DOMANDE RETI:

Indice generale delle domande:

Capitolo 2:

1. **Cosa si intende per serie di Fourier.**
2. **Bitrate e Baudrate.**
3. **Descrivere i vari tipi di cavo e confrontarli.**
4. **Caratteristiche e confronto fra i vari tipi di satellite, GEO,MEO,LEO.**
5. **Che cos'è la modulazione in frequenza.**
6. **Che cos'è la modulazione delta? (delta modulation)**
7. **Descrivere in dettaglio il GSM (Global System for Mobile communication)**
8. **Si descriva la tecnica CDMA (Code Division Multiple Access), possibilmente con un esempio.**
9. **Il GPRS: cos'è, pregi e difetti.**
10. **Handoff: cos'è e i vari tipi.**
11. **FDM, TDM, CDM: algoritmi per la selezione della banda.**
12. **QAM e QAM16.**

Capitolo 3:

13. **Che cos'è il byte stuffing.**
14. **Che cos'è il bit stuffing.**
15. **Numero di bit necessari per riconoscimento(correzione) degli errori di trasmissione.**
16. **Si descriva cos'è il CRC. Si calcoli inoltre il CRC di 10011101 usando il polinomio generatore di x^4+x+1.**
17. **Descrivere il protocollo stop and wait, pregi e difetti.**
18. **Cos'è il piggybacking.**
19. **Si descriva la tecnica del Sliding window.**
20. **Si descriva l'idea dei protocolli “go back N”, indicandone pregi e difetti.**
21. **Si descriva cos'è la tecnica del selective repeat.**

Capitolo 4:

22. **Descrivere la differenza fra ALOHA e ALOHA-SLOTTED.**
23. **Si illustri CSMA, indicandone pregi e difetti.**
24. **Basic bitmap.**
25. **Spiegare in cosa consiste il protocollo collision free binary countdown, pregi e difetti.**
26. **Spiegare che cos'è l'adaptive tree walk protocol.**
27. **Ethernet e i vari tipi di cavo.**
28. **Codifica Manchester.**
29. **Frame Ethernet.**
30. **Cos'è il binary exponential backoff?**
31. **Stazione nascosta e stazione esposta: cosa sono e come si comportano.**
32. **Bluetooth.**

Capitolo 5:

33. **Si descriva l'algoritmo statico Flooding.**
34. **Descrivere il distance vector routing.**
35. **Linkstate routing.**
36. **Choke bucket.**
37. **Token hop-by-hop.**
38. **Load shedding.**
39. **RED.**
40. **Reverse Path Forwarding.**
41. **Quality of Service.**
42. **Leaky Bucket, pregi e difetti.**
43. **Descrivere il token bucket, pregi e difetti.**
44. **Descrivere l'ARP.**
45. **Si descriva DHCP.**
46. **IPV6.**
47. **Elencare e descrivere brevemente i secondi (primi) 32b dell'header IPv4 (Ipv6).**
48. **CIDR.**

Capitolo 6:

49. **Si descriva l'header UDP.**
50. **Descrivere l'header del TCP/IP e commentarlo.**

Capitolo 7:

51. **DNS.**

Capitolo 8:

52. **Cos'è un cifrario a sostituzione.**
53. **Si l'algoritmo DES e triplo DES.**
54. **Si descriva il cipher block.**
55. **ECB.**
56. **Stream cipher.**
57. **RSA.**
58. **Sicurezza in 802.11**
59. **Si descriva la sicurezza del bluetooth.**
60. **Il replay attack.**
61. **Man in the middle.**
62. **DNS spoofing.**

RISPOSTE:

CAPITOLO 2: “LO STRATO FISICO”

1. **Cosa si intende per serie di Fourier.**

Le informazioni possono essere trasmesse via cavo variando alcune proprietà fisiche, come la tensione e la corrente. Rappresentando il valore di questa tensione o corrente, attraverso una funzione, f(t), e' possibile rappresentare le informazioni trasmesse via cavo.
Questa funzione e' composta da una serie infinita di somme di seni e coseni, ed e' in grado di rappresentare un segnale periodico e regolare.
Qualunque forma di onda vogliamo, possiamo dunque rappresentarla in onda grazie alla trasformata di Fourier!
Se tutti i componenti di Fourirer fossero attenuati in modo uniforme, il segnale risultante verrebbe ridotto in ampiezza, ma non risulterebbe distorto. Sfortunatamente, i mezzi di trasmissione non attenuano i componenti della serie di Fourier in modo uniforme, e ciò genera una distorsione. Di solito, le ampiezze sono trasmesse senza modifiche da 0 fino ad una certa frequenza, e attenuate per tutte le frequenze superiori a questo limite.
L'intervallo di frequenze trasmesse senza una forte attenuazione è chiamato -banda passante- .
Questa, è una proprietà fisica del mezzo di trasmissione, e di solito dipende dalla costruzione, dallo spessore e dalla lunghezza del mezzo.

Anche in un ipotetico canale perfetto, la velocità di trasmissione è limitata, e il limite massimo alla quantità di dati trasmissibili è definita dal Teorema di Nyquisit:
B=Banda
L=Livelli di segnale che usiamo.
Vmax= $2*B \log_2 L$

Dobbiamo inoltre considerare che è sempre presente del rumore(termico) all'interno del canale.
Il teorema di Nyquisit vale invece per un canale perfetto, privo di rumore(interferenze).
Il rapporto Segnale(S)/Rumore(R) è pari a S/N.

Il Teorema di Shannon, definisce il massimo data-rate, tenendo conto del rumore.
B=Banda
S/N=Rapporto segnale/rumore
Vmax=$B \log_2 (1+S/N)$

Sono inoltre presenti ulteriori problemi, oltre all'attenuazione e al rumore:
-Dispersione: E' il cambio della forma d'onda. Cambiano cioè le armoniche. Più lontano trasmetto, più la forma d'onda cambia!
Una possibile soluzione a ciò, è quella di utilizzare dei ripetitori di segnale.

2. **Bitrate e Baudrate.**

Il Data rate indica quanta informazione trasmettiamo al secondo.
-Bitrate: Indica quanti BIT trasmettiamo al secondo. La velocità trasmissiva viene indicata in bit/s. Il teorema di Nyquisit mette in relazione il bitrate con la banda disponibile:
B=Banda
L=Livelli di segnale(=simboli) che usiamo.
Vmax= $2*B \log_2 L$

Massimo Bitrate, tenendo conto del rapporto segnale/rumore:

Dobbiamo inoltre considerare che è sempre presente del rumore(termico) all'interno del canale. Il teorema di Nyquisit vale invece per un canale perfetto, privo di rumore(interferenze).
Il rapporto Segnale(S)/Rumore(R) è pari a S/N.

Il Teorema di Shannon, definisce il massimo data-rate, tenendo conto del rumore.
B=Banda
S/N=Rapporto segnale/rumore
Vmax=$B \log_2 (1+S/N)$

-Boudrate: indica quanti SIMBOLI trasmettiamo al secondo.
Bitrate $\subseteq$ Boudrate.

Differenza tra bitrate e boudrate:
Abbiamo un alfabeto di trasmissione che ha un certo numero di simboli.
Il Boudrate misura relativamente a quell'alfabeto.
Il Bitrate=Boud che utilizza 0 e 1 come alfabeto.

Esempio: Posso trasmettere un impulso usando 4 frequenze diverse. => alfabeto composto da 4 simboli. Ognuno di questi simboli porta quindi l'informazione di 2 bit (per rappresentare 4 simboli mi servono 2 bit). Il bitrate sarà quindi doppio rispetto al boudrate.

In generale:
$\log_2 (V)$ = Misura in termini di bit, esprimibile con un alfabeto composto da V simboli.
Es: $\log_2 (4) = 2$, cioè 2 bit per 4 simboli.

La relazione finale tra bitrate e boudrate:
Bitrate=Boudrate*$\log_2 (V)$

3. **Descrivere i vari tipi di cavo e confrontarli.**

Principalmente, esistono 3 tipi di cavo: Il classico doppino, il cavo coassiale e la fibra ottica.

Il Doppino: (Twisted Pair)

UTP=Unshilded Twisted Pair (Unshilded= non schermati, o bassa schermatura).
E' composto da due fili di rame isolati, spessi circa 1mm, avvolti uno intorno all'altro in una forma elicoidale, che ricorda un po' quella della molecola del DNA.
I cavi vengono intrecciati per evitare interferenze; cavi paralleli formano infatti un'eccellente antenna, mentre con cavi intrecciati i campi elettromagnetici generati dai due conduttori si annullano a vicenda. Il "twist" serve dunque ad evitare il fenomeno del crosstalk, ovvero l'interferenza reciproca.
L'applicazione più comune del doppino e' il sistema telefonico: può estendersi per molti Km senza che il segnale si indebolisca, ovvero senza bisogno di amplificazione.
I doppini possono trasmettere segnali analogici e digitali.
L'ampiezza di banda dipende dal diametro del cavo e dalla distanza percorsa; per il loro basso costo e discreto livello di prestazioni(per tratti lunghi pochi km, possono raggiungere la velocità di alcuni Mbps), sono largamente utilizzati.
Esistono varie tipologie di UTP:
UTP-3: 16 MHz
UTP-5: 100 MHz
UTP-6: 250 MHz
UTP-7: 600 MHz
Il numero rappresenta il numero di annodamenti che hanno. Più questo valore è alto, maggiore sarà la velocità e minore l'interferenza dovuta ai campi magnetici che essi generano. Più spire comportano però un maggior costo, a causa del cavo che deve (ovviamente) essere presente in maggior quantità.

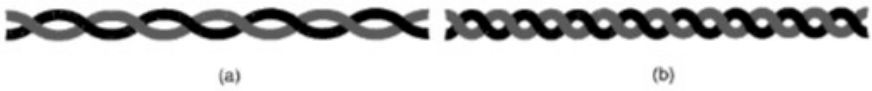


Figura 2.3. (a) UTP Categoria 3. (b) UTP Categoria 5.

Cavo coassiale:

Essendo più schermato del doppino, il cavo coassiale può estendersi per distanze più lunghe e consente velocità più elevate.
Un cavo coassiale è composto dal nucleo conduttore (anima di rame), coperto da un rivestimento isolante, a sua volta circondato da un conduttore cilindrico solitamente realizzato con una calza di conduttori sottili(calza conduttrice) che infine è avvolto da una guaina protettiva di plastica.
I cavi moderni hanno un'ampiezza di banda vicina a 1 Ghz.
Questa tipologia di cavo è molto utilizzata per le MAN e per le tv via cavo.

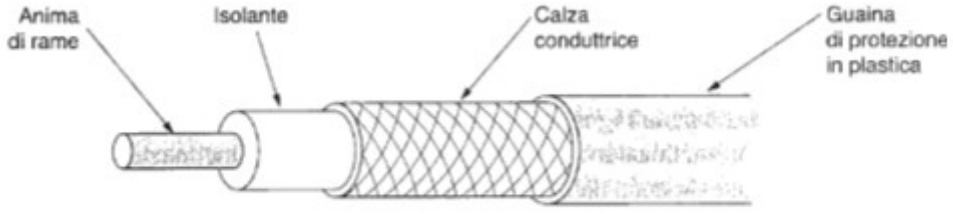


Figura 2.4. Un cavo coassiale.

Fibra ottica:

Un sistema di trasmissione ottico è formato da 3 componenti fondamentali:
- sorgente luminosa;
- mezzo di trasmissione;
- rilevatore di luce.
La sorgente di luce è rappresentata da led, oppure laser, anche se il secondo è meno diffuso e più costoso.
Il mezzo trasmissivo è la fibra, composta da un nucleo(core) di vetro di pochi micron, attraverso il quale si propaga la luce. Il nucleo è poi avvolto in un rivestimento di vetro(cladding), il quale ha un indice di rifrazione più basso: ciò costringe la luce a rimanere nel nucleo. Lo strato successivo è una sottile fodera di plastica, che protegge il rivestimento. La luce che attraversa la fibra è riflessa al suo interno, da un'estremità all'altra del cavo.
Le fibre, sono di solito raggruppate in fasci, protetti da una fodera più esterna.

Nonostante si trasmetta alla velocità della luce, la velocità viene limitata alla velocità di decodifica che avviene all'estremità.

Il principio che utilizza, è quello di far passare un impulso di luce in un pezzo di vetro. Quando un rilevatore viene colpito dalla luce, questo genera un impulso elettrico. Collegando quindi a un estremo della fibra una sorgente di luce e un rilevatore all'altro, si crea un sistema di trasmissione unidirezionale che accetta un segnale elettrico, lo converte e lo trasmette sotto forma di impulso luminoso; all'altra estremità della fibra converte nuovamente l'output in segnale elettrico.
La maggioranza della protezione, serve a proteggere il fragile e fine vetro all'interno.
E' inoltre possibile mettere più anime di vetro all'interno di una singola guaina.

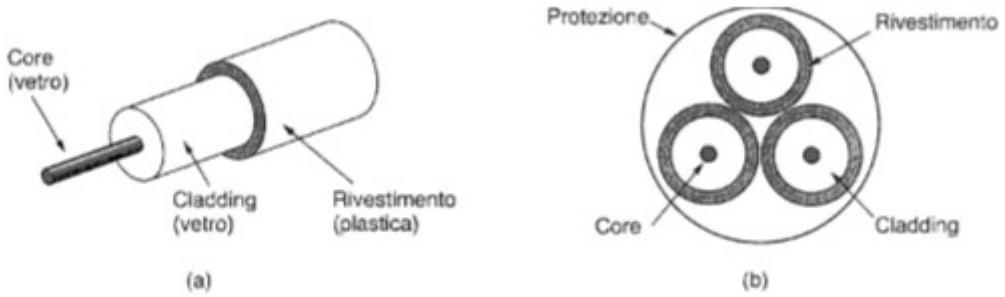


Figura 2.7. (a) Vista laterale di una singola fibra. (b) Vista frontale della guaina con tre fibre.

Problema: Come connettere le fibre tra loro?

1) Connettori: Si perde il 10-20% di luce.
2) Allineatori meccanici: Si perde il 10% di luce.
3) Fusione: Si fondono due pezzi di vetro, si perde circa l'1%

Problema: Quale luce usare?
Laser,led,lampadine ad incandescenza....Ogni luce ha proprietà diverse!

Elemento	LED	Laser a semiconduttore
Cadenza dei dati	Bassa	Alta
Tipo di fibra	Multimodale	Multimodale o monomodale
Distanza	Breve	Lunga
Durata	Lunga	Breve
Sensibilità alla temperatura	Scarsa	Elevata
Costo	Basso	Alto

Datarate:

Distanze:
Durata dell'apparecchio:
Alterazione del segnale dovuto alla temperatura:
Costo:

Figura 2.8. Confronto tra le sorgenti luminose realizzate con diodi a semiconduttore e LED.

Fibre ottiche o cavi in rame?

Vantaggi Fibra:

Primo fra tutti, la maggiore ampiezza di banda e tiene meglio il segnale. Non è influenzata dalle interferenze elettriche. E' sottile, piccola e leggera. Le componenti sono passive. L'unione di più reti in un'unica rete, o la divisione di una rete in più reti è più semplice.
La fibra è interessante anche dal punto di vista della sicurezza.

Svantaggi Fibra:

Costa di più, si piega molto meno facilmente e perciò si può danneggiare se si piega troppo. Inoltre, richiede ripetitori più complessi rispetto a quelli necessari per i cavi in rame.

4. **Caratteristiche e confronto fra i vari tipi di satellite, GEO,MEO,LEO.**

Un satellite è un grande ripetitore di microonde posizionato in cielo. Ci sono 3 tipi di satelliti, che si differenziano per la loro distanza dalla superficie terrestre.
-Satelliti geostazionari (GEO=Geostationary Earth Orbit)
-Satelliti medio-orbitali(MEO=Medium Earth Orbit satellites)
-Satelliti basso-orbitali(LEO=Low Earth Orbit satellites)

Dettagli:

Più basso è il satellite, più satelliti sono necessari per coprire l'intero pianeta.
-3 GEO
-10 MEO
-50 LEO

- I tempi di latenza(ritardo) diminuiscono man mano che ci si avvicina al suolo terrestre. I satelliti più bassi sono dunque avvantaggiati sotto questo punto di vista.
- La potenza richiesta per generare l'impulso (che rappresenta un'informazione), diminuisce mano a mano che diminuisce la distanza. (Satelliti bassi avvantaggiati).
- Il lancio costa molto di più per i satelliti più alti.

