

Telecomunicazioni Riassunti/Appunti

INTRODUZIONE CAPITOLO 1

Infrastruttura:

Host – sistema terminale che fa girare programmi applicativi (web, e-mail) che si interconnette alla rete attraverso dei **Collegamenti (link)** come ad esempio: Cavo coassiale/fibra ottica/onde elettromagnetiche.

Router – Si occupa dell’instradamento dei dati, tra i vari collegamenti vi sono dei dispositivi di interconnessione che creano i nodi di rete

Frequenza di Trasmissione connessa all’ampiezza di banda = ovvero quanti bit/s posso trasferire.

Obiettivo – della connessione è di trasferire informazioni e comunicare a distanza

Protocollo – definisce il formato, l’ordine e le regole dei messaggi scambiati tra due o più entità in comunicazione (es. TCP, IP, http etc.), si tratta di dispositivi sia hardware che software

Struttura

Confini – parte esterna ci sono le reti domestiche o aziendali fatte da sistemi terminali e applicazioni. Essi comunicano con due tipi di architetture:

- **Client/Server** – L’host client richiede e riceve un servizio da un programma offerto dal server in esecuzione su un altro terminale con struttura gerarchizzata (Browser, Web)

- **Peer to Peer** – nodi non gerarchizzati in maniera fissa ma chi è client può diventare server e viceversa (Skype, Bit Torrent), uso limitato o inesistente dunque di server dedicati

I sistemi P2P creano reti overlay di tipo dinamico: Host che sono online in un istante si connettono direttamente uno all’altro per permettere la condivisione delle proprie risorse

Intermezzo – Dispositivi fisici che permettono la comunicazione (Collegamenti)

Centro – i router interconnessi che creano poi internet etc.

Reti di Accesso – si indica la parte di rete destinata al collegamento fra la sede dei singoli utenti finali (terminali) e fino alla parte centrale (provider). Si distinguono per risorse:

- **Dedicate** – dedicate a una singola comunicazione

- **Condivise** – dedicate a molteplici comunicazioni e sono di tre tipologie:

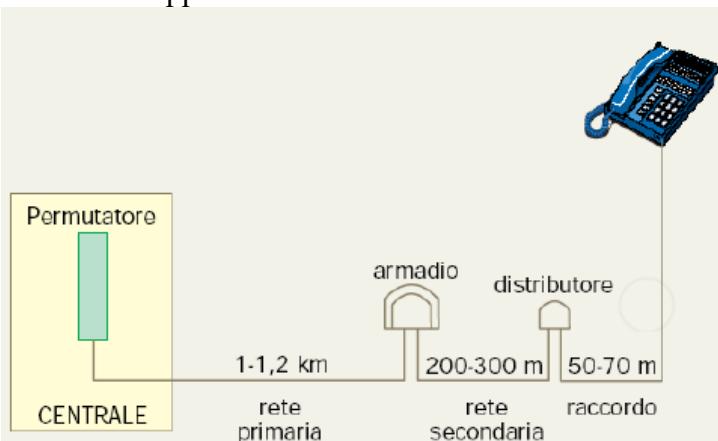
Reti di accesso residenziale – (punto a punto) due tipologie:

- **Modem Dial-Up:** si richiede la connessione al modem e realizza la connessione tra computer usando la banda fornita a bassa frequenza. Non è possibile navigare e telefonare allo stesso momento. (fino a 56 Kbps)

- **DSL** - linea dedicata per la connessione = connessione permanente, (30 Mbps in upstream, 100 Mbps in downstream)

- **Gerarchia:** Dial-Up / ADSL free (256-640 kbps) / ADSL Flat (2-6 Mbps) / ADSL2+ (10-20 Mbps) / VDSL2 (50-100 Mbps)

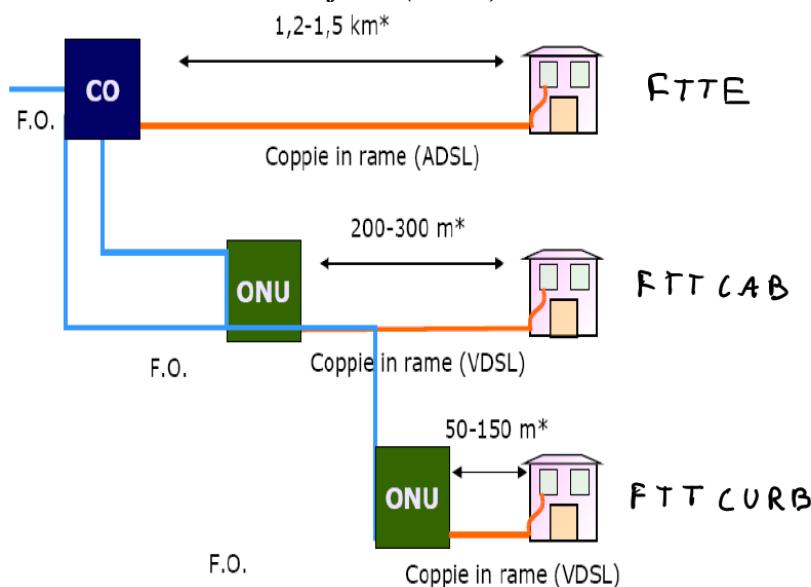
Rete di Distribuzione Telefonica – Ha come obiettivo il trasporto e il trattamento del segnale dalla centrale all'apparecchio del cliente.



Parte dal **permutatore** (centrale) passa per la rete primaria (~ 1 km) per arrivare all'**armadio**, poi si dirige attraverso la rete secondaria (~ 200 m) per arrivare al **distributore**, da cui attraverso il raccordo (~ 50 m) arriva alle case.

Reti di accesso aziendale – (locali) chiamate LAN collegano i sistemi terminali di aziende e università ad una LOCAL AREA NETWORK ovvero ad un router. A seconda dell'estensione geografica della rete si potrebbe arrivare ad ottenere una WAN (WIDE AREA NETWORK) che collega reti con estensione geografica maggiore.

Architetture ibride rame-fibra (FTTx)



ONU(Optical Network Unit): converte il segnale ottico trasmesso attraverso la fibra in segnale elettrico.

FTTE = Fiber to the Exchange

FTTCab = L'ONU è situato nell'armadio telefonico nelle strade, Fiber to the Cabinet

FTTCurb = L'ONU è situato in prossimità degli edifici dove si trovano gli utenti, Fiber to the Curb

FTTB = Nelle aree con edifici a sviluppo verticale, Fiber to the Building

FTTH = Anche nel caso di case individuali, Fiber to the Home (Onu presente nell'edificio)

Mezzi Trammissivi

Mezzo fisico – fisicamente ciò che sta tra trasmittente e ricevente

Mezzo guidato – il mezzo fisico tramite il quale il segnale si propaga (fibra ottica, filo di rame, cavo coassiale etc.)

Mezzi a onda libera – i segnali si propagano nell'atmosfera o spazio

ESEMPI

Twisted Pair – due fili di rame attorcigliati che trasferiscono il segnale:

- **Categoria 3**: tradizionale cavo telefonico (10 Mbps Ethernet)
- **Categoria 5**: 100 Mbps Ethernet

Cavo coassiale – due conduttori in rame concentrici, bidirezionale

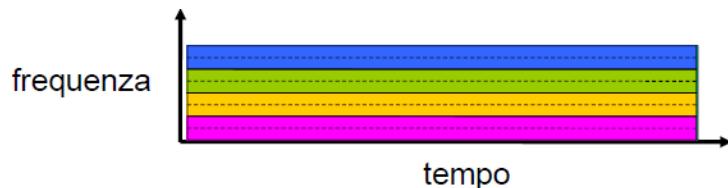
Fibra ottica – Mezzo sottile e flessibile che conduce impulsi di luce, alta frequenza trasmissiva e basso tasso di errore, immune all'interferenza elettromagnetica.

Canali Radio – in un canale radio non tutte le frequenze sono uguali e a seconda della frequenza la trasmissione può essere più veloce o più lenta ovvero più o meno forte nell'attraversare le cose. Microonde, Satellitare, Wifi etc.

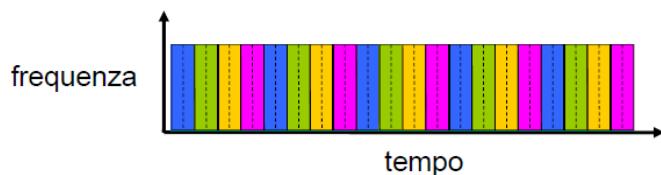
Modi di trasferimento dati

Commutazione di Circuito – (Circuit Switching) è un circuito dedicato per l'intera sessione che garantisce la prestazione della comunicazione ma non permette più connessioni allo stesso tempo e necessita dell'impostazione iniziale della chiamata. Per sorvolare il problema della mancata condivisione si possono suddividere le risorse di rete (**banda**) in pezzi(**canali**) e ogni pezzo viene allocato ad un collegamento. Questa divisione si può fare in due modi:

Divisione per FREQUENZA (Frequency Division Multiplex) FDM: a ogni collegamento affido una frequenza



Divisione per TEMPO (Time Division Multiplex) TDM: a ogni collegamento affido uno slot di tempo



Commutazione di Pacchetto – (Packet Switching) il flusso dei dati viene suddiviso in pacchetti di bit. Non c'è una linea dedicata e le risorse vengono utilizzate a seconda della necessità (**Multiplazione Statistica**). Utilizzo delle risorse più dinamico ma potrebbe incorrere in congestione di pacchetti = ritardi. Questo trasferimento inoltre utilizza il metodo **Store and Forward**: il commutatore deve ricevere l'intero pacchetto prima di poterlo trasmettere sul collegamento in uscita.

Ottima per dati a “burst” poiché più semplice della CS e non necessita dell'instaurazione della chiamata, ma sono necessari protocolli per il trasferimento affidabile dei dati e per il controllo della congestione.

Multiplazione Statistica: non vi è una sequenza fissa di chi utilizza risorse (condivisione di risorse su richiesta), l'ordine è casuale a seconda della necessità. = maggiore utilizzo efficiente della banda.
Se il tasso di arrivo dei pacchetti eccede la capacità del collegamento, il nodo memorizza i pacchetti nei Buffer dei router, ma se lo spazio non bastasse il pacchetto potrebbe essere perso. = **BEST EFFORT**

Ritardi: i pacchetti si accodano nel buffer dei router, se il tasso di arrivo è maggiore della capacità di inoltro del collegamento si crea una coda e si creano dei ritardi:

Ritardo di elaborazione - ritardo nel controllo di errori e nell'instradamento

Ritardo di accodamento – ritardo di trasmissione (può essere solo stimato), livello di congestione

Ritardo = $\frac{aL}{R}$ dove a = tasso medio di arrivo pacchetti, tanto più si avvicina ad 1 tanto più grande è il ritardo di attesa

Ritardo di trasmissione – dipende dalla velocità della linea

Ritardo = $\frac{L}{R}$ dove R = velocità/frequenza di trasmissione in bps e L = lunghezza del pacchetto in bit

Ritardo di propagazione – dipende dal tempo dei pacchetti

Ritardo = $\frac{d}{s}$ dove d = lunghezza collegamento fisico, s = velocità di propagazione del collegamento (circa 2×10^8 m/sec)

Ritardo di link

$$d_{\text{link}} = d_{\text{elab}} + d_{\text{queue}} + d_{\text{trasm}} + d_{\text{prop}}$$

d_{elab} = ritardo di elaborazione (**processing delay**): In genere pochi microsecondi, o anche meno

d_{queue} = ritardo di accodamento (**queuing delay**): Dipende dalla congestione

d_{trasm} = ritardo di trasmissione (**transmission delay**): Significativo sui collegamenti a bassa velocità

d_{prop} = ritardo di propagazione (**propagation delay**): Da pochi microsecondi a centinaia di millisecondi

Perdita di pacchetti

Una coda ha capacità finita: Quando il pacchetto trova la coda piena, viene scartato (e quindi va perso)
Un pacchetto perso può essere: ritrasmesso dal nodo precedente, dal sistema terminale che lo ha generato, o non essere ritrasmesso affatto

Altri valori

Throughput – Frequenza alla quale i bit sono trasferiti tra mittente e ricevente in bit/s.

Esso può essere calcolato in un preciso istante (**Istantaneo**) o in un periodo (**Medio**). Ma se il mio collegamento è fatto di diversi collegamenti con frequenze diverse alla fine quello con portata minore determinerà la portata complessiva (**collo di bottiglia anche detto Bottleneck**).

Traceroute - Programma diagnostico che fornisce una misura del ritardo dalla sorgente al router lungo i percorsi Internet punto-punto verso la destinazione.

Esempio: Invio tre pacchetti che raggiungono il router i sul percorso verso la destinazione, una volta ricevuti il router li restituirà al mittente e il mittente calcolerà così l'intervallo tra trasmissione e risposta.

Administration Maintenance e Billing: I flussi di traffico devono essere monitorati e controllati, mentre le attuali funzioni esistenti devono essere evolute con i cambiamenti

Tecnologie di Elaborazione

Legge di Moore: raddoppio della densità di transistor per circuito integrato ogni due anni

RAM: tabelle più grandi, sistemi più grandi

Digital signal processing: trasmissione, multiplazione, framing, error control, crittografia

Network processors: hardware dedicato per routing, switching, forwarding e traffic management

Microprocessors: supporto di sofisticate applicazioni e protocolli applicativi con maggiore velocità e throughput

Mercato

Network effect: il vantaggio di un servizio aumenta con la dimensione della comunità che lo utilizza

- **Metcalfe's Law:** il vantaggio di un servizio è proporzionale al quadrato del numero di utenti

Economia di scala: il costo per utente diminuisce all'aumentare del volume di produzione (numero di dispositivi su cui può girare il software)

S-curve: l'evoluzione di un nuovo servizio è rappresentata da una "S-shaped curve", il punto è raggiungere una crescita verso l'alto

Standards: Le nuove tecnologie sono spesso costose e rischiose, gli **Standard** permettono agli attori in gioco di condividere il rischio e gli eventuali benefici di un nuovo mercato

Internet

Rete delle reti con struttura gerarchica e standardizzata:

Esistono provider ISP di tre livelli diversi: Gli **ISP di livello 1** sono direttamente connessi a ciascuno degli altri ISP di livello 1, gli **ISP di livello 2**: ISP più piccoli (nazionali o distrettuali) si possono connettere solo ad alcuni ISP di livello 1 e possibilmente ad altri ISP di livello 2, gli **ISP di livello 3** o ISP locali sono gli ISP finali a cui si collega l'utente.

Internet Engineering Task Force – coloro che sviluppano gli standard e li pubblicano poi in

RFC = Request for Commento – documento di testo con regole con tempo di vita, se viene commentato ed eseguito allora viene approvato.

Medium access control (MAC) regola l'accesso ai mezzi condivisi.

Indirizzi identificano il punto di accesso alla rete (interfaccia)

ARCHITETTURA A STRATI CAPITOLO 2

La rete è un sistema complesso e per organizzare il sistema si utilizza una stratificazione gerarchica delle funzionalità. In questa maniera ogni processo della comunicazione risulta indipendente dagli altri, lo si può aggiornare o testare individualmente (manutenzione). I vari strati vengono poi collegati dai protocolli che effettuano delle chiamate ai servizi offerti dallo strato inferiore.

Architetture monolitiche sono costose, scarsamente flessibili e sono soggette a rapida obsolescenza

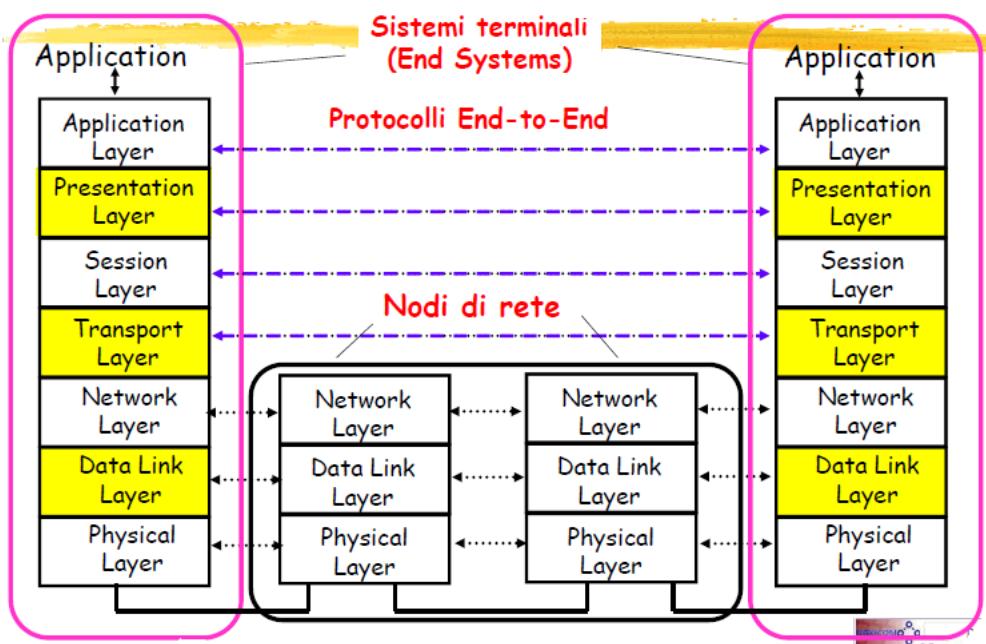
Il modello di riferimento per quest'architettura è:

OSI = Open Systems Interconnection: che definisce un modello di rete a 7 strati e i protocolli di ogni strato per permettere l'interconnessione tra sistemi che utilizzano anche architetture diverse.

Problema dell'OSI: Dagli anni '70 ogni produttore ha sviluppato la propria architettura a strati, dunque computers di "vendor" diversi non potevano essere interconnessi

Oggi questo sistema è stato superato dal TCP/IP che utilizza però gli stessi concetti.

Due terminali comunicano attraversando i loro 7 strati di funzioni. Alcune funzioni avvengono solo (end to end) nei terminali meno onerose ma meno immediate, altre vengono ripetute anche nei nodi (sezione per sezione) che vengono inseriti per motivi di distanze e topologia della rete.



STRATO FISICO (BIT) Physical Layer

Strato che effettua operazioni sezione per sezione, ma viene implementato in maniera diversa a seconda del mezzo trasmissivo.

Scopo – trasferire bit informativi sui mezzi trasmissivi

Caratteristiche – Meccaniche (tipo cavo), Elettriche / ottiche (potenza segnali, tensione) , Funzionali (procedure per attivare, mantenere o disattivare link fisici)

STRATO DI COLLEGAMENTO (FRAME) Data Link Layer

Scopo – realizzare trasferimento affidabile

Operazioni:

- Riceve bit dallo strato fisico e li incapsula in unità denominate **frame** (**Frame**)
- Controlla che i bit ricevuti siano corretti ed eventuale loro correzione
- Controlla il flusso, regolarizza la velocità alla quale i frame viaggiano
- Controlla la condivisione delle risorse tramite il MAC

STRATO DI RETE (PACCHETTI) Network Layer

Scopo – trasferire pacchetti attraverso una serie di reti diverse/link (**Internetworking**)

Operazioni:

Indirizzamento di rete (MultiCast, BroadCast , Unique Terminal)

Definisce procedure di **intradamento (routing)** per determinare nei nodi i cammini di rete

Definisce le procedure di **rilancio (forwarding)** dei pacchetti nei nodi

Controllo congestione – evita sovraccarico sulla rete

Definisce le procedure di **setup, gestione e teardown** delle connessioni di rete

STRATO DI TRASPORTO (SEGMENTI) Transport Layer

Scopo – Trasferire i dati end-to-end dal processo attivo in un host al processo dell'host remoto

Operazioni:

Garantisce l'affidabilità del trasferimento di stream di dati

Offre un trasferimento rapido e semplice di singoli blocchi di dati

Gestisce i numeri delle porte (indirizzi interni ai sistemi terminali)

Segmenta i pacchetti nell'invio e li riassembla all'arrivo

STRATO DI APPLICAZIONE E SUPERIORI (MESSAGGIO) Application Layer

Scopo – fornire servizi richiesti dalle applicazioni (DNS, web access, file transfer, email...)

Sono incorporati in esso due funzioni:

Presentation Layer – interpretare significato dei dati (cifratura, compressione, convenzioni specifiche della macchina)

Session Layer – sincronizzare e controllare il dialogo + recupero dati

INTERAZIONE TRA STRATI

Lo strato è un'entità che implementa delle funzioni e che comunica attraverso le unità dati **protocol data unit (PDU)** che rappresenta la SDU + informazioni di controllo (PCI) per eseguire le funzioni dello strato seguendo delle regole condivise dai sistemi ai quali è interconnesso.

Le entità che eseguono le funzioni di uno strato all'interno di sistemi comunicanti sono dette:

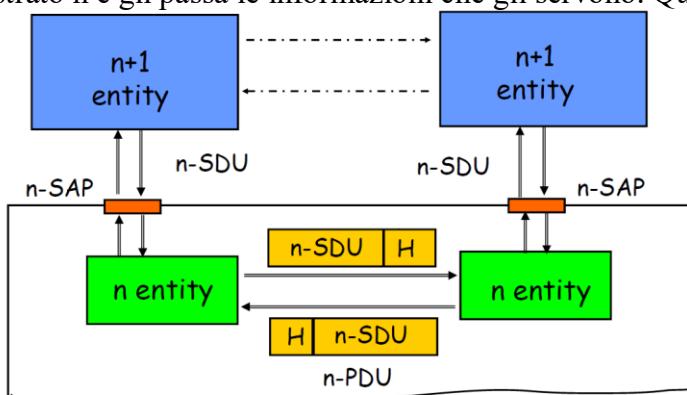
peer processes.

La cooperazione tra entità dello stesso strato è regolata dal protocollo di strato n (**layer-n protocol**)

Se esiste uno strato in un terminale deve esistere anche nel terminale con cui comunica.

I dati che vengono ricevuti da uno strato dallo strato superiore sono denominate **Service Data Unit (SDU)**, le SDU sono incapsulate nelle PDU nelle quali sono anche aggiunte le informazioni di controllo per l'esecuzione delle funzioni di strato.

La comunicazione tra strati è virtuale ed indiretta, infatti lo strato n +1 invoca il servizio fornito dallo strato n e gli passa le informazioni che gli servono. Questo meccanismo avviene tramite l'interfaccia.



Segmentazione & Riassemblaggio

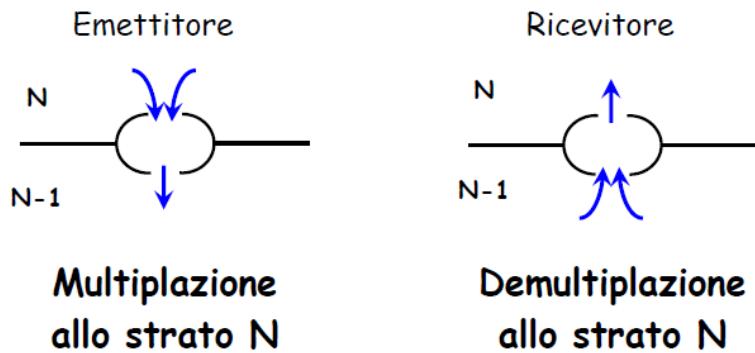
Uno strato può imporre un limite massimo alla dimensione del blocco dati che può essere trasferito, se le n-SDU superano questo limite non possono essere trasferite in un'unica n-PDU

Lato emittente: la SDU viene segmentata in PDU multiple

Lato ricevente: la SDU viene riassemblata a partire dalla sequenza di PDU ricevute

Alla ricezione dei bit nel lato ricevente ogni strato toglie la porzione di informazione relativa al suo strato e lo rimanda in su

Multiplexing: Condivisione del servizio di strato n da utenti multipli (è necessaria l'utilizzazione di etichette in ogni PDU per identificare a quale utente appartiene la SDU)



Headers & Trailers

PCI (Protocol control information). Ogni protocollo usa **un'intestazione (header)** per ogni strato e un **suffisso (trailer)** sul controllo dei bit che contengono le informazioni di controllo necessarie per l'esecuzione delle funzioni di strato: Indirizzi, numeri di sequenza, flag, codici di controllo d'errore...

Service Access Point

Interazione può avvenire in due modi:

Servizio con Connessione (Connection Oriented): due entità stabiliscono la connessione e le regole per il trasferimento e ad ogni trasferimento si attende la conferma di ricezione ed infine dichiarano terminata la connessione.

Strutturazione in tre fasi temporali:

- 1) Negoziazione dei parametri di trasferimento
- 2) Indirizzamento con identificatori di connessione
- 3) Legame logico tra i segmenti informativi scambiati

Servizio senza Connessione: non si stabilisce la connessione iniziale, assenza di negoziazione con uso di indirizzi esplicativi per l'origine e la destinazione, c'è una sola fase temporale e non c'è conferma di ricezione. (**Connectionless**)

STRATO DI APPLICAZIONE CAPITOLO 3

Fornisce i concetti base dei protocolli delle applicazioni di rete: modelli di servizio, paradigma client-server, paradigma peer-to-peer.

Protocolli di applicazione più diffusi:

HTTP (Hypertext Transfer Protocol) per i servizi Web

Una pagina web è costituita da un file base HTML che include diversi oggetti referenziati da un **URL** (Uniform Resource Locator) che contiene il nome del server e il percorso dell'oggetto

Nelle richieste HTTP viene usato il protocollo TCP:

- 1) Il client inizializza la connessione TCP (crea una socket) con il server (porta 80)
- 2) Il server accetta la connessione TCP dal client
- 3) Scambio di messaggi HTTP fra browser (client HTTP) e server web (server HTTP)
- 4) Chiusura della connessione TCP

HTTP è un protocollo **stateless**: Il server non mantiene informazioni sulle richieste fatte dal client, infatti due richieste consecutive dello stesso oggetto danno luogo a trasmissioni distinte

I protocolli che mantengono lo “stato” sono complessi (e lenti)

Se il server e/o il client si bloccassero, le loro viste dello “stato” potrebbero essere contrastanti e dovrebbero essere riconciliate

Connessioni HTTP

Connessioni non persistenti: Gli oggetti sono trasmessi su connessione TCP distinte tra client e server

- **Svantaggi:** Richiedono un tempo pari a 2 RTT per ogni oggetto, overhead del sistema operativo per ogni connessione TCP

Connessioni persistenti: Più oggetti possono essere trasmessi su una singola connessione TCP tra client e server (connessione TCP condivisa) dunque un solo RTT per tutti gli oggetti referenziati
La connessione TCP rimane attiva dopo il termine dell'invio di un oggetto fino allo scadere di un **timeout**

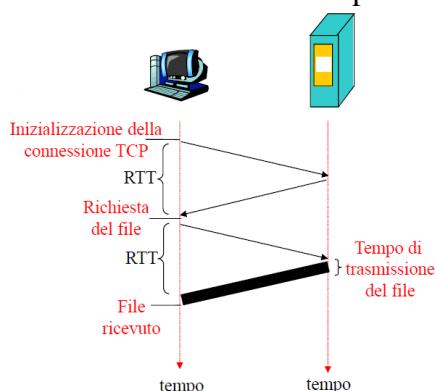
Schema del tempo di risposta

Round Trip Time (RTT): Tempo impiegato da un pacchetto per andare dal client al server e ritornare.

Tempo di risposta:

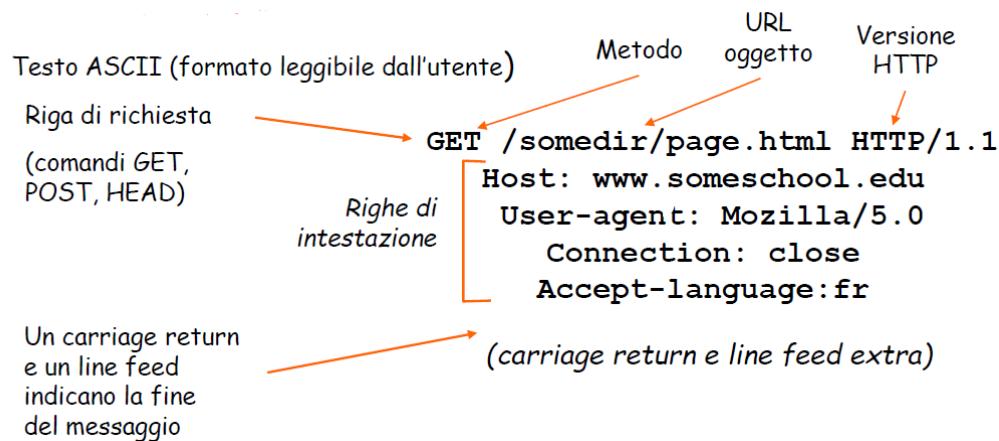
- 1) un RTT per inizializzare la connessione TCP
- 2) un RTT perché ritornino la richiesta HTTP e i primi byte della risposta HTTP
- 3) tempo di trasmissione del file

Ritardo totale: 2 RTT + tempo di trasmissione del file



Messaggi HTTP

Richiesta:

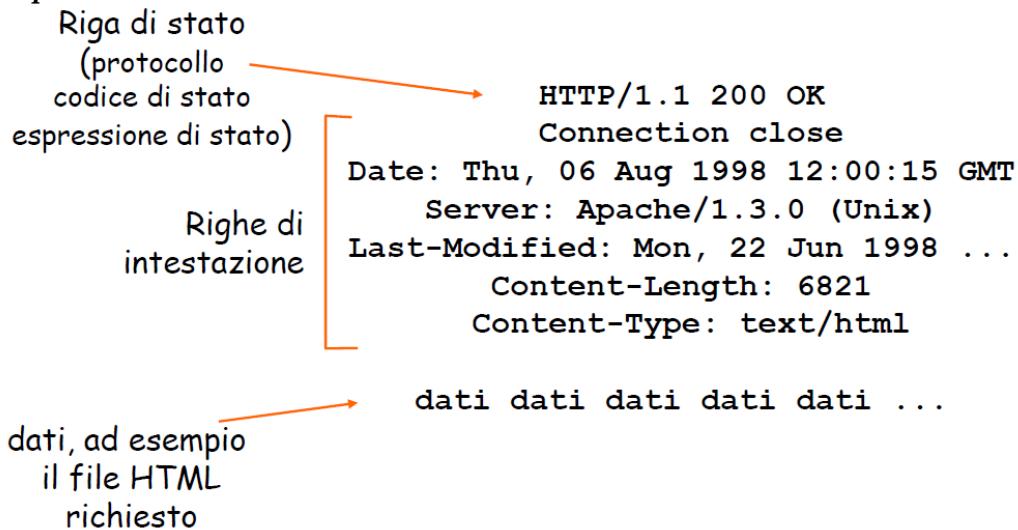


Tipi di metodi

HTTP/1.0: GET, POST, HEAD (chiede al server di escludere l'oggetto richiesto dalla risposta, usato per diagnostica)

HTTP/1.1: GET, POST, HEAD, PUT (include il file nel corpo dell'entità e lo invia al percorso specificato nel campo URL), DELETE (cancella il file specificato nel campo URL)

Risposta:



Codici della risposta HTTP

200 (OK): La richiesta ha avuto successo; l'oggetto richiesto viene inviato nella risposta

301 (Moved Permanently): L'oggetto richiesto è stato trasferito; la nuova posizione è specificata nell'intestazione Location della risposta

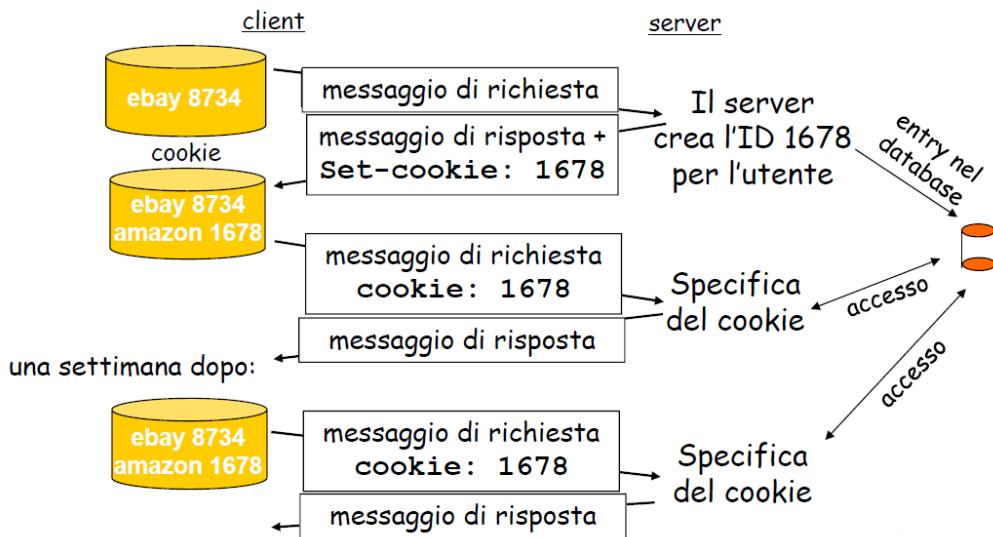
400 (Bad Request): Il messaggio di richiesta non è stato compreso dal server

404 (Not Found): Il documento richiesto non si trova su questo server

505 (HTTP Version Not Supported): Il server non supporta la versione richiesta del protocollo HTTP

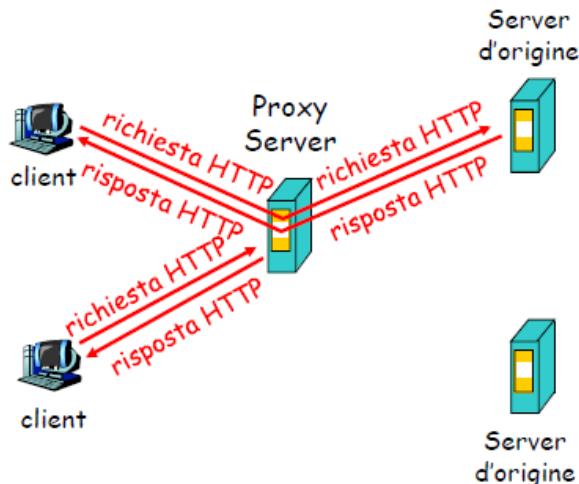
Interazione utente-server: i cookie

I cookie sono dei file mantenuti sul sistema terminale dell'utente e vengono gestiti dai browser, sono file identificativi che possono contenere dati di autorizzazione, carte per acquisti, stato della sessione dell'utente (mantengono lo stato del mittente e del ricevente per più transazioni, esempio e-mail)



Web cache (proxy server)

L'obiettivo è quello di soddisfare la richiesta del client senza coinvolgere il server originale, possiamo impostare un proxy server intermedio che crea una cache del sito originale (già in memoria nel proxy server oppure richiedendolo direttamente al server)



La cache dunque può operare sia come client che come server.

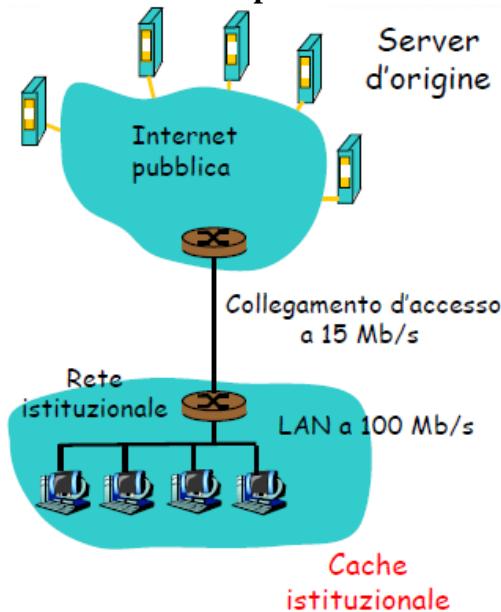
Il caching web viene usato poiché riduce i tempi di risposta alle richieste dei client e riduce il traffico sul collegamento di accesso a internet. Un altro pro (che è un contro per l'utente) è il fatto che internet viene arricchita di cache che consentono ai provider "scadenti" di fornire dati con efficacia

Esempio di caching

Ipotesi: Dimensione media di un oggetto = 1 Mbit, Frequenza media di richieste dai browser istituzionali ai server d'origine = 15 richieste/sec

Ritardo dal router istituzionale a qualsiasi server d'origine e ritorno al router = 2 sec (ritardo di Internet)

Ritardo totale di risposta = Ritardo LAN + Ritardo accesso + Ritardo internet



Altro Esempio:

Intensità del traffico nella rete locale (A_{LAN}):

$$A_{LAN} = 15 \text{ r/s} \times 1 \text{ Mbit} = 15 \text{ Mbit/s}$$

Grado di utilizzazione della LAN (ρ_{LAN}):

$$\rho_{LAN} = \frac{A_{LAN}}{100 \text{ Mbit/s}} = 0.15 \quad \text{Ritardo limitato}$$

Intensità del traffico nella rete di accesso (A_{acc}):

$$A_{acc} = 15 \text{ r/s} \times 1 \text{ Mbit} = 15 \text{ Mbit/s}$$

Grado di utilizzazione della rete di accesso (ρ_{acc}):

$$\rho_{acc} = \frac{A_{LAN}}{15 \text{ Mbit/s}} = 1 \quad \text{Ritardo molto elevato}$$

Conseguenze

ritardo totale = 2 sec + x minuti + y millisecondi

Il collo di bottiglia (**bottleneck**) è il segmento di accesso

Soluzione possibile

Aumentare l'ampiezza di banda del collegamento d'accesso a 100 Mbps per esempio

Conseguenze:

utilizzo sulla LAN = 15%

utilizzo sul collegamento d'accesso = 15%

ritardo totale = ritardo di Internet + ritardo di accesso + ritardo della LAN = 2 sec + msec + msec

Purtroppo, l'aggiornamento della rete di accesso è spesso molto costoso

Soluzione possibile: installare una cache

supponiamo una percentuale di successo (hit rate) pari a 0,4

Conseguenze:

il 40% delle richieste sarà soddisfatto quasi immediatamente

il 60% delle richieste sarà soddisfatto dal server d'origine

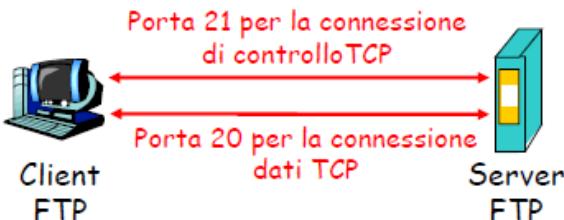
l'utilizzazione del collegamento d'accesso si è ridotta al 60%, determinando ritardi trascurabili (circa 10 msec)

ritardo totale medio = ritardo di Internet + ritardo di accesso + ritardo della LAN =

= $0,6 * (2,01)$ sec + millisecondi < 1,4 sec

FTP

Utilizzato per il trasferimento di file a/da un host remoto attraverso un modello client/server



- 1) Il client FTP contatta il server FTP alla porta 21, specificando TCP come protocollo di trasporto
- 2) Il client ottiene l'autorizzazione sulla connessione di controllo
- 3) Il client cambia la directory remota inviando i comandi sulla connessione di controllo
- 4) Quando il server riceve un comando per trasferire un file, apre una connessione dati TCP con il client
- 5) Dopo il trasferimento di un file, il server chiude la connessione
- 6) Il server apre una seconda connessione dati TCP per trasferire un altro file.
- 7) Connessione di controllo: "fuori banda" (out of band)
- 8) Il server FTP mantiene lo "stato": directory corrente, autenticazione precedente

Comandi e risposte FTP

I comandi comuni sono inviati come testo ASCII sulla connessione di controllo e possono essere:

USER: username

PASS: password

LIST: elenca i file della directory corrente

RETR filename: recupera(get) un file dalla directory corrente

STOR filename: memorizza(put) un file nell'host remoto

Mentre i codici di ritorno più comuni sono codici di stato ed espressione come in HTTP:

331: Username OK, richiesta la password

125: dati di connessione già aperti, inizio del trasferimento

425: Impossibile aprire i dati di connessione

452: Errore nella scrittura del file

SMTP/POP3/IMAP

La posta elettronica si suddivide in tre componenti principali:

Agente utente: detto anche “mail reader”, strumento attraverso la quale un utente può comporre, editare o leggere la posta, ad esempio Outlook

Server di posta suddiviso in:

- **Cassetta di posta(mailbox):** contiene i messaggi di arrivo per l'utente
- **Coda di messaggi** da trasmettere
- Cassetta di posta dell'utente

Il protocollo utilizzato è l'**SMTP** che usa TCP per trasferire in modo affidabile e **diretto** i messaggi dal client al server (porta 25) attraverso connessioni di tipo **persistente**. I messaggi nell'interazione comando (testo ASCII) / risposta (codice di stato ed espressione) devono essere nel formato ASCII a 7 bit e la fine del messaggio deve essere determinata da CRLF.CRLF

Il trasferimento si divide in tre fasi:

- 1) Handshaking
- 2) Trasferimento di messaggi
- 3) Chiusura connessione TCP

Confronto SMTP – HTTP

HTTP (pull): gli utenti accedono al server per effettuare il download dei contenuti

SMTP (push): gli utenti collocano i messaggi sul server

Entrambi hanno un'interazione comando/risposta, codici di stato

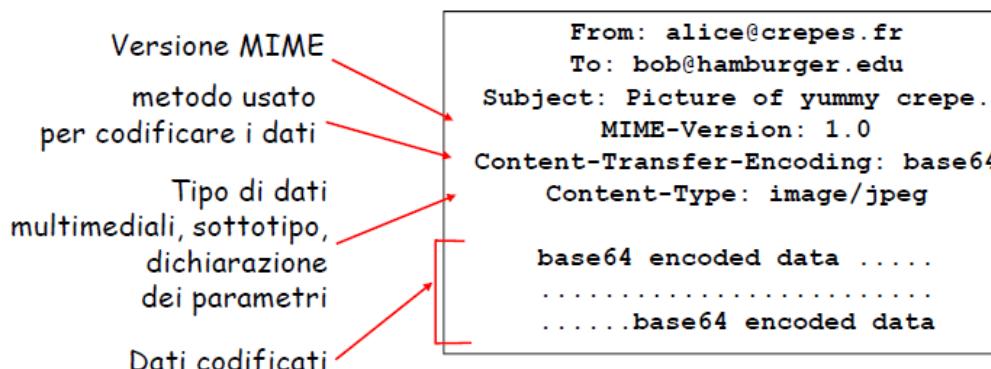
HTTP: ciascun oggetto è incapsulato nel suo messaggio di risposta

SMTP: più oggetti vengono trasmessi in un unico messaggio

Formato messaggi di posta elettronica



Un altro formato per i messaggi può essere il MIME che è un'estensione di messaggi di posta multimediali con RFC 2045,2056



Protocolli di accesso alla posta

Per inviare/memorizzare sul server del destinatario usiamo il protocollo SMTP, mentre per accedere e ottenere i messaggi dal server possiamo usare:

- **POP: Post Office Protocol** [RFC 1939]: composto solo da autorizzazione tra agente e server e successivo download
- **IMAP: Internet Mail Access Protocol** [RFC 3501]: più funzioni (più complesse) che consentono anche la manipolazione di messaggi memorizzati sul server
- **HTTP**: gmail, Hotmail, Yahoo! Mail, ecc.

