Reti di calcolatori 2010/2011

antiufo Ricky SyKoM

ITU IETF RFC MUX DEMUX FDM TDM ISP MODEM WAN LAN MAN WAN circuit switching PSTN ISDN WAN packet switching FRAME RELAY ATM IP xDSL ADSL OSI FISICO DATA LINK RETE TRASPORTO SESSIONE APPLICAZIONE SMTP MIME-Version DNS MUA MTA MDA POP IMAP HELO ENVELOPE MAIL FROM MAIL RCPT TO DATA HEADER BODY CC BCC MX NVTASCII SMTPe POP3 HTTP GET HEAD POST Cosici di stato URI URL URN HTML FTP TFTP Telnet ACK ARQ TCP UDP RTT HLEN DWORD URG ACL PSH RST SYN FIN MSS SYN_SENT ENSTABILISHED FIN_WAIT_2 FIN_WAIT_1 TIME_WAIT CLOSED SRTT RTTVAR Go back N Selective Repeat Congestion Window CWIND RECWND SSTHRESH Forwarding Routing JITTER CRB ABR VC VCI FCFS WFQ Drop tail Random early detection IPv4 IPv6 Subnet Mask DHCP NAT UPnP ICMP DV LS Hot potato RIP OSPF BGP CIDR AS-PATH NEXT-HOP FLOODING UNICAST MULTICAST ANYCAST HALF-DUPLEX FULL-DUPLEX FRAMING BIT STUFFING CHARACTER STUFFING Pysical layer coding violations MAC NIC PARITY CHECK CHACKSUM CRC TDMA FDMA CDMA PURE ALOHA SLOTTED ALOHA CSMA CSMA/CD CSMA p-persistent POLLING PROTOCOL TOKEN PASSING PROTOCOL ETHERNET IEEE 802.3 PPP SDH ARP 10 BASE-T 10 BASE-2 100 BASE-T 1000 BASE-LX 10G BASE-T SWITCH ROUTER HUB BRIDGE CAPACITà IMPEDENZA INDUTTANZA RESISTENZA RJ45CADDING CORE DIAFONIA FADING SHANNON BACKBONE WI-MAX GPRS UMTS nBmB MODULAZIONE BANDA BASE BANDA TRASLATA P2P SOCKET

Introduzione

sabato 14 maggio 2011
22:17

- ITU, IETF
  - International Telecommunication Union: de jure - reccomendation
  - Internet Engineering Task Force: de facto RFC

Definizioni

- **Trasmissione**: trasferimento segnali da 1 a 1 o più punti
- **Commutazione**: processo di interconnessione di unità funzionali, canali di trasmissione o circuiti di telecomunicazione per il tempo necessario al trasferimento
- **Segnalazione** (informazioni di controllo): apertura, chiusura
- **Banda** (teoria segnali): ampiezza spettrale
- **Banda** (telecomunicazione): bps
- **Capacità di un canale**: massima velocità trasmissiva (bps)
- **Traffico offerto**: dati che la sorgente cerca di inviare
- Traffico smaltito (**throughput**): porzione del traffico offerto consegnata correttamente a destinazione.
  $$throughput \leq capacità\ del\ canale$$
  $$throughput \leq traffico\ offerto$$
- **Nodo, canale**
- Tipi di canale
  - Punto-punto
  - Multi-punto
  - Unidirezionali/Bidirezionali

Topologia delle reti

- Maglia: vantaggio tolleranza guasti, svantaggio elevato numero di canali (usata solo a livello astratto, tipo p2p)
- Albero: vulnerabile ai guasti, vantaggio basso numero canali, costa meno, un solo percorso fra nodi
- Stella: vulnerabile al centro (reti locali, satellite, radio cellulari), vantaggio pochi canali, riduce costi, 1 percorso
- Maglia (mesh): instradamento complesso, topologia non regolare, tolleranza guasti, flessibile, la più usata
- Anello (metropolitane, bidir o unidir)
- Bus (locali e metropolitane)

Protocolli

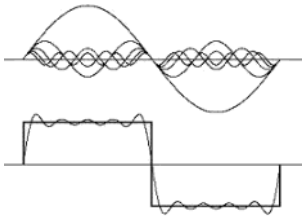
- Sintassi (formati)
- Semantica (funzionamento, cosa fare nelle varie situazioni)
- Temporizzazione (timeout, eventi...)

Covelli - Introduzione

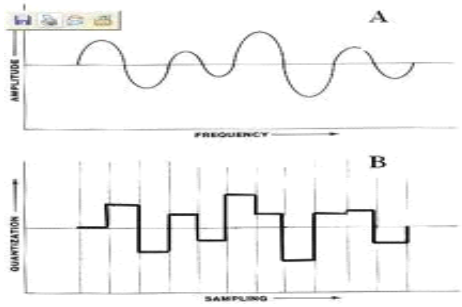
domenica 15 maggio 2011
00:10

Fourier

Ogni segnale periodico può essere scomposto come somma di sin e cos di frequenze ed ampiezze diverse. Dalla trasformata di Fourier deriva lo spettro del segnale.



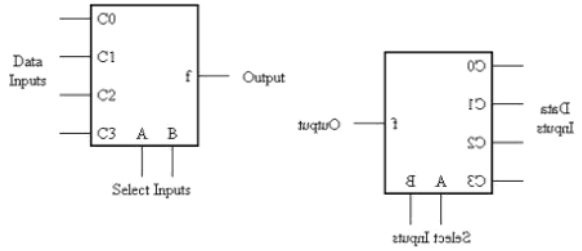
Analogico vs Digitale



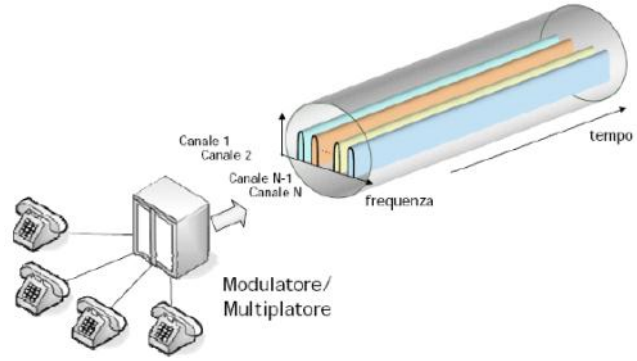
- **Digitale**: associa ad un range di frequenza un unico valore. Più valori ci sono, più è precisa la conversione, attenzione: potresti convertire anche il rumore, se troppo sensibile. Più facile da ricostruire (ovviamente).
- **Analogico**: segnale naturale, infiniti valori. Subisce molto il rumore, impossibile poi ricostruirlo. Per evitare si riduce la distanza massima fra i ripetitori.

PCM: Conversione analogica in digitale con compressione (complicato, ci vorrebbe una vita).

Multiplexing-demultiplexing

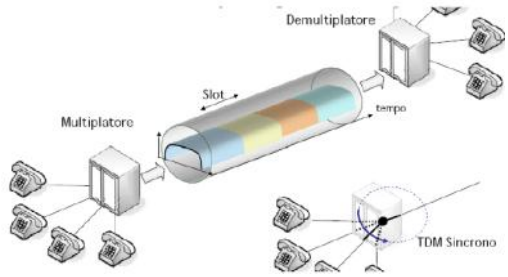


FDM



Frequency division multiplexing: modula modula modula modula. Meno diffusa della TDM.

TDM



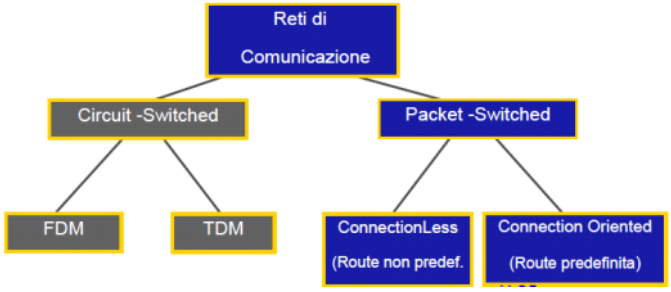
Time division multiplexing: un po' tu, un po' tu, un po' tu. Alla fine parlo a scatti che non si sentono. È modulare, quando finisco con l'ultimo inizio dal primo. Ce ne sono due tipi molto diffusi:

- **T1**: 24 canali logici per trunk
- **E1**: 32 canali logici per trunk, 2 riservati a sincronizzazione e segnalazione

Seguono una gerarchia: N vengono raggruppati, quel gruppo viene raggruppato con N altri gruppi ecc. Man mano che ci alziamo di gruppo andiamo sempre più veloci.
Se abbiamo da mettere assieme i dati provenienti da una T1 con una E1, è un casino, ma c'è lo standard PDH che lo fa. Ultimamente però si sta usando SDH (su fibre specialmente) perché fa tutto meglio, senza complicazioni, è usato come standard Europeo di livello 1, sia per voci che per dati.

Connessione ad ISP con modem analogico

WAN



WAN circuit switching

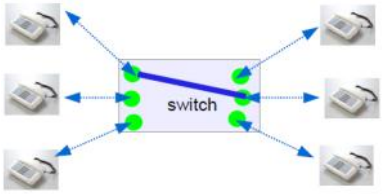
Si viene a creare un circuito temporaneo dedicato fra chiamante e chiamato. Il numero di connessioni fra due switch (le chiamate effettuabili contemporanemente) dipende dal numero di linee di interconnessione fisicamente disponibili (dette trunk). Usa FDM e TDM: su un unico collegamento fisico più canali logici contemporanei. Gli switch sono organizzati in maniera gerarchica.
Presentano importanti limiti quando utlizizzate per trasmissione dati (in quanto progettate per la voce):

- La voce presenta esigenze di trasporto sincrone, di solito non necessaria per i dati, costosa
- Banda disponibile limitata e usata discontinuamente (IDLE time)
- TDM ha velocità di trasmissione bassa
- L'affitto di una linea di T1 dedicata, balle varie, costa molto

Varie tipologie:

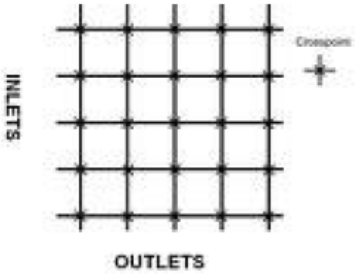
- **Mesh:** tanti link di tipo point to point. Costosa, poco utilizzata, complessa da gestire

- **Star:** tanti link di tipo point to point connessi ad un unico switch (es: Bell Telephony Company)



Distanze limitate tra telefono e switch, ottimizzazione del numero connessioni rispetto a mesh, molta attenuazione, merda varia. All'inizio lo switch era un beota che collegava a mano. Poi nel 1960 li hanno sterminati e sono passati agli switch elettronici

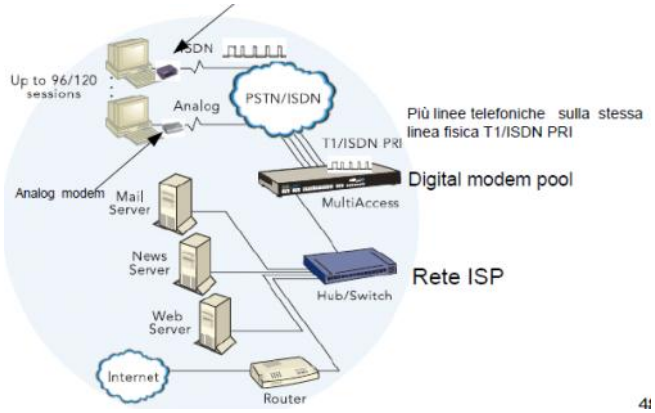
- **Switched network**: usa un crossbar switching (dispositivo elettromeccanico in grado di creare il collegamento fra linea entrante, uscente)



Per gestire lunghe distanze si ricorre a più switch connessi in point to point. Questa switched network crea connessioni temporanee fra i dispositivi.

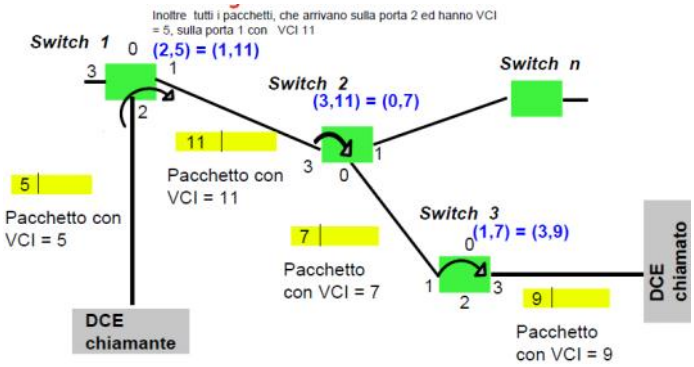
WAN di tipo circuit switching:

- **PSTN**: public switched telephone network
- **ISDN**: integrated services digital network. Usa una TDM, convertendo prima in digitale. Ultimamente si usano altre tecniche per poter sostituire le infrastrutture telefoniche con le stesse usate per la trasmissione dati (UDP, SIP, RTP). Se serve superspeed possiamo noleggiare un flusso primario solo per noi. Molto usato per connettere centralini aziendali alla centrale telefonica



WAN packet switching

- **Connection oriented**: le VC di livello 3, fa sempre la stessa strada una volta deciso il percorso (Frame Rela, ATM ecc)



Alla fine della trasmissione, il VCI viene rilasciato. Hanno overhead limitato, riducendosi al virtual channel e poche altre informazioni di controllo

- **Connection less**: pacchetti possono percorrere strade diverse (IP). Maggior overhead, ma c'è l'instradamento alternativo.

Risolvono i problemi delle circuit switching, sono più adatte alla trasmissione dei dati ed inviano pacchetti in modalità asincrona. Si basano sulla suddivisione dei messaggi applicativi in pacchetti di dimensioni standard. Ogni pacchetto contiene:

- Parte del messaggio da trasmettere
- Header

Utilizzano degli stack di protocolli. Bisogna passare il carico da un layer all'altro, fino alla trasformazione in segnale analogico o digitale. **Vantaggi**:

- Pacche to ha dimensioni limitate e usa per poco tempo il canale fisico
- Consente accesso multiplo al canale
- Banda è utilizzata interamente solo quando ci sono dati da trasmettere
- Migliore utilizzo del canale, meno costi: tutta la banda va ad un pacchetto, se il canale è occupato lo mettiamo in buffer (Store n forward) tramite TDM

Svantaggi rispetto al circuit switching:

- Ritardo [variabile(jitter)] introdotto dagli switch
- Se il ritardo ha poca importanza per i dati veri e propri, può diventare un problema nelle reti multimediali
- Gli header non sono nulli, devono essere trasmessi

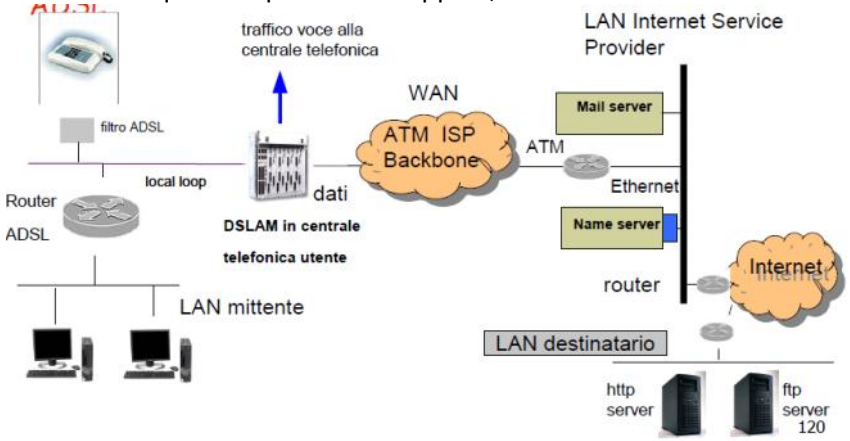
WAN di tipo packet switching:

- **X.25**, (connection oriented) la prima inventata, offre 64kbps di banda
- **ATM**, nasce con lo scopo di creare un'unica rete standard a livello mondiale per voce, video, dati. Ci sono diversi qualità di servizio in base alle esigenze. Frame di lunghezza fissa, per non creare eccessivi ritardi nella voce
- **FameRelay**, (connection oriented) migliora le prestazioni e la semplicità di X.25, costa di meno, sfrutta meglio le migliorie delle linee.

Non è detto che il tipo di rete rimanga sempre lo stesso nel percorso dei pacchetti.

ADSL (xDSL)

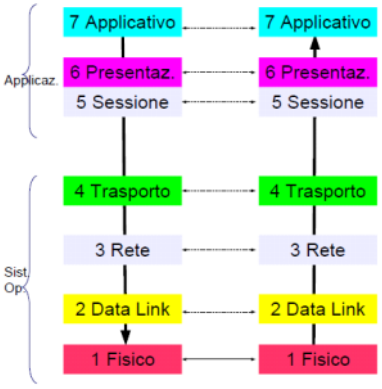
Tecnologia digitale che sta sostiutento ISDN. Va su doppini telefonici frequenze superiori a 1.1 MHz. Modula, modula, modula: voce in alto, dati in basso. Per essere sicuri, ci si mette anche un filtro, per eliminare i dati dal telefono. La qualità dipende dal doppino, distanza max: telefono-centrale



Modello ISO/OSI

- **ISO**: international standard organization
- **OSI**: organization standard international , scherzo: Open system Interconnection

È un modello concettuale delle architetture di rete, nessuno poi lo ha veramente implementato (che pensata). È solo un riferimento teorico, in realtà c'è il TCP/IP.



Quelli sopra usano servizi da quelli sotto. Si scorre in giù per inviare, in su per ricevere; procedendo per incapsulazioni-deincapsulazioni successive. For DUMMIES: non posso comunicare direttamente col livello applicativo del client, devo prima scorrere tutti gli N protocolli. Si sa cosa fa ogni livello (banale, easy oppure facile, dirsivoglia). I **layer Casati** (desaparesidos) 5 e 6, vengono fusi assieme al 7 nel TCP/IP.

Storia

- Primi sistemi di telecomunicazione ottica furono sviluppati in Francia a fine 1700.
- Primi esperimenti con l'elettricità:
  - Galvani fine 1700: rana
  - Volta 1800: una specie di pila
  - Morse (1850): telegrafo, prime cose in digitale

Invenzione del telefono

Meucci, 1871. Bell, 1876. Da questa invenzione nascono le reti circuit switching, che verranno utilizzate successivamente anche come rete dati. I primi telefoni usano solo l'analogico, dal 1960 ci sono i digitali.

Layer 5 - Livello applicativo

00:10

Obiettivo: interfacciare utente e macchina. Fornisce un insieme di protocolli che operano a stretto contatto con le applicazioni. È errato identificare un'applicazione utente come parte del livello applicazione.
I protocolli delle applicazioni tipiche di questo livello realizzano operazioni come:

- Posta elettronica
- Trasferimento di file
- Terminale virtuale

SMTP

Simple Mail Transfer Protocol. Supporta MIME-Version: 1.0.Progettato inizialmente per essere semplice, ma ha difetti:

- Vulnerabile a spamming
- Informazioni inviate in chiaro
- Mancanza di meccanismo di autenticazione

Utilizzato per l'invio di posta elettronica con indirizzi del tipo user@example.com (case insensitive). Il nome del mailserver viene recuperato dai record di tipo MX del DNS (ovvero associare al nome del dominio, il nome del/dei server di posta)

- DNS:
    - mail1.example.com. IN A 192.168.100.50
    - mail2.example.com. IN A 192.168.100.54
    - example.com. IN MX 10 mail1.example.com.
    - example.com. IN MX 20 mail2.example.com.
    - Il numero (10, 20) indica la priorità del mailserver

- **Mail User Agent (MUA)** è il client di posta (Outlook, Eudora...) che invia la posta al MTA (il mail server) del dominio tramite SMTP, IMAP o protocolli proprietari.
- **Mail Transfer Agent (MTA)** è il server (Microsoft Exchange, IBM Lotus Domino, Sendmail...)che inoltra la posta ricevuta dal MUA. L'MTA del dominio mittente è server nei confronti dell'user agent e client nei confronti del mail server destinatario. Lo scambio d'informazioni fra MTA avviene tramite protocollo SMTP.
- **MDA (Mail Delivery Agent)** è la componente software che riceve da MTA il messaggio di posta e lo memorizza nello spazio (message store) dove sono configurate le caselle degli utenti (possono essere file o database relazionale).Dispone normalmente di funzioni di controllo del contenuto (content filtering, antivirus, etc..). Alcuni prodotti includono entrambe le componenti MDA e MTA.
- **POP/IMAP SERVER** sono due protocolliche offrono funzioni di MDA:
    - Componenete software che interagisce con i client di posta degli utenti
    - Accede al message store e trasmette ai client le email contenute nelle caselle di posta
    - Protocolli alternativi: POP3 e IMAP

Comandi SMTP

- **HELO** :comando che segnala l'inizio dell'Envelope
- **Envelope**: serie di comandi, in protrocollo SMTP, scambiati tra MUA e MTA e fra I vari MTA per la corretta consegna del messaggio
- **MAIL FROM, RCPT TO**: identificano essenzialmente gli indirizzi email del mittente (envelope sender) e dei destinatari (envelope recipient). Tali indirizzi possono coincidere o meno con quelli presenti nell'header. Ad esempio I vari BCC sono presenti in envelope ma non nell'header.
- **DATA**: riporta il testo dell'email da inviare (header + body)
- **HEADER**: metadati del messaggio, ad esempio
    - Content-Transfer-Encoding: base64
    - Content-Type: audio/wav
    - Content-Id: un identificativo qualunque
    - Content-Disposition: inline; filename="ding.wav"
- **BODY**: messaggio vero e proprio ed allegati (attachments)

Invio di un messaggio

Un messaggio corrisponde ad N invii, l'utente specifica al client il mittente (FROM) ed il destinatario del messaggio (o vari destinatari):

- TO , destinatari diretti
- CC (Carbon Copy), per conoscenza: se lo invii a CC, significa che lo ricevono anche loro, ma non sono i destinatari principali
- BCC (Blind Carbon Copy): lo ricevono tutti, ma ogni singolo destinatario non vede a chi altri è stato mandato

- L'indirizzo del mittente (FROM) origina un comando SMTP del tipo MAIL FROM
- Gli indirizzi dei destinatari (TO, CC, BCC) originano un comando SMTP di tipo RCPT TO (uno per ogni destinatario)
- Con il comando DATA, il client invia header e body del messaggio (nell'header son riportati I destinatari del messaggio diretti (TO) e in CC).

Esempio:

<Connessione SMTP su TCP porta 25, nome mailserver presente su MX del DNS>
HELO client.example.com
MAIL FROM:<sender@example.com>
RECPT TO:<recipient@example.org>
DATA
From: Sender <sender@example.com>
To: Recipient <recipient@example.org>
Date: Sun, 11 Apr 2009 22:36:51 +0200
Subject: Mail and email

Dear recipient,
Regards.
<CRLF>.<CRLF>

Altri header: Cc, Bcc, Message-Id, Keywords, Subject, X-* (estensioni), Reply-To, In-Reply-To
Ogni MTA di passaggio aggiunge un header Received

```
Received: from amphissa.erlm.siemens.de ([146.254.164.8])
          by cz.siemens.net
          with Microsoft SMTPSVC;
          Thu, 29 Dec 2005 08:14:58 +0100
Received: from tegea.erlm.siemens.de
          by amphissa.erlm.siemens.de
          with ESMTP
          id 17862158526
          for <libor.dostalek@siemens.com>;
          Thu, 29 Dec 2005 08:14:58 +0100 (CET)
Received: from zetes.siemens.com (zetes.siemens.com [217.194.34.75])
          by tegea.erlm.siemens.de
          with ESMTP
          id CA10D1774B8
          for <libor.dostalek@siemens.com>;
          Thu, 29 Dec 2005 08:14:56 +0100 (CET)
Received: from imap.packtpub.com (unknown [217.207.125.60])
          by zetes.siemens.com (Postfix) with ESMTP
          for <libor.dostalek@siemens.com>;
          Thu, 29 Dec 2005 08:14:55 +0100 (CET)
Received: from paramita (unknown [203.122.53.88])
          by imap.packtpub.com (Postfix)
          with ESMTP
          id B6D24970B36
          for <libor.dostalek@siemens.com>;
          Thu, 29 Dec 2005 07:22:12 +0000 (GMT)
From: "Abhishek" <abhisheks@packtpub.com>
To: "'Dostalek Libor'" <libor.dostalek@siemens.com>
Subject: RE: TCP/IP DNS_Chapter 14
Date: Thu, 29 Dec 2005 12:44:44 +0530
Message-ID: <000001c60c47$8ce00220$0d00a8c0@paramita>
X-Mailer: Microsoft Office Outlook 11
Return-Path: abhisheks@packtpub.com

Message text
```

Set code e gestione d'errori

- La comunicazione avviene con setcode NVT ASCII (7 bits per carattere)
- Se l'MTA non riesce a passare l'email al MTA di destinazione, si ritenta la trasmissione ogni 1000 secondi per ad esempio 5 giorni

SMTP passi fondamentali

- Il client MUA risolve, via DNS, l'indirizzo ip del mailserver del dominio del mittente
- Il client si connette alla porta TCP 25 del mailserver che risponde con un messaggio 220<ready>
- Il client segnala l'inizio dell'envelope con un comando HELO, seguito opzionalmente dal proprio nome completo di dominio (FQDN). Il server risponde con 250<OK>
- Il client specifica il mittente (envelope sender) con MAIL FROM <indirizzo>, e il server risponde con 250<OK>
- Il client precisa gli indirizzi del destinatario (envelope recipients) con RCPT TO <indirizzo>, la risposta è ancora 250<OK>
- Il client dichiara di esser prontro a trasmettere il vero messaggio con: DATA. Il server risponde con : "354 End data with <CR><LF>. <CR><LF>".
- Il client inserisce l'header ed il body del messaggio e termina con una riga contenente solo il carattere '.'.
- Il client chiude la sessione con il comando QUIT.
- Il processo continua in modo analogo fra MTA rispettivamente dei domini mittente e destinatario/i.
- Una significativa differenza tra MUA e MTA è data dal fatto che, a fronte dell'unica connessione da MUA contenente n richieste RCPT TO ricevute dal client, MTA innesca n connessioni con I vari mailserver dei domini destinatari. (relay)
- Inoltre, MTA, nel processo di relay, lascia inalterati header e body del messaggio, ad eccezione del campo Received che indica I vari mailserver attraverso I quali il messagio è transitato.