Analizziamo ora i satelliti in questo ordine:
-MEO
-GEO
-LEO

MEO:

Collocati tra le due fasce di Van Allen, i satelliti MEO sono stati i primi satelliti utilizzati. (Primo satellite Sputnik, 1957).
Questa tipologia di satelliti impiega circa 6 ore per compiere un giro intorno al pianeta.
Coprono un'area più piccola rispetto ai satelliti GEO, e possono essere raggiunti usando trasmettitori meno potenti.
Sono necessari 10 MEO per coprire l'intera superficie terrestre
Esempi di satelliti MEO: Gps.

GEO:

Sono collocati lungo l'equatore in quanto è l'unica orbita stabile per i satelliti, ed hanno un orbita circolare.
Questa tipologia di satelliti, viene posizionata a circa 35'000 km dalla superficie ed hanno un tempo medio di ritardo della trasmissione di circa 300 msec.
Con 1 satellite GEO, copro circa 1/3 della superficie terrestre.
Limite: Massimo 180 satelliti, in quanto per evitare interferenze tra essi, devono essere posti ad una distanza minima di 2 gradi nel piano equatoriale (360°/2=180).
Usi tipici: satelliti spia, satelliti meteo, satelliti DBS(=direct broadcast satellites, per la televisione via cavo)

LEO:

Scendendo di altezza, si incontrano i satelliti LEO. Poiché si spostano rapidamente, la realizzazione di un sistema completo richiede l'utilizzo di numerosi satelliti di questo tipo. D'altra parte, poiché i satelliti sono molto vicini alla superficie terrestre, le stazioni terrestri non hanno bisogno di molta energia e il ritardo nelle comunicazioni è di pochi msec.
Questa tipologia di satelliti, è la migliore per le telecomunicazioni.
Esempi di satelliti LEO: Iridium(comunicazione intra-satellitare), Globalstar (comunicazione a terra, per quanto possibile).

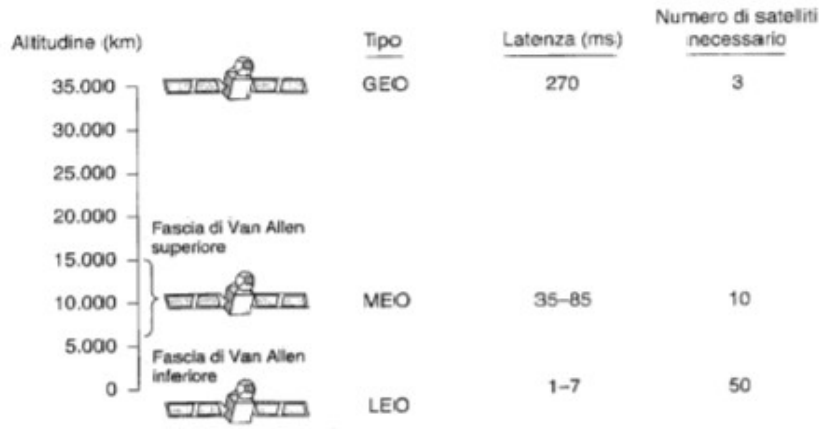


Figura 2.15. Satelliti di comunicazione e alcune delle loro proprietà, incluse le altitudini, il ritardo di andata e ritorno del segnale e il numero di satelliti necessari per fornire una copertura globale.

5. **Che cos'è la modulazione in frequenza(FM)? E in ampiezza(AM)?**

FM:
La modulazione in frequenza (FM, Frequency Modulation), è una delle tecniche di trasmissione utilizzate per trasmettere informazioni: si varia la frequenza dell'onda portante in maniera proporzionale all'ampiezza del segnale modulante(informazione che si intende trasmettere).

Rispetto alla modulazione di ampiezza, ha il vantaggio di essere molto meno sensibile ai disturbi e di permettere una trasmissione di miglior qualità.

AM:
La modulazione in ampiezza (AM, Amplitude Modulation), è una delle tecniche di trasmissione utilizzate per trasmettere informazioni: si varia l'ampiezza dell'onda portante in maniera proporzionale all'ampiezza del segnale modulante(informazione che si intende trasmettere).

Questa tipologia di modulazione è piuttosto semplice da realizzare ed è perciò stata utilizzata dagli albori delle trasmissioni radio. Nel caso della trasmissione binaria, ad una potenza bassa corrisponde lo zero, mentre ad una potenza alta corrisponde l'uno.
Tra i principali inconvenienti che si hanno, vi troviamo l'eccessiva sensibilità ai disturbi e all'attenuazione.

6. **Che cos'è la modulazione delta? (delta modulation)**

La modulazione delta è una variante della DPCM (differential pulse-code modulation).
Con questo metodo di compressione, si richiede che ogni valore campionato differisca dal precedente di +1 o -1; sotto queste condizioni, può essere trasmesso un singolo bit che dice se il nuovo campione è maggiore o minore del precedente.
Questa tipologia di codifica può creare problemi se il segnale cambia troppo rapidamente: in questo caso si perdono informazioni.
Perché delta? Perché invece di mandare informazioni su tutti i punti della curva, mando solo la differenza (delta) tra l'istante precedente e quello attuale.
Le comunicazioni "compresse", come ad esempio la modulazione delta, vengono gestite in multiplexing usando una classica TDM (Time Division Multiplexing).
La modulazione delta è adottata dai sistemi di telecomunicazioni militari(NATO e non) sin dagli anni settanta.

7. **Descrivere in dettaglio il GSM (Global System for Mobile communication)**

Il GSM (Global System for Mobile communication), è una tecnologia simile al D-AMPS, appartenente alla seconda generazione di cellulari, con qualche modifica.
Proprio come il D-AMPS, il GSM utilizza il multiplexing a divisione di frequenza (FDM), con anche divisione a tempo (TDM).
I canali del GSM, sono però molto più ampli rispetto a quelli utilizzati in AMPS (200 kHz rispetto a 30 kHz); da ciò, ne deriva anche una maggior capienza di utenti (8 vs 3) ed hanno un data-rate per utente più alto.
Un sistema GSM ha 124 coppie di canali simplex; ognuno è ampio 200 kHz e supporta 8 connessioni separate mediante multiplexing a divisione di tempo (TDM), 4 per ogni direzione. Trasmissione e ricezione non avvengono nello stesso intervallo, poiché il sistema non è in grado di gestirlo.
Come nel caso di AMPS, il tempo di elaborazione dati consuma gran parte dell'ampiezza di banda e lascia in definitiva soltanto 24,7 kbps per utente a disposizione del carico utile, al lordo della correzione degli errori. Dopo la correzione degli errori, lo spazio a disposizione della voce è di soli 13 kbps. In ogni caso, la qualità della voce è molto migliore (D-AMPS è di 4/8 kbps), così come la trasmissione dei dati è più decente.

Il sistema GSM ha inoltre portato all'introduzione delle SIM (Subscriber Identity Module), che consentono la divisione del numero telefonico dal cellulare. Queste, contengono varie informazioni, tra le quali spiccano per importanza la IMSI(Internetional Mobile Subscriber Identity, ovvero l'identificativo della sim) e la KI(chiave di identificazione).
Come avviene il collegamento con l'operatore:

- **Il cellulare manda l'IMSI della SIM all'operatore.**
- **L'operatore genera un numero casuale e lo manda al cellulare.**
- **Il cellulare firma il numero con la Ki e lo manda all'operatore.**
- **L'operatore ha nel suo database l'IMSI e la Ki associata: firma anche lui il numero casuale con la Ki, e controlla che il numero sia lo stesso di quello inviatogli dal cellulare. Identificazione riuscita!**

Struttura della cella GSM:
Nel protocollo GSM, ci sono 4 tipi di celle: Macro, Micro, Pico, Umbrella. Le prime sono le più grandi, sopraelevate rispetto agli edifici ed hanno un raggio massimo di 35 km. Le micro, sono più piccole, coprono un'altezza pari agli edifici. Le pico sono molto piccole, usate in aree molto dense, tipicamente indoor. Umbrela infine è una piccola estensione, usata per coprire i buchi tra le varie celle delle altre tre tipologie.

Il problema del GSM:
Essenzialmente, GSM è stato costruito per trasmettere voce.
Questo significa che se si vuole usare il GSM per trasmettere dati, ci sono gravi problemi:

- **il problema principale è dato dal fatto che viene riservato un canale intero alla comunicazione.**
- **→ navigare in internet spreca un intero canale voce anche quando il traffico è poco.**
- **Inoltre, la tariffa corrispondente è a tempo e non a traffico.**

8. **Si descriva la tecnica CDMA (Code Division Multiple Access), possibilmente con un esempio.**

CDMA funziona in modo completamente differente rispetto a D-AMPS e GSM, che utilizzano FDM e TDM.
CDMA(Code Division Multiple Access) invece di dividere l'intervallo di frequenze assegnate in poche centinaia di canali a banda stretta, permette a ogni stazione di trasmettere per tutto il tempo attraverso l'intero spettro di frequenza.
Questo sistema, rende meno rigida la premessa secondo la quale i frame che entrano in collisione, di dispositivi che "parlano" in contemporanea, si alterano completamente.
CDMA presume infatti che segnali sovrapposti si sommino linearmente.
Facendo un esempio, immaginiamoci di trovarci ad un party:
-TDM: ognuno parla a turno, mentre tutti gli altri stanno zitti.
-FDM: ognuno parla su "toni" diversi.
-CDMA: tutti parlano contemporaneamente, dove però ogni coppia comunica in un linguaggio diverso.
La chiave dello schema CDMA è pertanto la capacità di estrarre il segnale desiderato scartando tutto il resto.
Per risalire al messaggio originale, ovvero ciò che viene "detto", basta togliere il rumore aggiunto dalle altre conversazioni.
Questo sistema, lavora su uno spazio multidimensionale. Vengono stabiliti degli assi adeguati, e poi si usano regole di composizione (somma) e proiezione (prodotto scalare) per fare encoding/decoding.
Per questo sistema, vengono utilizzate le matrici di Walsh, che sono essenzialmente derivate dalle matrici di Hadamard.

Il CDMA è stato introdotto di recente, in quanto il sistema assume che ognuno dei dispositivi (cellulari 2G, in questo caso) parli il suo linguaggio con lo stesso volume. Se qualcuno parla più forte, si sballa tutto. La soluzione adottata è quella di far "parlare" il dispositivo più o meno forte a seconda di quanto è lontano dalla base, in modo tale da far sentire alla base tutti i cellulari allo stesso modo ("volume"). Il dispositivo/cellulare, in base alla potenza del segnale che riceve dalla base (la quale trasmette sempre a potenza fissa), determina quanto è lontano dalla base e di conseguenza stabilisce il "volume".

Vediamo un esempio di CDMA:
H4 (Hadamard)
Ci sono 4 canali

1 1 | 1 1
1 -1 | 1 -1
---
1 1 | -1 -1
1 -1 | -1 1

| $H2^k = H2^{k-1}\ H2^{k-1}$
| $H2^{k-1}\ -H2^{k-1}$
|
//Sostanzialmente, copio
|nelle prime 3 caselle
|l'H precedente, e nell'ultima
|metto tutti quanti con il segno
|al contrario.

4 Cellulari:
Italiano: Spaghetti, vino
Americano: Hamburger, coca-cola
Tedesco: salsicce, birra
Giapponese: sushi, sakè

spaghetti= prima riga di H4; 1 1 1 1
vino= prima riga di H4 (Riflessa) -1 -1 -1 -1
hamburger=seconda riga di H4 1 -1 1 -1
coca-cola=seconda riga di H4 (Riflessa) 1 -1 1 -1
ecc...

Ognuno trasmette la sua forma d'onda nello stesso istante; le forme d'onda si sommano:
Italiano: spaghetti 1 1 1 1
Americano: coca-cola -1 1 -1 1
Tedesco: sta zitto
Giapponese: sushi 1 -1 -1 1

1 1 -1 3

Quando qualcuno vuole ascoltare:
-Deve controllare se c'è una parola nella sua lingua:
-L'italiano:
[1 1 1 1] * [1 1 -1 3] = +1 +1 -1 +3 = 4 diverso da 0 => c'è una parola in italiano.
Essendo il valore positivo, sta dicendo spaghetti.
-L'americano:
[1 -1 1 -1] * [1 1 -1 3] = +1-1-1-3 =-4 diverso da 0 => c'è una parola in americano
Essendo il valore negativo, sta dicendo coca-cola.
-Il tedesco:
[1 1 -1 -1] * [1 1 -1 3] = +1 +1 +1 -3 =0 => non c'è alcuna parola in tedesco
-Il giapponese:
[1 -1 -1 1] * [1 1 -1 3]= +1 -1 +1 +3 = 4 => c'è una parola in giapponese
Essendo il valore positivo, sta dicendo sushi.

9. **Il GPRS: cos'è, pregi e difetti.**

Il sistema GPRS(General Packet Radio Service) è classificato 2.5G, ovvero si colloca tra il sistema 2G e il sistema 3G.
Consiste in una rete a pacchetti costruita sopra D-AMPS e GSM.
GPRS permette alle stazioni mobili di inviare e ricevere pacchetti IP in una cella basata su un sistema vocale. Si tratta quindi di un servizio aggiuntivo, che permette la gestione del traffico a pacchetti, per il sistema GSM.

Il sistema GPRS risolve i problemi riscontrati in GSM.
Essenzialmente, GSM è stato costruito per trasmettere voce.
Questo significa che se si vuole usare il GSM per trasmettere dati, ci sono gravi problemi:

- **il problema principale è dato dal fatto che viene riservato un canale intero alla comunicazione.**
- **→ navigare in internet spreca un intero canale voce anche quando il traffico è poco.**
- **Inoltre, la tariffa corrispondente è a tempo e non a traffico.**

GPRS permette quindi la navigazione a pacchetti, e non a messaggi interi.
I vantaggi che questo approccio comporta, sono:

- **non si spreca banda**
- **non serve un canale dedicato**
- **si possono usare tariffe a traffico**

GPRS supporta inoltre i classici protocolli IP e PPP ed è in grado di allocare dinamicamente i canali "internet" e quelli voce, a seconda delle richieste di traffico.
GPRS permette di inviare e ricevere pacchetti IP in una cella basata su sistema vocale.
Quando attivo, alcuni slot temporali vengono dedicati al traffico dei pacchetti. Questi slot sono divisi in canali logici. Ogni canale è utilizzato per scaricare i pacchetti nei quali c'è indicato il destinatario. Per inviare un pacchetto, la stazione mobile richiede uno o più slot temporali ed effettua la richiesta alla base. La base invia poi il pacchetto via internet tramite rete via cavo.

10. Handoff: cos'è e i vari tipi.

Nell'ambito della telefonia mobile, con il termine handoff (o handover), si intende la procedura per la quale viene cambiato il canale usato dalla connessione di un cellulare alla rete, mantenendo attiva la connessione.
Nelle connessioni mobili infatti, ogni telefono è connesso alla rete ad una sola cella finché non si sposta, ed il segnale diventa troppo debole. Quando ciò avviene, è necessario cambiare la cella precedente con una più vicina:
lo switching center chiede alle celle vicine quanta potenza ricevono dal cellulare, e la cella che ottiene la maggior potenza, ottiene l'assegnamento del cellulare.

La disconnessione da una cella può avvenire secondo le seguenti modalità:

-Hard Handoff:
La vecchia stazione "molla" il cellulare, che viene poi agganciato dalla nuova stazione. Ciò provoca del lag, e in alcuni casi la linea può anche cadere. Il tempo richiesto è di cerca 300 msec (0.3 secondi), eccessivi per una chiamata in corso.
Inoltre, se la nuova cella non è in grado di prendere il controllo del dispositivo, la chiamata viene interrotta bruscamente.

-Soft Handoff
Nel soft handoff, la nuova cella acquisisce il cellulare prima che la vecchia lo rilasci. In un certo momento, il cellulare deve essere connesso a due frequenze (celle) contemporaneamente. I costi per l'implementazione sono ovviamente maggiori, e la batteria dura meno.
Il problema di questa modalità, sta nel fatto che il cellulare deve essere in grado di collegarsi a due frequenze(celle) contemporaneamente. 1G e 2G non gestiscono questo tipo di handoff, ed utilizzano invece l'hard handoff.

In AMPS (1G), la gestione dell'handoff è di competenza dello switching center; in D-AMPS l'onere passa invece ai singoli cellulari.

11. FDM, TDM, CDM: algoritmi per la selezione della banda.

FDM: E' l'acronimo di Frequency Division Multiplexing.

