (e.g. campus, building, home)
- Wireless Personal Area Network (WPAN)
 - reduced local area coverage (e.g. house, office)
- Wireless Indoor Area Network (indoor)

- short range coverage (e.g. room, office)

WWAN/WMAN

- rete satellitare
 - orbita geostazionaria → la posizione apparente del satellite è sempre la stessa → l'orbita del satellite segue la rotazione terrestre
 - vantaggio: per ricevere da un satellite mediante una parabola la posizione rimane fissa → una volta direzionata la parabola

footprint → area di copertura di una rete wireless (access point)

last mile → parte finale del collegamento

- rete cellulare
 - insieme di access point collegati da dorsali (backbones) → la rete cellulare diventa in questo senso identica a una rete domestica
 - dal momento che un dispositivo si sposta allacciandosi di volta in volta ad access point differenti, il suo IP cambia → mobile IP
- ad hoc → connessione estemporanea tra oggetti di una rete, sfruttando direttamente le onde radio emesse da dispositivi vicini
 - comunicazione peer2peer
 - non essendoci un'infrastruttura, la rete è di fatto senza costi (salvo scheda wifi dei vari dispositivi)

Wired vs Wireless

Come ovviamente alla forte eterogeneità protocollare e di tecnologie utilizzate su internet?

- Utilizzo di **bridges** → dispositivi che adottano tecnologie di rete multiple, che fanno da ponte

Limiti delle reti wireless:

- capacità del canale ridotta rispetto alle tecnologie wired
- spettro limitato di frequenze disponibili su cui trasmettere (di cavi se ne può collegare quanti se ne vuole)
- necessari protocolli conservativi in termini di energia → l'energia delle batterie è limitata
- problemi di interferenza e rumore radio

- sicurezza → il segnale passa nell'ambiente

Best effort → non è garantita la qualità del servizio

Multiplexing

Come facciamo a massimizzare il numero di comunicazioni contemporanee data una banda di frequenza, un tempo, uno spazio e eventualmente l'uso di chipping sequence differenti?

- uso della stessa frequenza in spazi diversi → si può comunicare su una frequenza già utilizzata purché i dispositivi che la sfruttano siano sufficientemente lontani (fuori dalla detection zone del nostro dispositivo)
- in tempi diversi → trasmissioni che non si sovrappongono temporalmente possono avvenire su stessa frequenza
- frequenza (e tempo) → utilizzando frequency hopping su una banda di frequenza, è possibile avere più comunicazioni su stessa banda, purché i salti di frequenza delle varie comunicazioni siano sincronizzati in modo tale da evitare (quanto più possibile) di sovrapporle sulla stessa frequenza
- codice → NON è una dimensione fisica → sfruttando chipping sequence per diversificare le comunicazioni su una stessa banda è possibile comunicare nonostante le collisioni, purché le sequenze utilizzate per codificare l'informazione che si attende siano autocorrelate in maniera ottimale (più avanti approfondiremo questa tecnica vedendo cosa si intende con **CDMA, Code-division multiple access**)

Per un ragionamento sulle varie possibilità di multiplexing, a prescindere dalle tecnologie implementate nella realtà per le comunicazioni di rete, vedere lucidi RDC-Wireless2 slide 24-28

Spazio di guardia → spazio di frequenze inutilizzate tra canali differenti, perché se l'ambiente devia le caratteristiche del segnale semplicemente non viene considerato e non ci sono problemi

L'assegnazione di frequenze alle comunicazioni al fine di sfruttare al massimo lo spazio disponibile è detto **frequency planning**

- assegnamento fisso: assegnamento di una frequenza a uno specifico canale
- assegnamento dinamico: frequenze di trasmissione assegnate rispetto a che frequenze sono già assegnate nella zona → più versatile per ovviare a interferenze e canali a carico elevato

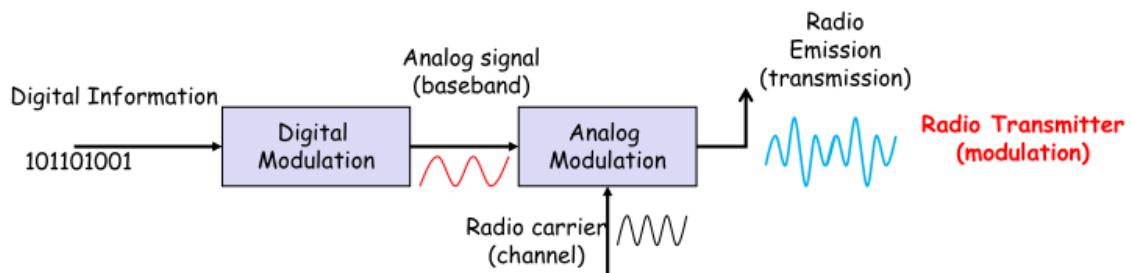
Modulazione

Per la trasmissione wireless, i dati digitali sono tradotti in un segnale analogico da modulare, ovvero da predisporre per essere trasmesso tramite un mezzo fisico, come un'antenna o un cavo Ethernet, a una determinata frequenza di trasmissione.

In particolare, in caso di modulazione digitale, ci serve innanzitutto uno strumento hardware chiamato **modulatore digitale** per convertire la codifica digitale (0 o 1) in un segnale analogico, ovvero una sinusoide → questo si fa tramite dei metodi che trasformino quindi dei simboli di una codifica in onde da caratteristiche distinguibili, che sono ASK (differenza di intensità), FSK (differenza di frequenza) e PSK (differenza di fase)

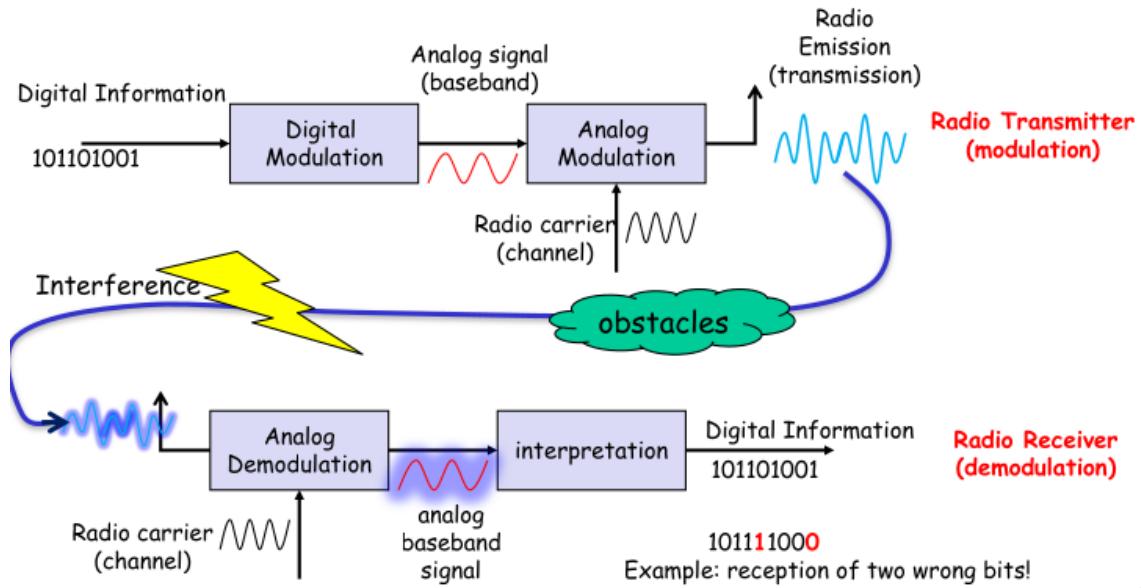
A questo punto abbiamo la nostra **baseband**, un segnale analogico che deve ancora essere manipolato prima di essere trasmesso sul canale. In particolare serve una altro modulatore, detto **modulatore analogico**, che ci dia un segnale idoneo a essere trasmesso su un determinato canale → il modulatore prende in input la frequenza a cui dobbiamo trasmettere, detta **radio carrier** (per poter comunicare sul canale della trasmissione) e la baseband, facendo una sorta di sovrapposizione tra questi due segnali. Nuovamente, ci sono una serie di metodi che ci permettono di fare questa sovrapposizione, speculari a quelli citati sopra: AM (informazione codificata nella variazione di intensità del carrier), FM (variazione di frequenza) e PM (variazione di fase).

Il segnale viene ricevuto dall'antenna e trasmesso



Ora che il segnale è stato trasmesso, può essere soggetto a interferenze di vario tipo e incappare in ostacoli che possono deviare il suo percorso e variarne la fase, finché non raggiunge l'antenna del ricevente.

Dal lato del ricevitore, si ha un percorso speculare a quello sopra: un **demodulatore analogico**, che prende in input il carrier e il segnale ricevuto, estrae la baseband e la manda a un **interprete**, che la converte in informazione digitale. Il segnale ricevuto non è mai identicamente uguale a quello trasmesso, ma in qualche modo lo contiene. Sta all'interprete capire a quale codifica corrisponde il segnale rumoroso ricevuto.



Il problema che si pone ora è come scegliere una codifica in modo da poter riconoscere eventuali bit mal interpretati rispetto al segnale trasmesso, di modo da minimizzare le volte in cui è necessario chiedere nuovamente una trasmissione.

Tecniche di modulazione digitale

Vediamo nello specifico le varie codifiche che otteniamo da un modulatore digitale: ASK, FSK e PSK

- **ASK**

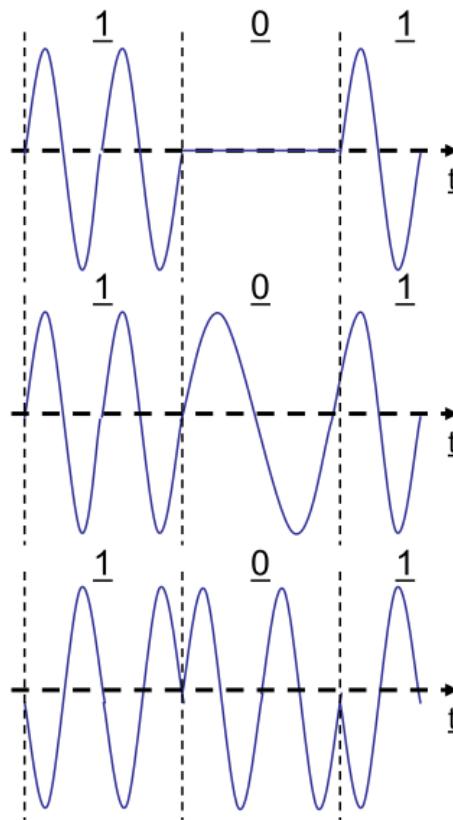
Semplice on/off del segnale, usa poche risorse dello spettro ma è fortemente inadatto in ambienti dove è possibile forte interferenza costruttiva o distruttiva

- **FSK**

Utilizza più risorse dello spettro visto che a ogni elemento della codifica si assegna una frequenza