ESMTP

- EHLO (per distinguersi da SMTP)
- 8BITMIME (invio di dati binari senza usare base64)
- SIZE (specifica dimensione email)

POP3

- Porta predefinita: 110
- Consente ai MUA di recuperare la posta presente nelle loro caselle e consente il download delle relative email
- USER user
- PASS pass
- STAT (ottiene numero email, spazio occupato...)
- RETR messageId (scarica un messaggio)
- DELE messageId (elimina un messaggio)
- TOP numEmail numLines (mostra le prime .. linee dei primi .. messaggi)

IMAP

- Protocollo che consente di mantenere sincronizzate le proprie cartelle sia sulla casella sia sul proprio computer

HTTP

Hyper Text Transport Protocol. Utilizzato per il web (protocollo di livello 7, applicativo). Nasce per interscambio di risorse fra utente e server (solitamente documenti html statici/dinamici, ma posson essere pure file .pdf, mp3, etc..)
Nel caso in cui il protocollo dell' URL corrisponde ad HTTP, il browser traduce la richiesta in una sequenza di comandi, che vengono inviati al server attraverso una socket basata su protocollo TCP.

- Primo server: httpd, '89
- Primi client: Viola, Cello, Mosaic

Prima di HTTP, occorrevano N terminali connessi a N differenti computer ed occorreva imparare N differenti programmi per accedere ai dati.

- Ci sono gli hyperlink
- Non c'è un controllo centrale
- Porta predefinita: 80
- Metodi:
    - **GET**: richiede una risorsa. È quello che succede quando scrivi qualcosa nella barra degli indirizzi, scarichi un file, o viene visualizzata un immagine dentro la pagina
    - **HEAD**: richiede solo i metadati di una risorsa (gli header, senza il corpo)
    - **POST**: invia dei dati e si aspetta una risposta. È quello che succede quando hai una form e premi il tasto Invia

- I dati sono trasmessi nel body
- Di norma vi sono due header (contentType, contentLength) ad indicare il tipo e la lunghezza dei dati

Richiesta

GET /path/to/file.txt HTTP/1.1<CRLF>
User-Agent: Mozilla/4.0 (compatible; Mozilla Firefox 4.0; DamaLinux 11.06)<CRLF>
Connection: keep-alive<CRLF> (consente di riciclare la connessione TCP per richieste future)
Header-3: Value3<CRLF>
<CRLF>

Risposta

HTTP/1.1 200 OK<CRLF>
Response-Header-1: Value1<CRLF>
Content-Length: 4<CRLF>
Content-Type: text/plain<CRLF>
<CRLF>
body

Richiesta con POST

POST /path/submit.php HTTP/1.0
Content-Type: application/x-www-form-urlencoded (significa che i dati della form usano la stessa codifica degli URL)
Content-Length: 23
Il corpo del post potrebbe essere: name=value&name2=value2
Rispetto al GET, non vediamo nell'URL i dati, possiamo inviare qualsiasi cosa, file ecc. Non abbiamo limiti di lunghezza imposti dall'URL

Codici di stato

200 OK, 404 Not Found, 500 Internal Server Error, 301 Moved Permanently, 403 Forbidden...

Riassumendo:

- HTTP è un protocollo stateless: ogni connessione è indipendente dalle altre
- Qualora sia necessario che il server sia in grado di associare una richiesta ad altre, inviate in precedenza, si ricorre alla tecnica dei cookie
- Il cookie non è altro che un numero di riconoscimento, inviato dal server al client all'atto della prima connessione, memorizzato ed inviato successivamente dal client per farsi riconoscere

URI

- Uniform Resource Identifier: identifica una risorsa
  - Uniform Resource Locator:
    - Oltre che a identificarla, dice anche dove la risorsa si trova
    - mss://streaming.example.com/stream.asx
  - Uniform Resource Name:
    - Dice solo il nome, o identificatore della risorsa
    - isbn:4554-66734-67
- Uniform Resource Locator (URL, particolare tipo di URI)
- È un indirizzo che permette di identificare in modo univoco una risorsa, per risorsa s'intende generamente un file (es: pagina html statica)
- Formato: schema:schema-specific-part
  http: //www.example.com/, mailto: user@example.com,
  gopher: //gopher.example.com/, ftp: //ftp.example.com

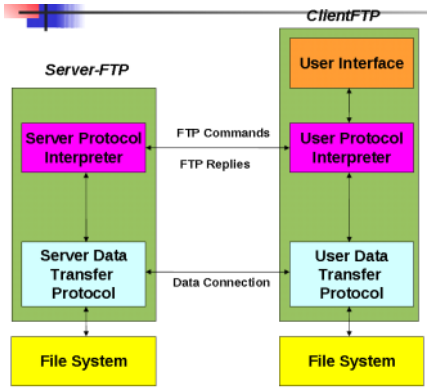
HTML

- Hyper Text Markup Language
- Linguaggio con il quale vien rappresentato il documento inviato dal web server al client
- Consente di indicare al browser il contenuto di un documento in forma di:
  - testo + riferimenti ad immagini + links

    - modalità di visualizzazione
- All'interno sono presenti dei tag che dicono al browser come rappresentare il contenuto.
- I tag sono comandi di rappresentazione ad alto livello.

FTP

- File Transfer Protocol



- Protocollo per accedere ad una specifica directory di un server ed eseguirvi operazioni di download e upload di files
- Deve tenere in considerazione l'esigenza di fornire dei comandi indipendenti dal tipo di sistema operativo sottostante
- "Protocollo storico" di internet - 1971
- Sostituito per limiti di sicurezza (le password viaggiavano in chiaro)
- Comandi forniti:
    - Collegarsi ad un server ed effettuare login
    - Navigare nelle directories del server (cd, ls, size, cdup, dir, etc...)
    - Navigare nelle directories del client (lcd)
    - Settare le modalità di trasferimento (ascii, binary)
    - Trasferire files (get, put, mget, mput, etc...)
    - Chiudere la connessione (close, disconnect)
- Prevede due diverse connessioni (socket)
    - Trasferimento comandi FTP
    - Trasferimento dati

Trivial File Transfer Protocol

- Usato per aggiornare il firmware di apparati di rete (telefoni SIP, switch, router) e in tutti I quei casi dove non si necessita di un'iterazione tra utente e server

Telnet

- Altro "Protocollo storico" 1969, sostituito poi dal piu sicuro ssh
- Consente l'accesso ad un server da remoto, fornendo una shell per eseguire comandi in formato testo come se l'utente fosse connesso localmente
- SCOPO:
    - da parte del client: inviare al server caretteri digitati a tastiera come se fossero stati direttamente digitati su una shell dal server
    - da parte del server: trasmettere al client l'output dei vari comandi
- Come FTP deve tener conto dell'eterogeneità dei dati
- Per risolvere questo problema, Telnet definisce un terminale virtuale standard (Network Virtual Machine) in modo da disaccoppiare il protocollo dai vari sistemi operativi
- Come funziona:
    - I caratteri digitati dal client venogno intercettati dall'applicazione telnet e trasformati nel formato previsto da NVT.
    - Vengono trasmessi al server, dove l'applicazione telnet trasforma I caratteri NVT in quelli specifici del server.
    - Analogo il passaggio per l'output

Layer 4 - Transport layer

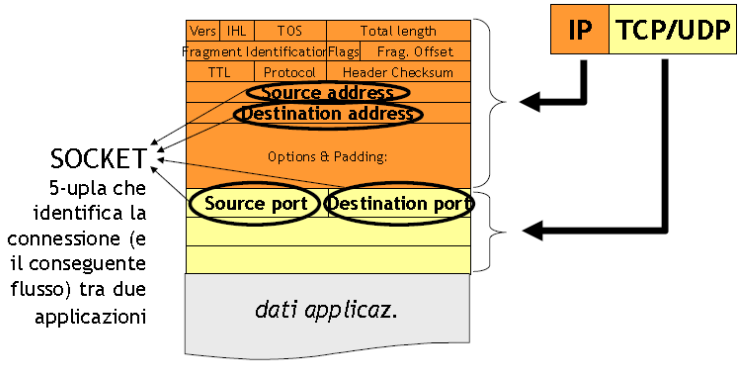
sabato 14 maggio 2011
22:31

Generalità

Non ha visibilità dei dettagli della rete. Fornisce un canale di trasporto end-to-end ideale e privo di errori tra due utenti, indipendentemente dalla rete. Per compiere questo obiettivo, come tutti i livelli OSI, il livello di trasporto offre, attraverso delle primitive, dei servizi al livello superiore e svolge una serie di funzioni:

- **Rilevamento di errori**:
  - Bit di parità
  - A ripetizione (si triplicano i dati)
  - Parità riga e colonna (correzione)
- **Acknowledgements** (conferme di ricezione):
  - ARQ: Automatic Repeat reQuest
    - Stop and wait
    - Go Back N
    - Selective Repeat
    - Piggybacking (mandare gli ACK assieme ai dati nell'altro senso)
    - Se si riceve qualcosa non in sequenza, si rimanda l'ack dell'ultimo segmento ricevuto in sequenza
  - Possono essere
    - Individuale, selettivo (ho ricevuto questo, questo e questo)
    - Cumulativo (fino al byte N-esimo, tutto bene)
    - Negativo (si informa esplicitamente che qualcosa è andato storto)

Prima di trasmettere bisogna mettersi d'accordo su che tecnica utilizzare. Dato che sul livello di trasporto si appoggiano più applicazioni, per distinguerle nasce il concetto di **porta**. Socket = indirizzo:porta



Le porte possono essere statiche (well known, tipo ssh sulla 20, servizi web sulla 80) oppure dinamiche (assegnate dal sistema operativo al momento della creazione della connessione)

Stop and Wait

Il trasmettitore invia una PDU dopo avere fatta una copia, attiva un orologio (tempo di timeout) e si pone in attesa della conferma di ricezione (acknowledgment - ACK). se scade il timeout prima dell'arrivo della conferma, ripete la trasmissione.
Quando riceve un ACK controlla la correttezza dell'ACK, controlla il numero di sequenza e se l'ACK è relativo all'ultima PDU trasmessa, si abilita la trasmissione della prossima PDU.
Il **ricevitore** controlla la correttezza della PDU, il numero di sequenza; se è giusta invia l'ack e la consegna al livello superiore. Poco efficiente a causa dei ritardi delle attese.

Go back N

Invia tante PDU prima di fermarsi ad aspettare.

- **Finestra di trasmissione**: la quantità massima di PDU in sequenza che il trasmettitore è autorizzato ad inviare in rete senza averne ricevuto riscontro. La sua dimensione dipende dalla memoria allocata in trasmissione
- **Finestra di ricezione**: la sequenza di PDU che il ricevitore è disposto ad accettare e memorizzare. Dipende dalla quantità di memoria allocata in ricezione

Il **trasmettitore** invia N PDU, facendo una copia ed attiva un orologio per quelle N. Si mette in attesa. Se scade il

timeout reinvia tutta la finestra non confermata, altrimenti invia la prossima.
Il ricevitore quando riceve una PDU il numero di sequenza; se è giusta invia l'ack e la consegna al livello superiore.
Considerazioni:

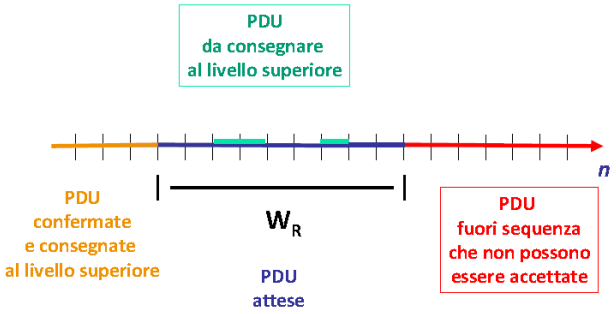
- Semplice da implementare
- È stato pensato per un ambiente in cui ci sono perdite multiple contemporanee (burst di segmenti persi)
- Inefficiente in caso di perdite singole
- Le prime implementazioni di TCP utilizzavano la tecnica per il recupero degli errori Go-ack-N;

Selective repeat

Accettare PDU corrette ma fuori sequenza migliora le prestazioni

- Finestra di trasmissione e finestra di ricezione di dimensione > 1
- Con ack selettivi oppure cumulativi
- Con timer delle singole PDU oppure della finestra

Con ack cumulativi e rimer associati alla finestra:



Il **trasmettitore** invia fino a N PDU, facendone una copia, timer per le N PDU che viene resettato ad ogni trasmissione di PDU. Si mette in attesa. Se scade il timeout prima che si riceva l'ack di chi ha resettato il timeout, reinvio tutte le PDU non ancora confermate.
Il **ricevitore**, riceve una PDU, controlla la correttezza, controlla il numero di sequenza, se è corretta ed in sequenza la consegna su. Se è corretta ma non in frequenza:

- Se è dentro la finestra di ricezione la memorizza
- Se è fuori la scarta

Invia un ack relativo all'ultima PDU ricevuta in sequenza (ack cumulativi). g:

- Esistono diverse modalità di ritrasmissione selettiva e quindi sia sorgente che destinazione devono implementare la stessa modalità
- Complesso
- Efficiente in caso di perdite singole
- Non sovraccarica la rete con ritrasmissioni
- Attualmente usato in TCP, con ack cumulativi.

Piggybacking

Va con flussi bidirezionali. Numerazione ciclica delle PDU in modulo 2^k. Mando l'ack assieme ai dati.

TCP e UDP

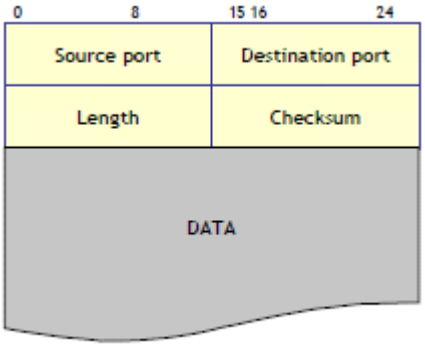
Entrambi offrono queste funzionalità:

- Porte
  - Permettono (de)multiplexing delle connessioni, cioè distinguere con un numero di porta a che applicazione sono destinati dei dati
  - Questi numeri vanno da 0 a 65535, quelli da 0 a 1023 sono well-known.
- Rilevazione errori header

Un socket è identificato da IP sorgente/destinazione, e porta sorgente/destinazione. La porta di sorgente spesso è effemerale, cioè è assegnata dal sistema operativo e serve solo per distinguere tra loro connessioni con gli altri valori simili. (Es due download dallo stesso server sullo stesso computer, tutti su porta 80 (HTTP))

UDP

- Non connesso, non affidabile, stateless. Praticamente non aggiunge quasi niente ad IP, solo i numeri di porta. Se arriva arriva, altrimenti niente. L'ordine di consegna non è garantito. Ideale per richieste singole abbastanza piccole, come le richieste DNS.
- Non c'è l'overhead di dover aprire una connessione, si può inviare subito i dati.
- Spesso scelto per applicazioni multimediali dove si vuole avere un controllo fine sui tempi, ritentativi, eccetera, però bisogna stare attenti a non intasare la rete (non c'è controllo di flusso). La checksum include header, dati e IP sorgente/destinazione.



TCP

Connesso, affidabile, controllo flusso e congestione, con stato. Offre ai livelli superiori uno stream di dati per leggere e uno per scrivere. I dati arrivano in ordine, e non vengono saltati pezzi di dati:

- Per il trasferimento dei dati viene attivata una connessione
- Recupero degli errori
- Consegna ordinata dei segmenti
- Controllo di flusso (non intasare ricevente)
- Controllo congestione

La versione base di TCP usa ACK cumulativi, go back n:

Eventi	Azioni ricevitore TCP
arrivo segmento in ordine, inviato ACK correttamente per tutti segmenti precedenti	**invia ACK**
arrivo segmento fuori sequenza con numero maggiore di quello atteso: vuoto rilevato	**invia ACK duplicato, indicando come numero di sequenza il prossimo byte che si attendedi ricevere**
arrivo di segmento che riempie vuoti parzialmente o completamente	**ACK immediato se il segmento copre parte iniziale della finestra**

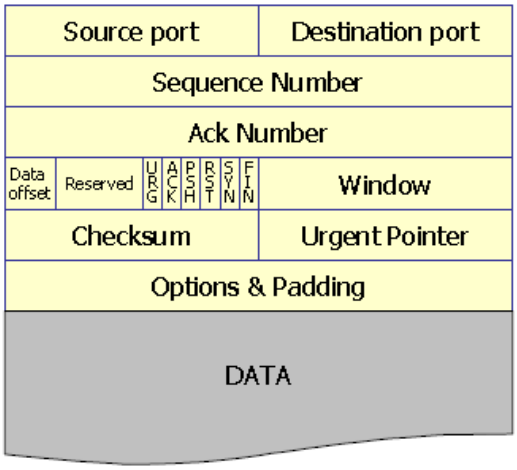
- Dimensioni finestra variabili durante la comunicazione
- Stima RTT per impostare il timeout
- Annuncia negli ACK lo spazio libero nel buffer di ricezione per controllare velocità trasmettitore (controllo di flusso)
- Velocità in assenza di errori: finestra trasmissione / RTT
- Il trasmettitore si autoimpone una dimensione massima di finestra in base alle perdite riscontrate per mancato arrivo di ACK.

TCP impiega un controllo end-to-end basato su controllo della dimensione della finestra del trasmettitore:

- controllo di flusso: il ricevitore impone la dimensione massima della finestra del trasmettitore, indicando negli ACK la finestra di ricezione disponibile
- controllo di congestione: il trasmettitore si autoimpone una dimensione massima della finestra in funzione delle perdite riscontrate per mancato arrivo di ACK

Segmento TCP

Può variare dal solo header per gli ack fino ad un valore massimo MSS concordato col ricevitore, che dipende dallo stream dei livelli superiori.



- Ack number: indice prossimo byte da ricevere (valido solo con ACK settato)
- HLEN: lunghezza header in dword
- Flags
  - URG
  - ACL
  - PSH: forza passaggio immediato dei dati all'applicazione ricevente
  - RST: reset connessione: il serve non ce la fa, non ha risorse, chiede di non ritentare
  - SYN: apertura connessione
  - FIN: chiusura connessione
- Nella pratica, PSH e URG non sono mai usati
- Receiver window: numero di byte, a partire da quello nel campo ACK, che il ricevitore è disposto ad accettare (max 64K)
- Checksum su dati, header e pseudoheader con IP e tipo protocollo
- Urgent pointer: offset rispetto a seq, esempio ctrl-C in telnet

- Maximum Segment Size: riferito ai dati, senza contare l'header TCP
- Maximum Transmission Unit: massima dimensione pacchetti fornita dal data link, in base a questa si sceglie MSS

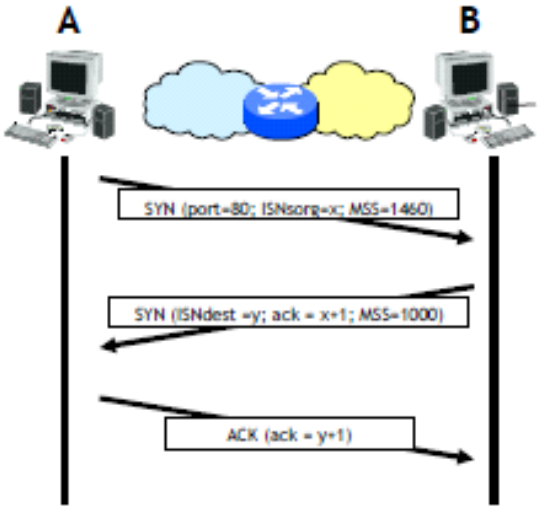
Aiuto qualcuno mi spieghi relazione RTT, rwnd, banda, banda*ritardo

Stati della connessione TCP, apertura e chiusura

- SYN_SENT: si ha inviato il SYN per richiedere al server la connessione
- ENSTABILISHED: si ha ricevuto il SYN ACK del server, si trasmettono e ricevono dati dell'applicazione
- FIN_WAIT_1: si ha deciso di chiudere la connessione, per quanto riguarda la scrittura di dati, ed il FIN è stato inviato
- FIN_WAIT_2: anche l'altro computer ha chiuso la connessione, dobbiamo inviargli un ACK per il suo FIN
- TIME_WAIT: aspettiamo 30 secondi prima di chiudere, per assicurarsi che l'ultimo ACK sia stato ricevuto
- CLOSED: tutte le risorse sono state liberate

Aprire una connessione

si manda un SYN e l'initial sequence number

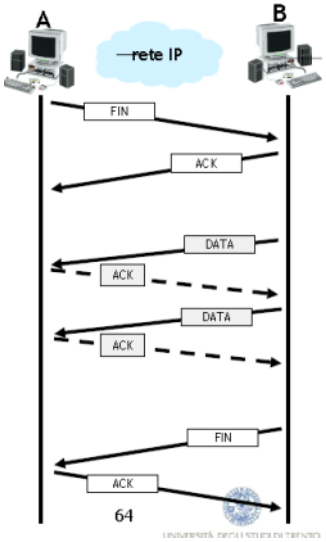


A annuncia a B di poter accettare solo segmenti la cui dimensione massima del campo dati è pari a 1460 byte;
B annuncia ad A che la sua MSS è invece di 1000 byte; le due stazioni dunque trasmetteranno segmenti con un campo dati lungo al massimo 1000 byte

- Anche se A chiude la trasmissione, può continuare a ricevere

Chiudere una connessione

La terminazione deve avvenire da entrambe le parti



- La stazione che non ha più dati da trasmettere e decide di chiudere la connessione invia un segmento FIN (segmento con il campo FIN posto a 1 e il campo dati vuoto)
- La stazione che riceve il segmento FIN invia un ACK e indica all'applicazione che la comunicazione è stata chiusa nella direzione entrante.

Poi si riesegue questa procedura nell'altro lato.

Calcolo del Retransmission Timeout

$$SRTT = (1- \alpha)\ SRTT + \alpha\ RTT$$

RTT = valore del campione attuale di RTT
SRTT = stima smoothed di RTT

Tipicamente a = 0.125 (1/8)
In pratica, la vecchia stima e l'ultimo roundtrip-time, pesato 1/8

$$RTTVAR = (1- b)\ RTTVAR + b\ |SRTT - RTT|$$

RTTVAR = stima smoothed della varianza di RTT

Tipicamente b= 0.25 (1/4)

La stima è più reattiva
In pratica, il vecchio valore, e la differenza attuale, pesata 1/4

RTO = SRTT + 4 RTTVAR
In pratica, la stima smussata, più 4 volte la varianza

Quando scade il timeout, si ritrasmette, e si raddoppia il timeout (probabilmente il ritardo è causato da una congestione)
Appena si riceve la risposta, si riporta RTO al valore calcolato.

0.2s < RTO < 60s+

Controllo di flusso

- Controllo di flusso: azione preventiva finalizzata a limitare l'immissione di dati in rete in caso il ricevitore non li possa ricevere
- Controllo della congestione: azioni da intraprendere come reazione alla congestione di rete (e non del ricevitore)

Problemi associati al controllo di flusso:

- Com'è possibile da parte di una stazione determinare la capacità e lo stato del ricevitore?

Stop and wait è di per sé una tecnica dicontrollo di flusso, ma inefficiente. Utilizziamo la **finestra scorrevole**:

- Se la finestra è troppo piccola sottoutilizzo la banda
- Impostiamo dinamicamente RCWIN (è la dimensione della finestra ricevente)
- I dati provenienti dall'applicazione vengono ordinati in una sequenza di byte numerati progressivamente (a partire dall'ISN, Initial Sequence Number)
- In base alla dimensione della MSS, tali byte sono suddivisi in segmenti
- Il meccanismo a finestra scorrevole è applicato alla sequenza di segmenti così determinata

Le ritrasmissioni rischiano di essere inutili nel caso in cui i pacchetti vengono buttati via per mancanza di risorse (congestione).

Congestione

Stato della rete quando il traffico offerto è > della capacità della rete. In caso di congestione, il controllo di flusso a finestra, protegge implicitamente anche la rete:

- Se la rete è congestionata, arriveranno meno riscontri e quindi il tasso di immissione diminuisce automaticamente
- l'attesa dello scadere del timeout lascia un intervallo di tempo in cui nonvengono immessi nuovi segmenti

Causa perdite:

- Errori trasmissione (singolo segmento)
- Congestione (più segmenti)

Potrebbe non bastare, bisogna scegliere una dimesnione ottimale della finestra, tale finestra è dimanica e si chiama **Congestion window,** la cui dimensione è scelta da vari algoritmi:

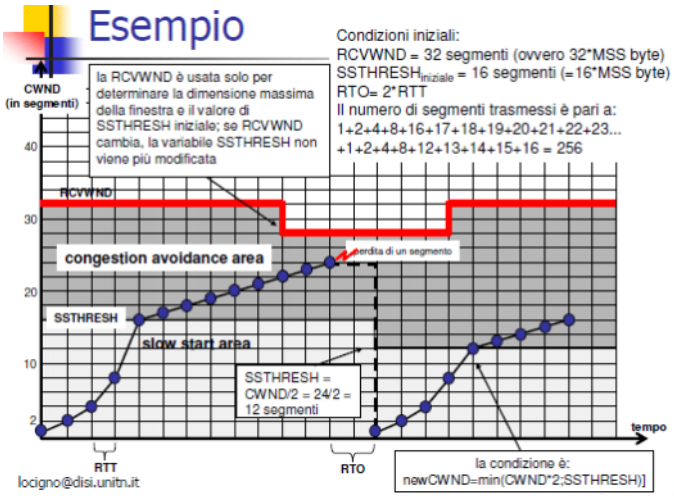
- **Slow Start**: ad ogni ACK, raddoppia dimensione finestra
- **Congestion Avoidance**: aumenta linearmente
- **Fast Recovery - Retransmit**: due algoritmi specificatamente progettati per gestire le perdite singole
    - Fast recovery: la CWND viene chiusa eccessivamente
    - Fast tretransmit: il segmento perso viene subito ritrasmesso
- **SACK**

Questi algoritmi vengono combinati assieme.