Per la condivisione di un canale, divide lo spettro in varie bande di frequenza, e ad ogni utente viene assegnata una certa banda, avendone uso esclusivo per inviare il proprio segnale.
Un semplice esempio per capirne il funzionamento, è quello di considerare un'autostrada la cui carreggiata viene divisa in più corsie per far transitare più auto in contemporanea.

TDM: E' l'acronimo di Time Division Multiplexing.

L'intera banda viene assegnata agli utenti a turno, secondo una politica round-robin; ognuno di loro, periodicamente, prende possesso dell'intera banda per un periodo di tempo limitato(slot temporale).
Nell'esempio dell'autostrada, l'intera carreggiata verrebbe assegnata per un determinato slot temporale ad una sola auto.

CDM: E' l'acronimo di Code Division Multiplexing.

La comunicazione è a spettro distribuito, in cui un segnale a banda stretta viene sparso su una banda di frequenza più ampia. Ciò rende il segnale più tollerante alle interferenze e permette a più segnali di utenti diversi di condividere la stessa banda di frequenza. Viene anche chiamato CDMA.
La chiave dello schema CDMA è pertanto la capacità di estrarre il segnale desiderato scartando tutto il resto.
Per risalire al messaggio originale, ovvero ciò che viene "detto", basta togliere il rumore aggiunto dalle altre conversazioni.

12. QAM e QAM16.

Il segnale digitale può essere trasmesso in 3 modi di base:
-Modulando in ampiezza (ASK=amplitude shift keying)
-Modulando in frequenza (FSK=frequency shift keying)
-Modulando in fase (PSK=phase shift keying)

La modulazione di fase:

Possiamo usare vari "sfasamenti" in ogni singolo impulso in modo da poter avere un alfabeto di simboli più capiente, e quindi a parità di baud, aumentare i bitrate.

Il QPSK (Quadrature Phase Shift Keying):

Si basa sull'utilizzo di 4 sfasamenti, avendo dunque un alfabeto di 4 simboli.
Il bitrate è doppio rispetto al baudrate, in quanto per rappresentare 4 simboli sono necessari 2 bit, e l'invio di 1 boud corrisponde quindi all'invio di 2 bit.

Se cambiassimo solo fase per incrementare il nostro alfabeto, un maggior numero di simboli dell'alfabeto comporterebbe sempre più piccole differenze di fase tra un simbolo e l'altro, e sarebbero quindi sempre meno distinguibili gli uni dagli altri. (Pensiamo ad esempio ad 1 miliardo di simboli..!)
Quindi, per aumentare la banda, posso incrementare sì il numero dei simboli, ma fino ad un certo limite.

Un approccio migliore è quello di combinare più tipi di modulazione assieme.

QAM(=Quadrature Amplitude Modulation) si basa basa proprio su questo principio, combinando la modulazione in ampiezza con la modulazione in fase.

QAM-16:

Utilizzando 4 ampiezze e 4 fasi, riesce a rappresentare 16 diversi simboli, permettendo così di trasmettere 4 bit per simbolo.
QAM-16 ha quindi bitrate quadruplo rispetto al baudrate.

QAM-64:

Utilizza sempre la modulazione in fase e in ampiezza, e permette di rappresentare 64 simboli, permettendo così di trasmettere 6 bit per simbolo.
Il bitrate è quindi sestuplo rispetto al baudrate.

Per rappresentare QPSK, e QAM, vengono utilizzati i diagrammi a costellazione, che mostrano le combinazioni valide di ampiezza e fase.
Ogni modem ad alta velocità ha uno suo schema di costellazione.
Con molti punti nello schema di costellazione, anche un piccolo livello di rumore nell'ampiezza o nella fase rilevata può provocare un errore, e potenzialmente, può far perdere molti bit.

I QAM ottimali sarebbero quelli circolari (es QAM-8); in genere però, trovare i QAM ottimali è tutt'altro che banale. Nella pratica vengono utilizzati i QAM rettangolari invece di quelli circolari e/o ottimali, in quanto sono più facili da generare e da codificare.

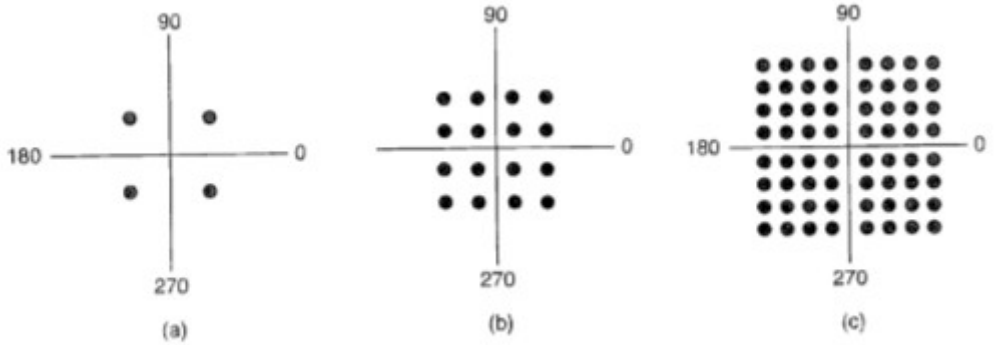


Figura 2.25. (a) QPSK. (b) QAM-16. (c) QAM-64.

FINE CAPITOLO 2.
CAPITOLO 3:

13. Che cos'è il byte stuffing.

Il byte stuffing è uno dei metodi di framing, utilizzato per suddividere il flusso di bit in frame. Questo metodo, utilizza un byte speciale all'inizio e alla fine di ogni frame, definito "flag byte", ed ha appunto lo scopo di delimitare sia l'inizio sia la fine dei frame.
Quando il destinatario perde la sincronizzazione, può semplicemente cercare il flag byte per trovare la fine del frame corrente. Due flag byte consecutivi indicano la fine di un frame e l'inizio del successivo.
Un problema può presentarsi qualora all'interno del flusso di dati, fosse presente un flag byte, che compare naturalmente dentro ai dati, interferendo così con le operazioni di framing.
Una possibile soluzione è quella di far inserire alla sorgente un byte di escape (ESC), subito prima di ogni occorrenza accidentale del flag byte all'interno dei dati. Lo strato data link della destinazione provvederà a rimuovere il byte di escape prima di passare i dati allo strato network. Se un byte di escape si trova dentro ai dati, questo deve essere preceduto da un byte di escape.
Questa tecnica, è chiamata byte stuffing o anche character stuffing.
Questo metodo di framing ha lo svantaggio di utilizzare caratteri a 8 bit, e non tutte le codifiche dei caratteri usano 8 bit (UNICODE ne usa 16, ad esempio).
La tecnica del bit stuffing, consente invece di gestire codifiche di carattere con un numero arbitrario di bit, risolvendo così il problema riscontrato in byte stuffing.

14. Che cos'è il bit stuffing.

Il bit stuffing è uno dei metodi di framing, utilizzato per suddividere il flusso di bit in frame. Per risolvere il problema riscontrato dal byte stuffing, si utilizza una tecnica analoga che permette però di creare data frame che contengono sia un numero arbitrario di bit, sia codifiche di carattere con un numero arbitrario di bit. Il funzionamento è il seguente:

Ogni frame inizia e finisce con un gruppo speciale di bit, 01111110 (sei '1'), che ha la funzione di flag byte. Ogni volta che lo strato data link della sorgente incontra cinque 1 consecutivi nei dati, inserisce automaticamente un bit con valore 0 nel flusso in uscita.
Questa operazione è chiamata bit stuffing.
La destinazione, quando riceve cinque bit consecutivi con valore 1 seguiti da uno 0, automaticamente elimina lo 0.
Nel caso in cui i dati da trasferire contengono la sequenza che corrisponde al flag, ovvero 01111110, tale sequenza verrà trasmessa come 011111010 e, dopo l'operazione di destuffing, il valore ritornerà uguale a quello originario.
Con questa modalità, il confine tra due frame viene riconosciuto in modo inequivocabile tramite l'uso della sequenza flag.

15. Numero di bit necessari per il riconoscimento(correzione) degli errori di trasmissione.

Le proprietà di rilevazione e correzione degli errori di una codifica dipendono dalla sua distanza di Hamming, che rappresenta il numero di bit diversi nei messaggi.
Per trovare(rilevare) m errori, è necessaria una codifica con distanza m+1: stiamo quindi sprecando 1/(m+1) per il controllo dell'errore. Il data-rate effettivo sarà dunque m/(m+1). Ad esempio, con m=3, sprechiamo un quarto del data-rate.
Un semplice esempio di codifica a rilevazione d'errore si può realizzare aggiungendo un bit di parità ai dati.

Se invece vogliamo progettare una codifica in grado non solo di rilevare, ma anche di correggere m errori, avremo bisogno di una codifica con distanza 2m+1.

Esempi di metodi per la rilevazione degli errori:

- **bit di parità**
- **codice di Luhn**
- **Hamming**
- **CRC**

16. Si descriva cos'è il CRC. Si calcoli inoltre il CRC di 10011101 usando il polinomio generatore di x^4+x+1.

CRC è un codice che fa solamente error detection.
La codifica polinomiale, nota anche come CRC (Cycle Redundancy Check = controllo di ridondanza ciclico), è basata sull'aritmetica polinomiale in base 2, nella quale le sequenze di bit vengono trattate come dei polinomi a coefficienti che possono assumere solo i valori 0 oppure 1. In questo modo, un numero binario viene visto come un polinomio nella quale ogni bit viene considerato come il coefficiente di potenze sempre crescenti. Es. 1011 => x^3+x+1
Quando si utilizza una codifica polinomiale, la sorgente e la destinazione devono mettersi d'accordo in anticipo su un polinomio generatore, G(x). E' necessario che il frame sia più lungo del polinomio generatore. L'idea è quella di aggiungere un checksum alla fine del frame, in modo che il polinomio rappresentato dal frame con checksum sia divisibile per G(x).
Quando la destinazione riceve il frame con checksum, prova a dividerlo per G(x). Se c'è un resto, vuol dire che c'è stato un errore di trasmissione.
L'algoritmo per calcolare il checksum è il seguente:

1. **posto r il grado di G(x), aggiungere r bit con valore zero dopo la parte di ordine più basso del frame, così che adesso contenga m+r bit e corrisponda al polinomio $x^r M(x)$.**
2. **dividere la sequenza di bit corrispondente a G(x) per la sequenza corrispondete a $x^r M(x)$, usando la divisione modulo 2.**
3. **sottrarre il resto (che contiene sempre al massimo r bit) dalla sequenza corrispondente a $x^r M(x)$ usando la sottrazione in modulo 2. Il risultato è il frame con checksum pronto per la trasmissione. [T(x)]**

17. Descrivere il protocollo Stop and Wait, pregi e difetti.

[Strato Data Link / protocollo]
Il protocollo stop & wait cerca di gestire il problema del flow control (controllo di flusso), nella quale in un canale semplice (simplex), se inviamo troppi dati, rischiamo che il ricevente non riesca a gestirli, e che questi vadano quindi persi.
Questo protocollo funziona sotto l'ipotesi in cui il canale di comunicazione sia senza errori e che il traffico sia di tipo simplex. Un canale half-duplex può bastare per questo protocollo.

La soluzione adottata dal protocollo Stop & Wait, è quella di disegnare la rete in modo tale da far parlare il ricevente con chi manda i dati: l'idea, è quella di far mandare al mittente un blocco di dati (un frame), e poi di farlo aspettare (stop & wait) fino a che il ricevente non gli manda un messaggio di conferma ACK (acknowledge), dando il via libera al mittente per l'invio di un altro frame.

Lo svantaggio principale è l'attesa, ma in compenso non c'è bisogno di regolare la velocità.

L'utilizzo di questo protocollo, può portare però ad avere due diversi errori:

- **-Il frame non arriva mai a destinazione e il mittente aspetta all'infinito: c'è bisogno di un tempo limite di rinvio. L'introduzione di un timer, risolve questo problema: se entro un certo periodo di tempo non riceviamo il messaggio di conferma (l'ACK), rimandiamo il pacchetto.**
- **-L'altro problema riguarda l'ACK: potrebbe non arrivare al mittente, il quale rinvia il pacchetto e al destinatario arriva più volte; grazie al numero di messaggio, questo però può essere tranquillamente scartato. E' sufficiente distinguere solo pacchetti che sono consecutivi, ed è quindi sufficiente 1 bit (due simboli, 0 e 1).**

18. Cos'è il piggybacking

[Strato Data Link / tecnica]
Nella maggior parte delle situazioni reali, risulta necessario poter trasmettere in entrambe le direzioni (full-duplex). Un modo per ottenere una trasmissione di dati full-duplex è quello di inframezzare dati e frame di controllo sullo stesso canale.
Per quanto riguarda l'acknowledgement, guardando al campo kind dell'intestazione del frame in arrivo, la destinazione può determinare se si tratta di dati oppure di un ACK.
Una miglioria, consiste nel far aspettare la destinazione che ha ricevuto un frame di dati, fino a che lo strato network gli passa il successivo pacchetto. L'ACK viene aggiunto al frame di dati in uscita usando il campo ACK nell'intestazione del frame.
In questo modo ACK si procura un passaggio gratis insieme al successivo frame dati trasmesso.

Questa tecnica, che consiste quindi nel ritardare gli ackowledgement in uscita in modo da agganciarli al successivo frame di dati, è chiamata piggybacking (=trasportare in groppa).
Il vantaggio principale di questa tecnica rispetto agli ACK separati, sta nel miglio uso della banda disponibile.
Per sfruttare al meglio questa tecnica, è necessario prestare attenzione a non aspettare troppo per la trasmissione del ACK: se lo strato data link aspetta più a lungo del periodo di timeout della sorgente, il pacchetto verrà infatti ritrasmesso, vanificando l'ACK.
Quindi, se il nuovo pacchetto arriva rapidamente viene fatto il piggybacking dell'ACK altrimenti, se alla fine del periodo di attesa il pacchetto non è arrivato, lo strato data link invia un frame di ACK separatamente.

19. Si descriva la tecnica del Sliding Window

I protocolli che utilizzano la tecnica “Sliding Window”, vengono utilizzati per il controllo di flusso dei dati, sono full-duplex e sfruttano il piggybacking.
Nei protocolli Sliding window (“finestra scorrevole”), ogni frame in uscita contiene un numero di sequenza che va da 0 fino a un valore massimo. Il massimo è di solito 2^n-1, cosicché il numero di sequenza rientra in n bit. Stop-and-wait sliding window a 1 bit, restringe i numero di sequenza a 0 e 1. Versioni più sofisticate utilizzano un numero arbitrario di n.

L'essenza dei protocolli sliding window, è che ad ogni istante, la sorgente tiene traccia di un insieme di numeri di sequenza corrispondenti ai frame che è autorizzata a inviare. Questi frame, si trovano nella cosiddetta “finestra di invio”.
Allo stesso modo, la destinazione tiene traccia della “finestra di ricezione”, ovvero l'insieme dei frame che può accettare.
Le finestre di mittente e destinatario non devono necessariamente avere la stessa dimensione, né uguali limiti inferiori o superiori.

Sorgente:
Il numero di sequenza nella finestra d'invio rappresenta i frame che sono già stati trasmessi correttamente, oppure che sono in transito, ma che non hanno ancora ricevuto l'ACK.
Quando un nuovo pacchetto arriva dallo strato network, gli viene assegnato il numero di sequenza successivo in ordine crescente e il limite superiore della finestra è incrementato di uno. Quando invece arriva un ACK, s'incrementa di uno il limite inferiore della finestra.
La finestra, continua così a mantenere la lista dei frame che ancora necessitano di ACK.
I frame dentro la finestra devono essere mantenuti in memoria (buffer) per la possibile ritrasmissione. Se il buffer è pieno, il livello data link deve costringere il livello network a sospendere la consegna degli pacchetti.

Destinazione:
Analogamente, il destinatario mantiene una finestra corrispondente agli indici dei frame che possono essere accettati dalla destinazione.

- **Se arriva un frame il cui indice è fuori dalla finestra, questo viene scartato.**
- **Se arriva un frame il cui indice è dentro la finestra:**
  - **Il frame viene accettato.**
  - **Viene spedito l'ACK.**
  - **La finestra viene spostata in avanti.**
- **La finestra del destinatario rimane sempre della stessa dimensione, e se essa è pari a 1 il livello accetta i frame solo nell'ordine corretto.**

-Fanno parte di questo tipo di protocolli:

- **Sliding window a 1 bit**
- **Go-back-n**
- **Selective repeat**

20. Si descriva l'idea dei protocolli “Go Back N”, indicandone pregi e difetti.

Se il tempo di andata e ritorno del segnale è alto (ritardo) , come ad esempio nel caso dei canali satellitari, c'è un enorme inefficienza con i protocolli stop-and-wait, perché si sta quasi sempre fermi ad aspettare l'ACK.
Per migliorare le cose, si può consentire l'invio di un certo numero di frame anche senza aver ricevuto l'ACK del primo. Questa tecnica va sotto il nome di pipelining.
Ciò pone però un serio problema, perché se un frame nel mezzo della sequenza si rovina molti altri frame vengono spediti prima che il mittente sappia che qualcosa è andato storto.
Ricordiamo che lo strato data link della destinazione è obbligato a mandare i pacchetti allo strato network nella corretta sequenza.
Un approccio possibile al problema, è quello del protocollo Go-Back-N.