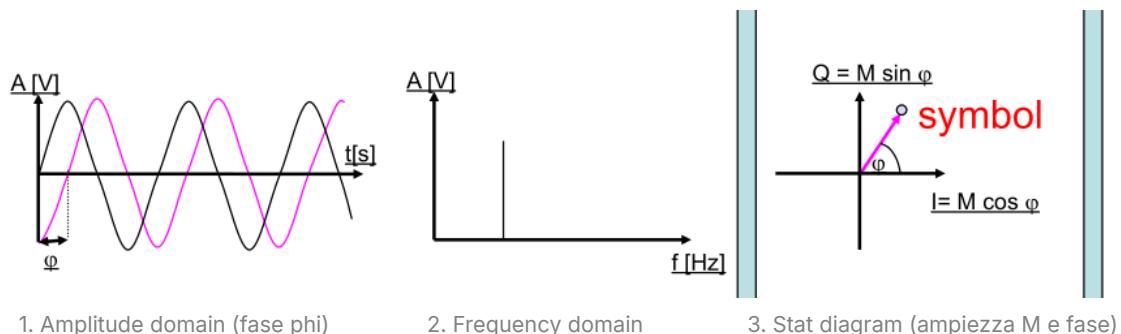
- **PSK**

Più complesso da implementare, più resistente all'interferenza, tante fasi di segnale possibili → adatto a codifiche più complicate



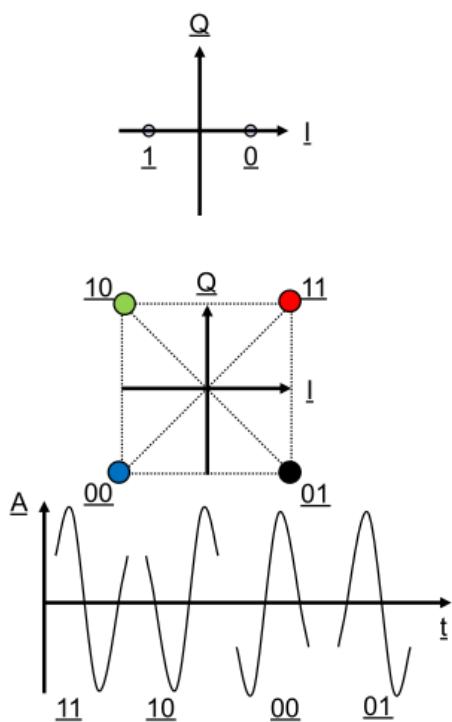
Rappresentazione del segnale

Come rappresentiamo graficamente le caratteristiche di un radiosegnale?



La terza rappresentazione si presta per rappresentare codifiche in PSK

- **BPSK** → codifiche diverse per opposizione di fase (0 e 1, solo due codifiche) → robusto ma sfrutta poco lo spettro
- **QPSK** → codifica in 4 fasi, permette la codifica di 4 simboli (00, 11, 01, 10) ma è più vulnerabile, ovvero è più facilmente mal interpretabile



Una specie di tiro al bersaglio

Ovviamente il segnale arriva al ricevitore alterato rispetto a come è stato inviato → non centra esattamente i 4 punti che abbiamo definito, ma possiamo ricondurlo al punto più vicino (in che quadrante cade?) → chiaramente, maggiore è l'area che associamo a una codifica, più sarà difficile sbagliare a decodificare il segnale quando arriva, tuttavia sprechiamo banda

NOTA: il modo più intelligente di assegnare codifiche a fasi, è quello di minimizzare il numero di bit differenti tra fasi vicine, così in caso di errore i bit errati sono il numero minore possibile → la cosa migliore è che questo numero sia 1

Perchè proprio 1?



Bit di parità

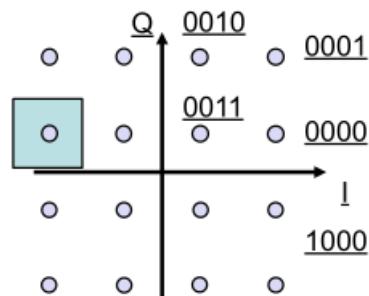
Già visto in tanti corsi, si dedica un bit di modo che il numero di bit a 1 sia pari

Se utilizziamo una struttura a matrice, riusciamo anche a correggere il bit sbagliato

Sender	Receiver
1 0 0 1 0	1 0 0 1 0
0 1 1 0 0	0 0 1 0 0
0 0 0 1 1	0 0 0 1 1
1 1 0 1 1	1 1 0 1 1
0 0 1 1	0 0 1 1

Se sappiamo che la trasmissione è di buona qualità, possiamo spingere ancora oltre la codifica

QAM → combinazione di modulazione di ampiezza e fase del segnale per ogni simbolo trasmesso
es. 16-QAM (16 symbols, 1 symbol = 4 bit)



Ovviamente bisogna stare attenti perchè più simboli vogliamo rappresentare, più la target area si riduce

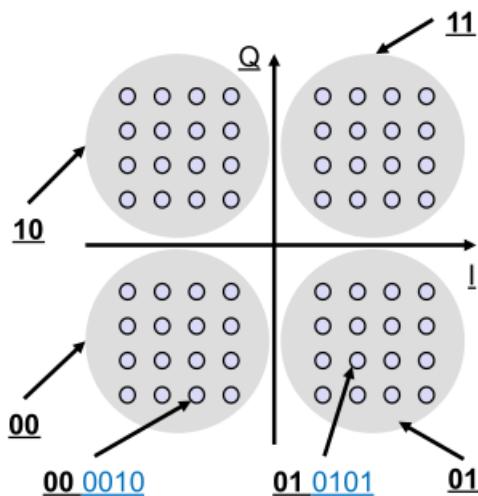
Si notino sulle bisettrici dei quadranti (e in generale, per ogni gruppo di simboli che giacciono sullo stesso angolo) si hanno (in questo caso coppie) simboli codificati con stessa fase ma ampiezza differente

Modulazione gerarchica

- Posso tramite QAM modulare due differenti sequenze di bit nello stesso segnale?

- Posso dare una priorità diversa alle due sequenze?

Sì, tramite **modulazione gerarchica** → vediamo di cosa si tratta



Ogni area grigia contiene 16 simboli (i vari punti) utilizzati per codificare la sequenza a priorità minore

Ogni quadrante invece è etichettato con una sequenza di due bit, quelli a priorità alta

Quando il canale ha poco rumore, si tende ad avere errori solo sulla parte a bassa priorità, chiaramente se il rumore aumenta è compromessa anche la codifica a priorità alta

| Vedi esempio "The mobile Video-call" su RDC-Wireless slides 62-77

Tecnologie spread spectrum

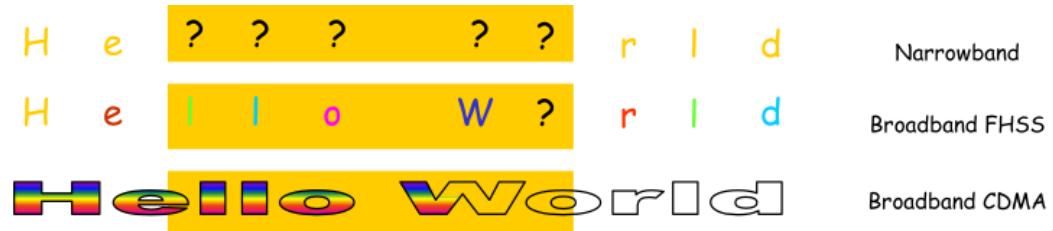
Un problema della tecnologia narrow band per le comunicazioni è che dedicando una sola frequenza alla comunicazione, se tale frequenza risente particolarmente di un certo tipo di interferenza, la comunicazione è compromessa finché l'interferenza non si attenua o svanisce

Utilizzando tecniche spread spectrum, che permettono di suddividere l'informazione in una banda di frequenze, il problema viene arginato. In particolare, per quanto riguarda la tecnica direct sequence, l'interferenza interesserà solo una delle componenti che formano il segnale

Inoltre si ricorda che:

- utilizzare direct sequence per una comunicazione impedisce a intrusi che non conoscono la codifica relativa alle trasmissioni di un utente di capire le informazioni che trasmette sul canale

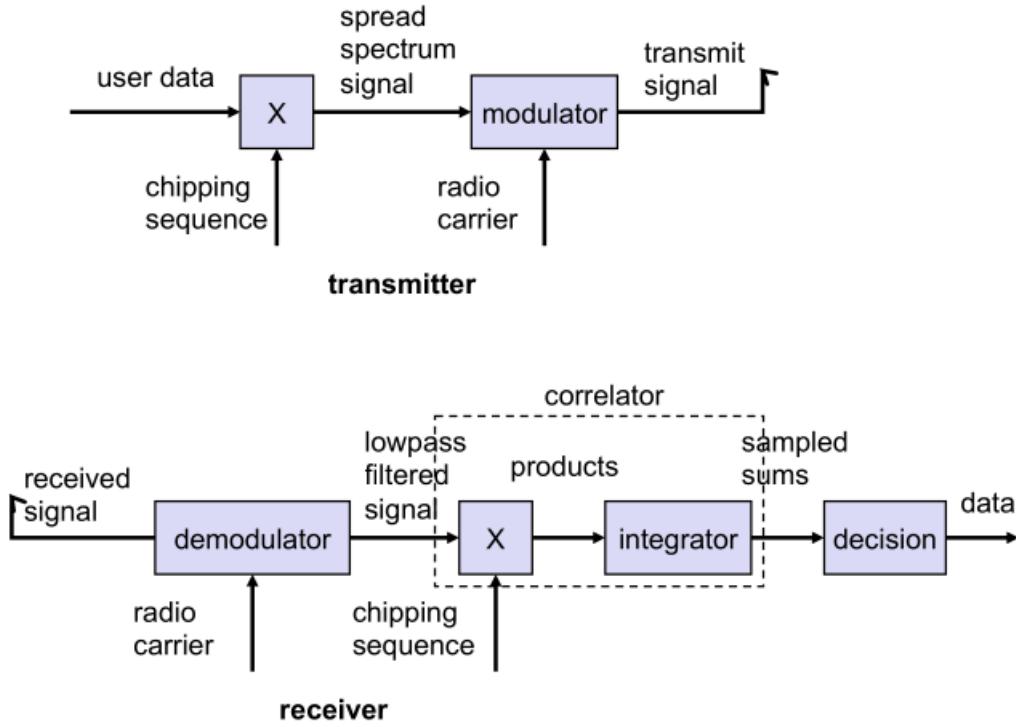
- la coesistenza di segnali diversi su una banda di frequenza non necessita di coordinazione



Direct Sequence Spread Spectrum (DSSS)

L'informazione binaria viene divisa in chip, come già visto → tramite XOR si va a correlare l'informazione inviata con la chiave (chipping sequence) di un utente

- riduce il fading relativo a frequenze specifiche (vedi sopra)
- nelle reti cellulari, le stazioni possono utilizzare lo stesso range di frequenze e, lato ricevente, possono rilevare e ricostruire il segnale
- necessaria sincronizzazione e power control (?)



X indica lo XOR

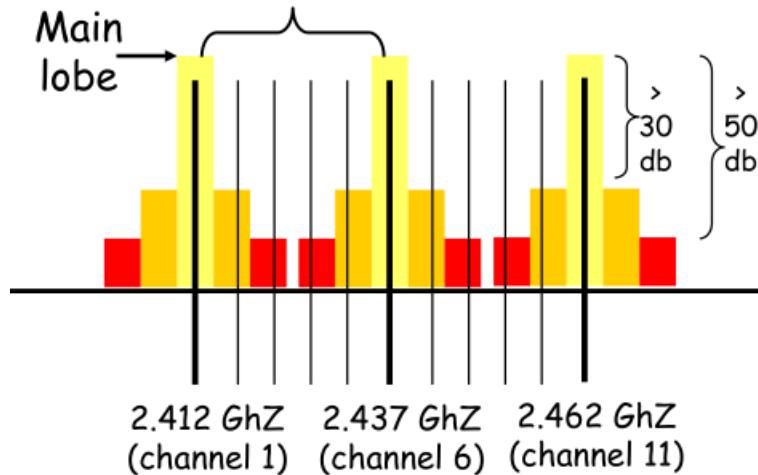
Il lato transmitter è abbastanza intuitivo → unica cosa, come gestiamo il carrier visto che dobbiamo trasmettere su una banda? cambia volta per volta?