Protocollo POP3

Fase di autorizzazione: Il client può dichiarare il proprio nome utente (comando **user**) e password (comando **pass**) mentre il server risponderà con +OK o -ERR

Fase di transazione: comandi client:

- **list**: elenca i numeri dei messaggi
- **retr**: ottiene i messaggi in base al numero
- **dele**: cancella il messaggio
- **quit**

The sequence diagram illustrates a POP3 session between a client and a server. It is divided into two main phases: 'Fase di autorizzazione' (Authorization phase) and 'Fase di transazione' (Transaction phase).
Authorization phase:
S: +OK POP3 server ready
C: user rob
S: +OK
C: pass hungry
S: +OK user successfully logged
Transaction phase:
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: retr 2
S: <message 1 contents>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off

Il protocollo pop3 è senza stato tra le varie sessioni.

Il precedente esempio usa la modalità **scarica e cancella** (l'utente non può rileggere le e-mail se cambia client) che differenzia dalla modalità **scarica e mantieni** (copia i messaggi su più client)

Il protocollo IMAP invece mantiene tutti i messaggi in un unico posto, il server e consente all'utente di organizzare i messaggi in cartelle conservando lo stato dell'utente tra le varie sessioni.

DNS: Domain Name System

È un protocollo che consente agli host, ai router e ai server DNS di comunicare per risolvere i nomi.

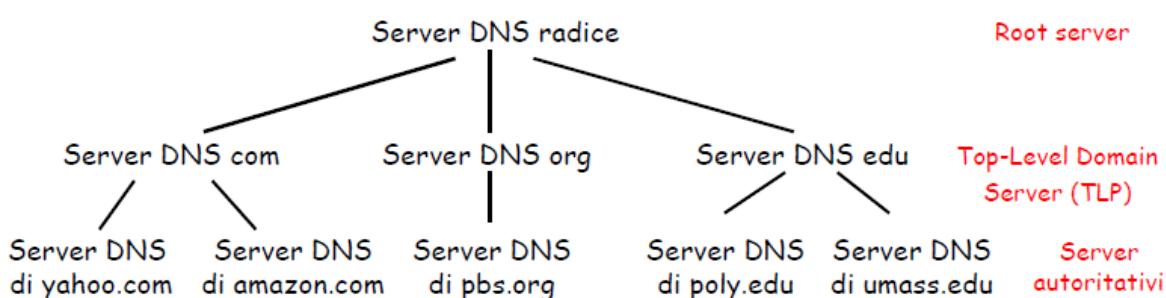
Host aliasing: un host può avere più nomi e il DNS traduce un alias nel nome canonico (indirizzo principale)

Distribuzione locale: un url può corrispondere a diversi IP

Perché non centralizzare DNS?

Singolo punto di guasto, volume di traffico, database centralizzato distante, manutenzione, un database centralizzato su un singolo server DNS non è scalabile

Database distribuiti e gerarchici



Il client vuole ottenere l'indirizzo IP del nome www.amazon.com:

- 1) Il client interroga il root server per trovare il server DNS .com
- 2) Il client interroga il server DNS .com per ottenere il server DNS amazon.com
- 3) Il client interroga il server DNS amazon.com per ottenere l'indirizzo IP di www.amazon.com

DNS: root server

Sono contattati da un server DNS locale che non può tradurre il nome, se non conosce la mappatura contatta un server DNS autorizzato per poterla ottenere e la restituisce al server DNS locale

Server TLD e server autoritativi

Server TLD (Top-Level Domain server): si occupano dei domini com, org, net, edu, ecc. e di tutti i domini locali di alto livello, quali uk, fr, ca e jp.

- Network Solutions gestisce i server TLD per il dominio com
- Educause gestisce quelli per il dominio edu

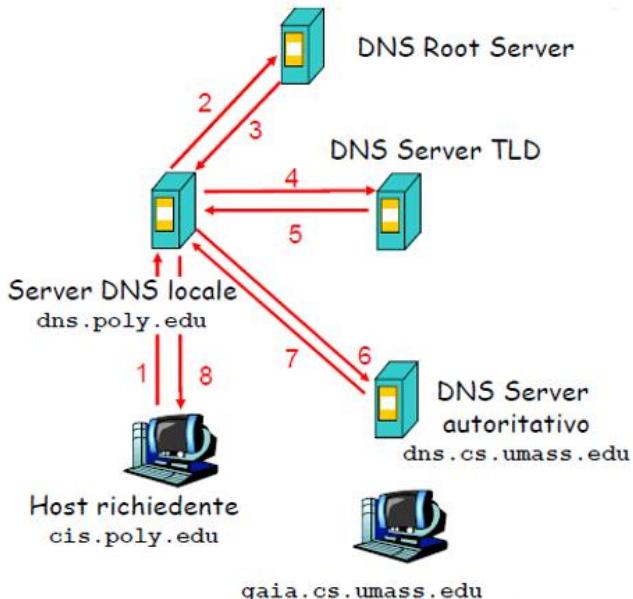
Server Autoritativi (authoritative server): ogni organizzazione dotata di host Internet pubblicamente accessibili (quali i server web e i server di posta) deve fornire i record DNS di pubblico dominio che mappano i nomi di tali host in indirizzi IP (possono essere gestiti dall'organizzazione o dal service provider)

Server DNS Locale

Ciascun ISP (università, società, ISP residenziale) ha un server DNS locale detto anche “**default name server**” che non appartiene alla gerarchia dei server

Quando un host effettua una richiesta DNS, la query viene inviata al suo server DNS locale che opera da proxy e inoltra la query in una gerarchia di server DNS

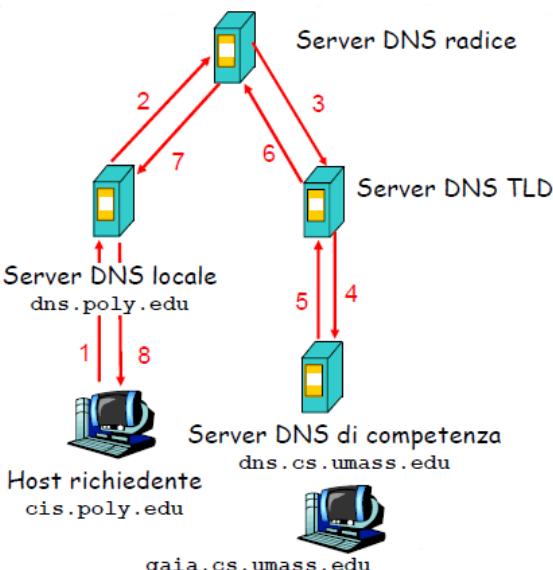
Esempio



L'host `cis.poly.edu` richiede l'indirizzo IP di `gaia.cs.umass.edu`

Query iterativa:

- 1) Il server contattato risponde con il nome del server da contattare
- 2) “Io non conosco questo nome, ma puoi chiederlo a questo server”.



Query ricorsiva

Affida il compito di tradurre il nome al server DNS contattato

Il traffico addizionale ed il delay introdotti dalla risoluzione dei nomi può essere diminuito mediante l'uso di cache

DNS Caching

Una volta che un server DNS conosce la mappatura «**nome, indirizzo**», la memorizza nella propria memoria cache dove le informazioni vengono cancellate dopo un certo periodo di tempo (tipicamente 48 ore)

Normalmente un server DNS locale memorizza nella cache gli indirizzi IP dei server TLD, dunque i server DNS radice non vengono visitati spesso

I meccanismi di aggiornamento/notifica sono definiti dall'IETF [RFC 2136]

DNS Resource Record (RR)

DNS è un database distribuito che memorizza i **Resource Record (RR)** che possono essere:

Formato RR: (**name, value, type, ttl**)

Type = A: **name** corrisponde al nome dell'host, **value** corrisponde all'indirizzo IP

Type = NS: **name** corrisponde al dominio (ad esempio foo.com), **value** corrisponde al nome dell'host del server autoritativo di questo dominio

Type = CNAME: **name** è il nome alias di un server (www.ibm.com è in realtà servereast.backup2.ibm.com), **value** è il nome canonico

Type = MX: **value** è il nome del server di posta associato a **name**

Formato messaggi DNS

Il protocollo DNS è formato da domande (**query**) e messaggi di **risposta**, entrambi con lo stesso **formato**:

Intestazione del messaggio composta da:

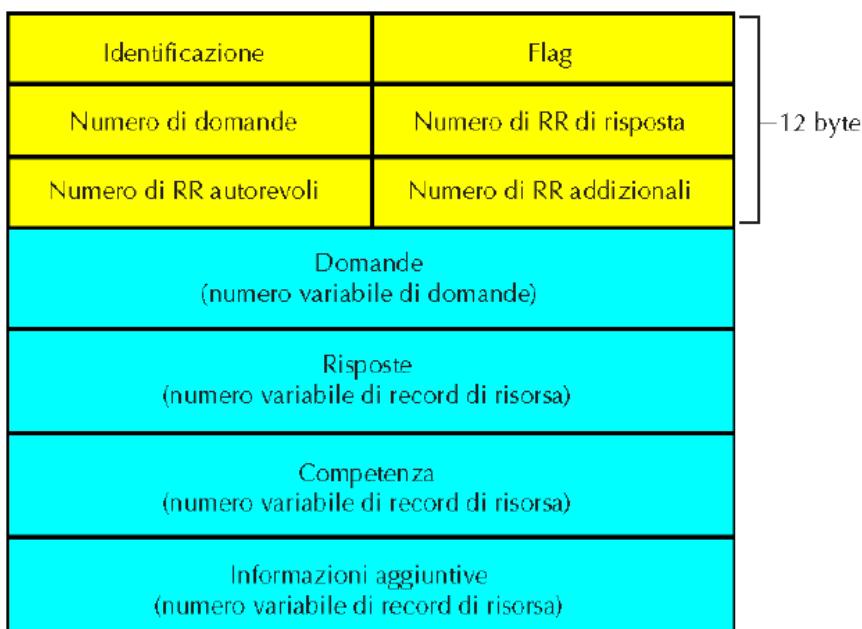
- **Identificazione:** numero di 16 bit per la domanda; la risposta alla domanda usa lo stesso numero
- **Flag** che mi indicano se è una domanda o una risposta, se è una richiesta di ricorsione oppure una ricorsione disponibile

Domande: Campi per il nome richiesto e il tipo di domanda

Risposte: RR nella risposta alla domanda

Competenza: record per i server autoritativi

Informazioni aggiuntive: informazioni extra



Architettura client-server

Il **server** è un host sempre attivo con un indirizzo IP fisso (oppure con servizi come dydns per assegnare un nome al dominio e rilevare il cambio IP automatico)

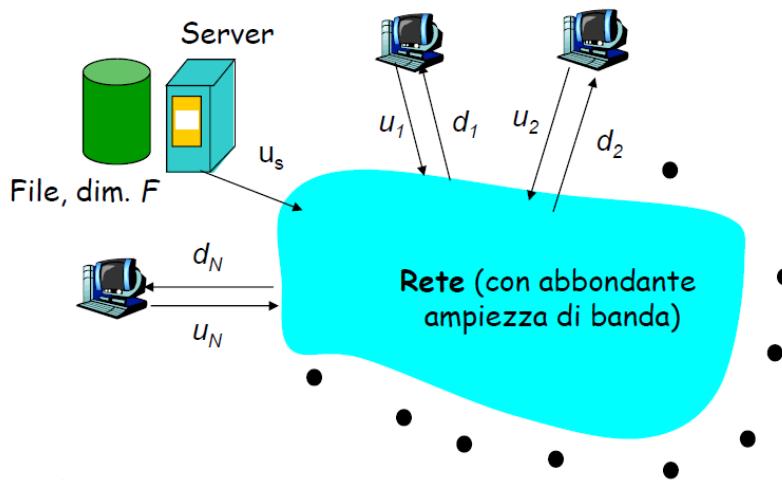
Il **client** comunica con il server che può contattare in qualunque momento e non necessita di avere indirizzi IP statici, ma non può comunicare direttamente con gli altri client

Architettura P2P

Non esiste un server sempre attivo poiché le coppie di host comunicano direttamente tra loro, ma i peer non devono necessariamente essere sempre attivi e avere indirizzo statico. Facilmente scalabile ma difficile da gestire.

Tempo di distribuzione di un file Architettura client-server

Domanda: Quanto tempo ci vuole per distribuire file da un server a N peer?



u_s : banda in upload del collegamento di accesso del server (bit/s)

u_i : banda in upload del collegamento di accesso dell'i-esimo peer (bit/s)

u_i : banda in download del collegamento di accesso dell'i-esimo peer (bit/s)

F : dimensione del file (bit)

N : numero dei peer

Il server invia in sequenza N copie:

$$Tempo = \frac{NF}{u_s}$$

Il client i impiega $\frac{F}{d_i}$ per scaricare

D_{cs} = Tempo di distribuzione di un file di dimensione F a N client usando l'approccio client/server che aumenta linearmente con N peer

$$D_{cs} = \max \left\{ \frac{NF}{u_s}, \frac{F}{\min(d_i)} \right\}$$

Tempo di distribuzione di un file Architettura P2P

Il server invia solo una volta il file poi sono i peer che redistribuiscono porzioni (**chunk**) del file verso gli altri peer utilizzando la propria banda di upload

Il server deve inviare una copia nel tempo $\frac{F}{u_s}$ (bottleneck del server)

Il client con la banda di download d_{min} più bassa riceve il file nel tempo $\frac{F}{d_{min}}$ (bottleneck del peer che riceve il file)

Devono essere scaricati **NF** bit

La banda massima possibile di upload dei peer è: $\mathbf{u}_s + \sum \mathbf{u}_i$

Ipotesi: piena utilizzazione della banda di upload

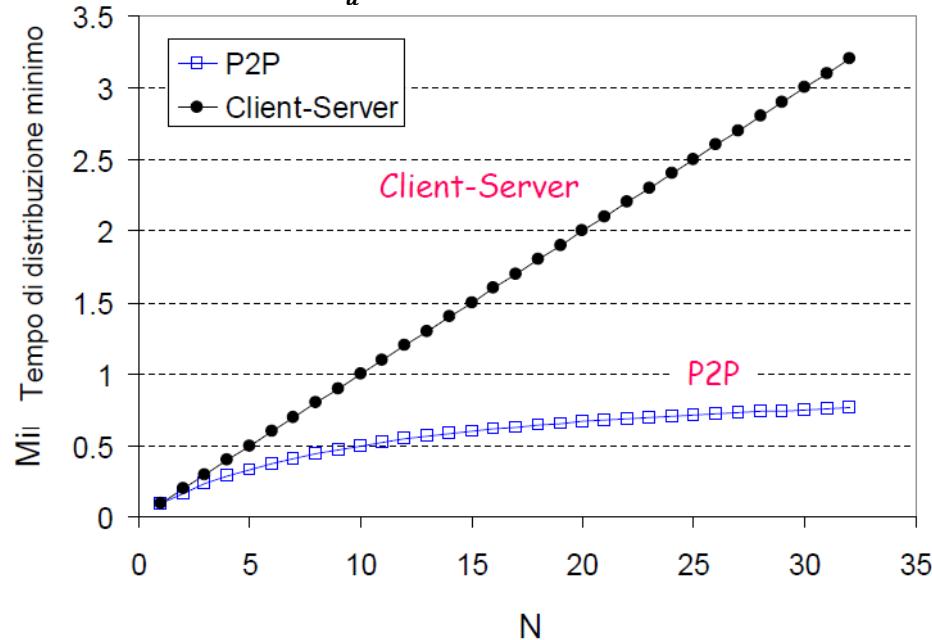
Il tempo minimo D_{P2P} di distribuzione del file tra gli N peer è dato da:

$$D_{P2P} = \max \left\{ \left(\frac{NF}{u_s}, \frac{F}{\min(d_i)} \right), \frac{NF}{u_s + \sum u_i} \right\}$$

Con $\frac{NF}{u_s + \sum u_i}$ dove il bottleneck è dato dalla distribuzione tra i peer

Tempo di distribuzione di un file: Server-Client vs P2P

Tasso di upload del client $\frac{F}{u} = 1$ ora, $\mathbf{u}_s = 10$ u, $d_{min} \geq u_s$



P2P: ricerca di informazioni

Indice nei sistemi P2P: corrispondenza tra le informazioni e la loro posizione negli host

File sharing: L'indice tiene traccia dinamicamente della posizione dei file che i peer condividono. I peer possono comunicare all'indice ciò che possiedono oppure consultare l'indice per determinare dove trovare i file.

Messaggeria istantanea: L'indice crea la corrispondenza tra utenti e posizione, quando l'utente lancia l'applicazione, informa l'indice della sua posizione in modo da essere individuabile agli altri peer.

P2P: directory centralizzata

Un unico server tiene traccia degli IP e dei contenuti di ogni Peer, in modo che quando un utente cerca un determinato file, il server gli invia la lista dei peer in cui può trovarlo.

Problemi: Unico punto di guasto, collo di bottiglia per le prestazioni, violazione del diritto d'autore
Il trasferimento dei file è distribuito, ma il processo di localizzazione è fortemente centralizzato

Bit Torrent

Il download di un file è chiamato **torrent**, per ogni torrent esiste un nodo centrale chiamato **tracker**, il nodo centrale però non contiene la lista dei contenuti dei peer ma solo gli indirizzi

Nel momento in cui un peer si aggiunge ad un torrent deve registrarsi sul nodo tracker e ottiene da lui la lista degli altri peer che può contattare per il download del file.

Un peer può aggiungersi o lasciare il torrent in qualsiasi momento, ed aggiorna il suo stato periodicamente.

Processo

I peer scaricano da altri peer **chunk** del file di uguale dimensione (256 kbyte)

Il nuovo peer tenta di stabilire delle connessioni TCP con i peer della lista, i peer connessi al nuovo peer sono detti **neighboring peer**

Il nuovo peer chiederà periodicamente ai vicini il download dei chunk del file, normalmente si usa un algoritmo detto (*rarest first*) richiesta dei chunk più rari

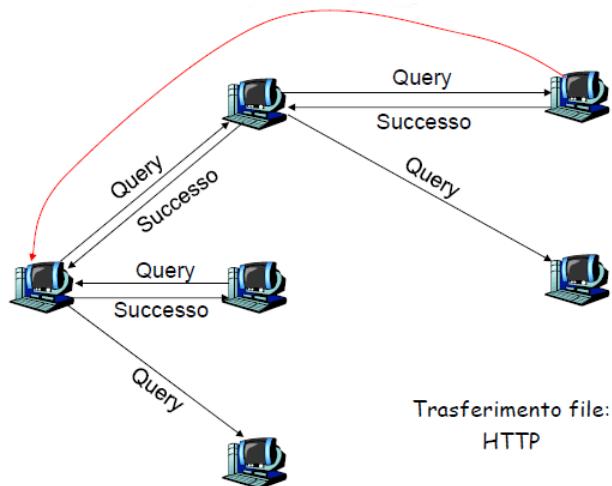
Query flooding

Il Sistema è completamente distribuito (nessun server centrale) e ciascun peer indicizza i file che rende disponibili per la condivisione (solo quelli).

Ha una **rete di copertura a grafo**:

Tutti i peer attivi e gli archi (collegamento virtuale) formano la rete di copertura, si crea un arco tra due peer solo se c'è una connessione TCP

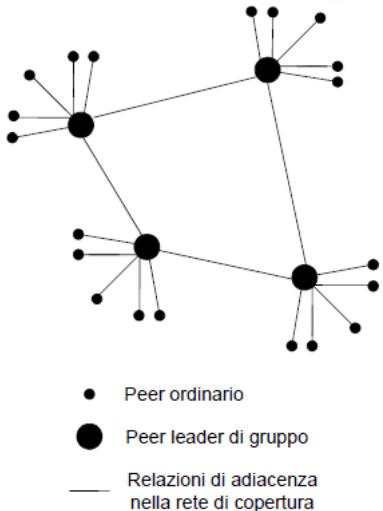
Un dato peer sarà solitamente connesso con meno di 10 peer vicini nella rete di copertura



Il messaggio di richiesta viene trasmesso sulle connessioni TCP esistenti e il messaggio di successo verrà trasmesso sul percorso inverso a quello di richiesta.

Copertura gerarchica

La copertura gerarchica combina le caratteristiche di indice centralizzato e query flooding



Ogni peer è assegnato a un **leader di gruppo** che ha il compito di tenere traccia del contenuto di tutti i suoi figli, si creano connessioni TCP tra peer e il loro leader e connessioni tra qualche coppia di leader

NAT: evita che un host al di fuori della rete domestica crei una connessione con un host all'interno di questa

Architetture ibride (client-server + P2P)

Skype: Applicazione P2P di Voice over IP, ha un server centralizzato: ricerca e memorizza gli indirizzi correnti degli utenti ma la connessione è di tipo P2P diretta

Processi comunicanti

Processo: Programma in esecuzione su di un host.

All'interno dello stesso host due processi comunicano utilizzando schemi interprocesso definiti dal Sistema Operativo mentre processi su host differenti comunicano attraverso la rete mediante lo scambio di messaggi.

Servizio di trasporto di un'applicazione

Perdita di dati: alcune applicazioni (ad esempio, audio) possono tollerare qualche perdita, altre applicazioni (ad esempio, trasferimento di file, telnet) richiedono un trasferimento dati affidabile al 100%

Ritardo: alcune applicazioni (ad esempio, telefonia, Internet, giochi interattivi) per essere “realistiche” richiedono piccoli ritardi

Throughput: alcune applicazioni (ad esempio, quelle multimediali) per essere “efficaci” richiedono un’ampiezza di banda minima (bit rate) garantita, mentre altre applicazioni (“applicazioni elastiche”) utilizzano l’ampiezza di banda che si rende disponibile

STRATO DI TRASPORTO CAPITOLO 4

Socket

Una **socket** è analoga a una porta: un processo che vuole inviare un messaggio, lo emette attraverso la propria “porta” (socket), il processo presuppone l’esistenza di un’infrastruttura esterna che trasporterà il messaggio attraverso la rete fino alla “porta” del processo di destinazione

Fornisce un **collegamento logico** tra processi in host remoti

Le funzioni di trasporto vengono eseguite nei sistemi terminali:

- **Lato emittente**: divide i messaggi (A-PDU) in **segmenti** (T-PDU) e li inoltra allo strato di rete
- **Lato ricevente**: esegue la ricostruzione dei messaggi e li inoltra allo stato di applicazione



Struttura del segmento TCP/UDP

Multiplexing

L’host emittente raccoglie i dati da varie socket, li incapsula in pacchetti IP con la sua intestazione e li trasmette all’host destinatario.

Demultiplexing

L’host ricevente riceve i pacchetti IP che trasportano i segmenti, ogni pacchetto ha un **indirizzo IP di origine** e un **indirizzo IP di destinazione**.

Ogni pacchetto trasporta un solo segmento a livello di trasporto e ogni segmento ha un numero di **porta di origine** e un numero di **porta di destinazione**

L’host usa gli indirizzi ip e i numeri di porta per inviare il segmento alla socket appropriata, la scelta dipende dal tipo di comunicazione:

Senza connessione: (prediletto il protocollo UDP) Un host crea la socket UDP identificata da:

- **Indirizzo IP di destinazione**
- **Numero di porta di destinazione**

Quando l’host riceve il segmento controlla il numero di porta e invia il segmento alla socket con quel numero di porta

Importante: Pacchetti IP con lo stesso numero di porta di destinazione, ma con indirizzi IP di sorgente, e/o numeri di porta di sorgente diversi vengono inviati alla stessa socket

Con connessione: (prediletto il protocollo TCP) Una socket TCP è identificata da:

- **Indirizzo IP di origine**
- **Numero di porta di origine**
- **Indirizzo IP di destinazione**
- **Numero di porta di destinazione**

L’host ricevente usa questi parametri per inviare i segmenti alla socket appropriata.

L’host server può supportare più socket TCP contemporaneamente

I server web hanno socket differenti per ogni connessione: HTTP non-persistente avrà una socket differente per ogni richiesta

Indirizzamento

Statico: Alcune delle app più diffuse hanno numeri di porta assegnati (**well-known port numbers**) gestiti dalla IANA ed aggiornato in tempo reale: intervallo 0 - 1023

Numero	Applicazione	Numero	Applicazione
7	Echo	37	Time
21	FTP (File Transfer Protocol)	53	Domain Name Server
23	TELNET	80	HTTP
25	SMTP (Simple Mail Transport Protocol)	119	NNTP (USENET New Transfer Prot.)

Dinamico: sono identificativi assegnati direttamente dal sistema operativo al momento dell'apertura della connessione (si utilizzano valori maggiori di 1023)

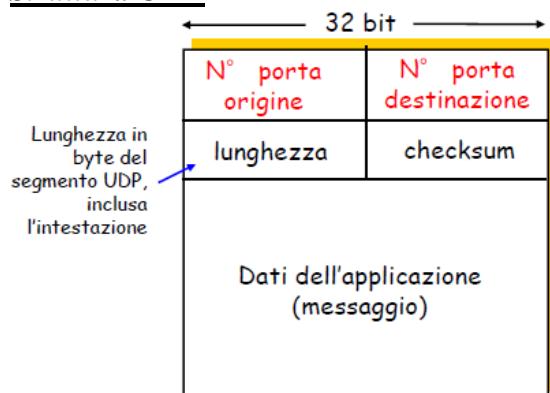
UDP (USER DATAGRAM PROTOCOL) datagramma

Servizio senza connessione (**no handshaking**, **ogni segmento gestito in modo indipendente dagli altri**) in cui il trasferimento dei dati è inaffidabile.

Non ha il setup della connessione, il controllo di flusso, il controllo della congestione, il controllo del ritardo e non garantisce né throughput né sicurezza poiché i segmenti possono essere perduti o consegnati fuori sequenza.

Vengono impiegati per DNS, SNMP o nelle applicazioni multimediale

Struttura UDP



Struttura del segmento UDP

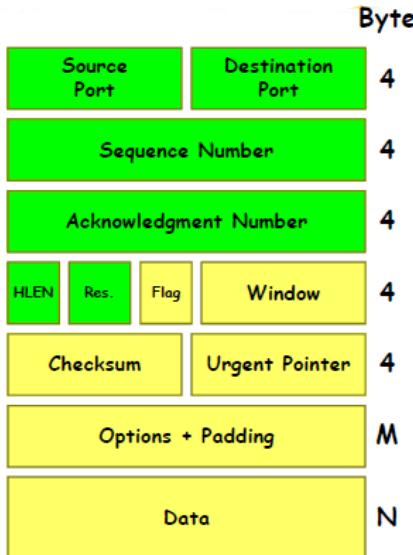
TCP

È un protocollo con connessione che interpreta il flusso dati proveniente dallo strato applicativo.

Funzioni:

- Indirizzamento di un'applicazione
- Controllo di sequenza delle unità informative
- Controllo e recupero di errore
- Controllo di flusso (regolare quantità dati in trasmissione e fare in modo che l'host ricevente li riceva correttamente)
- Controllo di congestione (regolare quantità dati in trasmissione per non sovraccaricare la rete)

Struttura TCP



Source port - Destination port (16 bit ciascuno)

Sequence number (32 bit): numero d'ordine del primo byte dati contenuto nel campo dati

Acknowledgment Number (ACKNum 32 bit): se il bit di ACK in Flag vale 1, contiene il numero di sequenza del prossimo byte che il ricevente si aspetta di ricevere.

HLEN (4 bit): numero di parole da 32 bit contenute nell'intestazione del segmento, la quale non supera i 60 ottetti ed è sempre un multiplo di 32

Reserved (6 bit): riservato per usi futuri

Flag: a seconda del bit che imposto ad 1 ho diversi significati:

- **URG (urgent)**: è uguale a uno quando il campo "Urgent Pointer" contiene un valore significativo
- **ACK (riscontro)**: è uguale a uno quando il campo "Acknowledgement Number" contiene un valore valido
- **PSH (push)**: è uguale a uno quando l'applicazione indica che i dati vengano consegnati all'applicazione ricevente prescindendo dal riempimento dei buffer di ricezione
- **RST (reset)**: è uguale a uno in caso di richiesta di re-inizializzazione della connessione
- **SYN (sincronizzazione)**: è uguale a uno solo nel primo segmento inviato durante la fase di sincronizzazione fra le entità TCP
- **FIN (fine)**: è uguale a uno quando la sorgente ha esaurito i dati da trasmettere

Window (16 bit): larghezza della finestra misurata in ottetti che ad iniziare dal valore di ACK Number, l'emittitore del segmento autorizza a trasmettere

Checksum (16 bit): protegge il segmento più alcuni campi dell'header IP (pseudo header)

Urgent pointer (16 bit): contiene il numero di sequenza dell'ultimo byte dei dati che devono essere consegnati urgentemente al processo ricevente

Options (di lunghezza variabile): sono presenti solo raramente

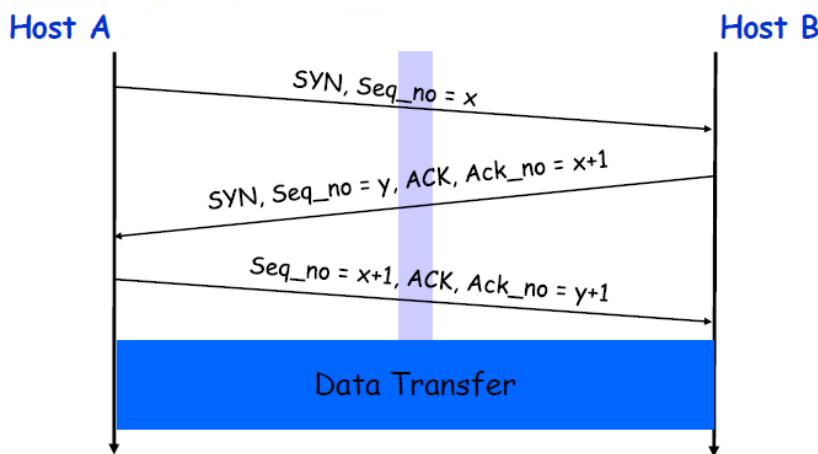
Padding (di lunghezza variabile): impone che l'intestazione abbia una lunghezza multipla di 32 bit

Funzionamento

Nella fase di instaurazione della connessione le due entità TCP remote si sincronizzano scambiandosi gli identificatori dei socket (porta, indirizzo IP), il proprio numero di sequenza iniziale, che rappresenta il numero a partire dal quale tutti gli ottetti emessi saranno sequenzialmente numerati e il valore iniziale della finestra di ricezione

Handshake a tre passi:

- 1) **Instaurazione della connessione:** host A invia segmento SYN all'host B, specificando il numero di sequenza iniziale utilizzato nel verso A → B
- 2) Host B riceve SYN, alloca i buffer per la ricezione e risponde con un segmento SYN ACK specificando il numero di sequenza iniziale del server utilizzato nel verso B → A
- 3) Host A riceve un segmento SYN ACK e risponde con un segmento ACK che può contenere dati.



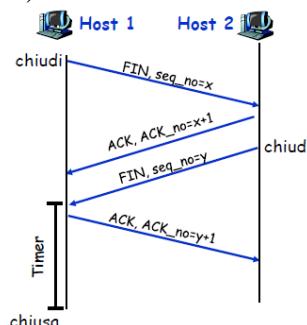
Maximum Segment Size (MSS)

Quando l'entità TCP emittente invia la prima TCP- PDU (SYN) può inserire l'informazione relativa alla massima dimensione del campo dei dati di utente di una TCP-PDU (Maximum Segment Size - MSS)

L'entità ricevente risponde comunicando la propria MSS, nel caso di uno scambio bidirezionale, la dimensione della MSS è scelta in modo indipendente nei due versi e può quindi essere diversa nelle due direzioni

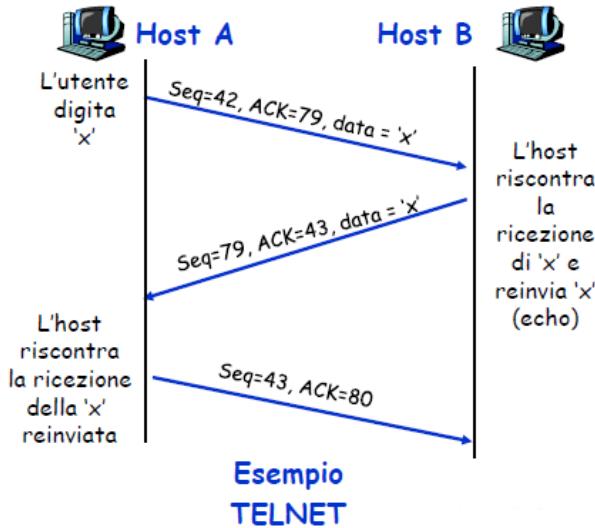
Chiusura della connessione

- 1) Host 1 invia un segmento di controllo FIN al server
- 2) Host 2 riceve il segmento FIN e risponde con un ACK
- 3) Host 2 chiude la connessione e invia un FIN
- 4) Host 1 riceve FIN e risponde con ACK
- 5) Alla ricezione dell'ACK si avvia un timer al termine del quale si chiude la connessione.



Controllo di sequenza

Il **numero di sequenza** è il numero del primo byte del prossimo segmento, esso viene trasmesso insieme all'**ACK** che viene aggiornato dal ricevente per dire all'emittente da che punto in poi trasmettere ogni volta. (es. N* seq = 42, lunghezza segmento = 6 – prossimo n*seq = $42+6+1 = 49$)
 La gestione dei segmenti fuori sequenza non è specificata però in questo standard, ma dipende dall'implementazione



Controllo d'errore

L'entità TCP emittente inserisce una codifica per la rivelazione d'errore nell'intestazione del segmento (**checksum**), e l'entità TCP ricevente la utilizza per rilevare eventuali errori.

L'entità ricevente può mandare i suoi riscontri ACK o con segmenti vuoti (senza dati) appena fa il controllo (**immediata**) o in modalità **piggybacking** se sta trasmettendo anch'essa, tenendoli quindi da parte finché anche lei trasmette qualcosa.

Retransmission timeout (RTO):

Temporizzatore di valore variabile in modo adattativo attivato dall'entità emittente che risulta:

- **attivato** nel momento in cui un segmento viene inoltrato su una connessione uscente (il timer è associato all'ultimo segmento non riscontrato)
- **disattivato** nel momento in cui viene ricevuto un ACK relativo al segmento corrispondente e quando tale ricezione avviene prima che l'RTO si esaurisca

Riscontri

L'entità TCP ricevente può emettere i riscontri (ACK) secondo due modalità:

Immediata: appena vengono accettati i dati, emette immediatamente un segmento vuoto che contiene l'appropriato numero di riscontro

Cumulativa: appena vengono accettati i dati, tiene memoria della necessità di inviare un riscontro, ma aspetta un segmento in uscita nel quale inserirlo.

Per evitare lunghi ritardi, attiva un timer di finestra, se il tempo di questo timer si esaurisce prima che venga inviato un riscontro, emette un segmento vuoto che contiene l'appropriato numero di riscontro

Round Trip Time (RTT) e timeout

SampleRTT: tempo misurato dalla trasmissione di un segmento fino alla ricezione dell'ACK relativo (ignora le ritrasmissioni), esso varia, quindi occorre una stima "smoothed" di RTT:

$$\text{EstimatedRTT} = (1 - \alpha) * \text{EstimatedRTT} + \alpha * \text{SampleRTT}$$

L'influenza dei vecchi campioni decresce esponenzialmente; Valore tipico: $\alpha = 0,125$

Più grande è la variazione dell'EstimatedRTT, maggiore è il margine di sicurezza

Stima della deviazione standard dell'EstimatedRTT

$$\text{DevRTT} = (1 - \beta) * \text{DevRTT} + \beta * |\text{SampleRTT} - \text{EstimatedRTT}| \quad \text{con } \beta \text{ tipico} = 0,25$$

Valore Retransmission TimeOut (RTO)

$$\text{RTO} = \text{EstimatedRTT} + 4 * \text{DevRTT}$$

↑ ↑
Estimated RTT margine di sicurezza

Exponential RTO Backoff

Determina il valore di RTO associato ad un segmento riemesso:

è consigliabile variare il valore di RTO sui segmenti riemessi perché l'esaurimento dell'RTO è dovuto a congestione in rete, allo stesso modo è consigliabile variare il valore di RTO delle sorgenti che sono coinvolte nella congestione per evitare riemissioni contemporanee

Una sorgente TCP aumenta il valore di RTO per ogni riemissione (**exponential backoff process**) (normalmente $q = 2$)

$$RTO_{i+1} = q \cdot RTO_i$$

Algoritmo di Karn

L'entità TCP ricevente non distingue se il riscontro si riferisce alla prima emissione del segmento (RTO troppo elevato con perdita di efficienza e inutili ritardi) o alla riemissione del segmento (RTO troppo breve e quindi riemissioni eccessive e nuovi errori di misura).

L'algoritmo di Karn stabilisce di non considerare il RTT dei segmenti riemessi, usare come RTO il valore dato dalla procedura di exponential backoff e ricalcolare il nuovo valore di RTO solo al momento della ricezione di un ACK di un segmento non riemesso

Generazione di ACK (riassunto)

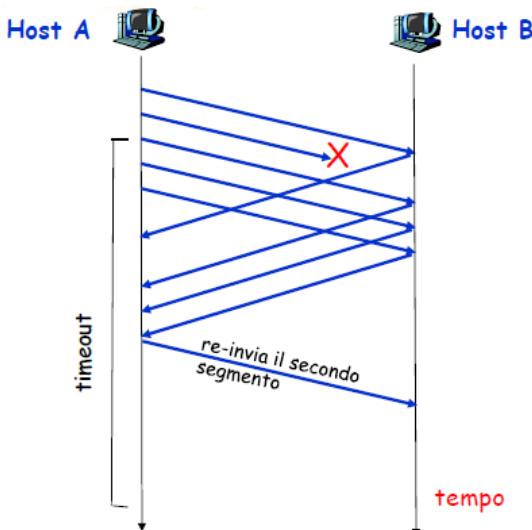
Evento presso il ricevente	Azione del ricevente
Arrivo ordinato di un segmento. Tutti i dati fino al numero di sequenza atteso sono già stati riscontrati	ACK ritardato. Attende fino a 500 ms l'arrivo del prossimo segmento. Se il segmento non arriva, invia un ACK
Arrivo ordinato di un segmento. Un altro segmento è in attesa di trasmissione dell'ACK	Invia immediatamente un singolo ACK cumulativo, riscontrando entrambi i segmenti
Arrivo non ordinato di un segmento con numero di sequenza superiore a quello atteso, viene rilevato un "fuori sequenza"	Invia immediatamente un ACK duplicato, indicando il numero di sequenza del prossimo byte atteso
Arrivo di un segmento che ripristina parzialmente o completamente il "fuori sequenza"	Invia immediatamente un ACK, ammesso che il segmento sia in sequenza con l'ultimo segmento riscontrato

Fast retransmit

Il periodo di timeout spesso è relativamente lungo perciò l'entità TCP emittente può rivelare precocemente i segmenti perduti tramite l'analisi degli ACK duplicati:

L'entità TCP emittente spesso invia molti segmenti e se un segmento viene smarrito, è probabile che ci saranno molti ACK duplicati

Se l'entità TCP emittente riceve 3 ACK duplicati per lo stesso dato, suppone che il segmento che segue il dato riscontrato sia andato perduto e effettua una ritrasmissione rapida prima che scada il timer



Controllo di Flusso

Ha lo scopo di limitare il ritmo di emissione dei dati da parte di un host per evitare la saturazione della capacità del buffer di ricezione.