-Se arriva un frame danneggiato o con un numero di sequenza non progressivo, il destinatario ignora tale frame e tutti i successivi, non inviando i relativi ACK.
Ciò corrisponde ad una finestra di dimensione 1 nel ricevitore, che quindi accetta i frame solo nell'ordine giusto.
-Il mittente ad un certo punto va in time-out sul frame sbagliato, e poi su tutti quelli successivi (scartati dal destinatario!), e quindi provvede a ritrasmettere la sequenza di frame che inizia con quello per il quale si è verificato il time-out.

Osservazione: Il mittente deve mantenere in un apposito buffer tutti i frame non confermati (ACK), per poterli eventualmente ritrasmettere. Se il buffer si riempie, il mittente deve bloccare il livello network fino a che non si ricrea spazio nel buffer. Inoltre, vi è spreco di banda se il tasso d'errore è alto e/o il time-out è lungo.

Osservazioni:
-E' necessaria la gestione di timer multipli (uno per ogni frame inviato e non confermato).
-Il ricevente, per inviare gli ACK, usa il piggybacking se possibile, altrimenti invia un apposito frame.

21. Si descriva cos'è la tecnica del Selective Repeat.

Se il tempo di andata e ritorno del segnale è alto, come ad esempio nel caso dei canali satellitari, c'è un enorme inefficienza con i protocolli stop-and-wait, perché si sta quasi sempre fermi ad aspettare l'ACK.
Per migliorare le cose, si può consentire l'invio di un certo numero di frame anche senza aver ricevuto l'ACK del primo. Questa tecnica va sotto il nome di pipelining.
Ciò pone però un serio problema, perché se un frame nel mezzo della sequenza si rovina molti altri frame vengono spediti prima che il mittente sappia che qualcosa è andato storto.
Ricordiamo che lo strato data link della destinazione è obbligato a mandare i pacchetti allo strato network nella corretta sequenza.
Oltre all'utilizzo del protocollo Go-Back-N, un secondo approccio più efficiente è chiamato Selective Repeat.

- **Il destinatario mantiene nel suo buffer tutti i frame ricevuti successivamente ad un eventuale frame rovinato: non appena questo arriva nuovamente (senza errori), esso e tutti i successivi frame contigui che il destinatario ha mantenuto nel buffer, vengono consegnati al livello network.**
- **Per ogni frame arrivato bene, il destinatario invia un ACK col numero più alto della sequenza completa arrivata fino a quel momento.**
- **Quando si verifica un timeout, il mittente rispedisce il frame corrispondente.**

Alcune considerazioni del protocollo Selective-Repeat:

- **Mittente e destinatario devono entrambi gestire un buffer per mantenervi i frame:**
  - **Non confermati (mittente);**
  - **Successivi ad un errore (destinatario).**
- **Vi è un basso spreco di banda, dato che si può ulteriormente diminuire mandando un NACK (Negative ACKnowledgement) quando:**
  - **Arriva un frame danneggiato;**
  - **Arriva un frame diverso da quello atteso.**

Osservazioni:
-E' necessaria la gestione di timer multipli (uno per ogni frame inviato e non confermato).
-Il ricevente, per inviare gli ACK, usa il piggybacking se possibile, altrimenti invia un apposito frame.

FINE CAPITOLO 3.
CAPITOLO 4:

22. Descrivere la differenza fra ALOHA e ALOHA-SLOTTED.

[Sottostrato MAC]
Tra i vari algoritmi esistenti per l'assegnazione di un canale ad accesso multiplo, vi troviamo il sistema denominato "ALOHA".
Esistono due versioni di ALOHA: quella pura (Pure ALOHA), e quella slotted (Slotted ALOHA).
ALOHA puro non richiede una sincronizzazione temporale globale, a differenza invece dello slotted ALOHA; in quest'ultima versione, il tempo è inoltre diviso in intervalli discreti dove tutti i frame devono inserirsi.

ALOHA puro:
In questa versione, le stazioni trasmettono quando vogliono, però durante la trasmissione ascoltano il canale e confrontano ciò che ricevono con ciò che hanno spedito, in modo tale da verificare se il frame è andato distrutto.
Se si verifica una collisione quindi, le stazioni se ne accorgono e, in tal caso, dopo aver lasciato passare una quantità di tempo casuale, ritrasmettono il frame.
Basta che il primo bit di un nuovo frame si sovrapponga all'ultimo bit di un frame quasi completato, per considerarli entrambi totalmente distrutti, e quindi da ritrasmettere.
La scelta di attendere per una quantità di tempo casuale deriva dal fatto che altrimenti una collisione ne ricrea infinite altre.
Funzionamento a basso livello:
Definiamo come frame time il tempo necessario alla trasmissione di un frame, che ha lunghezza fissa. Supponiamo che vengano complessivamente generati dei frame con una distribuzione di Poisson, con media N frame per frame time.

- **Se N>1, le stazioni stanno generando frame a una frequenza più elevata della frequenza che il canale è in grado di gestire, e quasi ogni frame subirà una collisione.**
- **Se 0<N<1, viene garantita una ragionevole capacità di trasporto.**

Supponiamo inoltre che anche la distribuzione di tutti i frame (vecchi e nuovi) sia di Poisson, con valore medio pari a G frame per frame time.
Sotto qualunque condizione di carico, il throughput (cioè la quantità di pacchetti che arrivano a destinazione) è uguale al carico offerto, moltiplicato per la probabilità che la trasmissione abbia successo, ossia:

$$\text{Throughput} = G * P(0)$$

con: P(0)=probabilità che la trasmissione abbia successo.

La probabilità che k frame siano generati durante un dato frame time è data dalla distribuzione di Poisson $Pr[k]=(G^k e^{-G}) / k!$

Perciò la probabilità di zero frame, è e^{-G}**.**

Nel "periodo di vulnerabilità" di un frame, cioè nell'intervallo di tempo nel quale esso è a rischio di collisioni, mediamente vengono generati 2G frame. Di conseguenza, la probabilità che non si generino nuovi frame per tutto il periodo di vulnerabilità di un frame è:

$$P(0)=e^{-2G}$$

Il throughput raggiungibile dal protocollo ALOHA Puro, è quindi

$$\textbf{Throughput} = Ge^{-2G}$$

In altre parole, con ALOHA puro, al massimo si può sperare di utilizzare il 18% del canale.
ALOHA SLOTTED:
Un metodo per aumentare l'efficienza di ALOHA (Puro), consiste nel dividere l tempo in intervalli discreti, ciascuno corrispondente ad un frame time.
Le stazioni non possono iniziare a trasmettere quando vogliono, ma solamente all'inizio dell'intervallo. Questo protocollo, che prende il nome di Slotted Aloha, dimezza il periodo di vulnerabilità che diventa quindi uguale ad un solo frame time.
Il throughput ottenibile col protocollo Slotted ALOHA, è:

$$\textbf{Throughput} = Ge^{-G}$$

In altre parole, con Slotted ALOHA, al massimo si può sperare di utilizzare il 37% del canale.

23. Si illustri CSMA, indicandone pregi e difetti.

[Sottostrato MAC]
Nelle reti locali, le stazioni possono ascoltare il canale e regolarsi di conseguenza, ottenendo un'efficienza molto più alta rispetto ai protocolli Aloha e Aloha-slotted nei quali invece le stazioni trasmettono senza preoccuparsi di verificare se il canale è libero.
I protocolli nei quali le stazioni ascoltano il canale prima di iniziare a trasmettere, si dicono carrier sense.

Ci sono vari protocolli carrier sense: CSMA (Carrier Sense Multiple Access):

CSMA 1-Persistente:
- **Quando una stazione deve trasmettere, ascolta il canale:**
  - **se è occupato, aspetta finché si libera e quindi trasmette;**
  - **se è libero, trasmette (con probabilità 1, da cui il nome)**
- **Se avviene una collisione, la stazione aspetta un tempo random e riprova a trasmettere.**
- **Problemi:**
  - **Una stazione A trasmette, e prima che il suo segnale arrivi a B, anche B inizia a trasmettere; dunque, si verifica una collisione. Più alto è il tempo di propagazione fra A e B e più grave è il fenomeno.**
  - **A e B ascoltano contemporaneamente durante la trasmissione di C, e non appena quest'ultima termina iniziano entrambe a trasmettere: anche in questo caso si verifica una collisione.**

CSMA Non-persistente:
- **Quando una stazione deve trasmettere, ascolta il canale:**
  - **se è occupato, invece di trasmettere non appena si libera come in 1-Persistente, la stazione aspetta comunque un tempo random e ripete tutto il procedimento da capo;**
  - **se è libero, si comporta come in 1-Persistente (inizia ad inviare dati con p=1).**
- **Intuitivamente, ci si aspettano maggiori ritardi prima di riuscire a trasmettere un frame e meno collisioni rispetto a 1-Persistente.**

CSMA P-Persistente:

- **-Si applica a canali slotted;**
- **-Quando una stazione deve trasmettere, ascolta il canale:**
  - **-se è occupato, aspetta il prossimo slot e ricomincia da capo;**
  - **-se è libero:**
    - **-con probabilità p , trasmette subito.**
    - **-con probabilità 1-p , aspetta il prossimo slot; se anch'esso è libero, riapplica il procedimento.**
- **-Il processo si ripete finché:**
  - **-il frame è trasmesso, oppure**
  - **-qualcun altro ha iniziato a trasmettere. In questo caso la stazione si comporta come in una collisione: aspetta un tempo random e ricomincia da capo.**
- **-Intuitivamente, al diminuire di p ci si aspettano crescenti ritardi prima di riuscire a trasmettere un frame ed una progressiva diminuzione delle collisioni.**

Un miglioramento, si può ottenere consentendo ad ogni stazione di annullare la propria trasmissione in caso di collisione. Ciò permette di risparmiare tempo e banda. Un protocollo che soddisfa questo miglioramento, è CSMA/CD (CSMA con Collision Detection).

24. Basic bitmap.

[Sottostrato MAC]
Basic bitmap fa parte di quei protocolli che risolvono la contesa per il canale senza generare alcuna collisione, nemmeno durante il periodo di contesa.
Supponiamo che ci siano N stazioni, ognuna associata a un indirizzo univoco che varia da 0 a N-1.
Nel metodo a mappa di bit elementare, ogni periodo di contesa è composto da esattamente N intervalli. Se la stazione 0 ha un un frame da inviare, trasmette un bit 1 durante l'intervallo 0. Durante quest'intervallo, a nessun'altra stazione è concesso trasmettere.
In generale, la stazione j trasmette un 1 nello slot temporale j, se ha un frame da inviare.
Una volta trascorsi tutti gli N intervalli, ogni stazione sa quali sono le stazioni che devono trasmettere. A questo punto, le stazioni iniziano a trasmettere in ordine numerico. Essendo tutti d'accordo su chi sarà il prossimo, non ci sarà mai alcuna collisione.
Dopo che l'ultima stazione pronta ha trasmesso il proprio frame (può essere facilmente monitorato da tutte le stazioni), ha inizio un altro periodo di contesa.
Questo genere di protocolli, nei quali la volontà di trasmettere è comunicato a tutti prima di iniziare la trasmissione, si chiamano protocolli a prenotazione.
Efficienza:
Se il carico è elevato, ossia se tutte le stazioni hanno sempre qualcosa da trasmettere, il periodo di contesa di N bit è distribuito proporzionalmente su N frame, quindi un solo bit per frame è usato per il controllo e l'efficienza è d/(d+1) dove d rappresenta la taglia del frame.
Quindi, c'è un solo bit "sprecato".

Problemi:
-Il problema del bit-map è che abbiamo un bit per ogni stazione, quindi con tante stazioni il periodo di contesa può diventare molto lungo.

25. Spiegare in cosa consiste il protocollo Collision free Binary Countdown, pregi e difetti.

[Sottostrato MAC]
Basic bitmap è poco adatto a reti con molte stazioni, in quanto con tante stazioni il periodo di contesa può diventare molto lungo.
Una possibile alternativa, è l'utilizzo del protocollo Collision free Binary Countdown (=Conteggio Binario).
Anche questo protocollo, fa quindi parte di quei protocolli che risolvono la contesa per il canale senza generare alcuna collisione, nemmeno durante il periodo di contesa.
Supponiamo che ci siano N stazioni, ognuna associata a un indirizzo univoco che varia da 0 a N-1.
Una stazione che desidera utilizzare il canale deve comunicare a tutti il proprio indirizzo sotto forma di stringa binaria, partendo dal bit più significativo. Tutti gli indirizzi hanno la stessa lunghezza. I bit che occupano la stessa posizione negli indirizzi di stazioni diverse sono elaborati mediante operatore OR.
Per evitare conflitti, si deve applicare una regola di arbitraggio: la stazione rinuncia non appena si accorge che è stata sovrascritta da un 1 una posizione di bit di ordine elevato che nel proprio indirizzo vale 0.
In questo modo, la precedenza l'avrà chi ha il numero di stazione più grande.
Dopo aver "vinto la gara", la stazione può trasmettere un frame, e al termine della trasmissione ricomincia un nuovo turno.
Le stazioni con il numero più alto, hanno quindi priorità maggiore rispetto alle stazioni con un numero più basso. (es, 0010,0100,1001, 1010: nel primo tempo di bit, le stazioni trasmettono rispettivamente 0,0,1,1. solamente le ultime due stazioni proseguono la contesa. ... vince 1010).
Efficienza:
-Con questo metodo, l'efficienza del canale è $d/(d+\log_2 N)$
-Visto che di solito il frame ha già l'informazione sull'indirizzo della stazione, l'efficienza può arrivare anche al 100%.
Problema:
- **-Le stazioni sono in priorità, quindi si rischia lo stallo di quelle a priorità più bassa.**

Possibili soluzioni:
- **-Metodo probabilistico: ad ogni round, ogni stazione si riassegna una priorità random.**
  - **-possibili collisioni**
  - **-rischio di ritardi o situazioni non eque, e perdiamo l'informazione!**
- **-Manteniamo l'informazione che abbiamo, cioè quella che ci dice se la stazione ha o non ha trasmesso.**
  - **-la stazione aumenta la priorità di 1 se non ha trasmesso;**
  - **-priorità a 0 se ha trasmesso.**
  - **-si mantiene la separazione delle priorità e si evita lo stallo.**

26. Spiegare che cos'è l'Adaptive Tree Walk Protocol.

[sottostrato MAC]
Un modo particolarmente semplice per eseguire l'assegnazione del canale, si basa su un algoritmo facente parte dei protocolli a contesa limitata, chiamato Adaptive Tree Walk Protocol.
I protocolli a contesa limitata:

- **aumentano la competizione quando ci sono poche stazioni**
- **diminuiscono la competizione quando ci sono tante stazioni.**

E' utile immaginare le stazioni come foglie di una struttura ad albero binaria.
Nel primo intervallo di contesa che segue una trasmissione senza collisioni, l'intervallo 0, tutte le stazioni possono tentare di acquisire il controllo del canale.
Se una ci riesce, bene; altrimenti, in caso di collisione, durante l'intervallo 1 possono competere solo quelle stazioni che si trovano sotto il nodo 2 della struttura ad albero. Se una di queste riesce ad acquisire il controllo del canale, l'intervallo successivo è riservato alle stazioni che si trovano sotto il nodo 3. Se d'altra parte, due o più stazioni sotto il nodo 2 vogliono trasmettere, allora durante l'intervallo 1 ci sarà una collisione e di conseguenza durante l'intervallo 2 il turno passerà al nodo 4.

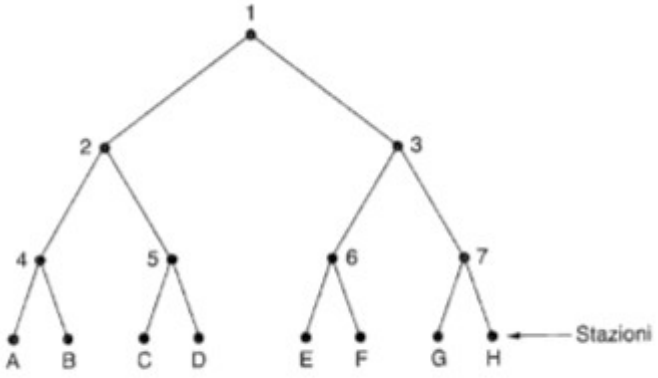


Figura 4.9. La struttura ad albero nel caso di otto stazioni.