Il lato receiver abbiamo una parte più complessa relativa alla ricostruzione del segnale demodulato nel dato originale, tenendo conto che il segnale ottenuto è soggetto a interferenza

Main lobe

Con DSSS abbiamo lo spettro suddiviso in 14 canali (effettivamente utilizzati solo 11 negli Stati Uniti) ma non sono tutti utilizzati in maniera uguale.

Si dicono **main lobe** i canali che contengono la maggior parte dell'informazione trasmessa, tra cui effettivamente i dati che si vogliono trasmettere. Essi sono inoltre i canali in cui la potenza del segnale è massima. I **side lobe**, che sono i canali limitrofi, si utilizzano per funzionalità di controllo, ridondanza dei dati (per tentare di ricostruirli in caso di problemi) e in comunicazioni multicast



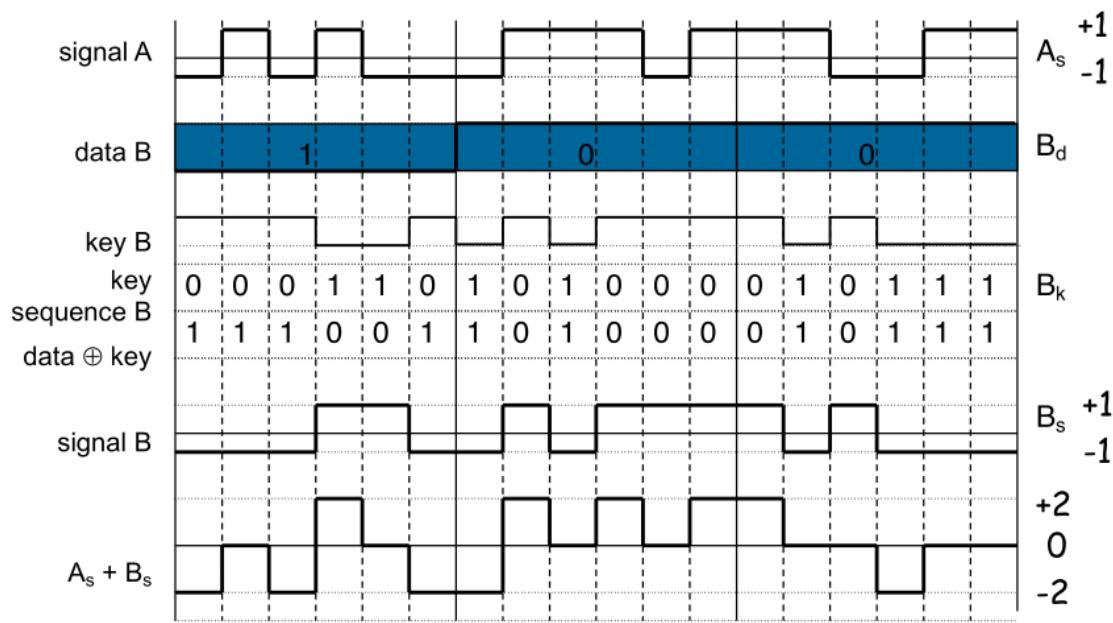
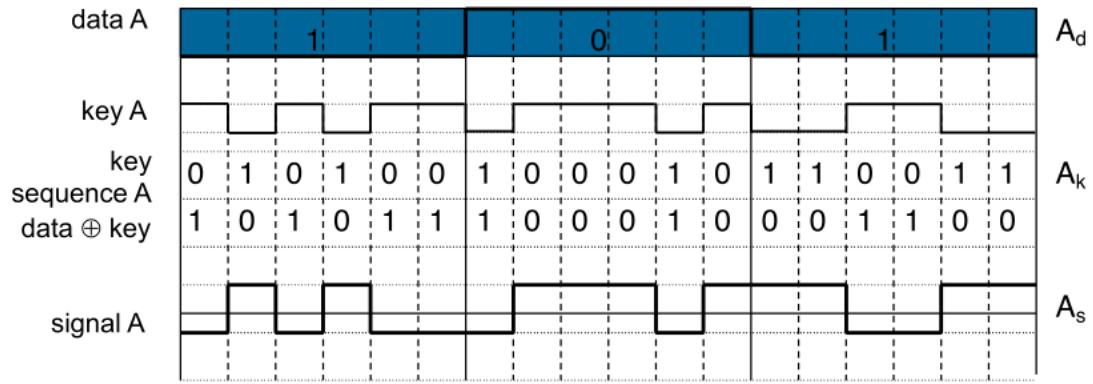
Nel caso di canali sovrapposti si parla di **overlapping channels** → è evidente che le sovrapposizioni tra canali in comunicazioni diverse sia inevitabile, ma si cerca di non fare interferire tra loro i main lobe delle varie comunicazioni

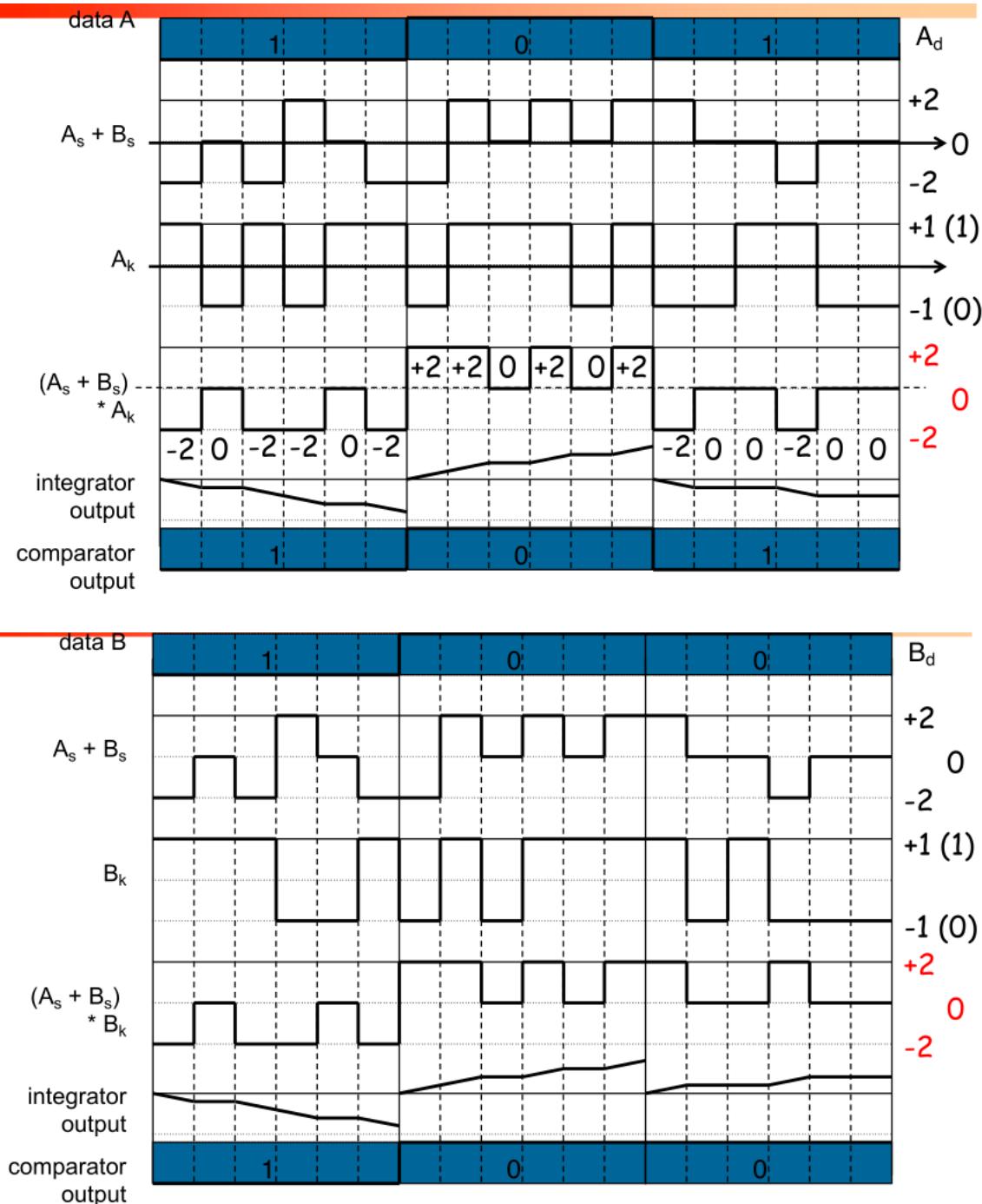
Code Division Multiple Access (CDMA) - ricostruzione del segnale

Abbiamo due sender A e B, ognuno con il messaggio (bit) che vuole trasmettere e la propria chiave (chipping sequence?)

- Chiave e messaggio vengono codificati con $0 = -1$ e $1 = +1$
- Il segnale da inviare si ottiene tramite XOR tra messaggio e chiave
- I segnali si sovrappongono, andando a sommare le componenti
- A quel punto, dalla marmellata di segnale, si va ad estrarre il segnale desiderato applicando la chiave corrispondente (moltiplicando) → compito dell'**integrator**
 - a seconda se il segnale è positivo o negativo, si risale a quale fosse il bit originale inviato → compito del **comparator**