TCP utilizza un controllo di flusso basato su una finestra scorrevole di lunghezza variabile (**Recwindow**) la cui larghezza e scorrimento sono controllati dal ricevente

Il controllo di flusso opera a livello di byte: Gli ottetti sono numerati in sequenza a partire dal numero scelto durante il 3-way handshaking (procedura di instaurazione della connessione)

La procedura di controllo di flusso TCP utilizza i seguenti parametri:

- **SN (Sequence Number)**: riferito al primo byte (ottetto) contenuto nel segmento
- **AckN (Acknowledgement Number)**: riferito al prossimo ottetto che l'entità ricevente aspetta di ricevere
- **RecWindow (Window)**: esprime il numero massimo di ottetti che l'entità emittente può emettere senza ricevere un riscontro per alcuno di questi

Un riscontro (**AckN = X** e **RecWindow = W**) significa che sono riscontrati tutti gli ottetti ricevuti fino a quello numerato con $X - 1$;

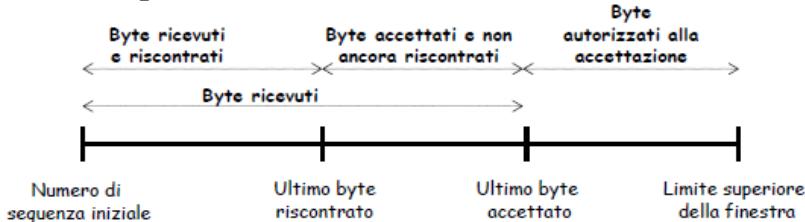
L'entità TCP emittente è autorizzata a trasmettere fino a ulteriori W ottetti, ovvero fino all'ottetto numerato con $X + W - 1$

Controllo della finestra

Puntatori per il controllo a finestra lato emittente:



Puntatori per il controllo a finestra lato ricevente:



Throughput di una connessione TCP

L'efficienza (throughput - TH) di una connessione TCP, nell'ipotesi di overhead nullo e di assenza di ritrasmissioni, è dato da:

$$TH = \frac{\text{tempo di ritrasmissione utile in un RTT}}{RTT} = \min \left[1, \frac{W/C}{2\alpha/C} \right] = \min \left[1, \frac{W}{2\alpha} \right]$$

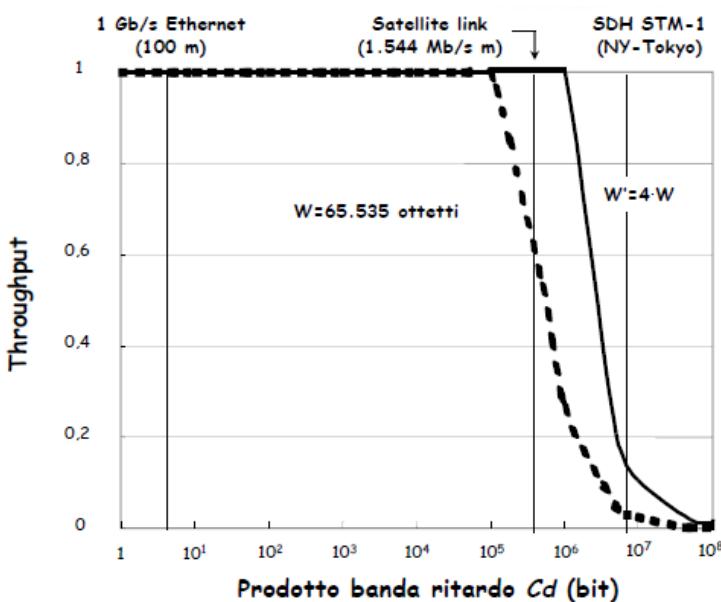
$$= \begin{cases} 1 & \text{se } W \geq 2\alpha \\ \frac{W}{2\alpha} & \text{se } W < 2\alpha \end{cases}$$

Dove:

- **W**: larghezza(byte) della finestra in trasmissione
- **C**: bit rate della connessione
- **d**: ritardo di propagazione sulla connessione
- **Cd**: prodotto banda-ritardo
- $\alpha = C \frac{d}{8}$ (rapporto tra ritardo di propagazione e tempo di trasmissione di un ottetto) prodotto banda ritardo espresso in byte

Il TCP non è adatto a collegamenti ad elevato bit-rate o con elevato ritardo, infatti se il prodotto banda ritardo è elevato il throughput del TCP decresce sensibilmente

NOTA: L'aumento della dimensione della finestra non è un intervento risolutivo



Controllo di congestione

Ha l'obiettivo di non sovraccaricare la **rete** per evitare di avere ritardi enormi (accodamento nei buffer dei router) o peggio perdita di pacchetti (overflow nei buffer dei router)

Approcci al controllo della congestione (non si capisce granché dalle slide)

Controllo di congestione assistito dalla rete: i router forniscono un feedback ai sistemi terminali:

un singolo bit per indicare la congestione (**Explicit Congestion Notification -ECN**)

L'entità ricevente comunica in modo esplicito al mittente il bit-rate in trasmissione

Il metodo adottato dal TCP per effettuare quest'operazione invece è il controllo **punto-punto** ovvero deduce la disponibilità osservando le perdite e i ritardi nei sistemi terminali

Il protocollo TCP utilizza i seguenti meccanismi:

- Esaurimento dell'RTO come sintomo di cogestione

- **Congwindow (finestra di congestione)** che si affianca a quella di ricezione operante nel controllo di flusso per impostare un limite aggiunto alla quantità di traffico che un host può inviare in una connessione

- Soglia (**Threshold**): il valore è pari alla metà del valore della congwin al momento in cui viene rilevata una perdita.

All'inizio della connessione (slow start) la soglia è posta uguale a ∞

L'entità emittente determina nel tempo il valore della finestra disponibile (**Available Window - Awdn**):

Awdn = numero di segmenti di lunghezza massima (MSS) che possono essere inviati senza riscontro

Il valore di Awdn non deve superare il minimo tra le larghezze **Congwin della finestra di congestione** e **RecWindow della finestra di ricezione**:

$$Awdn \leq \min \{ Congwin, RecWindow \}$$

Congwin ed RecWindow sono quantità espresse in numero di segmenti MSS

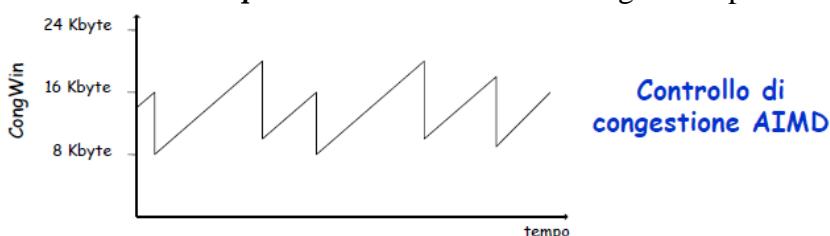
RecWindow è la larghezza comunicata nell'ultimo ACK ricevuto e ottenuta dall'entità TCP emittente dividendo il numero contenuto nel campo Window di questo ACK per il numero di ottetti che compongono una MSS

Additive-Increase Multiplicative-Decrease

Aumenta il valore di **CongWin** (sondando la rete) fino a quando non si verifica una perdita

Incremento additivo: Aumenta CongWin di 1 MSS a ogni RTT in assenza di eventi di perdita

Decremento Moltiplicativo: Riduce a metà CongWin dopo un evento di perdita



Controllo di congestione TCP

Approssimativamente il rate di emissione dei segmenti è dato da:

$$\text{Frequenza d'invio} = \frac{\text{CongWin}}{\text{RTT}} \text{ byte/sec}$$

Il mittente percepisce la congestione se si esaurisce il timeout o riceve 3 ACK duplicati

Il mittente TCP riduce la frequenza d'invio (CongWin) dopo un evento di perdita

Fasi della procedura

Per evitare la congestione, l'emettitore TCP segue una **procedura ciclica** in cui ogni ciclo è composto da due fasi:

- **Slow Start**: Incremento esponenziale della Congwin
- **Congestion Avoidance**: Incremento lineare della Congwin

Slow Start

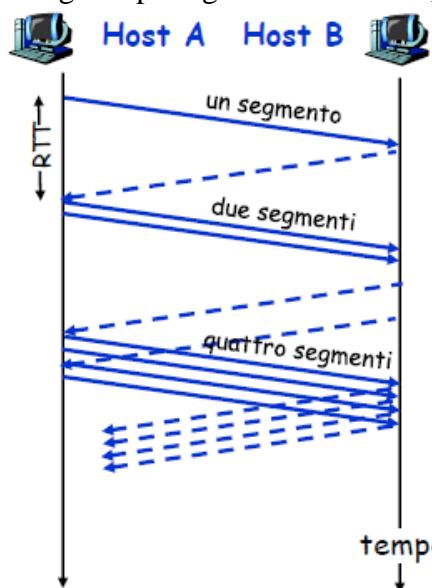
Quando si stabilisce una connessione:

- CongWin = 1 MSS
- Soglia = ∞

La larghezza di banda disponibile potrebbe essere $>>$ MSS/RTT il che consente di raggiungere rapidamente una bit-rate d'invio significativo

Quando inizia la connessione, la frequenza aumenta in modo esponenziale, fino a quando non si verifica un evento di perdita:

Raddoppia CongWin a ogni RTT, ciò avviene incrementando di un segmento MSS il valore della CongWin per ogni ACK ricevuto, l'incremento iniziale è lento, ma poi cresce in modo esponenziale



Quando si verifica un evento di perdita si pone (decremento moltiplicativo):

$$\text{CongWin(new)} = 1 \text{ MSS}$$

$$\text{Soglia} = \frac{\text{CongWin(old)}}{2}$$

Congestion avoidance

Se l'aumento che si ha nella fase Slow Start raggiunge e supera il valore di soglia e cioè se

Congwin \geq Soglia, l'incremento di Congwin diventa **lineare** al crescere di RTT:

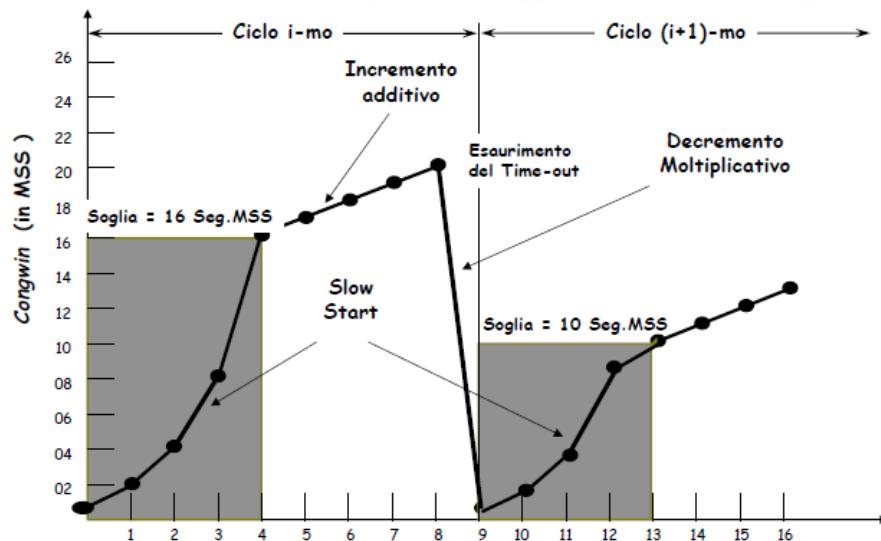
Se Congwin = w e se $w \geq$ Soglia, dopo l'arrivo di w riscontri consecutivi, la larghezza Awdn viene incrementata di 1 segmento MSS in ciascun RTT in cui si registra l'arrivo di un intero gruppo di riscontri dei contenuti della finestra di congestione

Questo incremento lineare continua finché i riscontri arrivano prima dei loro rispettivi RTO

Questo aumento ha un limite superiore corrispondente al raggiungimento di uno stato di saturazione su uno dei collegamenti lungo il percorso o in uno dei nodi attraversati.

Nell'ipotesi che **Congwin < Recwindow**, il limite superiore dell'aumento della Congwin è determinato dal verificarsi di un evento di perdita di un segmento e di un conseguente raggiungimento del relativo RTO

Esempio



- Il valore Soglia iniziale è uguale a 16 MSS
- Durante la fase di slow start, la Soglia è raggiunta all'istante 4
- Il valore di Congwin cresce poi linearmente, finché non si verifica una perdita (istante 8) e quando Congwin = 20 MSS
- Il valore Soglia è allora ridotto a 0,5 Congwin = 10 MSS e la finestra di congestione è successivamente posta a 1 MSS
- La fase di slow start ricomincia poi all'istante 9 e ha termine all'istante 13, quando Congwin ha raggiunto il valore 10 MSS
- Da quest'ultimo valore ricomincia l'incremento additivo di Congwin che avrà termine quando si verificherà una nuova perdita

Riassunto

Quando **CongWin** è sotto la **soglia**, il mittente è nella fase di **slow start**; la finestra cresce in modo esponenziale

Quando **CongWin** è sopra la **soglia**, il mittente è nella fase di **congestion avoidance**; la finestra cresce in modo lineare

Quando si verificano **tre ACK duplicati**, il valore di Soglia viene impostato a $\frac{\text{CongWin}}{2}$ e CongWin viene impostata al valore di Soglia

Quando **scade il timeout**, il valore di Soglia viene impostato a $\frac{\text{CongWin}}{2}$ e CongWin è impostata a 1 MSS

Stato	Evento	Azione del mittente TCP	Commenti
Slow Start (SS)	Ricezione di ACK per dati precedentemente non riscontrati	$CongWin = CongWin + MSS$, If ($CongWin > Threshold$) imposta lo stato a "Congestion Avoidance"	$CongWin$ raddoppia a ogni RTT
Congestion Avoidance (CA)	Ricezione di ACK per dati precedentemente non riscontrati	$CongWin =$ $= CongWin + MSS * (MSS/CongWin)$	Incremento additivo: $CongWin$ aumenta di 1 MSS a ogni RTT
SS o CA	Rilevato un evento di perdita da tre ACK duplicati	$Threshold = CongWin/2$, $CongWin = Threshold$, imposta lo stato a "Congestion Avoidance"	Ripristino rapido con il decremento moltiplicativo. $CongWin$ non sarà mai minore di 1 MSS
SS o CA	Timeout	$Threshold = CongWin/2$, $CongWin = 1 MSS$, imposta lo stato a "Slow Start"	Entra nello stato "Slow Start"
SS o CA	ACK duplicato	Incrementa il conteggio degli ACK duplicati per il segmento in corso di riscontro	$CongWin$ e $Threshold$ non variano

Throughput TCP

Ignoriamo le fasi di slow start; Sia W la dimensione della finestra quando si verifica una perdita. Quando la finestra è W, si ha:

$$TH1 = \frac{W}{RTT}$$

Subito dopo la perdita, la finestra si riduce a W/2, quindi:

$$TH2 = \frac{W}{2*RTT}$$

Poiché l'aumento è lineare:

$$TH = \frac{TH1+TH2}{2} = 0,75 \text{ W/RTT}$$

STRATO DI RETE (PARTE 1) CAPITOLO 5

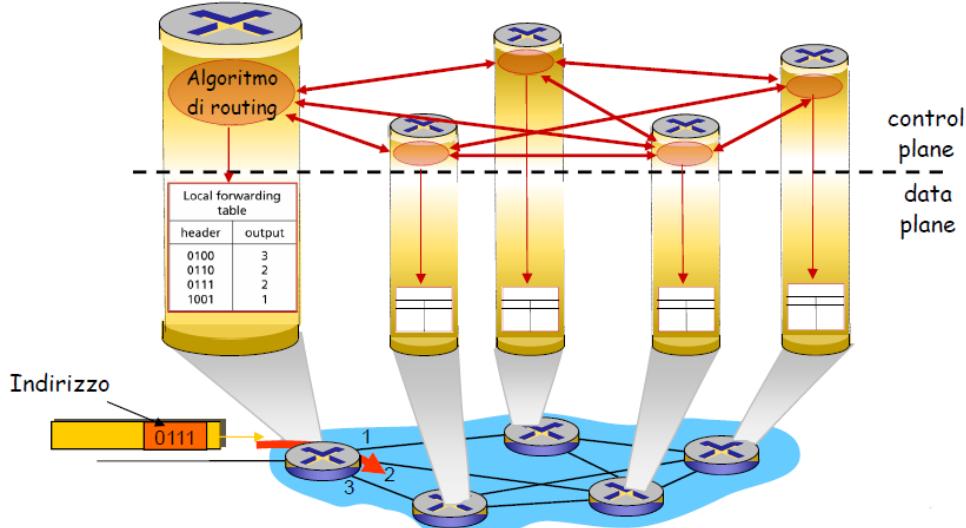
Funzioni

Routing: instradamento per determinare il percorso dei pacchetti dall'origine alla destinazione, lo fa attraverso protocolli e algoritmi specifici (**Funzione Decisionale - Control Plane**)

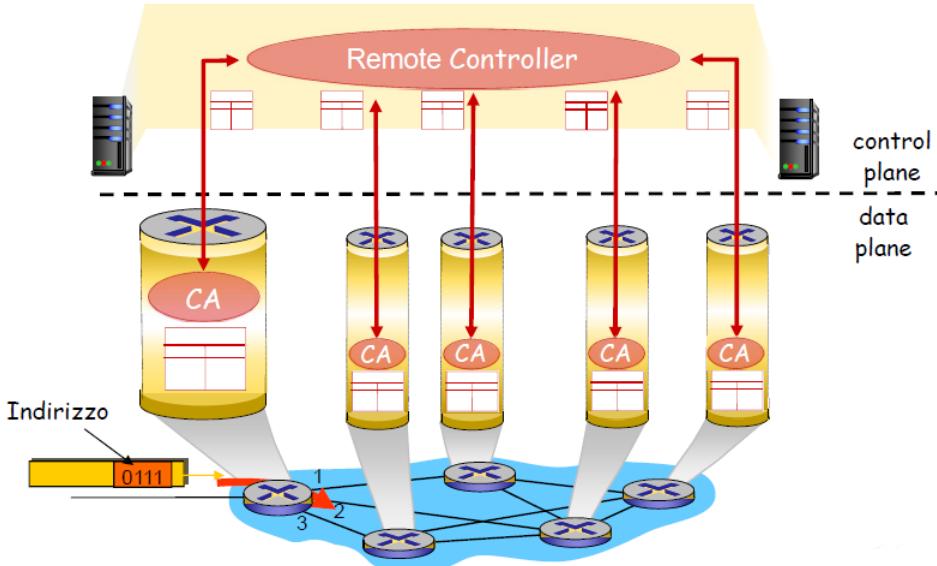
Forwarding: inoltro di pacchetti da un'interfaccia di ingresso di un nodo verso l'opportuna interfaccia di uscita individuata dalla funzione di routing (**Funzione attuativa – Data Plane**)

Due possibili architetture:

- **Distribuito:** algoritmi di routing implementati in ogni router che cooperano tra singole componenti



- **Centralizzato:** Software Defined Networking (SDN), un controller remoto interagisce con i Control Agent (CA) presenti in ogni router



Tipologie di servizio di rete

Senza connessione (reti a datagramma): i pacchetti vengono inviati senza un preventivo accordo tra sorgente e destinazione, ogni pacchetto è indipendente dagli altri, operazione di instradamento deciso pacchetto per pacchetto. I router hanno un funzionamento **stateless**

Con connessione (reti a circuito virtuale (VC) o Label Switching (es.MPLS): instaura una connessione di rete prima dell'invio dei pacchetti nella quale si decide il cammino(path) dei pacchetti. I nodi mantengono info riguardo allo stato delle connessioni (funzionamento **statefull**)

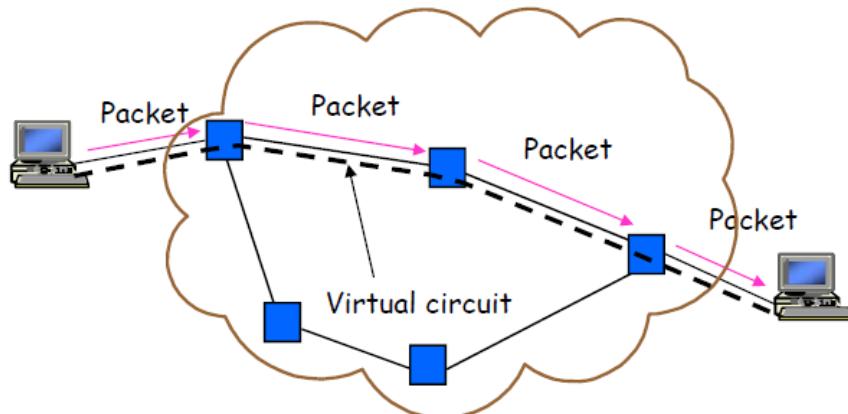
Tipologia Reti

Switching Circuit: Rete con connessione garantisce affidabilità e qualità ma perde tempo nell'istaurazione della connessione

Packet Switching (Virtual Circuit):

È necessaria una fase di set-up della connessione e un protocollo di segnalazione

Tutti i pacchetti seguono lo stesso path in rete e la consegna dei pacchetti avviene in sequenza
L'informazione di indirizzamento contenuta nell'header di ogni pacchetto è l'identificatore della connessione a cui appartiene (l'identificazione della connessione avviene "per link")



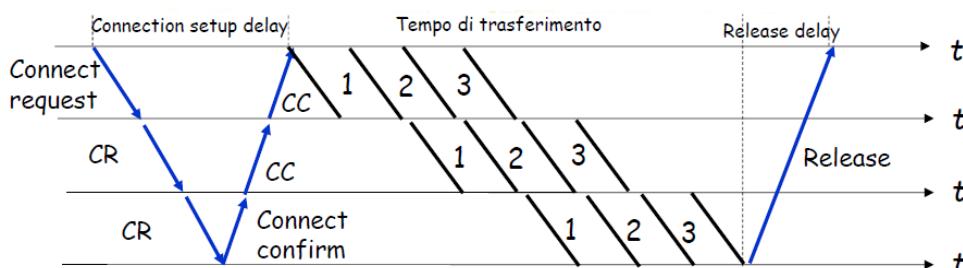
Virtual Circuit: Non stabilisco le risorse allocate per i pacchetti ma stabilisco la strada che i pacchetti faranno e tento di utilizzare quella strada ogni volta essa è libera.

Utilizza un protocollo di segnalazione in cui i messaggi di segnalazione vengono trasmessi lungo il path della connessione e fa due operazioni: inizializza le tabelle di forwarding dei nodi e determina la funzione di routing di ogni nodo per identificare il nodo successivo.

La connessione è identificata su ogni link da un "local tag" **VCI (Virtual Circuit Identifier - VCI)** in modo che gli switch possano aggiornare le loro tabelle di forwarding con la relazione tra il tag di entrata e di uscita.

Una volta che le tabelle di forwarding sono inizializzate i pacchetti possono essere trasmessi in rete

Connection Setup Delay



Il ritardo di instaurazione della connessione (**connection setup delay**) si somma al ritardo di transito dei pacchetti

Efficienza (ρ)

$$\rho = \frac{\text{transfer delay}}{\text{setup delay} + \text{transfer delay} + \text{release delay}}$$

Il ritardo addizionale è **tollerabile** se è inferiore al ritardo di trasferimento dei dati, mentre è **inaccettabile** se devono essere trasferiti pochi pacchetti

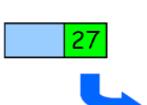
Virtual Circuit Forwarding Table

Ogni porta di ingresso di un router ha una propria forwarding table

Si utilizza il VCI contenuto nell'header di ogni pacchetto come indice di accesso alla tabella

Si individua il record corrispondente al VCI, si legge la porta di uscita e il valore del VCI sul link d'uscita scritto nell'header del pacchetto

Input VCI	Output port	Output VCI
12	13	44
15	15	23
27	13	16
58	7	34



Riassumendo

Un circuito virtuale consiste in un percorso tra origine e destinazione, identificatori di connessione (VCI), uno per ciascun link e righe nella tabella di forwarding in ciascun nodo (switch)

Il pacchetto di un circuito virtuale trasporta il VCI nella propria intestazione

Il VCI del pacchetto cambia su tutti i collegamenti lungo un percorso

Router (Esempio di Rete senza Connessione)

I router utilizzano gli **indirizzi di destinazione** per effettuare il forwarding poiché i pacchetti possono seguire percorsi diversi in rete (la consegna in sequenza non è garantita). Per l'indirizzamento si utilizzano delle Tabelle di Routing con gli indirizzi di rete. Ovviamente non è che ogni router ha una lista di tutti i possibili indirizzi, ma ha una tabella che per ogni intervallo di indirizzi ha l'interfaccia d'uscita e il nodo successivo, al resto penseranno i nodi successivi.

Il router non conosce tutto il cammino completo del pacchetto, ma per scegliere l'interfaccia d'uscita ne utilizza il **Prefisso**.

Il router possiede un Database Topologico dove memorizza le informazioni sulla topologia della rete a cui è connesso e lo aggiorna in base alle info che riceve da altri router con messaggi sui protocolli di routing.

Sulla base di questo database l'algoritmo di routing determina ogni volta il cammino minimo tra il router e le possibili reti di destinazione e aggiorna così il next-hop nella tabella di routing.

Concetto di Prefisso

Il pacchetto è instradato sulla porta di uscita corrispondente al prefisso di lunghezza maggiore contenuto nell'indirizzo di destinazione (**longest prefix matching**)

È necessaria una fase di ricerca del prefisso: Algoritmi di lookup oppure Memorie particolari (TCAM)

Content Addressable Memory

Una CAM è una memoria specializzata per eseguire operazioni di matching in modo parallelo
Rispetto ad una memoria RAM, una CAM, oltre alle celle di memoria, contiene i circuiti di confronto per rilevare un match tra i bit memorizzati e i bit di input

Elevate prestazioni: oltre 100x106 lookup/sec, ma anche **Alto consumo:** circa 10-15 W per chip
Le CAM sono memorie speciali ottimizzate per le operazioni di confronto

Operazione di lettura in una memoria tradizionale (RAM)

Input: indirizzo di una locazione di memoria

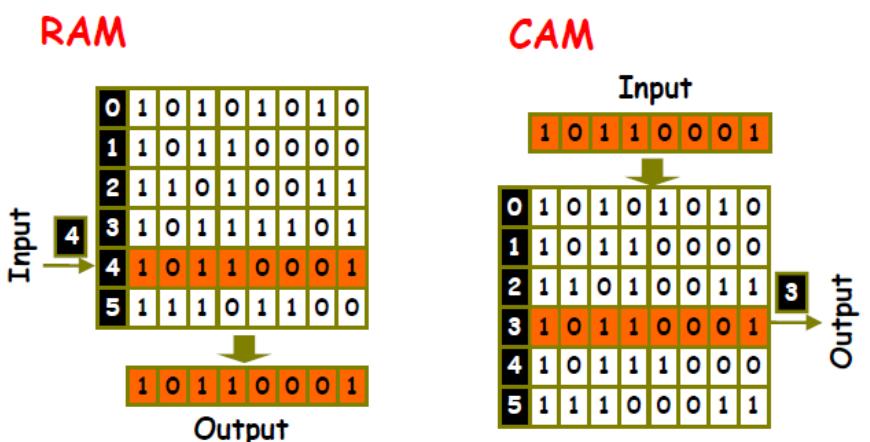
Output: contenuto della locazione di memoria

Nelle CAM l'operazione di lettura è inversa

Input: parola da confrontare

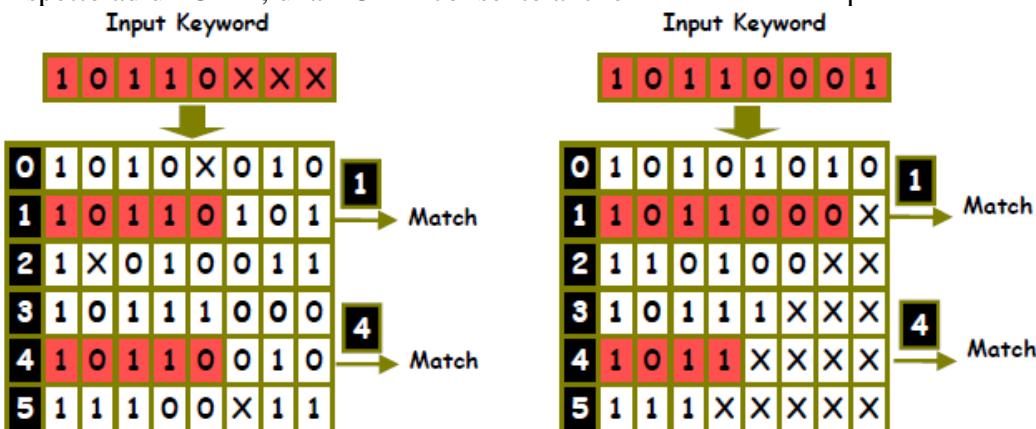
Output: indirizzo della locazione che contiene la parola

RAM vs CAM



Ternary Content Addressable Memory (TCAM)

Rispetto ad un CAM, una TCAM consente anche il confronto con presenza di “don’t care” (X)

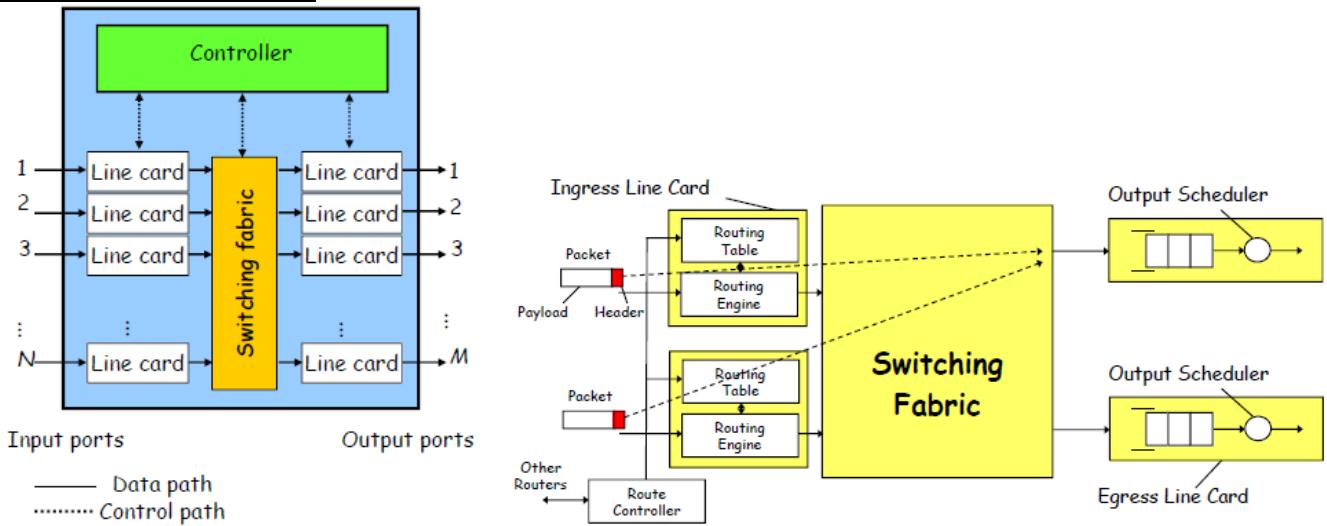


Longest Prefix Matching con TCAM

I prefissi sono ordinati in ordine di lunghezza crescente

Un indirizzo di destinazione viene confrontato con i prefissi in ordine crescente di lunghezza, il primo match indica l'indirizzo del record della Routing Table che contiene le informazioni di instradamento

Architettura del router

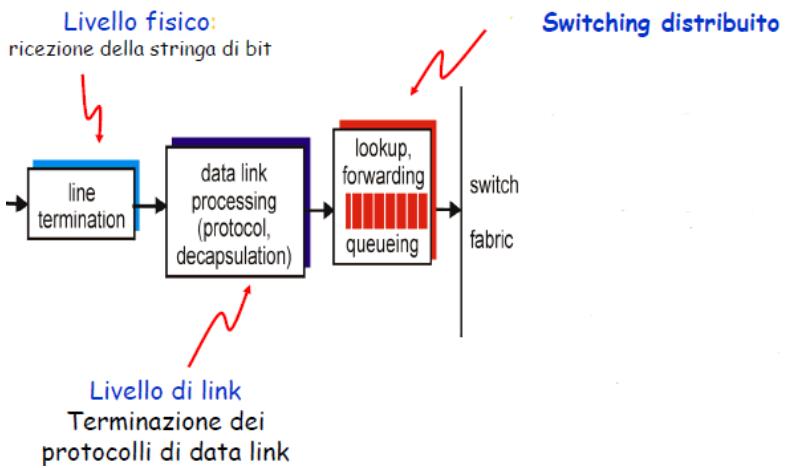


Controller: dove vengono bufferizzate le risorse se il tasso di arrivo è maggiore del tasso di inoltro
(Funzioni di controllo e di resource allocation)

Input Line Card Porte d'ingresso

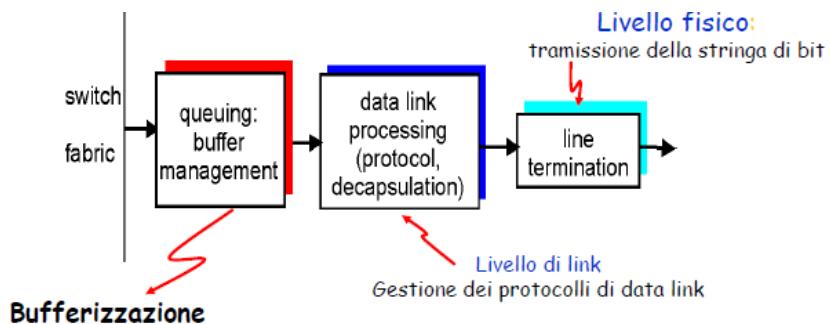
Determina la porta d'uscita dei pacchetti utilizzando le informazioni della tabella di routing
completando l'elaborazione allo stesso tasso della linea

Funzione di bufferizzazione se il tasso di arrivo dei pacchetti è superiore a quello di inoltro



Output Line Card

Porte di uscita che definiscono lo scheduling per la trasmissione dei pacchetti. Riceve i pacchetti, se sono troppi li bufferizza, e stabilisce in che ordine ritrasmetterli.



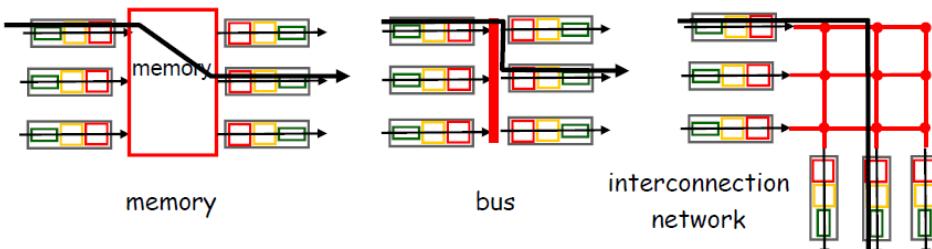
Switching Fabric

Funzione di instradamento tra porta di ingresso e di uscita

Switching rate: rate massimo al quale i pacchetti possono essere trasferiti nella switching fabric

Obiettivo: N porte di ingresso: switching rate N volte il rate di linea

Tre tipi di switching fabric:



Architettura memory-based

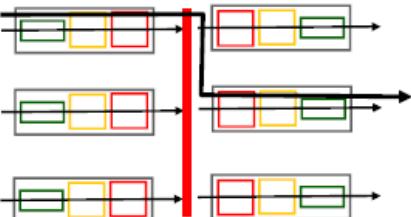


Prima generazione di router in cui i pacchetti vengono copiati nella memoria centrale e poi passato alla porta di uscita, con a disposizione operazioni di scrittura e di lettura

Switching rate limitato dalla velocità della memoria

Ogni pacchetto attraversa due volte il bus

Architetture a bus

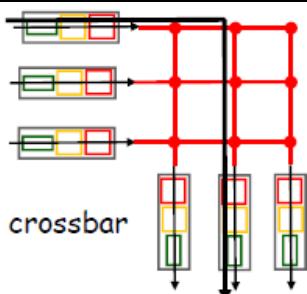


Le porte di ingresso sono collegate alle porte di uscita tramite un bus e trasferiscono il pacchetto da sole. Tutte le porte si contendono il bus.

Switching rate limitato dal data rate del bus

Adatta per router di piccole dimensioni

Architetture con interconnection network (crossbar)



Tanti bus per collegare ogni porta di ingresso a quella di uscita, architetture derivate dalle reti multiprocessore

Le unità dati trasferite all'interno della rete di interconnessione sono celle a lunghezza fissa

Necessità di segmentazione e ricostruzione dei pacchetti IP

Lo switching rate cresce con le dimensioni della rete

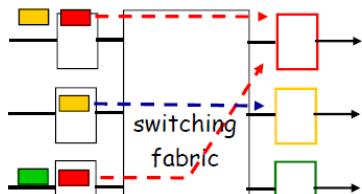
Adatta per router di grandi dimensioni

Input port queuing

La memorizzazione dei pacchetti in caso di congestione avviene in buffer posti nelle input card
Possibilità di ritardi e perdita di pacchetti

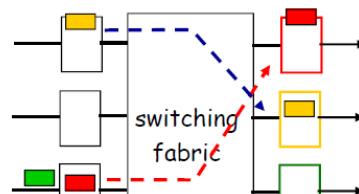
Problema: Head-of-the-Line (HOL) blocking, output port contention

Il primo pacchetto memorizzato nella coda in input può bloccare i pacchetti successivi



Output port contention

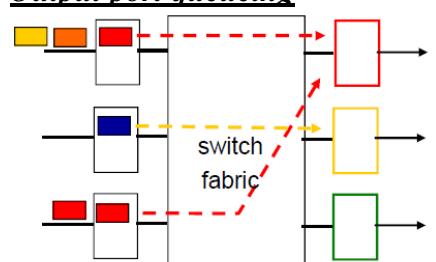
Solo uno dei due pacchetti rossi può essere trasferito, l'altro deve essere memorizzato



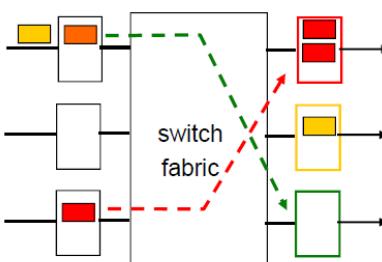
HOL blocking

il pacchetto verde subisce un blocco anche se la porta di uscita è libera

Output port queueing



Contention sulla linea di uscita



Memorizzazione di uno dei pacchetti rossi

I buffer sono collocati nelle card di output

Si ha bufferizzazione quando il tasso di arrivo dei pacchetti verso una singola card supera il rate di trasmissione sul link di uscita

Non esiste il problema dell'HOL

Dimensionamento dei buffer

Regola per bufferizzazione: La dimensione dei buffer B dipende dal prodotto RTT * C (**prodotto banda ritardo**)

Dove RTT è il Round Trip Time tipico del router e C è il bit rate del link di uscita

Se sono presenti N flussi si ha

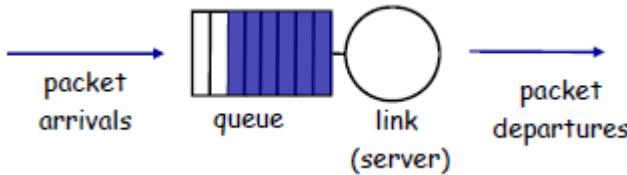
$$B = RTT \cdot \frac{C}{\sqrt{N}}$$

Politiche di scheduling: FIFO

La funzione di scheduling ha lo scopo di determinare il successivo pacchetto da trasmettere sulla linea di uscita: FIFO (first in first out)

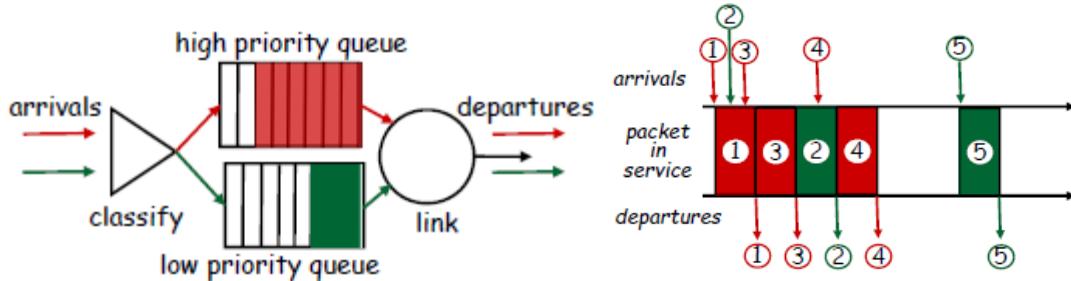
Politica di scarto dei pacchetti in caso di saturazione del buffer:

- **tail drop:** si scarta l'ultimo pacchetto arrivato
- **priority:** scarta i pacchetti secondo il loro livello di priorità
- **random:** si sceglie il pacchetto da scartare in modo casuale



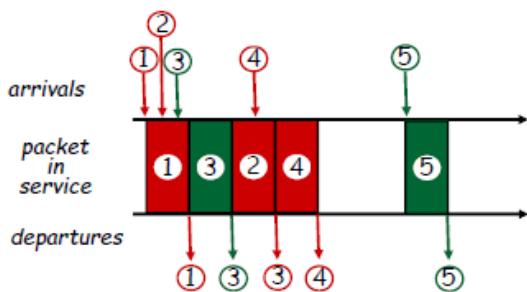
Politiche di scheduling: Priorità

Classi multiple con livelli di priorità diversi identificati attraverso una funzione di classificazione
Es. IP source/dest, port numbers, etc.