Abbiamo quindi queste variabili:

- CWND (finestra trasmissione)
- RECWND (finestra ricezione, è il max limite della CWND)
- SSTHRESH (Slow Start Threshold: soglia oltre cui usare Congestion Avoidance)

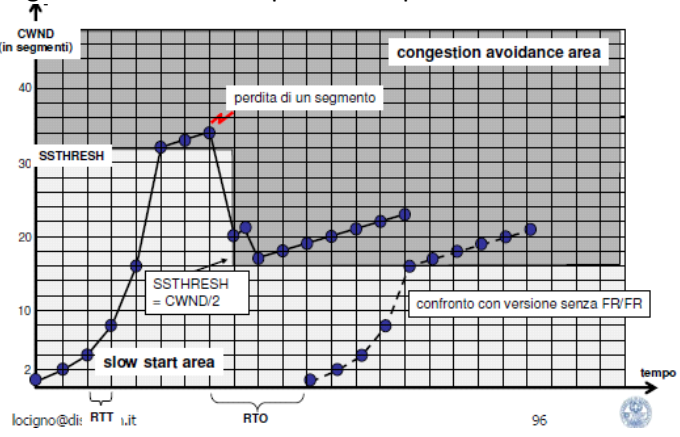
- In caso di errore o di perdita dei segmenti:
  - la trasmissione si interrompe (la finestra non si sposta non permettendo l'immissione di nuovi segmenti in rete)
  - si attende lo scadere del timeout RTO
  - allo scadere di RTO, si pone
    - SSTHRESH = CWND / 2
    - CWND = 1
  - si riprende a ritrasmettere con la tecnica di Go-back-N
  - l'evoluzione della finestra segue l'algoritmo di Slow Start fino al raggiungimento della SSTHRESH...
- In caso di errore o perdita consecutiva
  - al primo errore o perdita, quando il timeout scade e si ritrasmette il primo segmento non riscontrato, il timeout per quel segmento viene raddoppiato (exponential back off)
    - RTO_new=2*RTO_old
  - la CWND rimane = 1, mentre SSTHRESH si pone = 2 segmenti



- Prima di inviare gli ACK, aspetto un po di tempo, così magari ricevo dall'applicazione dei nuovi dati da trasmettere, e metto l'ACK nello stesso segmento, se non arriva, invio l'ACK da solo. Inoltre se ricevo altri pacchetti dall'altra parte, invio un ACK unico che li riscontra tutti.
- Se ricevo un segmento non ordinato, e mi rimane un buco, trasmetto immediatamente l'ACK duplicato
- Se ricevo dei segmenti che riempiono anche parzialmente il buco (dall'estremità inferiore), invia subito ACK
- In caso di perdita di un solo segmento, la reazione è eccessiva. Quindi, si ritrasmette subito il segmento perso

Fast Retransmit – Fast Recovery

- Arrivano 3 ACK duplicati (1+3) per lo stesso segmento: è improbabile che sia dovuto a dei pacchetti in disordine, quindi probabilmente il destinatario ha perso un pacchetto.
  - SSTHRESH = CWND / 2
  - Si ritrasmette senza attendere il RTO (**fast retransmit**)
  - CWND = SSTHRESH + 3
  - Per ogni successivo ACK duplicato, CWND aumenta di 1
- Quando si riceve l'ACK con la conferma del segmento ritrasmesso
  - CWND = SSTHRESH
  - Si procede con Congestion Avoidance
- Segmento ritrasmesso perso? Si riparte da 1 in slow start



- I dati in arrivo vanno letti dell'applicazione. Se però questa è troppo lenta, il buffer si riempie, e TCP deve fermarsi (controllo di flusso).
- I rallentamenti possono anche avvenire per congestione della rete (controllo di congestione)
- Ciascuno dei due computer è a conoscenza dell'ultimo byte confermato dall'altro, e dello spazio libero nella finestra del ricevente. Il mittente non invia quindi segmenti riferiti a byte che sono fuori dalla finestra del ricevente (il ricevente li scarterebbe)
- Se la finestra di ricezione va a zero, il mittente non può inviare dati, ma allora quando si libera spazio non può sapere che si è liberato. Allora, il mittente deve continuare a inviare segmenti con un (1) byte di dati, così nel riscontro si potrà leggere se la finestra è cresciuta, e ricominciare a trasmettere seriamente.

Congestione in generale

- Scenario 1, due mittenti e un router con buffer limitati: più di tot dati non passano per via della capacità della rete, e allora si accodano nei buffer. Il throughput diventa massimo,ma il ritardo medio cresce verso l'infinito.
- Scenario 2, due mittenti e un router con buffer finiti
- Scenario 3, quattro mittenti, router con buffer finiti e percorsi multihop

- Controllo congestione end-to-end: il controllo di congestione avviene sullabase di osservazioni del comportamento della rete
- Controllo congestione assistito dalla rete: i router inviano un feedback esplicito al mittente sullo stato della rete
  - Direttamente al mittente
  - Lo segnalano nel pacchetto da inviare al destinatario, che poi notificherà il mittente

Congestione TCP

- IP non offre feedback espliciti di congestione
- TCP si autoimpone un limite sulla frequenza di invio, in funzione della congestione percepita.
- La quantità di dati inviabili è limitata da Congestion Window (un limite che ci imponiamo) e da Receiver Window (quanto spazio ha il destinatario nel buffer)
- La frequenza di invio è circa CongWin/RTT
- Definiamo "evento di perdita" la scadenza di un timeout o la ricezione di tre ACK duplicati.
- Se riceviamo gli ACK, aumentiamo man mano la finestra di congestione (di invio) di uno alla volta (**additivo**). Se riceviamo tutti i riscontri, significa che probabilmente che esiste banda inutilizzata che iniziamo a sfruttare un po alla volta
- Slow start: all'inizio è un peccato perdere tempo a crescere linearmente, quindi **raddoppiamo**, finché non superiamo la soglia SSTHRESH, dopo di che andiamo in Congestion Avoidance e proseguiamo linearmente, di 1 MSS alla volta
- Se qualcosa va storto, impostiamo **SSTHRESH** a **metà** della dimesione attuale della **CongWin**, e poi:
  - Se riceviamo tre ACK duplicati, **dimezziamo** CongWin (Fast Recovery)
  - Se un timeout è scaduto, CongWin **riparte da 1** MSS
  - La prima versione di TCP (Tahoe) ripartiva sempre da 1, quella nuova, Reno usa Fast Recovery. Se riceve ACK duplicati, significa che almeno la rete riesce a consegnare parte dei pacchetti, quindi non è un problema poi così grave
- All'inizio SSTHRESH viene messa molto alta, in modo che non abbia praticamente nessun effetto, cioè continuiamo a raddoppiare finché non avviene un problema, e lì settiamo veramente SSTHRESH

Throughput

Senza contare i timeout, da cui comunque ci si riprende piuttosto in fretta, il throughput è approssimabile tra metà e 1 di W/RTT, quindi 3/4 W / RTT, dove W è la dimensione della finestra a cui si verificano solitamente le perdite di pacchetti.
Throughput medio di una connessione TCP:

$$\frac{1.22 \cdot MSS}{RTT\sqrt{L}}$$

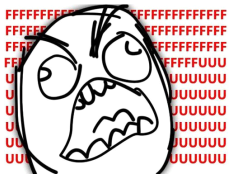
Futuro di TCP

- TCP si è evoluto negli anni e continua a farlo
- Quello che andava bene per SMTP, FTP, Telnet non va necessariamente bene per l'odierna internet dominata da HTTP, e nemmeno per i servizi futuri che oggi neppure immaginiamo
- Alta velocità:
    - 10 Gbps, 1500 byte per segmento, RTT di 100 ms
    - La dimensione media della finestra sarebbe di 83.333 segmenti: sono tantissimi, e se se ne perde uno?

Layer 3 - Routing

domenica 15 maggio 2011
00:08

livello del modello OSI e TCP/IP che si occupa del traffico dei pacchetti (vd. Datagramma) dalla macchina sorgente alla macchina destinataria, superando gli ostacoli dovuti alle diversità dei protocolli utilizzati tra le reti (eterogenee) come alla grandezza dei pacchetti (MTU).
A tal fine i router si occupano di trasferire i pacchetti attraverso i diversi segmenti di rete, evitano anche le linee sovraccariche (controllo della congestione) e definendo percorsi secondo tabelle statiche o dinamiche. In questo livello si risolvono le interconnessioni tra reti dissimili nei protocolli (es: IP, IPX, CLNP) ma non l'affidabilità. Altri protocolli del Livello di rete: ICMP, ARP, RARP, BOOTP. Si veda anche un'illustrazione globale del modello TCP/IP.



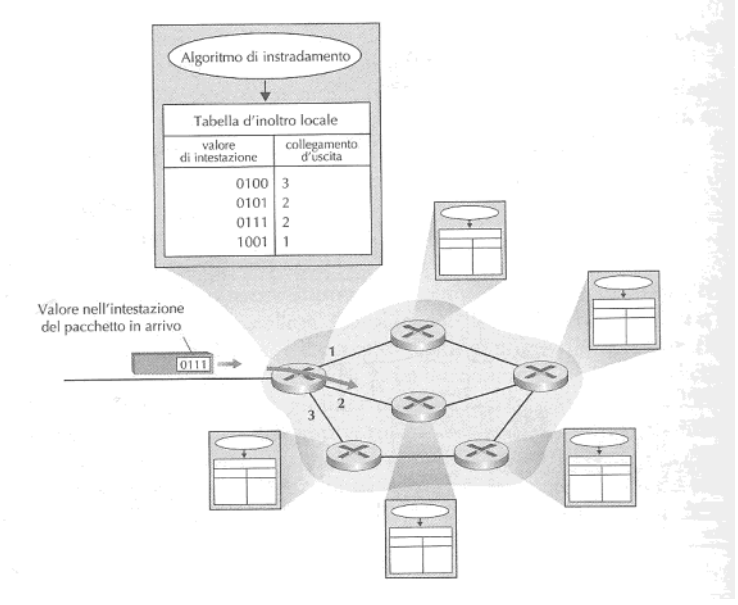
NB: le slide trattano solo algoritmi di instradamento, il resto l'ho preso qui e li fra le slide, ma il riassunto è fatto dal libro

Inoltro ed instradamento

Il ruolo del livello di rete è quello di trasferire i pacchetti da un host all'altro:

- **Inoltro (forwarding)**: trasferire i pacchetti dall'interfaccia di ingresso a quella di uscita
- **Instradamento (routing)**: stabilire il percorso che i pacchetti devono fare

Instradamento è il percorso che i pacchetti che devono eseguire, inoltro è quello che succede quando si passa da un nodo all'altro all'interno del percorso.
Per inoltrare i pacchetti i router hanno una tabella d'inoltro (forwarding table). Estraggono un indice dal campo di intestazione dei pacchetti e leggono la tabella da quell'indice. Il risultato indice quale interfaccia di collegamento deve essere utilizzata.



L'algoritmo di instradamento determina il contenuto delle forwarding table. Il router riceve dei messaggi di instradamento che vengono utilizzati per riempire queste tabelle.

Commutatore di pacchetto (packet switch): generico dispositivo che si occupa del trasferimento dell'interfaccia di ingresso a quella di uscita. Alcuni leggono gli header del livello di rete (router), altri si basano su altri livelli (trasporto).

Instaurazione della connessione (connection setup): dipende dal tipo di rete.

Modelli dei servizi di rete

Definiscono le caratteristiche del trasporto end-to-end di dati tra il trasmittente ed il ricevente. Sono in pratica i servizi che il livello di rete quando qualcosa gli viene dal livello di trasporto:

- **Consegna garantita**
- **Consegna garantita con ritardo limitato**: il pacchetto è sicuramente consegnato entro N unità di tempo
- **Consegna in ordine**
- **Minima ampiezza di banda garantita:** finchè trasmettiamo a velocità minore o uguale di quella garantita non si possono verificare perdite di pacchetti
- **Jitter limitato:** il lasso di tempo fra la trasmissione di due pacchetti è uguale a quello di ricezione
- **Servizi di sicurezza:** crittorafia

- Many others

Il livello di rete di Internet mette a disposizione solo il servizio "best-effort" che in realtà non garantisce nulla.

Architettura di rete	Modello di servizio	Garanzia sulla banda	Garanzia di consegna	Ordina-mento	Temporizza-zione	Indicazione di congestione
Internet	Best-effort	Nessuna	Nessuna	Non garantito	Non mantenuta	No
ATM	CBR	Tasso costante garantito	Sì	Rispettato	Mantenuta	Non si verifica congestione
ATM	ABR	Minimo garantito	Nessuna	Rispettato	Non mantenuta	Sì

I due modelli CBR e ABR di ATM invece offrono vari servizi:

- **Servizi di rete ATM a bit rate costante (CBR):** il primo dei modelli ATM ad essere standarizzato. Ha come scopo dotare ad un flusso di pacchetti un canale virtuale che abbia caratteristiche simili a quello dedicato ad una banda fissa host to host. Jitter, ritardo end-to-end e percentuale di celle perdute o consegnate in ritardo sono inferiori ai valori specificati (concordati tra l'host di invio e la rete ATM).
- **Servizi di rete ATM a bit rate disponibile (ABR):** è un po' migliore di Internet: possono essere perse delle celle ma il loro ordine rimane inalterato, è garantito un minimo tasso trasmissivo (MCR).

Reti a circuito virtuale e datagramma

Il livello di trasporto può offrire un servizio senza connessione o orientato alla connessione. Ad esempio il livello di trasporto di Internet mette a disposizione TCP ed UDP. Anche il livello di rete può offrire servizi con o senza connessione (molto simili a quelli del livello di Trasporto). Differenze:

- I servizi di rete vengono forniti al livello di trasporto (quelli di trasporto al livello di applicazione)
- In quasi tutte le architetture (Internet, ATM, frame-relay ecc) il livello di rete offre un servizio host-to-host senza connessione o con connessione ma non entrambi. Le reti che mettono a disposizione solo il servizio con connessione sono dette reti virtuali, quelle senza connessione sono dette a datagramma

Reti a circuito virtuale

Internet è una rete a datagrammi. ATM , frame-relay & co sono reti a circuito virtuale. Un circuito virtuale consiste in:

- Un percorso fra origine e destinazione
- Numeri VC per ciascun collegamento lungo il percorso
- Righe nelle tabelle di inoltro per ciascun router

Il pacchetto di un circuito virtuale ha un numero VC nel proprio header. Dato che ogni circuito virtuale può avere un numero VC diverso per ogni collegamento ogni router lo deve sostituire con il nuovo numero rilevato dalla tabella d'inoltro: ogni arco del nodo ha un VC diverso. Inizialmente il pacchetto ha il VC del primo arco, poi man mano che gli attraversa bisogna aggiornarlo col VC dell'arco successivo. Esempio di tabella di instradamento di un router

Interfaccia in ingresso	Numero VC entrante	Interfaccia in uscita	Numero VC uscente
1	12	2	22
2	63	1	18
3	7	2	17
1	97	3	87
...	...	...	...

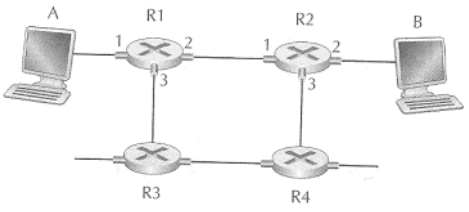


Figura 4.3 Rete a circuito virtuale.

Sostituire il VC di arco in arco serve per:

- Diminuire la quantità di bit assegnati al VC (risparmio banda)
- Disponendo di più numeri, ciascun nodo può scegliere indipendentemente dagli altri che strada fargli fare

Nelle reti a circuito virtuale i router devono mantenere informazioni sullo stato delle connessioni attive. Ogni volta che stabilisce una nuova connessione aggiunge una riga alla tabella, ogni volta che ne rilascia una, toglie una riga.
Il ciclo di vita dei circuiti virtuali è articolato in 3 fasi:

- **Instaurazione:** il livello di trasporto del mittente contatta quello di rete, specifica l'indirizzo del destinatario ed

aspetta che il protocollo di rete imposti il circuito virtuale. Il livello di rete crea il percorso ed imposta i numeri VC lungo il percorso, aggiunge una riga alle tabelle dei router può anche riservare nel frattempo delle risorse (ampiezza di banda ecc).

- **Trasferimento di dati**
- **Terminazione**: quando uno dei due capi si stufa, lo comunica al livello di rete che distrugge il circuito

Stabilire il circuito virtuale a livello di trasporto coinvolge solo due terminali (usa tutto quello che fa il livello di rete fregandosene dei dettagli), mentre in quello di rete vengono coinvolti tutti i nodi.
I messaggi inviati per iniziare o concludere un circuito virtuale sono detti **messaggi di segnalazione**, i protocolli utilizzati per scambiare questi messaggi sono detti **protocolli di segnalazione**.

Reti a datagramma

Il pacchetto deve contenere l'indirizzo del destinatario (senza alcun circuito virtuale). Non esistono circuiti virtuali. I pacchetti seguono un percorso attraverso i router che usano l'indirizzo di destinazione per inoltrarli. C'è una tabella dalla quale i router leggono l'interfaccia di uscita in base all'indirizzo di destinazione.
L'idea di avere in ogni router un tabellone con 4 miliardi di righe (1 per ogni indirizzo disponibile) è assolutamente impensabile:
I router memorizzano parte di questo tabellone, e riescono ad inoltrare correttamente pacchetti ad indirizzi a loro sconosciuti leggendone i prefissi. Se nessun prefisso corrisponde a qualche riga della tabella c'è una riga di "default" dove vengono inviati gli indirizzi sconosciuti. Se si verificano corrispondenze multiple, il router sceglie quella che corrisponde per il numero più lungo di bit (**regola di corrispondenza a prefisso più lungo**).
Per facilitare questo meccanismo gli indirizzi vengono assegnati in modo gerarchico. I router nelle reti a datagrammi non conservano informazioni sullo stato della connessione ma solo su quello di inoltro. Per renderla usabile quindi bisogna spesso aggiornare queste tabelle (circa ogni 5 minuti).
Dato che le tabelle possono essere modificate in qualsiasi istante, i pacchetti possono raggiungere la destinazione in modo disordinato.

Origini delle reti a circuito virtuale e datagramma

Internet è nato così ed è rimasto così. Nato pensato per non complicare l'interlacciamento fra i vari tipi di calcolatori, offre praticamente nessun servizio. I servizi aggiuntivi, Web, DNS ecc vengono forniti dai livelli superiori. IP, fornendo 0 servizi è particolarmente flessibile e permette l'interlacciamento di reti che utilizzano protocolli svariati.
Le reti a circuito virtuale invece derivano dalla telefonia: serviva un collegamento fisso e costante fra i due host per permettere la comunicazione.

Che cosa si trova all'interno di un router?

Nei router possiamo individuare quattro componenti:

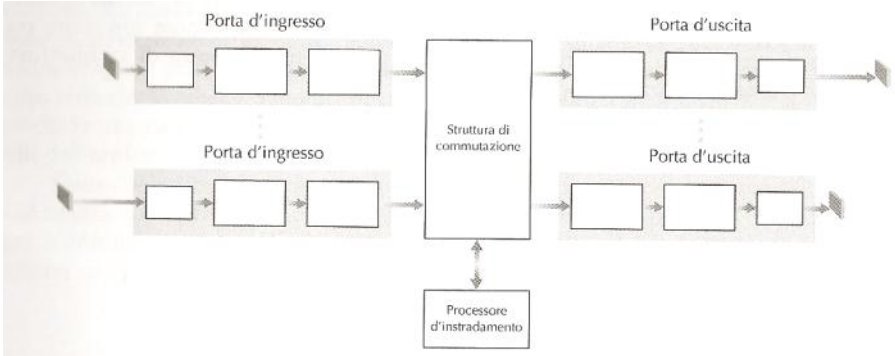
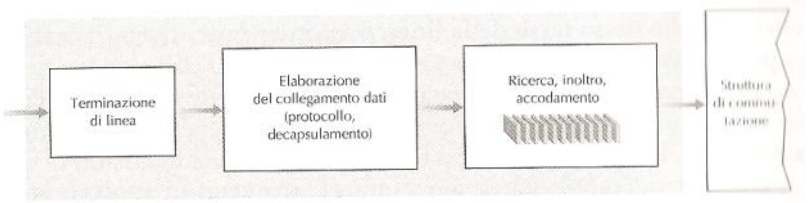


Figura 4.6 Architettura del router.

- **Porte d'ingresso (input port):** svolgono a livello fisico le funzioni di terminazione di un collegamento fisico in ingresso al router. Svolgono anche funzioni di ricerca ed inoltro. I pacchetti ricevuti nelle porte in ingresso vengono inoltrati alla struttura di commutazione. Spesso più porte sono raggruppate su un'unica line card.
- **Struttura di commutazione (switching fabric):** connette fisicamente le porte in ingresso a quelle d'uscita (è una rete interna al router).
- **Porte d'uscita (output port):** memorizzano i pacchetti forniti dalla struttura di commutazione e trasmettono sul collegamento in uscita.
- **Processore d'instradamento (forwarding processor):** esegue i protocolli di instradamento, conserva le informazioni di instradamento ed effettua funzioni di gestione della rete all'interno del router.

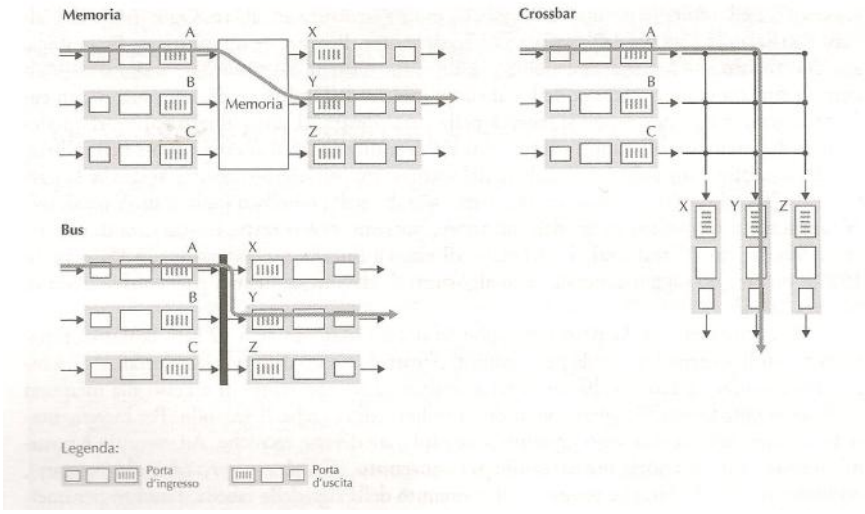
Porte d'ingresso



Elaborazione alle porte d'ingresso.

Lavorano a livello fisico (terminazione della rete elettrica) e a livello di data link. Eseguono anche operazioni di inoltro per non creare un collo di bottiglia unico**: inoltro decentralizzato** (forwarding processor). Se non riescono a inoltrare per qualche motivo, mandano i pacchetti al processore d'instradamento (esempio: un router di dorsale deve essere veloce, se nessuno gli da una mano esplode. L'inoltro decentralizzato serve a garantire che il tempo di lettura dalle tabelle di instradamento non alteri il tasso della linea. Dato che serve tanta velocità non possiamo andare a cercare nelle tabelle in O(n). Esse attualmente sono implementate tramite alberi di ricerca. Ogni bit dell'indirizzo corrisponde ad un livello diverso nell'albero. Ma anche così siamo troppo lenti: si utilizzano **memorie indirizzabili per contenuto** (**CAM**). L'algoritmo migliore attualmente è log(N).

Struttura di commutazione

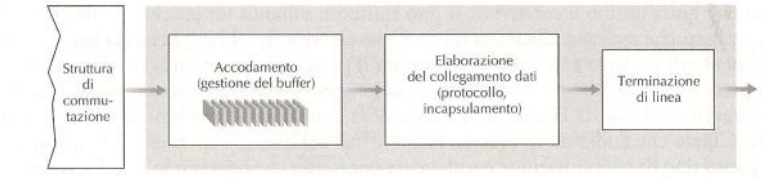


La commutazione è ottenuta in vari modi:

- **Commutazione in memoria:** old style. Le porte non avevano funzionalità d'inoltro, se ne occupava tutto lui.
- **Comunicazione tramite bus:** le porte d'ingrasso trasmettono alle porte d'uscita tramite un bus condiviso.
- **Comunicazione attraverso rete d'interconnessione**

Dove si verifica l'accodamento?

Se le code diventano grosse i buffer si possono riempire e quindi dei pacchetti andranno persi.



igura 4.9 Elaborazione alle porte d'uscita.

Se la velocità di commutazione è pari almeno a n volte la velocità della linea d'ingresso non si verificano accodamenti nelle porte d'ingresso. Possono però accumularsi sulle porte d'uscita. In questo caso lo **schedulatore dei pacchetti** deve stabilire in quale ordine trasmetterli:

- **FCFS:** first come first served.
- **WFQ:** accodamento equo ponderativo. Guarda quanti pacchetti accodati hanno le varie connessioni e li ripartisce equamente.

Se però di memoria non ce n'è più bisogna decidere quali pacchetti scartare:

- **Drop tail:** quello appena arrivato.
- **Gestione attiva della coda (active queue management)**: in alcuni casi è conveniente eliminare un pacchetto prima che il buffer sia pieno:
  - **Random early detection**: mantiene una media ponderata della lunghezza della coda di output. Se la media è inferiore ad una soglia minima i pacchetti vengono messi in coda, altrimenti vengono marcati o eliminati.

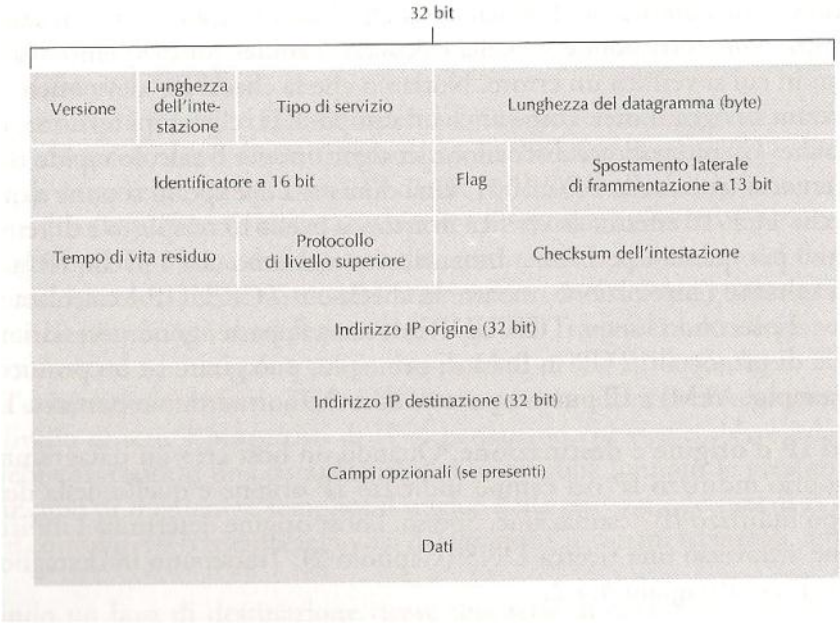
Se la struttura di commutazione non è sufficientemente rapida si può verificare un accodamento nelle porte d'ingresso. Può verificarsi il fenomeno del **blocco in testa alla fila**: quando un pacchetto in ingresso deve attendere

perché ce n'è già un altro in fila.

Protocollo internet (IP): inoltro e indirizzamento in internet

- IPv4: 32 bit

Formato dei datagrammi



- **Numero di versione**: IPv4-6.
- **Lunghezza dell'intestazione**: per gestire le opzioni variabili.
- **Tipo di servizio**: flag per descrivere le caratteristiche desiderate (consegnalo subito ecc).
- **Lunghezza del datagramma:** intestazione + dati.
- **Identificatore, flag, spostamento laterale di frammentazione:** hanno a che fare con la frammenzazione.
- **Tempo di vita:** decrementato ad ogni salto, se 0 drop.
- **Protocollo:** di livello superiore.
- **Checksum dell'intestazione:** da controllare con coppie di bytedell'intestazione in complemento a 1.
- **Opzioni:** deprecated in IPv6.

Frammentazione dei datagrammi IP

MTU: massima unità di trasmissione. Essendoci MTU dobbiamo frammentare i pacchetti se sono troppo grandi. I frammenti devono poi essere riassemblati. Per riassemblarli si leggono i campi **Identificatore, flag, spostamento laterale di frammentazione.** La frammentazione viene usata nei DoS: i bersagli possono collassare cercando di ricostruire i datagrammi.