▼ Esempio esplicativo

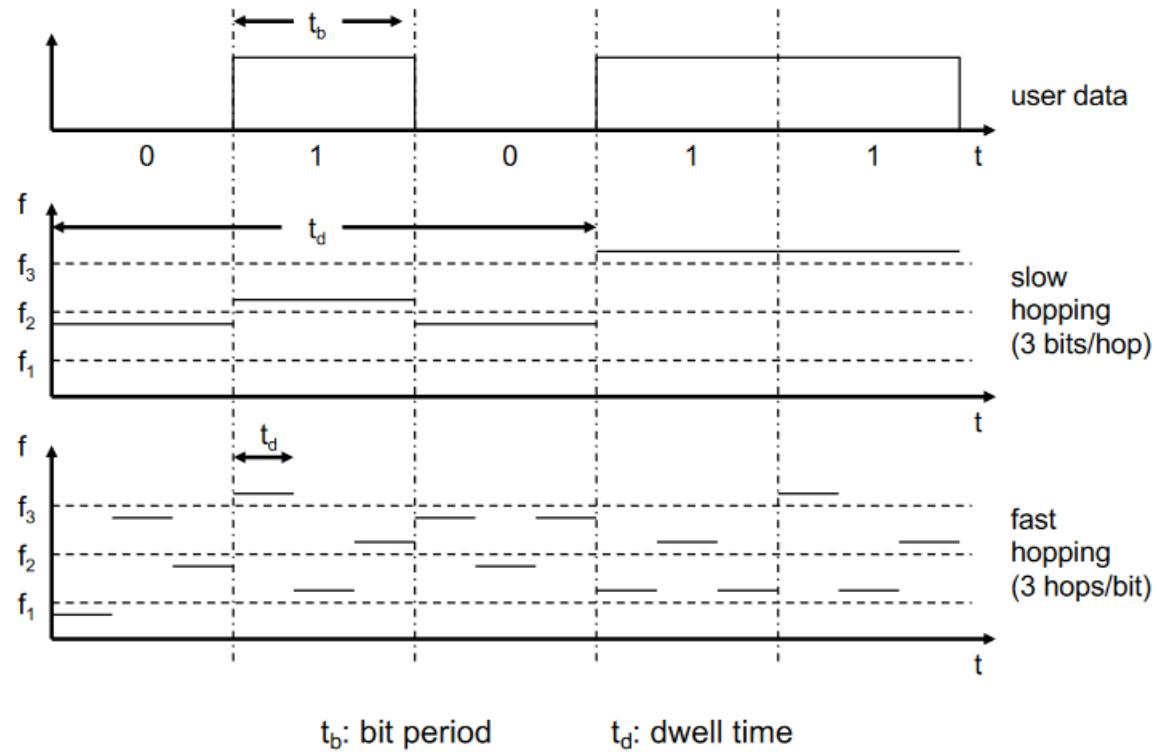


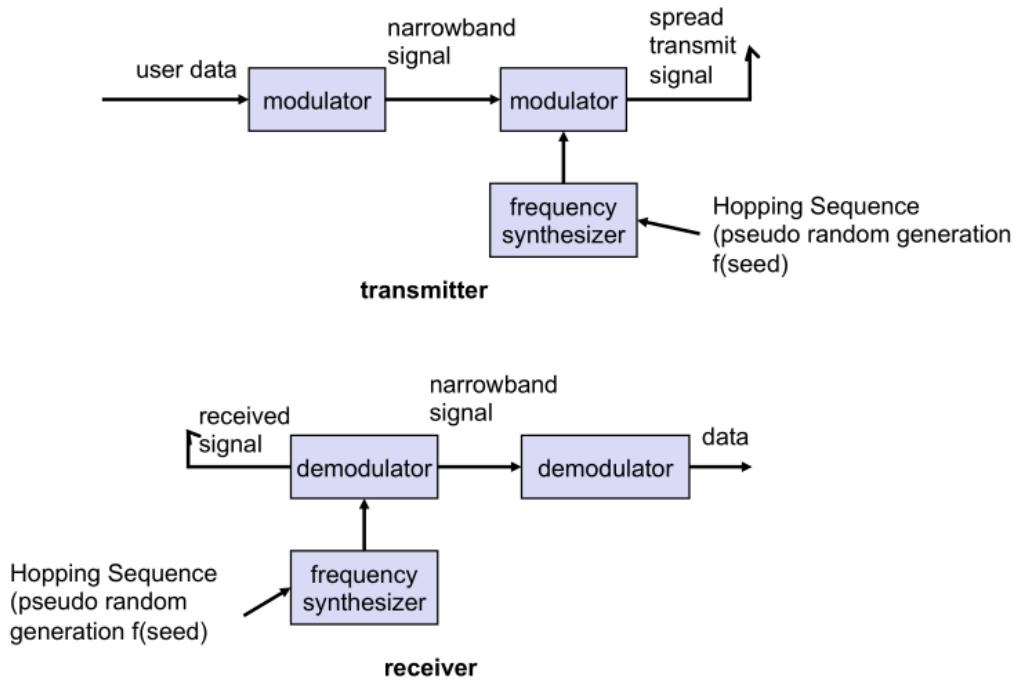


Frequency Hopping Spread Spectrum (FHSS)

Il frequency hopping consiste, come abbiamo visto, in cambi della frequenza di trasmissione successivi dati da una sequenza di numeri determinati pseudo casualmente. Vi sono **fast hopping** e **slow hopping**, semplicemente varia il rate di bit inviati tra un cambio di frequenza e l'altro

- vantaggi: limitato il frequency selective fading (interferenza "selettiva" su una frequenza), implementazione semplice e utilizzo di una porzione esigua di spettro al momento t
- svantaggi: meno robusto di DSSS, meno sicuro, quindi più semplice un detect del segnale





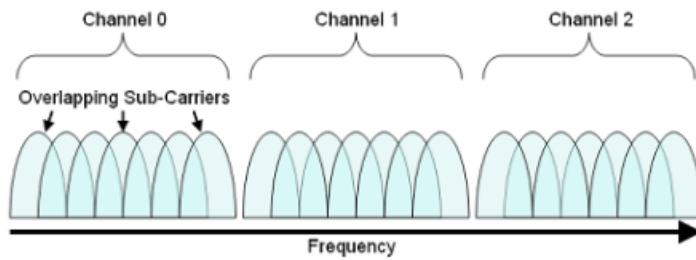
- A lato transmitter, il segnale modulato sul carrier viene rimodulato per fittare sulla frequenza corretta prevista dall'algoritmo pseudocasuale
- A lato receiver, quando viene demodulato il canale viene suggerito il canale corretto con cui effettuare la demodulazione

OFDM

Tecnica di trasmissione che sfrutta canali adiacenti in una banda → 20 Mhz divisi in sottobande → ogni banda suddivisa in 52 subcarriers, di cui 4 utilizzati per gestione della trasmissione (pilot), e 48 utilizzati per trasferire i dati (un simbolo per subcarrier)

Vediamo un po' tutti principi che stanno alla base di questo sistema

Se il Frequency Division Multiplexing sfrutta canali separati non overlapping (evita che le trasmissioni si sovrappongano), l'OFDM divide la banda in canali di frequenze overlapping, senza spazi di guardia

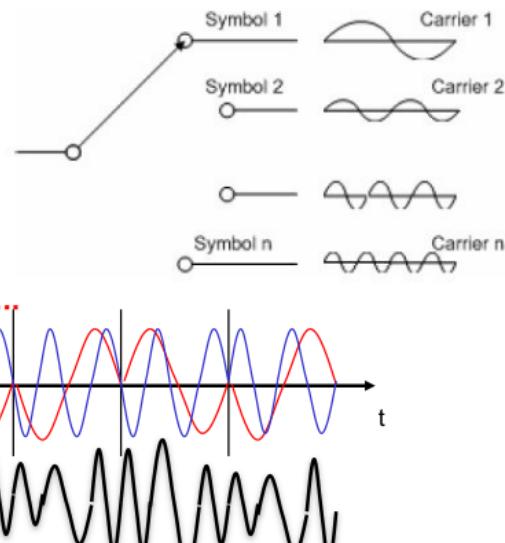


Un canale OFDM ospita da 128 to 2048 sub-carriers, per una bandwidth di conseguenza variabile tra 1.25 MHz e 20 MHz

Subcarrier adiacenti sono tra loro ortogonali → questo permette trasmissioni simultanee su non overlapping → è possibile estrarre tramite IFFT (Inverted Fast Fourier Transformation) uno dei segnali dalla marmellata di segnale OPPURE comporre il segnale tramite le varie frequenze uscenti da ciascun canale (processo inverso)

L'ortogonalità tra i segnali si nota dal grafico sotto: dove il segnale rosso ha un picco, gli altri (in questo caso uno soltanto) ha intensità nulla, e idem per quello blu, e per gli altri non visualizzati

- **C1 (1 Hz):** 1 1 1 -1 1 1 -1 1 1 -1 ...
- **C2 (2 Hz):** 1 1 -1 1 1 -1 -1 -1 1
- **C3 (3 Hz):** -1 1 1 1 1 -1 -1 1
- **C4 (4 Hz):** -1 -1 1 -1 -1 1 1 -1



- **OFDM encoding: ≈ 250.000 phase modulations per second**

Data Rate (Mbps)	modulation	Bits coded per phase transition	R = fraction of carriers used for convolution	Length of 1 symbol at the given data rate (#subcarriers * bits coded per symbol)	Data bits encoded in 1 symbol
6	DBPSK	1	1/2	48	24
9	DBPSK	1	3/4	48	36
12	DQPSK	2	1/2	96	48
18	DQPSK	2	3/4	96	72
24	16-QAM	4	1/2	192	96
36	16-QAM	4	3/4	192	144
48	64-QAM	6	2/3	288	192
54	64-QAM	6	3/4	288	216

OFDM codifica 250.000 modulazioni di fase al secondo (250.000 simboli trasmessi al secondo)

La frazione di carrier usati per la convoluzione indica quanti carrier sono effettivamente utilizzati per l'invio di dati

I bit coded per phase transition indicano quanti bit sono rappresentati da un simbolo nella codifica

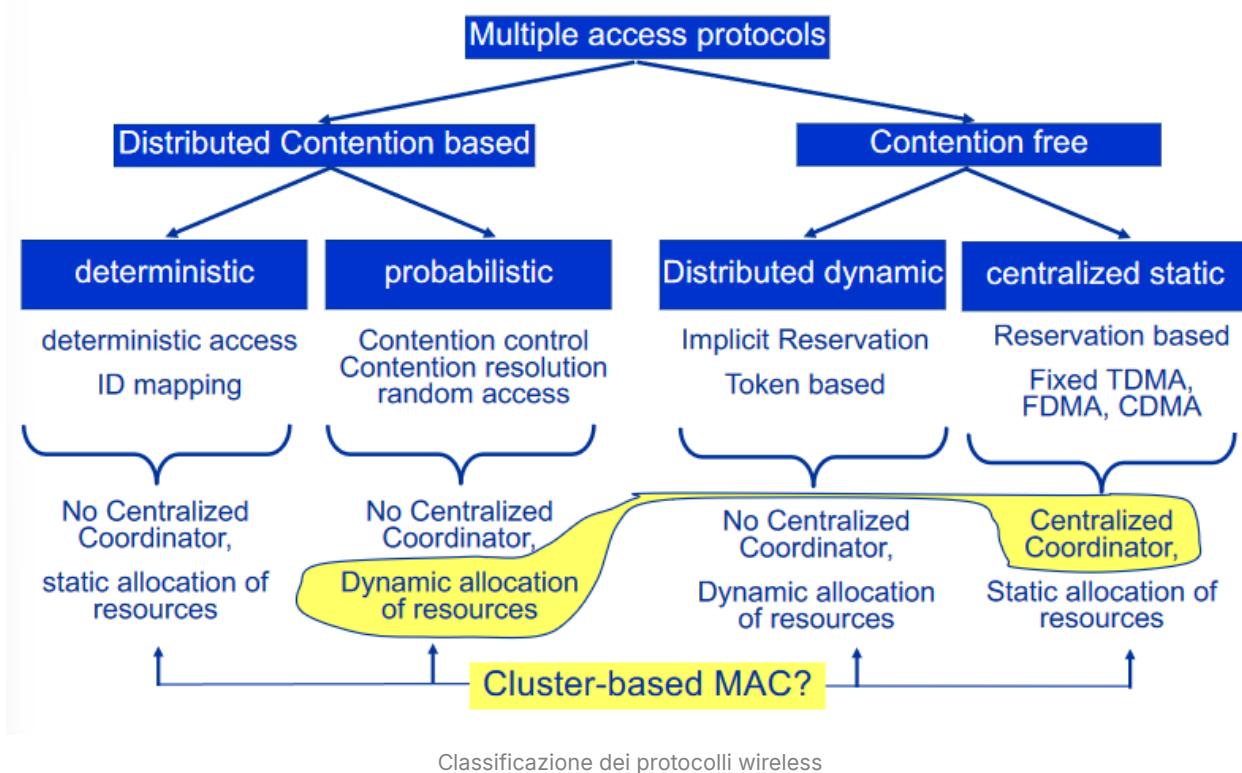
Protocolli MAC wireless

Collisioni di segnali wireless

Le collisioni hanno effetto distruttivo sul ricevitore:

- Spreco di potenza e di canale (canale occupato per una trasmissione mai arrivata a destinazione che deve essere nuovamente occupato per rinviare il messaggio)
- Nelle reti cablate, il rilevamento delle collisioni è pratico e può essere gestito, ma nelle reti wireless diventa più complicato
- Per limitare le collisioni si cerca di controllare la contesa del canale dal lato mittente
- **effetto cattura:** → si verifica quando un segnale più forte sopprime i segnali più deboli, consentendo al dispositivo ricevente di decodificare e demodulare solo il segnale più forte
 - può essere sfruttato questo meccanismo
- **dominio di collisione** → insieme di nodi che condividono lo stesso canale di comunicazione

Wireless MAC protocols' classification



Protocolli time domain first

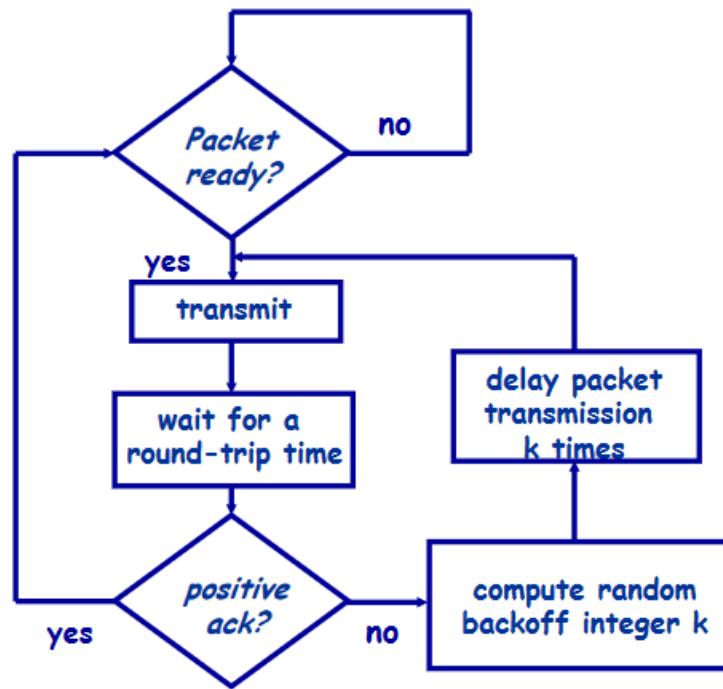
time domain first → il tempo è la dimensione primaria utilizzata per organizzare e gestire le trasmissioni dei dati

- Maggior enfasi su controllo della latenza, il ritardo e la sincronizzazione delle trasmissioni

ALOHA - puro

- Funzionamento
 - Viene spedito il pacchetto
 - Aspetto un Round Trip Time (RTT), il tempo impiegato da un pacchetto di dati per viaggiare dal mittente al destinatario e tornare indietro
 - Se ricevo un ack passo al prossimo pacchetto altrimenti genero un tempo random di attesa prima di riprovare → **backoff**

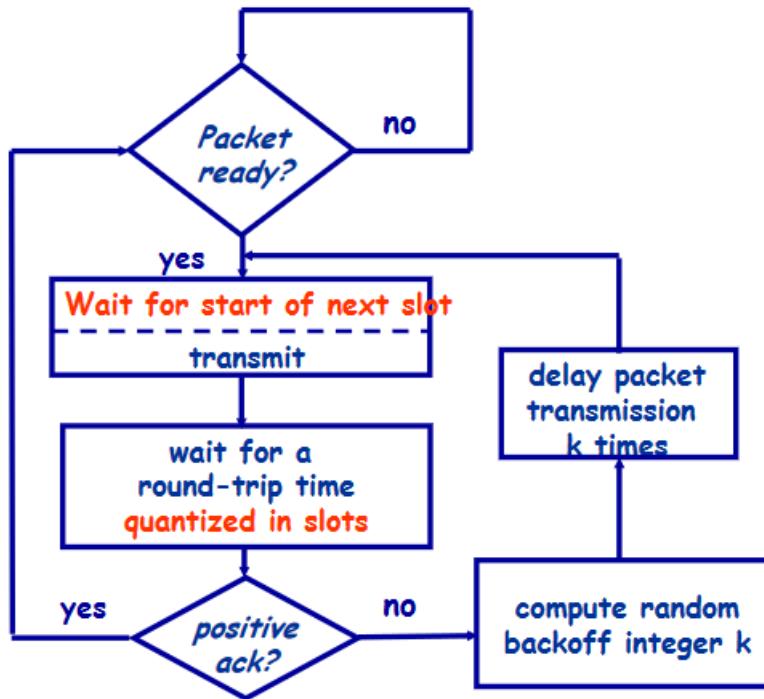
- Il tempo di vulnerabilità del frame è pari a 2 volte la sua dimensione
 - Ovvero la finestra durante la quale un frame può subire collisione
 - Rischio elevato di collisione



ALOHA - slotted

Tempo dell'algoritmo quantizzato in slot, si può iniziare a trasmettere solo all'inizio di uno slot → il tempo di vulnerabilità diminuisce a un solo frame, poiché la collisione avviene solo se la trasmissione avviene nello stesso slot di tempo

Per il resto, è in tutto e per tutto uguale all'algoritmo sopra

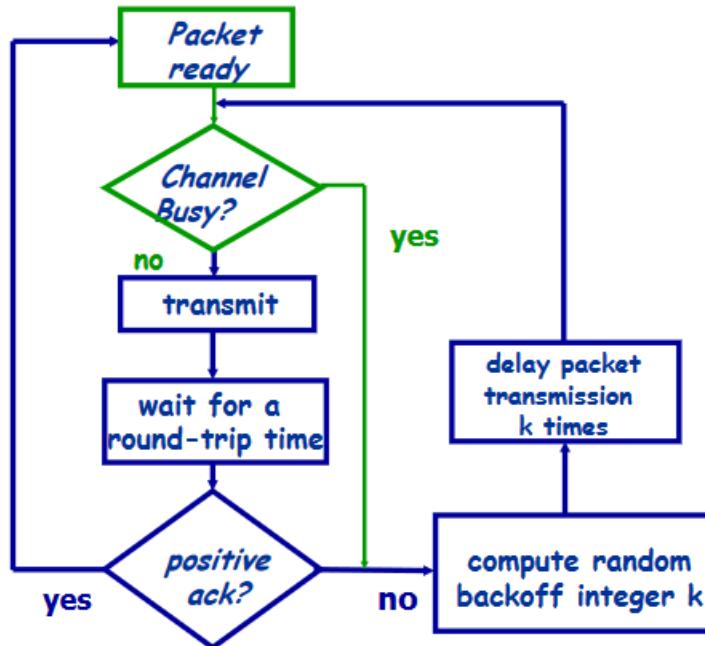


CSMA (Carrier Sens Multiple Access) puro e slotted

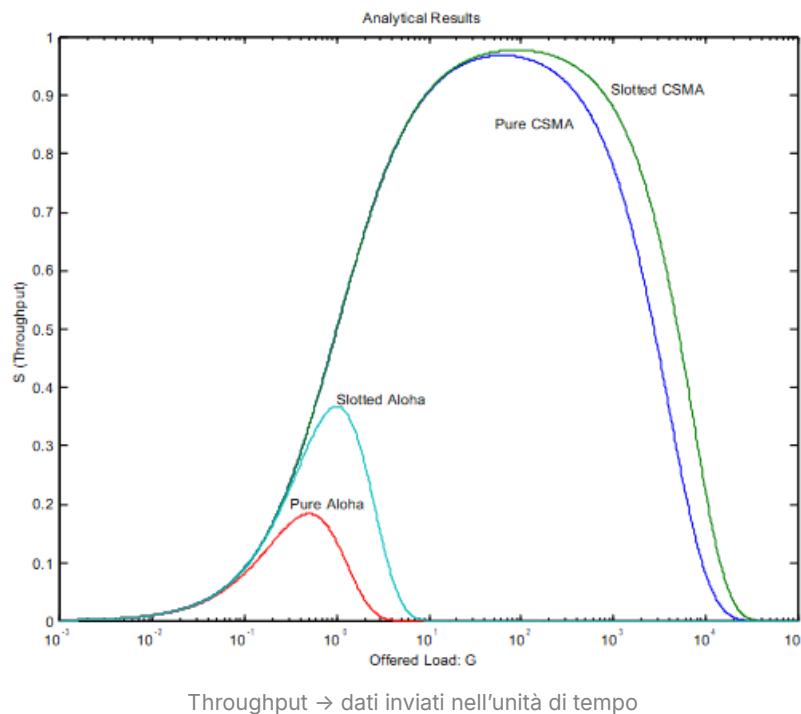
Basato su chip radio che ascoltano il canale prima di trasmettere

- funzionamento
 - Simile all'ALOHA puro, ma la trasmissione dipende dal chip che ascolta il canale → se il canale è occupato, la trasmissione è ritardata
- Il tempo di vulnerabilità è 2 volte il delay di propagazione

Introducendo gli slot la vulnerabilità è pari ad una volta sola al delay di propagazione



Throughput comparison



Protocolli space domain

I **protocolli space domain** sono protocolli di comunicazione che si concentrano sull'uso efficiente dello spazio fisico all'interno di una rete per ottimizzare la trasmissione dei dati

- maggior enfasi su distribuzione dei nodi, copertura del segnale, riduzione dell'interferenza e massimizzazione dell'efficienza spaziale

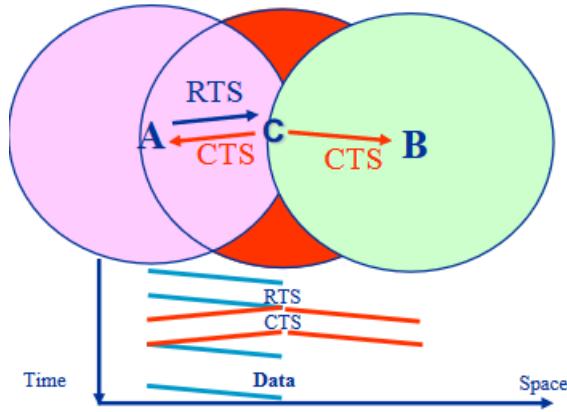
Problema dei terminali nascosti (hidden terminals) → due terminali A e B inviano frame a un terminale C nel mezzo, A e B non si percepiscono ma su C avviene una collisione

Problema dei terminali esposti (exposed terminals) → se un nodo A sta trasmettendo a un nodo B, un nodo C che si trova nel raggio di trasmissione di A potrebbe essere esposto se non trasmette nonostante potrebbe farlo senza interferire con la comunicazione tra A e B

Meccanismo RTS/CTS

Protocollo space domain per risolvere il problema degli hidden ed exposed terminals

- funzionamento
 - Una stazione, per verificare se la trasmissione possa avere successo, invia un pacchetto RTS (request to send) in attesa di un pacchetto di ritorno CTS (clear to send)
 - Il pacchetto CTS viene mandato in broadcast a tutti i nodi collegati al destinatario della comunicazione (colui che riceve RTS) → in questo modo i nodi vicini al receiver sono informati del fatto che tale nodo è occupato e, per evitare collisioni, aspettano a trasmettere
- problemi:
 - overhead aggiuntivo → bisogna inviare ogni volta RTS e CTS
 - Ritardi di latenza dovuti allo scambio di pacchetti RTS e CTS prima dei dati effettivi
 - Non considera asimmetrie dei canali → se A invia RTS a B ma B ha potenza minore e non riesce a raggiungere A, A non si vede mai arrivare il CTS



MACA

Implementato tramite uno slotted RTS/CTS senza carrier sensing

- RTS/CTS da 30 byte ciascuno e slot di tempo pari alla durata di invio di un pacchetto RTS o CTS
- Il carrier sensing non è presente (come in RTS/CTS) perché la contesa è sul ricevente, non sul mittente
- Allow exploitation of concurrent spatial transmission if the receiver is not exposed to two hidden transmitter terminals??