Round Robin (RR) scheduling

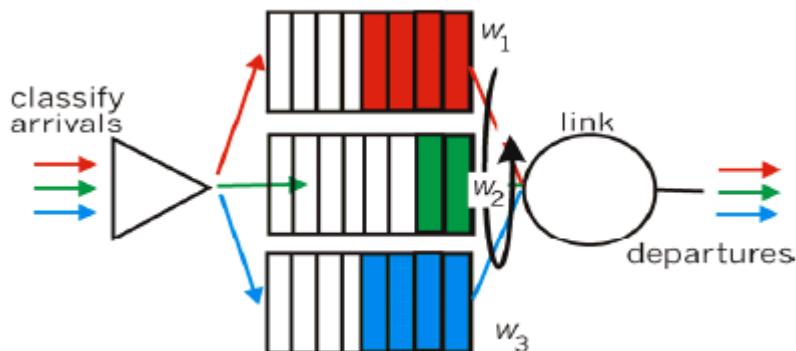
Gestione di classi multiple, ad ogni classe è associato un buffer logico diverso
Lo scheduler esamina ciclicamente le code ed emette, se esiste, il primo pacchetto di ciascuna coda



Weighted Fair Queuing (WFQ)

Generalizza il funzionamento dello scheduler RR

Ad ogni classe è associato un peso che corrisponde alla frequenza con cui viene esplorata la coda



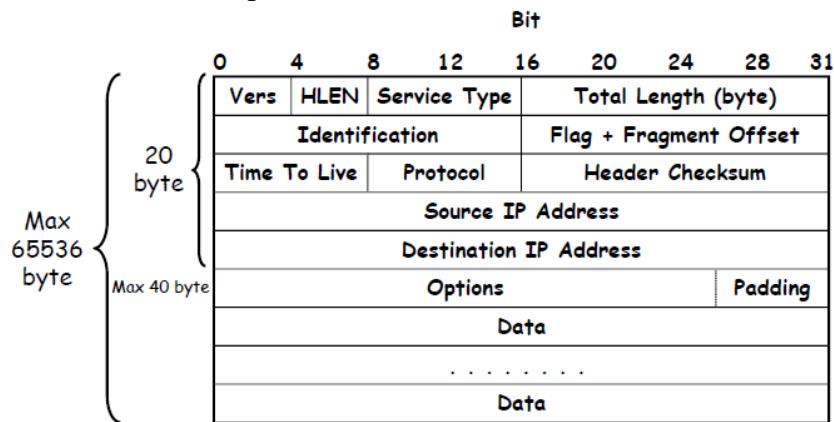
STRATO DI RETE (PARTE 2) CAPITOLO 6

Protocollo IP

Nella Rete abbiamo protocolli di instradamento (che servono al controllo e al riempimento delle tabelle di routing), le Routing Table, i Protocolli ICMP (gestiscono gli errori che IP non gestisce, controlla flusso del trasferimento e risolve eventuali situazioni anomale) ed infine il Protocollo IP. Esso opera con modalità di trasferimento **senza connessione** (servizio “best effort”) ed esegue le seguenti funzioni:

- Definisce il formato dei pacchetti, la lunghezza massima di un pacchetto è di 2^{16} byte = 65536 byte
- Definisce lo schema di indirizzamento
- Definisce le modalità di instradamento dei pacchetti
- Esegue, se necessario, frammentazione e ri-assemblaggio delle unità dati.

Ogni Rete fisica ha un valore massimo di lunghezza della propria unità informativa = **MTU (Max Transmission Unit)**. Il valore minimo di MTU è 68 byte. Se un pacchetto IP è più grande della MTU della sottorete allora serve frammentazione, fatta prima del rilancio nella sottorete che poi l'host destinazione ricomporrà.



Formato del pacchetto

Vers: 4 bit, versione del protocollo

HLEN (Header lenght): 4 bit, lunghezza dell'intestazione (specificata in parole di 32 bit), comprende la parte fissa (20 byte) e la parte opzionale, valore massimo: 60 byte

Total Length: 16 bit , lunghezza complessiva del pacchetto(specificata in byte), valore massimo 65536 byte (perchè con 16 bit al max posso rappresentare questo numero), comprende la lunghezza dell'header e del payload

Service Type: 8 bit, specifica i parametri di qualità del servizio richiesti dall'utente per il pacchetto

 - **Precedence**: 3 bit, indica il livello di priorità del pacchetto, usati per implementare i meccanismi DiffServ

Precedence	Delay	Thput	Reliab.	Cost	0
1	2	3	4	5	6 7 8

Type of Service(TOS): 4 bit, indica il tipo di servizio richiesto a seconda di quale bit è posto a 1.

(1000-minimize delay, 0100 max throughput, 0010 max realiabilty, 0001 minimize cost, 0000 normal)

Identification: 16 bit, numero identificativo del pacchetto da frammentare assegnato dal processo sorgente

Flags: 3 bit, X = posto a zero, DF (Don't Fragment) = 0 permessa frammentazione, = 1

frammentazione vietata, MF (More Fragment) = 0 ultimo frammento del pacchetto, = 1 non ultimo frammento

Fragment Offset: 13 bit, posizione del frammento all'interno del pacchetto (espresso in unità di 8 byte), per valutare l'integrità del pacchetto.

L'unità massima di trasmissione (MTU) è la massima quantità di dati che un frame a livello di collegamento può trasportare

Time to Live (TTL): 8bit, numero massimo di router che possono essere attraversati dal pacchetto. Inizializzato dall'host sorgente e decrementa ogni salto, se arriva a zero viene scartato e viene emesso un messaggio ICMP di notifica verso l'host sorgente

Protocol: 8bit, indica a quale protocollo di strato superiore deve essere trasferito il contenuto informativo del pacchetto (es. TCP = 6, UDP = 17, ICMP = 1)

Header Checksum: 16 bit protegge solo l'intestazione del pacchetto, se viene rivelato un errore il pacchetto viene scartato.

Source address(32bit) e Destination Address(32bit)

Options: lunghezza variabile a multipli di 8

- **Record Route Option (RRO):** lista vuota di indirizzi IP, ogni router inserisce il suo

- **TimeStamp Option:** come RRO ma con in più l'istante in cui il pacchetto attraversa ogni router

- **Loose Source Routing Option (LSRO):** specifica una lista di router che devono essere attraversati dal pacchetto

- **Strict Source Route Option (SSRO):** specifica tutti i router attraverso i quali deve transitare il pacchetto

Padding - rende l'intestazione multipla di 32 bit introducendo zeri

Protocollo ICMP (Internet Control Message Protocol)

Protocollo che è parte integrante di quello IP, encapsulato nella parte dati del pacchetto IP.

Obiettivo: Permettere ai router di inviare messaggi all'host sorgente in caso di anomalie nell'elaborazione di un pacchetto (errori di instradamento, TTL scaduto, congestione eccessiva). Non definisce che operazioni fare in caso di errore, **notifica solamente l'errore** e non vengono elaborati dai router intermedi.

Non vengono generati nuovi messaggi ICMP in seguito ad errori causati da pacchetti contenenti messaggi ICMP

Struttura: Per ogni pacchetto vi è un messaggio ICMP e nel caso di frammentazione il messaggio sarà solo nel frammento 0.

Byte			
0	8	16	24
Type	Code	Checksum	
ICMP data			

Esso è composto da:

- **Type:** 4bit, identifica il particolare tipo di messaggio a seconda dell'errore rilevato

0	Echo replay	11	Time exceeded
3	Dest. Unreachable	13	Time stamp request
4	Source Quench	14	Time stamp replay
5	Redirect	17	Address mask req.
8	Echo	18	Address mask rep.

- **Code:** 4bit, contiene il codice dell'errore

- **Data:** contiene una parte del pacchetto IP per poter risalire a quale pacchetto ha causato l'errore

- **Redirect Message:** se emesso da un router significa che i prossimi pacchetti devono essere trasmessi non più al router che manda il messaggio ma a quello specificato nell' ICMP, quindi si modifica la tabella di instradamento dell'host

- **Source Quench**: Se viene emesso da un router intermedio indica che il router non ha buffer sufficiente per memorizzare il pacchetto, se invece viene emesso dall'host di destinazione allora il pacchetto non è stato processato
- **Time exceeded**: TTL si è esaurito
- **Echo & Echo Replay**: usati per stabilire l'attività di un elemento di un host
- **Destination Unreachable**: indica che l'instradamento di un pacchetto non è stato completato
- **Time Stamp Request & Time Stamp Replay**: utilizzati per effettuare misure di prestazioni (es. ritardi di transito)
- **Address mask Request & Address mask Replay**: utilizzati per determinare la maschera della sottorete a cui è connesso un host, vengono usati da host molto semplici (diskless) dopo aver individuato il proprio indirizzo con il protocollo RARP

PING

Usato per vedere se la macchina con cui vogliamo comunicare è attiva ed il tempo di transito tra host sorgente e host destinazione (utilizza i messaggi ICMP Echo e Echo Replay)

TraceRoute

Protocollo utilizzato per capire che strada farà il pacchetto e quindi che nodi attraversa. Ogni volta che il TTL scade infatti il router manda un messaggio all'host sorgente dicendo "Time Exceeded e il nome del router che lo manda".

TraceRoute forza quindi l'host a mandare una serie di pacchetti partendo dal TTL=1 e per poi incrementarlo, di modo che per ogni pacchetto individua il nodo successivo ed il tempo dalla sorgente (RTT) a quel router.

Traceroute fa questo procedimento per tre volte, ma i tempi potrebbero anche esser diversi ogni volta.

Criteri di arresto dell'invio

Quando un segmento UDP arriva all'host di destinazione, quest'ultimo restituisce un messaggio ICMP di porta non raggiungibile (tipo 3, codice 3).

Quando l'origine riceve questo messaggio ICMP, si blocca

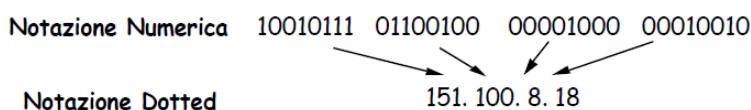
Indirizzamento in Ipv4

L'indirizzo IP identifica un'interfaccia di Rete. Esso ha una lunghezza di 32 bit ed è unico nella rete.

Se un host è connesso a più di una rete (multihomed) avrà un indirizzo IP per ogni interfaccia

Un router ha tanti indirizzi IP quanto sono le interfacce di rete che gestisce

Noi convertiamo i 4 byte da binario (notazione numerica) a decimale (notazione dotted) e quindi per ognuna delle quattro parti potrò mettere un numero da 0 a 257.



Sottorete

Rete isolata composta da terminali collegati all'interfaccia di un host o router (anche detta rete IP).

Anche un link diretto tra due router è considerato una sottorete.

Struttura Indirizzo IP

Composto da due parti:

- **Net_id** = identificativo della sottorete (**prefisso**)
- **Host_id** = identificativo dell'host all'interno della sottorete.

Schema Indirizzamento Classfull

In origine le sottoreti erano divise in classi, gli indirizzi dovevano essere univoci e così a seconda di quanti host avevi bisogno ti davano un certo intervallo di indirizzi di una certa classe. Le classi erano contraddistinte dai bit iniziali dell'indirizzo e un numero di bit per il net_id fisso.

Classi di Indirizzi IP

	0	8	16	24	31
Classe A	0	Net_Id		Host_Id	
Classe B	1 0	Net_id		Host_Id	
Classe C	1 1 0	Net_Id		Host_Id	
Classe D	1 1 1 0		Multicast Address		
Classe E	1 1 1 1 0		Reserved		

Convenzioni Indirizzi Speciali:

Se un host si muove dalla rete in cui si trova, il suo indirizzo deve essere cambiato

- **Indirizzo Host in fase di boot** – tutti zeri
- **Host nella rete locale** – tutti zeri (Net_Id) e poi Host_Id
- **BroadCast su rete locale** – tutti uni
- **Broadcast sulla rete Net_Id** – net_id e tutti 1(Host_Id)

Si è cominciato ad avere un problema quando gli utenti avevano un numero di indirizzi IP enorme ma non li usavano, ed essendo univoci non si potevano sprecare.

Nasce così il **Subnetting** in cui gli utenti prendevano un gruppo di indirizzi e li dividevano in due creando un terzo livello gerarchico, la **Subnet**.

Si utilizzano alcuni bit dell'Host_Id per codificare il Subnet_Id

Original address	1 0	Net ID	Host ID
------------------	-------	--------	---------

Subnetted address	1 0	Net ID	Subnet ID	Host ID
-------------------	-------	--------	-----------	---------

I campi Net_Id e Subnet_Id vengono identificati da una maschera denominata **Subnet Mask** che corrisponde a una parola di 32bit in cui i bit uguali a 1 identificano i bit del Net_Id e del Subnet_Id mentre quelli uguali a 0 identificano i bit dell'Host_Id

La Subnet_Id ha significato solo nel router a cui sono connesse le sottoreti

IP address	1 0	Net ID	Subnet ID	Host ID
------------	-------	--------	-----------	---------

Subnet Mask	1 1 1 1 1 1 1 1	...	1 1 1 1 0 0 0 0 0 0 0 0
-------------	-------------------	-----	---------------------------

Esistono due tipi di subnet:

Subnetting Statico in cui tutte le subnet hanno la stessa maschera e una lunghezza fissa.

	0	Net_id	Host_id	
Subnet Mask	1	8	16	24
	255	255	255	192
	11111111111111111111111111111111000000			

numero massimo di sottoreti possibili = $2^{18} = 262.142$

numero massimo di host per sottorete = $2^6 - 2 = 62$

Subnetting a lunghezza variabile: permette di gestire diverse sottoreti di dimensione diversa.

Vedere esempio slide 6_strato_di_rete_parte_2 pagina 40

Per trovare dall'indirizzo di una sottorete la sua subnet si converte l'indirizzo della sottorete in binario, si scrive la sua subnet = tanti zeri quanti bit riservati alla subnet e poi si fa l'operazione AND. Il risultato convertito in decimale è l'indirizzo subnet.

Routing in reti IP

Sia gli host che i router hanno una **Routing Table**. Se l'host deve inviare ad un host nella sottorete, invia il pacchetto direttamente tramite l'interfaccia di rete (la frame in cui viene encapsulato il pacchetto conterrà l'indirizzo MAC della destinazione). Se invece invia fuori dalla rete, invia al **default router** (la frame in cui viene encapsulato il pacchetto conterrà l'indirizzo MAC del router)

Il Router esamina l'indirizzo IP di destinazione nel pacchetto entrante, se la destinazione è su una delle reti a cui è connesso il router, il pacchetto viene inviato direttamente usando l'interfaccia di rete, altrimenti accede alla routing table per verificare il next-hop per quel pacchetto.

Tutto questo avviene con la tecnica del **Longest Prefix Matching**, ovvero il router sceglie la direzione corrispondente al prefisso di lunghezza maggiore.

Instradamento

- indirizzo 198.15.7.3
- indirizzo 198.15.7.4

198.15.7.3

- porta 1: matching prefisso 16
- porta 7: matching prefisso 24
- porta 4: matching prefisso 32

198.15.7.4

- porta 1: matching prefisso 16
- porta 7: matching prefisso 24
- porta 4: no matching

Tabella di instradamento

Prefix	Porta d'uscita
198.15.0.0/16	1
198.15.7.0/24	7
198.15.7.3/32	4

198.15.7.3 \Rightarrow porta 4

198.15.7.4 \Rightarrow porta 7

Routing Table

Contiene gli IP di destinazione + next hop associato + identificatore della porta di uscita + informazioni statistiche

Vedere esempio slide 6_strato_di_rete_parte_2 pagina 46

Schema Indirizzamento ClassLess (CIDR , Classless Inter Domain Routing)

Nasce dal fatto che gli indirizzi erano in via di esaurimento e più aumentavano, più le tabelle di routing crescevano in dimensione.

La struttura Classfull era inefficiente: gli indirizzi di Classe B erano troppo grandi per la maggior parte delle organizzazioni e gli indirizzi di Classe C troppo piccoli

Il **CIDR** rappresentava una soluzione short-term al problema, mentre la soluzione a long-term era l'aumento dello spazio di indirizzamento (IPv6, indirizzi a 128 bit)

CIDR è stato ideato per rendere più efficiente l'impiego dello spazio di indirizzamento di IP e diminuire la complessità delle tabelle di instradamento nei router

Ad una rete viene assegnato un certo numero di blocchi contigui di indirizzi (**Supernetting**): la rete sarà caratterizzata da un unico **prefisso** (insieme dei bit più significativi) potendo essere individuata nei router solo dal suo prefisso

Address Allocation Policy

Indirizzi di Classe A e B sono assegnati solo in caso di dimostrata necessità

Sono assegnati blocchi consecutivi di classe C (fino a 64 blocchi): tutti gli indirizzi IP hanno un prefisso comune la cui lunghezza può essere arbitraria

La metà inferiore degli indirizzi di classe C è assegnata su base geografica

Tutte le reti appartenenti ad una regione geografica sono identificate dagli stessi 7 bit di prefisso

Esempio: Europa da 194 (1100001 00) a 195 = (1100001 11)

Multiregional	192.0.0	193.255.255
Europe	194.0.0	195.255.255
Others	196.0.0	197.255.255
North America	198.0.0	199.255.255
Central/South America	200.0.0	201.255.255
Pacific Rim	202.0.0	203.255.255
Others	204.0.0	205.255.255
Others	206.0.0	207.255.255

Supernetting

Vedere esempi slide 6_strato_di_rete_parte_2 pagine 55-56-57

CIDR Allocation Principles (RFC 1518-1520)

L'assegnazione degli indirizzi IP riflette la topologia fisica della rete che segue i confini continentali e nazionali (Gli indirizzi IP devono essere assegnati su questa base)

I domini di transito (TRD) hanno un prefisso IP unico:

- Trasportano traffico tra domini terminali
- La maggior parte dei domini terminali sono single-homed: connessi ad un solo TRD
- A tali domini sono assegnati indirizzi con lo stesso prefisso del TRD
- Tutte le reti connesse ad un TRD sono aggregate in un solo entry delle tabelle di routing

STRATO DI RETE (PARTE 3) CAPITOLO 7

DHCP

Un host deve essere configurato con indirizzo IP, Subnet mask, Default router e Server DNS

DHCP (**Dynamic Host Configuration Protocol**) auto compila i suddetti campi.

Consente ad un host di ottenere dinamicamente il suo indirizzo IP dal server di rete: È possibile rinnovare la proprietà dell'indirizzo in uso, è possibile il riuso degli indirizzi e supporta anche gli utenti mobili che si vogliono unire alla rete

Panoramica del DHCP

- L'host invia un messaggio broadcast “**DHCP discover**”: È emesso in modo broadcast da un client per trovare un DHCP server
- Il server DHCP invia l'indirizzo con il messaggio “**DHCP offer**” in risposta al DHCP discover
- L'host richiede la configurazione con il messaggio “**DHCP request**”
- Il server DHCP invia la configurazione richiesta con il messaggio “**DHCP ack**”

Supporta tre meccanismi per la gestione degli indirizzi IP:

Allocazione automatica: DHCP assegna permanentemente un indirizzo IP

Allocazione dinamica: DHCP assegna un indirizzo IP per un intervallo limitato di tempo (lease)

Allocazione manuale: L'indirizzo IP è assegnato dall'amministratore di rete

Struttura

code	HW type	length	hops
Transaction ID			
Seconds	Flags field		
Client IP address			
Your IP address			
Server IP address			
Router IP address			
Client HW address (16 bytes)			
Server host name (64 bytes)			
Boot file name (128 bytes)			
Options 312 bytes)			

Code: Indica una richiesta o una risposta

HW type: Tipo di hardware (es. ethernet, IEEE 802)

Length: Lunghezza del campo client HW address

Transaction ID: Pacchetti di richiesta e di risposta hanno lo stesso numero

Seconds: Indica il tempo trascorso dall'avvio della procedura di boot

Flag: Indica se il pacchetto è unicast o broadcast

Client IP address: È settato dal client, se il client non conosce il proprio indirizzo il suo valore è 0.0.0.0

Your IP address: Indirizzo IP del client assegnato dal server

Server IP address: Indirizzo IP del server

Client HW address: Indirizzo MAC del client

Options: Parametri di configurazione addizionali: router di default, subnet mask, domain name server...

Messaggi DHCP

DHCP_Discover: È emesso in modo broadcast da un client per trovare un DHCP server

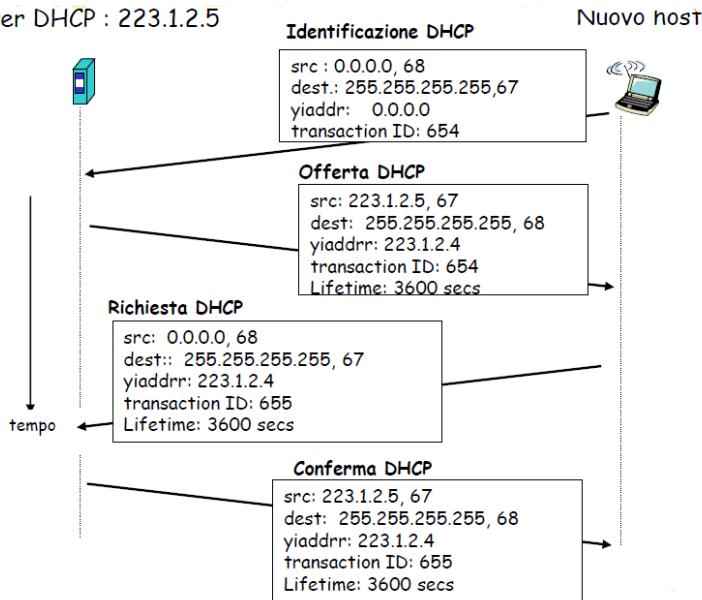
DHCP_Offer: È la risposta di un DHCP server ad un messaggio DHCP_Discover e assegna l'indirizzo IP richiesto

DHCP_Request: È emesso da un DHCP client verso un server, richiede i parametri di configurazione ad un server e rifiuta le offerte degli altri in caso di ricezione multipla di messaggi DHCP Offer
Verifica la consistenza della propria configurazione in caso di cambio di rete o di sistema (es. reboot)
Richiede l'estensione temporale dell'uso di un indirizzo (lease extension)

DHCP_Ack: Riscontro inviato dal DHCP Server al client ad un DHCP_request
Contiene la configurazione richiesta dal client

Procedura DHCP

server DHCP : 223.1.2.5



Pro

Semplifica la gestione amministrativa degli indirizzi in rete e l'accesso in rete di utenti in mobilità (Nomadic Computing)

Rende possibile l'uso efficiente di un insieme di indirizzi IP dimensionando opportunamente il tempo di lease

Contro

Non garantisce un vero e proprio "plug and play":

Deve essere previsto un server DHCP in rete e gli host devono essere configurati per usare DHCP

DHCP non è sicuro, un utente non autorizzato può accedere alla rete

Problemi di interoperabilità con DNS in caso di riallocazione dinamica degli indirizzi

NAT (Network Address Translator)

Riduce l'utilizzazione dello spazio di indirizzi IP

È utilizzato in una Intranet (ISP locale) in cui vengono assegnati un insieme di indirizzi IP pubblici che sono visibili dalle reti esterne

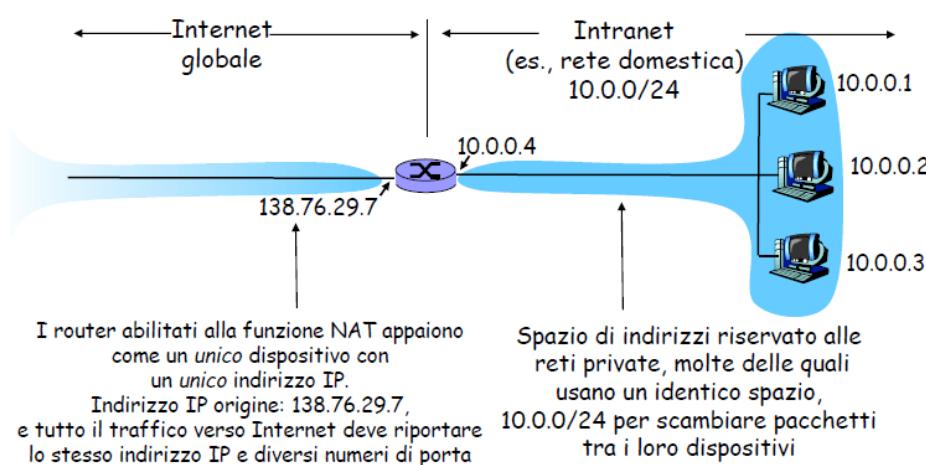
All'interno della Intranet possono essere utilizzati liberamente indirizzi IP **privati**, anche non unici in rete, appartenenti alle seguenti classi:

Indirizzi di classe A: 10.0.0.0

Indirizzi di classe B: da 172.16.0.0 a 172.31.0.0

Indirizzi di classe C: da 192.68.0.0 a 192.168.255.0

Il dispositivo NAT assegna un indirizzo pubblico ad un host solo nel momento in cui questo deve comunicare con l'esterno eseguendo la conversione dell'indirizzo privato con un indirizzo pubblico. Un NAT nasconde i dettagli di una Intranet al mondo esterno infatti è possibile cambiare gli indirizzi delle macchine di una rete privata senza doverlo comunicare all'Internet globale



Quando un router NAT riceve il pacchetto dalla rete locale (source address IP privato, source port number qualsiasi):

- Genera un nuovo numero di porta d'origine (es. 5001)
- Sostituisce l'indirizzo IP di sorgente (privato) con il proprio indirizzo IP (pubblico) sul lato WAN (es. 138.76.29.7)
- Sostituisce il numero di porta origine iniziale (es. 3348) con il nuovo numero (5001)

Quando un router NAT riceve il pacchetto da Internet (destination IP pubblico del router, destination port number fissato dal router):

- Legge il numero di porta (es. 5001) ed individua il mapping con l'indirizzo interno
- Sostituisce l'indirizzo IP di destinazione con l'indirizzo IP privato dell'host di destinazione
- Sostituisce il numero di porta di destinazione (5001) con il numero di porta iniziale (3348)

Traduzione degli indirizzi di rete

Il campo numero di porta è lungo 16 bit e il protocollo NAT può supportare più di 60.000 connessioni simultanee con un solo indirizzo IP sul lato WAN

L'uso di un NAT è controverso:

È contrario ai principi dell'architettura a strati TCP/IP: i dispositivi di rete dovrebbero elaborare i pacchetti esclusivamente fino allo strato 3

Elimina la trasparenza della rete: un host non è visibile dall'esterno della rete a cui appartiene
Interferisce con le applicazioni P2P

Al momento di un cambio di indirizzo IP deve essere ricalcolato il checksum dei pacchetti UDP e TCP
Incompatibilità con il protocollo ICMP

Difficoltà di una connessione tra un host ed un server dietro ad un NAT (Connection reversal)

STRATO DI RETE (PARTE 4): IPv6 CAPITOLO 8

Funzionalità

Aumento dello spazio di indirizzamento: Indirizzi a 128 bit con indirizzamento gerarchico basato sul concetto di prefisso

Semplificazione della struttura dell'header dei pacchetti: Header di lunghezza fissa, diversa modalità di codifica del campo “Options” ed eliminazione dei campi checksum e quelli dedicati alla frammentazione

Possibilità di identificazione dei flussi dei pacchetti (Flow label)

Meccanismo integrato di autoconfigurazione delle interfacce di rete

Le funzioni e il formato dei pacchetti IPv6 sono specificate nei seguenti RFC

RFC 2460: Internet Protocol, Version 6 (IPv6) Specification

RFC 3513: IP version 6 Addressing Architecture

RFC 3587: IPv6 Global Unicast Address Format

RFC 3177: IAB/IESG Recommendations on IPv6 Address Allocations to Sites

RFC 2461: Neighbor Discovery for IP version 6 (IPv6)

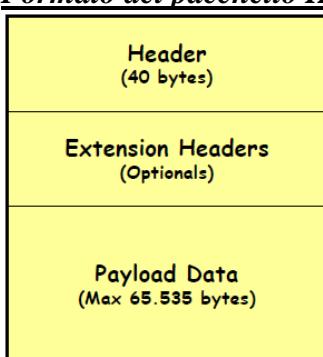
RFC 2462: IPv6 stateless address autoconfiguration

RFC 2893: Transition mechanism for IPv6 hosts and routers

RFC 3056: Connection of IPv6 domains via IPv4 clouds

RFC 3315: Dynamic Host Configuration Protocol for IPv6 (DHCPv6)

Formato del pacchetto IPv6



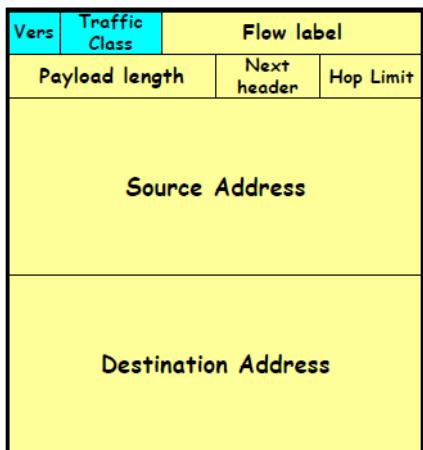
Header: contiene le informazioni comuni a tutti i pacchetti

Extension Headers: contengono le opzioni utilizzate dai router intermedi e/o dall'host di destinazione

Payload Data: sono i bit informativi elaborati dall'host di destinazione

IPv6 Datagram Basic Header

0 4 8 12 16 20 24 28 31



- **Version (4 bit)**: versione del protocollo, è possibile la coesistenza di più versioni di IP
- **Traffic Class (8 bit)**: stabilisce la classe di servizio e la priorità del pacchetto
È compatibile con la specifica del campo DSCP dell'architettura DiffServ
- **Flow label (20 bit)**: identifica, insieme al campo source address, un particolare flusso di pacchetti
Consente di instradare i pacchetti in hardware
Consente l'applicazione delle procedure di riservazione di risorse per traffico con qualità di servizio garantita
- **Payload Length (16 bit)**: lunghezza in byte del pacchetto IP (escluso header)
Normalmente la lunghezza massima del payload è 65.535 byte
Se la lunghezza del pacchetto è maggiore di 64 K il suo valore è "0", l'opzione "jumbo payload" indica la lunghezza effettiva
- **Next Header (8 bit)**: Contiene il codice che identifica l'header che segue nel pacchetto

0	Hop-by-hop options header
4	Internet Protocol (IP)
6	Transmission Control Protocol (TCP)
17	User Datagram Protocol (UDP)
43	Routing
44	Fragment Header
50	Encapsulating Security Payload
51	Authentication Header
58	Internet Control Message Protocol
59	No Next Header
60	Destination Options Header

- Hop Limit (8 bit)**: numero massimo di tratte di rete che il pacchetto può attraversare, ogni router decrementa di una unità tale campo, se il contatore si azzera prima che la destinazione sia raggiunta, il datagramma viene scartato
Evita gli effetti di eventuali loop in rete e può essere utilizzato per effettuare delle ricerche di host in rete a distanza prefissata
- Source e Destination Address (128 bit)**: indicano gli indirizzi IP degli host sorgente e di destinazione

IPv6 Extension Headers

Sono utilizzati per inviare informazioni opzionali alla destinazione o ai sistemi intermedi

Sono definiti 6 tipi di Extension Headers (EH):

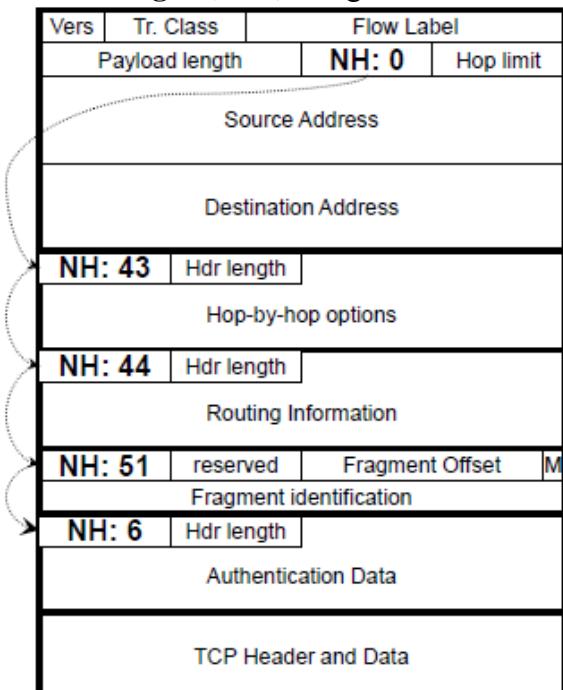
Next Header	Extension Header
0	Hop-by-hop options header
43	Routing Header
44	Fragment Header
51	Authentication Header
50	Encapsulation Security Payload Header
60	Destination Options Header

Un pacchetto può trasportare un numero qualsiasi di EH poiché ciascun EH è identificato dal valore del campo Next Header dell'header che lo precede.

Tutti gli EH tranne l'hop by hop sono elaborati dal nodo identificato dal destination address

Il formato generale di un EH prevede:

- **Next Header (8 bit):** identifica l'EH successivo
- **Hdr Length (8 bit):** lunghezza dell'EH esclusi i primi 8 ottetti



L'ordine di inserimento degli EH nel pacchetto è il seguente:

- 1) IPv6 header
- 2) Hop by hop options header
- 3) Destination options header: opzioni che devono essere elaborate dal nodo che appare nel campo Destination Address e, successivamente, dalle seguenti destinazioni indicate nel Routing header
- 4) Routing header
- 5) Fragment header
- 6) Authentication header
- 7) Encapsulating Security Payload header
- 8) Destination Options header: opzioni che devono essere elaborate dal nodo destinazione finale
- 9) Upper layer header

Hop by Hop Options Header

Contiene informazioni che devono essere elaborate da ogni sistema intermedio sul percorso del pacchetto

È identificato dal codice Next Header = 0 ed è costituito dai seguenti campi:

- **Type (8 bit)**: indica il tipo di opzione
- **Length (8 bit)**: indica la lunghezza del campo Data
- **Data (multiplo di 64 bit)**: trasporta il valore dell'opzione e alcune indicazioni per il router utili per l'elaborazione dell'opzione



I primi due bit del campo Type indicano la reazione che un router deve avere nel caso non riconosca l'opzione

Type	Action
00xxxxxx	Ignora l'opzione e elabora ugualmente il datagramma
01xxxxxx	Scarta il datagramma
10xxxxxx	Scarta il datagramma ed invia un messaggio ICMP
11xxxxxx	Scarta il datagramma ed invia un messaggio ICMP solo se la destinazione non è multicast

Il terzo bit stabilisce se il campo “data” può essere modificato

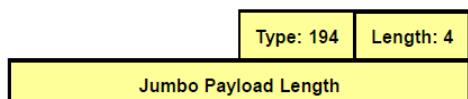
Type	Action
xx0xxxxx	Il campo “data” non deve essere modificato
xx1xxxxx	Il campo “data” può essere modificato

Jumbo Payload Length Option

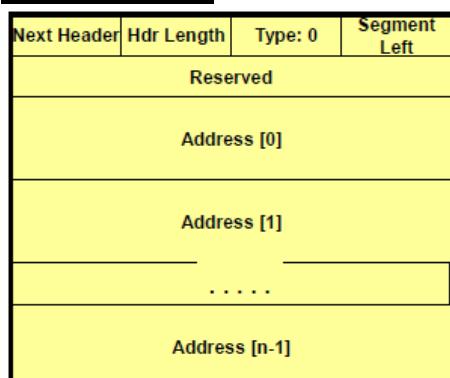
È individuata dal valore 194 del campo Type e serve ad aumentare la lunghezza massima del pacchetto rispetto a quanto consentito dall'header principale

Nel caso tale opzione sia utilizzata, il campo payload length del basic header deve contenere il valore 0
Il campo data ha lunghezza 32 bit

La lunghezza massima del pacchetto è quindi $2^{32} - 1$ byte



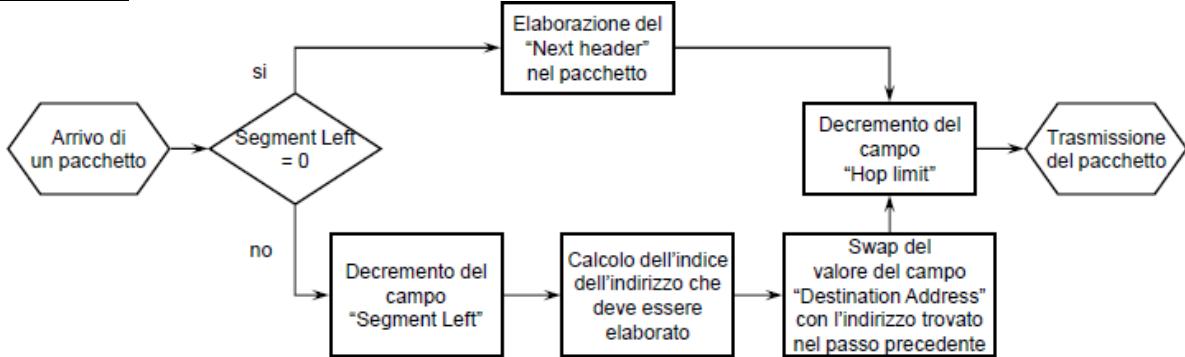
Routing Header



Permette l'instradamento di un pacchetto su un cammino predefinito
Fornisce ai router indicazioni per l'instradamento del datagramma

Segment left (8 bit): indica il numero di indirizzi che devono essere ancora elaborati, ogni router decremente tale campo, se il campo ha valore “0” il router ignora l'intero “extension header”

Algoritmo



Indirizzamento IPv6

Dimensione 128 bit (16 bytes): Lo spazio equivale a circa $340 \cdot 10^{36}$ indirizzi

Utilizzato con la stessa efficienza dello spazio degli indirizzi IPv4 consente di gestire circa 50.000 indirizzi per metro quadro

Un indirizzo è rappresentato da otto numeri esadecimali (ogni numero equivale a 16 bit) divisi dal simbolo “.” FE80:0000:0000:0001:0800:23E7:F5DB

Si possono omettere i gruppi di “0” iniziali in ogni numero: FE80:0:0:0:1:800:23E7:F5DB

Si possono omettere completamente un gruppo (uno solo) di numeri consecutivi di valore “0” sostituendoli con il simbolo “::” : FE80::1:800:23E7:F5DB

Sono definiti tre tipi di indirizzi:

- **Unicast:** Identifica una singola interfaccia, un pacchetto che reca un indirizzo di questo tipo è consegnato esclusivamente a quella interfaccia
- **Anycast:** Identifica un insieme di interfacce, un pacchetto che reca un indirizzo di questo tipo è consegnato ad una delle interfacce identificate dall'indirizzo (normalmente la più vicina)
- **Multicast:** Identifica un insieme di interfacce, un pacchetto che reca un indirizzo di questo tipo è consegnato a tutte le interfacce identificate dall'indirizzo

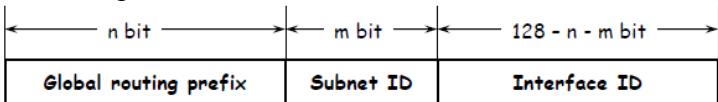
Lo spazio degli indirizzi è organizzato ad albero mediante prefissi IPv6 address / prefix length

Tipi di indirizzi IPv6

Tipo di indirizzo	Prefisso (binario)	Notazione IPv6
Non specificato	000...0 (128 bit)	::/128
Loopback	000...1 (128 bit)	::1/128
Multicast	11111111	FF00::/8
Link-local Unicast	1111111010	FE80::/10
Site-local Unicast	1111111011	FEC0::/10
Global Unicast	Qualsiasi altro	

Indirizzi Global Unicast

Formato generale:



- **Global Routing Prefix (n bit) (normalmente n = 48)**: Identifica un sito complesso (un cluster di reti)
- **Subnet ID (m bit) (normalmente m = 16)**: Identifica una specifica sottorete all'interno di un sito
- **Interface ID (128-n-m bit) (normalmente 64 bit)**: Identifica un'interfaccia fisica in una subnet, normalmente equivale all'indirizzo fisico dell'interfaccia (Indirizzo MAC a 64 bit)

Indirizzi Speciali Unicast

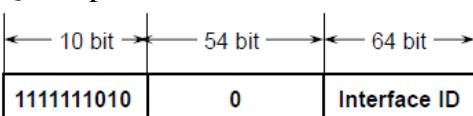
L'indirizzo "Non Specificato" (0:0:0:0:0:0:0) indica l'assenza di un indirizzo, non può essere assegnato ad un interfaccia fisica, ma viene ad esempio utilizzato come source address nei pacchetti di richiesta di indirizzo nelle procedure di inizializzazione automatica degli host (Non può essere usato come Destination Address)

L'indirizzo "Loopback" (0:0:0:0:0:0:1) è utilizzato da un nodo per inviare un pacchetto a se stesso
Non può essere assegnato ad un'interfaccia fisica e non può essere usato come Source Address

Indirizzi Link-local

Prefisso: FE80::/10, possono essere usati solo sulla rete fisica alla quale è connessa l'interfaccia dell'host e servono a individuare gli host su un link

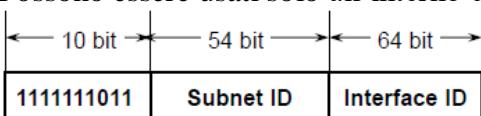
Questi pacchetti non devono essere rilanciati da un router



Indirizzi Site-local

Prefisso: FEC0::/10, equivalgono agli indirizzi privati IPv4 (es. 10.0.0.0)

Possono essere usati solo all'interno di una Intranet



Indirizzi Anycast

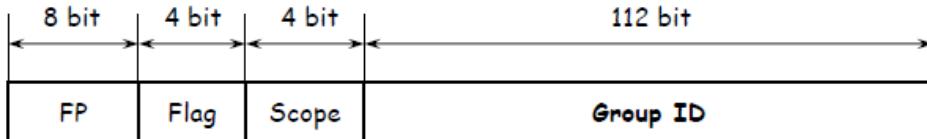
Un indirizzo Anycast Identifica un insieme di interfacce: un pacchetto inviato ad un indirizzo di questo tipo è instradato verso l'interfaccia più vicina appartenente all'insieme

Gli indirizzi anycast sono ricavati all'interno dello spazio degli indirizzi unicast: sono indistinguibili dagli indirizzi unicast e i router devono essere configurati in modo da riconoscere tali indirizzi

Esempio:

Possono essere utilizzati per identificare un insieme di router che appartengono ad uno stesso provider
Tale indirizzo può essere posto nel Routing extension header di un pacchetto IPv6 affinché raggiunga il provider

Indirizzi Multicast



Format Prefix (FP) (8 bit): 1111 1111

Flags (4 bit):

- 0000: indirizzo permanente
- 0001: indirizzo transitorio

Group ID (112 bit): Identifica il gruppo multicast

Scope (4 bit): Individua l'ambito di validità dell'indirizzo ovvero l'ambito in cui sono compresi i nodi appartenenti al gruppo multicast

Esempi

- 1 solo interfacce di un nodo
- 2 solo nodi del link locale (link scope)
- 5 solo nodi del sito locale (site scope)

Regole di allocazione degli indirizzi

Un indirizzo IPv6 può essere modellato come diviso in due campi:

- **Network number (64 bit)**
- **Host number (64 bit)**

Il problema è quello di individuare dei criteri per individuare il corretto spazio di indirizzi IPv6 da assegnare ad un singolo sito:

- Rete di un provider
- Rete privata (intranet)
- Postazioni d'utente (domestica, mobile, ecc.)
- Singolo host

Le regole adottate sono le seguenti:

- Prefisso /48 (16 bit di indirizzamento per le sottoreti) per un qualsiasi provider tranne quelli di grandi dimensioni
- Prefisso /64 nel caso in cui il sito è riconosciuto gestire solo una singola sottorete
- Prefisso /128 per un singolo dispositivo

Ad un provider di grandi dimensioni può essere assegnato uno spazio delimitato da un prefisso più breve di 48 bit

Osservazioni

Partizione dello spazio quasi statica

Tutte le reti hanno lo stesso spazio di indirizzamento (16 bit di identificativo di sottorete e 64 bit di identificatore di host)

Una rete mobile (veicolo o un terminale con interfacce multiple) ha la possibilità di gestire una molteplicità di dispositivi terminali

Un PC che si connette alla rete riceve un singolo indirizzo

IPv6: Autoconfigurazione

Il processo di autoconfigurazione permette ad un host di creare il proprio link-local address e verificarne l'univocità e determinare la sottorete a cui appartiene e quindi il proprio prefisso

Stateless autoconfiguration: Non è necessaria la presenza di un server e l'indirizzo di un host viene individuato sulla base di due informazioni:

- **Identificatore di interfaccia** (disponibile localmente)
- **Identificatore di sottorete** (comunicato da un router)

Non permette di controllare l'assegnazione degli indirizzi in un sito

Stateful autoconfiguration: L'host riceve le informazioni di configurazione da un server e permette di controllare strettamente il processo di assegnazione degli indirizzi

Ogni indirizzo è caratterizzato da un tempo di validità (lifetime):

- **Infinito** (assegnazione permanente)
- **Finito** (assegnazione dinamica)

Se il tempo di validità scade, l'associazione tra indirizzo e host non è più valida e l'indirizzo può essere riassegnato ad un altro host

L'unicità di un indirizzo è garantita da un algoritmo di rivelazione di indirizzi duplicati (**Duplicated Address Detection Algorithm**): l'algoritmo viene eseguito in entrambe le procedure stateless o stateful prima di utilizzare l'indirizzo assegnato

Stateless Autoconfiguration

Permette ad un host di ottenere un indirizzo unico per ognuna delle sue interfacce, si suppone che ogni interfaccia sia caratterizzata da un identificatore predefinito e unico

Evita la presenza di un server di configurazione in siti di piccola e media dimensione: un host può determinare automaticamente il prefisso associato alla rete a cui è connesso

Facilita l'eventuale ri-numerazione della rete in caso di cambio di provider

Procedura

- 1) Inizia al momento dello startup di un host
- 2) Viene generato il link-local address per l'interfaccia (prefisso FE80::/10)
- 3) Viene inviato sulla rete un messaggio di "**Neighbor Solicitation**" contenente il nuovo indirizzo per verificarne l'unicità: se un nodo risponde negativamente la procedura di autoconfigurazione è bloccata
- 4) Viene inviato un messaggio "**Router Solicitation**" per ottenere dal router l'indicazione del prefisso di rete per la formazione degli indirizzi site local address e global address
- 5) Un router emette i messaggi "**Router Advertisements**" per rispondere alla richiesta del nodo: i messaggi "Router Advertisements" sono comunque emessi periodicamente per consentire le operazioni di verifica e di aggiornamento degli indirizzi

Statefull Autoconfiguration

Permette la configurazione automatica di un host con l'ausilio di un server

Il protocollo di colloquio tra Host e Server è denominato **Dynamic Host Configuration Protocol IPv6** (DHCPv6) che è un'estensione del protocollo DHCP utilizzato in IPv4

DHCPv6 sfrutta i meccanismi specifici di IPv6 e ha le seguenti caratteristiche:

- Un host usa il proprio link-local address per comunicare con il DHCP server
- Ha la possibilità di fornire una molteplicità di indirizzi per un'interfaccia
- I messaggi sono contenuti in pacchetti IPv6
- Rende possibili cambi di configurazione automatici

Protocollo DHCPv6

I messaggi DHCPv6 sono trasferiti tramite il protocollo UDP

Un DHCPv6 server riceve messaggi da un client mediante indirizzi multicast di tipo link-scope

Sono possibili meccanismi di relay per consentire l'accesso a DHCPv6 server anche non residenti sullo stesso link del client

Messaggi DHCPv6

Solicit: È emesso da un client per localizzare un DHCPv6 server

Advertise: È emesso da un server in risposta ad un messaggio Solicit per indicare che è disponibile un servizio DHCPv6

Request: È emesso da un client per richiedere ad uno specifico server DHCPv6 l'assegnazione di un indirizzo e i relativi parametri di configurazione

Confirm: È emesso da un client verso qualsiasi server per verificare che l'indirizzo assegnato è valido sul link a cui il client è connesso

Renew: È emesso da un client verso il server che ha eseguito l'assegnazione per estendere il lifetime della configurazione stessa

Rebind: È emesso da un client verso qualsiasi server, dopo aver ricevuto la risposta ad un messaggio Renew, per comunicare l'estensione del lifetime della configurazione stessa

Replay: È emesso da un server in risposta ai messaggi Solicit, Request, Renew, Rebind e contiene gli indirizzi assegnati e i parametri di configurazione

Release: È emesso da un client verso il server che ha eseguito l'assegnazione per indicare il rilascio di uno o più indirizzi

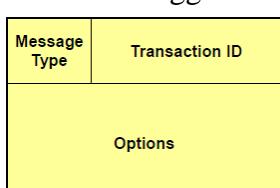
Decline: È emesso da un client verso il server che ha eseguito l'assegnazione per indicare che uno o più indirizzi non sono validi perché già in uso

Reconfigure: È emesso da un server per informare i client che deve essere iniziata una procedura di variazione nei parametri di configurazione tramite l'invio di messaggi di Renew o Information-Request

Information-Request: È emesso da un client verso un server per richiedere i parametri di configurazione, ma non l'assegnazione di un indirizzo

Formato Messaggi DHCPv6

Tutti i messaggi DHCPv6 emessi da un client hanno lo stesso formato dell'header

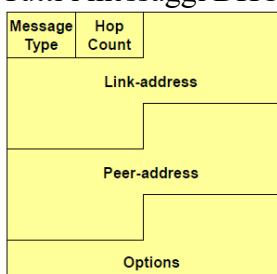


- **Message Type (8 bit):** Identifica il tipo di messaggio

- **Transaction ID (24 bit):** Identifica la transazione client server a cui si riferisce il messaggio

- **Options (lunghezza variabile)**

Tutti i messaggi DHCPv6 emessi da un server hanno lo stesso formato dell'header



- **Message Type (8 bit)**: Identifica il tipo di messaggio
- **Hop-count (8 bit)**: Numero di relay node che hanno rilanciato il messaggio
- **Link address (128 bit)**: Global address o site-local address
- **Peer address (128 bit)**: Indirizzo del client
- **Options (lunghezza variabile)**

Strategie di transizione da IPv4 a IPv6

È indispensabile definire dei meccanismi che assicurino la compatibilità tra sistemi che supportano le due versioni del protocollo e la possibilità di una migrazione graduale senza degradazione del servizio offerto da Internet

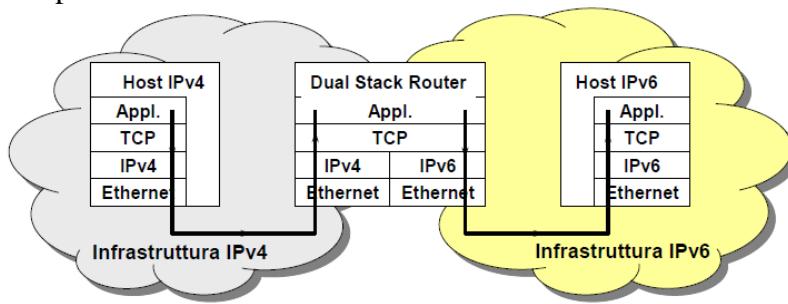
I meccanismi che sono stati definiti identificano le modalità di funzionamento dei router IPv6 quando devono interlavorare con i router IPv4 o utilizzano un'infrastruttura IPv4

I meccanismi definiti sono:

- **Dual IP layer (Dual stack)**: Fornisce il supporto completo di entrambe le versioni del protocollo in un router
- **Indirizzi IPv4 immersi (embedded) nella struttura IPv6**: Strutture di indirizzi IPv6 che contengono indirizzi IPv4
- **Configurazione di tunnel IPv6 in IPv4**: Configurazione amministrativa di tunnel punto-punto tra router IPv6 attraverso reti IPv4, i pacchetti IPv6 vengono incapsulati in pacchetti IPv4
- **Tunnel automatico di IPv6 in IPv4**: Creazione automatica di tunnel IPv6 attraverso reti IPv4 mediante l'uso degli indirizzi IPv4 contenuti negli indirizzi IPv4 compatibili

Dual Stack

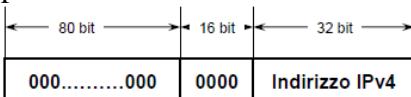
Un nodo gestisce entrambi le versioni del protocollo, è il modo più diretto per mantenere la compatibilità tra due sezioni di rete utilizzanti le due versioni del protocollo



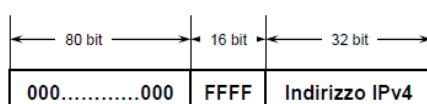
Indirizzi IPv4 immersi (embedded)

Sono definite tre tipologie:

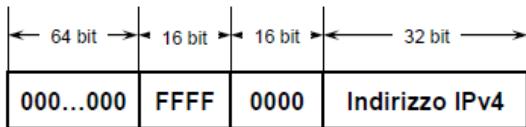
IPv4 compatible address (0::0:a:b:c:d): Sono utilizzati quando è necessario effettuare tunnel di pacchetti IPv6 attraverso reti IPv4



IPv4 mapped IPv6 address (0::ffff:a:b:c:d): Sono utilizzati da nodi IPv6 per indirizzare nodi che supportano solo il protocollo IPv4



IPv4 translated IPv6 address (0:ffff:0:a:b:c:d): Identificano un host IPv6 quando questi comunica con un nodo IPv4



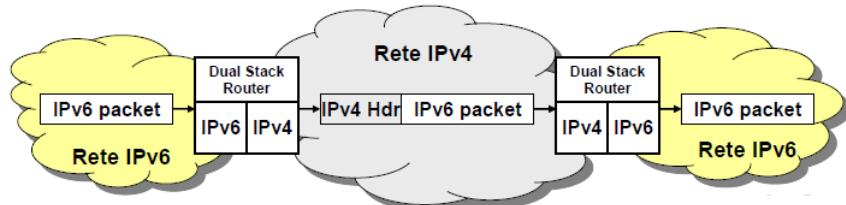
ESEMPIO pagina 47/54 slide 8

Tunnelling

La rete IPv6 si svilupperà a isolare all'interno dalla rete IPv4 esistente

Il meccanismo del Tunnelling consente di connettere aree IPv6 attraverso un'infrastruttura IPv4

La presenza di un payload IPv6 in un pacchetto IPv4 è indicata dal valore del campo Protocol uguale a 41



Sono possibili diverse modalità di tunnelling:

- **Router to Router**: Due router IPv6/IPv4 sono interconnessi da un tunnel IPv4 che è un segmento intermedio del cammino end-to-end IPv6
- **Host to Router**: Un Host IPv6/IPv4 può accedere ad un router IPv6/IPv4 tramite una rete IPv4, il tunnel è il primo segmento del cammino end-to-end IPv6
- **Host to Host**: Due host IPv6/IPv4 sono interconnessi da un tunnel IP che coincide con l'intero cammino end-to-end IPv6
- **Router to Host**: Un Router IPv6/IPv4 è connesso all'host finale IPv6/IPv4 tramite una rete IPv4. Il tunnel è l'ultimo segmento del cammino end-to-end IPv6

Le modalità di tunnelling si differenziano in base al meccanismo con cui il nodo che effettua l'incapsulamento del pacchetto determina l'indirizzo del nodo terminale del tunnel (Endpoint):

- **Tunnel configurati**: L'endpoint del tunnel è un router e quindi la destinazione finale del pacchetto non coincide con l'endpoint del tunnel

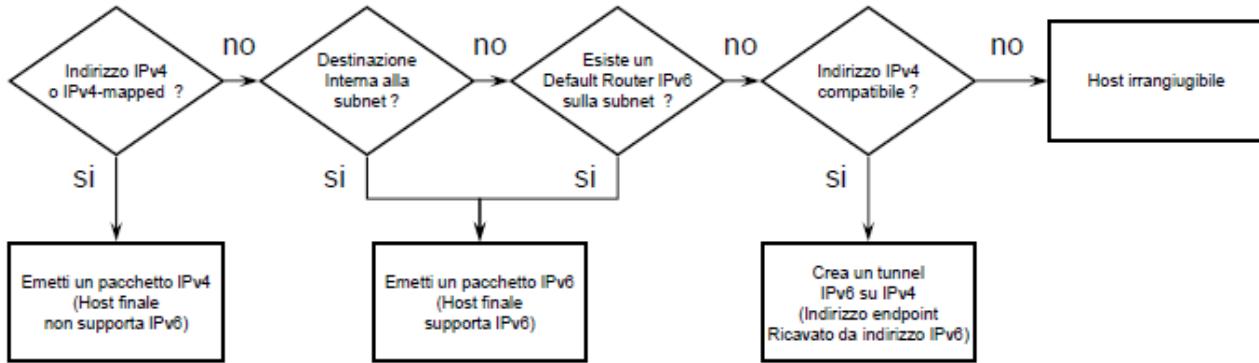
L'indirizzo IPv6 contenuto nel pacchetto non identifica l'indirizzo IPv4 dell'endpoint del tunnel. Tale informazione deve essere resa disponibile tramite configurazione

- **Tunnel automatici**: L'endpoint del tunnel è un host e quindi la destinazione finale del pacchetto coincide con l'endpoint del tunnel

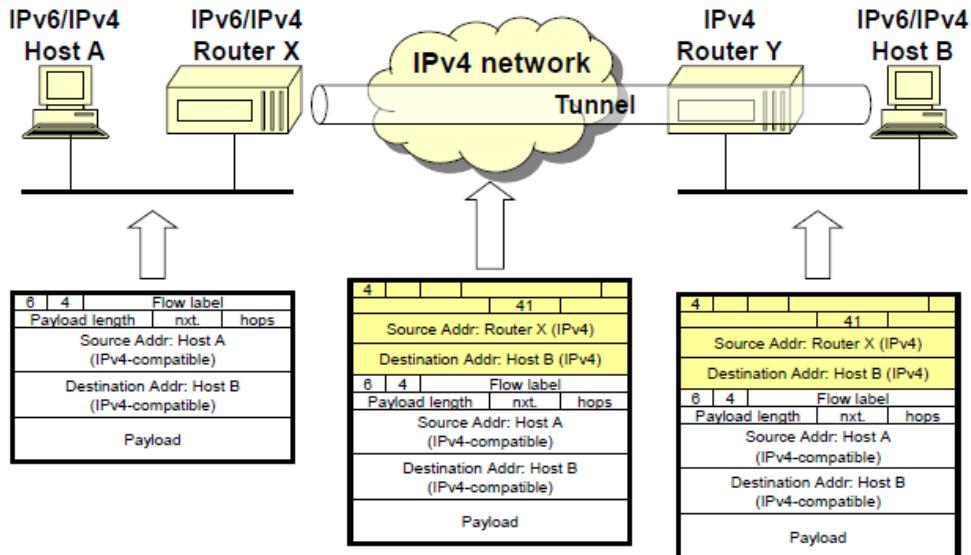
È indispensabile utilizzare un indirizzo IPv6 che identifica automaticamente anche l'indirizzo IPv4 dell'endpoint del tunnel (indirizzi IPv4 compatibili)

Tunnel automatici

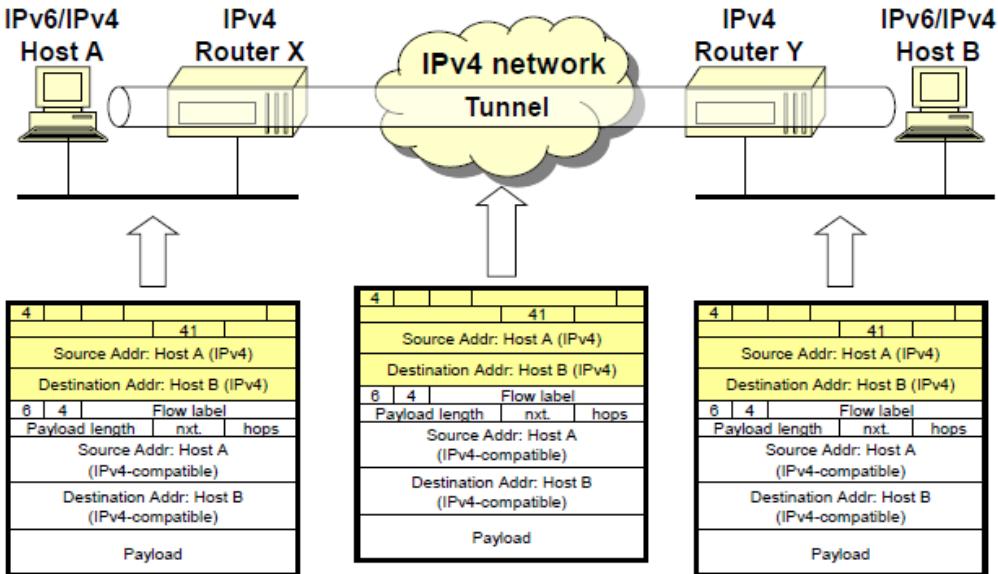
Algoritmo di decisione sull'effettuazione di un Tunnel automatico, l'algoritmo privilegia l'uso dell'infrastruttura IPv6 se esiste



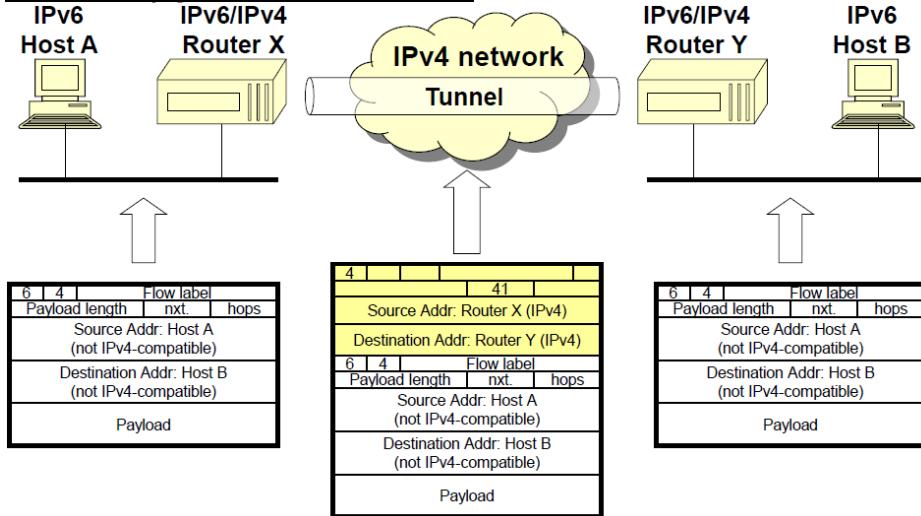
Tunnel automatico router-to-host



Tunnel automatico host-to-host



Tunnel configurati router-to-router



STRATO DI RETE (PARTE 5) CAPITOLO 9

Instradamento

Per avviare l'instradamento occorre innanzitutto creare le tabelle di routing, ovvero definire le informazioni che voglio avere riguardo ai link, definire ogni quanto voglio ricevere queste informazioni, e calcolare i cammini migliori per i vari nodi.

Requisiti

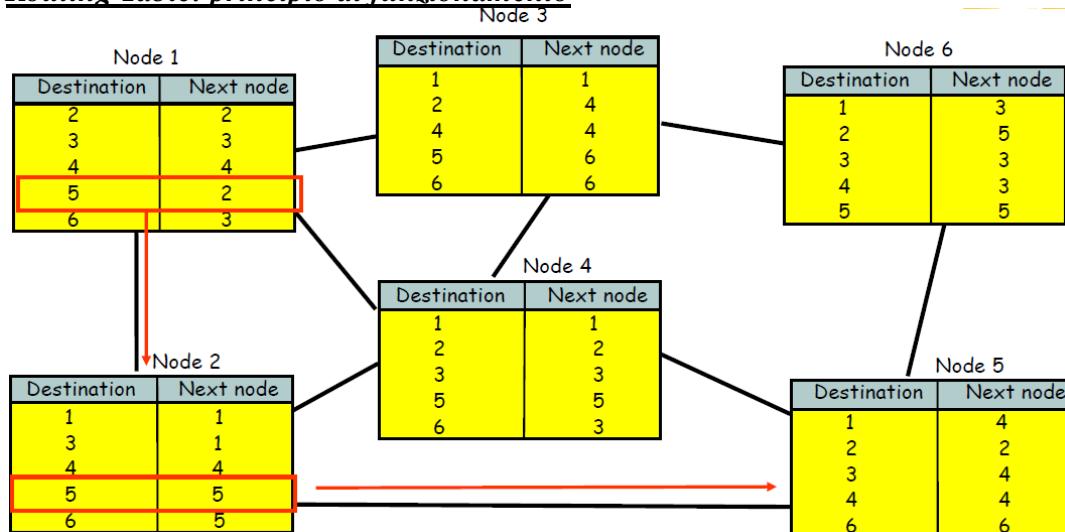
Risposta alle variazioni di stato: Variazioni di topologia o banda dei link, stato di congestione, rapida convergenza e assenza di loop

Ottimalità: Utilizzazione ottima delle risorse di rete e minimizzazione della lunghezza dei cammini

Robustezza: Continuità di servizio in presenza di condizioni anomale (alto carico, congestione di rete, guasti, errori di implementazione)

Semplicità: Basso carico di elaborazione

Routing Table: principio di funzionamento



Indirizzamento e instradamento non gerarchici

Nessuna relazione tra indirizzi e localizzazione geografica (vicinanza) delle destinazioni

Routing table composte da 16 record ciascuna: possibilità di routing table explosion

Indirizzamento e instradamento gerarchici

I prefissi indicano la rete a cui un host è connesso e le reti con lo stesso prefisso sono “vicine” (Routing table composte da 4 record ciascuna)

Instrandamento in reti IP

La scelta del router verso cui inviare il pacchetto avviene utilizzando la Routing Table contenuta in ogni host e in ogni router

Ogni elemento di una RT contiene: Indirizzo IP di destinazione (host address o network address), indirizzo del router successivo (next hop router) sul cammino verso la rete di destinazione e indicazione dell’interfaccia fisica di uscita

Un router non conosce il cammino completo verso la destinazione

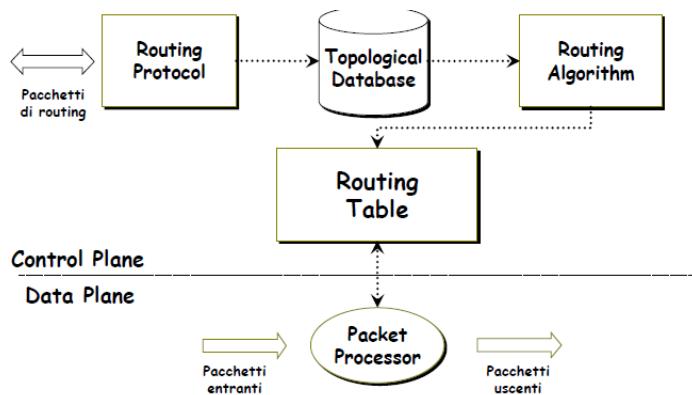
Un router esegue i seguenti passi:

- 1) Estraie dal pacchetto entrante il contenuto del campo Destination Address
- 2) Ricerca all’interno della RT il record che contiene il “longest prefix matching” con il DA del pacchetto entrante
- 3) In caso di fallimento del passo precedente, ricerca l’indirizzo del “router di default”
- 4) Se nessuno dei passi precedenti da esito positivo, il pacchetto viene classificato come **“undeliverable”** e viene scartato ed inviato un messaggio ICMP all’host sorgente

Un router possiede un **Database Topologico** in cui sono memorizzate le informazioni sulla topologia della rete che vengono aggiornate dai messaggi del **Protocollo di routing**

Sulla base delle informazioni contenute nel Database Topologico, l'**Algoritmo di routing** determina periodicamente i percorsi a costo minimo tra il router e le possibili reti di destinazione (network prefix)

La **Routing Table** viene costruita inserendo, per ogni destinazione, l’informazione relativa al next hop verso cui instradare il pacchetto



Le Routing Table sono dinamiche: Ogni router ed ogni host aggiornano periodicamente le informazioni relative alla topologia di rete

L'aggiornamento dinamico è necessario perché Internet non può essere considerato stabile, la topologia può cambiare (Aggiunta/eliminazione di reti/router/host), in caso di guasto di router/link alcuni cammini non sono utilizzabili

Possibile scelta del cammino in base allo stato di occupazione delle risorse di rete

Le RT devono essere aggiornate periodicamente (anche ad intervalli di pochi secondi)

L'aggiornamento delle RT è attuato mediante protocolli di instradamento (**Routing Protocol**)

Sistemi Autonomi

Un sistema autonomo (**Autonomous System AS**) è un insieme di router ed host controllato da una singola autorità amministrativa.

Un AS particolare è il **CORE AS** (Backbone di internet) il cui router è detto **Core Router**.

Ogni AS ha il proprio protocollo di instradamento indipendente dai protocolli usati negli altri AS

Uno Stub AS deve avere almeno un router connesso ad un core router che viene chiamato **Exterior Gateway**, mentre i router interni all'AS sono chiamati **Interior Gateway**.

I protocolli di instradamento all'interno di un AS sono detti **Interior Gateway Protocols (IGP)** e le informazioni di instradamento che coinvolgono più di un sistema autonomo sono gestite mediante gli **Exterior Gateway Protocols (EGP)**

Le informazioni di instradamento degli EGP vengono inviate agli Exterior Gateway di ogni sistema autonomo

L'instradamento all'interno di un sistema autonomo e la raccolta di dati da inviare ai core router avviene per mezzo degli IGP

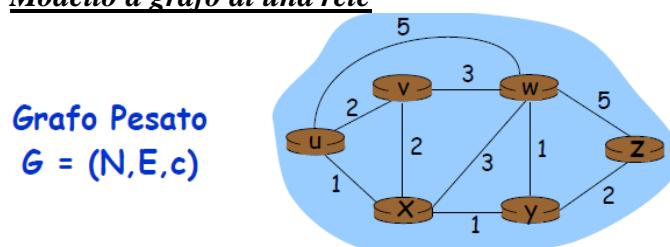
EGP - svolge tre funzioni:

Individuazione dei router adiacenti con cui scambiare le informazioni di instradamento

Verifica continua della funzionalità dei router interlocutori

Scambio periodico delle informazioni di instradamento, queste riguardano la sola raggiungibilità delle reti, non la distanza

Modello a grafo di una rete



N = insieme di nodi (router) = { u, v, w, x, y, z }

E = insieme di archi (collegamenti) = { (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), y,z }

c = insieme dei costi associati ai rami

- $c(x,x')$ = costo associato ramo (x,x')

- Se il ramo (x,x') non esiste $c(x,x') = \infty$

Il **costo di un cammino** è dato dalla somma dei costi associati agli archi componenti il cammino

$$\text{Costo di un cammino}(x_1, x_2, x_3, \dots, x_p) = c(x_1, x_2) + c(x_2, x_3) + \dots + c(x_{p-1}, x_p)$$

Il costo di un cammino è anche detto lunghezza del cammino o distanza

Il **protocollo di instradamento** mette in grado ogni router di determinare il modello a grafo della rete

L'**algoritmo di instradamento** determina il cammino a costo minimo tra due nodi della rete

Metriche

Misurano la “qualità” di un link o di un cammino:

- **Costo basso**: link ad alta qualità (es. banda elevata), da includere, se possibile, nei cammini
- **Costo elevato**: link di bassa qualità (es. banda limitata), da escludere, se possibile, nei cammini

Lunghezza di un cammino (**Path Length**) = somma dei costi dei link componenti (**Distanza**)

Possibili metriche:

- **Numero di hop**: misura approssimata delle risorse utilizzate
- **Affidabilità**: grado di disponibilità del cammino; BER
- **Ritardo**: somma dei ritardi lungo il path
- **Bandwidth**: capacità disponibile lungo un path
- **Carico**: Grado di utilizzazione dei link e dei router lungo il path

Approcci Shortest Path

Distance Vector Protocol: i nodi adiacenti si scambiano la lista delle distanze verso le destinazioni, viene determinato il next hop migliore per ogni destinazione (**Algoritmo di Bellman Ford**)

Link State Protocol: lo stato dei link (costi) è diffuso in rete (flooding)

I router conoscono l'intera topologia della rete, ogni router calcola lo shortest path ed il next hop verso ogni destinazione (**Algoritmo di Dijkstra**)

Algoritmo di Bellman-Ford

Ogni nodo ha una riga per ogni destinazione, distanza dal nodo X a se stesso = 0 mentre per tutte le altre distanze viene posta a infinito. A ogni iterazione la destinazione avvisa i suoi vicini sui costi e i vicini aggiornano la tabella.

Se però qualcosa si rompe e la connessione con la destinazione si interrompe avviene un ciclo di conteggi all'infinito tra i nodi

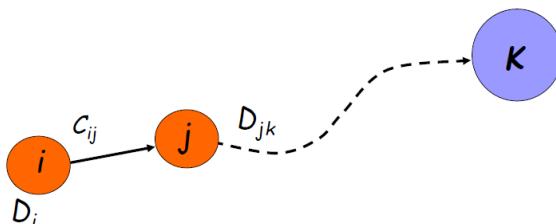
Routing Table: Per ogni destinazione vengono memorizzati Next Hop e Distanza (costo del cammino minimo)

Dest	Next	Dist

Router adiacenti si scambiano i Distance Vector periodicamente o dopo un cambio di stato
DV = (destinazione, distanza)

Ogni nodo determina per ogni destinazione il next hop migliore

Calcolo dei cammini minimi



Se D_{jk} è la distanza minima dal nodo j e k e se j è il nodo adiacente a i che si trova sul percorso a costo minimo dal nodo i verso k, si ha:

$$D_i = C_{ij} + D_j$$

Il nodo i riceve le informazioni dai nodi adiacenti $D_x(d)$ e conosce i costi dei rami verso i nodi adiacenti (C_{ix})

$$D_i = \text{Min}_x\{C_{ix} + D_x(k)\}$$

Algoritmo con vettore distanza

Iterativo, asincrono: Ogni iterazione locale è causata da una variazione del costo di uno dei link locali o dalla ricezione da un nodo adiacente di un vettore distanza aggiornato

Distribuito: Ogni nodo aggiorna i suoi vicini solo quando il suo DV cambia, un router avvisa i nodi adiacenti solo se necessario

Ciascun Nodo quando riceve un messaggio del cambio del costo da parte del suo vicino effettua il Calcolo del costo dei percorsi e se il DV cambia, invia la notifica ai suoi adiacenti

Esempio Algoritmo di Bellman-Ford slide 9 pagine 29-37 (IMPORTANTE)

Conteggio all'infinito

I nodi credono che esista un cammino in realtà non disponibile

Soluzioni al conteggio all'infinito

Split Horizon: Un router non trasmette il proprio DV aggiornato verso il router da cui ha ricevuto l'aggiornamento

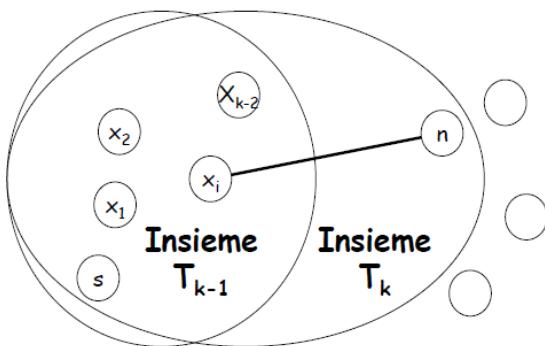
Poisoned Reverse: Un router trasmette il proprio DV aggiornato anche verso il router da cui ha ricevuto l'aggiornamento, ma indicando per la distanza aggiornata al valore ∞ e si interrompe immediatamente il loop di conteggio (questa soluzione non funziona in caso di loop complessi)

Algoritmo Dijkstra

Analizza il costo minimo tra sé stesso e tutti gli altri nodi di un grafo G (spanning tree), aumentando a ogni iterazione la distanza.

L'algoritmo procede a passi successivi:

- al passo k -mo sono individuati i k nodi raggiungibili dal nodo sorgente tramite i cammini a costo più basso, tali k nodi formano l'insieme T_k
- al passo $k+1$ -mo si individua il nodo n che è caratterizzato dal cammino dal costo più basso dal nodo s che transita esclusivamente nei nodi dell'insieme T_k
- viene formato l'insieme T_{k+1} aggiungendo il nodo n all'insieme T_k
- l'algoritmo termina quando sono stati aggiunti all'insieme T_k tutti i nodi del grafo



Notazioni

- N : insieme dei nodi del grafo
- s : nodo sorgente
- T_k : insieme dei nodi raggiunti dall'algoritmo al passo k
- $c(i,j)$: peso (costo) del ramo i,j :
 - $c(i,i) = 0$
 - $c(i,j) \geq 0$ se i vertici i e j sono connessi direttamente

- $c(i,j) = \infty$ se i vertici i e j non sono connessi direttamente

$L_k(n)$: costo del cammino minimo, individuato dall'algoritmo fino al passo k, tra il nodo s ed un generico nodo n

Inizializzazione ($k = 1$)

- $T_1 = \{s\}$
- $L_1(n) = c(s,n)$ per $n \neq s$

Aggiunta di un nodo (passo $1 \leq k \leq N$)

- trovare $x \neq T_{k-1}$ tale che:

$$L_{k-1}(x) = \min_{j \notin T_{k-1}} \{L_{k-1}(j)\}$$

- aggiungere all'insieme T_{k-1} il nodo x ed il ramo incidente a x

Aggiornamento dei cammini minimi

- $L_k(n) = \min[L_{k-1}(n), L_{k-1}(x) + c(x,n)]$ per tutti i valori di $n \notin T_k$

Al termine l'insieme T_N è uno spanning tree del grafo di partenza contenente i cammini a costo minimo tra il nodo sorgente e tutti gli altri nodi del grafo

$L_N(n)$ indica il costo del cammino a costo minimo tra il nodo s ed il nodo n

Si noti che al passo k-mo viene aggiunto all'insieme T_{k-1} il k-mo nodo ed è individuato il cammino a costo minimo tra tale nodo ed il nodo sorgente, questo cammino transita esclusivamente attraverso i nodi sinora compresi nell'insieme T_{k-1}

La complessità dell'algoritmo è $O(N^2)$

ESEMPIO algoritmo di dijkstra slide 9 pagina 47

Complessità dell'algoritmo (n nodi)

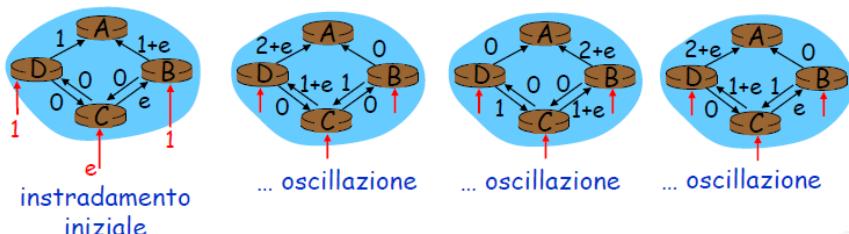
Ciascuna iterazione: controllo su tutti i nodi, w, non in N

$$\frac{n(n+1)}{2} \rightarrow O(n^2)$$

La più efficiente implementazione possibile: $O(n \log n)$

Possibili oscillazioni

Es. costo del collegamento = quantità di traffico trasportato



Instradamento gerarchico

Da pagina 53 a pagina 58 slide 9 (esempi)

STRATO DI RETE (PROTOCOLLI DI INSTRADAMENTO) CAPITOLO 10

I protocolli IGP più comuni sono:

- **RIP (Routing Information Protocol)**: utilizzato in reti di piccole e medie dimensioni, utilizza l'algoritmo Distance Vector, la metrica dei rami dipende normalmente dal loro stato (sano/guasto)
Conteggio degli hop come metrica di costo (max = 15 hop)
È molto semplice, ma la convergenza è lenta e lo stato di equilibrio può essere un sub-ottimo
- **OSPF (Open Shortest Path First)**
- **IGRP (Interior Gateway Routing Protocol)**: protocollo proprietario Cisco

RIP

I messaggi RIP sono trasportati dal protocollo UDP (port 520) ed emette due tipi di messaggi:

- 1) **Request** (per chiedere ai vicini il distance vector)
- 2) **Response** (per annunciare il suo distance vector).

Ogni messaggio Response contiene un elenco comprendente fino a 25 sottoreti e la distanza tra l'origine del messaggio e ciascuna di queste.

I router adiacenti si scambiano periodicamente gli aggiornamenti d'instradamento con un valore di default di 30 secondi

RIP v1

Command	Version	0
Address Identifier		0
	IP Address 1	
	0	
	0	
	Metric for address 1	
Address Identifier		0
	IP Address 2	
	0	
	0	
	Metric for address 2	
⋮	⋮	⋮
Address Identifier		0
	IP Address N	
	0	
	0	
	Metric for address N	

Address 1
distance
Address 2
distance
Fino a 25
addresses

Header: Presenta i campi **Command** (request, risponde) e **Version**

Block: Composto dai campi **IP address** (rete, sottorete o host) e **Metric** (distanza dalla rete indicata nell'IP address)

RIP v2

Command	Version	Reserved
Address Identifier		Reserved
	IP Address 1	
	Subnet Mask	
	Next Hop	
	Metric for address 1	
Address Identifier		Reserved
	IP Address 2	
	Subnet Mask	
	Next Hop	
	Metric for address 2	
⋮	⋮	⋮
Address Identifier		Reserved
	IP Address N	
	Subnet Mask	
	Next Hop	
	Metric for address N	

Address 1
distance
Address 2
distance
Fino a 25
addresses

- **IP address:** rete, sottorete o host
- **Subnet Mask:** specifica come interpretare i bit dell'indirizzo
- **Next Hop:** indica a quale next hop router il router emittente il messaggio RIP invierà i pacchetti diretti all'indirizzo specificato
- **Metric:** distanza dalla rete indicata nell'IP address

ESEMPIO INIZIALIZZAZIONE SLIDE 10 PAGINE 9-16

RIP: guasto sul collegamento e recupero

Se un router non riceve messaggi da un nodo adiacente per un intervallo di 180 sec, il nodo adiacente viene considerato spento o guasto:

RIP modificherà la tabella d'instradamento locale propagando l'informazione mandando annunci ai router adiacenti che verrà propagata su tutta la rete

I nodi adiacenti invieranno nuovi messaggi (se la loro tabella d'instradamento è cambiata)

L'utilizzo dell'inversione avvelenata (**poison reverse**) evita i loop (distanza infinita = 16 hop)

ESEMPIO RIP GUASTO DI UN RAMO SLIDE 10 pagine 18-21

OSPF (Open Shortest Path First)

Utilizzato in reti backbone, utilizza l'algoritmo **Link State Protocol** e diffonde un messaggio sulla rete con il **flooding** (Esplora tutti i possibili cammini tra origine e destinazione) che si chiama **Link State Advertisement (LSA)** che viene emesso ogni volta che si verifica un cambiamento sul suo stato.

Se due percorsi hanno lo stesso costo non devo sceglierne uno, posso usare tutti e due a differenza di **RIP**.