In definitiva, in caso di collisione durante l'intervallo 0, l'algoritmo analizza l'intera struttura ad albero partendo dal basso per individuare tutte le stazioni pronte. Ogni intervallo di bit, è associato a qualche particolare nodo dell'albero.
In caso di collisioni, la ricerca continua in modo ricorsivo con gli elementi figli posti a sinistra e a destra del nodo. Se un intervallo di bit è libero, oppure se una sola stazione trasmette durante quel periodo, la ricerca del suo nodo può interrompersi perché tutte le stazioni pronte sono state individuate.
Il principio utilizzato, consiste nel diminuire in modo dinamico la competizione, selezionando una sottoparte dell'albero.
Possiamo fare ancora meglio: analizzando il traffico recente, possiamo ad esempio renderci conto di quante sono le stazioni che cercano di trasmettere in quel lasso di tempo.
Se sappiamo quante sono le stazioni attive, è inutile sprecare tempo a cercare proprio dalla cima dell'albero. Potremmo dunque cominciare direttamente da un sotto-pezzo dell'albero, più precisamente in $P=\log_2 A$, con P=profondità dell'albero, A=stazioni attive.

27. Ethernet e i vari tipi di cavo.

Un importante esempio pratico di protocollo multi accesso, è dato dallo standard IEEE 802.3, cioè Ethernet.
Il nome Ethernet, si riferisce al cavo (definito "etere"); analizziamo questo componente.
Generalmente, si utilizzano quattro tipi di cablaggio(cavi):

{X Base Y,
con X=Banda in Mbps,
Base=connessione baseband(a frequenza unica),
Y=Tipo di cavo.
}

-Coassiale spesso (10 Base5)
-Coassiale sottile (10 Base 2)
-Doppino intrecciato (10 Base-T)
-Fibra ottica (10 Base-F)

-10 Base 5

-E' il più vecchio;
-Consiste di un cavo coassiale spesso. (Thick Ethernet).
-10 Mbps;
-Baseband signaling;
-500m di lunghezza massima
-Possono essere installate fino a 100 macchine su un segmento.
-Ogni stazione contiene un'interfaccia di rete, alla quale viene collegata un' estremità di un cavo corto (pochi mm), detto transceiver drop cable, all'altra estremità del quale è connesso un transceiver che si aggancia con un dispositivo detto a vampiro, al cavo thick (che non viene interrotto).

-10 Base 2

-L'evoluzione di 10 Base 5, più economico e più affidabile.
-Fatto con cavo coassiale sottile, si piega più facilmente (Thin Ethernet).
-10 Mbps;
-Baseband signaling;
-200 metri di lunghezza massima per un singolo segmento.
-Possono essere installate 30 macchine su un segmento.
-Di norma, l'interfaccia di rete contiene anche il transceiver.
-L'allaccio di una stazione alla rete avviene con una giunzione a T , alla quale sono collegati il cavo che porta alla stazione e due cavi thin che costituiscono una porzione del segmento.

-10 Base T

-Usa il doppino telefonico. 10 Base T=Twisted, poco costoso e ancora più affidabile.
-Prevede il collegamento fra una sola coppia di stazioni.
-Per connettere più di due stazioni, serve un HUB (=ripetitore multiporta)
-Un ripetitore è un dispositivo che opera al livello fisico: riceve il segnale da un segmento, lo amplifica e lo ritrasmette su tutti gli altri segmenti.
-Svantaggi:
-La massima lunghezza di ogni segmento scende a 100 metri.

-Vantaggi:

-Il numero di stazioni per segmento cresce a 1024.

-10 Base F

-Usa la fibra ottica

-Vantaggi:

-Permette segmenti fino a 2km

-Molto fine=>ideale per connessioni tra edifici

-Buona immunità alle interferenze

-Più sicuro (tapping molto più difficile).

-Svantaggio:

-E' un'alternativa costosa

Ogni versione di Ethernet, ha una lunghezza massima per segmento. Per creare reti più grandi, si possono collegare tra loro più cavi facendo uso di ripetitori.
Un ripetitore è un dispositivo che opera sullo strato fisico: riceve, amplifica e ritrasmette il segnale.

Nome	Cavo	Lunghezza max. del segmento	Nodi/ segmento	Vantaggi
10Base5	Coassiale spesso (thick Ethernet)	500 m	100	Cavo originale, ora obsoleto
10Base2	Coassiale sottile (thin Ethernet)	185 m	30	Non occorre un hub
10Base-T	Doppino intrecciato	100 m	1.024	Il sistema più economico
10Base-F	Fibra ottica	2.000 m	1.024	Il migliore fra edifici

Figura 4.13. I tipi più comuni di cavi Ethernet.

28. Codifica Manchester.

In 802.3 (Ethernet) non si usa una codifica diretta dei dati (ad esempio, zero volt per lo zero e cinque volt per l'uno), perché sarebbe difficile rilevare le collisioni. Inoltre, si vuole delimitare con facilità l'inizio e la fine di ogni singolo bit.
Viene quindi utilizzata una codifica, detta Manchester, che prevede una transizione del valore del segnale nel mezzo di ogni bit, zero o uno che sia. Ogni periodo di bit è infatti diviso in due intervalli uguali. (Es: L'1 binario è inviato scegliendo un livello di tensione alto durante il primo intervallo e un livello basso durante il secondo; lo schema contrario è utilizzato per rappresentare invece lo 0).
Questa tecnica aiuta la sincronizzazione del trasmettitore e del ricevitore.
-Vantaggi:

-facilità di sincronizzazione fra mittente e destinatario;

-è facile rilevare le collisioni.

-Svantaggi:

-dimezza la banda.

Vi è poi una variante: codifica Manchester differenziale. Richiede dispositivi più complessi, ma offre una maggiore immunità ai rumori.
Ethernet la codifica Manchester "semplice".

29. Frame ethernet.

Il frame Ethernet costituisce l'unità elementare di informazione per il sottolivello MAC di IEEE 802.3. La struttura di un frame è la seguente:

-Preamble: 7 byte
- **-7 byte tutti uguali a 10101010. Servono a "svegliare" il ricevente e a sincronizzarlo con il mittente.**

-Start of the Frame: 1 byte
- **-10101011: indica al destinatario che dal prossimo byte, avrà inizio il frame vero e proprio.**

-Destination Address: 6 byte
-Source Address: 6 byte
- **-gli indirizzi sono univoci a livello mondiale. E' possibile specificare un singolo destinatario, un gruppo di destinatari (multicast), oppure un invio in broadcast a tutte le stazioni.**

-Lunghezza dei dati: 2 byte
- **-indica quanti byte ci sono nel campo dati (0-1500)**

-Dati: 0-1500
- **-Sono i dati veri e propri. Se sono meno di 46 byte, occorre aggiungere dei byte supplementari di riempimento per arrivare almeno a 46 byte. Ogni frame sarà dunque lungo almeno 64 byte.**

-PAD: 0-46
- **-Se il frame è più corto di 64 byte, con questo campo lo si porta a 64 byte.**

-Checksum: 4 byte
- **-è un codice CRC.**

30. Cos'è il Binary Exponential Backoff?

E' un algoritmo che gestisce l'attesa casuale dopo una collisione (in IEEE 802.3 = Ethernet) ; il tempo di attesa viene scelto dinamicamente e casualmente.

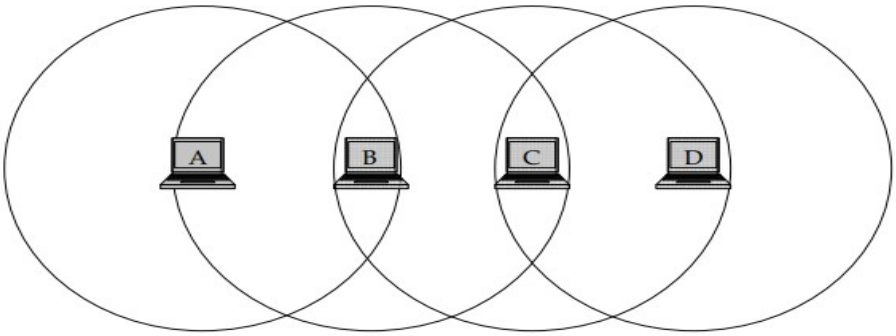
Vediamone il funzionamento:
-Dopo una collisione, il tempo è diviso in intervalli discreti, la cui lunghezza è uguale al tempo di propagazione di andata e ritorno nel caso peggiore sul mezzo di trasmissione (51.2 μsec, o 512 bit).
-Il tempo di attesa prima della prossima ritrasmissione è un multiplo intero dell'intervallo di tempo, e viene scelto a caso in un intervallo i cui estremi dipendono da quante collisioni sono avvenute;
-Dopo n collisioni, il numero r di slot temporali da lasciar passare è scelto a caso nell'intervallo $0<=r<=2^n-1$.
- **-dopo dieci collisioni (n=10), il tetto massimo dell'intervallo di scelta rimane n=10.**
- **-dopo sedici collisioni, viene comunicato un errore al livello superiore.**

-La crescita esponenziale dell'intervallo garantisce una buona adattabilità ad un numero variabile di stazioni, infatti:

- **-se il range fosse sempre piccolo, con molte stazioni si avrebbero praticamente sempre collisioni;**
- **-se il range fosse sempre grande, non ci sarebbero quasi mai collisioni ma il ritardo medio causato da una collisione sarebbe molto elevato.**

31. Stazione nascosta e stazione esposta: cosa sono e come si comportano.

Ogni apparato trasmittente è caratterizzato da una portata, dipendente dalla potenza trasmissiva impiegata, che è la distanza massima alla quale il segnale emesso può essere rilevato. Tutte le apparecchiature entro la portata di un apparato ricevono il segnale trasmesso da tale apparato, mentre quelle al di fuori di tale portata non lo ricevono.



Problema della Stazione nascosta:
Supponiamo ora che la stazione A voglia trasmettere a B. Se C controlla la presenza di portante sul mezzo di trasmissione (CSMA!), non potrà rilevare A, perché si trova al di fuori della sua portata; di conseguenza C pensa erroneamente di poter trasmettere a B. Se inizia a trasmettere, C interferisce con B distruggendo il frame inviato da A.
Il problema di una stazione che non è in grado di rilevare i potenziali concorrenti sul mezzo di trasmissione a causa della distanza eccessiva è chiamato problema della stazione nascosta.

Problema della Stazione esposta:
Consideriamo ora la situazione inversa: B trasmette ad A. Se C controlla la presenza di portante sul mezzo di trasmissione, rileva una trasmissione in atto ed erroneamente pensa di non poter inviare dati a D; in realtà la trasmissione causerebbe una cattiva ricezione solo nella zona compresa tra B e C, che non ospita nessuno dei ricevitori designati.
La situazione genera quello che è chiamato problema della stazione esposta.

Una semplice soluzione è data dal protocollo MACA (Multiple Access with Collision Avoidance). Questo protocollo, sfrutta l'idea che chi deve trasmettere renda il suo spazio locale "conosciuto" anche agli altri, tramite frame RTS (request to send) e CTS (clear to send).

32. Bluetooth.

Bluetooth in origine, è il nome di un progetto volto alla realizzazione di uno standard wireless che permettere il collegamento tra dispositivi di calcolo, di comunicazione e accessori vari mediante un sistema radio wireless a basso costo, a bassa potenza e a portata ridotta.
-Architettura Bluetooth:
L'unità base di un sistema Bluetooth è la piconet, composta da un nodo master e da diversi (non più di sette) nodi slave attivi situati entro un raggio di 10 metri.
Un sistema di piconet interconnesse è chiamato scatternet.
Tutta la comunicazione avviene tra il nodo master e un nodo slave; non è ammessa alcuna comunicazione diretta tra nodi slave.
Il cuore della piconet usa TDM: il nodo master controlla chi può comunicare ad ogni intervallo.

Si può scegliere fra vari servizi Bluetooth:
- **-accesso generico**
- **-scoperta del servizio**
- **-dial-up**
- **-auricolari**
- **-trasferimento file**
- **-ecc...**

Tipi di link (tra master e slave)
- **-ACL (Asynchronous Connection-Less)**
  - **-i frame possono essere persi e quindi ritrasmessi**
- **-SCO (Synchronous Connetion-Oriented)**
  - **-i frame non vengono ritrasmessi ma si fa error correction**

//FRAME Bluetooth

FINE CAPITOLO 4.
CAPITOLO 5

33. Si descriva l'algoritmo statico Flooding(open loop).

[Strato Network]
La tecnica del flooding (“alluvione”) è un algoritmo di routing statico, che consiste nell'inviare ogni pacchetto su tutte le linee eccetto quella da cui è arrivato.
In linea di principio, il flooding può essere usato come algoritmo di routing (ogni pacchetto inviato arriva a tutti i router), ma presenta l'inconveniente di generare un numero enorme (teoricamente infinito) di pacchetti.

Ci sono delle tecniche per limitare il traffico generato:

1) **Inserire in ogni pacchetto un contatore che viene decrementato ad ogni hop (hop-counting). Quando il contatore arriva a zero, il pacchetto viene scartato (muore). Un appropriato valore iniziale può essere il diametro della subnet.**
2) **Inserire la coppia (source router ID, sequence number) in ogni pacchetto. Ogni router esamina tali informazioni e ne tiene traccia; quando vede per la seconda volta lo stesso pacchetto, lo scarta.**
3) **Selective flooding: i pacchetti vengono duplicati solo sulle linee che vanno all'incirca nella giusta direzione.**

Il flooding non è utilizzabile in generale come algoritmo di routing, però:

-utile in campo militare (massima affidabilità e robustezza);
-utile per l'aggiornamento contemporaneo di informazioni distribuite;
-utile come strumento di paragone per altri algoritmi, visto che trova sempre, fra gli altri, il cammino minimo.

34. Descrivere il Distance Vector Routing(DVR)(closed loop).

[Strato Network]
E' un algoritmo di routing dinamico, che tiene conto del carico istantaneo della rete.
L'idea è che ogni router ha una tabella di routing che contiene informazioni su quanto veloce è la connessione ad un altro router, e qual'è la via migliore (tra i primi vicini) per raggiungerlo.
Distance vector routing:

- **Funziona in modo semplice.**
- **Ogni router chiede ai suoi vicini la loro tabella ad intervalli regolari.**
- **Usa poi le loro tabelle, ed il tempo che c'è voluto per averle, per costruire la propria tabella selezionando i percorsi migliori.**

Pro:

- **Veloce a recepire le "buone notizie"**

Contro:

- **Non tiene conto della capacità della banda**
- **Quanto questo routing si comporta bene rispetto alle "buone notizie", tanto male, allo stesso modo, si comporta con le "cattive notizie", cioè se la linea cade. Ciò è legato al fatto che i router non conoscono la topologia della rete.**
- **Problema del conteggio all'infinito**

Questo algoritmo, è inoltre esposto al problema del cout-to-infinity (conteggio all'infinito): consideriamo questo esempio:

```
A     B     C     D     E   <- Router
*-----*-----*-----*-----*   <- Collegamenti (topologia lineare)
      1     2     3     4   <- Distanze da A
```

Se ora cade la linea fra A e B, dopo uno scambio succede questo:

```
A     B     C     D     E   <- Router
*     *-----*-----*-----*   <- Collegamenti
      3     2     3     4   <- Distanze da A (dopo uno scambio)
```

Ciò avviene perché B, non ricevendo risposta da A, crede di poterci arrivare via C, che ha distanza 2 da A.
A lungo andare, tutti i router vedono lentamente aumentare sempre più la distanza per arrivare ad A. Questo è il problema del cout-to-infinity.
Non è stata trovata una soluzione veramente efficacie a questo problema.

35. Link State Routing.

[Strato Network]
E' un algoritmo di routing dinamico, basato sullo stato dei collegamenti (Link State Routing). Soprattutto a causa della lentezza di convergenza (problema del conteggio all'infinito) del Distance Vector Routing, si è cercato un approccio diverso, che ha dato origine al Link State Routing.
L'idea è questa:

-Ogni router tiene sotto controllo lo stato dei collegamenti fra se e i suoi vicini immediati, misurando il ritardo di ogni linea, e distribuisce queste informazioni a tutti gli altri router;

-Sulla base di queste informazioni, ogni router ricostruisce localmente la topologia completa dell'intera rete e calcola il cammino minimo fra se e tutti gli altri.