MACAW

Implementato tramite uno slotted RTS/CTS senza carrier sensing e con ACK

- introduce l'ACK (RTS – CTS – DATA – ACK) → ritrasmissione efficiente dei dati
- meccanismo di backoff adattivo per regolare dinamicamente la lunghezza del ritardo prima di ritrasmettere in caso di collisioni

NOTA: sia mittente che destinatario si comportano da destinatario durante la trasmissione del pacchetto → nessuno sfruttamento spaziale concorrente del canale

FAMA

Implementato tramite uno slotted RTS/CTS con carrier sensing e con ACK

Il carrier sensing agisce prima dell'invio di un RTS

Protocolli ad-hoc

Le reti **ad-hoc multi-hop** sono reti in cui i dispositivi comunicano tra loro direttamente, senza la necessità di un'infrastruttura di rete centralizzata come un router o un access point

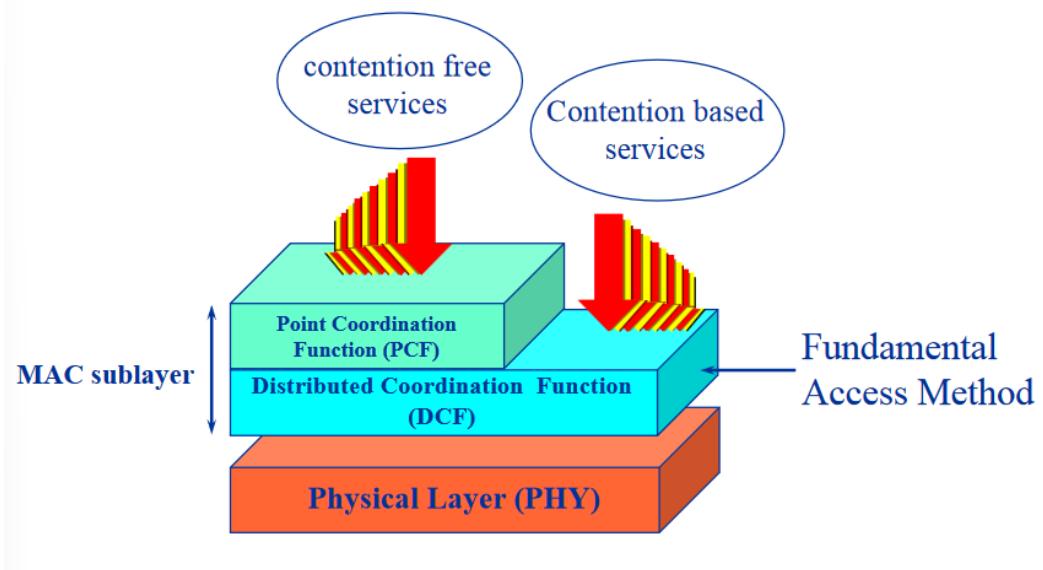
I vari dispositivi della rete agiscono sia come nodo di trasmissione che come nodo di ricezione comunicando entro il loro raggio di trasmissione, raggiungendo nodi più distanti della rete saltando tra un dispositivo e l'altro.

Queste reti sono utilizzate, ad esempio, in scenari di emergenza, reti militari sul campo, sensori mobili ecc...

I protocolli ad-hoc sono basati sulla combinazione data + ack caratteristica del protocollo TCP

- problemi di auto-contesa: e se il messaggio che devo inviare si scontra con l'ACK del messaggio precedente? O con altri pacchetti nel flusso?
- varie soluzioni

IEEE 802.11 Wireless LAN



Il protocollo MAC IEEE 802.11 ha un'architettura costruita su due funzioni coesistenti, **DCF** e **PCF**

Vengono poi utilizzati altri frame di controllo chiamati beacon frames

PCF

La cosiddetta point coordinated mode è un servizio opzionale che coesiste con la DCF

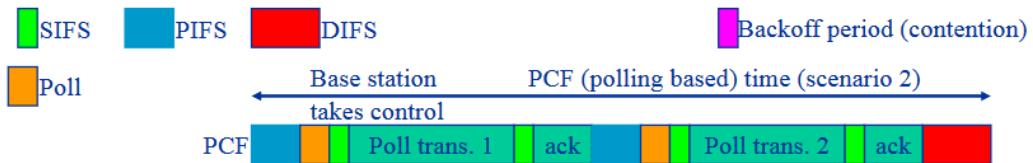
- in un access point, si occupa di gestire le stazioni che appartengono alla sua lista di accesso al canale
- garantisce un accesso non conteso al canale da parte delle varie stazioni a lui collegate

Questo accesso non conteso avviene innanzitutto grazie a una “**poll**”, ovvero un’interrogazione dei nodi legati all’access point, per determinare se hanno qualcosa da trasmettere (una poll interroga un nodo della rete)

Ogni stazione connessa al canale effettua un carrier sensing prima di accedere al canale

IFS → Inter Frame Spaces → possono essere Short (SIFS), Point (PIFS) o Distributed (DIFS) in ordine di ampiezza

- Dopo un SIFS, solo la stazione interrogata (detta polling station) dall’access point può trasmettere (o inviare una risposta ACK)
 - è quindi sostanzialmente il tempo che intercorre tra una trasmissione e l’altra di un pacchetto da parte di uno stesso nodo di rete
- Dopo un PIFS, solo il punto di accesso può trasmettere (e PCF prende il controllo) → serve a garantire il funzionamento della PCF sull’access point prima che il controllo passi al livello sottostante, alla DCF
- Dopo un DIFS, ogni stazione può trasmettere mediante la DCF

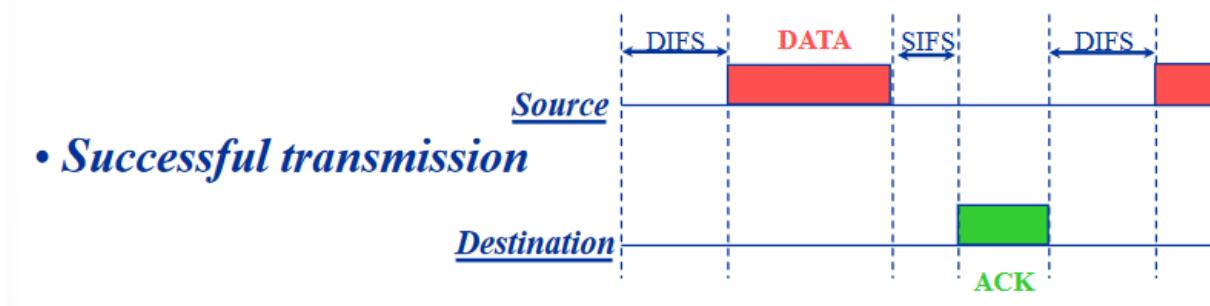
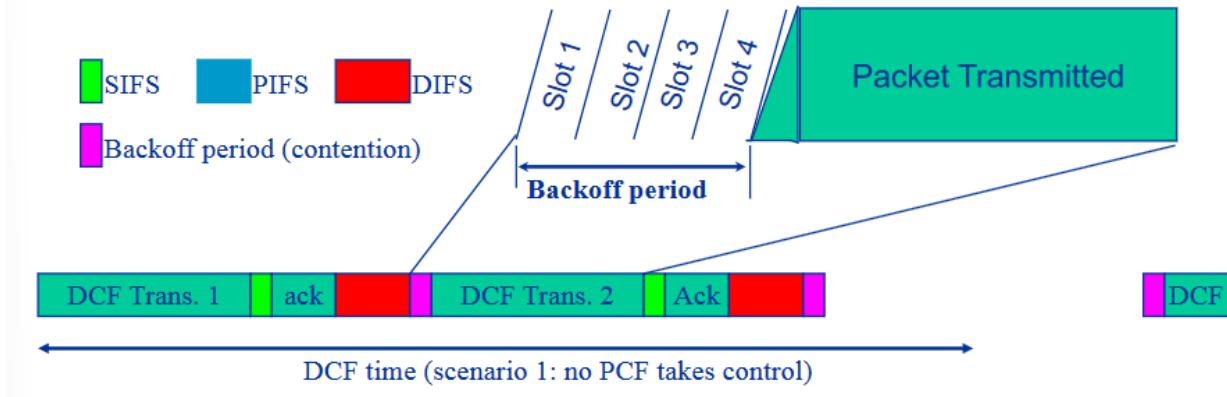


DCF

Il modello di accesso al canale è CSMA/CA (CSMA visto sopra, protocollo time domain first, CA sta semplicemente per Collision Avoidance)

- carrier sensing
- backoff esponenziale basato su slot → il backoff è esponenziale sui multipli del tempo di uno slot
 - approfondito dopo

- no Collision Detection, ma solo Collision Avoidance
- CSMA/CA utilizzato anche in Ethernet



Backoff time (da rivedere)

$$\text{BackoffTime}(i) = (CW_i \cdot \text{random}()) \cdot \text{SlotTime}$$

CW_i = dimensione della **finestra di contesa** all'i-esimo tentativo di trasmissione → la finestra di contesa è il tempo che intercorre tra un tentativo e l'altro di trasmissione del pacchetto (in slot)

- Dopo ogni DIFS, se il mezzo è libero, il backoff time viene decrementato di uno slot time per ogni slot durante il quale non si osservano attività
- Finché il mezzo è occupato, la procedura di backoff è sospesa
- random presumo serva per sfalsare il BackoffTime dei vari dispositivi
- Un nuovo tentativo di trasmissione avviene quando il BackoffTime raggiunge lo 0

Routing e trasporto wireless

Introduzione

In sistemi wireless, le reti non sono più vincolate a una topologia statica, ma hanno una forma che cambia dinamicamente visto che i dispositivi sono dotati di mobilità

Come ragioniamo in topologia dinamica? Come capiamo se un determinato host è raggiungibile all'interno della rete?

