



UNIVERSITÀ  
DI TRENTO

Dipartimento di Ingegneria e Scienza dell'Informazione

Corso di Laurea in  
Ingegneria dell'Informazione e Organizzazione d'Impresa

**APPUNTI DI RETI**  
Dal materiale del prof. F. Granelli

Autore  
Davide Parpinello

Anno accademico 2019-2020

# Indice

<b>1 Roadmap</b>	<b>4</b>
1.1 Internet . . . . .	4
1.2 Ai confini della rete . . . . .	4
1.3 Il nucleo della rete . . . . .	4
1.3.1 Esempio di commutazione di circuito . . . . .	5
1.3.2 Esempio di commutazione di circuito . . . . .	5
1.3.3 Confronto fra commutazione di pacchetto e di circuito . . . . .	5
1.3.4 Struttura gerarchica . . . . .	5
1.4 Ritardi, perdite e throughput nelle reti a comunicazione di pacchetto . . . . .	6
1.4.1 Ritardo di un nodo . . . . .	6
1.5 Livelli di protocollo e i loro modelli di servizio . . . . .	7
1.6 Reti sotto attacco: la sicurezza . . . . .	7
<b>2 Il livello Applicazione</b>	<b>8</b>
2.1 Principi delle applicazioni di rete . . . . .	8
2.2 Web e HTTP . . . . .	9
2.2.1 Connessioni non persistenti . . . . .	9
2.2.2 Connessioni persistenti . . . . .	10
2.2.3 Messaggi HTTP . . . . .	10
2.2.4 Cookies . . . . .	11
2.2.5 Cache web (server proxy) . . . . .	11
2.2.6 HTTP/2.0 . . . . .	11
2.3 FTP . . . . .	11
2.3.1 Comandi comuni . . . . .	12
2.3.2 Codici di ritorno comuni . . . . .	12
2.4 Posta elettronica . . . . .	12
2.5 DNS . . . . .	13
2.5.1 Protocollo DNS . . . . .	14
2.6 Condivisione di file P2P . . . . .	15
2.6.1 Confronto tra architettura server client e P2P . . . . .	15
2.7 Cloud Computing . . . . .	16
2.7.1 CDN . . . . .	16
<b>3 Il livello di Trasporto</b>	<b>17</b>
3.1 Servizi a livello di trasporto . . . . .	17
3.1.1 Demultiplexing senza connessione . . . . .	17
3.1.2 Demultiplexing orientato alla connessione . . . . .	18
3.2 Trasporto senza connessione: UDP . . . . .	18
3.3 Principi del trasferimento dati affidabile . . . . .	19
3.3.1 Rdt 1.0: trasferimento affidabile su canale affidabile . . . . .	19
3.3.2 Rdt 2.0: canale con errori nei bit . . . . .	19