I passi da seguire sono:

1) **Scoprire i vicini e identificarli.**
2) **Misurare il costo (ritardo) delle relative linee.**
3) **Costruire un pacchetto con tali informazioni.**
4) **Mandare il pacchetto a tutti gli altri router.**
5) **Calcolare il cammino più breve a tutti gli altri router.**

1)
Quando il router si avvia, invia un pacchetto HELLO su tutte le linee in uscita. In risposta riceve dai vicini i loro indirizzi (univoci su tutta la rete).
2)
Inviando vari pacchetti ECHO, misurando il tempo di arrivo della risposta (diviso 2) , è possibile calcolare una stima del ritardo della linea.
3)
Si costruisce un pacchetto con:

- **identità del mittente.**
- **numero di sequenza del pacchetto(per tenere in considerazione quello più aggiornato).**
- **età del pacchetto.**
- **lista dei vicini con i relativi ritardi.**

La costruzione e l'invio di tali pacchetti si verifica tipicamente:

- **A intervalli regolari; oppure**
- **Quando accade un evento significativo (es. una linea cade e ritorna su).**

4)
La distribuzione dei pacchetti è la parte più delicata, perché errori in questa fase possono portare qualche router ad avere idee sbagliate sulla topologia, con conseguenti malfunzionamenti.
Di base, si utilizza il flooding, inserendo nei pacchetti le coppie (source router ID, sequence number) per eliminare duplicati; numero di sequenza, serve a prevenire il caso in cui un pacchetto con l'informazione locale arrivi dopo un altro pacchetto con informazione locale più aggiornata. Tutti i pacchetti sono confermati. Inoltre, per evitare che pacchetti vaganti (per qualche errore) girino per sempre, l'età del pacchetto viene decrementata ogni secondo, e quando arriva a zero il pacchetto viene scartato.

5)
Combinando tutte le informazioni arrivate, ogni router costruisce il grafo della subnet e calcola il cammino minimo a tutti gli altri router.

Problemi:

-Più grande è la rete → tabelle di routing più grandi → maggior tempo di refresh.

36. Choke Bucket.

[Strato Network]
Choke Bucket è un algoritmo per il controllo della congestione.
Come nel flow control, alle volte il ricevente può segnalare che le cose non vanno bene: anche qui l'idea è che se un router si accorge che c'è cogestione, può inviare un pacchetto speciale (choke packet, to choke= soffocare) a chi sta inviando dati, dicendogli di rallentare.
Quando l'host sorgente riceve il choke packet, diminuisce il flusso (tipicamente lo dimezza) e ignora i successivi choke packet per un tempo prefissato, perché tipicamente ne arriveranno molti in sequenza.
Trascorso tale tempo prefissato, l'host si rimette in attesa di altri choke packet. Se ne arrivano altri, riduce ancora il flusso. Altrimenti, aumenta di nuovo il flusso, adottando però incrementi più piccoli in modo tale da evitare la ricomparsa della congestione. Non appena si riceve un choke, per un certo periodo di tempo si ignorano le altre richieste di choke che vengono dalla stessa destinazione.
Problema/inconveniente:

-Se la rete è grande (internet!), una richiesta di choke può metterci troppo tempo per sistemare le cose. → soluzione: hop-by-hop choke

37. Token hop-by-hop.

[Strato Network]
Token hop-by-hop è un algoritmo per il controllo della congestione.
Se la rete è grande, (Internet ad esempio), una richiesta di choke può metterci troppo tempo per sistemare le cose. L'unico problema del choke packet, è infatti dovuta alla lentezza di reazione, perché l'host che produce i pacchetti ci mette un certo tempo a ricevere i choke packet e a diminuire di conseguenza il ritmo della trasmissione. Per migliorare le cose, si può costringere ogni router sul percorso, appena riceve tali pacchetti, a rallentare subito il ritmo. In tal caso si parla di hop-by-hop choke packet.
Questa tecnica, rende molto più veloce il sollievo del router che ha per primo i problemi di congestione, ma richiede più spazio di buffer nei router sul percorso dall'host originario a quel router.

38. Load Shedding.

[Strato Network]
Quando nessun altro metodo riesce a eliminare la congestione(choke packet/token hop-by-hop), i router possono tirar fuori "l'artiglieria pesante"; gettare via il carico. La tecnica Load Shedding fa sì che in caso di sovraccarico, alcuni pacchetti semplicemente vengano buttati via. Ovviamente, si può fare in modo intelligente, non buttando via pacchetti a caso, ma con raziocinio.

-Wine (vino)
- **-Il vecchio è migliore del nuovo; ad esempio, per la trasmissione dei file, conviene buttar via i pacchetti più nuovi rispetto ai vecchi.**

-Milk (latte)
- **-Il nuovo è migliore del vecchio; ad esempio, in un file multimediale conviene buttar via un pacchetto vecchio; anche se c'è qualche salto nella trasmissione, l'importante è che "the show must go on" .**

Un ulteriore passo avanti richiede la cooperazione dei trasmettitori. Per molte applicazioni, alcuni pacchetti sono più importanti di altri. Per implementare un criterio di eliminazione intelligente, le applicazioni devono contrassegnare i loro pacchetti in classi di priorità in modo da indicare la loro importanza. Se lo fanno, i router possono scartare i pacchetti partendo da quelli di classe più bassa.

39. RED.

[Strato Network]
Fa parte degli algoritmi per il controllo della congestione.
Risulta più semplice gestire la congestione appena viene rilevata piuttosto che cercare di porvi rimedio dopo averle dato il tempo di bloccare tutto. Questa osservazione conduce all'idea di scartare i pacchetti prima che tutto lo spazio del buffer sia completamente esaurito. Un celebre algoritmo usato per mettere in pratica questo schema è chiamato RED (Random Early Detection).
In pratica, facendo in modo che i router scartino i pacchetti prima che la situazione diventi senza speranza (early!), è possibile bloccare il problema sul nascere. Per stabilire quando è il momento giusto per iniziare a scartare i pacchetti, i router mantengono una media mobile delle lunghezze delle code. Quando la lunghezza media della coda su una linea supera una soglia di guardia, la linea è considerata congestionata e viene intrapresa l'azione di correzione.
Viene scelto a caso, un pacchetto dalla coda che ha attivato l'azione di correzione.
In che modo il router dovrebbe avvisare la sorgente del problema?
Una strategia è quella che scarta semplicemente il pacchetto selezionato, senza rendere nota l'operazione. La sorgente alla fine noterà l'assenza del pacchetto ACK e prenderà un'iniziativa. La sorgente,sapendo che i pacchetti persi sono causati generalmente da congestioni, risponderà rallentando il flusso anziché ritentare in modo più accanito l'invio del pacchetto.

Nelle reti wireless, dove la maggior parte delle perdite è causata dal rumore presente nel collegamento aereo, questo approccio non può essere utilizzato.

40. Reverse Path Forwarding.

[Strato Network]
Reverse Path Forwarding è un algoritmo di routing, usato per trasmettere in modalità broadcast, che tenta di approssimare il comportamento dell'algoritmo “spanning tree”, anche quando i router non sanno nulla degli spanning tree.

Quando riceve un pacchetto broadcast, il router verifica se il pacchetto è giunto attraverso la linea che normalmente è utilizzata per inviare i pacchetti alla sorgente della trasmissione broadcast.

- **In caso affermativo, c'è una forte probabilità che il pacchetto broadcast stesso abbia seguito il percorso migliore dal router, e che perciò sia la prima copia arrivata al router. In questo caso, il router inoltra le copie del pacchetto attraverso tutte le linee esclusa quella di input.**
- **Se al contrario, il pacchetto broadcast è giunto attraverso una linea diversa da quelle che viene preferita per raggiungere la sorgente, il pacchetto è scartato in quanto è probabile che si tratti di un duplicato.**

Sostanzialmente, funziona come il flooding, solo che vengono considerati solo i pacchetti provenienti dal cammino migliore, mentre tutti gli altri sono ignorati.
Il vantaggio principale del Reverse Path Forwarding è che si tratta di un sistema ragionevolmente efficiente e facile da implementare, non richiede una lista di destinazioni o una mappa di bit per ogni pacchetto broadcast e non necessita nemmeno di un meccanismo speciale d'interruzione del processo, come invece richiesto ad esempio dal flooding.

41. Quality of Service

[Strato Network]
Un flusso di pacchetti diretto da una sorgente ad una destinazione è chiamato semplicemente flusso.
In una rete orientata alle connessioni tutti i pacchetti che appartengono a un flusso seguono lo stesso percorso, a differenza invece delle reti senza connessioni dove i pacchetti possono seguire percorsi differenti.
Le esigenze di ogni flusso possono essere caratterizzate da quattro parametri, che insieme determinano la QoS (Quality of Service), ossia la qualità del servizio richiesta dal flusso, e sono:
-Affidabilità
-Ritardo
-Jitter = misura il grado di variazione nei tempi di arrivo dei pacchetti.
-Banda

AFFIDABILITÀ:
Il trasferimento dei file ad esempio, ha requisiti rigidi sull'affidabilità; nessun bit può essere trasmesso in modo scorretto. Questo obiettivo di solito è raggiunto creando il checksum di ogni pacchetto e verificandolo alla destinazione. Il pacchetto in transito che arriva danneggiato non riceve ackowledgement, e sarà di conseguenza ritrasmesso.
Audio e video, possono invece tollerare errori, perciò nessun checksum è elaborato o verificato.
RITARDO:
Le applicazioni di trasferimento file, non sono sensibili al ritardo. Se tutti i pacchetti subiscono un ritardo uniforme di pochi secondi non accade nulla di male.
Applicazioni in tempo reale (telefonia, videoconferenza) hanno invece requisiti severi nel ritardo.
JITTER:
Il trasferimento dei file ad esempio, non è sensibile ai pacchetti che arrivano a intervalli irregolari. Il video e soprattutto l'audio sono estremamente sensibili allo jitter. Se un utente sta guardando un video trasmesso attraverso la rete e i fotogrammi hanno un ritardo di esattamente 2 secondi, non accade nulla di male; se però il tempo di trasmissione dei pacchetti varia in modo casuale tra 1 e 2 secondi, il risultato sarà terribile.
BANDA:
La posta elettronica non richiede molta banda; il video invece, ne richiede molta.

Applicazione	Affidabilità	Ritardo	Jitter	Banda
Posta elettronica	Alta	Bassa	Bassa	Bassa
Trasferimento file	Alta	Bassa	Bassa	Media
Accesso al Web	Alta	Media	Bassa	Media
Login remoto	Alta	Media	Media	Bassa
Audio a richiesta	Bassa	Bassa	Alta	Media
Video a richiesta	Bassa	Bassa	Alta	Alta
Telefonia	Bassa	Alta	Alta	Bassa
Videoconferenza	Bassa	Alta	Alta	Alta

Figura 5.30. Rigidità dei requisiti relativi alla qualità del servizio.

Tecniche per ottenere una buona QoS:

- **sovradimensionamento**
- **utilizzo del buffer**
- **Traffic Shapping**
- **choke packet**
- **token hop-by-hop**
- **load shedding**
- **Leaky Bucket**
- **Token Bucket**

42. Leaky Bucket, pregi e difetti.

[Strato Network]
Leaky Bucket è una tecnica per migliorare la QoS, attraverso il controllo della congestione. L'idea è semplice, e trova un analogia reale in un secchio che viene riempito da un rubinetto (che può essere continuamente manovrato in modo da risultare più o meno aperto) e riversa l'acqua che contiene attraverso un forellino sul fondo, a ritmo costante. Se viene immessa troppa acqua, essa fuoriesce dal bordo superiore del secchio e si perde.

Sull'host si realizza (nell'interfaccia di rete o in un software) un leaky bucket, che è autorizzato a riversare sulla rete pacchetti con un fissato data rate (diciamo b bps) e che mantiene, nei suoi buffer, quelli accodati per la trasmissione.

Se l'host genera più pacchetti di quelli che possono essere contenuti nei buffer, essi si perdono.

In questo modo, l'host può anche produrre un traffico bursty senza creare problemi sulla rete; finché il data rate medio non supera i b bps tutto funziona regolarmente, oltre si cominciano a perdere pacchetti.

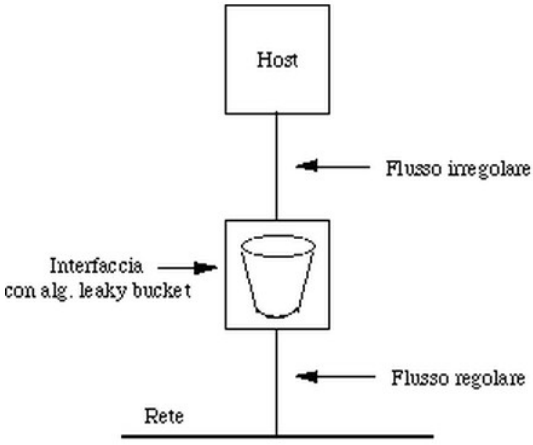


Figura 5-4: Leaky bucket

Il meccanismo trasforma dunque un flusso irregolare di pacchetti provenienti dai processi dell'utente dell'host, in un flusso regolare di pacchetti immesso nella rete, appianando i picchi e riducendo enormemente la possibilità di congestioni.

E' facile implementare l'algoritmo originale leaky bucket. Esso, è composto da una coda finita. All'arrivo del pacchetto, se c'è spazio il pacchetto viene aggiunto alla coda, altrimenti viene scartato. Ad ogni ciclo di clock viene trasmesso un pacchetto (a meno che la coda non sia vuota).

-Vantaggio/Svantaggio:
- **-Data rate sempre costante.**
  - **-quando il traffico è più sostenuto, converrebbe ad esempio aumentare un po' il data rate.**
  - **-non segue la variabilità del traffico.**
- **-Facile da implementare**

43. Descrivere il token bucket, pregi e difetti.

[Strato Network]
Token Bucket è una tecnica per migliorare la QoS, attraverso il controllo della congestione.
Algoritmo token bucket (secchio a gettoni)
E' una tecnica per consentire un grado di irregolarità controllato anche nel flusso che esce sulla rete.
Essenzialmente, si accumula un credito trasmissivo con un certo data rate (fino al massimo consentito) quando non si trasmette nulla.
Quando poi c'è da trasmettere, lo si fa sfruttando tutto il credito disponibile per trasmettere, fino all'esaurimento di tale credito, alla massima velocità consentita dalla linea.
Il secchio contiene dei token, che si creano con una cadenza prefissata (es ogni msec), fino a che il loro numero raggiunge un valore M prefissato che corrisponde all'aver riempito il secchio di token.
Per poter trasmettere un pacchetto (o una certa quantità di byte), deve essere disponibile un token.
Se ci sono k token nel secchio e h>k pacchetti da trasmettere, i primi k sono trasmessi subito (al data rate consentito dalla linea) e gli altri devono aspettare dei nuovi token. (figura).
Potenzialmente quindi, dei burst di M pacchetti possono essere trasmessi in un colpo solo, fermo restando che mediamente non si riesce a trasmettere ad una velocità più alta di quella di generazione dei token.
Poiché possa essere trasmesso, un pacchetto deve catturare e distruggere un token.

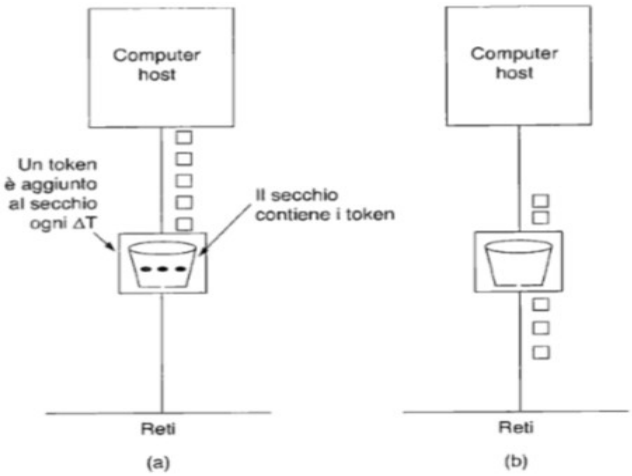


Figura 5.34. L'algoritmo token bucket. (a) Prima. (b) Dopo.

Nella figura, il secchio contiene inizialmente solamente tre token, con cinque pacchetti in attesa di essere trasmessi.
Nella seconda parte della figura, notiamo che tre dei cinque pacchetti sono passati, ma gli altri due sono bloccati in attesa che vengano generati altri token.