Bisogna (spesso) utilizzare un protocollo di routing che ci trovi la strada tra mittente e destinatario

Non c'è un protocollo unico adatto a tutte le situazioni

- **protocolli proattivi** → Questi protocolli mantengono costantemente informazioni aggiornate sulla topologia della rete e sui percorsi verso tutte le destinazioni possibili (NON LI AFFRONTIAMO)
- **protocolli reattivi** → In questi protocolli, i nodi reagiscono solo quando è richiesto un percorso verso una destinazione, quindi vi è una ricerca dinamica della route al momento della richiesta (TUTTI QUELLI A SEGUIRE)

I protocolli che seguono sono protocolli ad-hoc → i nodi si comportano sia come mittenti/destinatari, sia come ripetitori per i nodi più distanti

Flooding

Sostanzialmente "inondazione" di pacchetti

S vuol inviare un pacchetto a D → Inserisce nell'header del pacchetto l'indirizzo mittente e destinatario e invia il pacchetto in tutte le direzioni (broadcast)

- NOTA: ovviamente senza ack, non si può aspettare degli ack da un broadcast perché farebbero collisione

il pacchetto passa al livello mac di ogni dispositivo che riceve il pacchetto

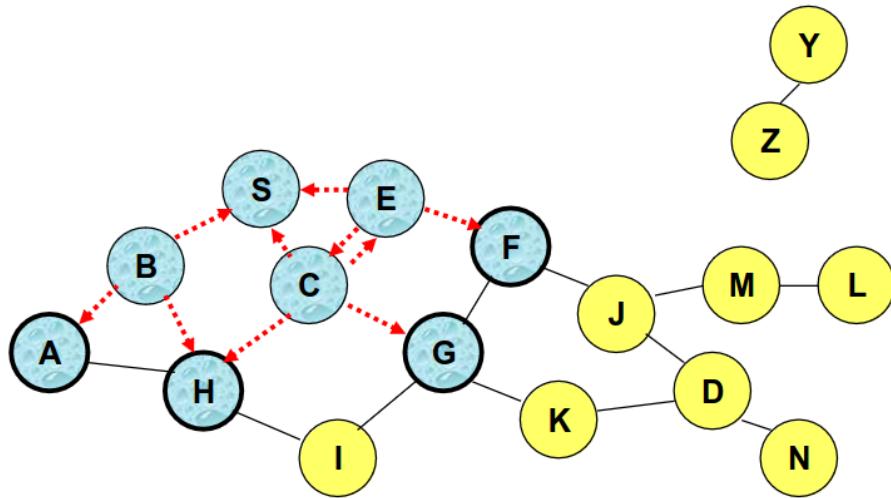
- se conoscono dove si trova il destinatario, inviano il pacchetto
- se il destinatario è sconosciuto, viene inviato nuovamente un broadcast

Il processo è iterato finché non è raggiunto il destinatario, che può ricevere contemporaneamente il pacchetto da più strade differenti OPPURE potrebbe non riceverlo mai se i pacchetti collidono

D non fa broadcast perché è il destinatario, ma solo lui sa che il pacchetto è arrivato a destinazione, quindi tutti gli altri nodi della rete continueranno a inoltrare il pacchetto

Come risolvere in caso di cicli?

- il pacchetto ha un time to live, dopo un certo numero di passi il pacchetto muore
- Si assegna un id al pacchetto, se il pacchetto è già stato lanciato in broadcast da un nodo non viene lanciato nuovamente



Vantaggi e svantaggi (soprattutto svantaggi):

- È un modo molto semplice
- Può essere utile se ci sono poche trasmissioni di piccoli pacchetti e la topologia cambia molto frequentemente → totalmente assenti tempi di discover della route da S a D
- Non è un modo efficiente perché vengono dati tantissimi contributi inutili → tempo di utilizzo del canale sprecato
- il metodo è potenzialmente reliable, perché il pacchetto arriva verosimilmente più di una volta a destinazione, ma potenzialmente non reliable, perché il broadcasting implica un numero molto elevato di collisioni e di pacchetti persi → tutto un po' a caso
- Rischio di non delivery se un nodo collo di bottiglia non collabora (o se ha problemi di vario genere)
- Rischio di scadenza di time to live del pacchetto

Dynamic Source Routing (DSR)

Basato su route discovery tramite **pacchetti di controllo** → flooding di un pacchetto **Route Request** (RReq) prima dei dati veri e propri

La sorgente invia in broadcast una RReq → ogni nodo che riceve la RReq fa broadcasting del pacchetto e fa un append del nodo visitato alle informazioni di tale pacchetto → ogni RReq traccia la strada che

ha percorso

- potenziali collisioni tra pacchetti RReq
- chi ha già ricevuto e rinvia la RReq, non ripete la procedura quando arriva un'altra RReq

Il nodo destinazione, all'arrivo della prima RReq, non rinvia il pacchetto ma invia una **Route Reply** (RRep), che include la strada da sorgente a destinazione ricavata dalla RReq (la RRep in se fa la strada opposta partendo dalla destinazione e arrivando alla sorgente)

NOTA: la Route Reply viene inviata direttamente solo se i link attraversati dalla RReq sono bidirezionali, altrimenti ricomincia un processo speculare per ottenere la strada di ritorno

- **piggybacking** → viene lasciato spazio sul pacchetto della RReq per inserire la RRep, per limitare il numero di pacchetti di controllo in giro per la rete
- Se utilizziamo il protocollo IEEE 802.11 MAC per l'invio dei dati, i link devono obbligatoriamente essere bidirezionali, o l'ACK non può essere utilizzato

La RRep viene cashata da S una volta arrivata → ogni volta che viene inviato un pacchetto a D, viene inclusa nell'header → tutti i pacchetti intermedi consultano la source route per capire dove inoltrare il pacchetto



Caching - ottimizzazione

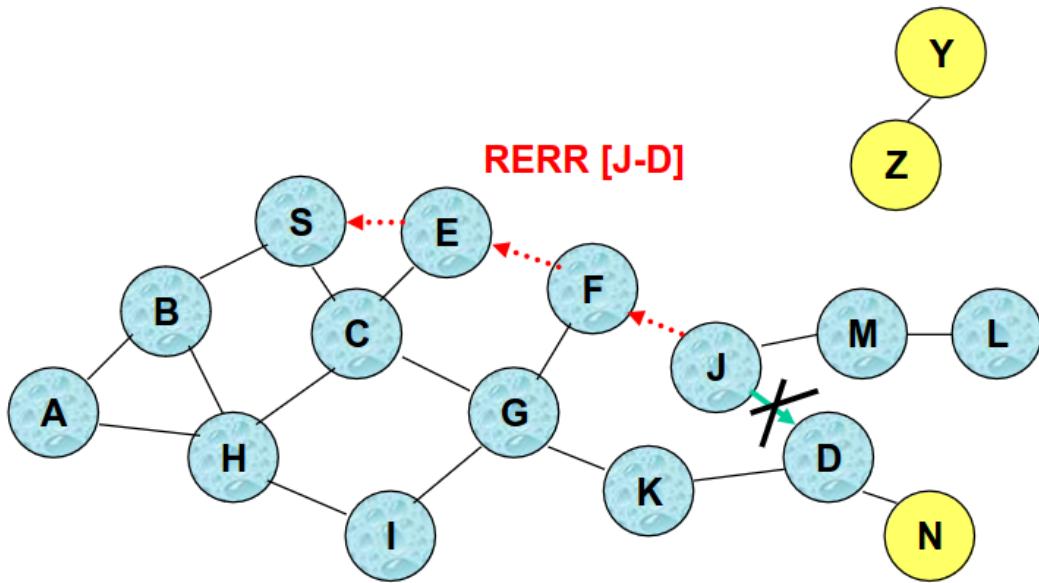
Non viene solo utilizzata la route scoperta in S per raggiungere D:

- Quando il nodo S trova il percorso [S, E, F, J, D] per il nodo D, il nodo S apprende anche il percorso [S, E, F] per il nodo F.
- Quando il nodo K riceve la RReq [S, C, G] destinata al nodo, il nodo K apprende il percorso [K, G, C, S] per il nodo S.
- Quando il nodo F inoltra la RRep [S, E, F, J, D], il nodo F apprende il percorso [F, J, D] per il nodo D.
- Quando il nodo E inoltra i dati [S, E, F, J, D], apprende il percorso [E, F, J, D] per il nodo D.
- Un nodo può anche apprendere un percorso quando intercetta pacchetti di dati

NOTA: Il nodo X, al ricevimento di una RReq per un certo nodo D, può inviare una RRep se il nodo X conosce un percorso per il nodo D.

Sfruttare al massimo il caching garantisce un route discovery più veloce e riduce la propagazione di RReq

Route Error (RErr) → quando un invio di dati fallisce e una route si rivela fallace (la topologia della rete si è modificata) bisogna effettuare nuovamente un'operazione di route discovery per il nodo non raggiungibile → per comunicare questo, il nodo che si accorge dell'impossibilità di inviare un pacchetto nella route avvisa tramite RErr tutti i pacchetti della route all'indietro



Tentativo di inviare dati nella route [S-E-F-J-D] con fallimento in J-D

Vantaggi e svantaggi

- Route salvate solo tra nodi che devono comunicare → overhead ridotto
- Molto efficiente grazie al caching
- Bisogna prestare attenzione alle collisioni tra pacchetti di controllo e al potenziale flooding di RReq
- Dimensione dell'header proporzionale alla route length → può diventare significativo in reti espansse
- Da gestire il Route Reply Storm → quando si manda una RReq, se molti nodi mandano una Reply perché hanno la route in cache la rete si riempie di RRep
- Rischio di diffusione di RRep obsolete, per come funziona il caching → servono meccanismi aggiuntivi

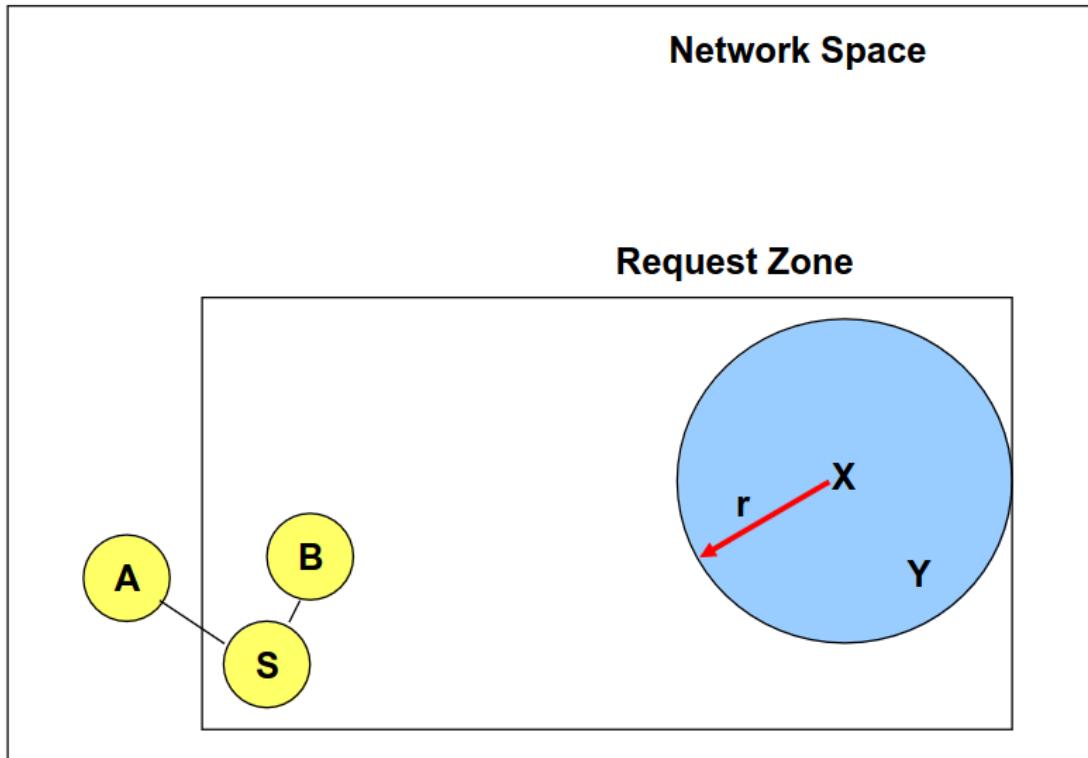
Ci servono protocolli che riducano il numero di pacchetti di controllo in rete