Indirizzamento IPv4

Ogni interfaccia di host e router ha indirizzo IP univoco (eccetto NATTAMENTI vari)

- Rete
- Sottorete
- Maschera di sottorete (/24)
- Classes Interdomain Routing: assegnazione degli indirizzi IP gerarchica
- Bit di rete = prefisso
- Broadcsast: x.x.x.255

Come ottenere un blocco di indirizzi

Si contatta l'ISP, si ottiene un blocco e lo si divide come vuole. Anche l'ISP richiede un blocco di indirizzi, che gli è fornito dall' ICANN. ICANN deve anche gestire i server DNS.

Come ottenere l'indirizzo di un host: DHCP

Dynamic Host Configuration Protocol: gli indirizzi vengono assegnati automaticamente a chi è collegato ad una sottorete. Gli indirizzi sono temporanei. DHCP è un protocollo plug-and-play. Facilita l'amministrazione della rete. Da anche dei vantaggi agli ISP, tiene degli indirizzi liberi: 200 ut/500000 IP.
È un protocollo client server: per ottenere un IP temporaneo bisogna conoscere l'indirizzo del server DHCP.

- L'host manda un **messaggio d'identificazione DHCP** (invia un pacchetto UDP sulla porta 67 all'indirizzo di broadcast).
- Il server riceve il messaggio e risponde con un **messaggio di offerta DHCP**.
- Il client riceve la/e offerta/e del/i server DHCP e ne sceglie una, confermandolo al server specifico.
- Il server riceve la conferma e risponde con un ACK DHCP per comunicare al client che tutto è andato a buon fine

Traduzione degli indirizzi di rete

Network Address Translation (NAT):

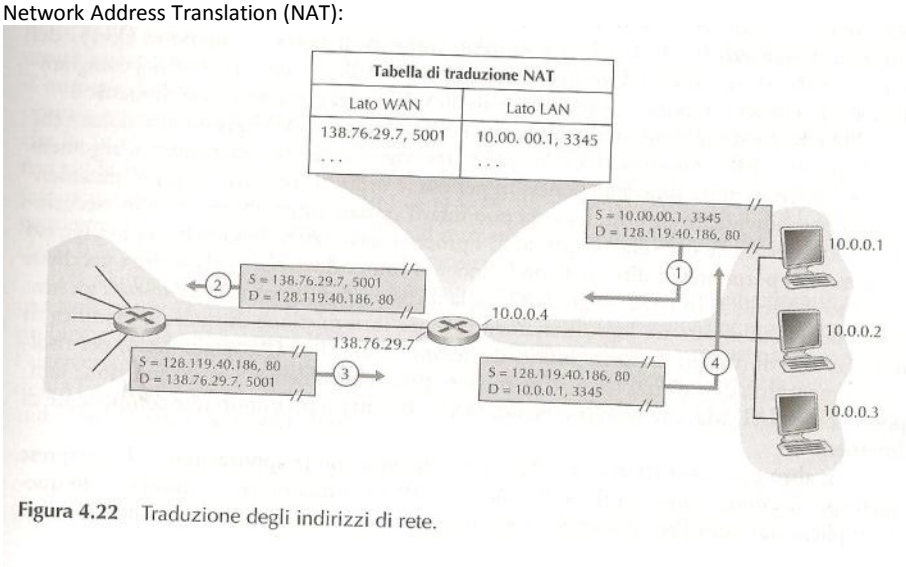


Figura 4.22 Traduzione degli indirizzi di rete.

Tabella di traduzione NAT.

UPnP

Universal plug and play: permette all'host di configurare un NAT vicino. Deve essere supportato sia dal host che dal NAT. Comodo quando un'applicazione P2P vuole comunicare con un'altra su una specifica porta: da 10.10.0.1:3232 (privato) a 125.57.98.156:5001 (pubblico) dove 5001 è una porta scelta dall'applicazione. In pratica serve per far coumunicare un host esterno con uno protetto da NAT.

Internet control message protocol (ICMP)

Il livello di rete di internet presenta 3 componenti principali:

- Protocollo IP
- Protocolli di instradamento (RIP, OSPF, BGP)
- ICMP

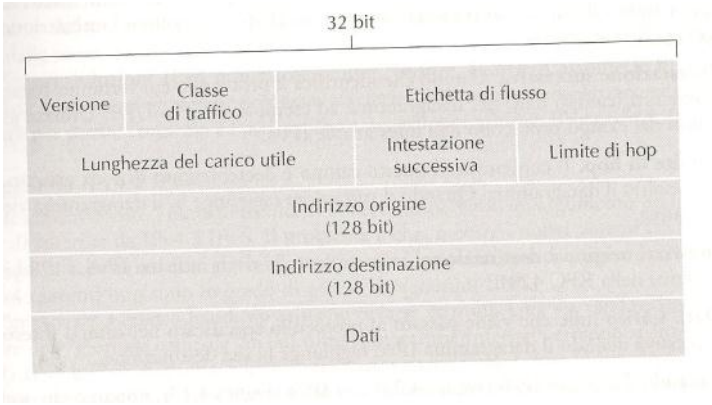
ICMP viene usato da host e router per scambiarsi informazioni sugli errori a livello di rete. I messaggi ICMP vengono trasportati nei datagrammi IP (è quindi di livello più alto). Contengono due campi: codice, tipo che trasportano informazioni sul datagramma che ha generato l'errore.

Tipo ICPM	Codice	Descrizione
0	0	risposta eco (a ping)
3	0	rete destinazione irraggiungibile
3	1	host destinazione irraggiungibile
3	2	protocollo destinazione irraggiungibile
3	3	porta destinazione irraggiungibile
3	6	rete destinazione sconosciuta
3	7	host destinazione sconosciuto
4	0	riduzione (controllo di congestione)
8	0	richiesta eco
9	0	annuncio del router
10	0	scoperta del router
11	0	TTL scaduto
12	0	errata intestazione IP

IPv6

Nato negli anni 90 per ovviare al problema della carenza di IP (IPv4). Gli ideatori integrarono nuove funzionalità.

Formato dei datagrammi IPv6



Cambiamenti rispetto a IPv4:

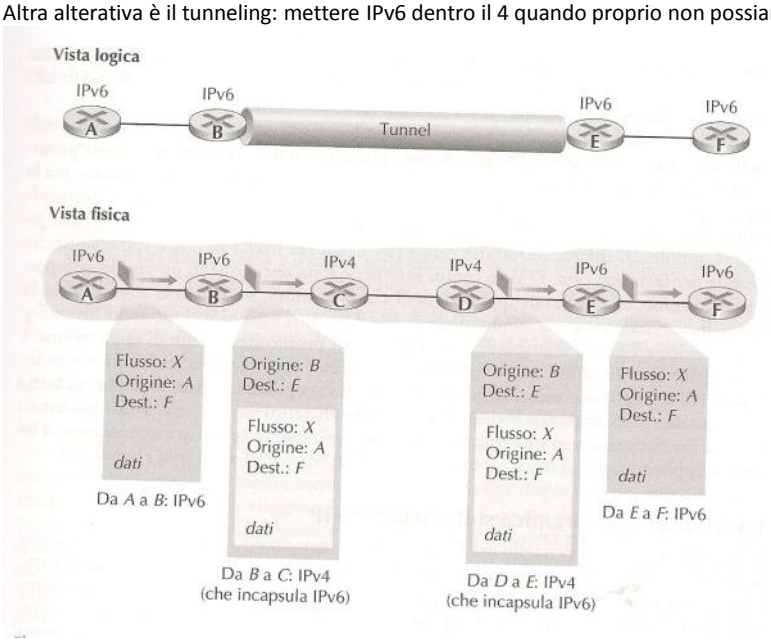
- **Possibilità d'indirizzamento:** da 32 a 128 bit di indirizzo
- Oltre a unicast e multicast è stato inserito **anycast** (per consegnare a gruppi)
- **Intestazione a 40 byte a linea di flusso:** eliminati o resi opzionali alcuni campi IPv4. Ora c'è una parte fissa di 40 byte ed una variabile per le opzioni
- **Architettura e priorità di flusso:** audio-video-chi paga più sono considerati flusso, non applicazioni tradizionali
- **Versione**: IPv4/6
- **Classe di traffico:** priorità (tipo TOS IPv4)
- **Lunghezza del carico utile**
- **Protocollo Intestazione successiva:** TCP/UDP ecc
- **Limite di hop:** se 0 scarta
- **Indirizzi origine, destinazione e dati**

Vengono invece eliminati:

- **Frammentazione/riassemblaggio:** non è consentita frammentazione fra router intermedi, fatte solo da origine, destinazione. Se qualcuno riceve un messaggio troppo grande lo elimina e manda un ICMP
- **Checksum dell'intestazione:** se occupano già altri livelli
- **Opzioni**

Passaggio da IPv4 a IPv6

I nuovi sistemi sono retrocompatibili, quelli vecchi non possono usare IPv6. Si pensa all'approccio doppia pila: nodi capaci di rivevere ed inviare sia il 4 che il 6, che hanno tabelle valide sia per il 4 che per il 6. Il DNS dovrà dare un 4 o un 6 in base al nodo.
Altra alterativa è il tunneling: mettere IPv6 dentro il 4 quando proprio non possiamo usarlo.



Sicurezza di IP

Nato senza bisogno di sicurezza, ora serve, quindi ci si inventa nuovi protocolli di rete che la offrano
IPSec, usato nelle VPN: offre una cifratura ed è retrocompatibile con IPv4/6. Nella modalità di trasporto viene instaurata una connessione fra i due host (è orientato alla connessione). Solo src e dest sapranno che stanno usando IPSec, mentre i nodi interni penseranno di usare IPv4/6. Serve quindi:

- Accordo crittografico
- Cifratura del carico utile
- Autenticazione della sorgente

Una volta stabilita la connessione, TCP e UDP sono cifrati e viaggiano nella rete.

Algoritmi di instradamento

- **Algoritmo di instradamento globale:** calcola il cammino con costo minimo avendo una conoscenza della completa rete (link-state algorithm)
- **Algoritmi di instradamento decentralizzato:** cammino calcolato in modo iterativo. Nessun nodo sa tutto di tutta la rete, lo scopre man mano (distance vector)

Si suddividono poi in:

- **Statici**: i cammini cambiano molto raramente
- **Dinamici:** si adattano al traffico e ai cambiamenti di topologia

Ulteriore suddivisione:

- **Load sensitive:** i costi degli archi sono dinamici
- **Load insensitive**

Algoritmo d'instradamento a stato del collegamento (Link State)

La topologia e tutti i costi della rete sono conosciuti. Lo si ottiene facendo inviare dei pacchetti tutti a tutti (OSPF). Una volta costruito l'input si usa **Dijkstra** (implementazione più sofisticata è quella che usa lo heap dell'amico Johnson (quello del sapone per sedere dei neonati)):

> *Supponiamo di avere un grafo con n vertici contraddistinti da numeri interi {1,2,...,n} e che 1 sia scelto come nodo di partenza. Il peso sull'arco che congiunge i nodi j e k è indicato con p(j,k). Ad ogni nodo, al termine dell'analisi, devono essere associate due etichette, f(i) che indica il peso totale del cammino (la somma dei pesi sugli archi percorsi per arrivare al nodo i-esimo) e J(i) che indica il nodo che precede i nel cammino minimo. Inoltre definiamo due insiemi S e T che contengono rispettivamente i nodi a cui sono già state assegnate le etichette e quelli ancora da scandire.*
>
> Incollato da <http://it.wikipedia.org/wiki/Algoritmo_di_Dijkstra>

Algoritmo d'instradamento con vettore distanza (Distance Vector)

Bellman ford distribuita. È un algoritmo (le iterazioni sono dovute a cambiamenti di qualche nodo in qualsiasi istante) asincrono e distribuito (ogni nodo calcola qualcosa). Ciascun nodo riceve parte dell'informazione da uno o più suoi vicini, calcola e restituisce i dati. I nodi stanno in ascolto finché non c'è un cambio nei vicini. Non richiede che tutti i nodi operino al passo con gli altri.
Percorso minimo da A a B viene calcolato con la formula di Bellman Ford:

$$d_x(y) = \min\{c(x,v) + d\}$$

Tramite questa formula si costruiscono le tabelle nell'intorno di un nodo x. Ciascun nodo x inizia con $D_x(y)$, una stima del costo del percorso a costo minimo da se stesso al nodo y, per tutti i nodi del grafo. Sia $D_x = [D_x(y) in N]$ il vettore distanza del nodo x, che è il vettore delle stime dei costi da x a tutti gli altri nodi y in N. Con l'algoritmo DV, ciascun nodo x mantiene i seguenti dati d'instradamento:

- Per ciascun vicino v, il costo c(x,v) da x a v
- Il vettore del nodo x, contenente la stima presso x del costo verso tutte le destinazioni y in N.
- I vettori distanza di ciascuno dei suoi vicini.

Quando un nodo riceve un nuovo vettore distanza de un vicino, riformula Bellman Ford ed invia ai propri vicini il risultato.
Se un cammino viene a costare meno, le buone notizie viaggiano rapidamente, infatti in 3 passi la rete ha aggiornato tutte le tabelle di instradamento.
Se un costo aumenta, pericolo: dopo un po' si continua a cambiare col proprio vicino. Ci si ferma quando si è raggiunti il costo maggiore dei due. Questo può portare ad un conteggio infinito (che bisogna rilevare).
Per ovviare al problema si ricorre all'**inversione avvelenata**: se z instrada tramite y per giungere alla destinazione x, allora z avvertirà y che la sua distanza verso x è infinita, anche se in realtà z sa che vale N. Continuerà a dire questa piccola bugia finché instrada verso x passando per y. Dato che y crede che z non abbia un percorso verso x, non tenterà mai di instradare verso x passando per z. Questa però risolve solo i cicli fra nodi adiacenti.
Vantaggi:

- Semplice da implementare
- Semplice da gestire
- Uso ampio

Svantaggi:

- Converge lentamente
- Va bene per reti piccoline
- Conteggio all'infinito (risolvibile col max hop)

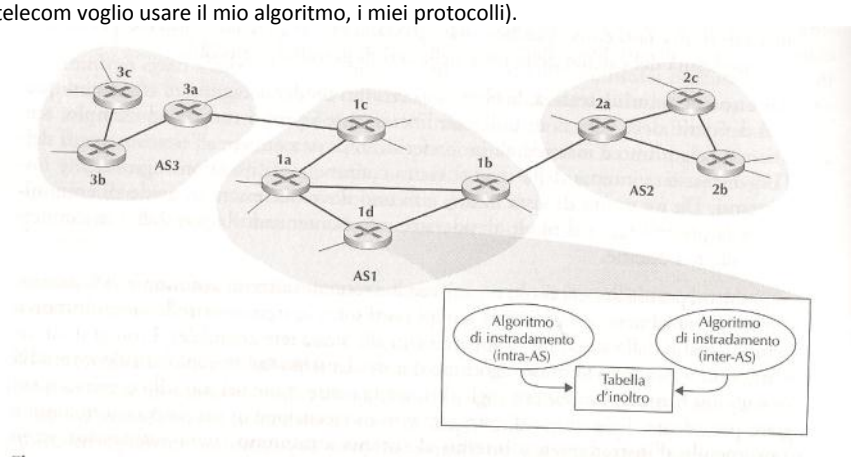
Confronto fra LS e DV

- **Complessità dei messaggi:**
  - LS richiede di costruire la rete in O(N*E), e quando qualcosa cambia bisogna informare tutti
  - DV lancia messaggi solo ai nodi adiacenti, ma può entrare in cicli infiniti
- **Velocità di converfenza:**
  - LS: O(N^2) + O(N*E) messaggi
  - DV può convergere lentamente o entrare in loop
- **Robustezza:**
  - LS è abbastanza robusto, se qualcosa va male manda in broadcast e si ricostruisce la rete, ma questi messaggi potrebbero essere alterati

- DV conunica alla rete errori con costo basso. Pericolo!: un calcolo errato si diffonde velocemente nella rete, la quale potrebbe intasare determinati nodi

Instradamento gerarchico

I router nella rete vengono organizzati in sistemi autonomi per problemi di scala e di autonomia amministrativa (io telecom voglio usare il mio algoritmo, i miei protocolli).



Questi sistemi autonomi devono essere connessi tra loro tramite i router gateway. Se un pacchetto è interno al sistema va tutto bene, ma se devo inoltrare ad un altro sistema? Lo mando ad uno dei gateway, quale? Usiamo l'ennesimo **prodocollo d'instradamento tra sistemi autonomi**.
I gateway devono conoscere quali sottoreti possono raggiungere. **Hot potato routing**: mi libero della patata bollente inviandola al gateway con costo minore, che possa raggiungere la destinazione.

Instradamento in internet

Link State

- Topology information is flooded within the routing domain
- Best end-to-end paths are computed locally at each router.
- Best end-to-end paths determine next-hops.
- Based on minimizing some notion of distance
- Works only if policy is shared and uniform
- Examples: OSPF

Distance Vector

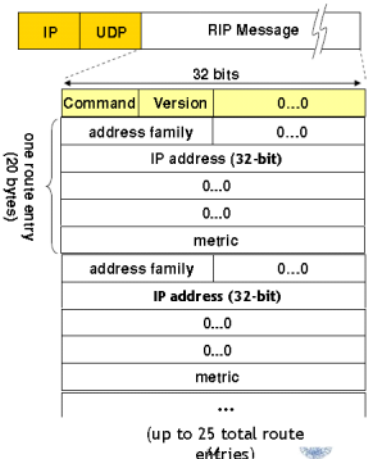
- Each router knows little about network topology
- Only best next-hops are chosen by each router for each destination network.
- Best end-to-end paths result from composition of all next-hop choices
- Does not require any notion of distance
- Does not require uniform policies at all routers
- Examples: RIP

In internet i **protocolli gateway interni** (quelli di routing dentro un SA) utilizzati sono RIP e OSPF.

Routing information protocol (RIP)

A vettore di distanza che opera in modo simile all'algoritmo DV, utilizza il conteggio degli hop come metrica di costo (per hop si intendono il numero di sottoreti (ma secondo me più che sottoreti sono archi…)attraversate). Il costo massimo di un percorso è limitato da 15 hop. I router adiacenti si scambiano ogni 30 secondi il DV utilizzando un messaggio di risposta RIP. Tale messaggio contiene fino a 25 sottoreti raggiungibili oltre alle informazioni interne al SA. Quindi ogni router ha una tabella d'instradamento ed una d'inoltro. Se non ricevo informazioni entro 180 secondi dal vicino, allora lo reputo irraggiungibile. Ogni 120 secondi fa il garbage collection. Por scambiarsi messaggi usiamo UDP sulla porta 520.

- Command: 1=request 2=response
  - Updates are replies whether asked for or not
  - Initializing node broadcasts request
  - Requests are replied to immediately
- Version: 1
- Address family: 2 for IP
- IP address: non-zero network portion, zero host portion
  - Identifies particular network
- Metric
  - Path distance from this router to network
  - Typically 1, so metric is hop count



locigno@disi.unitn.it

Open shortest path first (OPSF)

I tre principali criteri di progettazione del protocollo OSPF sono:

- Distinzione tra host e router
- Reti broadcast
- Suddivisione delle reti di grandi dimensioni

Generalmente utilizzato da ISP di livello superiore. Open perché è open source. È a Link state ed utilizza il **flooding** e Dijkstra. Si costruisce il grafo, il costo degli archi è dato dall'amministratore della rete. Se cambia la rete ci sono i messaggi broadcast, in più invia ogni 30 minuti lo stato dei collegamenti (anche se non sono cambiati). Questi annunci sono contenuti in messaggi OSPF trasportati da IP come protocollo di livello superiore.

Invia messaggi **HELLO** per vedere se il vicino è vivo.

Common header (type = 1, hello)		
Network mask		
Hello interval	*Options*	*Priority*
Dead interval		
Designated router		
Backup Designated router		
Neighbor		

Neighbor: lista di nodi adiacenti da cui ha ricevuto un messaggio di Hello negli ultimi dead interval secondi.

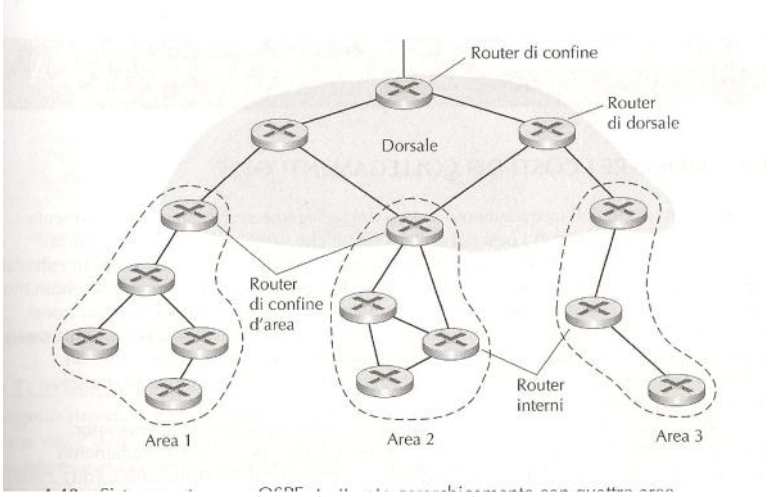
Utilizza anche il protocollo **EXHCANGE:**

- Sincronizzazione dei database link state (bring up adjacencies) tra due router che hanno appena verificato l'operatività bidirezionale del link che li connette
- Protocollo client-server
- Messaggi:
  - Database Description Packets
  - Link State Request
  - Link State Update (mandato col flooding)

Vantaggi:

- **Sicurezza:** Gli scambi tra router OSPF possono essere autenticati tramite MD5 + chiavi di cifratura. Soli i pacchetti fidati vengono presi in considerazione.
- **Percorsi con lo stesso costo:** vengono mantenuti + percorsi con lo stesso percorso minimo in modo da non caricarne solo uno
- **Supporto unicast e multicast**
- **Supporto alle gerarchie di instradamento:**possibilità di strutturare SA in modo gerarchico

Un sistema autonomo OSPF può essere configurato in aree che eseguono diversi algoritmi d'instradamento OSPF.



In ogni area i router di confine si fanno carico dell'instradamento dei pacchetti indirizzati all'esterno. Un'area di un SA OSPF è configurata per essere l'area di dorsale il cui ruolo principale è d'instradare il traffico tra le altre aree del sistema autonomo. La dorsale contiene tutti i router di confine. Gerarchia ei router:

- **Router interni:** interni, non di dorsale
- **Router di confine d'area:** fanno parte sia di un'area gerarchica che di dorsale
- **Router di dorsale**: interni, di dorsale
- **Router di confine**: scambiano informazioni con altri sistemi autonomi

Instradamento tra sistemi autonomi

Viene utilizzato il **border gateway protocol** versione 4 (**BGP4**):

- Ottenere informazioni sulla raggiungibilità delle sottoreti da parte dei sistemi confinanti
- Propagare le informazioni di raggiungibilità tra tutti i router interni ad un sistema autonomo
- Determinare percorsi buoni verso le sottoreti sulla base delle informazioni di raggiungibilità e delle politiche del sistema autonomo

Fondamenti di BGP

Utilizza connessioni semi permanenti TCP fra SA (eBGP) e interne ad ogni SA (iBGP). La connessione TCP con tutti i dati che vengono inviati è detta sessione BGP (esterna, interna). Non sempre i collegamenti BGP corrispondono a collegamenti fisici. In BGP le destinazioni non sono host ma **prefissi CIDR** che rappresentano una sottorete o una collezione di sottoreti. Nella figura AS3 contiene 3 sottoreti. AS3 quindi computa i prefissi delle sue sottoreti e manda un unico prefisso ad AS1, ma potrebbe anche annunciare un prefisso specifico della sottorete.
Nella sessione eBGP fra i gateway 3a e 1c AS3 e AS1 si inviano la lista dei prefissi raggiungibili tra loro tramite i gateway. Quando un gateway vuole inviare tramite eBGP dai prefissi utilizza le proprie sessioni iBGP.

Attributi del percorso e rotte BGP

Un SA viene identificato dal suo numero di sistema (ASN) globalmente univoco. Non tutti gli SA hanno ASN, quelli che trasportano solo traffico di cui sono origine o destinazione non lo hanno. ASN è assegnato da ICANN (come per gli indirizzi IP). Oltre al prefisso in usa sessione GBC ci sono anche gli **attributi BGP**:

- **AS-PATH:** indica per quali SA è passato il prefisso
- **NEXT-HOP**

Selezione dei percorsi BGP

Ad ogni rotta nei messaggi iBGP viene associato una preferenza locale. La imposta direttamente uno dei router o la sceglie per lui qualche altro router. Tra rotte con lo stesso valore di preferenza locale, si sceglie quella con AS-PATH più breve. Tra quelle che hanno stessa preferenza, stesso AS-PATH minimo si sceglie quella col NEXT-HOP più vicino.

Instradamento broadcast

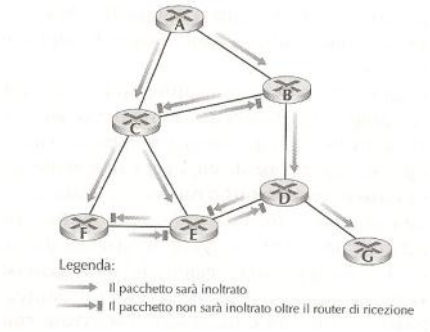
Unicast a N vie: invia il pacchetto a tutte le destinazioni. Inefficiente. Non è detto che gli indirizzi di tutti siano conosciuti dal mittente (bisognerebbe ottenerli, na sega). È contro logica usare un meccanismo unicast per fare un broadcast. Si scelgono quindi altre modalità.

Flooding incontrollato

Invio il pacchetto ai miei vicini, che lo duplicano e lo inoltrano ai loro vicini ecc. Se il grafo è connesso dovrebbe fuzionare. BALLE: se c'è un ciclo scoppia tutto. Altro difettone (tempesta di broadcast): se creo più pacchetti per la stessa destinazione anche quella farà così sui vicini, e alla fine si intasa tutto.

Flooding controllato

- **Flooding controllato con numeri di sequenza:** l'origine pone un proprio identificatore ed un numero di sequenza di broadcast nei pacchetti prima di inviarli. Ogni volta che un nodo riceve un pacchetto si segna l'identificativo da qualche parte, se l'identificativo è già stato visto scarta il pacchetto, altrimenti lo duplica ed inoltra. Difetto: non riesce ad eliminare l'invio di pacchetti ridondanti:



- **Inoltro su percorso inverso** (**RPF** reverse path flooding): il nodo inoltra a tutti i vicini, tranne quello da cui ha ricevuto il pacchetto, solo se il pacchetto viene da un percorso minimo rispetto alla sorgente. Difetto: non riesce ad eliminare l'invio di pacchetti ridondanti.
- **Broadcast con spanning tree:** invia pacchetti secondo un albero di copertura. Elimina così la ridondanza dei pacchetti.