I messaggi OSPF vengono trasportati direttamente in pacchetti IP, non viene utilizzato un protocollo di trasporto (TCP o UDP)

Vantaggi di OSPF

Sicurezza: gli scambi tra router sono autenticati

Multipath: quando più percorsi verso una destinazione hanno lo stesso costo, OSPF consente di usarli senza doverne scegliere uno, come invece avveniva in RIP (**Equal Path Cost Multipath ECMP**)

Su ciascun collegamento, vi possono essere più metriche di costo per differenti TOS

es: il costo di un link via satellite sarà “basso” per un pacchetto best effort; “elevato” per un pacchetto real time

Supporto integrato per l'instradamento unicast e multicast: Per consentire l'instradamento multicast viene impiegato MOSPF (OSPF multicast) che utilizza il database topologico di OSPF

Supporto alle gerarchie in un dominio d'instradamento

Tipi di LSA

Gli LSA sono emessi quando un router contatta un nuovo router adiacente, quando un link si guasta, quando il costo di un link varia o periodicamente ogni fissato intervallo di tempo

La rete trasporta gli LSA mediante la tecnica di flooding: un LSA è rilanciato da un router su tutte le sue interfacce tranne quella da cui è stato ricevuto

Gli LSA trasportano dei riferimenti temporali (time stamp) o numeri di sequenza per evitare il rilancio di pacchetti già rilanciati e consentire un corretto riscontro dal ricevente

Tecnica Flooding

Obiettivi di OSPF: Tutti i router di una rete abbiano un database topologico contenente lo stato della rete e le stesse informazioni sullo stato dei link

Alla ricezione di un LSP un router esamina i campi di un LSP: link identifier, metrica, time stamp o numero di sequenza, se il dato non è contenuto nel database, viene memorizzato e l'LSP è rilanciato su tutte le interfacce del router tranne quella di ricezione, se il dato ricevuto è più recente di quello contenuto nel database, il suo valore è memorizzato e l'LSP è rilanciato su tutte le interfacce del router tranne quella di ricezione, se il dato ricevuto è più vecchio di quello contenuto nel database, viene rilanciato un LSP con il valore contenuto nel database esclusivamente sull'interfaccia di arrivo dell'LSP, se i due dati sono della stessa età non viene eseguita alcuna operazione

La tecnica flooding ha i **vantaggi** di esplorare tutti i possibili cammini tra origine e destinazione, è estremamente affidabile e robusta e almeno una copia di ogni LSP seguirà la via a minor costo
Il traffico di controllo generato dipende dalle dimensioni della rete e può essere molto elevato

Suddivisione di grandi reti in aree

Se la rete è di grandi dimensioni cresce il numero di record del database e quindi la memoria necessaria in ogni router, il tempo necessario al calcolo dei percorsi e il traffico di segnalazione dovuto all'invio degli LSP

OSPF supporta un instradamento di tipo gerarchico in cui una rete viene suddivisa in aree con sezioni indipendenti di rete, database separati e meccanismi di flooding indipendenti

Le singole aree sono interconnesse da un'area di backbone

Open Shortest Path First

OSPF è il protocollo IGP più utilizzato nelle aree di grandi dimensioni poiché è basato sullo scambio di LSP, supporta metriche relative a diversi valori del campo TOS, supporta l'uso del concetto di variable length subnet mask (CIDR), supporta il servizio di autenticazione tra router, supporta l'indicazione di specific routes, ha una riduzione delle dimensioni delle tabelle di routing grazie all'uso del concetto di Designated Router (DR) e supporta l'indicazione di virtual link per l'interconnessione di aree non contigue

Terminologia OSPF

Area: è un insieme logico di reti e di router (geografico, amministrativo, ...) che ha lo scopo di limitare la dimensione dei database di descrizione della topologia di rete all'interno dei router

All'interno di un'area i router devono avere database identici che descrivono la topologia di rete

Border Router: Un Area Border Router trasmette LSA contenenti informazioni sulle reti esterne all'interno dell'area (costo di raggiungimento)

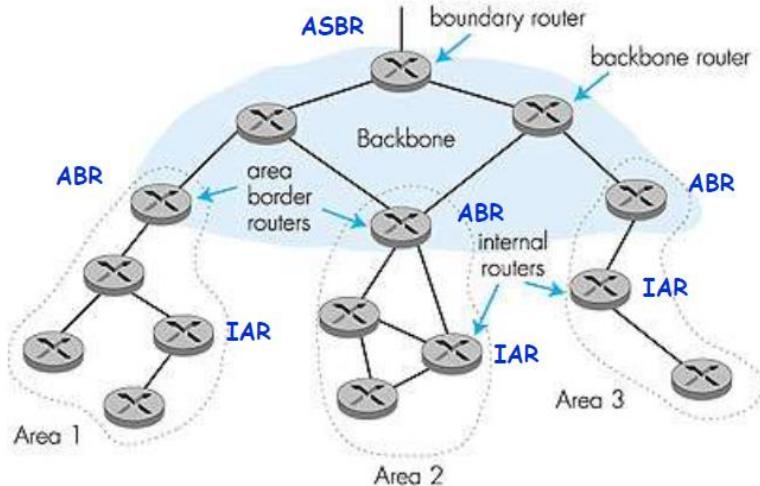
Tutte le reti OSPF devono essere composte da almeno un'area, denominata area di backbone

Intra-Area Router (IAR): Sono i router che sono situati all'interno di una area OSPF che scambiano LSA con tutti gli altri router dell'area e gestiscono il database relativo alla topologia dell'area

Area Border Router (ABR): Sono i router che sono connessi a due o più aree OSPF e gestiscono i database di topologia di tutte le aree a cui sono connessi trasmettendo all'interno di ogni area LSA relativi alle reti presenti in ogni area

AS Boundary Router (ASBR): Sono i router che sono situati a bordi del dominio OSPF e scambiano LSA contenenti informazioni di raggiungibilità di reti di altri AS, inviando LSA all'interno del dominio con informazioni sui percorsi esterni

OSPF strutturato gerarchicamente



Suddivisione di grandi reti in aree

Un ABR contiene i database di tutte le aree a cui appartiene ed emette degli appositi messaggi (**summary records**) che contengono la lista delle sottoreti raggiungibili attraverso le aree a cui appartiene

Instradamenti esterni

Un AS è connesso ad altri AS attraverso uno o più “AS Border Router” ASBR

Se un'area ha un unico ASBR è sufficiente indicare a tutti i router interni all'area l'instradamento di default verso l'esterno

Se gli ASBR sono più di uno, ognuno di essi indicherà ai router interni il costo della via verso l'esterno (**External record**).

Tipologie di LSA

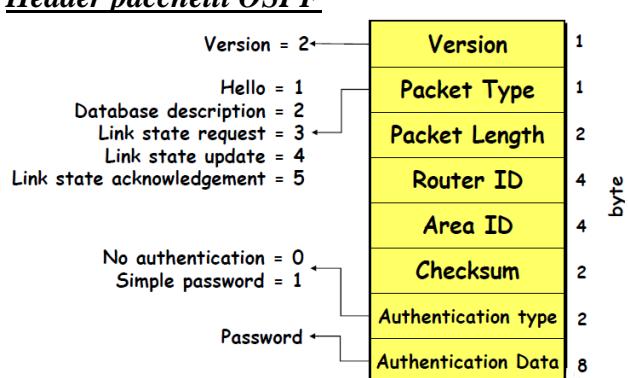
Link State Advertisements (LSA): Sono i messaggi scambiati tra router OSPF per aggiornare i link state database (LSDB) e i percorsi inter-area e inter-AS.

Router link advertisement: Indicano lo stato dei link uscenti da un router, sono inviati all'interno di una singola area.

Summary link advertisement: Sono generati dagli ABR e individuano le reti contenute nelle altre aree ed i relativi costi di raggiungimento, sono inviati all'interno di tutte le aree gestite da un ABR.

AS external link advertisement: sono generati dagli ASBR e indicano i cammini verso le reti esterne al dominio OSPF, sono inviati all'interno di tutta la area di un dominio OSPF.

Header pacchetti OSPE



Tutti i tipi di LSA hanno lo stesso header:

OSPF Link State Header

Link State Age	2
Options	1
Link State Type	1
Link State ID	4
Advertising Router	4
Link State Sequence Number	4
Link State Checksum	2
Length	2

Ottetti

Link State Age: Indica il tempo (in secondi) di emissione dell'advertisement

Link State Type:

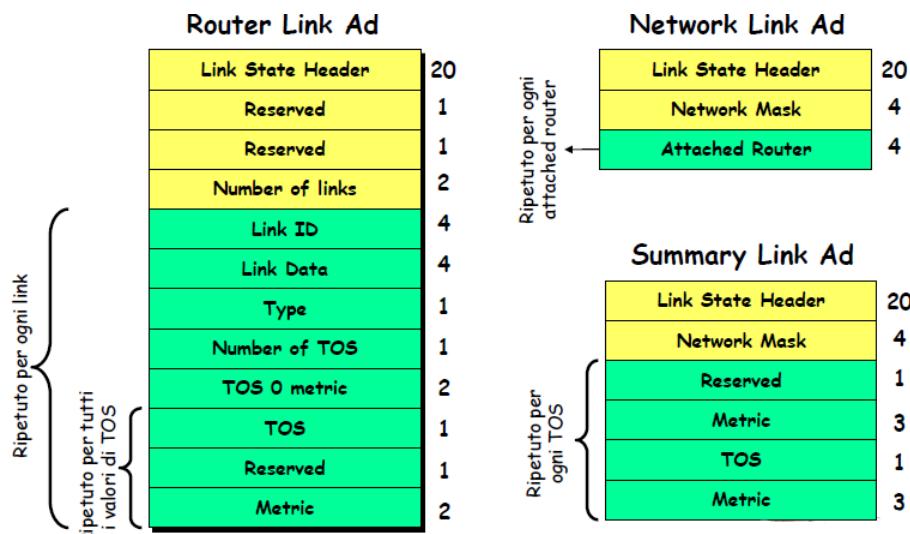
- 1: Router link
- 2: Network link
- 3: Summary link: Inter-area, intra-AS route
- 4: Summary link: Route verso l'AS Boundary Router
- 5: AS External link: Route verso reti esterne all'AS

Link State ID: Indica il tipo di link a cui si riferisce il messaggio

- Tipo 1 e 4: indirizzo IP del Router emittente
- Tipo 3 e 5: indirizzo IP della rete a cui si riferisce il messaggio
- Tipo 2: indirizzo IP del DR emittente

Advertising Router: Indirizzo IP del router che ha emesso il messaggio

- Tipo 1 : identico al campo Link State ID
- Tipo 2: indirizzo IP del DR
- Tipo 3 e 4: indirizzo IP del ABR
- Tipo 5: indirizzo IP del ASBR
- Tipo 5: indirizzo IP del ASBR



External Link Ad

Link State Header	20
Network Mask	4
Reserved	1
Metric	3
Forwarding Address	4
External Route Tag	4
TOS	1
TOS metric	3
Forwarding Address	4
External Route Tag	4

Ripetuto per tutti i valori di TOS

Network Mask: Maschera della rete a cui si riferisce il pacchetto, l'indicazione della rete è contenuta nell'header

Metric: Costo del cammino

Forwarding Address: Indirizzo IP a cui deve essere inviato il traffico diretto alla rete indicata

External Route Tag: Suffisso ad uso degli ASBR

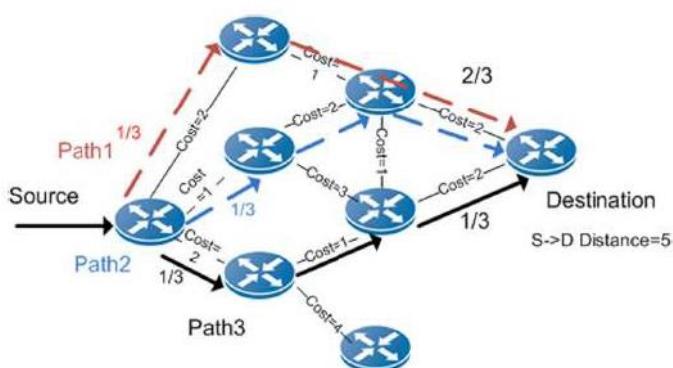
Equal Cost Multi Path (ECMP)

ECMP è una tecnica che rende possibile l'uso di “equal cost path” tra le sottoreti sorgente e destinazione tra cui distribuire il traffico

Gli “equal cost path” calcolati dall'algoritmo di Dijkstra sono memorizzati nella tabella di bilanciamento del carico (**Load Balancing Table**)

Il throughput di rete aumenta di un valore variabile tra il 50% e 110% e i percorsi alternativi possono essere utilizzati come backup reciproco in caso di guasto in rete

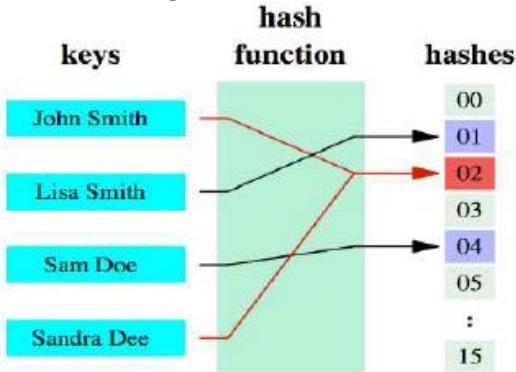
Esempio



Il traffico tra S e D è ripartito sui tre cammini in modo uguale

Solo il nodo sorgente supporta l'ECMP, gli altri nodi si comportano come nel caso di single path

Flow Hashing ECMP



Funzione Hash: Funzione che trasforma dati di grandi dimensioni e di lunghezza variabile in una stringa di dimensioni piccole e di lunghezza costante

Spesso l'obiettivo è quello di utilizzare l'hash di un dato come indice di accesso per il lookup in una tabella

Flow Hashing ECMP: Esegue la ripartizione del traffico sui cammini disponibili a costo uguale per bilanciare il carico in rete

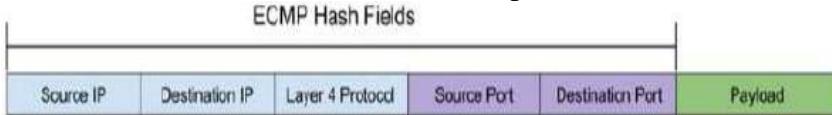
Calcola l'hash dell'header del pacchetto che viene utilizzato come input per il lookup della porta di uscita dello switch

È conservata la corretta sequenza dei pacchetti: I pacchetti di uno stesso flusso (es. stessa coppia sorgente, destinazione) sono instradati sullo stesso path

Implementazione

La tabella di routing contiene per ogni destinazione entry multipli associati ai path con costo uguale
L'hash dell'header di un pacchetto è usato come indice per l'accesso alla tabella di routing per la decisione della porta di uscita dal router

Oltre all'informazione dell'interfaccia d'ingresso, i campi di un pacchetto IP/TCP utilizzati per la funzione di hash sono normalmente: IP protocol, Source & Destination IP e Source & Destination Port



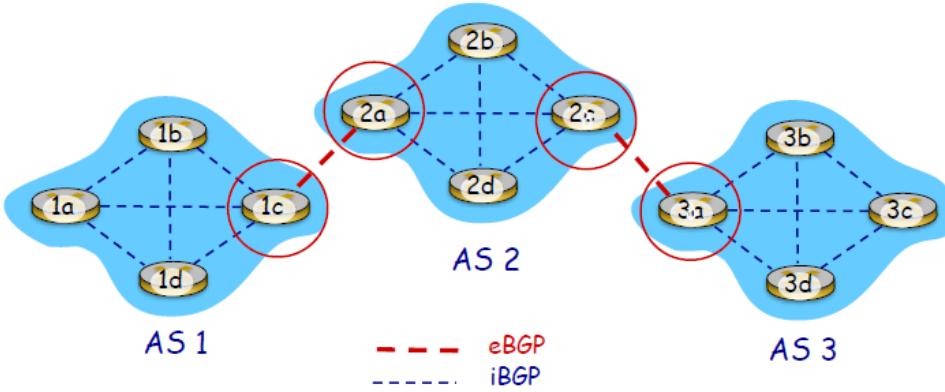
Poiché la funzione hash è deterministica l'output è lo stesso per un dato input, tutti i pacchetti appartenenti allo stesso flusso (header identico) saranno instradati sullo stesso path

Border Gateway Protocol (BGP)

Rappresenta lo standard dei protocolli EGP

BGP mette a disposizione di ciascun AS un modo per ottenere informazioni sulla raggiungibilità delle sottoreti da parte di AS confinanti (**iBGP**), propagare le informazioni di raggiungibilità a tutti i router interni di un AS (**eBGP**) e determinare percorsi “buoni” verso le sottoreti sulla base delle informazioni di raggiungibilità e delle politiche dell'AS

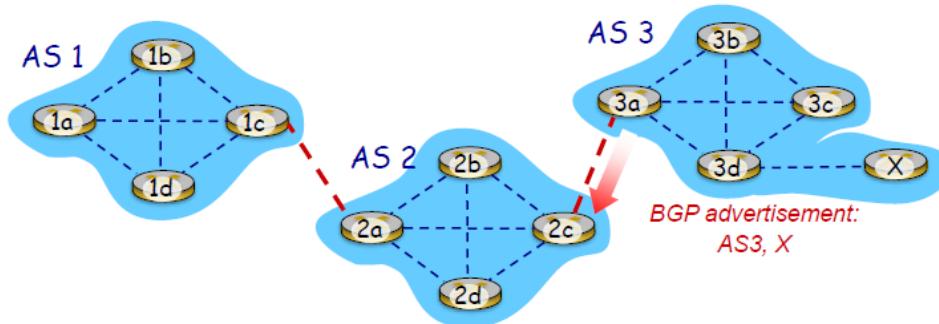
BGP consente a ciascuna sottorete di comunicare la propria esistenza al resto di Internet ed è un protocollo **path vector**: annuncia i cammini (path) verso le sottoreti prefissi di destinazione



I gateway router eseguono ambedue i protocolli eBGP and iBGP

BGP session: due BGP router (peers) scambiano i messaggi BGP utilizzando una connessione TCP annunciando i cammini verso le reti di destinazione (network prefix)

Quando il router 3a (AS3 gateway router) annuncia il cammino (AS3,X) a router 2c (AS2 gateway router) assicura che AS3 rilancerà i pacchetti verso la destinazione X



Terminologia BGP

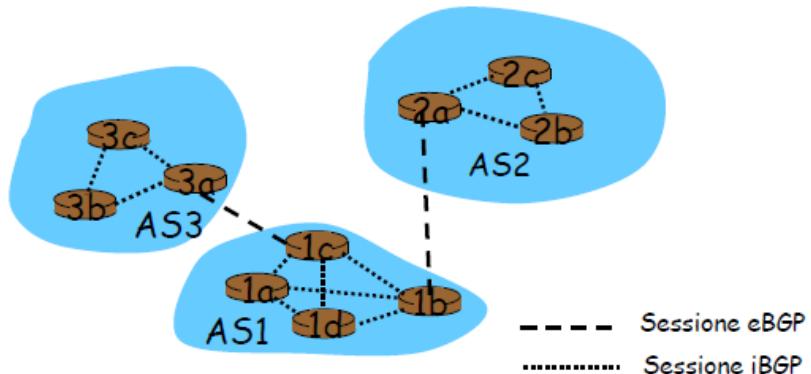
BGP speaker: Un router che supporta il protocollo BGP, il quale non necessariamente coincide con un border router

BGP Neighbors: Una coppia di BGP speaker che si scambiano informazioni di instradamento inter-AS
Possono essere di due tipi:

Interni: se appartengono allo stesso AS

Esterne: se appartengono ad AS diversi

BGP session: la connessione TCP che supporta il colloquio tra due BGP speaker



Attributi del percorso e rotte BGP

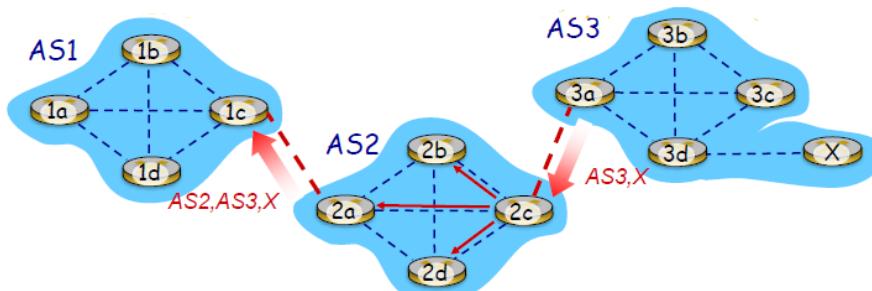
Quando un router annuncia un prefisso per una sessione BGP, include anche un certo numero di **attributi BGP**: prefisso + attributi = “rotta”

Due dei più importanti attributi sono l'**AS-PATH** che elenca i sistemi autonomi attraverso i quali è passato l’annuncio del prefisso e il **NEXT-HOP**

Quando si deve inoltrare un pacchetto tra due sistemi autonomi, questo potrebbe essere inviato su uno dei vari collegamenti fisici che li connettono direttamente.

Quando un router gateway riceve un annuncio di rotta, utilizza le proprie politiche d’importazione per decidere se accettare o filtrare tale rotta

BGP path advertisement



Il router 2c riceve un messaggio di path advertisement **AS3,X** (protocollo eBGP) dal router 3a

In base alla policy di AS2, il router 2c accetta il cammino AS3,X e lo rilancia (protocollo iBGP) a tutti i router di AS2

In base alla policy di AS2, il router 2a annuncia (protocollo eBGP) il cammino **AS2 , AS3, X** al router 1c

I gateway router possono apprendere l’esistenza di path multipli verso una destinazione:

Il router 1c acquisisce il path AS2,AS3,X dal router 2a

Il router 1c acquisisce il path AS3,X dal router 3a

In base alla policy di AS1, il gateway router 1c sceglie il path AS3,X e annuncia il path all’interno di AS1 (protocollo iBGP)

Politiche d’instradamento BGP

A, B, C sono reti di provider

W, Y sono reti d’utente

X è una rete stub dual-homed (interconnessa a due reti): X non vuole supportare il traffico di transito da B a C e X non annuncerà a B la rotta verso C

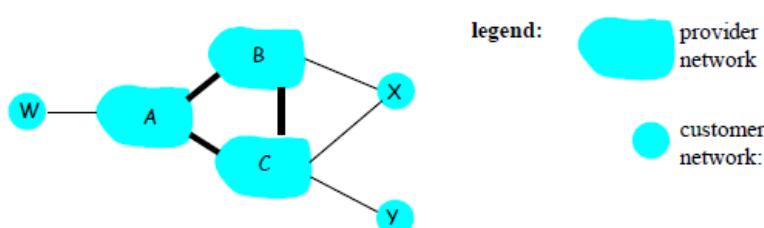


Figure 4.5-BGPnew: a simple BGP scenario

A annuncia a B e C del percorso AW

B annuncia a X del percorso BAW

B non annuncia a C del percorso BAW se B non ha nessun “interesse commerciale” nella rotta CBAW poiché nessuna tra le reti A, C e W è cliente di B, oppure se B vuole costringere C ad instradare verso W attraverso A oppure se B vuole instradare solo da/verso i suoi clienti

Esempi Routing table BGP e OSPF e Routing Hot Potato slide 19 pagine 58-60

Terminologia BGP

AS number: Identificatore a 16-bit che identifica univocamente un AS

AS path: è la lista di AS che sono attraversati in un cammino

Politiche di routing: nel protocollo BGP non sono definite regole fisse per la scelta dei cammini inter-AS, ma le regole sono definite dal gestore di ogni AS:

Un AS multi-homed può rifiutare di operare come AS di transito o farlo solo per alcuni AS

Un AS può scegliere a quale altro AS affidare il traffico di transito

Tra le possibili scelte un BGP speaker sceglie quella da preferire in base alla politica di routing fissata dal gestore, in caso di cammini alternativi, un BGP speaker li mantiene tutti ma ne comunica uno solo agli altri AS

Selezione dei percorsi BGP

Un router può ricavare più di una rota verso un determinato prefisso, e deve quindi sceglierne una

Regole di eliminazione:

Alle rotte viene assegnato come attributo un valore di preferenza locale. Si selezionano quindi le rotte con i più alti valori di preferenza locale

Si seleziona la rota con valore AS-PATH più breve

Si seleziona quella il cui router di NEXT-HOP più vicino: instradamento “hot potato”

Se rimane ancora più di una rota, il router si basa sugli identificatori BGP

Messaggi BGP

I messaggi BGP vengono scambiati attraverso TCP e possono essere di tipo:

OPEN: Apre la connessione TCP e autentica il mittente

UPDATE: Annuncia il nuovo percorso (o cancella quello vecchio)

KEEPALIVE: Mantiene la connessione attiva in mancanza di UPDATE

NOTIFICATION: Riporta gli errori del precedente messaggio; usato anche per chiudere il collegamento

Protocolli inter-AS vs. protocolli intra-AS

Politiche:

- **Inter-AS:** il controllo amministrativo desidera avere il controllo su come il traffico viene instradato e su chi instrada attraverso le sue reti.

- **Intra-AS:** unico controllo amministrativo, e di conseguenza le questioni di politica hanno un ruolo molto meno importante nello scegliere le rotte interne al sistema

Scala: L’instradamento gerarchico fa “risparmiare” sulle tabelle d’instradamento, e riduce il traffico dovuto al loro aggiornamento

Prestazioni:

- **Intra-AS:** orientato alle prestazioni

- **Inter-AS:** le politiche possono prevalere sulle prestazioni

CAPITOLO 11

Indirizzi MAC

Indirizzo IP a 32 bit (Strato di rete): Indirizzo a livello di rete analogo all'indirizzo postale di una persona, ha una struttura gerarchica e deve esser aggiornato quando una persona cambia residenza (cambia rete)

Indirizzo MAC a 48 bit (strato di data link): Analogico al numero di codice fiscale di una persona Ha una struttura orizzontale e non varia a seconda del luogo in cui la persona si trasferisce (indipendente dalla rete)

Ciascuna scheda di rete ha un indirizzo MAC univoco

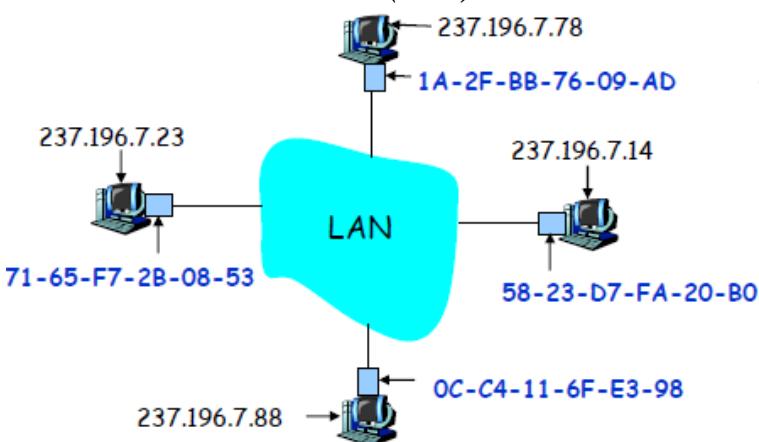
Indirizzo broadcast: FF-FF-FF-FF-FF-FF

Una LAN usa l'indirizzo MAC per consegnare le frame a destinazione

La IEEE gestisce gli indirizzi MAC: quando una società vuole costruire schede di rete compra un blocco di spazio di indirizzi (**unicità degli indirizzi**)

Gli indirizzi IP hanno una struttura gerarchica e devono essere aggiornati se il terminale cambia rete dipendono dalla sottorete IP cui il nodo è collegato.

Address Resolution Protocol (ARP)



Ogni nodo IP (host, router) nella LAN ha una **tavella ARP** che contiene la corrispondenza tra indirizzi IP e MAC: < Indirizzo IP; Indirizzo MAC; TTL> dove **TTL** (tempo di vita) che indica quando bisognerà eliminare una data voce nella tabella, il tempo di vita tipico è di 20 min

Protocollo ARP nella stessa sottorete

Un host A vuole inviare un messaggio ad un host B in cui l'indirizzo MAC di B non è nella tabella ARP di A

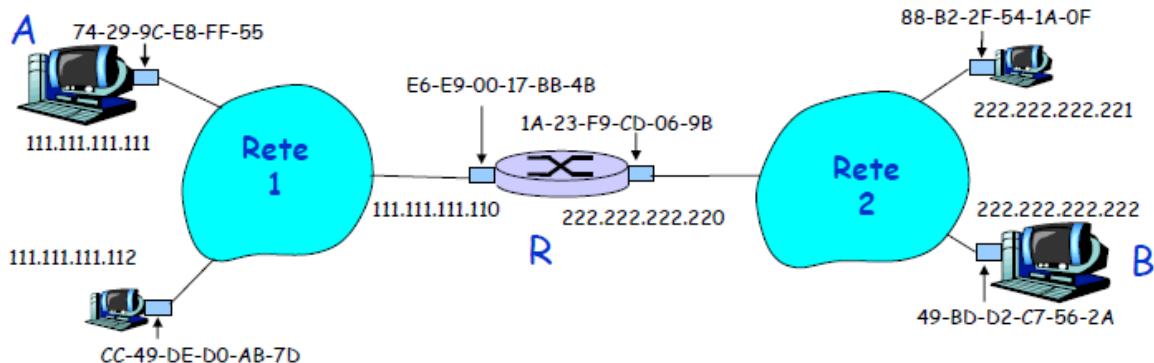
A trasmette in una frame broadcast il **messaggio di richiesta ARP**, contenente l'indirizzo IP di B Tutti gli host della LAN ricevono la richiesta ARP

L'host B riceve la frame ARP e risponde ad A comunicandogli il proprio indirizzo MAC: la frame viene inviata all'indirizzo MAC di A che è scritto nel messaggio ARP

Il messaggio di risposta ARP è inviato in una frame standard

ARP è “plug and play”: La tabella ARP di un nodo si costituisce automaticamente e non deve essere configurata dall'amministratore del sistema

Invio verso un nodo esterno alla sottorete



Invio di un pacchetto tra due host A a B, localizzati in due LAN diverse (Reti 1 e 2) attraverso un router R, è necessario che A conosca l'indirizzo IP di B

Il router R ha due tabelle ARP, una per ciascuna LAN

A crea un pacchetto con origine A e destinazione B

A usa ARP per ottenere l'indirizzo MAC di R (scheda della rete 1)

A invia il pacchetto a R

R rimuove il pacchetto IP dalla frame Ethernet, e vede che la destinazione è B

R usa ARP per ottenere l'indirizzo MAC di B

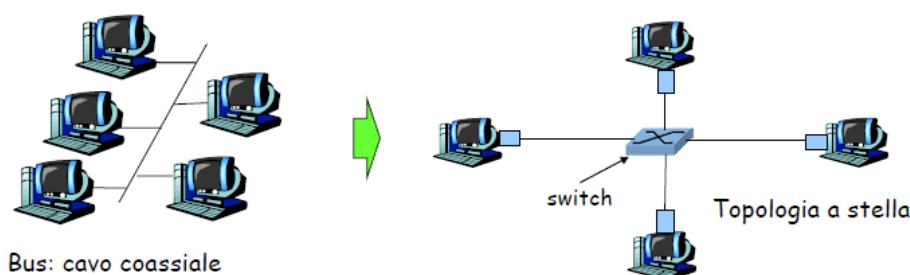
R crea un frame contenente il pacchetto IP e lo invia a B

Ethernet (Standard IEEE 802.3)

Topologia a bus o a stella

Al centro della stella è collocato un elemento denominato **switch** che esegue le funzioni di commutazione delle frame sui rami della stella

Ciascun nodo esegue un protocollo Ethernet separato e non entra in collisione con gli altri



Slot Time (Parametro principale di sistema):

- limite superiore per rivelare una collisione ($2t_{prop}$)
- limite superiore per acquisire il canale in trasmissione
- limite superiore per la lunghezza di una frame in caso di collisione
- quanto per il calcolo del tempo di ritrasmissione in caso di collisione
- $\max\{\text{round-trip propagation, MAC jam time}\}$

IEEE 802.3: Parametri originali

Transmission Rate: 10 Mbit /s

Lunghezza minima di una frame: 512 bit = 64 byte

Slot time: $\frac{512 \text{ bit}}{10 \text{ Mbit/s}} = 51.2 \mu \text{sec}$:

- $51.2 \mu\text{sec} \times 2 \times 10^5 \text{ km/sec} = 10.24 \text{ km}$ (round trip delay)

- 5.12 km estensione massima della rete

Lunghezza massima della rete: 2500 metri + 4 repeater (5 tratte di 500 metri ciascuna)

Regola: ogni incremento di 10 volte del bit rate, determina la diminuzione di 10 volte della lunghezza massima della rete

Frame Ethernet (IEEE 802.3)



Struttura

La scheda di rete incapsula i pacchetti IP in una frame Ethernet formata da:

Preambolo: 7 byte che serve per attivare le schede dei riceventi e sincronizzare i clock con quelli dell'emittente, Ogni byte ha la configurazione 10101010 (onda quadra)

Start Delimiter: 1 byte che indica l'inizio della frame (10101011)

Source e Destination Address: 6 byte ciascuno, sono gli indirizzi MAC di ricevente e destinatario, Quando una scheda di rete riceve una frame contenente nel campo destination address il proprio indirizzo MAC o l'indirizzo broadcast (es.: un pacchetto ARP), copia la frame nel buffer di ricezione mentre le frame con altri indirizzi MAC vengono ignorati.

Length: 2 byte indica la lunghezza del campo informativo che può essere al massimo 1500 bytes, se poi invece calcolo la lunghezza massima del frame senza preambolo e SD allora sono 1518

PAD: assicura la lunghezza minima del frame sia 64 byte

CRC: 4 byte per il controllo

Funzionamento

Si tratta del funzionamento del protocollo **CSMA/CD**, il protocollo rimane standard e anche la struttura dei frame, ciò che si differenzia sono le velocità e i mezzi trasmissivi

La scheda di rete prepara una frame Ethernet e se il canale è inattivo, inizia la trasmissione.

Se invece il canale risulta occupato, resta in attesa fino a quando non rileva più il segnale

Durante la trasmissione verifica la presenza di eventuali segnali provenienti da altri terminali, se non ne rileva considera il pacchetto spedito

Se invece rileva segnali da altri adattatori (evento di **collisione**), interrompe immediatamente la trasmissione del pacchetto e invia un **segnale di disturbo (jam)** che ha una lunghezza di 48 bit

La scheda di rete calcola l'intervallo di backoff e se si è arrivati all'n-esima collisione consecutiva,

stabilisce un valore K tra $\{0,1,2,\dots,2^n - 1\}$ di attesa prima di riprendere

Intervallo di backoff: adatta il tempo di attesa al numero di nodi coinvolti nella collisione

Prima collisione: sceglie K tra $\{0,1\}$; il tempo di attesa è pari a K volte 512 bit.

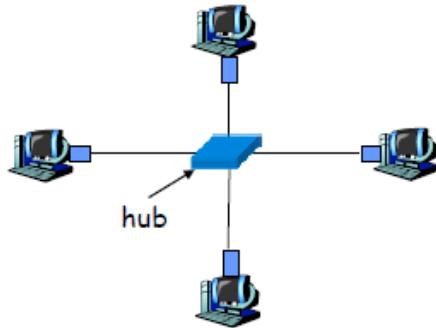
Dopo la seconda collisione: sceglie K tra $\{0,1,2,3\}\dots$

Dopo dieci collisioni, sceglie K tra $\{0,1,2,3,4,\dots,1023\}$

Esempi differenti tipi di rete, gigabit slide 11 pagina 22

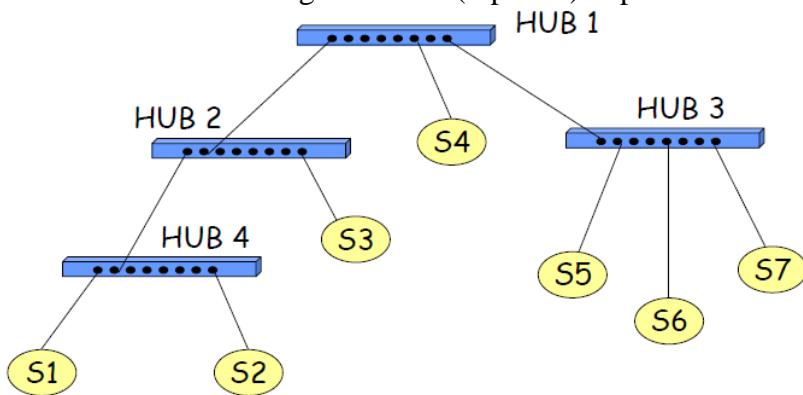
Hub

Ripetitore che opera allo strato fisico che rigenera il segnale analogico (re-shaping, retiming re-transmitting) e lo ritrasmette su tutte le interfacce, decodifica e ri-codifica il codice di linea, rileva collisioni e le inoltra su tutte le porte, isola segmenti di rete se si verificano 30 collisioni consecutive e permette di aumentare le dimensioni di una LAN rispettando il limite teorico imposto dal CSMA/CD e i limiti al numero massimo di ripetitori utilizzabili



Dominio di collisione: Sezione di rete in cui qualsiasi coppia di stazioni che trasmettono contemporaneamente generano una collisione.

La sezione di rete collegata da hub (repeater) fa parte di un unico dominio di collisione



Switch

Dispositivo che opera a livello di collegamento che esamina l'indirizzo MAC di destinazione e, se possibile, lo invia all'interfaccia corrispondente alla sua destinazione, collega diversi domini di collisione (reti collegate ad un hub) in cui le reti non sanno che c'è uno switch tra di loro.

Gli switch non hanno bisogno di essere configurati, apprendono autonomamente la topologia di rete e le regole di instradamento delle frame poiché viene collegato direttamente a ogni host

Permette trasmissioni simultanee tra 4 reti

Gli host hanno (normalmente) collegamenti dedicati e diretti con lo switch

Gli switch, se necessario, bufferizzano le frame

Il CSMA/CD è usato su ciascun collegamento in entrata, anche se non si verificano collisioni (collegamenti full duplex)

La trasmissione simultanea non è possibile con gli hub

Tabella di commutazione (Switch Table)

Ogni switch ha una tabella di commutazione che comprende l'indirizzo MAC di un nodo, l'interfaccia collegata al nodo e il timestamp.

Autoapprendimento: Lo switch apprende quali nodi possono essere raggiunti attraverso determinate interfacce, infatti ogni volta che un nodo gli manda frame, lo switch prende i dati del mittente e registra la coppia mittente/indirizzo nella sua tabella di commutazione (inizialmente vuota).

Gli switch possono essere interconnessi tra loro

Esempio slide 11 pagina 35

Adaptive Learning

In una rete statica il processo di apprendimento conduce ad uno stato in cui tutti gli indirizzi sono memorizzati nelle switch table

In situazioni pratiche, in una rete i nodi sono aggiunti, rimossi o spostati:

Si introduce un timeout che forza periodicamente l'apprendimento di ogni indirizzo.

Le informazioni che non vengono rinfrescate da molto vengono cancellate dopo un tempo massimo.
(ageing time 300 s valore consigliato dallo standard)

Se una frame arriva su una porta che differisce da quella memorizzata nella switch table, questa viene aggiornata immediatamente

Se trova la destinazione nella tabella = **Selective Send**, se invece non la trova allora la inoltra a tutti = **Flood**

Switch e router a confronto

Entrambi sono di tipo store-and-forward.

I **Router** sono attivi a livello di rete e mantengono le tabelle di routing e implementano algoritmi di instradamento, mentre gli **Switch** sono attivi a livello di collegamento e mantengono le tabelle di commutazione e implementano il filtraggio e algoritmi di autoapprendimento.

Protocollo Spanning Tree (STP)

Regola il processo di forwarding in presenza di loop nella rete.

Da una rete a maglia crea un albero e lo tiene in memoria.

L'algoritmo viene avviato ogni volta che si rileva un cambiamento nella topologia della rete.

Operazioni: Inizialmente si determina lo switch radice (**Root Bridge**): all'inizio tutti si credono root bridge e mandano trame e messaggi (in particolare **BPDU**(bridge protocol data unit)), in cui mettono nel campo root id il loro indirizzo MAC. Quando ricevono una trama con l'id diverso dal loro lo confrontano. Se l'id è minore del proprio lo definiscono root, altrimenti continuano a credersi root.

Alla fine il root sarà quello con l'id minore.

Determina poi per tutti gli altri switch la **Root Port** (la porta che li conduce al Root Bridge): una volta eletta la root bridge ogni switch elegge la porta a costo minimo per raggiungere il root bridge. Seleziona la **designated port**, la porta per ogni lan per ricevere e trasmettere frame: una volta che gli switch hanno scelto le loro porte con i costi per raggiungere il root bridge, le lan sceglieranno lo switch che raggiunge il root a costo minore.

Tutte le altre porte di lan e switch vengono disabilitate

Le **BPDU** (Bridge Protocol Data Unit) contengono:

Root id: identificativo del root bridge candidato a diventare il root bridge

Switch id: identificativo del bridge emittente

Root path cost: costo totale per raggiungere root bridge (posto a 0 dal Root Bridge e aggiornato da ogni altro switch)

Flag: con il Topology Change Notification = 1 trasmesso se cambia qualcosa nella topologia della rete verso il Root Bridge

ESEMPIO STP SLIDE 11 pagine 55-60

STP: cambiamenti di topologia

Cambiamenti della topologia vengono notificati al Root Bridge attraverso **Topology Change Notification BPDU**

Il Root Bridge invia Conf. BPDU con flag TC = 1 verso tutti gli altri bridge

I bridge reagiscono al cambiamento della topologia impostando il timer ageing-time al valore forward delay (trasportato nelle Conf. BPDU ... raccomandato 15 s)

	Octet
Protocol Identifier	1
	2
Protocol Version Identifier	3
BPDU Type	4

Topology Change Notification BPDU

Posizione dello Switch?

Accessi dedicati: Utilizzo pesante e continuativo di risorse di rete (server)

Accessi condivisi: Stazioni che generano traffico discontinuo

STRATO FISICO (PARTE 1) CAPITOLO 12

Trasmissione Digitale.

Due modi per trasmettere messaggi:

Informazione a blocchi – il messaggio è strutturato in unità indipendenti (blocchi), con un numero di bit per blocco fisso. (Text message, Data file, JPEG image, MPEG file)

Informazione Stream – trasmissione continua di bit (es. streaming).