Location-Aided Routing (LAR)

Sfrutta informazioni sulla posizione dei nodi per limitare il request flood (es. via GPS)

- **Expected zone** → regione in cui ci aspettiamo di trovare la posizione attuale della destinazione → determinata a partire da vecchie posizioni della destinazione e la sua velocità

- **Request zone** → zona in cui è limitata la ricerca della destinazione → comprende posizione del sender e expected zone



- Solo i nodi presenti nella request zone inoltrano la RReq → la request zone è esplicitata nella RReq e tutti i nodi conoscono la loro propria posizione fisica, per poter determinare se sono dentro o fuori
- Se una route discovery fallisce, viene ampliata la request zone e ritentata
- Il resto del protocollo è simile a DSR

Vi sono due varianti di LAR:

1. **Adaptive Request Zone** → ogni nodo ha il potere di modificare la request zone a seconda delle informazioni che possiede e della sua posizione nella rete
2. **Implicit Request Zone** → le RReq vengono inoltrate solo ai nodi che sono ritenuti più vicini all'expected zone

Si può inoltre variare il modo in cui si conosce la posizione dei nodi della rete → inizialmente ogni nodo conosce solo la propria posizione, e vengono man mano scoperte solo tramite route discovery (**e salvate in qualche cache?**)

Variazioni:

- piggybacking delle informazioni di posizione di un nodo su qualsiasi messaggio che attraversa quel nodo
- un nodo comunica la sua posizione attivamente ai nodi limitrofi

Vantaggi e svantaggi

- riduce il flood di pacchetti di controllo
- riduce l'overhead
- necessario che ogni nodo conosca la propria posizione fisica
- non tiene conto di eventuali ostruzioni nella radiotrasmissione

Variante - GEDIR

- assumiamo come conosciuta la posizione della destinazione
- ogni nodo conosce la posizione dei nodi adiacenti
- ogni nodo inoltra al pacchetto ai nodi adiacenti più vicini alla destinazione

problema: rischio di infilarsi in vicoli ciechi in caso di ostruzioni

Ad Hoc On-Demand Distance Vector Routing (AODV)

Identico a DSR per la gestione di RReq e RRep, ma si vuole limitare l'header → routing tables ai nodi

Differenze:

- Vengono instaurati dei link durante il passaggio della RReq tra un nodo e il successivo → utilizzati dalla RRep per tornare indietro, così da non avere bisogno di headers (**da controllare**)
 - timeout che elimina il link in un tempo utile perché la RRep possa tornare alla sorgente (a quel punto il link non serve più)
- Routing tables generate dinamicamente durante la trasmissione della RREP ogni nodo lungo il percorso memorizza le informazioni sul percorso nella sua tabella di routing. Queste informazioni includono l'indirizzo del nodo di destinazione, il prossimo salto verso il nodo di destinazione e il numero di sequenza più recente associato al percorso (numero di sequenza = numero che indica quanto è nuova la sequenza, serve a preferire sequenze aggiornate rispetto a quelle obsolete)
 - Nodi intermedi possono inviare una RRep se conoscono una route più recente rispetto a quella conosciuta dal nodo sorgente (**quando viene inviata? durante l'invio dei dati? perché se S sta esplorando una nuova route, sicuro il suo numero di sequenza è più alto?**)

- Creando i link prima dell'invio dei dati facciamo in modo di non dover includere un header nei dati da inviare
- timeout che elimina il link se non viene utilizzato in una finestra di tempo
- AODV assume link bidirezionali
- AODV conserva nelle routing tables soltanto un link per destinazione, a contrario di DSR dove i nodi possono tenere in cache più route per la stessa destinazione

Mobile IP

Se cambia il router a cui un dispositivo è connesso, cambia l'indirizzo di rete del dispositivo → come fa a non cadere la comunicazione?

Vari protocolli per mobile IP: mobile IPv4, mobile IPv6, mobile IPv4 gerarchico...

Home agent

Mobile IPv4 lascia nella **home network** (rete in cui siamo inizialmente e in cui siamo per la maggior parte del tempo) un processo acceso chiamato **Home Agent**

- Tale processo non termina ma sta in ascolto anche nel momento in cui mi disconetto
- Al momento della riconnessione con un altro router viene inviato un pacchetto all'Home Agent che comunica la mia nuova posizione → HA ascolta i dati in arrivo che sarebbero stati inviati a me (ma io non ci sono), e me li manda conoscendo il mio nuovo IP
- Se mi sposto ancora mi connetto al nuovo router, aggiorno l'Home Agent dicendo il mio nuovo indirizzo → cambia solo la destinazione dell'inoltro

Problema: i pacchetti che ha inviato HA mentre mi spostavo dal vecchio al nuovo router come li gestisco?

- Rimando i pacchetti senza ACK
- Con un **Foreign Agent** → il mio collegamento prima rimane aperto come Foreign agent e i dati seguono il percorso HA → FA → IP attuale

Per evitare un giro dell'oca dopo tanti ricollegamenti, quando ho una fase di respiro nella comunicazione, chiedo all'Home Agent di creare una scorciatoia diretta a dove sono, recapitando nuovamente i pacchetti senza ACK, e chiudendo tutti i FA

Vantaggio: privacy → a prescindere da dove sei, in rete appari nella posizione dell'home agent

NOTA. le comunicazioni con HA sono criptate e autenticate

Come viene determinato HA?

TCP nelle reti mobile ad-hoc

Protocolli stop and wait

Un protocollo stop and wait attende la corretta ricezione di un pacchetto prima della trasmissione del successivo

La gestione della perdita e dell'alterazione dei dati ricevuti avviene tramite:

- feedback ACK/NAK (negative ACK, conferma della perdita del pacchetto) dal ricevente
- trasmissione subito dopo l'arrivo del ACK/NAK
- timeout per la gestione della perdita del pacchetto
- numeri di sequenza (per disambiguare la ritrasmissione a seguito della perdita dell'ACK, ovvero poter mantenere l'ordine di trasmissione dei pacchetti)

Ma quali sono le prestazioni del sistema? Vi sono alcuni parametri per capirlo

1. **Round Trip Time** (RTT) → tempo che va dall'invio di un pacchetto alla recezione dell'ACK relativo
2. **T_tx_i** → tempo di trasmissione (tx sta per trasmissione) del pacchetto i, ovvero tempo necessario per trasmettere un pacchetto nel canale

$$T_{tx_i} = \text{Size(packet } i) / \text{channel bit rate}$$

3. **Channel Utilization** → tempo in cui il canale di comunicazione è effettivamente utilizzato per la trasmissione di dati rispetto al tempo totale di ciclo

$$\text{Channel Utilization} = T_{tx_i}/(RTT + T_{tx_i})$$

es. invio segmenti di 1000 Byte, rete a 1 Gbps, con RTT 30 ms

$$T_{tx_i} = 8000 \text{ bit} / 2^{30} \text{ bps} = 8 \text{ microSec}$$

$$\text{Channel Utilization} = 8 / (8+30000) = 266 \text{ Kbps (basso utilizzo!)}$$

$$\text{Prodotto della rete} = \text{Bandwidth} * \text{Delay}$$

Il prodotto della rete calcola la quantità di dati presenti nella rete in un determinato momento, e si ottiene moltiplicando la capacità della rete (**Bandwidth**) con il tempo necessario a un pacchetto per raggiungere il destinatario (**Delay**)

Se il mezzo della rete è sfruttato al massimo ritmo sostenibile, il destinatario riceve costantemente dati senza interruzioni → massimo throughput (quantità di dati nel tempo) del sistema

Gestione del canale a pipeline

Non si attende di ricevere l'ACK dei segmenti precedenti prima di inviare i segmenti successivi (se disponibili):

- Aumentano i numeri di sequenza man mano che i pacchetti sono inviati (ACK non ambigui)
- Necessità di buffer su mittente e destinatario

Come ci comportiamo in caso di perdita di segmenti o ACK/NAK?

Protocollo Go-Back-N

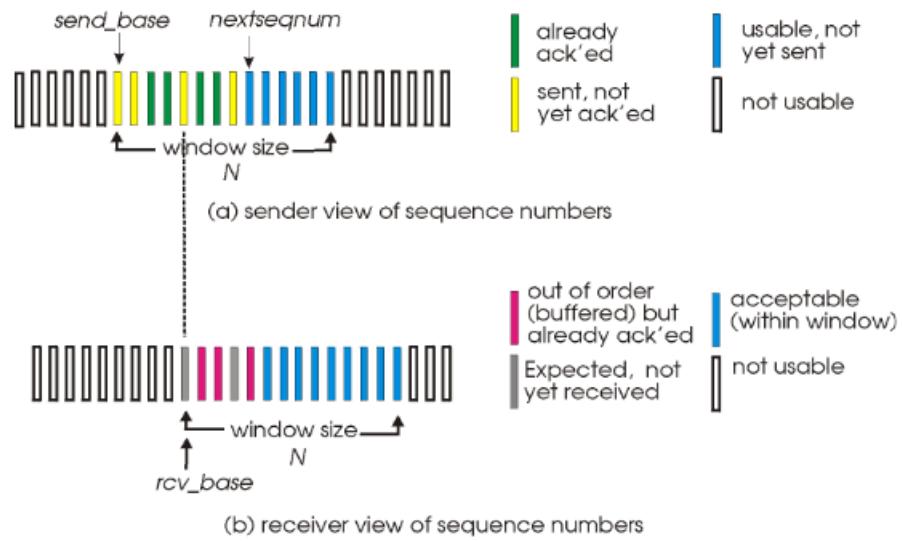
Viene scelta una massima finestra (scorrevole) di segmenti in sospeso (trasmessi ma non ricevuti)

- Il destinatario invia ACK solo se il segmento è quello atteso (oppure ripete ultimo ACK valido) e non inserisce in buffer segmenti fuori ordine
- NOTA: gli ACK sono cumulativi per il mittente → un ACK ricevuto per il pacchetto N conferma la ricezione di tutti i pacchetti prima di N
- Timer per la ritrasmissione → in caso di timeout: ritrasmissione di tutti i segmenti successivi della finestra
- numero di pacchetti della finestra = min(capacità buffer destinatario, capacità di smaltimento del router più lento)
 - non può essere un numero troppo grande per evitare saturazione della rete in caso di trasmissione
- Problema: dati corretti scartati perché fuori sequenza



Protocollo Selective Repeat

- Finestra di ricezione non superiore a un buffer
- invio di ACK specifici anche fuori ordine di ricezione MA dentro alla finestra di ricezione → la finestra si sposta mano a mano che i pacchetti più vecchi vengono confermati
- invia Ack in caso di segmenti ripetuti anche precedenti nel range [Expected_seq#-N... Expected_seq#]





Nota su ridimensionamento della finestra di congestione

Ha senso in comunicazioni wireless dove spesso pacchetti vengono persi di resettare la finestra di congestione tutte le volte? Soluzione alternativa → ridimensionare