Multicast

Difficile, in unicast si specifica un solo indirizzo, in broadcast anche (perché nessuno escluso deve ricevere). I pacchetti vengono indirizzati utilizzando l'**indirezionabilità degli indirizzi**. Si usa un identificatore singolo per un gruppo di indirizzi. I gruppi sono creati secondo IGMP (in internet).o

Layer 2 - Data link

00:09

Obiettivo principale: fornire al livello di rete di due macchine adiacenti un canale di comunicazione il più possibile affidabile.
Definisce il formato dei pacchetti scambiati fra due nodi agli estremi del collegamento, non si occupa di instradamento, utilizza il canale una volta trovato un percorso. Le unità di dati scambiati a livello di collegamento (data link, DL) sono detti **frame**. Si occupa di trasportare i datagrammi del livello di rete. Il protocollo utilizzato all'inizio del cammino non è necessariamente lo stesso per tutto il percorso. (ethernet -> WAN->wlan). In pratica il DL è molto simile a quello di trasporto, solo lavora fra nodi adiacenti a livello più basso.

Servizi offerti

- **Servizio connectionless senza acknowledge**: non viene attivata nessuna connessione, invio delle trame senza attendere alcun feedback dalla destinazione. Se una trama viene persa non ci sono tentativi per recuperarla, il compito viene lasciato ai livelli superiori. La maggior parte delle LAN utilizzano questa tipologia di servizio
- **Servizio connectionless con acknowledge**: non viene attivata nessuna connessione, ogni trama inviata viene "riscontrata" in modo individuale
- **Servizio connection-oriented con acknowledge**: viene attivata una connessione e, al termine del trasferimento, essa viene abbattuta. Ogni trama inviata viene "riscontrata" in modo individuale
- **Half-duplex, full-duplex**

Principali funzioni

- **Framing**: siccome il livello 1 opera solo sui bit, bisogna rendere riconoscibile una trama dall'altra . I frame sono costituiti da un campo dati nel quale è inserito il datagramma e da campi di intestazione e trailer. Esistono diverse tecniche per implementare il framing:
    - Inserire intervalli temporali fra trame consecutive:
        - Problema: per natura intrinseca le reti di telecomunicazione non danno garanzie sul rispetto delle caratteristiche temporali delle informazioni trasmesse
        - Gli intervalli inseriti potrebbero essere espansi o ridotti generando problemi di ricezione
    - Marcare inizio e termine di ogni trama:
        - **Character count**: un campo nell'header indica il numero di caratteri che ci sono nel frame. Attenzione, se un carattere si perde per strada, rovina tutto quello che c'è dopo.
        - **Character stuffing**: ogni trama inizia e termina con una sequenza scelta:
            - DLE(data link escape) + STX (Start of Text)
            - DLE + ETX (end of text)
        - Se nella trasmissione troviamo una di queste sequenze note la sorgente duplica il DLE (character stuffing). Svantaggio: legata al modulo base dei caratteri ad 8 bit e alla codifica ASCII
        - Starting and ending flags (**bit stuffing**): ogni trama inizia e termina con una sequenza nota di bit (01111110). Il trasmittente quando trova 5 uni consecutivi ci sbatte li uno 0.
        - **Physical layer coding violation**s: 1->10, 0->01, quindi 00 e 11 non sono mai usati, e si possono usare come delimitatori
- **Accesso al collegamento**: controlla l'accesso al mezzo (MAC, medium access control), specifica le regole con cui immettere i frame nel collegamento.
- **Consegna affidabile**: gestiscono il trasporto dei datagrammi senza alcun a perdita di dati in maniera simile al livello di trasporto (ack ecc). Lo scopo è correggere localmente gli errori trasmissivi prima di ingigantirli quando attraverseranno la rete.
- **Controllo di flusso**: i nodi agli estremi del collegamento presentano una limitata capacità di buffering di conseguenza il destinatario potrebbe ricevere frame a un ritmo più elevato di quello con cui riesce a processarli. Serve quindi a garantire che il mittente non saturi il destinatario. È basato su feedback inviati alla sorgente dalla destinazione indicando:
    - Bloccare la trasmissione fino al comando successivo

    - La quantità di informazione che la destinazione è ancora in grado di gestire

  I feedback possono essere:
    - Nei servizi con riscontro, gli ack stessi
    - Nei servizi senza riscontro, dei pacchetti appositi
- **Rilevazione e correzione degli errori.**

Dov'è implementato il livello di collegamento?

Nelle schede di rete: NIC (network interface card). Di solito è un chip unico che fa tutto il lavoro. Tempo fa le schede di rete nei PC erano separate. Possiamo individuare due parti principali del livello 2:

Anche se in linea di principio il livello MAC gestisce l'accesso al mezzo e il livello "high" gestisce le altre funzionalità, nella pratica il livello MAC gestisce anche il framing e il controllo di errore, mentre il livello 2 "high" si occupa del controllo di flusso. Nello stack TCP/IP ove il livello 2 non fa controllo di flusso, il livello 2 "high" è completamente assente o, se c'è, non svolge nessuna funzione.

Tecniche di rilevazione e correzione degli errori

Oltre ad inviare i dati normali, si inviano anche dei bit EDC (error detection and correction).

<figura a pagina 392>

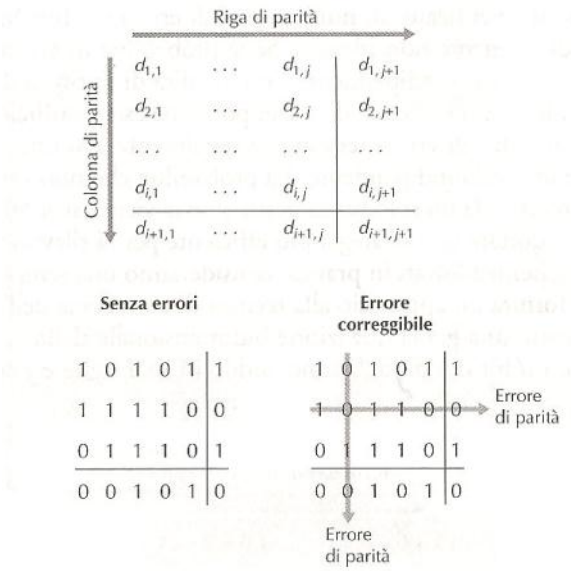
Anche con le tecniche di rilevazione degli errori possono comunque verificarsi degli errori non rilevati. Ci sono 3 tecniche:

- Controllo di parità (parity check)
- Somma di controllo (di solito usato nel livello di trasporto)
- Controllo a ridondanza ciclica (CRC: cyclic redundancy check)

Controllo di parità

Supponiamo che le informazioni da inviare "D" siano composte da d bit. In uno schema di parità pari, il mittente include un bit addizionale e sceglie il suo valore in modo da rendere pari il numero totale dei bit a 1 nei d+1 bit trasmessi.

Quindi per controllare, il ricevente deve solo contare il numero di bit che sono ad 1.
NON FUNZIONA BENE: se si è verificato un numero pari di errori nei bit? Abbiamo un errore non rilevato.
Essendo che la probabilità di errore in un bit è molto bassa, quella su più bit è ancora più bassa, potrebbe bastare un solo bit di parità. Tuttavia non siamo ancora sicuri che effettivamente l'errore non c'è.
Usiamo quindi la parità bidimensionale



Con questa tecnica non solo ci accorgiamo dell'errore, ma possiamo pure correggerlo.

Somma di controllo

Checksum. Sommiamo i bit dati come interi da k bits, usiamo il risultato per rilevare gli errori. In internet la checksum funziona: dati son interi di 16 bit sommati, il complemento a 1 di questi bits corrisponde alla checksum, che viene inviata assieme ai pacchetti. In IP la checksum è fatta solo sugli header, dato che TCP e UDP ne hanno una propria. TCP e UDP rilevano gli errori via software, mentre DL hardware (velocemente, semplicemente). Sono piuttosto limitate rispetto CRC.

CRC

Largamente utilizzata. Codici di controllo a ridondanza ciclica. I CRC sono detti anche **polinomiali** perché possiamo immaginare i bit come coefficienti di un polinomio.
"d" bit sono i dati da trasmettere (D). Sorgente e destinazione sono d'accordo su una stringa di r+1 bit chiamata generator (G). Il più significativo di G è a 1. Dato un blocco di dati D il mittente sceglie r bit ® eli unisce a D in modo da ottenere una stringa di d+r bit che interpretata in binario sia esattamente divisibile per G.
(r+d)/G ha resto 0: errore. Siccome tutto è fatto in modulo 2 senza riporto, + e - equivalgono ad uno XOR.
Quindi r+d equivale a fare: $D \cdot 2^r \; XOR \; R$
Bisogna scegliere R in modo che R+D sia divisibile per G.

Protocolli di accesso multiplo

Problema di decisione dell'utilizzo del mezzo:

- Concedi a ciascuno la possibilità di parlare
- Non parlare finchè non sei interrogato
- Non monopolizzare la conversazione
- Alza la mano se devi porre una domanda
- Non interrompere se qualcuno sta parlando
- Non addormentarti quando qualcuno parla

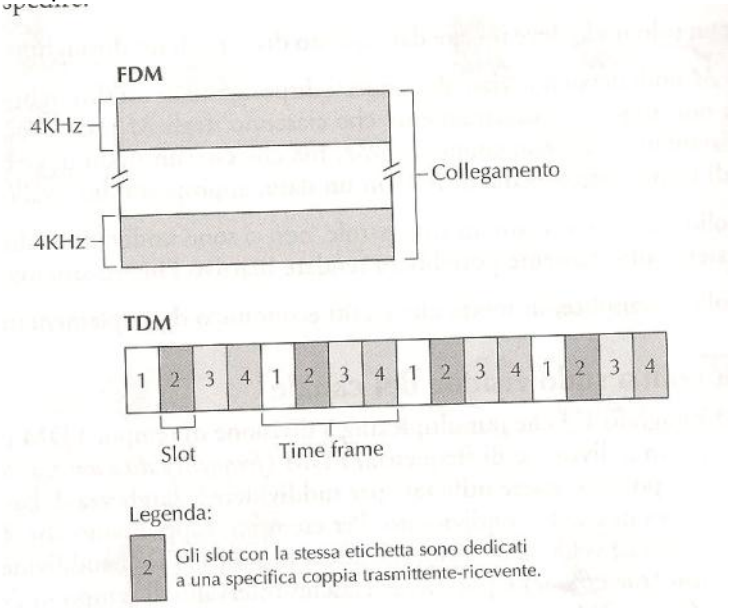
I **protocolli di accesso multiplo** fissano le modalità con cui i nodi regolano le loro comunicazioni sul canale condiviso. Siccome tutti possono trasmettere frame è facile avere delle collisioni: nessuno dei sender riuscirà a fare quello che vuole, tutto va perso. Categorie di protocolli di accesso multiplo:

- Allocazione statica (a suddivisione del canale):
  - TDMA
  - FDMA
  - CDMA
- Allocazione dinamica(ad accesso casuale):
  - Ad accesso casuale (a contesa): i più utilizzati.
    In generale tutti rispettano (R è la velocità del canale trasmissivo):
    - Quando un solo nodo deve inviare dati dispone di un throughput pari a R bps
    - Quando M nodi devono inviare hanno throughput pari a R/M (intesa come velocità media)

        - Protocollo decentralizzato
        - Protocollo semplice, economico da implementare
    - A rotazione (a turno): gestione complessa

Protocolli ad allocazione statica

Usano TDM oppure FDM (vedi livello fisico). Vanno bene con pochi utenti, con molto poco carico costante nel tempo.



- **Caso TDMA** (time division multiple acces): ogni slot di tempo è assegnato a uno degli N nodi. Ogni volta che un nodo deve inviare, trasmette i bit del pacchetto durante lo slot di tempo assegnatoli a rotazione nel frame TDM. Di solito le dimensioni degli slot sono tarati per poter trasmettere almeno un pacchetto. Inefficiente quando ci sono pochi broadcast
- **Caso FDMA**: assegna un range di frequenza ad ogni nodo. Difetto: larghezza di banda del canale non è illimitata
- **Caso CDMA** (code division …): ogni nodo ha un proprio codice, che utilizza per codificare i dati inviati. Se i codici sono scelti bene si può anche trasmettere simultaneamente . È molto usato nei canali wireless.

Protocolli ad accesso a contesa

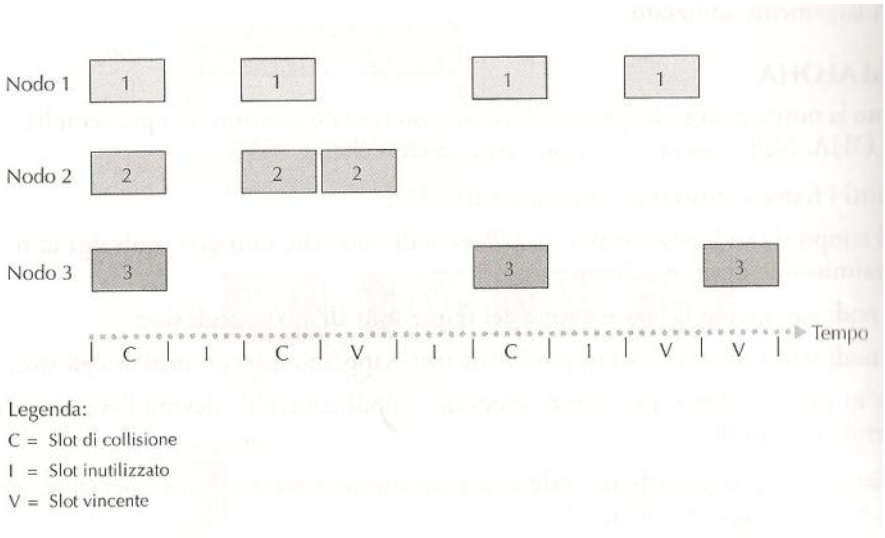
Tutti trasmettono alla massima velocità. Quando c'è una collisione, si ritrasmette il frame finché non arriva a destinazione senza collisioni. La ritrasmissione non è immediata ma bisogna aspettare un random delay. Scegliere bene questo delay può far causare meno collisioni.
Definiamo **periodo di vulnerabilità** il lasso di tempo in cui ci potrebbero essere delle collisioni.

Slotted ALOHA

Assumiamo che:

- Tutti i frame siano lunghi L bit
- Il tempo sia suddiviso in slot di L/R secondi (slot = tempo trasmissione pacchetto)
- I nodi comincino la trasmissione dei frame solo all'inizio di ogni slot
- Se in uno slot due o più frame collidono, i nodi coinvolti rilevino l'evento prima del termine dello slot

Indichiamo con p la probabilità di successo. ALOHA fa queste cose:



- Quando un nodo ha un nuovo frame da spedire, attende fino all'inizio dello slot successivo e poi trasmette l'intero frame
- Se non si verifica una collisione, l'operazione ha avuto successo, quindi non occorre effettuare una ritrasmissione e possiamo inviare un nuovo frame
- Se si verifica una collisione il nodo la rileva prima del termine dello slot, e ritrasmette con probabilità p il suo frame durante gli slot successivi, finché non trasmette con successo

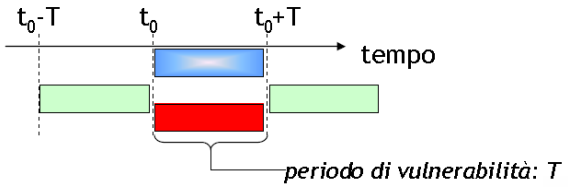
Con proprietà p significa: lancia una moneta, se esce testa trasmetti, se esce croce aspetta e rilancia la moneta.
Vantaggi:

- Posso trasmettere al massimo finché parlo solo io
- Decentralizzato: ognuno rileva le proprie collisioni
- Semplicissimo

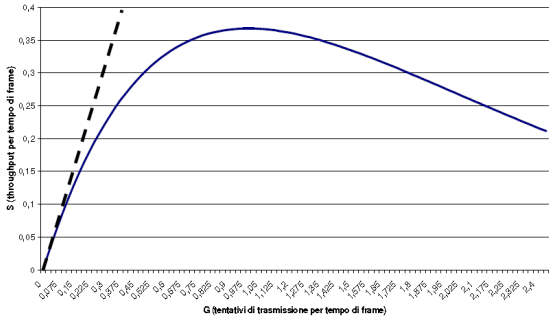
Problemi:

- Quando c'è una collisione, ci saranno degli slot sprecati
- Potrebbero esserci degli slot vuoti dovuti alla probabilità

Il periodo di vulnerabilità in questo caso è il tempo di una trama.



L'efficienza di un protocollo slotted è data da slot vincenti/nodi attivi che spediscono. Se N nodi trasmettono assieme abbiamo efficienza dello slot pari a $Np(1-p)^{N-1}$, quindi bisogna pesare p in modo che massimizzi la formula: 1/e (viene dal limite che tende all'infinito, robe della Barozzi che nessuno si ricorda più). Quindi l'effettiva velocità di trasmissione non è R ma 0,37R. Il 37% degli slot viaggia vuoto ed il 26% ha delle collisioni. Quindi non va bene quando tanti trasmettono assieme, povero amministratore della rete, che è stato millantato da qualche simpaticone Hawaiano. NB: un'attesa deterministica prima della ritrasmissione porterebbe ad attese infinite.



Pure ALOHA

Privo di slot, decentralizzato. Appena si vuole inviare un frame, il nodo lo strasmette integralmente sul canale broadcast. Se un frame va in collisione il nodo lo ritrasmette con probabilità p, immediatamente.

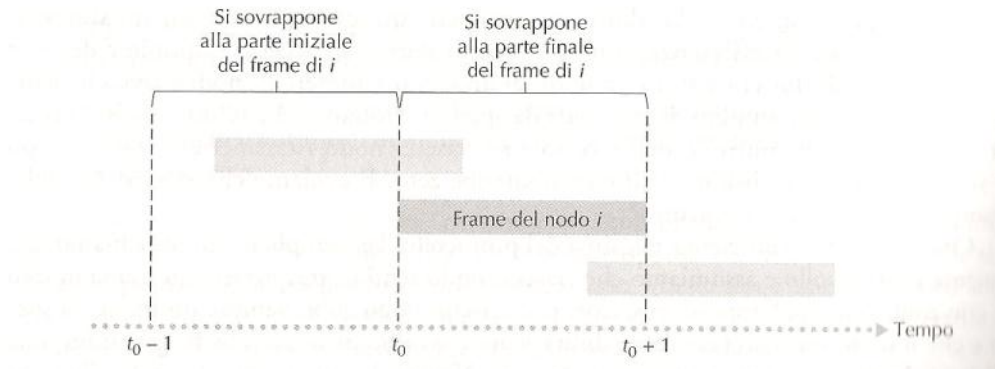
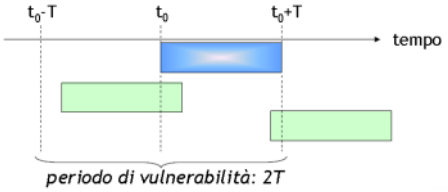
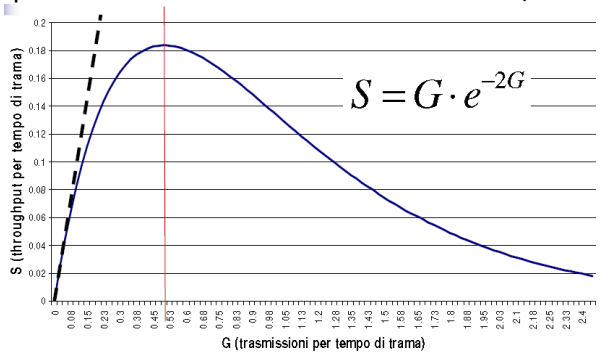


Figura 5.12 Interferenza delle trasmissioni in ALOHA puro.

Il periodo di vulnerabilità è il doppio della durata di una trama: detto T il tempo di trama e t0 l'inizio della trasmissione da parte di una sorgente, il periodo di vulnerabilità è pari al doppio del tempo di trama nel momento in cui inizia a trasmettere (t0), nessuna altra sorgente deve aver iniziato la trasmissione dopo l'istante di tempo t0 -T e nessuna altra sorgente deve iniziare la trasmissione fino a t0 +T



Quando un nodo si accorge che qualcuno sta interferendo, non fa nulla, continua a trasmettere, bovinamente. Ogni nodo può trasmettere in ogni stante con probabilità p, quindi ci possono essere collisioni in qualsiasi istante. La probabilità che un nodo non trasmetta mentre sta trasmettendo qualcun altro è $p(1-p)^{2(N-1)}$, quindi l'efficienza è metà di aloha slotted: 1/2e.



CSMA

Carrier sense multiple acces. Se gli aloha sono due maleducati che non si preoccupano che qualcuno potrebbe star parlando, mentre loro vogliono parlare, **CSMA è più educato**:

- **Ascolta prima di parlare**: carrier sensing (o rilevazione della portante), se qualcuno sta parlando sta zitto ed aspetta un po' prima di riprovare a parlare, magari hanno finito gli altri.
- **Se qualcuno comincia a parlare assieme a lui, smette di parlare**: collision detection, quando trasmetto sto anche in ascolto, se qualche maleducato parla mentre parlo io, mi fermo (perché sono molto educato), ci penso un po' e decido quando è il momento giusto di parlare.
- **Se il canale è occupato:**
  - Non persistent: rimanda la ritrasmissione ad un nuovo istante casuale
  - Persistent: appena si libera ritrasmetto

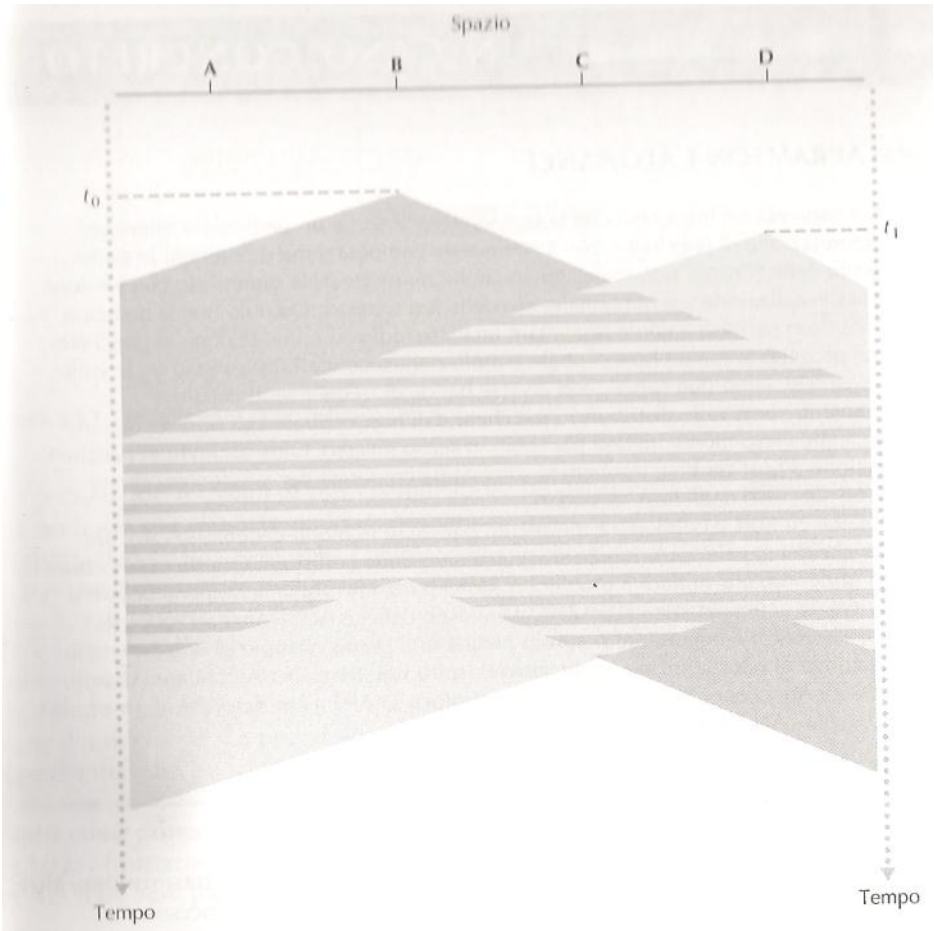
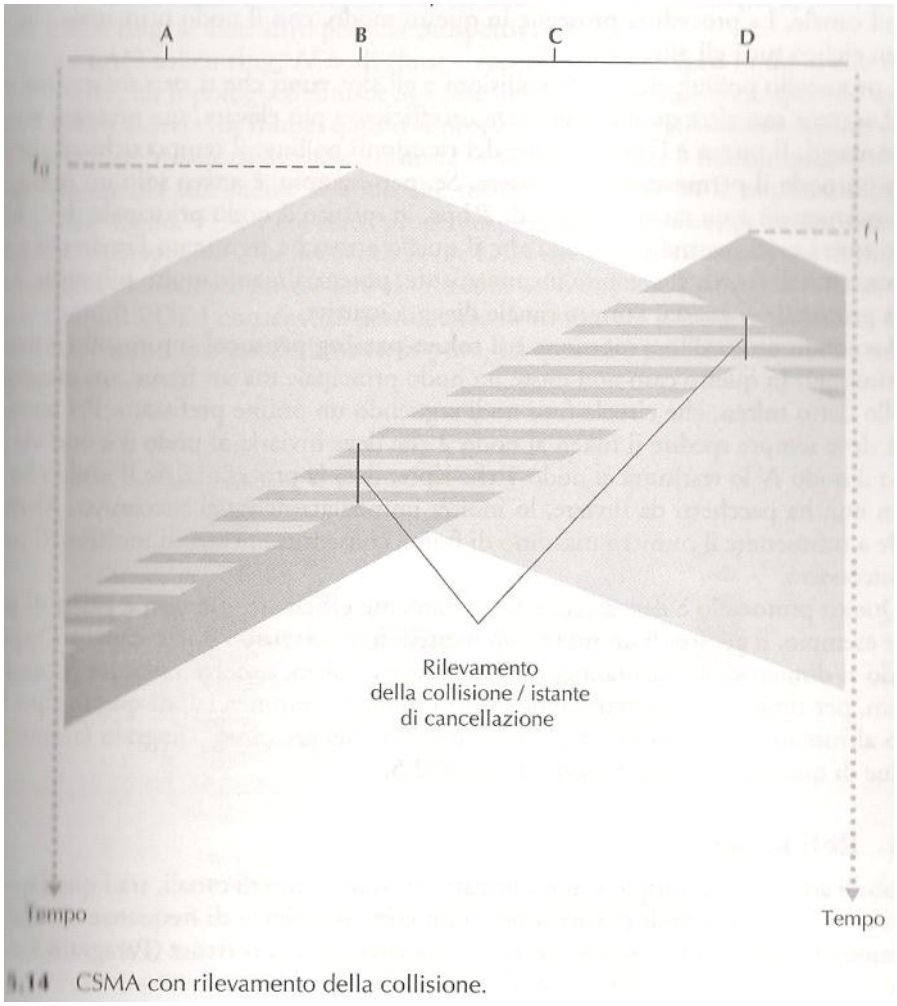


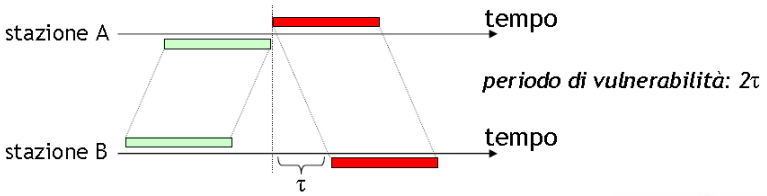
Figura 5.13 Diagramma spazio tempo di due nodi CSMA con trasmissioni in collisione.