Esse poi possono avere:

Bit Rate = frequenza di bit/s che riesco a trasmettere

- **Costant Bit Rate (CBR)** – flussi a bit rate costante (la rete deve fornire un canale con banda minima uguale al bit rate)
- **Variable Bit Rate (VBR)** – flussi con bit rate variabile nel tempo (la rete deve supportare la variabilità)

Delay di trasferimento di un messaggio

L = Numero di bit in un messaggio, si riduce mediante tecniche di compressione

R = Velocità del sistema di trasmissione (bit/s), si aumenta mediante adeguate tecniche di trasmissione

T_{prop} = tempo di propagazione lungo il mezzo trasmissivo

d = lunghezza del collegamento, si riduce riducendo la lunghezza del collegamento

c = velocità di propagazione sul mezzo trasmissivo (3×10^8 m/s nel vuoto, 2×10^8 m/s nei mezzi guidati)

$$\text{Delay minimo: } T_{\text{prop}} + \frac{L}{R} = \frac{d}{c} + \frac{L}{R}$$

Compressione

Si riducono i bit necessari a rappresentare l'informazione, riducendo la ridondanza.

Senza perdita (Lossless): l'informazione originale è ricostruita esattamente: zip, GIF, fax

Con perdita (lossy): l'informazione decompressa non è identica all'originale: JPEG

Rapporto di compressione = $R_c = \frac{B_{orig}}{B_{compr}} \left(\frac{\#bit\ originali}{\#bit\ compressi} \right)$, con $R > 1$ per non

perdere informazioni, più è grande R e più efficiente sarà la mia compressione.

Digitalizzazione dell'analogico.

Un segnale vocale nella forma originale è di tipo analogico ed esso deve essere digitalizzato e trasmesso in tempo reale

Un segnale analogico varia nel tempo quindi ci saranno momenti in cui userò più bit e altri meno.

Campionamento: Divido il segnale in livelli. Ogni tot tempo prendo un campione (*sampling*) e lo approssimo al livello più vicino. A seconda del livello sul quale cade gli assegno un certo numero di bit. Il numero di bit dipende dal numero di livelli in cui divido il segnale. (3 bit = 2^3 8 livelli, 8 bit = 2^8 256 livelli)

Il bit rate **BR** in questo caso diventerà $\frac{\text{numerodibit}}{\text{campione}} * \frac{\text{numerodicampioni}}{\text{secondo}}$ e da esso dipenderà la qualità del segnale.

Teorema del campionamento

La larghezza di Banda (Bandwidth) Ws(Hz) indica quanto velocemente il segnale varia nel tempo. Più essa è grande più dovrò prendere campioni frequentemente per avere buona qualità.

ESEMPIO VOCE E AUDIO SLIDE 12 PAGINA 6

SEGNALE VIDEO SLIDE 12 PAGINE 7-9

Si giunge così al fatto che la frequenza di campionamento minima per preservare le informazioni è:

$$F_c = 2 \times W_s \text{ e da questo ottengo che } T_c = \frac{1}{F_c} \text{ è il tempo di ogni quanto campiono.}$$

Sebbene ci sia un minimo, se campiono più frequentemente del dovuto non ho un guadagno.

Parametri di qualità per servizi di tipo Stream

Possibili problemi introdotti dal transito in rete (**Network Impairment**)

Ritardo (Delay): Per ogni servizio occorre individuare il vincolo sul ritardo massimo di attraversamento della rete

Variabilità del ritardo (Jitter): Per ogni servizio occorre individuare il vincolo sulla variabilità massima consentita del ritardo di attraversamento della rete

Perdita di informazioni (Loss): Per ogni servizio occorre individuare il vincolo sulla percentuale massima di bit persi (per errori o congestione) sul totale dei bit trasmessi

Sistema di Trasmissione

Trasmettitore: converte flusso informativo prodotto da una sorgente in un **segnale** adatto alla trasmissione

Canale di Comunicazione: Cavi, Fibra, Radio, interposto tra trasmettitore e ricevitore

Ricevitore: converte il segnale ricevuto in forma utilizzabile

Possibili alterazioni: Attenuazione del segnale / Distorsione segnale / Rumore additivo / Interferenza con altri segnali

Posso trasmettere

Segnale Analogico in cui il ricevente deve ricostruire tutti i dettagli, maggiore possibilità di errore, limite di distanza.

Nel caso di trasmissioni analogiche a lunga distanza devo far uso di **ripetitori**.

Ogni ripetitore ha lo scopo di **rigenerare** il segnale in uscita in modo che sia quanto più possibile simile a quello ricevuto in ingresso

La rigenerazione è non ideale: le distorsioni non sono completamente eliminabili, il rumore e le interferenze sono solo parzialmente rimosse, per questo la qualità del segnale diminuisce al crescere del numero di ripetitori

Segnale Digitale in cui il ricevente deve ricostruire solo i livelli discreti del segnale, probabilità di errore piccola, possibili comunicazioni a lunga distanza.

Nel caso di trasmissioni analogiche a lunga distanza devo far uso di **rigeneratori**.

Un rigeneratore ricostruisce la sequenza iniziale di bit e la ritrasmette sulla tratta successiva

Il segnale rigenerato è in pratica identico a quello originale

Trasmissione ad impulsi

Obiettivo: Rendere massimo il rate di trasmissione degli impulsi in un canale, ovvero rendere T il più piccolo possibile facendo attenzione che gli impulsi brevi e ravvicinati potrebbero sovrapporsi (**interferenza intersimbolica**).

Esiste un minimo di frequenza affinché questo non accada:

Frequenza di Nyquist: $F = 2 \times W_c$ dove W_c è la larghezza della banda del canale(**bandwidth**)

Se non c'è interferenza intersimbolica e non c'è rumore il massimo rate di trasmissione è $R_{max} = 2W_c$

Larghezza di banda di un canale trasmissivo

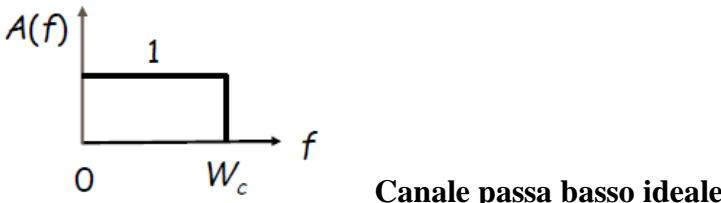
Se il segnale di ingresso ad un canale è una sinusoide di frequenza f allora l'uscita sarà una sinusoide della stessa frequenza f attenuata di un fattore $A(f)$ che dipende dalla frequenza f

Segnale trasmesso: $X(t) = a \cos(2\pi ft)$

Segnale ricevuto: $Y(t) = A(f) a \cos(2\pi ft)$

$A(f) \approx 1 \Rightarrow$ il segnale transita inalterato; $A(f) \approx 0 \Rightarrow$ il segnale è bloccato

La larghezza di banda W_c è definita come l'intervallo di frequenze per cui $A(f) \approx 1$



Trasmissione multilivello (PAM)

Si consideri un canale con larghezza di banda W_c e una trasmissione ad un rate $2W_c$ impulsi/sec (senza interferenza), se l'ampiezza degli impulsi può assumere due valori (-A o +A), ogni impulso può rappresentare un solo bit informativo, quindi:

$$\text{Bit Rate} = 1 \text{ bit/impulso} \times 2W_c \text{ impulsi/sec} = 2W_c \text{ bit/s}$$

Se l'ampiezza degli impulsi può assumere valori appartenenti all'insieme $\{-A, -A/3, +A/3, +A\}$, ogni impulso può rappresentare 2 bit quindi:

$$\text{Bit Rate} = 2 \text{ bit/impulso} \times 2W_c \text{ impulsi/sec} = 4W_c \text{ bit/s}$$

Se il segnale può assumere $M = 2^m$ livelli, si ha:

$$\text{Bit Rate} = m \text{ bit/impulso} \times 2W_c \text{ impulsi/sec} = 2^m W_c \text{ bit/s}$$

In assenza di rumore il bit rate può essere incrementato aumentando il valore di m (livelli del segnale)
Attenzione: aumentando m si riduce la distanza tra livelli adiacenti

SPIEGATO MEGLIO

Raggruppa i bit in parole di dimensione $N = \log_2 M$

M : numero di livelli; N : numero di bit trasmessi in un unico impulso;

Assegna ad ogni parola di N bit un livello tra gli M disponibili:

I livelli adiacenti corrispondono a parole di codice che differiscono per un solo bit

Un errore tra due livelli adiacenti comporta un errore su un solo bit

Se il segnale può assumere $M = 2^m$ livelli, si ha un Bit Rate uguale a:

$$m \text{ bit/impulso} \times 2W_c \text{ impulsi/sec} = 2^m W_c \text{ bit/s}$$

Il bit rate può essere aumentato incrementando il numero di livelli, tuttavia il segnale include il rumore additivo che limita il numero di livelli che possono essere usati.

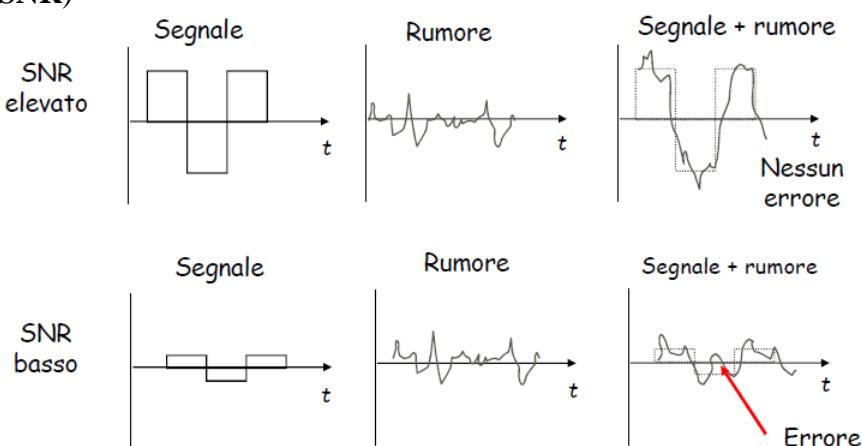
Rumore

La presenza di rumore limita l'accuratezza della misura dell'ampiezza del segnale ricevuto

L'effetto del rumore è modellabile come un **segnale additivo** rispetto al segnale utile

Gli errori nella rivelazione del segnale ricevuto appaiono quando la separazione tra i livelli del segnale è comparabile con il livello di rumore

Il **Bit Error Rate (BER)** aumenta quando diminuisce il rapporto segnale-rumore (**signal-to-noise ratio SNR**)



$$SNR = \frac{\text{Potenza media del segnale}}{\text{potenza media del rumore}} \quad SNR(\text{dB}) = 10 \log_{10} (\text{SNR})$$

Se l'SNR è elevato il segnale ha un rumore trascurabile, se è basso l'errore è evidente.

Limite di Shannon alla capacità di un canale

Dato un canale con banda W e rumore Gaussiano e fissato un valore di SRN, il massimo rate di trasmissione raggiungibile per cui è ottenibile un BER arbitrariamente piccolo è dato da:

$C_{\max} = W_c \log_2 (1+SNR)$ bit/s che è una funzione della larghezza di banda e del rapporto segnale/rumore.

Gli errori nella rivelazione del segnale ricevuto appaiono quando la separazione tra i livelli del segnale è comparabile con il livello di rumore

Il **Bit Error Rate (BER)** = $\frac{\text{numero di bit giusti}}{\text{numero di bit complessivi}}$ aumenta quando diminuisce l'SNR

Il rumore pone un limite al numero di livelli che possono essere utilizzati nella trasmissione di impulsi e quindi un limite al bit rate in trasmissione

Se il bit rate di trasmissione R è inferiore a C_{\max} ($R < C_{\max}$) è possibile ottenere un BER arbitrariamente piccolo.

Se $R > C_{\max}$, non è possibile ridurre il BER a valori arbitrariamente piccoli

La capacità C_{\max} può essere utilizzata come una misura di riferimento per stabilire quanto un sistema di trasmissione reale è vicino alle migliori prestazioni possibili

STRATO FISICO (PARTE 2) CAPITOLO 13

Potenza di un segnale $x(t)$



$$P_x = \lim_{\Delta t \rightarrow \infty} \frac{1}{\Delta t} \int_{-\frac{\Delta t}{2}}^{\frac{\Delta t}{2}} |x(t)|^2 dt \geq 0$$

Un segnale è detto **di potenza** se $0 < P_x < \infty$

VEDERE SLIDE 13, per me roba inutile

POTENZA DI UN SEGNALE FORMULE

Sviluppo in serie di Fourier per un segnale periodico

Teorema di Parseval

Trasformata di Fourier: Il generico segnale è costituito da una composizione di sinusoidi che rappresentano le frequenze. Maggiori sono le frequenze, maggiore è il segnale che si avvicina all'originale.

Digitalizzazione di segnali analogici

Campionamento: estrazione di campioni del segnale $x(t)$ uniformemente spaziati nel tempo

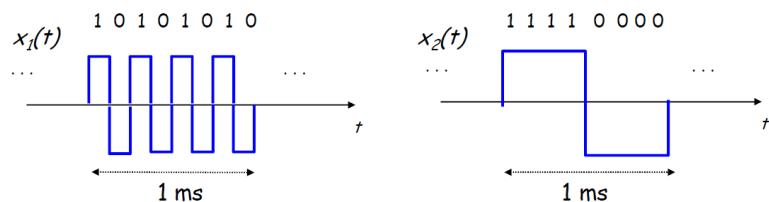
Quantizzazione: codifica di ogni campione con una stringa di bit (con precisione finita)

Compressione: applicazione di metodi di riduzione del bit rate

Frequenza di campionamento e larghezza di banda

Segnali che variano più velocemente nel tempo devono essere campionati con maggiore frequenza

Larghezza di banda (Bandwidth): misura quanto velocemente varia un segnale



Segnali periodici

Un segnale reale periodico di periodo T può essere rappresentato come somma di sinusoidi usando lo sviluppo in serie di Fourier:

$$x(t) = a_0 + a_1 \cos(2\pi f_0 t + \phi_1) + a_2 \cos(2\pi 2f_0 t + \phi_2) + \dots + a_k \cos(2\pi kf_0 t + \phi_k) + \dots$$

"Componente continua";
media a lungo termine

Frequenza
fondamentale $f_0 = 1/T$
(prima armonica)

k -ma armonica

$|a_k|$ determina la potenza della k -ma armonica

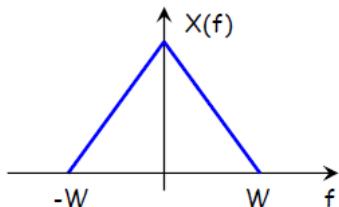
Spettro di ampiezza = $\{|a_0|, |a_1|, |a_2|, \dots\}$

Spettro & Bandwidth di un segnale

Lo **Spettro** di un segnale è rappresentato dalle ampiezze di ciascuna componente di frequenza

La larghezza di banda (Bandwidth) W_s di un segnale è definita come l'intervallo di frequenze del segnale che hanno potenza non trascurabile

Un segnale reale $x(t)$ si dice limitato in banda $[-W, W]$ se la sua trasformata di Fourier $X(f)$ è nulla per $f \notin [-W, W]$ dove W è la Larghezza di banda del segnale $x(t)$

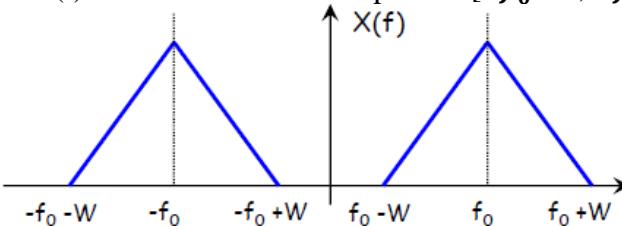


La quantità W è definita come la **Larghezza di Banda** del segnale $x(t)$

Poiché $X(f) \neq 0$ in un intorno $[-W, W]$ di $f = 0$, il segnale $x(t)$ si dice **segnale di banda base**

Un segnale reale $x(t)$ si dice limitato in banda, con banda $2W$ centrata intorno alla frequenza f_0 se:

- $f_0 > W$
- $X(f)$ è identicamente nulla per $f \notin [-f_0 - W, -f_0 + W] \cup [f_0 - W, f_0 + W]$



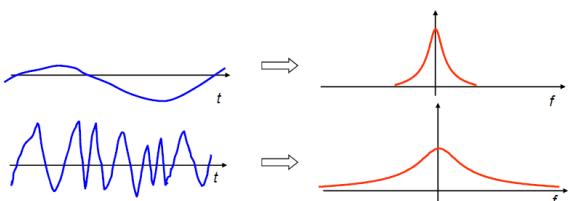
La quantità $2W$ è la larghezza di banda del segnale $x(t)$

Poiché $X(f) \neq 0$ in un intorno di $\pm f_0$ non adiacente all'origine, il segnale $x(t)$ si dice **segnale in banda traslata**

Relazioni tempo-frequenza

Segnali lentamente varianti in $t \rightarrow$ banda stretta (in f)

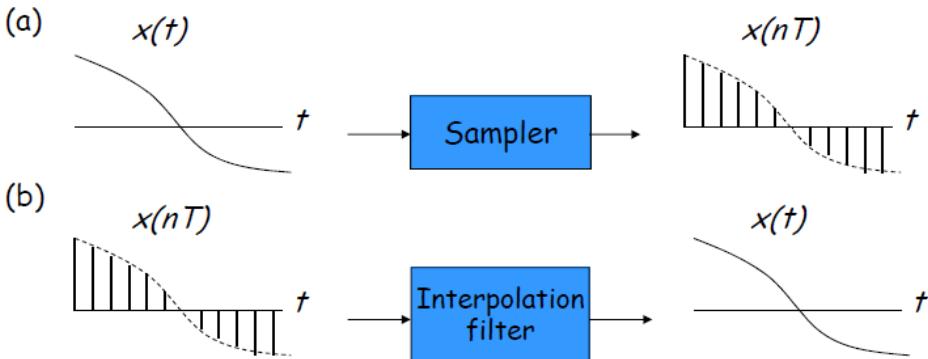
Segnali rapidamente varianti in $t \rightarrow$ banda larga (in f)



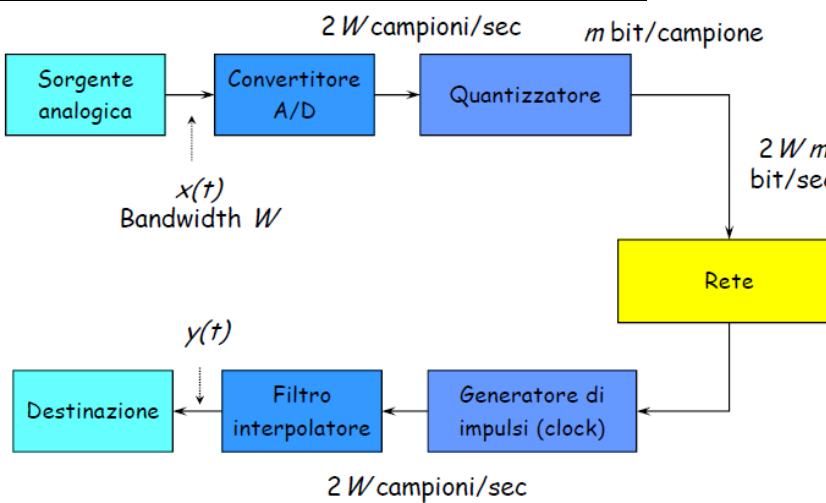
Teorema del campionamento

Un segnale limitato in banda W_s può essere perfettamente ricostruito a partire dalla sequenza dei suoi campioni se la frequenza di campionamento:

$$F_c = \frac{1}{T} > 2W_s \quad (\text{Frequenza di Nyquist})$$

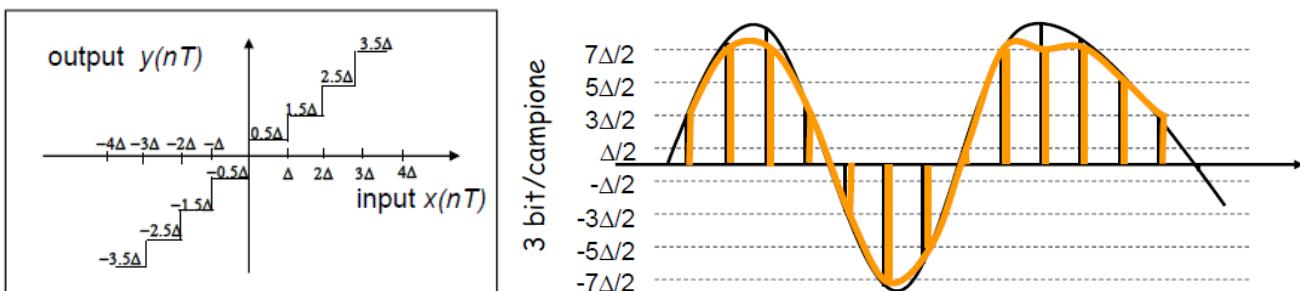


Trasmissione digitale di informazioni analogiche



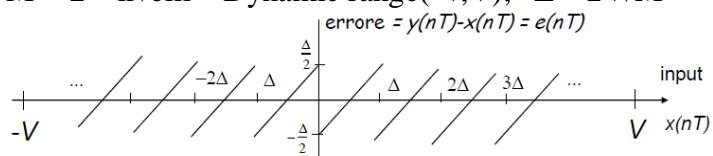
Quantizzazione di segnali analogici

Il quantizzatore associa il valore di ingresso al valore 2^m (livello) più vicino
 Errore di quantizzazione (rumore di quantizzazione) = $y(nT) - x(nT)$



Prestazioni del quantizzatore

$M = 2^m$ livelli Dynamic range (-V, V); $\Delta = 2V/M$



Se il numero di livelli M è sufficientemente elevato, allora l'errore è uniformemente distribuito tra $(-\Delta/2, \Delta/2)$

Potenza media del rumore di quantizzazione (Errore quadratico medio):

$$\sigma_e^2 = \frac{1}{\Delta} \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} x^2 dx = \frac{\Delta^2}{12}$$

Sia σ_x^2 potenza del segnale si ha:

$$SNR = \frac{\sigma_x^2}{\sigma_e^2} = \frac{\frac{12\sigma_x^2}{\Delta^2}}{\frac{4V^2}{M^2}} = 3 \left(\frac{\sigma_x}{V} \right)^2 M^2 = 3 \left(\frac{\sigma_x}{V} \right)^2 2^{2m}$$

usualmente $\frac{V}{\sigma} = 4$, quindi esprimendo SNR in dB:

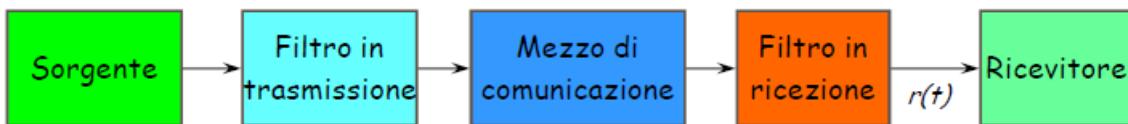
$$SNR(dB) = 10 \log_{10} \frac{\sigma_x^2}{\sigma_e^2} = 10 \log_{10} (3 \cdot 4^{-2} \cdot 2^{2m}) = 6m - 7.27 \text{ dB}$$

STRATO FISICO (PARTE 3) CAPITOLO 14

Canali di comunicazione

Unione dei mezzi trasmittivi e dei dispositivi (elettronici o ottici) che sono attraversati dal segnale lungo il percorso tra sorgente e destinazione: Equalizzatori, amplificatori, ecc.

Spesso si usa il termine **filtro** per indicare gli effetti del canale sul segnale che lo attraversa



Canale: set di sinusoidi che passano inalterate nel canale

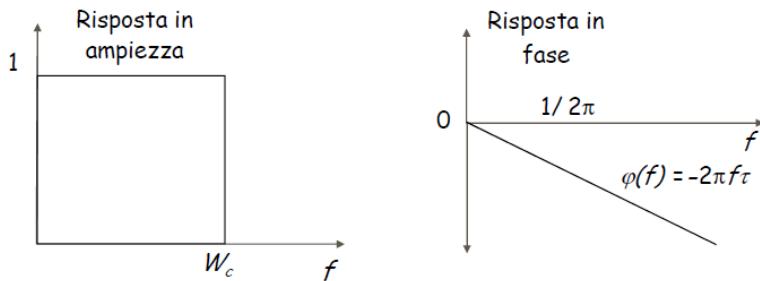
Segnale: set di ampiezze che sono lo spettro del segnale. La parte più alta della banda canale costituisce la banda segnale.

Canale con banda infinita: canale sul quale passano tutte le frequenze ma non è ideale perché gli impulsi (funzione che trasporta i bit) si allargano e la coda di un segnale finisce sull'inizio di un altro ottenendo un'interferenza intersimbolica.

Filtro passa basso ideale

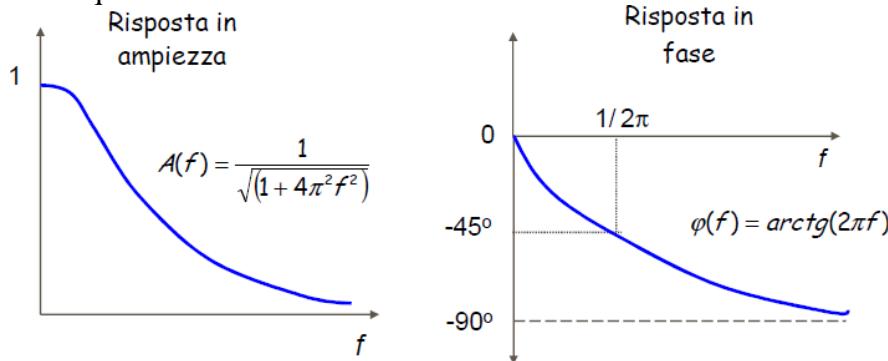
Tutte le frequenze $f < W_c$ non subiscono attenuazione e sono ritardate di τ secondi, invece le frequenze $f > W_c$ vengono bloccate

$$y(t) = A_{in} \cos (2\pi f t - 2\pi f \tau) = A_{in} \cos (2\pi f (t - \tau)) = x(t - \tau)$$



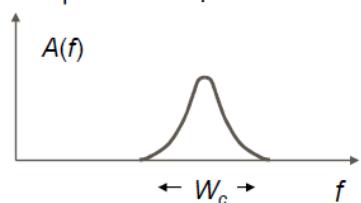
Filtro passa basso reale

Le frequenze sono attenuate in modo diverso e subiscono ritardi diversi



Canale passabanda

Amplitude Response



Alcuni canali di comunicazione si comportano come un filtro passa-banda quindi **bloccano le basse e le alte frequenze**

La larghezza di banda è l'ampiezza dell'intervallo di frequenze per cui il segnale in uscita ha una potenza non trascurabile

Distorsione

$$x(t) = \sum a_k \cos(2\pi f_k t + \theta_k) \rightarrow \text{Canale} \rightarrow y(t)$$

Il canale introduce sul segnale in ingresso $x(t)$ due effetti:

- Se la risposta in **frequenza** non è “piatta”, le componenti di frequenza del segnale d’uscita $y(t)$ avranno ampiezza diversa rispetto a quelle del segnale d’ingresso $x(t)$
- Se la risposta in **fase** non è “piatta”, le componenti di frequenza del segnale d’ingresso $x(t)$ subiranno ritardi diversi

$$y(t) = \sum A(f_k) a_k \cos[2\pi f_k t + \theta_k + \Phi(f_k)]$$

Esempio: Distorsione di ampiezza capitolo 14 pagine 4-5

Risposta impulsiva di un sistema lineare

La risposta impulsiva $h(t)$ di un sistema lineare e permanente (filtro) è definita come l’uscita $y(t)$ del sistema quando all’ingresso è applicato l’impulso unitario $x(t) = \delta(t)$

La larghezza della risposta impulsiva fornisce un’indicazione di quanto velocemente l’uscita segue l’ingresso e quindi di quanto velocemente possono essere trasmessi gli impulsi in ingresso

FORMULE + CONVOLUZIONE pagine 7-8

Risposta impulsiva di un filtro ideale

Per canali ideali passa basso di larghezza di banda W_c , la risposta impulsiva è rappresentata dalla funzione impulso di Nyquist $h(t) = s(t - \tau)$, dove $T = \frac{1}{2} W_c$, e $s(t)$ vale zero in $t = kT$, $k = \pm 1, \pm 2, \dots$

CAPITOLO 15 ASSEGNAZIONE RISORSE

Pre-assegnazione

- **Individuale** (ed esempio su base banda di picco) si impiega nel modo di trasferimento a circuito
- **Collettiva** (ad esempio su base banda media) si impiega nel modo di trasferimento a pacchetto con connessione

Assegnazione a domanda: si impiega nel modo di trasferimento a pacchetto senza connessione

Si indica con C la capacità della linea e con N il numero di sorgenti multiplate

Pre-assegnazione su base banda di picco di una linea di capacità C

Le sorgenti vengono caratterizzate mediante il loro **ritmo binario di picco** F_p

Affinché le sorgenti possano essere multiplate sulla linea deve essere sempre soddisfatta la relazione:

$$\sum_{i=1}^N F_{p,i} \leq C$$

Pre-assegnazione su base banda media di una linea di capacità C

Le sorgenti vengono caratterizzate mediante il loro **ritmo binario medio** F_m

Affinché le sorgenti possano essere multiplate sulla linea deve essere sempre soddisfatta la relazione:

$$\sum_{i=1}^N F_{m,i} \leq UC \quad \text{con } U < 1$$

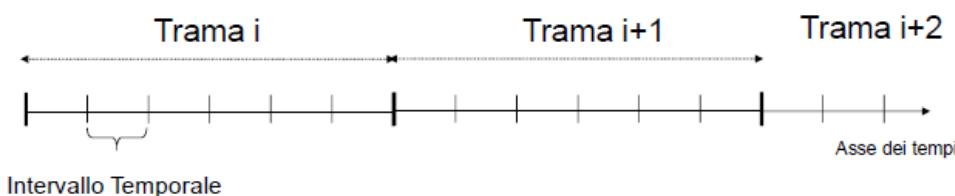
U rappresenta il rendimento di utilizzazione massimo della linea multiplata

Schemi di multiplazione statica

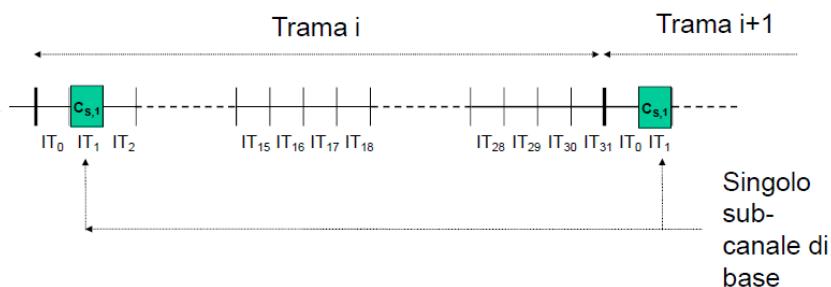
Le risorse vengono pre-assegnate in modo individuale

La linea multiplata è divisa in **Intervalli Temporali** (IT) organizzati in trame

In genere ad una sorgente viene assegnato un IT ripetuto a cadenza di trama



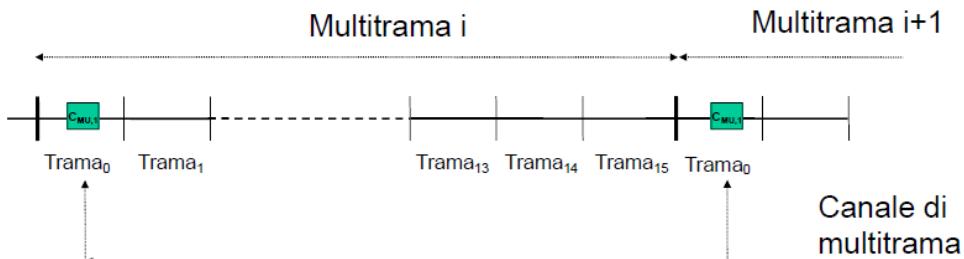
Sub-canali di base



Ogni sub-canale di base ha capacità:

$$C_s = \frac{Nbit_{IT}}{T_{trama}} \text{ nel caso del PCM europeo } C_s = \frac{8 \text{ bit}}{125 \mu\text{s}} = 64 \text{ kbit/sec}$$

Multitrama



Multitrama = struttura organizzativa superiore alla trama costituita da M trame

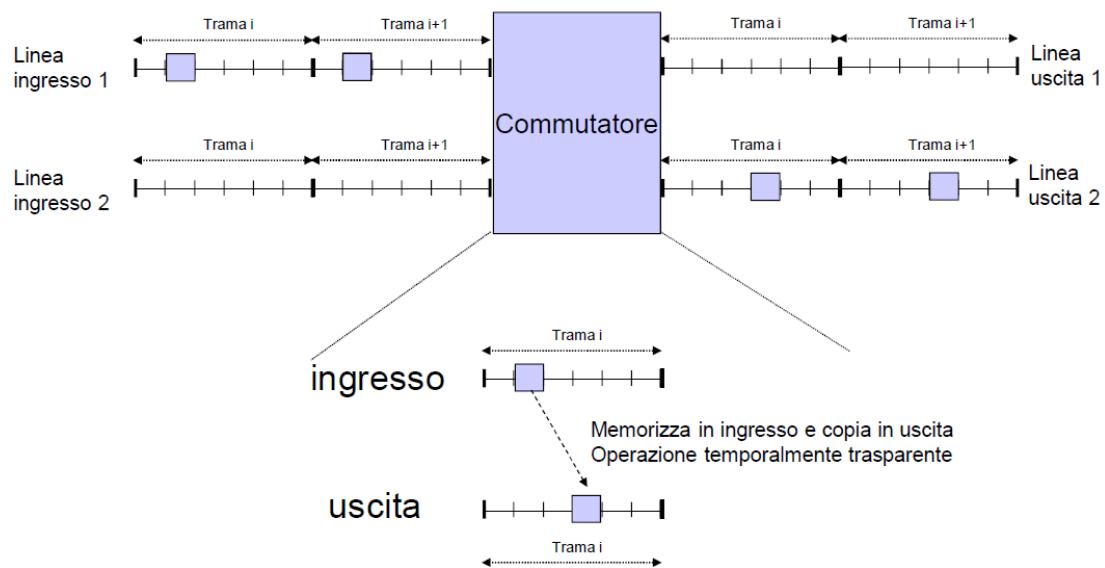
Un canale di multitrama è definito da un IT ripetuto a cadenza di multitrama e ogni canale di multitrama ha capacità:

$$C_{MU} = \frac{Nbit_{IT}}{M * T_{trama}}$$

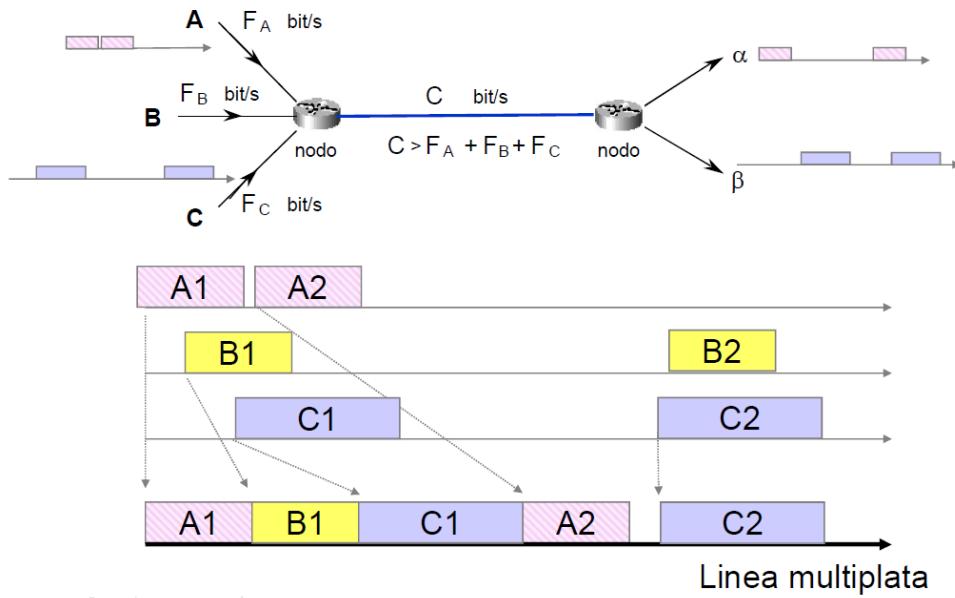
Nel PCM Europeo la segnalazione associata al canale è convogliata in una struttura a multitrama con $M = 16$, l'informazione di segnalazione occupa in ogni trama della multitrama l' IT_{16}

8 bit dell' IT_{16} della prima trama nella multitrama portano la parola di allineamento di multitrama
4 bit dell' IT_{16} di ogni trama vengono associati ad un singolo canale fonico
ogni singolo canale di segnalazione ha capacità di 4 bit / $125\mu\text{s} * 16 = 2 \text{ Kbit/s}$

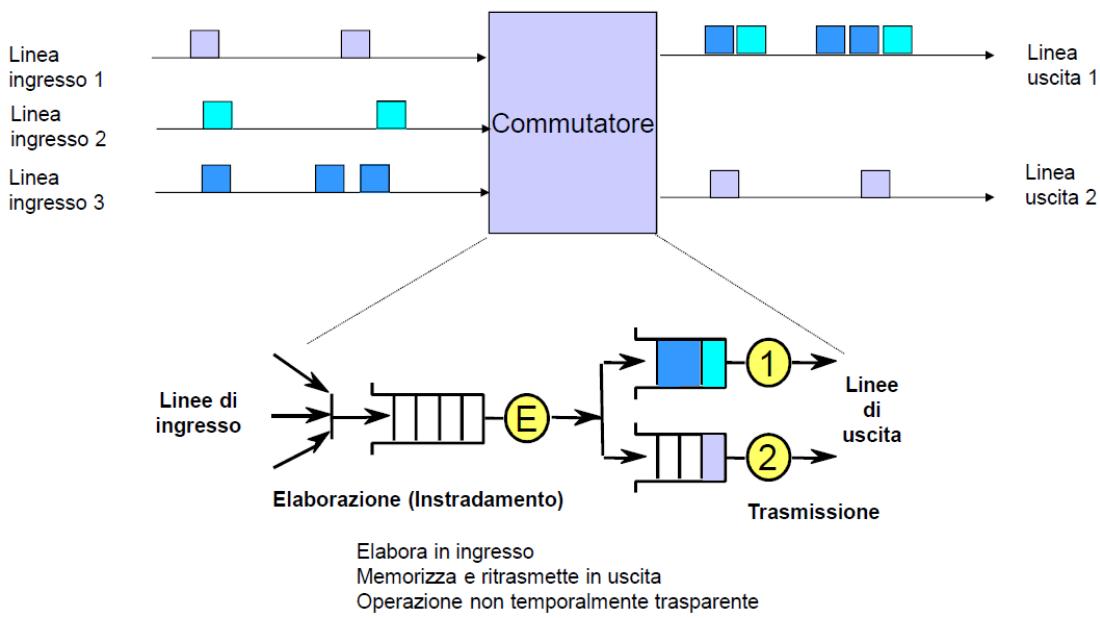
Schema di commutazione (con connessione diretta)



Schemi di multiplazione dinamica

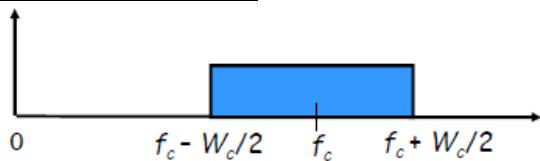


Schema di commutazione (ad immagazzinamento e rilancio)



STRATO FISICO (PARTE 4) CAPITOLO 16

Canali passa-banda



Ammettiamo di avere un segnale ad una certa frequenza, ma vogliamo spostarlo e centrarlo attorno ad una frequenza che chiamiamo Frequenza Portante.

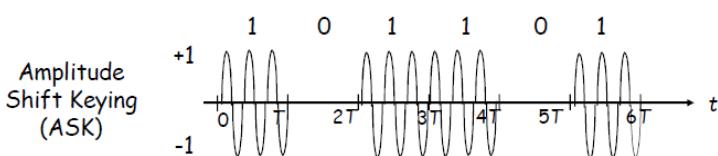
Quest'operazione si chiama **Modulazione in Banda** ed il canale che passa per frequenze tutte centrate attorno ad f_c viene chiamata non più Canale Passa Basso ma Canale Passa Banda.

I dispositivi che effettuano la modulazione vengono chiamati **Modulatori Numerici** (es Modem) ed essi moltiplicano la frequenza base per la sinusode $\cos(2\pi f_c t)$

Quest'operazione è necessaria perché ogni canale ha una frequenza portante differente, e devo poter trasmettere il mio segnale a canali diversi con f_c diversi.

Quest'operazione diminuisce il limite della velocità della banda che invece di essere $2W_c$ diventa W_c .