Differenze con l'algoritmo leaky bucket:
-L'algoritmo leaky bucket non permette agli host inattivi di risparmiare permessi in modo da inviare successivamente grandi raffiche di dati; l'algoritmo token bucket, consente invece di risparmiare fino a riempire la dimensione massima del secchio, M.
-Un'altra differenza col leaky bucket è che i pacchetti non vengono mai scartati (il secchio contiene token, non pacchetti). Se necessario, si avverte il livello superiore, produttore dei dati, di fermarsi per un po'.

44. Descrivere l'ARP.

[Strato Network]
-ARP (Address Resolution Protocol)
Il protocollo ARP serve per derivare, dall'indirizzo IP dell'host di destinazione, l'indirizzo di livello data link(ad esempio gli indirizzi Ethernet, di 48 bit, MAC) necessario per inviare il frame.

Esso opera appoggiandosi direttamente sul livello data link e non su IP:

- **viene inviata a tutte le stazioni della LAN, in broadcast, una richiesta del tipo: “chi ha l'indirizzo IP uguale a xxx.xxx.xxx.xxx?”**
- **solo l'host che ha quell'indirizzo IP risponde, inserendo nella risposta il proprio indirizzo data link;**

Possibili ottimizzazioni:

- **Quando riceve la risposta, l'host mantiene in memoria l'indirizzo (circa 15 minuti), caso mai dovesse essere necessario ricontattare lo stesso computer.**
- **L'host (mittente) include la propria associazione IP-Ethernet nel pacchetto ARP, che viene poi inserita nella cache dal destinatario, caso mai dovesse contattare l'host mittente.**
- **Fare in modo che ogni computer/host trasmetta in broadcast la propria associazione durante l'accensione.**

Le voci nella cache ARP dovrebbero scadere dopo pochi minuti (circa 15).

45. Si descriva DHCP.

[Strato Network]
-DHCP (Dynamic Host Configuration Protocol).

-Permette l'assegnazione manuale oppure automatica degli indirizzi IP.
-Si basa sull'idea di un server speciale (server DHCP) che assegna gli indirizzi IP agli host che ne richiedono uno. Questo server, non deve trovarsi necessariamente nella stessa LAN dell'host richiedente.
Poiché il server DHCP potrebbe non essere raggiunto dalle trasmissioni broadcast, è necessario installare in ogni LAN un agente di inoltro DHCP (DHCP relay agent).
Per trovare il suo indirizzo IP, una macchina appena accesa invia in modalità broadcast un pacchetto DHCP DISCOVER. L'agente di inoltro DHCP presente su quella LAN intercetta tutte le trasmissioni DHCP; quando individua un pacchetto DHCP DISCOVER, l'agente trasmette il pacchetto in modalità unicast al server DHCP, che può anche trovarsi su una rete distante. L'agente di inoltro ha bisogno di una sola informazione: l'indirizzo IP del server DHCP.

Problema:
Se un host abbandona la rete e non restituisce il proprio indirizzo, quell'indirizzo andrà definitivamente perso, perciò dopo un po' di tempo molti indirizzi IP potrebbero non essere più disponibili.
Per impedire ciò, l'assegnazione dell'indirizzo IP può essere fissata per un periodo di tempo, mediante una tecnica chiamata leasing.
Poco prima della scadenza del leasing, l'host deve chiedere al server DHCP il rinnovo dell'indirizzo. Se non riesce ad ottenere il rinnovo, l'host non può più utilizzare l'indirizzo IP che gli era stato assegnato precedentemente.

46. IPv6.

-E' una nuova versione di IP, successore di IP versione 4.
-I requisiti principali di progetto erano:
- **-aumentare il numero di indirizzi ormai quasi esauriti;**
- **-ottenere una maggiore efficienza nei router (tavole più piccole, routing più veloce);**
- **-supportare meglio il traffico real time;**
- **-offrire maggiore sicurezza ai dati riservati.**

Le principali differenze rispetto alla versione 4 di IP, sono:
- **-indirizzi di 16 byte (anziché 4), il che significa disporre di 2^{128} indirizzi IP;**
- **-header semplificato: 7 campi contro 13 di IPv4;**
- **-IPv6 non ha il campo checksum.**
- **-funzioni di autenticazione e privacy, basate su crittografia;**
- **-gestione della qualità di servizio attraverso un campo flow label che consente di istituire delle pseudo-connessioni con caratteristiche negoziate in anticipo.**

Vediamo i vari campi che compongono l'header di IPv6:
- **-Version: 4 bit (come in IPv4) → ora contiene '6' anziché '4'**
- **-Traffic class: 8 bit**
  - **-è circa l'equivalente di type of service di IPv4 (6 bit)**
- **-Flow label: 20 bit**
  - **-IPv6 ha in più la gestione dei flussi: l'idea è quella di poter creare flussi separati, anche partendo dalla stessa macchina, i quali (flussi) hanno magari hanno esigenze diverse.**
- **-Payload lenght: 16 bit**
  - **-simile al campo Total Lenght di IPv4 , solo che ora dà la lunghezza del payload, escludendo l'header.**
- **-Next header:8 bit**
  - **-serve a gestire le estensioni in IPv6**
- **-Hop limit: 8 bit (come in IPv4)**
  - **-è come il TTL di IPv4**

47. Elencare e descrivere brevemente i secondi (primi) 32 bit dell'header IPv4 (IPv6).

[Strato Network]
La colla che tiene unito Internet è il protocollo dello strato Network IP (Internet Protocol). La comunicazione in internet funziona in questo modo: Lo strato trasporto prende i flussi di dati e li divide in datagrammi. Ogni datagramma viene trasmesso attraverso Internet. Quando alla fine tutti i pezzi raggiungono la macchina destinazione, lo strato Network ricostruisce il datagramma originale e lo inserisce nel flusso di input del processo ricevente.

Un pacchetto IP è costituito da un header e da una parte dati.

L'header ha una parte fissa, di 20 byte e una parte, opzionale, di lunghezza variabile.

Vediamone i primi 32 bit.

-Version: 4 bit
-indica la versione di IP usata, permettendone quindi il versioning.
-IHL: 4 bit
-è la lunghezza dell'header (IP Header Lenght), minimo 5, massimo 15.
-Type of service: 8 bit
-caratterizza affidabilità e velocità richieste; questo campo è adatto per la selezione della QoS, ma di fatto, viene ignorato dai router.
-Total lenght: 16 bit
-indica la lunghezza del pacchetto, inclusi i dati; massimo 65'535 byte

Vediamo anche gli altri campi dell'header di IPv4.

-Identification:
- tutti i frammenti di uno stesso pacchetto hanno lo stesso valore
-DF: dont' fragment
-se uguale a 1, non si deve frammentare il pacchetto, anche a costo di scegliere una strada meno veloce.
-MF: more fragments
-se uguale a 1, il pacchetto non è ancora finito. E' importante sapere quando sono arrivati tutti i frammenti di un datagramma.
-Fragment offset:
-indice del frammento nel pacchetto.
-Time to live:
-contatore (inizializzato a 255), che viene decrementato di uno a ogni hop. Quando arriva a zero, il pacchetto viene scartato.
-Protocol:
-codice del protocollo di livello transport a cui consegnare i dati (es. TCP,UDP)
-Header Checksum:
-checksum di controllo del solo header:
-si sommano, in complemento ad 1, le parole a 16 bit dello header.
-Source e destination address:
-indirizzi mittente e destinatario.
-Options

48. CIDR

[Strato Network]
-CIDR= Classes InterDomain Routing

Dato che l'assegnazione degli indirizzi IP tramite classi causava lo spreco di migliaia di indirizzi, è stato introdotto il CIDR, ovvero un metodo di assegnazione di indirizzi IP che non rispetta i rigidi limiti dettati dalle classi, ma che permette di assegnare gli indirizzi IP rimanenti in blocchi di dimensioni variabili.
Con CIDR, l'inoltro diventa però più complicato: ogni voce della tabella di routing necessita di una maschera di 32 bit (subnet mask, maschere di sottorete).
Questa maschera, contiene l'informazione su quanto grande è il blocco di indirizzi; dei 32 bit, ci sono tanti 0 quanto è largo il blocco di indirizzi, mentre gli altri bit sono a 1.
Gli indirizzi CIDR di solito sono scritti usando la CIDR notation, cioè la dotted notation seguita da uno slash (“/”), e poi dal numero di 1 nella maschera.
Esempio:

- **Blocco di 2048 indirizzi (11 bit)**
- **Gli viene assegnato come primo indirizzo 194.24.0.0**
- **la maschera è composta da 11 bit a 0, e 32-11=21 bit a 1.**
- **11111111111111111111100000000000**
- **in CIDR notation, avremo dunque indirizzi 194.24.0.0/21 (=21 '1')**

Quando arriva un pacchetto, il router estrae l'IP di destinazione e lo compara con la tabella di routing: applica la maschera all'IP estratto e vede se c'è un match tra IP mascherato e primo IP di quella linea di output. Se ci sono più match, viene scelto quello con maschera più lunga.
Se un router non ha linee di output specifiche e separate verso una zona, può utilizzare voci aggregate per diminuire la dimensione della tabella di routing.

FINE CAPITOLO 5.

Daniel De Gaspari

CAPITOLO 6.

49. Si descriva l'header UDP.

[Strato Trasporto]
UDP è un protocollo di trasporto senza connessione.
Offre alle applicazioni un modo per inviare datagrammi IP incapsulati senza dover stabilire una connessione. Essenzialmente, UDP equivale ad, IP con l'aggiunta di una breve intestazione di 8 byte.
UDP aggiunge l'informazione minimale per lo strato Trasporto:

- **la porta di chi manda**
- **la porta di chi riceve**
- **...e fa il controllo dell'errore tramite checksum (opzionale)**

UDP trasmette segmenti costituiti da un'intestazione di 8 byte, seguita dal carico utile.

Vediamo l'header di UDP:

- **Source port: 2 byte**
  - **la porta del mittente**
- **Destination port: 2 byte**
  - **la porta del destinatario**
- **UDP lenght: 2 byte**
  - **lunghezza UDP, include l'intestazione di 8 byte e i dati.**
- **UDP checksum: 2 byte**
  - **è facoltativo, contiene 0 se non elaborato**

Di cosa si occupa UDP:
UDP si occupa di fornire un'interfaccia al protocollo IP, facendo multiplexing delle singole risorse network (le macchine, identificate da indirizzi IP), tramite le porte.

IP+porta=socket

- **è la divisione in “rete logica” al livello Trasporto, di una singola entità in rete.**

50. Descrivere l'header del TCP/IP e commentarlo.

[Strato Trasporto]
TCP è un protocollo di trasporto orientato alla connessione; le connessioni TCP sono full duplex e point-to-point. Di default, il controllo del flusso è gestito con una sliding window a dimensione variabile, con goback-N, e c'è un'opzione per usare il selective repeat.
Ogni segmento TCP, inizia con un'intestazione di 20 byte, con formato fisso.
Un segmento TCP non può superare i 65'535 byte(64 KB); dunque, i dati trasportabili da un segmento sono al massimo 65'535 - 20 (header IP) - 20 (header TCP). Sono inoltre ammessi segmenti senza dati, usati ad esempio per gli ACK.

Analizziamo l'intestazione di TCP:
- **Source port: 2 byte**
- **Destination port: 2 byte**
  - **presenti anche in UDP.**
- **Sequence number: 4 byte**
- **ACK number: 4 byte**
- **TCP header lenght: 1 byte**
  - **detta la lunghezza dell'header, visto che possono esserci "options", come in IP. Questo campo è il numero di words(parole, da 32 bit) dell'header.**
- **6 bit inutilizzati**
  - **per eventuali estensioni**

poi c'è una serie di 6 flags:
- **URG:**
  - **1 se urgent pointer è usato, 0 altrimenti. Per dati urgenti.**
- **ACK:**
  - **indica che c'è un ACK (piggybacking!!!)**
- **PSH:**
  - **dati urgenti (pushed data), da consegnare senza aspettare che il buffer si riempia.**
- **RST:**
  - **indica che c'è un problema nella connessione, e serve a fare il reset.**
- **SYN:**
  - **usato nella fase di setup della connessione**
- **FIN:**
  - **usato per rilasciare una connessione; la terminazione avviene tramite handshaking (con ACK, anche in piggybackng).**

...fine flags
- **Window size:2 byte**
  - **il controllo di flusso è di tipo sliding window con goback-N, e c'è un'opzione per usare il selective repeat , di dimensione variabile.**
    Window size, indica quanti byte possono essere inviati a partire da quello che ha ricevuto ACK.
- **Checksum: 2 byte**
  - **per il controllo dell'errore. Viene calcolato, come anche in UDP, con semplici somme in complemento a 1, e poi si fa il complemento a 1 del risultato.**

FINE CAPITOLO 6.

Daniel De Gaspari

CAPITOLO 7.

51. DNS.

[Strato Applicazione]
DNS-Domain Name System
Poiché riferirsi ad una risorsa (sia essa un host oppure l'indirizzo di posta elettronica di un utente), utilizzando un indirizzo IP numerico (xxx.xxx.xxx.xxx) è estremamente scomodo, si è creato un meccanismo tramite il quale le risorse possono essere identificate tramite un nome logico, cioè una stringa di caratteri, molto più facile e comprensibile, es:
www.unipd.it

La corrispondenza fra gli indirizzi IP numerici ed i nomi logici si effettua mediante l'uso del DNS.
Esso consiste di:

1. **uno schema gerarchico di nominazione, basato sul concetto di dominio (domain).**
2. **un database distribuito, che implementa lo schema di nominazione.**
3. **Un protocollo per il mantenimento e la distribuzione delle informazioni sulle corrispondenze.**

Il funzionamento, in breve, è il seguente:

- **Quando un'applicazione deve collegarsi ad una risorsa di cui conosce il nome logico(www.unipd.it), invia una richiesta al DNS server locale (l'applicazione chiama per questo un'apposita procedura di libreria, detta resolver);**
- **Il DNS server locale, se conosce la risposta, la invia direttamente al richiedente. Altrimenti, interroga a sua volta un DNS server di livello superiore (ordine gerarchico!), e così via. Quando finalmente arriva la risposta, il DNS server locale la passa al richiedente;**
- **Quando l'applicazione riceve la risposta (costituita dal numero IP della risorsa in questione), crea una connessione TCP con la destinazione, usando l'indirizzo IP ricevuto.**

Lo spazio dei nomi DNS, è uno spazio gerarchico, organizzato in domini, ciascuno dei quali può avere dei sotto-domini.
Esistono inoltre un insieme di domini di massimo livello (top-level domain), i più alti nella gerarchia. Es: .com, .edu, .mil, .it, ecc...

Ogni dominio, ha la responsabilità di fornire il servizio DNS per quanto di propria competenza. Ossia, deve poter rispondere a interrogazioni riguardanti tutti gli host contenuti nel dominio stesso.

FINE CAPITOLO 7.

Daniel De Gaspari

CAPITOLO 8.

52. Cos'è un cifrario a sostituzione.

[Sicurezza delle reti]
Cifrario a Sostituzione:
Uno tra i cifrari più semplici, attribuito a Cesare, si basa sul principio di sostituzione di lettere con altre, spostando l'alfabeto di un numero k di caratteri, e k diventa la chiave.
Nel cifrario di Cesare, K=3.
Questi cifrari, conservano l'ordine dei simboli del testo in chiaro, limitandosi a mascherare la loro apparenza.
Un miglioramento, consiste nel far sì che ognuno dei simboli dell'alfabeto, corrisponda ad altre lettere. Questo sistema nel quale si ha una sostituzione simbolo a simbolo, viene chiamato sostituzione monoalfabetica.

Decriptazione:
Nonostante la semplicità, si può pensare che provare tutte le combinazioni sia proibitivo. Vero, ma non è l'approccio corretto per risolvere l'enigma.
In questi casi, si usa un approccio statistico, che si basa sul fatto che le singole lettere, in ogni lingua, hanno una certa frequenza, come ad esempio 'e' in inglese, guardando inoltre i diagrammi più comuni (coppie di lettere), e i trigrammi (combinazioni di tre lettere).
Un altro approccio, è quello di provare con una parola che dato il contesto, ha una buona probabilità di essere nel testo, e da li ricavare le varie lettere.