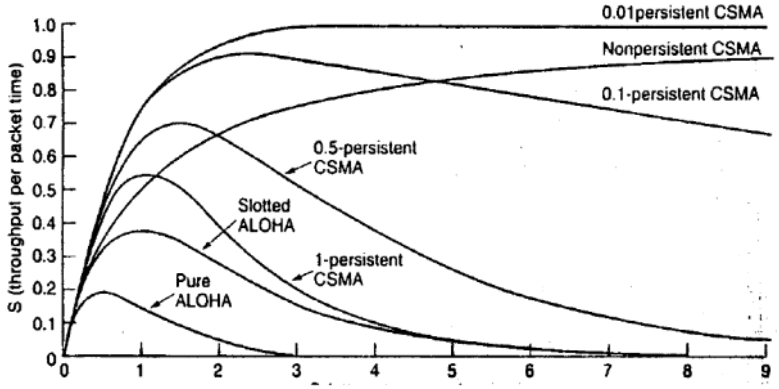
5.14 CSMA con rilevamento della collisione.

Perché se tutti ascoltano prima di parlare si verificano collisioni? Eccola la risposta: io trasmetto verso il Ghana perché ho il canale libero, e voglio far saltare in aria la macchina di un politico. Un egiziano decide di vendere della cocaina a qualche albanese, guarda se il canale è libero. Lo trova libero perché io che abito in Norvegia ho spedito qualche istante prima, ma il segnale non è arrivato ancora dallo spacciatore. Quindi si verificano collisioni perché le propagazioni dei segnali non avvengono in tempo nullo.

- **CSMA p-persistent**: il tempo viene suddiviso in intervalli la cui lunghezza è uguale al periodo di vulnerabilità. Trasmetto con probabilità p, se si è deciso di trasmettere vedo se è occupato (in quel caso aspetto un po' e ritrasmetto con probabilità p), se è libero e c'è una collisione aspetto un tempo casuale e provo a ritrasmettere con probabilità p. Se si è deciso di non trasmettere (p-1) aspetto un intervallo e riprovo.
  Il periodo di vulnerabilità è quindi dato dal tempo di propagazione.



- **CSMA/CD** (CSMA with collision detection): se la stazione che sta trasmettendo rileva la collisione, interrompe immediatamente. In questo modo, una volta rilevata collisione, non si spreca tempo a trasmettere trame già corrotte. Inoltre, per far sentire a tutte le stazioni che vi è stata collisione, si trasmette una particolare sequenza, detta di jamming.



(fonte: A. Tanenbaum, Computer Networks)

Protocolli a rotazione

Le proprietà auspicabili di un canale ad accesso multiplo sono:

- Se è solo lui a trasmettere dovrebbe avere un throughput pari a R (la capacità del canale)
- Quando sono attivi M nodi lo dovrebbe avere a R/M

Il maleducato ALOHA e l'educato CSMA possiedono la prima proprietà, ma non la seconda. Per questo, i cervelli si sono inventati i protocolli a rotazione (token run protocol).

Protocollo di polling: uno dei nodi designato come principale, sonda "a turno" gli altri: dice ad esempio al nodo 1 che può trasmettere fino a N frame. Quando il nodo 1 ha finito, avvisa il due che può inviare N frame.
Vantaggi:

- Elimina slot vuoti, collisioni

Svantaggi:

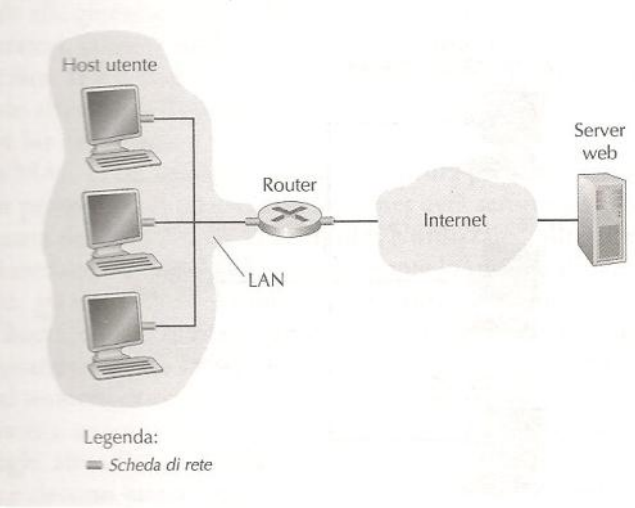
- Ritardo di polling: avvisare qualcuno che può fare qualcosa richiede del tempo (quando 1 ha finito di trasmettere, il boss sonda, e avverte il prossimo che può trasmettere)
- Se il boss si guasta, tutto il canale diventa inattivo

Token passing protocol: non esiste un nodo principale, ma un frame (il token) che circola fra i nodi in un ordine prefissato. Se un nodo riceve il token e non ha pacchetti da inviare, lo invia al successivo, altrimenti trasmette un numero massimo di frame e lo inoltra al successivo. È decentralizzato ed altamente efficiente, ma svantaggi:

- Il guasto di un nodo può mettere a $\frac{\pi}{2}$ l'intero canale
- Se il token per qualche bislacco motivo va perso, ci vogliono delle procedure per rimetterlo in circolazione

Reti locali

I protocolli qui sopra sono enormemente sfruttarti nelle reti locali.



- Velocità elevata
- Ristrette geograficamente

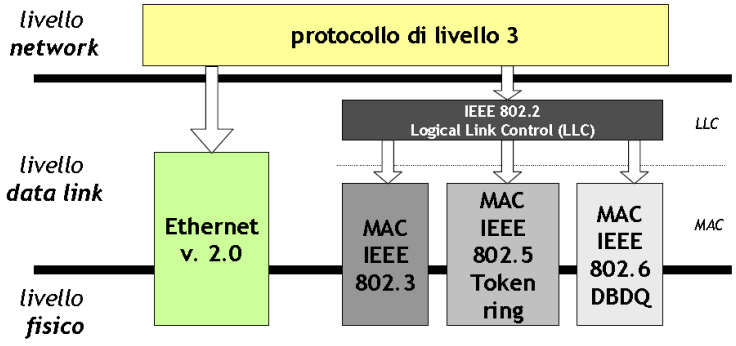
In una lan a token ring N nodi sono collegati in un anello tramite connessioni dirette. La topologia definisce l'ordine del passaggio del token. Quando qualcuno ha il token ed invia frame, lo fa propagandoli nell'intero anello, quindi quando giunge al destinatario, deve essere rimosso dall'anello, quindi non è un canale di broadcast puro.

Principalmente per LAN non di backbone si usa comunque:

- Ethernet (CSMA/CD 10Mbps)
- IEEE 802.3 (CSMA/CD 1-10MbpS)
- PPP

Mentre per le backbone:

- Frame relay
- ATM
- SDH
- Many others



Indirizzi MAC

Indirizzi a livello di collegamento (media acces control) che individuano univocamente a livello mondiale una scheda di rete, grazie alla sovrintendenza di IEEE: quando una società fa degli adattatori deve comperare di blocchi di indirizzi (2^{48}In totale). Non c'è alcuna gerarchia, non cambia mai.

IP: indirizzo di residenza di una persona. MAC: codice fiscale.

Protocollo di risoluzione degli indirizzi ARP

Converte da indirizzi di rete in indirizzi di collegamento. Per trasmettere un datagramma, il nodo trasmittente deve fornire al suo adattatore non solo l'indirizzo IP, ma anche l'indirizzo MAC. È per molti aspetti simile al DNS. Nella ram dei nodi vi è una **tabella ARP** che contiene le corrispondenze IP - MAC.

Indirizzo IP	Indirizzo MAC	TTL
222.222.222.221	88-B2-2F-54-1A-0F	13:45:00

222.222.222.223	5C-66-AB-90-75-B1	13:52:00

Il TTL (time to live) indica quando bisognerà eliminare la voce dalla tabella. La tabella non contiene per forza una linea per ogni nodo nella sottorete, alcuni possono essere cancellati/mai stati scoperti. TTL tipico: 20 min. Quando un nodo vuole inviare qualcosa a qualcuno di cui ha la corrispondenza nella tabella ARP è tutto semplice. Quando invece non ha la corrispondenza crea un **pacchetto ARP** che possiede molti campi, fra cui indirizzi MAC/IP mittente e di chi l'ha ricevuto. I pacchetti ARP di richiesta e di risposta hanno lo stesso formato. I pacchetti ARP sono incapsulati in frame a livello di collegamento. [Cit libro:] il protocollo ARP è simile ad una persona che si alza in mezzo all'ufficio ed inizia a gridare: "qual è il codice fiscale della persona che abita a <indirizzo>. L'unico nodo che ha l'IP corrispondente risponderà, la tabella del mittente verrà aggiornata. È plug-and-play perché la tabella non è fatta a mano, se un nodo si scollega prima o poi verrà cancellato da tutti.

Come inviare un datagramma ad un nodo esterno alla sottorete

Mente un host ha un solo indirizzo IP, ed una tabella ARP, i router dispongono di un indirizzo IP per ogni interfaccia, e quindi anche di diverse tabelle ARP.

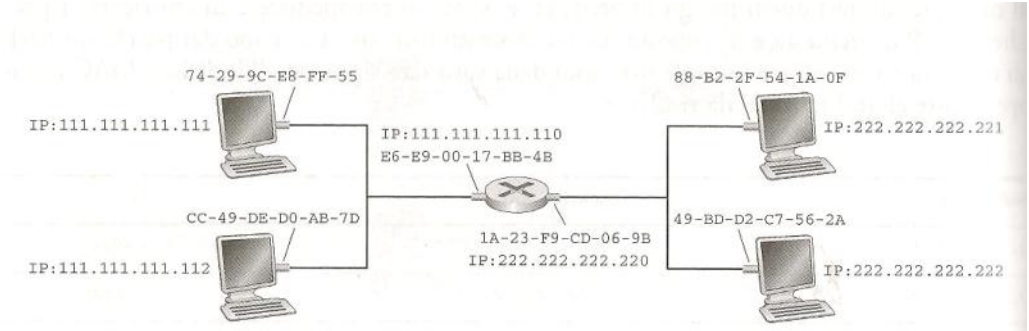


Figura 5.19 Due sottoreti connesse da un router.

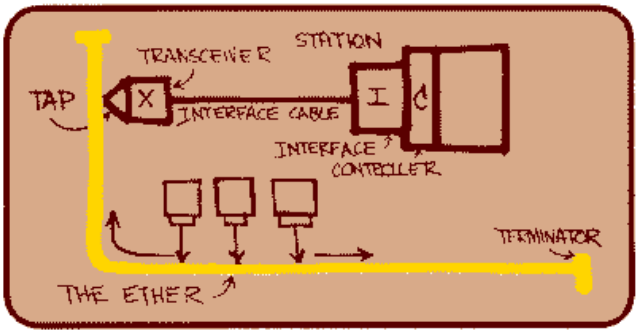
Se non è sulla sottorete invio al gateway, si, ma mi serve comunque il MAC. Quindi si mette l'indirizzo MAC del router, lui si accorge di non essere il vero destinatario e provvede a mandar in giro il pacchetto.

Ethernet

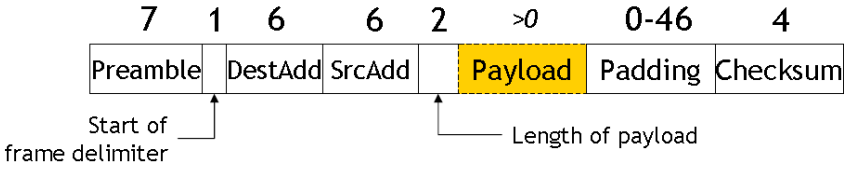
Detiene il mercato dominante nelle LAN cablate rappresentando nelle reti locali ciò che è internet per l'interconnessione globale delle reti.
È molto veloce e si è diffusa subito; FDDI, token ring e ATM sono molto più complesse e costose, ha il tasso trasmissivo più alto delle altre. La sua grande diffusione ha fatto si che i suoi adattatori siano ampiamente reperibili, ed economicamente pro.


Robert Metcalfe, inventore di Ethernet

La prima ethernet era una a bus, quindi broadcast. Successivamente si è diffusa quella a stella basata su un **hub**. Lo hub è un dispositivo a livello fisico che agisce sui singoli bit piuttosto che sui frame: quando arriva un bit, hub lo rigenera, amplificandolo e lo rimanda su tutte le interfacce: se si inviano frame contemporaneamente, collisone, perso i frame. Successivamente si è passati dallo hub allo **switch**: un commutatore di pacchetti store-and-forward privo di collisioni, lavora a livello 2 (il router a livello 3).



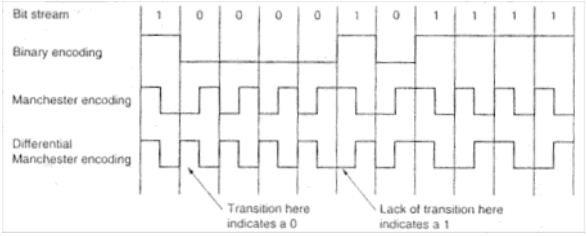
Struttura dei pacchetti ethernet



- **Campo dati:** contiene i pacchetti dei livelli superiori (datagramma IP). L'MTU (unità massima trasferita) è 1500 byte, quindi se il pacchetto IP è più grande bisogna frammentarlo. Se è più piccolo della dimensione minima invece, deve essere riempito (poi vengono rimossi usando il campo lunghezza)
- **Indirizzo di destinazione/sorgente**: MAC d/s
- **Campo tipo**: che tipo di protocollo di rete sta trasportando, in modo che il ricevitore possa fare il de multiplexing dei protocolli
- **CRC**
- **Preambolo**: composto da 8 bytes, i primi sette servono a sincronizzare il clock del trasmettitore con quello del ricevente, l'ultimo byte comunica che ora arriverà il frame vero e proprio.

Ethernet utilizza la trasmissione in banda base, ossia l'adattatore inserisce il segnale direttamente nel canale broadcast, non vi è modulazione.

Molte tecnologie ethernet (10BaseT ecc) utilizzano anche la **codifica di Manchester**.



Durante la ricezione di ciascun bit si verifica una transazione, da 1 a 0 è associato 1, 0 viceversa. Questo serve per sincronizzare sorgente e destinazione. È comunque un'operazione del livello fisico.

Servizio senza connessione non affidabile

Come il livello di rete verso quello di trasporto, il livello di collegamento offre a quello di rete un servizio senza connessione (non ci sono handshake ecc). Non è nemmeno affidabile, quando il CRC non combacia, non dice "ah! l'è sbaià!" ma semplicemente lo butta (se non può correggerlo). Questo fatto aiuta ethernet ed essere semplice ed economica, a discapito di possibili lacune nel flusso di datagrammi.
La rilevazione di errori, frame persi e balle varie, è destinata al livello di trasporto (se vuole (TCP!=UDP)). In realtà quindi ethernet riceve solo ordini dall'alto.

CSMA/CD: protocollo di accesso multiplo di ethernet

Il **collision domain** è quella porzione di rete Ethernet in cui, se due stazioni trasmettono simultaneamente, le due trame collidono. Spezzoni di rete connessi da repeater sono nello stesso collision domain. Spezzoni di rete connessi da dispositivi di tipo store and forward (bridge, switch o router) sono in collision domain diversi.
Con il termine **diametro di un collision domain** si indica la distanza massima tra ogni possibile coppia di stazioni. Il diametro massimo di un collision domain a 10Mbit/s è di 2800m e dipende da:

- lunghezza massima dei cavi (attenuazione del segnale che induce uso di repeater, con ritardo aggiuntivo)
- ritardo di propagazione (round trip delay)

Quando abbiamo un inutile HUB, abbiamo le collisioni, ethernet le risolve con il CSMA/CD con caratteristiche:

- Non può trasmettere un frame quando qualcun altro lo sta facendo
- Annulla la trasmissione se qualcuno si mette in mezzo
- Prima di ritrasmettere aspetta un tempo arbitrario
- Slot time = 512 bit time (51.2 µs): unità base di attesa prima di una ritrasmissione (pari ad un pacchetto di dimensione minima)
- In caso di n-esima collisione di un pacchetto, si ritrasmette dopo ritardo casuale estratto tra 0 e 2k-1slot time, con k=min(n, 10)
- Backoff limit = 10: numero di tentativi oltre al quale non aumenta più il valor medio del back-off
- Attempt limit n=16: massimo numero di tentativi di ritrasmissione

Quando il round trip delay del segnale è basso, CSMA/CD è efficiente circa al 100%. Per soddisfare il punto due e il 3, gli adattatori rilevano il livello di tensione nel mezzo. In un singolo adattatore il CSMA/CD fuziona:

- l'adattatore ottiene il datagramma dal livello di rete, prepara il frame a livello DL e lo prepara in un buffer
- Se il canale è in attivo, inizia a trasmettere, altrimenti aspetta.
- Se rileva una collisione mentre trasmette, la termina e manda un segnale di disturbo (jam)
- Dopo aver inviato il jam entra in un tempo di attesa esponenziale (facile immaginare cosa sia): fa in modo che ci siano meno collisioni possibili. Trasmetto, mentre A trasmette, A butta ritrasmette mentre io trasmetto, io butto, un casino: aspetto K tempi di bit. Al primo colpo $K \sim U([0,0])$ al secondo $K \sim U([0,1])$, al terzo $K \sim U([0,3])$, ecc. Questa scelta deriva dal fatto che quando c'è una collisione non sappiamo quanti ci sono in mezzo, quindi se siamo in pochi k va bene piccolo, se siamo in tanti k è meglio grande.

Efficienza di ethernet

Sopporta un carico medio del 30% (3 Mb/s) con picchi del 60% (6 Mb/s). Sotto carico medio:

- Il 2-3% dei pacchetti ha una sola collisione
- Qualche pacchetto su 10,000 ha più di una collisione

Tecnologie ethernet

- 10 BASE-T
- 10 BASE-2
- 100 BASE-T
- 1000 BASE-LX
- 10G BASE-T

Il numero è la velocità dello standard, "BASE" significa che è in banda BASE (no modulazione), la parte finale è il mezzo fisico. T sono i doppini intrecciati, poi ci sono fibre, ecc.
I primi standard furono 10-BASE2/5 su due diversi tipi di cavi coassiali, limitati ad una lunghezza di 500m. Se si vuole più lungo usa un ripetitore.
Attualmente le reti ethernet prevedono degli switch per collegamenti punto punto: tutto è cambiato, la velocità, il formato dei MAC, frame.
Autmento velocità == diminuzione distanza massima: la 100 Mbps è limitata a 100m su cavo in rame e a parecchi km su fabri fibra. Ora c'è addirittura la Gigabit Ethernet:

- Utilizza il formato del frame standard di ethernet (retrocompatibilità con 10/100 baseT)
- Consente l'uso di collegamenti punto punto, sia broadcast
- Se broadcast usa CSMA/CD
- Se punto punto va anche in flull duplex

È stata standardizzata anche la 10GBASE-T, una cosa colossale.

PPP: protocollo punto-punto

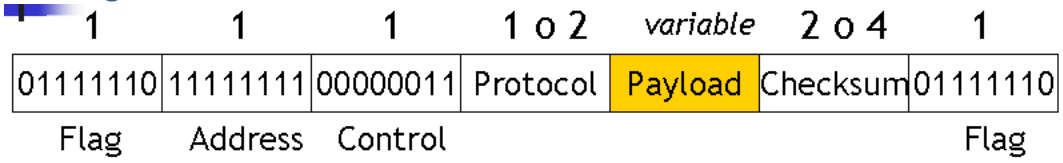
A livello di collegamento su reti punto punto il più utilizzato è il PPP. Attualmente si usa un protocollo simile HDLC (higl-level data link control). Requisiti:

- Framing dei pacchetti
- Trasparenza: non deve dare limitazioni ai livelli superiori
- Protocolli multipli del livello di rete: deve avere il campo protocollo
- Tipi di link multipli: deve far funzionare collegamenti seriali, paralleli, sincroni, asincroni, bassa/alta velocità, elettrici, ottici
- Rilevazione degli errori
- Disponibilità della connessione: rilevare eventuali guasti
- Negoziazione degli indirizzi di rete: deve fornire un meccanismo per ottenere/configurare gli indirizzi di reti
- Semplicità

Cosa NON fa PPP:

- **Correzione errori**
- **Controllo di flusso**: sincronizzazione
- **Sequenza**: ordine
- **Collegamento multi punto**

Framing dei dati



Questo è un frame che utilizza una delimitazione tipo HDLC

- **Flag**: si inizia e finisce con una sequenza nota
- **Indirizzo**: l'unico valore è 11111111
- **Controllo**: l'unico valore è 00000011

Questi due campi inutili sono stati fatti perché in futuro potrebbero servire, per ora possiamo anche inviare frame senza di loro.

- **Protocollo**: indica il protocollo del livello superiore
- **Informazioni**: i dati veri e propri
- **Checksum**: indovina un po'?

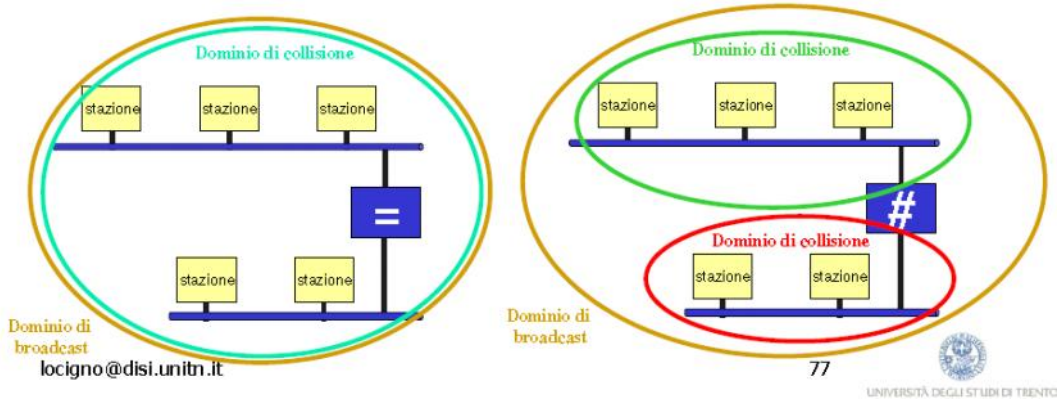
NB: bisogna escapare la sequenza di apertura/chiusura dentro i dati, altrimenti interpretiamo il frame come chiuso prima, per farlo si mette 0 ogni 5 uni consecutivi.

LAN estese

La scelta di utilizzare mezzi condivisi per l'accesso al canale di trasmissione è stata fatta sia per necessità (ad es. trasmissioni wireless) sia motivi economici. Grazie proprio agli aspetti economici, tale tecnologia è stata utilizzata e si è diffusa particolarmente nelle reti locali (Local Area Networks, LAN) La rappresentazione tipica di una LAN è una serie di stazioni (PC) connesse ad un segmento di cavo (bus). Poiché il segmento non può essere troppo lungo (attenuazione del segnale, disposizione spaziale delle stazioni all'interno di un edificio (ad es.: su più piani)) nasce il problema di come estendere le LAN. Esistono 3 tipi di apparati, in ordine crescente di complessità:

- Repeater o Hub (ciabatte della luce, solo livello fisico)
- Bridge (switch con 2 linee)
- Switch (pro, separano i domini di collisione, fino al livello data link)

Dominio di collisione: parte di rete per cui, se due stazioni trasmettono dati contemporaneamente, il segnale ricevuto dalle stazioni risulta danneggiato.
Dominio di broadcast (detto anche Segmento data-link): parte di rete raggiunta da una trama con indirizzo broadcast (a livello 2)
Stazioni appartenenti alla medesima rete di livello 2 condividono lo stesso dominio di broadcast. Gli apparati che estendo le LAN possono solo influire sul dominio di collisione



Switch a livello di collegamento

Lo switch è trasparente ai nodi, possono ricevere a velocità più veloci della loro ricezione grazie a dei Buf[STR_DIM].

Inoltro e filtraggio

- **Filtraggio**: il frame deve essere inoltrato a qualche interfaccia o deve essere scartato?
- **Inoltro**: individuare l'interfaccia giusta ed inviare

Ambedue sono fatti mediante una tabella di commutazione che contiene voci:

- MAC del nodo
- Interfaccia sulla quale è il nodo
- Quando la voce è stata inserita

La legge in questo modo (riassunto martinelliano, questa volta devo dire efficiente):

- Riceve un frame con destinazione a lui sconosciuta, lo invia su tutte le vie, tranne quella da cui proviene.
- Riceve un frame con destinazione che è nella stessa rete della sorgente, lo scarta, è già arrivato a destinazione
- Lo inoltra sull'interfaccia giusta altrimenti

Autoapprendimento

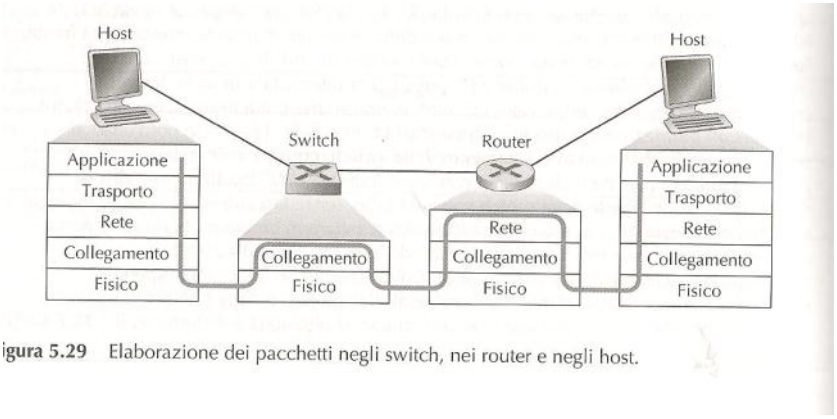
1. Tabella vuota
2. Quando arriva un frame si salva tutto: interfaccia, mac, tempo della sorgente ed inoltra su tutte le interfacce (tranne src). Quando tutti hanno inviato un frame la tabella p completa.
3. Dopo un po' di tempo cancella le voci troppo vecchie

Proprietà della commutazione a livello di collegamento

Vantaggi switch piuttosto che bus, hub, sflah, ping, pong, petrektek, niko petris:

- **Eliminazione delle collisioni**: grazie ai buffer
- **Collegamenti eterogenei**: dato che sono isolati l'uno dall'altro, possono andare a velocità differenti su mezzi diversi. Ideale per collegare roba vecchia già installata con roba nuova di zecca
- **Gestione**: se un adattatore è guasto e manda continuamente frame lo switch se ne accorge e lo elimina, oppure un cavo è tagliato: non serve un cinese obeso che riscriva la tabella.