Modulazione di Ampiezza (ASK)

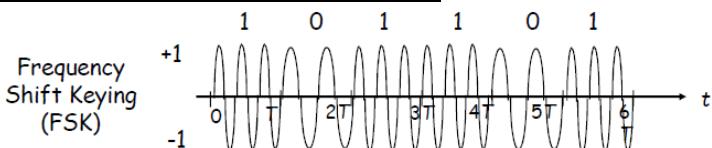


Mappa ogni bit informativo nell'ampiezza di una sinusode a frequenza f_c :

- “1” trasmissione del segnale sinusoidale
- “0” nessun segnale

Il demodulatore individua i periodi in cui è presente il segnale e i periodi in cui il segnale è assente

Modulazione di Frequenza (FSK)

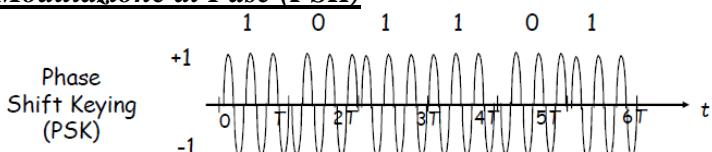


Mappa ogni bit informativo nella frequenza di un segnale sinusoidale

- “1” trasmissione di un segnale di frequenza $f_c + \delta$
- “0” trasmissione di un segnale di $f_c - \delta$

Un demodulatore individua la potenza intorno alle frequenze $f_c + \delta$ oppure $f_c - \delta$

Modulazione di Fase (PSK)

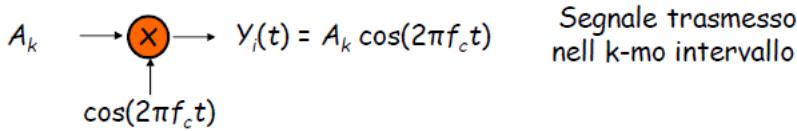


Mappa ogni bit informativo nella fase di un segnale sinusoidale:

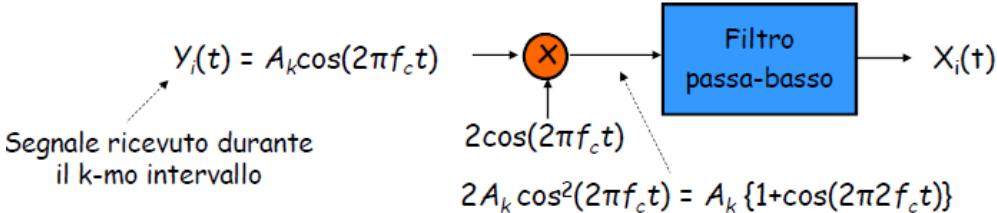
- “1” trasmissione del segnale $A \cos(2 \pi f t) \rightarrow$ fase 0
- “0” trasmissione del segnale $A \cos(2 \pi f t + \pi) \rightarrow$ fase π

Modulazione e Demodulazione PSK

Un segnale $\cos(2\pi f_c t)$ viene **modulato** moltiplicandolo per A_k per T secondi (durata di un simbolo):



Il segnale ricevuto viene **demodulato** moltiplicandolo per $2\cos(2\pi f_c t)$ per T secondi e successivamente filtrandolo con un filtro passa-basso



ESEMPIO DI MODULAZIONE E DEMODULAZIONE SLIDE 16 PAGINA 4

Banda in trasmissione

Se il segnale in banda base $x(t)$ ha banda $W_c/2$ Hz, il segnale modulato $x(t) \cos(2\pi f_c t)$ ha banda uguale a W_c Hz

Se il canale di comunicazione ha banda W_c Hz:

- Il canale in banda base ha una larghezza di banda disponibile uguale a $\frac{W_c}{2}$ Hz
- Un sistema di modulazione supporta $(\frac{W_c}{2}) \times 2 = W_c$ impulsi/secondo

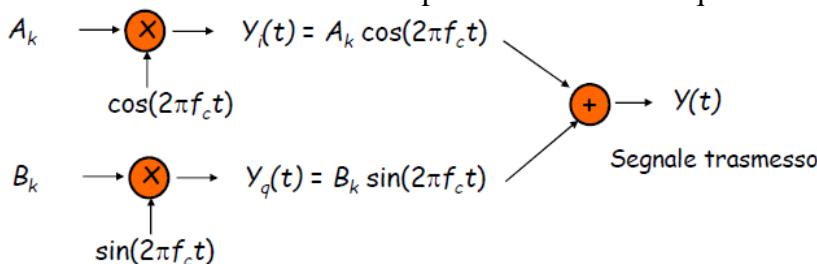
Quindi W_c impulsi/secondo per W_c Hz = 1 impulso/Hz, si ricorda che la trasmissione in banda base supporta 2 impulsi/Hz

QAM (Quadrature Amplitude Modulation)

Si tratta di una particolare modulazione che permette di trasmettere due segnali diversi (due impulsi)

Un segnale A_k **modulato in Fase** come abbiamo visto, ed il secondo B_k **modulato in quadratura**.

Si trasmette la somma delle componenti in fase ed in quadratura: $Y_i(t) + Y_q(t)$

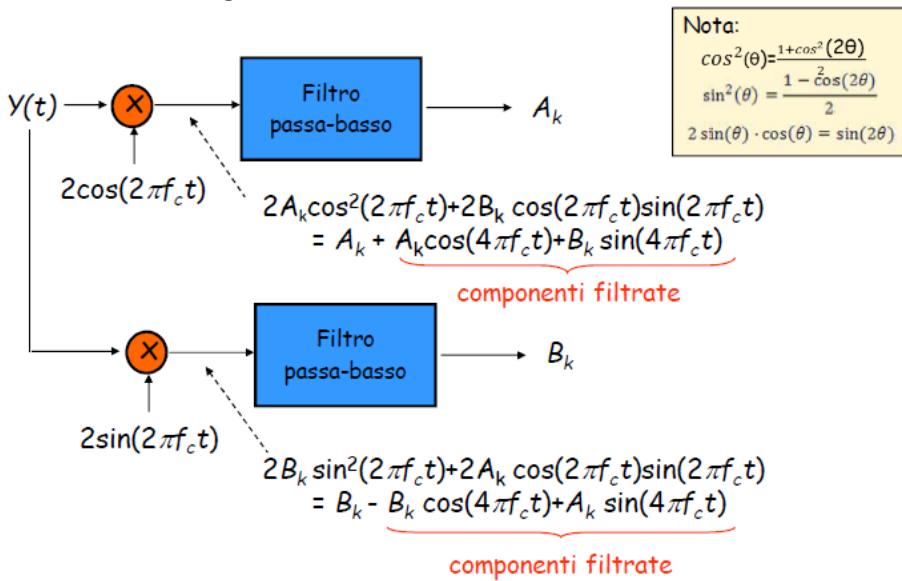


I segnali $Y_i(t)$ e $Y_q(t)$ occupano entrambi la banda passante del canale, la modulazione QAM supporta 2 impulsi/Hz

Questo è possibile perché il seno ed il coseno sono ortogonali tra loro e quindi possono viaggiare insieme comandandoli.

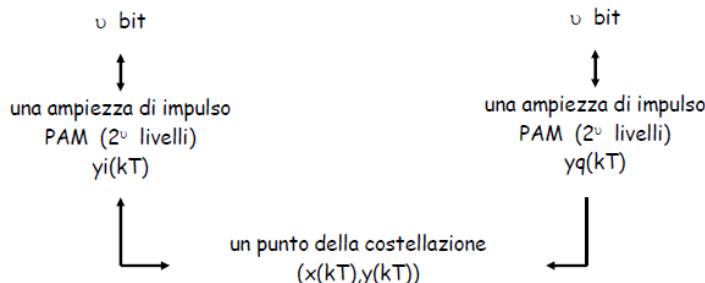
L'operazione di traslazione alla frequenza portante mi fa perdere metà della velocità della banda traslata e da W_c il limite diventa $\frac{W_c}{2}$, ma inviando poi due impulsi nella stessa trasmissione recupero la perdita.

Demodulazione OAM



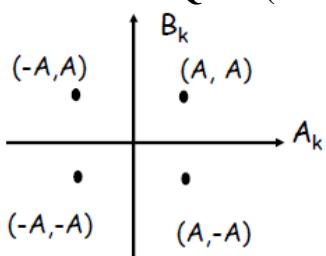
Costellazione dei Segnali

Ogni T secondi vengono trasmessi $2v$ bit del segnale di ingresso

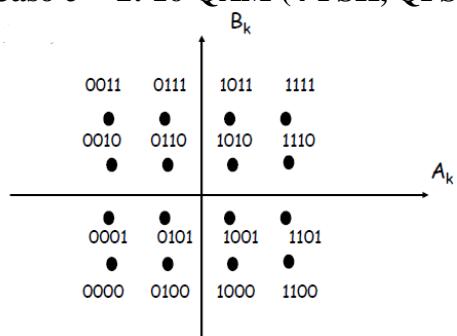


Ogni coppia (A_k, B_k) definisce un punto nel piano

La **costellazione** di un segnale è l'insieme dei punti che può assumere un segnale
Caso $v = 1$: 4-QAM (4-PSK, QPSK), 4 possibili punti in T sec (2 bit/impulso)



Caso $v = 2$: 16-QAM (4-PSK, QPSK), 16 possibili punti in T sec (4 bit/impulso)



Effetto del rumore

Si individua nel piano del signal set delle regioni di decisione associate ai punti della costellazione
La generica regione di decisione associata a un punto P è costituita da tutti i punti del piano più vicini a P che a tutti gli altri punti del signal set.

Si ha una decisione errata (corrispondente a uno o più bit errati nel segnale binario demodulato) quando il rumore è tale da far cadere il punto ricevuto R al di fuori della regione di decisione relativa al punto trasmesso P.

STRATO DI COLLEGAMENTO (PARTE 1) CAPITOLO 17

In una rete esistono diversi nodi (Host, Router) collegati da dei **link** che sono i canali di comunicazione.

Le unità dati scambiate dai protocolli a livello di link sono chiamate **Frame**

I frame possono essere gestiti da collegamenti diversi (cablati, wireless, LAN) e quindi da **protocolli** diversi che erogano servizi diversi. Per identificare origine e destinazione vengono utilizzati indirizzi "MAC"

Servizi

Framing: incapsulamento dei pacchetti dello strato superiore in Frame

Rivelazione e correzione degli errori: Gli errori sono causati dal transito del segnale nel mezzo trasmisivo, il nodo ricevente individua la presenza di errori e li corregge (è possibile grazie all'inserimento, da parte del nodo trasmittente, di bit di controllo di errore all'interno del frame)

Controllo di flusso: Evita che il nodo trasmittente saturi quello ricevente

Consegna affidabile dei dati e ritrasmissione: Nel caso i requisiti dell'applicazione impongano una consegna affidabile dei dati il protocollo di link può effettuare la ritrasmissione delle frame affette da errore

È normalmente utilizzata nei collegamenti soggetti a elevati tassi di errori (es.: collegamenti wireless)
Nella modalità **full-duplex** gli estremi di un collegamento possono trasmettere contemporaneamente.
Nella modalità **half-duplex** la trasmissione nei due versi è alternata

Implementazione

Si tratta di un livello che è implementato sia nel software che nell'hardware.

A livello pratico si tratta di una scheda di rete che implementa le funzioni dello strato fisico e di collegamento e si chiama **Network Interface Card (NIC)** ed è una combinazione di hardware, software e firmware

Rivelazione Correzione errori

Il **nodo mittente** incapsula un pacchetto in un frame ed imposta il bit di rilevazione degli errori

Il **nodo ricevente** rivela l'errore e possibilmente lo corregge

Framing

Ha lo scopo di formare la PDU di strato (**frame**) incapsulando la PDU di strato superiore (**pacchetto**)

L'entità ricevente deve essere in grado di riconoscere senza ambiguità l'inizio e la fine di ogni frame (**funzione di delimitazione**)

Ad ogni frame viene aggiunto all'inizio e alla fine una sequenza fissa di bit, denominata **flag**

L'entità ricevente esamina il flusso binario entrante e delimita le frame riconoscendo i flag di apertura e di chiusura

Problema della **simulazione del flag** all'interno della frame

Esempio di funzione di delimitazione

Una possibile configurazione del Flag di delimitazione è: 01111110

Per evitare la simulazione si utilizzano le funzioni di:

Bit Stuffing: In emissione, si aggiunge uno “0” dopo ogni sequenza di cinque “1” consecutivi all’interno della frame indipendentemente da quale sia il bit successivo

Bit Destuffing: in ricezione ogni sequenza di cinque 1 consecutivi, se il bit successivo è un 1 è finito il frame, se invece è 0 è un bit di stuffing e lo si può togliere.

Byte stuffing e de-stuffing

Utilizzata nel protocollo PPP (Point to Point Protocol)

Byte stuffing (si usa una sequenza di «control escape» 01111101): se nella sequenza di bit, c’è un pezzo identico al flag, allora lo ripeto due volte.

Byte destuffing: se incontro due byte vicini uguali ne tolgo uno

Controllo d’errore

La trasmissione introduce errori Bit Error Rate (BER)

Il controllo d’errore si usa quando il livello trasmissivo non soddisfa i requisiti dell’applicazione

Il controllo d’errore assicura un determinato livello di accuratezza nel trasferimento di uno stream dati

Due approcci possibili:

- **Error detection & retransmission (ARQ)** – il ricevente rileva l’errore e chiede di ritrasmettere l’informazione.

- **Forward Error Correction(FEC)** – il ricevente rileva l’errore e lo corregge senza chiedere la ripetizione

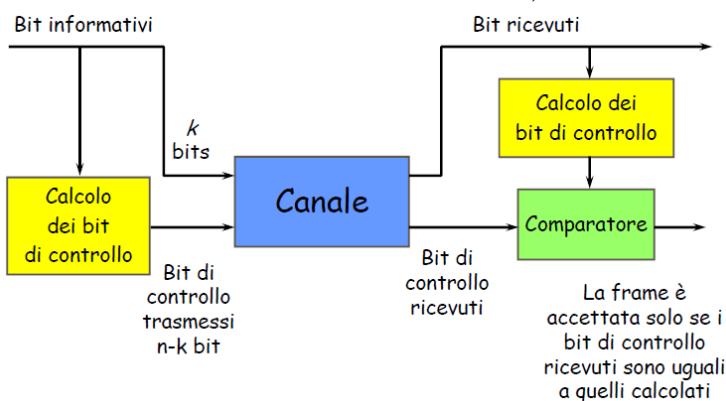
Principio base del controllo d’errore

È necessaria una **ridondanza** (**overhead**) costituita da un insieme di bit di controllo da aggiungere al blocco dati d’utente

È possibile che il canale trasformi la parola di codice trasmessa in una stringa di bit che è ugualmente una parola di codice

Rilevamento Errori

Codice di Parità: I dati trasmessi si organizzano in modo da mandare blocchi di codice(**codeword**), alla ricezione se il blocco non è una codeword, è considerato un errore.



Per fare questo è necessario aggiungere dei bit di controllo che vengono chiamati **Codici di controllo di Parità** che possono effettuare:

Controllo di Parità Singola:

Prendo la sequenza dei bit da trasmettere e conto quanti 1 ci sono, se ci sono un numero pari di 1 aggiungo uno 0 alla fine, invece se ci sono un numero dispari di 1 aggiungo un 1 alla fine

Il ricevente conta gli 1: se sono pari è corretto, se sono dispari vuol dire che qualcosa è andato storto.

Due problemi: una volta rivelato l'errore non posso correggerlo, se avviene un numero pari di errori non mi accorgo dell'errore.

Prestazioni del controllo di parità

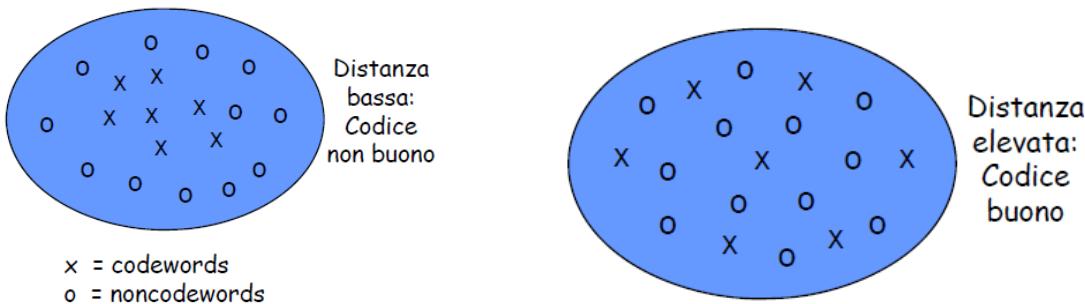
Vedere slide 17 pagina 11 (per me inutile al fine dell'esame)

Quanto è "buono" un codice?

In molti canali le configurazioni di errore più probabili sono quelle con un numero basso di bit errati Questi errori trasformano le codeword trasmesse in n-tuple "vicine", se le codeword sono

"vicine" tra loro allora la funzione di rivelazione può fallire

I buoni codici massimizzano la "distanza" tra le codeword trasmesse



Controllo di Parità Bidimensionale

Si struttura la sequenza dei bit informativi in colonne, per ogni colonna si aggiunge un bit di parità e si aggiunge poi una colonna di parità

1	0	0	1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0
1	1	0	1	1	0	0	0	0	0	0	0
<hr/>						1	0	0	1	1	1

La colonna finale è formata dai bit di parità di ogni riga

La riga finale è formata dai bit di controllo di ogni colonna

Problemi: se si verificano due errori sulla stessa colonna non posso rilevarli, quando gli errori sono maggiori di 4 non sempre è possibile rilevarli

Altri codici di rivelazione d'errore

I codici a parità singola hanno **scarse prestazioni**: Elevata probabilità di non rivelare errori

I codici bi-dimensionali hanno **overhead elevato**: Richiedono un numero elevato di bit di controllo

Internet Checksums: implementato nello strato di Trasporto, si aggiungono bit di controllo nell'header che controllino i bit di parità (controllo sul controllo), il checksum viene ricalcolato in ogni router e quindi deve essere di facile implementazione in software

Si considera la stringa di bit da proteggere composta da L parole di 16 bit, il checksum che aggiungo è anch'esso una stringa da 16 bit.

Con x pari alla somma degli interi

Prendo ogni parola della stringa e la converto in intero da binario e sommo tutti i vari interi e faccio:

$$- x |(2^{\# \text{bit}} - 1)|$$

Se lo si vuole fare in binario = sommo tutti i vari numeri in binario con complemento a 1(quindi se ho un bit di riporto lo sommo al risultato). Dopodiché ottengo il checksum come complemento a 1 del risultato della somma (ovvero invertendo tutti i bit)

Il ricevente somma tutte le sequenze e alla fine somma anche il checksum = se il risultato è composto da tutti 1 il pacchetto è valido

Non è comunque un metodo sicuro al 100% = se la somma ha qualche zero sono sicuro ci sia stato un errore, ma se la somma è tutti 1, non posso esser certa che non ci sia stato errore.

Più è robusta la codeword più facilmente potrò correggere gli errori

Esempio

Stringhe di 4 bit

Si usa l'aritmetica $mod_{-(2^4 - 1)} = mod_{-15}$

$$b_0 = 1100 = 12$$

$$b_1 = 1010 = 10$$

$$b_0 + b_1 = 12 + 10 = 7 \text{ mod } 15$$

$$b_2 = -7 = 8 \text{ mod } 15 = 1000$$

Codici polinomiali a ridondanza ciclica (CRC)

Le cifre della stringa da proteggere sono trattate come coefficienti (1 o 0) di un polinomio $P(x)$. Il polinomio avrà grado uguale alla lunghezza della stringa - 1 .

$$P(x) = a_{k-1}x^{k-1} + a_{k-2}x^{k-2} + \dots + a_1x^1 + a_0$$

In particolare, l'i-esimo bit (a_i) della stringa è il coefficiente del termine x^{i-1} di $P(x)$

Le entità emittente e ricevente utilizzano un polinomio comune $G(x)$, detto **polinomio generatore**

Coefficienti del grado massimo e grado nullo devono essere = 1

I coefficienti di $G(x)$ sono **binari**, come quelli di $P(x)$, supponiamo che questo polinomio sia di grado z

Emittente: prende il polinomio $P(x)$ lo moltiplica per x^z e lo divide per $G(x)$, il risultato è:

$$\frac{x^z P(x)}{G(x)} = Q(x) + \frac{R(x)}{G(x)}$$

Addizione e sottrazione sono operazioni identiche: equivalgono ad un XOR sui bit degli operandi

La **moltiplicazione** per x^z = significa aggiungere z zeri alla stringa originaria, il che equivale ad uno shift verso sinistra di k posizioni

Ricevente: prende il polinomio $T(x) = x^z P(x) + R(x)$ (parola di codice) e divide tutto per $G(x)$ ottenendo il $Q(x)$ se il quoziente è senza resto allora non ci sono errori, altrimenti ci sono.

Aritmetica polinomiale a coefficienti binari

Addizione e sottrazione: sono operazioni identiche ed equivalgono ad un XOR sui bit degli operandi
 $(x^7 + x^6 + 1) + (x^6 + x^5) = x^7 + x^6 + x^6 + x^5 + 1 = x^7 + x^5 + 1$

Moltiplicazione: la moltiplicazione di una stringa binaria per 2^k equivale ad uno shift verso sinistra di k posizioni

$$(x + 1) \cdot (x^2 + x + 1) = x^3 + x^2 + x^2 + x + x + 1 = x^3 + 1$$

Codici CRC

L'emittente calcola quindi il $T(x)$ e manda al ricevente il $P(x)$ di k bit e il $R(x)$ di z bit. Il **ricevente** riassembra il $T(x)$ e lo divide per $G(x)$ e controlla se c'è il resto o no.

Il ricevente ottiene una sequenza di bit di $T(x)$ e di $G(x)$. La divisione in binario viene fatta mettendo divisore sotto al dividendo a partire dalla prima cifra. E faccio operazione di xor. A ogni giro rimetto il divisore sotto, shiftato di uno a dx.

Rilevazione dell'errore

Rappresentiamo con un altro polinomio $E(x)$ il polinomio dell'errore. Il polinomio $T(x)$ errato sarà la somma di $T(x) + E(x)$, sempre divisibile per $G(X)$ e la storia è uguale: se il resto è nullo non ci sono errori, altrimenti ce ne sono.

Ogni bit “1” in $E(x)$ corrisponde ad un bit che è stato invertito e quindi a un errore **isolato**

Un errore a **burst** di lunghezza n è caratterizzato in $E(x)$ da un “1” iniziale, una mescolanza di “0” e “1”, e un “1” finale per un complesso di n coefficienti binari

$$\text{Resto} \left[\frac{T(x) + E(x)}{G(x)} \right] = \text{Resto} \left[\frac{E(x)}{G(x)} \right]$$

L'unico problema incorre quando $E(x) = G(x)$ perché ottengo zero anche se c'è l'errore.

Errori non rilevabili con CRC: tutti gli errori pari a $G(x)$ o multipli di $G(x)$

Codici CRC: polinomi generatori

Sono standard i seguenti polinomi generatori:

$$G(x) = x^{16} + x^{12} + x^5 + 1$$

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

Esempio calcolo CRC

Polinomio Generatore:

$$G(x) = x^3 + x + 1$$

Dati: (1,1,0,0)

$$P(x) = x^3 + x^2$$

$$x^3 P(x) = x^6 + x^5$$

$\begin{array}{r} x^3 + x^2 + x \\ \hline x^3 + x + 1 \quad \quad x^6 + x^5 \\ \quad \quad \quad x^6 + \quad \quad x^4 + x^3 \\ \hline \quad \quad \quad x^5 + x^4 + x^3 \\ \quad \quad \quad x^5 + \quad \quad x^3 + x^2 \\ \hline \quad \quad \quad x^4 + \quad \quad x^2 \\ \quad \quad \quad x^4 + \quad \quad x^2 + x \\ \hline \quad \quad \quad x \end{array}$	$\begin{array}{r} 1110 \\ \hline 1011 \quad \quad \boxed{11000000} \\ \quad \quad \quad 1011 \\ \hline \quad \quad \quad \boxed{1110} \\ \quad \quad \quad 1011 \\ \hline \quad \quad \quad \boxed{1010} \\ \quad \quad \quad 1011 \\ \hline \quad \quad \quad \boxed{010} \end{array}$
--	--

Codeword trasmessa: $b(x) = x^6 + x^5 + x$ (1,1,0,0,0,1,0)

Forward Error Correction (FEC)

Date due stringhe binarie di ugual lunghezza, X e Y e posto W(A) = numero di bit 1 della stringa A, si definisce **distanza di Hamming** tra X e Y la quantità:

$$HD(X, Y) = W(X \text{ xor } Y)$$

Un codice con parole di n bit può rappresentare simboli di m bit e la capacità di correzione è funzione della ridondanza $r = n - m$

Il valore minimo della HD tra tutte le coppie di parole di codice è la HD del codice(?)

Un codice con $HD = 2d+1$ può correggere fino a d errori binari e può rivelarne fino a $2d$

STRATO DI COLLEGAMENTO (PARTE 2) CAPITOLO 18

Protocolli di accesso multiplo (MAC)

Due tipi di Rete: **Punto-Punto(PPP)**, **BroadCast** (cavo o canale condiviso) nel quale comunicano centinaia o migliaia di nodi. Per evitare collisioni esistono protocolli di accesso multiplo che fissano le modalità di trasmissione su canali condivisi. Il protocollo viene implementato in tutti i nodi (decentralizzato) senza controllo centrale.

Protocolli a suddivisione del canale: suddivide canale in parti (slot di tempo, frequenza, codice) e per ogni nodo viene allocata una parte (es. telefonia, satelliti)

- **A divisione di tempo (TDMA)**: Si divide il canale x intervalli di tempo, ogni slot allocato a un nodo anche se il nodo non ha nulla da trasmettere.
- **A divisione di frequenza (FDMA)**: Si divide il canale x frequenze, ogni nodo ha una banda di frequenza, anche se il nodo non ha nulla da trasmettere.

Il Vantaggio è che se il canale è libero, il nodo trasmette al massimo R, se avviene collisione il protocollo deve saperla gestire (**aloha**, **ethernet**, **CSMA**).

Protocolli ad accesso dinamico:

- **Protocolli ad accesso casuale** (random access): Canali non divisi, se avviene collisione i nodi ritrasmettono il messaggio. La rete come se usasse un bus e tutti trasmettono quando vogliono.
- **Protocolli ad accesso controllato**: ogni nodo ha un turno, più o meno lungo a seconda della necessità. Il turno è di colui che possiede il token, quindi la rete è come se fosse un anello in cui ci si passa il token (wireless, LAN)

Protocolli ad accesso casuale

Quando un nodo deve inviare un pacchetto trasmette sempre alla massima velocità del canale, cioè R bit/s e non c'è nessun coordinamento a priori tra i nodi

Se due o più nodi trasmettono "contemporaneamente" si ha una "collisione"

Un protocollo ad accesso casuale definisce come rilevare un'eventuale collisione

Esempi di protocolli ad accesso casuale sono: **ALOHA**; **slotted ALOHA**; **CSMA**, **CSMA/CD**, **CSMA/CA**

Prodotto Banda Ritardo

$$PBR = Rd(\text{bit})$$

R (bit/s): banda del canale;

d (sec): ritardo di propagazione end-to-end

È il numero di bit che si trovano contemporaneamente sul canale. Ritardo prop = distanza km / velocità del collegamento

Calcolo dell'efficienza

Dato R : bit rate del canale (bit/s) e L lunghezza di una frame (bit)

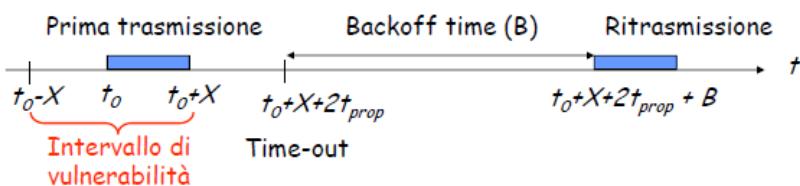
$$\text{Efficienza} = \rho_{\max} = \frac{L}{L + 2t_{prop}R} = \frac{1}{1 + 2t_{prop}R/L} = \frac{1}{1 + 2a}$$

$$\text{Throughput Massimo} = R_{\text{eff}} = \frac{L}{L/R + 2t_{prop}} = \frac{1}{1 + 2a} R \text{ bit/s}$$

$$a = \frac{t_{prop}}{L/R} = \frac{\text{Ritardo di Propagazione}}{\text{Tempo di trasmissione di una frame}}$$

Protocollo ALOHA

Ad accesso casuale, quindi il nodo trasmette quando ha una frame pronta, se c'è collisione la frame viene persa, il nodo aspetta l'ACK entro un tempo timeout, se scade, il nodo calcola il tempo di ritrasmissione (**backoff time**) e allo scadere di esso ritrasmette la frame.



Definizioni:

X = tempo di trasmissione frame (costante)

S = throughput (numero medio di trame trasmesse con successo nell'intervallo di X sec)

G = load (numero medio di tentativi di trasmissione in un intervallo X sec)

P_{succ} = probabilità di trasmissione della trama con successo

$$S = G * P_{\text{succ}} = G e^{-2G} (\text{Throughput})$$

Intervallo di vulnerabilità = $2X$

Slotted Aloha

I pacchetti hanno stessa dimensione ed il tempo diviso in slot uguali al tempo di trasmissione.

All'inizio degli slot sincronizzati i nodi trasmettono pacchetti, se collidono rilevano la collisione prima della fine dello slot e ritrasmettono frame negli slot dopo con probabilità p .

Ogni nodo può arrivare alla massima velocità di trasmissione e non c'è controllo centralizzato, ma una certa frazione degli slot presenterà collisioni e di conseguenza andrà "sprecata", mentre un'alta frazione degli slot rimane vuota, quindi inattiva.

$$\text{Intervallo di vulnerabilità} = X$$

L'efficienza dello Slotted Aloha

Definita come la frazione di slot in cui avviene una trasmissione utile in presenza di un elevato numero di nodi attivi, che hanno sempre un elevato numero di pacchetti da spedire.

Supponiamo N nodi con pacchetti da spedire, ognuno trasmette i pacchetti in uno slot con probabilità p
 La probabilità di successo di un dato nodo = $p(1-p)^{N-1}$

La probabilità che un nodo arbitrario abbia successo = $Np(1-p)^{N-1}$

Per ottenere la massima efficienza con N nodi attivi, bisogna trovare il valore p^* che massimizza

$$Np(1-p)^{N-1} \rightarrow p^* = \frac{1}{N}$$

Per un elevato numero di nodi, ricaviamo che nel caso migliore, solo il 36% degli slot sono utilizzati in modo utile

Pro

Consente a un singolo nodo di trasmettere continuamente pacchetti alla massima velocità del canale
È fortemente decentralizzato, ciascun nodo rileva le collisioni e decide indipendentemente quando ritrasmettere ed è estremamente semplice

Contro

Una certa frazione degli slot presenterà collisioni e di conseguenza andrà “sprecata”
Un’alta frazione degli slot rimane vuota, quindi inattiva

Protocollo CSMA (Carrier Sending Multiple Access)

I nodi prima di trasmettere ascoltano il canale, se è libero trasmettono, se è occupato aspettano prossimo intervallo di tempo.

Intervallo di vulnerabilità: $2t_{prop}$

Quando il nodo rileva il canale occupato si applicano degli algoritmi di persistenza:

1-persistent CSMA: appena il canale si libera il nodo inizia la trasmissione, basso ritardo e bassa efficienza

Non persistent CSMA: il nodo applica un **backoff** e ricomincia da capo con il CSMA, alto ritardo e alta efficienza

P-persistent CSMA: il nodo aspetta che il canale si liberi, con probabilità p trasmette, con probabilità $1-p$ attende un mini-slot e ricomincia il CSMA

CSMA/CD (with collision detection)

Il nodo ascolta prima di trasmettere e mentre trasmette, così se rivela una collisione invia un segnale di disturbo (**jam**) e interrompe la trasmissione e tutti i nodi coinvolti rischedulano dopo un tempo di backoff. Se si arriva all’ n -esima collisione consecutiva stabilisce un valore k (tra 0 e 2^{n-1}) e la scheda di rete aspetta un tempo pari a K volte 512 bit. Così non spreca i tempi di rilevazione di collisioni e la collisione viene rivelata al max ad un tempo $t = 2t_{prop}$

Confronto Protocolli

Per piccoli valori di prodotto banda ritardo, CSMA-CD ha il miglior throughput, per grandi valori di a , Aloha e Slotted Aloha hanno migliori prestazioni.

Protocolli ad accesso controllato

Polling: c’è un nodo master che gestisce i turni degli altri, elimina le collisioni e gli slot vuoti ma se si rompe il nodo master tutto il canale risulta inattivo.

Token-Passing: la rete è ad anello e ci si passa un messaggio di controllo (token) per chi ha il turno, quindi decentralizzato ma il guasto di un nodo può mettere inattivo tutto il canale.

I nodi in stato ready aspettano il token, quando lo ricevono e cominciano a trasmettere cambiano ultimo bit del flag per dire che è busy, alla fine della trasmissione rimette il bit per dire che è free.

Metodi di reinserimento del token

Ring Latency: numero di bit che possono esser trasmessi sul ring simultaneamente

MultiToken operation: appena viene trasmesso l’ultimo bit del frame il nodo trasmette il free token

Throughput = $\frac{1}{1+\frac{a}{M}}$ (M = nodi e a = tempo x bit di circolare sul ring/tempo trasmissione frame)

Single Token operation: Il Free token è inserito dopo che l'ultimo bit del busy token è ritornato al nodo origine. $\text{Throughput} = \frac{1}{\frac{a}{M} + \max(1, a)}$

Single Frame operation – Il Free token è inserito dopo che il nodo emittente ha ricevuto l'ultimo bit della sua frame. $\text{Throughput} = \frac{1}{(1+a)(1+\frac{1}{M})}$

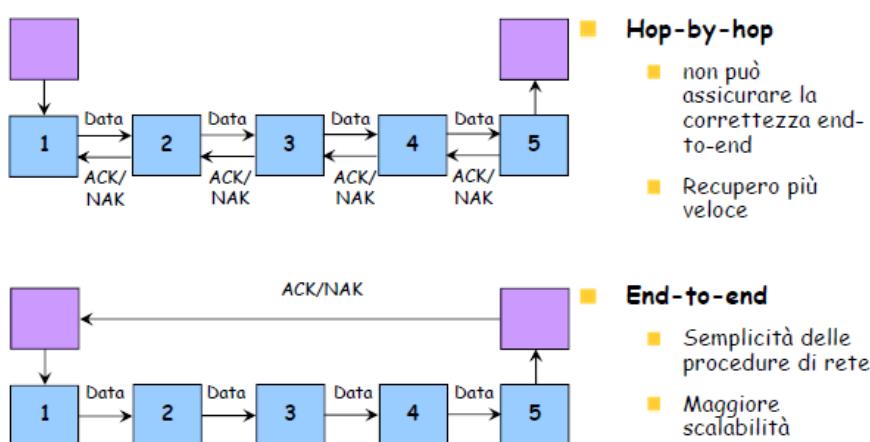
STRATO DI COLLEGAMENTO (PARTE 3) CAPITOLO 19

Controllo Errore – Controllo di Flusso

Hop-by-Hop = operazione effettuata tratta per tratta (Strato di Collegamento). Il controllo viene fatto a ogni hop e la reazione è immediata, nel caso di errore può essere ritrasmesso. Affidabile.

End-to-End = operazione effettuata da estremo a estremo (Strato di Trasposto).

Il controllo avviene tra sorgente e destinazione e i segmenti possono essere persi o subire ritardi. Il controllo è più complesso e meno affidabile.



Automatic Repeat Request (ARQ)

Servizio offerto dagli strati sottostanti per assicurare che una sequenza di PDU sia consegnata in ordine e senza errori o duplicazioni in presenza di un servizio offerto dagli strati sottostanti.

Stop-and-Wait

L'entità **emittente (Transmitter)** si trova inizialmente nello stato Ready in attesa che uno strato superiore gli richieda di inviare un pacchetto. All'arrivo del pacchetto trasmette un frame (composto da Pacchetto+Frame di controllo = Header+CRC) dove nell'Header c'è il numero di sequenza del frame S_{last} . Dopodiché passa allo stato Wait e attiva un temporizzatore nell'attesa di un ACK (riscontro positivo) da parte del ricevente.

In questo stato la ricezione delle richieste dallo strato superiore viene bloccata

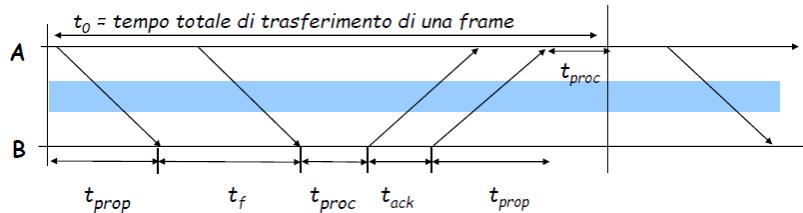
Se scade il timeout il frame viene ritrasmesso. Se ricevo l'ACK e il numero di sequenza $R_{next} = S_{last} + 1$ è corretto, torna nello stato ready, se invece il numero di sequenza è sbagliato l'ACK viene ignorato.

L'entità **ricevente (Receiver)** a sua volta sta sempre nello stato Ready in attesa di ricezione di una nuova frame. Quando arriva la frame effettua il controllo CRC.

Se non ci sono errori e il numero di sequenza $S_{last} = R_{next}$ è corretto, allora la frame viene accettata, aggiorna il valore di R_{next} , emette l'ACK con R_{next} (numero di sequenza che si aspetta di ricevere), consegna il pacchetto allo strato superiore.

Se non ci sono errori e il numero di sequenza è sbagliato: la frame viene scartata e viene mandato un ACK con R_{next} uguale a prima per richiedere l'ultimo frame
 Se ci sono errori: la frame viene scartata.

Tempo di trasferimento



$$\begin{aligned} t_0 &= 2t_{prop} + 2t_{proc} + t_f + t_{ack} && \text{Lunghezza di una frame} \\ &= 2t_{prop} + 2t_{proc} + \frac{n_f}{R} + \frac{n_a}{R} && \text{Lunghezza di un ACK} \\ &&& \text{Bit rate canale} \end{aligned}$$

Rate di Efficienza di trasferimento

$$R_{eff}^0 = \frac{\text{numero di bit informativi consegnati a destinazione}}{\text{tempo totale necessario per la consegna dei bit informativi}} = \frac{n_f - n_o}{t_0},$$

bit di overhead

Efficienza di trasmissione

$$n_0 = \frac{R_{eff}}{R} = \frac{\frac{n_f - n_o}{t_0}}{R} = \frac{\frac{n_f - n_o}{1 + \frac{n_a}{n_f} + \frac{2(t_{prop} + t_{proc})R}{n_f}}}{R}$$

Effetto dell'overhead di una frame

Effetto di un ACK

Effetto del prodotto Banda-Ritardo

Essa decresce all'aumentare della lunghezza del mezzo perché dipende molto dal prodotto banda-ritardo

Se poi vogliamo tener conto della probabilità di errori si moltiplica all'efficienza $(1 - p_f)$ che è la probabilità che un frame arrivi senza errori (che dipende dal mezzo trasmittivo).

Minore è p_f e maggiore sarà l'efficienza.

Problema: Anche quando i frame vengono mandati correttamente si spreca tempo nell'attesa degli ACK

Efficienza su un canale con errori

Sia $1 - P_f$ = probabilità che una frame arrivi senza errori

$\frac{1}{1-P_f}$ = numero medio di trasmissioni necessarie per avere una trasmissione corretta di una frame

$\frac{T_0}{1-P_f}$ = tempo medio di trasferimento di una frame

Go-back N

Il **Transmitter** rimane attivo perché manda una finestra di frame (il numero di frame mandabili è limitato dal W_s) e numera i frame con m bit.

Se riceve ACK delle frame emesse prima che termini quella finestra, allora la finestra viene aggiornata con quella successiva e continua la trasmissione

Se la finestra termina e non riceve ancora ACK : si mette in attesa degli ACK

Se non riceve ACK allo scadere di timeout: ritrasmette la finestra richiesta in R_{next}

Il **Receiver** se tutto va bene rimanda con l'ACK il numero di frame che si aspetta di ricevere in R_{next} al termine della finestra. Altrimenti se c'è un problema rimanda con ACK il numero del frame ricevuto male e continua a scartare gli altri fintanto che il Transmitter non ricomincia dal frame errato.

Efficienza: Nei casi di elevato Prodotto Banda Ritardo è meglio di S&W ma nei casi di elevato BER l'efficienza diminuisce

Problema: anche quando certe frame vengono consegnate correttamente, rimanda dal punto in cui l'ACK era negativo, ritrasmettendo quelle corrette

Sliding window

Il Transmitter attende gli ACK (con numero di sequenza $S \geq S_{last}$) e quando arriva un ACK, con numero di sequenza S , viene posto $S_{last} = S$

L'estremo superiore della finestra sarà quindi $S_{last} + W_s - 1$

Il massimo valore della finestra = $W_s = M-1 = 2^m-1$

PiggyBacking: quando due entità mandano e ricevono e quindi nel mandare il messaggio caricano sulla schiena anche gli ACK di ciò che hanno ricevuto.

Tempo Timeout(T_{out} componenti): Il timeout è il tempo che ci vuole per arrivare da punto a punto ed è uguale alla somma di: $2t_{prop} + t_{proc} + t$ trasmissione frame + t trasmissione ack.

W_s deve quindi essere abbastanza grande per mantenere il canale occupato per tutto il T_{out} .

Selective Repeat

Go-Back-N è inefficiente poiché, in caso di ritrasmissione, viene riemesso un numero elevato di frame, anche se ricevute correttamente dal receiver

Selective Repeat ritrasmette solo le frame che sono state perse

Transmitter: All'esaurimento del Timeout il Transmitter rimanda solo la frame corrispondente al NAK più vecchio.

Receiver: gestisce una finestra in ricezione con sequenza di numeri che possono essere accettati.

Il NAK inizia a mandare ACK sempre con R_{next} relativo al NAK.

Ma in realtà dentro di se aggiorna un altro contatore così quando viene rimandato quello, il NAK riprende dal punto lasciato. Quelle che arrivano fuori sequenza vengono bufferizzate.

Efficienza: meglio di GBN e S&W ma comunque l'efficienza diminuisce all'aumentare del BER.

Il massimo valore permesso è: $W_s + W_r = 2^m$