53. Si descriva l'algoritmo DES e triplo DES.

[Sicurezza delle reti]
DES:
E' un algoritmo a chiave simmetrica. E' un cifrario a blocchi.
Il testo in chiaro viene cifrato in blocchi di 64 bit, che generano 64 bit di testo cifrato.
L'algoritmo è parametrizzato con una chiave a 56 bit, e ha 19 stadi distinti.
Il primo stadio consiste nella trasposizione dei 64 bit del testo in chiaro. L'ultimo stadio è esattamente l'inverso del primo. Il penultimo stadio, consiste nello scambiare i 32 bit più a destra con quelli più a sinistra; i rimanenti 16 stadi intermedi, invece sono funzionalmente uguali, ma parametrizzati con diverse funzioni della chiave.
Ogni stadio intermedio prende due ingressi a 32 bit, e produce due uscite da 64 bit.

- **l'output di sinistra è semplicemente una copia dell'output di destra.**
- **L'output di destra consiste nello XOR bit per bit dell'input di sinistra, con una funzione dell'input di destra, e della chiave per questo stadio.**

Tutto ciò è strutturato in modo che la decifrazione, usi la stessa chiave della cifratura, proprietà necessaria in un algoritmo a chiave simmetrica.

TRIPLO DES:
Il triplo DES, ha un funzionamento molto semplice: è diviso in 3 fasi, nella quale la prima è un DES in criptazione con chiave K1, poi viene usato un DES in decriptazione, con chiave K2, e poi ancora in criptazione con K1. Si usano quindi 2 chiavi(da 112 bit), e 2 criptazioni e una decrittazione(EDE, encrypt, decrypt, encrypt), al posto che 3 criptazione (EEE, encrypt x3), per

una questione di retro-compatibilità con DES semplice: usando infatti due chiavi uguali (K1=K2), si torna a DES.

54. Si descriva il cipher block.

I block cipher (=cifrari a blocco), sono algoritmi a chiave simmetrica; per decriptare e criptare si usa quindi la stessa chiave.
I cifrari a blocco prendono n bit da testo in chiaro, e li trasformano utilizzando una chiave a n bit.
Sono detti "cifrari a blocchi" perché agiscono appunto su blocchi di bit (prendendo cioè n bit alla volta di testo in chiaro, e li traducono in n bit di messaggio criptato).
I Cipher Block si compongono essenzialmente con due elementi: i P-box, e gli S-box.

Questi algoritmi spesso utilizzano una trasposizione e una sostituzione per cifrare.
Per permutare, si usa una P-box (permutation box), che prende in input n bit e li permuta.
Per la sostituzione si usa invece una S-box, che compie la sostituzione.
Collegando questi due dispositivi in cascata, si ottiene il cosiddetto cifrario prodotto.
Possono essere realizzati sia via software, sia via hardware.
Uno dei primi esempi di cifrario a blocchi, è DES.

[Parlare ad esempio anche di DES...]

55. ECB.

[Sicurezza delle reti]
ECB=Eletronic Code Book.
E' una tecnica utilizzata per l'applicazione di algoritmi di crittografia simmetrica, che operano su blocchi di dati di lunghezza fissa B. Questo vuol dire che se si vuole cifrare un dato (testo in chiaro) di dimensione maggiore di B, occorre suddividerlo in blocchi di dimensione B ed eventualmente, riempire l'ultimo blocco, in modo tale da rendere anche l'ultimo blocco lungo B, potendo così applicare l'algoritmo di crittografia a blocchi (es DES, AES).
Un esempio di algoritmo che opera su dati di dimensione prefissata, è DES.

[Parlare ad esempio anche di DES...]

56. Stream cipher.

[Sicurezza delle reti]
Lo stream cipher, fa parte degli algoritmi di cifratura simmetrici.
Questa modalità funziona cifrando un vettore di inizializzazione con una chiave crittografica, per ottenere un blocco in uscita.
Quest'ultimo, viene cifrato per ottenere un secondo blocco in uscita, quindi si procede analogamente con il terzo, ecc.
La sequenza (di lunghezza arbitraria) di blocchi cifrati in uscita, chiamata il keystream (flusso delle chiavi), viene utilizzata come un blocco monouso (One-time-pad) e applicata in XOR sul testo in chiaro, per ottenere il testo cifrato.
Il vettore di inizializzazione, viene utilizzato solo per la prima iterazione, poi saranno i keystream ad essere cifrati.
Il keystream, è indipendente dai dati, così che può essere calcolato in anticipo, se necessario, ed è anche immune da errori di trasmissione.
Una cosa essenziale nella modalità stream cipher, è quella di non utilizzare mai la stessa coppia (chiave, vettore di inizializzazione).
L'uso ripetuto dello stesso keystream, espone il testo cifrato ad attacchi di tipo keystream riutilizzato.

Con la modalità stream cipher, 1 errore di 1 bit nel testo cifrato trasmesso, genera solamente un errore di 1 bit nel testo decifrato.

57. RSA

[Sicurezza delle reti]
RSA è un algoritmo di crittografia asimmetrica, basato sulla crittografia a chiave pubblica. Questo metodo, molto utilizzato, richiede però chiavi di almeno 1024 bit, il che lo rende abbastanza lento.

Come funziona una chiave pubblica:
Facendo un esempio pratico, se Alice vuole spedire un messaggio a Bob, e non vuole che altri all'infuori di Bob possano leggerlo, Alice cercherà sull'elenco la chiave pubblica di Bob, e con quella potrà cifrare il messaggio. Essendo Bob l'unico a possedere la chiave inversa, sarà anche l'unico a poter decifrare il messaggio, che rimarrà così segreto per tutti gli altri, Alice compresa, che non disponendo della chiave inversa non sarà in grado di decifrare il messaggio da lei stessa creato.
Firme a chiave pubblica:
Con questo metodo di cifratura, è inoltre possibile garantire anche la provenienza del messaggio: vediamo perché.
Nell'esempio precedente, Alice questa volta prima di cifrare il messaggio usando la chiave pubblica di Bob, lo cifrerà usando la propria chiave inversa, e solo in un secondo momento lo ri-crittograferà utilizzando la chiave pubblica di Bob.
Quando Bob riceverà il messaggio e lo decifrerà, usando la propria chiave inversa, otterrà ancora un messaggio criptato. Quel dato messaggio necessiterà poi della chiave pubblica di Alice per essere decifrato, garantendo così che il messaggio è stato spedito solo e soltanto da Alice, unica a possedere la chiave inversa con la quale era stato crittografato il messaggio.

RSA:
Per semplificare il funzionamento di RSA, immaginiamo che Alice debba spedire un messaggio segreto a Bob. Occorrono i seguenti passaggi:

1. **Bob sceglie due numeri primi, molto grandi (per esempio di 300 cifre), e li moltiplica con il suo computer.**
2. **Bob invia il numero che ha ottenuto ad Alice. Chiunque può vedere questo numero.**
3. **Alice usa questo numero per cifrare il messaggio.**
4. **Alice manda il messaggio cifrato a Bob, chiunque può vederlo, ma non decifrarlo.**
5. **Bob riceve il messaggio, e utilizzando i due fattori primi che solo lui conosceva, lo decifra.**

Alice e Bob hanno impiegato pochi secondi a cifrare e decifrare, ma chiunque avesse intercettato le loro comunicazioni impiegherebbe troppo tempo per scoprire i due fattori primi, con cui si può decifrare il messaggio.

In realtà, questo sistema non è così semplice, e per trasmettere grandi quantità di dati occorre tanto tempo per la decifratura, quindi Alice e Bob si scambieranno con questo sistema una chiave segreta (che non occupa molto spazio), che poi useranno per comunicare tra loro usando un sistema a crittografia simmetrica, più semplice e veloce.

58. Sicurezza in 802.11

Il protocollo di sicurezza prescritto dallo standard 802.11 è il WEP (Wired Equivalent Privacy).
Funziona usando encrypting a chiave simmetrica.
Utilizza un block cipher (cifrario a blocco), chiamato RC4.
La cifratura WEP usa uno stream cipher su RC4.

Quando in 802.11 la sicurezza viene attivata, ogni stazione stabilisce una chiave segreta condivisa (è a chiave simmetrica!) con la base.
Ad esempio, ciò può avvenire facendo generare alla stazione numeri casuali, che invierà con una trasmissione wireless cifrata usando la chiave pubblica del ricevente. La cifratura WEP utilizza uno stream cypher, basato sull'algoritmo RC4: viene generato un keystream che viene messo in XOR col testo in chiaro (plaintext), per produrre il testo cifrato (è il principio usato da stream cipher).

Incapsulamento dei pacchetti con WEP:
Per prima cosa viene calcolato il checksum del payload usando CRC-32 polinomiale.
Il checksum viene aggiunto al payload per formare il testo da cifrare.
Il testo viene messo in XOR con il keystream.
L'IV(Vettore di inizializzazione) utilizzato per inizializzare il keystream viene inviato insieme al testo cifrato.

I Problemi del WEP:
L'IV è di 24 bit. Ciò comporta che ce ne siano 2^24, e oltre 2^24 pacchetti, siamo costretti a riutilizzare lo stesso IV. Quindi, abbiamo la possibilità di fare attacchi del tipo keystream riutilizzato.
A questa debolezza, si aggiunge anche la vulnerabilità dell'algoritmo RC4.

59. Si descriva la sicurezza del bluetooth.

Bluetooth ha 3 modalità di sicurezza, che vanno dalla sicurezza 0 (nessuna sicurezza), ad una completa cifratura dei dati e controllo dell'integrità.
Bluetooth presenta soluzioni di sicurezza a più strati: in quello fisico viene usato il salto di frequenza (frequency hopping). La vera sicurezza si ha comunque quando uno slave comunica con il suo master. Si suppone che i dispositivi si siano scambiati le chiavi (condivise), a volte già inserita nei dispositivi. Questa chiave è detta passkey.
Per stabilire una connessione, si controlla che l'altro conosca la passkey. In caso affermativo, viene stabilito se il canale deve essere cifrato e viene scelta una chiave di sessione casuale a 128 bit.
La cifratura usa uno stream cypher detto E_0 , ed il controllo di integrità utilizza invece SAFER+. Entrambi, sono cifrari a blocco con chiave simmetrica.

Meccanismo di cifratura:
Il testo in chiaro (plaintext), viene applicato in XOR con il keystream per generare il testo cifrato.
Problemi:
E_0 Però potrebbe avere alcune vulnerabilità.
Un altro punto riguardante la sicurezza è dato dal fatto che bluetooth autentica solo i dispositivi, e non gli utenti: il furto di un dispositivo Bluetooth, può quindi dare al ladro l'accesso a qualunque dato dell'utente.
Bluetooth implementa però la sicurezza anche negli altri strati superiori, quindi nel caso in cui ci sia una violazione allo strato data link, rimane ancora un po' di sicurezza, specialmente per le applicazioni che richiedono l'inserimento manuale dalla tastiera di un codice PIN per poter completare la transazione.

60. Il replay attack.

Il replay attack è una forma di attacco di rete che consiste nell'impossessarsi di una credenziale di autenticazione comunicata da un host ad un altro, e riproporla successivamente simulando l'identità dell'emittente.
In genere, l'azione viene compiuta da un attaccante che s'interpone tra i due lati comunicanti. A differenza dell'attacco di tipo man in the middle, che opera sempre in tempo reale, il replay attack può operare anche in modo asincrono quando la comunicazione originale è terminata.
Si verifica un replay-attack quando Mallory intercettata la comunicazione tra Alice, che si sta autenticando con Bob, e si spaccia, agli occhi di Bob, per Alice.
Quando Bob chiede a Mallory (convinto di parlare con Alice) una chiave d'autenticazione, Mallory pronta invia quella di Alice, instaurando così la comunicazione.
Gli attacchi di tipo replay si evitano con l'uso di token di sessioni generati pseudocasualmente: Bob invia ad Alice uno di questi token usa e getta, che Alice utilizza per criptare la propria chiave da inviare a Bob. Bob effettua lo stesso calcolo e controlla che il suo risultato corrisponda con quello di Alice.
Questo attacco permette operazioni fraudolente come falsa autenticazione e/o transazioni duplicate, senza dover necessariamente decrittare la password, ma soltanto ritrasmettendola in un tempo successivo.

61. Man in the middle.

In crittografia, l'attacco “Man in the middle”, è un tipo di attacco nel quale l'attaccante è in grado di leggere, inserire o modificare a piacere, messaggi tra due parti senza che nessuna delle due sia in grado di sapere se il collegamento che li unisce sia stato effettivamente compromesso da una terza parte, ovvero un attaccante.
L'attaccante deve essere in grado di osservare, intercettare e replicare verso la destinazione prestabilita il transito dei messaggi tra le due vittime.
Supponiamo che Alice voglia comunicare con Bob e che Mallory voglia spiare la conversazione e, se possibile, consegnare a Bob dei falsi messaggi. Per iniziare, Alice deve chiedere a Bob la sua chiave pubblica. Se Bob invia la sua chiave pubblica ad Alice, ma Mallory è in grado di intercettarla, può iniziare un attacco Man in the middle.
Mallory può semplicemente inviare ad Alice una chiave pubblica della quale possiede la corrispondente chiave privata. Alice poi, credendo che questa sia la chiave pubblica di Bob, cifra i suoi messaggi con la chiave di Mallory ed invia i suoi messaggi cifrati a Bob. Mallory quindi li intercetta, li decifra, ne tiene una copia per sì e li re-cifra (dopo averli alterati se lo desidera), usando la chiave pubblica che Bob aveva originariamente inviato ad Alice.
Quando Bob riceverà il messaggio cifrato, crederà che questo provenga direttamente da Alice.
Un simile attacco è possibile, in teoria, verso qualunque messaggio inviato usando tecnologia a chiave pubblica, compresi pacchetti di dati trasportati su reti di computer.

Altro esempio:
Alice vuole visitare il sito Web di Bob. Per fare questo, digita l'URL di Bob nel browser e, dopo alcuni secondi, appare una pagina web. E' veramente la pagina web di bob?
Trudy, potrebbe intercettare ed esaminare tutti i pacchetti in uscita da Alice. Quando intercetta una richiesta http GET, indirizzata al sito di Bob, Trudy potrebbe andare a prendere la pagina di Bob, modificarla a piacimento e poi inviare ad Alice la versione alterata. Alice non si accorgerebbe di niente.
Un esempio, potrebbe essere dato dalla tecnica del DNS spoofing.

62. DNS spoofing.

DNS spoofing è un tipo di attacco “Man in the middle”.
Si svolge nel modo seguente:
Supponiamo che Trudy sia in grado di entrare nel sistema del DNS, magari anche solamente nella cache del DNS dell'IPS di Alice. Qui, Trudy rimpiazza l'indirizzo IP di Bob, con il proprio indirizzo IP.

1. **Alice chiede al DNS l'indirizzo IP di Bob.**
2. **Lo ottiene.**
3. **Chiede a Bob la sua home page.**
4. **La ottiene.**

Dopo che Trudy ha modificato il record di Bob nel DNS, in modo da contenere l'indirizzo di Trudy, quando Alice cerca l'indirizzo IP di Bob, ottiene invece quello di Trudy, quindi tutto il traffico che dovrebbe andare verso Bob va verso Trudy.

Come fa Trudy ad ingannare il DNS?

> **Riassumendo brevemente, Trudy inganna il DNS dell'ISP di Alice inviandogli una query per cercare l'ndirizzo di Bob. Sfortunatamente, visto che il DNS usa UDP, il server DNS non ha modo di sapere chi fornisce la risposta alla query. Trudy utilizza questa opportunità per falsificare la risposta alla query e quindi inserire un indirizzo IP falso nella cache del server DNS.**
> **Nel dettaglio:**
> **Trudy comincia inviando all'IPS di Alice la richiesta dell'indirizzo IP di bob.com.**
> **Visto che non ci sono informazioni per questo nome DNS, la richiesta viene passata al server top-level per il dominio com. Invece, quello che succede è che Trudy batte in velocità il server com, inviando una risposta falsa che dice: “bob.com è xxx.xxx.xxx.xxx”, dove l'indirizzo IP è quello di Trudy.**

Una cache che contiene un indirizzo IP intenzionalmente falsificato, è detto poisoned cache.

Nel caso in cui l'ISP di Alice controlli che la risposta contenga come indirizzo IP sorgente quello del server top-level, Trudy può aggirare questo ostacolo semplicemente cambiando il campo sorgente dell'indirizzo dell'IP, mettendo quello del server top-level (che sono pubblici).

Un'ulteriore difficoltà consiste nel fatto che tutte le richieste contengano un numero di sequenza, per consentire ai server DNS di far corrispondere le richieste con le risposte.
Per poter aggirare il problema, Trudy deve conoscere il numero di sequenza corretto. Può farlo ad esempio registrando un proprio dominio e facendo conoscere il suo DNS server all'ISP di Alice.