Switch vs router



igura 5.29 Elaborazione dei pacchetti negli switch, nei router e negli host.

- **Router** sono commutatori di pacchetto store and forward che lavorano a livello di rete (IP).
- **Shitch** sono commutatori di pacchetto store and forward che lavorano a livello DL (MAC).

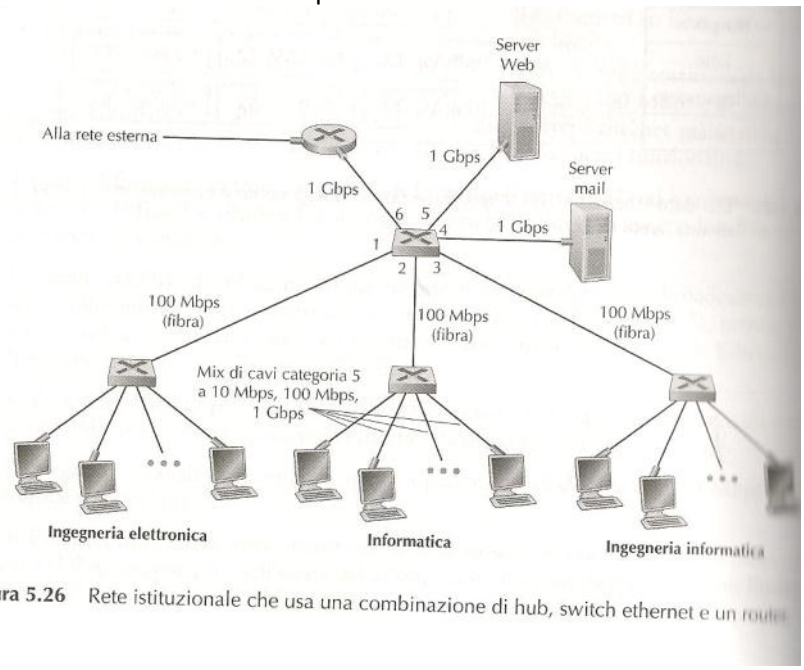


Figura 5.26 Rete istituzionale che usa una combinazione di hub, switch ethernet e un router

Per collegare le LAN dei dipartimenti, server e il router di gateway verso internet perché anche il router permette di separare le comunicazioni senza avere collisioni. Pro e contro di ambedue:

- **Pro switch**: plug-and-play, filtraggio, inoltro, velocissimi.
- **Contro switch**: rete limitata ad albero, per evitare cicli con i frame di broadcast. Un' ampia rete richiederebbe un grosso tabellone ARP. Non offrono nessuna protezione dalle tempeste di broadcast
- **Pro Router**: lavora con indirizzi gerarchici. I cicli ci possono essere se le tabelle di routing sono fatte male, comunque c'è sempre il numero di hop massimi che ci protegge dai cicli infiniti.
- **Contro router**: non sono plug-and-play, i loro indirizzi e quelli degli altri host devono essere configurati. Sono più lenti degli switch. IMPORTANTISSIMO: [cit. libro] -> 'esistono due differenti modi di pronunciare router: "ruuter" o "rauter", le persone perdono un sacco di tempo per decidere come si pronuncia' :|

	Hub	Router	Switch
Isolamento traffico	No	Sì	Sì
Plug and play	Sì	No	Sì
Instradamento ottimale	No	Sì	No

Plug and play	Sì	No	Sì
Instradamento ottimale	No	Sì	No
Cut	Sì	No	Sì

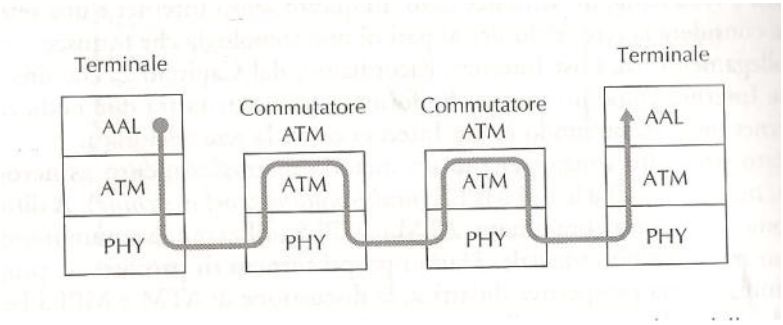
Ad c er reti da poche centinaia di host, conviene usare gli switch

NB: da qui in poi per il data link è finito, l'ho RIASSUNTO PER SBAGLIO

Canali virtuali: una rete come un livello di link

Trasferimento asincrono ATM ed MPLS (multi protocol label switching). A differenza della rete telefonica, ATM e MPLS utilizzano la commutazione di pacchetto e sono reti a circuito virtuale.

Trasferimento asincrono ATM



- Servizio a tasso costante
- Servizio a tasso variabile
- Servizio a tasso non specificato

È a commutazione di pacchetto, con circuiti virtuali. È divisa in 3 livelli:

AAL: (ATM adaption layer) "equivale" al livello di trasporto di Internet ed è presente solamente nei dispositivi di periferia. Dentro pacchetti AAL possiamo trovare tutti i protocolli superiori, ma anche IP. Il "frame" d AAL si chiama PDU: grandezza multipla di 48 byte perché deve adattarsi all'unità base "**cella ATM**".

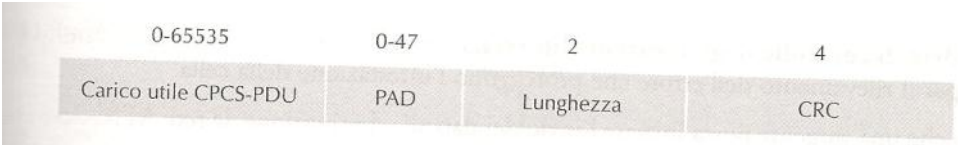
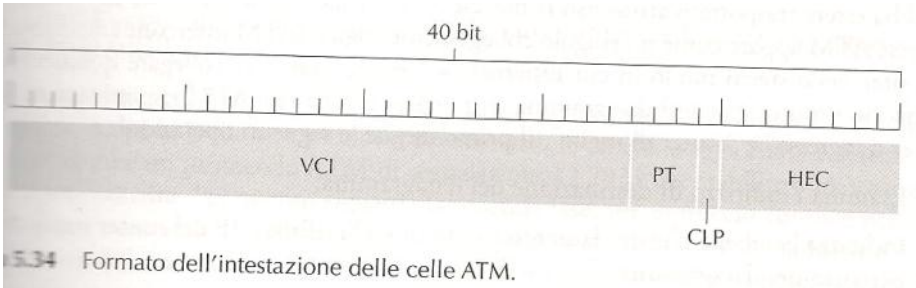


Figura 5.33 AAL5 PDU.

- AAL1 : servizio a tasso costante
- AAL2 : servizio a tasso variabile
- AAL5: servizio dati con trasporto di datagrammi

ATM: la cella ATM "equivale" ai datagrammi IP. I suoi campi:



5.34 Formato dell'intestazione delle celle ATM.

- **Identificatore del canale virtuale**: identifica il VC a cui appartiene la cella
- **Tipo di carico utile**: manutenzione/celle inattive ecc.
- **Bit di priorità sulla perdita di cella**: differenzia traffico di alta/bassa priorità. In caso di congestione, sceglie quali celle sacrificare

- **Bit di controllo degli errori nell'intestazione**

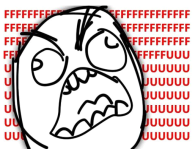
Prima di procedere all'invio deve essere stabilito un VC, su ciascuno dei nodi coinvolti è fissato un **identificatore del circuito virtuale**.
Il livello fisico si differenzia sostanzialmente in: apparecchiature per inviare frame, per inviare altro.

IP su ATM

Ultimo capitoletto a pagina 438, della lunghezza di 80 righe che ora non ho assolutamente voglia di fare

Layer 1 - Livello Fisico

martedì 31 maggio 2011
11:38



Mezzi e sistemi trasmissivi

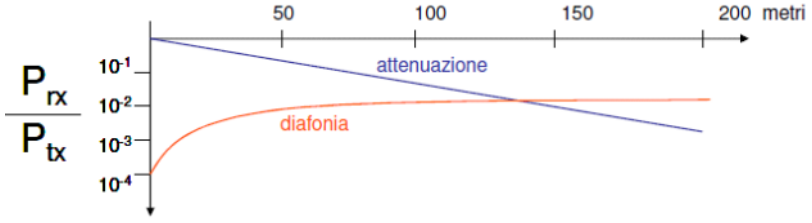
- Elettrici
  - Doppino non schermato
  - Cavo coassiale
- Ottici
  - Fibra ottica
  - Raggi laser
- Radio
  - Ponti radio
  - Satelliti
  - Reti cellulari

Ottimalmente
- Resistente, capacità, impedenze basse
- Resistenza alla trazione
- Flessibilità

Il che dipende da
- Geometria
- Numero di conduttori e distanza reciproca
- Isolante, schermatura

Parametri dei mezzi trasmissivi elettrici

- abstract class Impedence;
  - class Capacity : Impedence -> il voltaggio è in ritardo
  - class Induttance : Impedence -> la corrente è in ritardo
  - class Resistance : Impedence -> diminuisce la corrente
- Queste cose al cambiano in modo diverso in base alla frequenza.
- Velocità di propagazione del segnale (0.5c - 0.7c per cavi, 0.6c per fibre)
- Attenuazione cresce al crescere della distanza
- Diafonia o cross-talk



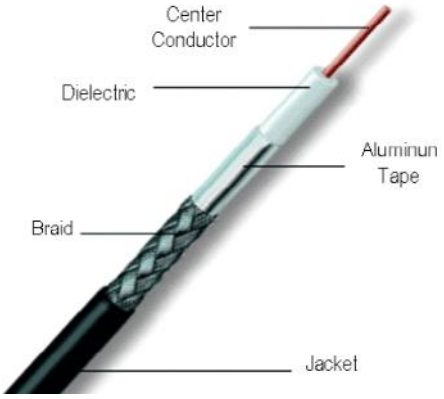
Doppino telefonico

Mezzo classico della telefonia, la ritorsione riduce le interferenze.

Costano poco, si installano in fretta. RJ45 sono quelli del telefono, con 4 coppie.
Versione senza schermatura, UTP: Unshielded Twisted Pair: usate per telefono e dati

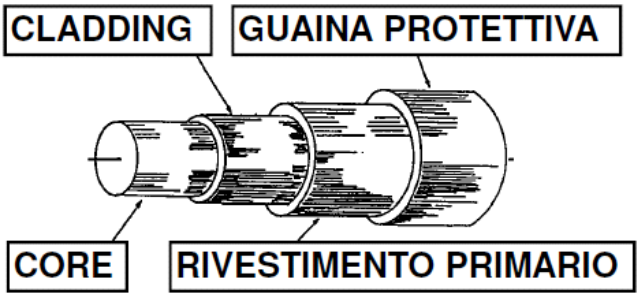
1	Telefonia analogica
2	Telefonia ISDN
3	Reti locali fino a 10 Mb/s
4	Reti locali fino a 16 Mb/s
5	Reti locali fino a 100 Mb/s
5e	Reti locali fino a 1 Gb/s
6	Reti locali fino a 1 Gb/s (migliore qualità di Cat.5e)
6a	Reti locali fino a 10 Gb/s

Cavo coassiale

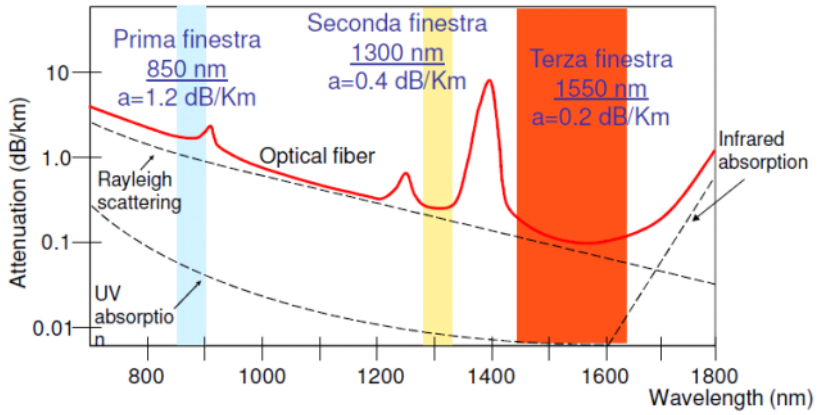
- Maggiore schermatura, costi elevati, difficile da installare
- Usato in USA per la tv via cavo
- 

Fibra ottica

- Filo di vetro costituito da Core e Cladding, con indici di rifrazione diversi
- Il raggio luminoso generato da un LED laser (con un certo angolo) rimane confinato nel Core



- Vantaggi
  - Completamente immune da disturbi elettromagnetici
  - Veloci (decine di terabit/sec)
  - Bassa attenuazione 0.1dB/km
  - Dimensioni ridotte, costi contenuti
- Svantaggi
  - Solo punto-punto
  - Difficili da collegare
  - Ridotto raggio di curvatura

- 

- I cavi vengono interrati sul fondo del mare (quelli oceanici sono flottanti)
- -amplificatori ogni 30/50 km

Trasmissione via radio (Etere)

- Fading ()variazione veloce dell'ampiezza del segnale dovuta alla combinazione in fase di "copie" dello stesso segnale
- Shadowing (variazione lenta dell'ampiezza del segnale)
- Generalmente molti più errori di quelli via cavo
- Interferenza da altri segnali
- Attenuazione
  - quadrato della distanza (condizioni ottime)
  - Potenze di 2.5-4 normalmente sulla Terra

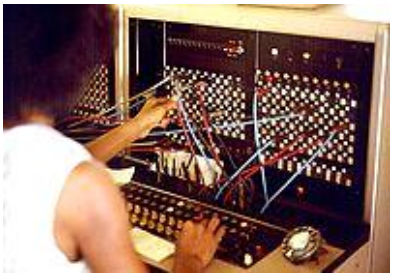
Reti di accesso - trasporto (non solo radio quindi)

Formula di Shannon: $C = B \log_2 \left(1 + \frac{S}{N}\right)$, dove S è il segnale, R il rumore, B la banda in Hz

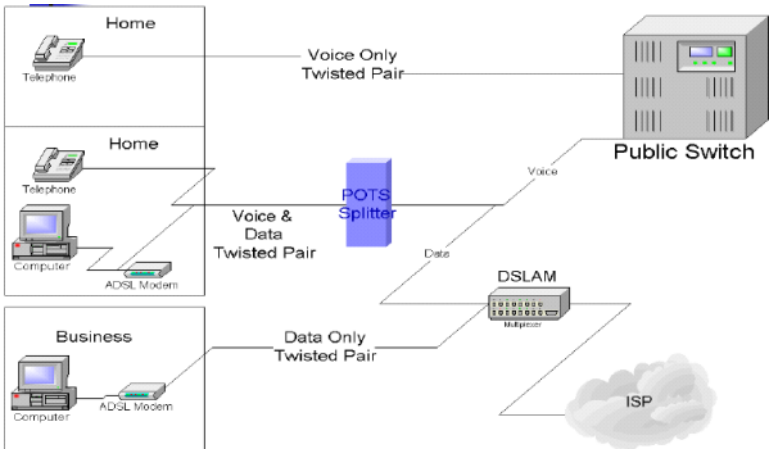
- **Rete di accesso**: collega l'utente al nodo di accesso del gestore dei servizi. Si tratta di "ultimo miglio" l'ultima parte di questa rete
- **Rete di trasporto**: backbone, tra uno o più gestori

Tecniche per l'ultimo miglio:

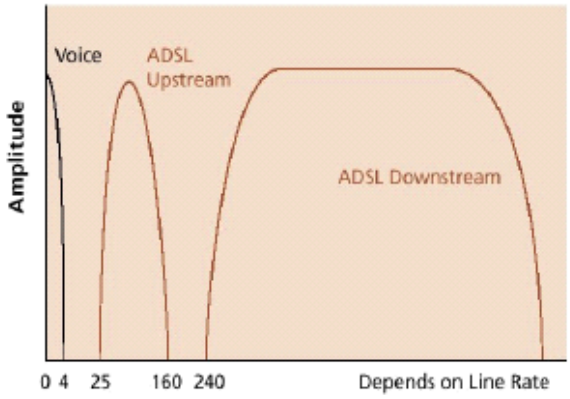
- Plain Old Telephone Service (POTS)
  - Modulatore/DEModulatore, rendono il segnale idoneo alla trasmissione su rete pubblica in tecnologia analogica su banda fonica
  - Banda 3000 Hz, segnale/rumore=35dB
  - Bitrate: 34 kbit/sec
  - I modem V.90 a 56 kbit/s (solo in download), mediante soppressione filtro fonico in download



- **ISDN/ADSL, Cable-TV**
- **Wi-MAX, GPRS, UMTS**
- **Ottiche**
- **Digital Subscriber Line (xD SL, tra cui Asymmetric DSL)**
  - ADSL, 2 e 2+: 6, 8, 24 Mb/s [downstream]
  - ADSL, 2 e 2+: 1 e mezzo, 3 e mezzo, 3 e mezzo [upstream]



- Filtro splitter: separa dati da voce



- DSLAM: converte da analogico a digitale il segnale proveniente dal subscriber e lo multiplexa

- **Fastweb**:
  - Offre fibra ottica quasi ovunque
  - Ha in progetto di offrire il satellitare
  - Usa il NAT, come se fosse un enorme LAN
  - La rete in questione è suddivisa per: città > zona della città > distretto > area elementare > progressivo edificio. Dal box di Fastweb, dove termina la fibra ottica, si dipartono infine i cavi che servono gli interni del palazzo o delle abitazioni.
  - Molti utenti della stessa zona hanno quindi lo stesso IP
  - Ha problemi per connettere utenti fra loro, infatti l'uno è invisibile all'altro.

Codifiche di linea e (cenni) alle tecniche di mo-demodulazione

- **Codifiche unipolari**
  - Molto semplici e primitive
  - Un livello di tensione per 0, un altro per 1
  - Problema: componente continua che può essere filtrata da alcuni sistemi
  - Perdita di sincronismo con lunghe sequenze dello stesso simbolo
  - In mezzi ottici, lunghe sequenze di 1 possono sovraccaricare il LED di trasmissione
- **Polari**
  - Due livelli di tensione con polarità diverse
  - Non-Return-Zero: non c'è transizione su tensione nulla
  - Return-to-Zero: transizione su zero tra due bit consecutivi
  - Bifase , es Manchester: ogni bit rappresentato da due livelli di tensione di polarità inversa (ideale per il sincronismo)
  - RZ e bifase richiedono doppia velocità per lo stesso bitrate
- **Bipolari**
  - Zero: tensione nulla
  - Uno: due polarità opposte, usate in alternanza
  - Permettono l'uso di simboli ternari, come 8B6T (o bit in 6 trit)
  - aka AMI

- **Codifiche nBmB:** codificano n bit su m bit
    - Richiedono meno banda delle codifiche polari
    - Permettono il controllo sulla scelta delle parole di codice, limitando troppi 0 e 1 consecutivi
    - Limita la componente continua
    - Fornisce caratteri speciali per delimitare pacchetti, idle o padding.

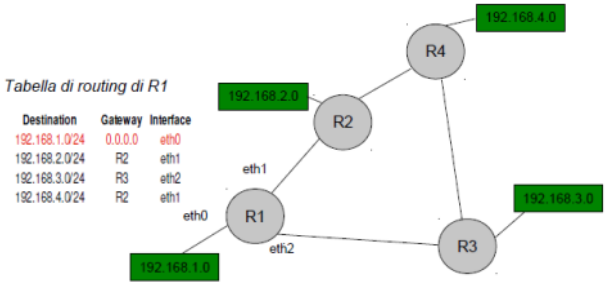
Modulazione: operazione di mappaggio bit su simboli analogici
Banda base: collegamenti cablati punto-punto a bassa velocità
Banda traslata: usati da fibre ottiche, ponti radio e tutti i sistemi moderni
La banda tralata consente la multiplazione in frequenza di canali diversi

Covelli - RIP, DHCP, DNS

Saturday, June 11, 2011
20:56

RIP

- Routing Information Protocol
- Le sottoreti collegate direttamente vengono rilevate in automatico
- Gli altri Next-Hop vanno aggiunti manualmente alla tabella di routing



- Questa gestione statica, effetuata manualmente diventa complicata ed error-prone quando aumentano il numero di sottoreti, se poi si rompono link c'è tutto da risistemare
- RIP consente di avere una configurazione dinamica delle tabelle di routing
- Algoritmo di distance vector (minimo numero di hop)
- All'accensione, la tabella viene popolata con le sottoreti direttamente collegate
- Ogni 30 secondi, il router broadcasta ai router adiacenti la propria tabella di routing, via UDP porta 520
- Il numero di hop può non essere sempre la scelta migliore, si usano allora protocolli alternativi (OSPF) che tengono conto non solo della topologia ma anche dello stato delle connessioni
- Se una metrica cambia improvvisamente (es. link down), gli aggiornamenti vengono inviati immediatamente ai router adiacenti
- Dopo 180 secondi senza aggiornamenti, una voce viene considerata obsoleta, e viene fatta una richiesta ai router adiacenti per ricevere aggiornamenti
- Formato
  - 1 byte: Command
  - 1 byte: Version number
  - 1 byte: unused
  - 1 byte: unused
  - 2 byte: Address Field Identifier (valore fisso corrispondente ad IP)
  - 2 byte: Route tag
  - 4 byte: Network Address
  - 4 byte: Subnet mask
  - 4 byte: Next hop
  - 4 byte: Metric

DHCP

- Funziona su UDP
- Usato per l'assegnazione automatica di IP address, subnet mask, default gateway, primary/secondary DNS
- Vantaggi
  - Gestione centralizzata
  - Si evitano errori di configurazione manuale
  - Riutilizzo indirizzo IP (quando non c'era NAT)
- Svantaggi
  - Single point of failure, se DHCP server non è ridondato
- Sostituisce il vecchio RARP
  - Richiedeva un server per ogni sottorete, perché faceva broadcast Ethernet
  - Forniva solo gli indirizzi IP

- Prima però è stato BOOTP a sostituire RARP
    - Invia tutti i parametri di configurazione
    - Consente di specificare il percorso del file con il sistema operativo con cui avviare l'host
    - Non consente assegnazione dinamica, IP è assegnato in base al MAC
    - Broadcast ethernet: le richieste sono numerate perché client diversi riconoscano solo le risposte destinate a loro
- DHCP supporta allocazione sia statica che dinamica
- Molto simile a BOOTP
- Come per BOOTP, le richieste vengono inviate in limited broadcast (255.255.255.255) - il client non sa ancora in che rete si trova. Ma questo non è un problema, tanto i router non lasciano fuoriuscire il traffico broadcast
- DHCP agent relay: consente l'inoltro di pacchetti di broadcast al server DHCP in modo da poter utilizzare lo stesso server su più sottoreti (anche BOOTP ha l'agent relay)
- DHCP DISCOVER su porta 68 (broadcast), è possibile richiedere un IP di propria preferenza
- Uno o più server gli rispondono con una DHCPOFFER (broadcast)
- Il client sceglie un IP (il primo, o quello con più lease time) e lo richiede con un DHCPREQUEST
- Il server risponde con un DHCPACK e gli riporta i parametri di connessione
- Scaduto il lease-time, viene inviata una nuova DHCPREQUEST

NAT

- Probabilmente non l'abbiamo fatto, comunque:
- Vantaggi
    - Riduce il numero di indirizzi pubblici utilizzati (con IPv6 non sarà necessario)
    - Maggior facilità di gestione degli host e nel cambio di ISP
    - In un certo senso, maggiore sicurezza, ma comunque un firewall è meglio
- Svantaggi
    - Rende complicata la vera connettività end-to-end
    - Problem di funzionamento con alcuni protocolli
    - Maggiore complessità di analisi di eventuali problemi di rete
- Terminologia:
    - Inside Local Address (es. 192.168.1.5)
    - Inside Global Address (es. 87.0.21.211)
    - Outside Global Address (indirizzo pubblico di un host esterno)
    - Outside Local Address (indirizzo privato di un host esterno)
- Port Address Translation (per sapere a che host è destinato un pacchetto proveniente dall'esterno)
- Per essere contattati dall'esterno (fare da server), occorre impostare il router o usare uPNP

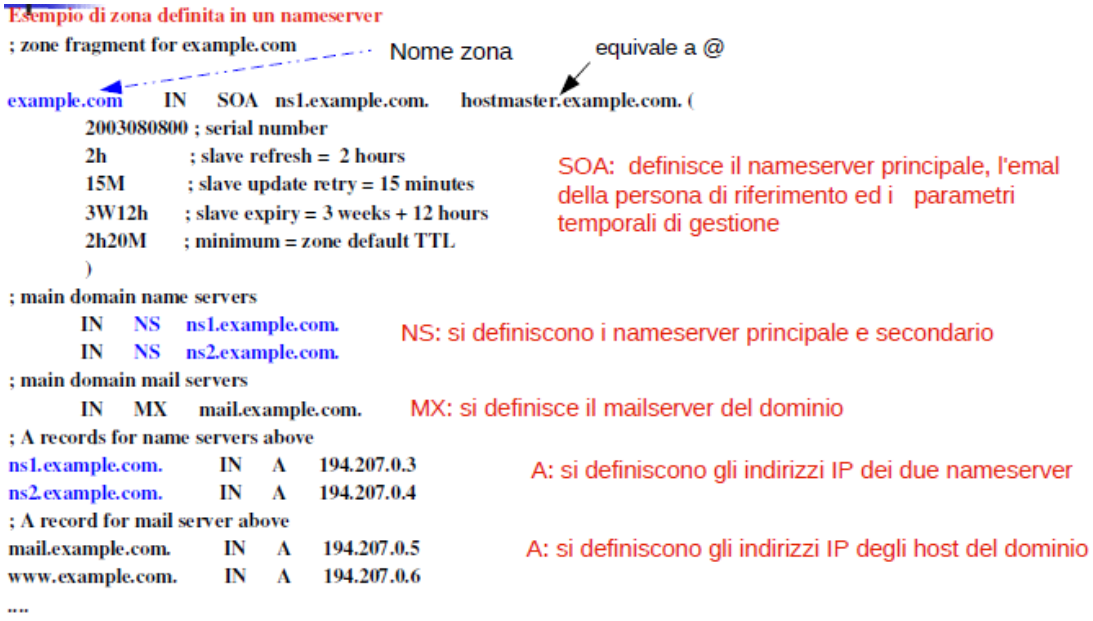
DNS

Introduzione:

- Ogni host visibile su internet viene identificato mediante uno specifico indirizzo, univoco a livello mondiale (32bits).
- Quest'indirizzo (IP) viene assegnato da Authorities (l'utente finale di solito è un Internet Service Provider) in modo da garantirne l'univocità.
- L'IP è uno dei parametri che vengono associati al messaggio, che serve per il riconoscimento degli host.
- I client non sanno quasi mai gli IP dei server, di conseguenza s'è deciso di instaurare un meccanismo che associ ad un ip, un nome mnemonico. Questa tipo d'associazione viene registrata in un particolare database (nameserver).
  NB: nell'applicazione viene utilizzato solo l'IP, il nome mnemonico è usato solo dall'utente dell'applicazione.
- Esiste un protocollo applicativo, richiamato da tutti gli altri protocolli applicativi, che consente la trasformazione del nome mnemonico in indirizzo IP, il DNS (Domain Name System) che definisce anche tutti gli elementi corollari e la struttura organizzativa che lo rendono possibile (DNS è incluso in uno

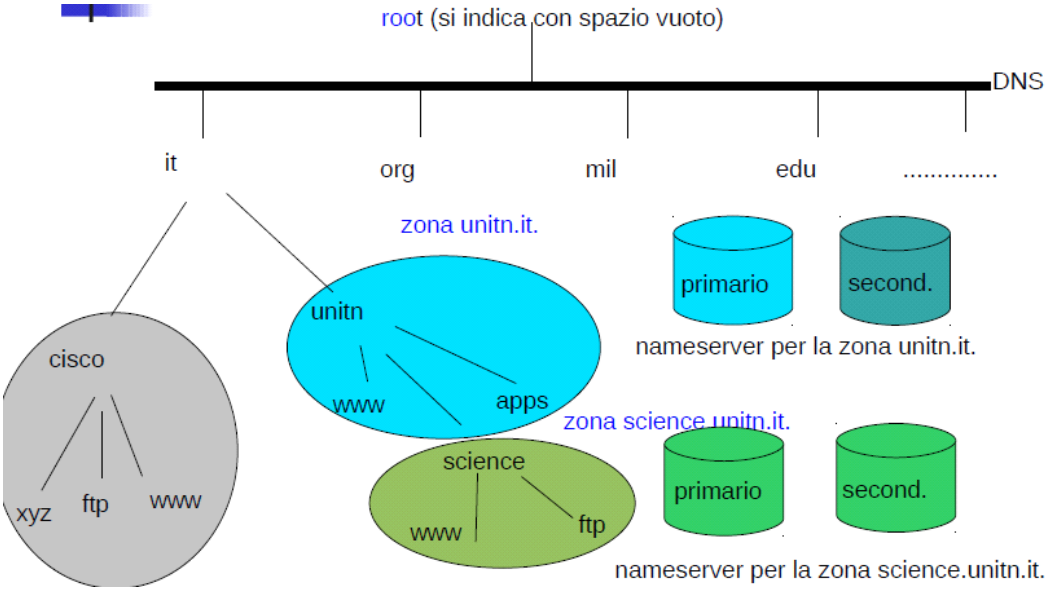
standard).
Fine intro

- Originariamente, c'era un file di testo "hosts" mantenuto dall'università di Stanford con l'associazione ip/mnemonico.
  I client scaricavano periodicamente via FTP quel file
- DNS consente una gestione distribuita del mapping, e nel frattempo una visione integrata ed unitaria
- Dominio di root: ""
- Dominio di primo livello: "com."
- Dominio di secondo livello: "example.com."
- Dominio di terzo livello: "dammuozz.selfip.com."



- Il nome di dominio (FQDN) è formato dai nomi, separati da punti, di tutte le label incontrate partendo dal nodo che ne rappresenta la foglia, salento verso la root (ogni nodo e foglia hanno una label).

- Organizzazione gerarchica anche per le authorities:
  - ICANN assegna i gTLD e ccTLD (generic e country-code Top Level Domains)
  - ICANN delega ad altre authorities (registry) la gestione dei domini di primo livello (it. al CNR di Pisa...)
  - I registry a loro volta delegano sotto domini di secondo livello (unitn.it) ad organizzazioni dette maintainer/registrar che registrano tale sottodominio presso il registy su richiesta del cliente richiedente (registrant). Le responsabilità di un certo sottodominio è del registrant, il maintainer è solo l'intermediario.
  - Zona: un sottoinsieme del namespace controllato da un'authority
  - Una zona può coincidere con un dominio, oppure ci possono essere più zone: unitn.it., science.unitn.it.
  - Il registrant mette solitamente a disposizione almeno 2 nameserver per la risoluzione dei nomi contenuti nella sua zona, oppure per comodità può essere gestita con i nameserver del registrant
  - Se un sottodominio in carico ad un registrant viene ulteriormente delegato, la zona può esser gestita mediante nameserver autonomi o tramite I nameserver del registrant delegante.



In sintesi, mentre il dominio è un concetto che si riferisce al contesto del namespace, il termine "zona" si riferisce al contesto della gerarchia delle autorithies.

- Il DNS dunque si basa sul concetto di delega.

Nameserver

- autoritativo per una certa zona
- è responsabile dei nomi di tutti gli host appartenenti a quella zona
- gestisce informazioni specifiche (RR resource record), le piu importanti sono:
  - SOA (Start of Authority), ne esiste uno per zona e riporta dei parametri di gestione
  - NS (name server), indica il nome dei nameserver (primario e secondari), dove reperire le informazioni di una zona
  - A (address), indica l'IP (un record per host)
    esempio -> slide 45

- Nel caso di zona delegata, sul nameserver delegato dovranno essere presenti delle informazioni (glue records) che consentano di puntare ai nameserver di competenza.
- Nameserver di root delegano dei nameserver che gestiscono il I° livello.
- Nameserver di I° livello delegano dei nameserver che gestioscono il II° livello.
- Il nameserver delegante deve riportare degli indirizzi IP dei nameserver della sua zona delegata (glue records) in modo che, in fase di risoluzione degli indirizzi, si possa , scorrendo la sequenza di deleghe che porta dai nameserver di root a quelli autoritativi, accedere direttamente a questi ultimi per reperire gli indirizzi IP degli host appartenenti alla zona cercata. (es slide 50)

  Database del registrar:
  example.com, ns1.example.com, NS
  example.com, ns1.example.com, NS
  ns1.example.com, 212.121.21.12, A

Vantaggi di DNS

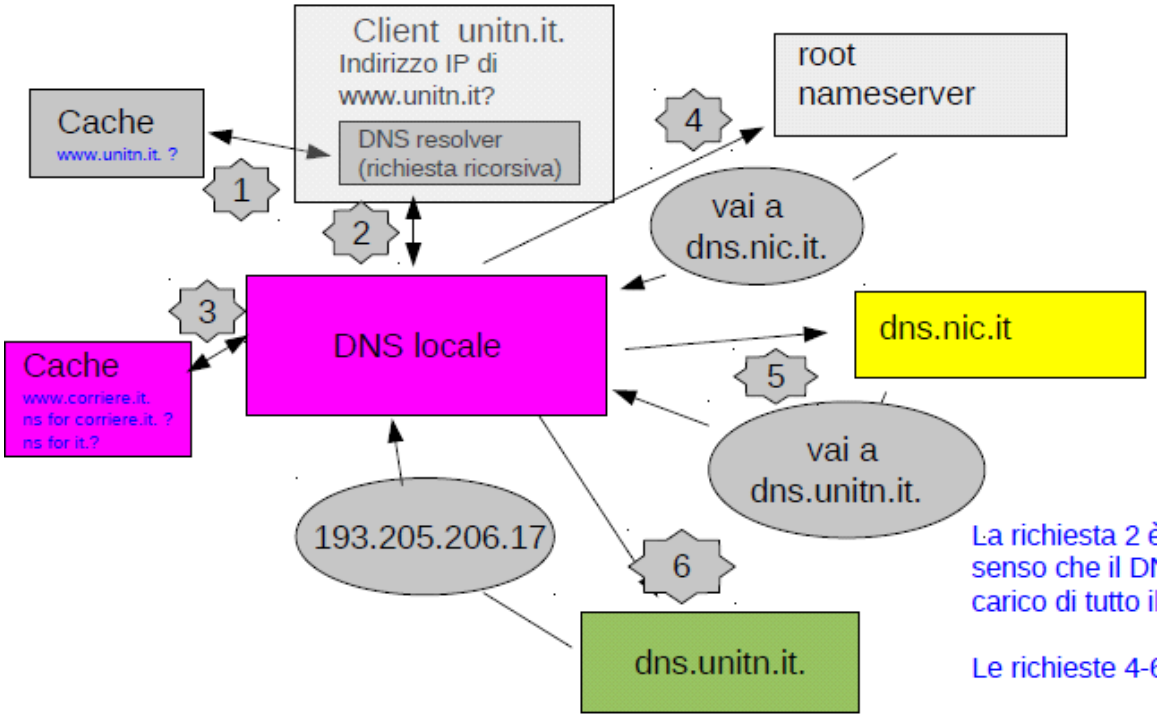
- Distribuzione del carico di lavoro e migliori performance
- Ridondanza e quindi affidabilità
- Ogni organizzazione è direttamente responsabile dei suoi dati
- Impossibilità di nomi duplicati

Modalità di risoluzione dei nomi

- Il client (browser) contatta il nameserver associato all'host sul quale è installato e chiede che venga risolto un certo nome (es: www.unitn.it) in modalità ricorsiva
- Il nameserver locale, se non ha già disponibile l'indirizzo richiesto in cache, contatta, in modalità

iterativa, uno dei rootserver richiedendo la risoluzione del nome cercato

- Il rootserver risponde fornendo gli indirizzi IP dei nameserver autoritativi per il dominio di I° livello presente nel nome cercato (nel nostro caso .it)
- Il nameserver locale contatta (iterativamente) uno di questi due nameserver che risponderà con gli indirizzi IP dei nameserver autoritativi per la zona unitn.it
- Il nameserver locale contatta uno dei due nameserver autoritativi per la zona unitn.it ed otterrà finalmente l'indirizzo IP dell'host cercato (www.unitn.it)
- L'indirizzo viene restituito al client.
- NB: parte di questo processo può essere semplificato tramite caching delle informazioni.
- Questa navigazione è resa possibile dai glue records



Covelli - Architetture Client/Server e p2p

mercoledì 15 giugno 2011
0.24

Introduzione: Architetture di rete dal punto di vista applicativo

- Due host, collegati in Internet, si scambiano fra loro dei messaggi, si trattan di messaggi codificati secondo determinati protocolli standard detti applicativi.
- Gli applicativi sono gestiti da applicazioni software apposite, realizzate ed installate negli host (livello 7-6-5 dallo stack ISO/OSI)
- Questi protocolli stabiliscono il formato dei messaggi e la loro sequenza di utilizzo
- Protocolli applicativi prevedono due differenti architetture:
    - Client/Server
    - Peer to peer (p2p)

Client / server

- Server
    - solitamente sono degli host ad elevata affidabilità
    - dislocati in apposite strutture (Data Center)
    - hanno installate applicazioni per soddisfare i servizi richiesti
- Client
    - indica una componente che accede ai servizi o alle risorse di un'altra componente, detta server

- Come funziona:
- Il client attiva una specifica connessione (socket) con il server
- richiede, mediante un particolare protocollo, un determinato servizio
- Il server riceve la richiesta
- Crea a sua volta una connessione specifica con il client (socket)
- Fornisce il servizio richiesto

- In questo tipo d'architettura il server è il punto piu vulnerabile (single point of failure)
- una sua compromissione, implicherebbe l'interruzione del servizio
- Tutta la richiesta di banda trasmissiva è concentrata su di esso
- Le performance posson dunque diventare critiche
- Soluzioni:
    - Server ridondanti
    - Servizio distribuito in modo trasparente su piu server fisici

Note importanti

- I messaggi applicativi non vengono trasmessi tali e quali ma sono suddivisi in pacchetti (blocchi di bytes di lunghezza fissa)
- I pacchetti contengono una parte del messaggio da trasmettere (payload) ed informazioni per la corretta consegna del pacchetto al destinatario (header)
- Trasmissione/ricezione di pacchetti tramite protocolli standard (TCP/UDP, IP)
- Questi protocolli standard sono necessari sia via Internet che via LAN e devon esser presenti sugli host client e server.
- I protocolli applicativi che analizzeremo si baseranno sui protocolli di internet, sono necessari per garantire l'internet working
- Questo concetto si esprime con il termine incapsulamento dei protocolli, ovvero i dati di protocollo di livello superiore (messaggi) vengono consegnati a quelli di livello inferiori (TCP/UDP, IP)

P2P

- Nato verso il 2000 con Napster
- Host è: client/server allo stesso tempo
- Punti di forza:

    - Maggiore banda utilizzabile per il download (ogni peer mette a disposizione la propria)
    - Ridondanza dei dati, replicati su piu host contemporaneamente
    - Minore gestione di costi (Non c'è il Datacenter)
- Lo scambio di dati avviene direttamente tra peers
- C'è un server che ha il compito di gestire gli indici per la ricerca
- Server possiede i nomi dei file scaricabili e ip attivi dei peers
- Messaggio : <length> (2byte)<type>(2byte)<data>
- Peer si logga al server
- Il server risponde con un ACK
- Peer invia la lista di file che vuole condividere
- Per il Download:
    - viene mandato il comando SEARCH al server dei file
    - Server risponde con un RESPONSE
    - Il messaggio di RESPONSE contiene <username> e <IP> dei peer che possiedono il file
    - Il peer sceglie uno dei peer e lo comunica al server
    - Il peer contatta l'altro peer e scarica

Pure P2P

- Orientato al filesharing
- Modello decentralizzato
- Per connettersi deve esserci almeno un peer già presente in rete
- Ricerca di un file:
    - messaggio di QUERY al peer connesso
    - indicando <file da cercare> <n° max di iterazioni di ricerca> (TTL)
    - se non c'è, decrementa di 1 il TTL e rinvia la richiesta ad altri peer conosciuti
    - risponde con un messaggio QUERYHIT che contiene l'elenco dei files in sharing
    - TTL = 0 blocca la diffusione
- Punti di forza:
    - Utilizzo condiviso della banda
    - informazioni sono ridondate e distribuite su piu peers (es: bittorrent)

Covelli - STACK TCP/IP, UDP, LINK, SOCKET

mercoledì 15 giugno 2011
1.19

Introduzione

- Socket: è un astrazione che rappresenta la comunicazione con un computer remoto
  Il client prepara un messaggio e lo invia al server, per far questo:
  - specifica una struttura in memoria (socket)
  - connette la socket al server
  - specifica porta e IP
- Socket: client applicativo
- canale di applicazione client/server semplificato tipo file virtuale
- dove van scritti i messaggi da trasmettere e letti i messaggi di risposta
- consente alle applicazioni di ignorare le problematiche connesse alla comunicazione
- messaggi = flusso indifferenziato di bytes

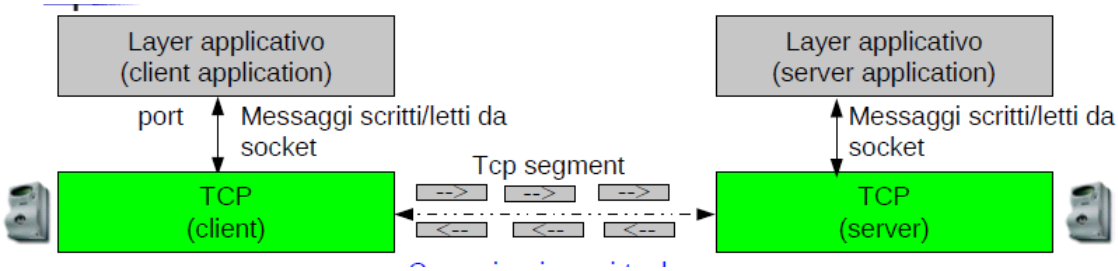
- Il server resta in listening su una socket specifica
- quando un client vuole connettersi crea un'altra socket specifica sulla quale verrano ricevuti i messaggi in partenza dal client ed inviati i messaggi di risposta (simmetricità della connessione)
- Altri strati software gestiscono i problemi di corretta trasmissione dei messaggi
- Non essendoci un unico strato software, perche ci sono problematiche molteplici e complesse, si preferisce far gestire il processo di trasmissione a strati a software diversi (**stack**)

TCP

- Primo layer protocollare: Transport Control Protocol
- Tcp riceve i messaggi dalla socket e svolge i seguenti compiti:
  - Suddivide i messaggi in pacchetti di dimensioni fisse (TCP Segment)
  - Ad ogni pacchetto aggiunge un intestazione con particolare informazioni (TCP Header). Questo per garantire che i pacchetti non vadano persi e vengano ricomposti dal ricevente secondo l'ordine con il quale essi sono stati inviati dal mittente.
  - Ad ogni TCP Segment viene associato un particolare contatore (Sequence Number)
  - Viene inizializzato in modo random in ogni nuova socket
  - Associato al primo byte del primo segmento
  - Sequence Number è un numero incrementale
  - Permette al destinatario TCP di ricostruire la corretta sequenza di pacchetti e di verificare la perdita eventuale di essi
- Quando il server riceve il pacchetto ed invia a sua volta i pacchetti TCP di risposta, viene memorizzato, in un particolare campo dei segmenti di risposta, l'acknowledgment number.
- Corrisponde al prossimo sequence number atteso
- Altro importante campo nell'header TCP è il numero di porta (destination port) che identifica in modo univoco il processo ricevente.
- Nell'header TCP viene specificato anche il numero di porta del mittente (source port) in modo da consentire la corretta consegna al processo client mittente dei TCP segment di risposta
- Altro campo dell'header TCP , è la window size.
- Informa il ricevente del numero massimo di byte che il mittente puo ricevere in risposta
- es: se windowsize = 0, il mittente non è in grado di ricevere alcun byte quindi il ricevente deve aspettare fino a quando riceve un nuovo segmento con windowsize > 0.

In sintesi

- TCP è lo strato software che si fa carico della corretta consegna dei messaggi e del controllo di flusso tra mittente e destinatario
- Esso dev'esser presente sia presso il mittente che presso il destinatario (controllo end to end)
- Il processo di risposta è simmetrico



- I messaggi scritti dal client su socket vengono suddivisi da TCP in segmenti sui quali son presenti informazioni di controllo
- questi segmenti arrivano al TCP del server dove vengon controllati e passati alla socket del server (individuata dal parametro "destination port" dell'header TCP.
- Il processo applicativo lato server riceve i messaggi ed invia il messaggio di risposta che segue un percorso simmetrico.
- In ogni pacchetto di risposta è presente un valore (acknowledgment number) che informa la controparte sul suo ultimo segmento ricevuto

UDP

- User Datagram Protocol
- Usato in alternativa al TCP
- UDP non suddivide i messaggi in pacchetti ma li ingloba in pacchetti (UDP datagram)
- Ogni messaggio è preceduto dall'header UDP
- I messaggi hanno lunghezza max 65000bytes circa.
- E' piu semplice di TCP
- Campi significativi :
  - porta mittente
  - porta destinatario
- Non esiste numerazione dei pacchetti e acknowledgment
- UDP non si fa carico alcuno della corretta consegna dei pacchetti (unreliable protocol)
- Modello piu snello e agevole
- Poco affidabile
- La gestione d'errore è lasciata ai protocolli applicativi
- Usato al posto di TCP quando:
  - si devono privilegiare le prestazioni rispetto alla perdita dei pacchetti (es: applicazioni multimediali)
  - si è in presenza di protocolli basati su un semplice meccanismo di messaggi domande-risposta (es: DNS)

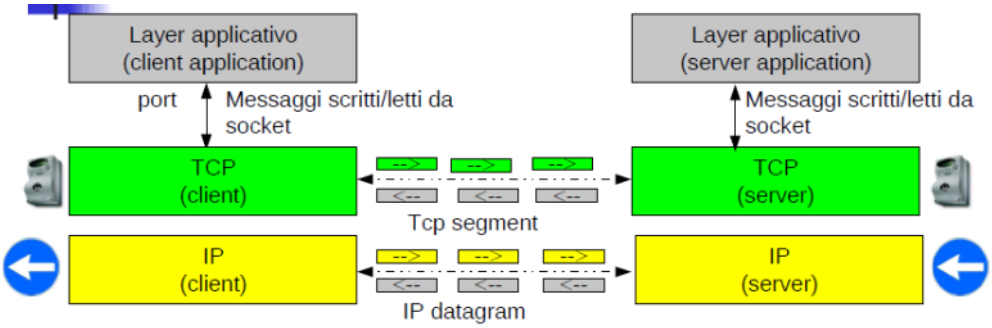
IP

- Internet Protocol
- Rappresenta il secondo layer software (dopo TCP / UDP)
- Prende in carico i messaggi in forma di TCP segment o UDP datagram
- Per ognuno di essi crea un pacchetto (IP datagram)
- Ci aggiunge il suo header, contenente fra le informazioni piu significative:
  - Indirizzo IP mittente (source address)
  - Indirizzo IP destinatario (destination address)
- Il nuovo pacchetto è quindi formato da:
  - IP header
  - TCP / UDP header
  - Dati originari (messaggi applicativi o una loro parte)
- Questo processo è chiamato incapsulamento (encapsulation)
- Le informazioni registrate nell'header vengono poi esaminate dal medesimo layer presente nel destinatario per realizzare i dovuti controlli
- Effettuati i controlli, il layer destinatario elimina il proprio header e passa i dati ricevuti al layer superiore (TCP / UDP)
- Il trattamento effettuato da IP è individuare il percorso necessario (route) per la consegna del pacchetto al destinatario.
- Ognuno di questi indirizzi ha una struttura che individua la LAN di appartenenza (net-id) e lo specifico host all'interno della LAN (host-id)

- Ci sono 2 casi:
  - stesso net-id (stessa LAN)
  - diverso net-id

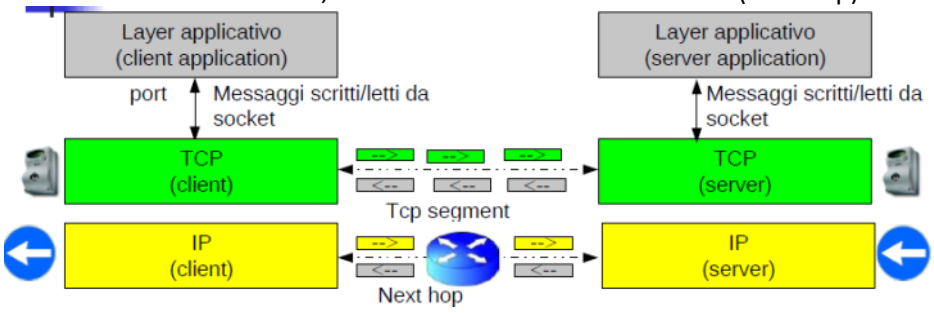
Stessa LAN:

- IP consegna il pacchetto al layer software sottostante (Datalink) per la trasmissione vera e propria dei dati al destinatario



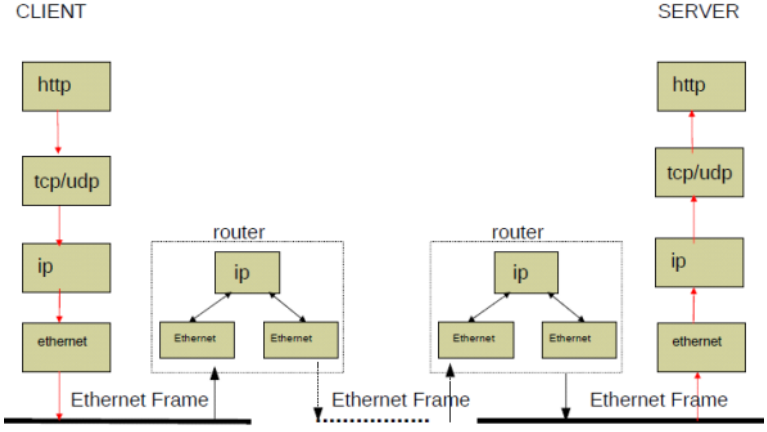
LAN diversa:

- IP consulta le tabelle di routing che indicano, per ogni IP di destinazione, a quale indirizzo IP il pacchetto vada consegnato (next hop)
- Una volta individuato il next hop, IP consegna il pacchetto al layer inferiore (Datalink)
- Poiché fra diverse reti (diverso net-id) è possibile l'interscambio di messaggi previsti dai vari protocolli di internet, occorre che esse siano fra loro collegate da router (Internetworking). Router: Tante schede di rete quante son le reti collegate
- IP è installato a bordo di ogni router ed il suo compito è proprio quello di esaminare i pacchetti da inoltrare decidendo, in base all'IP di destinazione, l'indirizzo IP del router successivo (next hop).



Link

- Ultimo layer software (link / network interface)
- Prende in carico i pacchetti IP dal livello immediatamente superiore (IP) e lo incapsula in un pacchetto
- Ethernet
- Ethernet inserisce il pacchetto IP in un frame che contiene fra le informazioni piu significative gli indirizzi fisici della scheda mittente e destinataria (MAC address)
- Ethernet invia in broadcast il frame sulla LAN
- Solo le schede di rete avente MAC eguale a quella del destinatario tratterà il frame
- Ethernet codifica pure i bits dei frame in segnali elettrici, ottici, radio per l'invio fisico dei dati al destinatario



Socket

- Area di memoria

- Vengono associati 5 parametri
  - Protocollo immediatamente inferiore da utilizzare (TCP / UDP)
  - Indirizzo IP mittente
  - Indirizzo IP destinatario (risolto eventualmente tramite DNS)
  - Porta mittente (emeferale)
  - Porta destinatario (well known)
- Il client creata la socket, si connette al server con un particolare comando messo a disposizione dalla libreria delle socket : Connect (viva la fantasia)
- Da questo punto in poi il client è in grado di inviare/ricevere messaggi sulla socket allo/dallo specifico server
- Il server deve già disporre di una socket in grado di ricevere le richieste di connessione (listening)
- La socket è un canale di comunicazione che consente lo sviluppo di applicazioni client/server.
- Alla socket si connetto contemporaneamente piu client.

Passaggi fondamentali

- Applicazione server crea una socket generica, in grado di ricevere messaggi (richieste di connessione) da parte di qualsiasi client.
- (bind) il server associa alla socket il suo IP e la porta well known (socket TCP / UDP)
- Il server mette in attesa le richieste di connessione da parte dei client, le richieste vengon poi messe in una specifica coda (listen)
- Ad ogni richiesta, il server accetta, creando una nuova socket, distinta da quella precedente e specifica per il client
- Nel caso iterativo (no fork) il server comunica con il client attraverso la nuova socket, terminata la connessione, riprende in considerazione la successiva richiesta presente nella coda di listening.
- Nel caso di server concurrent, il server crea con una fork un nuovo processo che gestirà la comunicazione con il client specifico attraverso la nuova socket generata dall'accept. Il padre invece tornerà a gestire la coda di listen.
- Il processo client si limita invece a creare una socket, ad associare ad essa una specifica porta ed uno specifico indirizzo IP e connettere tale socket al server