3.3.3	Rdt 2.1 . . . . .	20
3.3.4	Rdt 2.2: un protocollo senza NAK . . . . .	20
3.3.5	Rdt 3.0: canali con errori e perdite . . . . .	20
3.3.6	Pipelining . . . . .	21
3.4	Trasporto orientato alla connessione: TCP . . . . .	21
3.4.1	TCP: controllo di flusso . . . . .	22
3.4.2	Gestione della connessione . . . . .	23
3.5	Principi del controllo di congestione . . . . .	23
3.6	Controllo di congestione in TCP (AIMD) . . . . .	23
3.6.1	Partenza lenta . . . . .	24
3.6.2	Riassunto: controllo di congestione . . . . .	24
3.6.3	Throughput TCP . . . . .	24
3.6.4	Equità di TCP . . . . .	24
<b>4</b>	<b>Il livello di Rete</b>	<b>25</b>
4.1	Introduzione . . . . .	25
4.1.1	Funzioni chiave del livello di rete . . . . .	25
4.1.2	Modelli di servizio . . . . .	26
4.2	Reti a circuito virtuale e datagramma . . . . .	26
4.2.1	Reti a circuito virtuale . . . . .	26
4.2.2	Reti a datagramma . . . . .	26
4.2.3	Confronto tra le due tipologie . . . . .	27
4.3	Cosa si trova all'interno dei router? . . . . .	27
4.3.1	Architettura . . . . .	27
4.3.2	Porte d'ingresso . . . . .	28
4.3.3	Tecniche di commutazione . . . . .	28
4.3.4	Porte d'uscita . . . . .	29
4.3.5	Quale deve essere la capacità dei buffer? . . . . .	29
4.4	Protocollo Internet (IP) . . . . .	29
4.4.1	Formato dei datagrammi . . . . .	30
4.4.2	Indirizzamento IPv4 . . . . .	30
4.4.3	ICMP . . . . .	34
4.4.4	IPv6 . . . . .	34
4.5	Algoritmi di instradamento . . . . .	35
4.5.1	Stato del collegamento (link state) . . . . .	35
4.5.2	Algoritmo con vettore distanza . . . . .	36
4.5.3	Instradamento gerarchico . . . . .	37
4.6	Instradamento in Internet . . . . .	37
4.6.1	RIP (Routing Information Protocol) . . . . .	38
4.6.2	OSPF (Open Shortest Path First) . . . . .	38
4.6.3	BGP: instradamento inter-AS . . . . .	39
<b>5</b>	<b>Il livello di collegamento</b>	<b>41</b>
5.1	Livello di collegamento: introduzione e servizi . . . . .	41
5.1.1	Servizi offerti a livello di link . . . . .	41
5.1.2	Dov'è implementato il livello link . . . . .	42
5.2	Tecniche di rilevazione e correzione degli errori . . . . .	42
5.2.1	Controllo di parità . . . . .	42
5.3	Protocolli di accesso multiplo . . . . .	42
5.3.1	Protocolli a suddivisione del canale . . . . .	43
5.3.2	Protocolli ad accesso casuale . . . . .	43
5.3.3	Protocolli MAC a rotazione . . . . .	44
5.3.4	Riepilogo dei protocolli . . . . .	45

---

5.4	Indirizzi a livello di collegamento . . . . .	45
5.4.1	Indirizzi MAC e ARP . . . . .	45
5.5	Ethernet . . . . .	46
5.5.1	Topologia . . . . .	47
5.5.2	Struttura dei pacchetti Ethernet . . . . .	47
5.5.3	Servizio senza connessione non affidabile . . . . .	47
5.5.4	Fasi operative del protocollo CSMA/CD . . . . .	47
5.5.5	Efficienza di Ethernet . . . . .	48
5.5.6	Ethernet 802.3 . . . . .	48
5.5.7	Codifica Manchester . . . . .	48
5.6	Switch a livello di collegamento . . . . .	49
5.6.1	Hub . . . . .	49
5.6.2	Switch . . . . .	49
5.7	PPP: protocollo punto-punto . . . . .	50
5.7.1	Requisiti di IETF (RFC 1547) . . . . .	50
5.7.2	Formato dei pacchetti dati PPP . . . . .	51
5.8	Canali virtuali: una rete come un livello di link . . . . .	51
5.8.1	Il concetto di virtualizzazione . . . . .	51
5.8.2	Internet: virtualizzazione delle reti . . . . .	52
5.8.3	LAN virtuali . . . . .	53
5.8.4	ATM e MPLS . . . . .	55

# Capitolo 1

## Roadmap

### 1.1 Internet

Internet è costituito da milioni di dispositivi, chiamati **sistemi terminali**, collegati tra loro da collegamenti in rame, fibre ottiche, oppure via radio come onde elettromagnetiche o satelliti.

La frequenza di trasmissione in internet è data dall'ampiezza di banda disponibile.

Sulla rete internet inoltre sono presenti particolari host, denominati **router**, che si occupano di instradare i pacchetti verso la loro destinazione finale.

Nello scambio di messaggi tra host vengono implementati dei **protocolli**, che ne definiscono formato e ordine d'invio, così come le azioni intraprese in fase di trasmissione e/o ricezione di un messaggio o un altro evento.

Internet è un'infrastruttura di comunicazione per applicazione distribuita, viene anche chiamato "rete delle reti" ed è organizzato in modo gerarchico.

### 1.2 Ai confini della rete

Sul bordo della rete internet troviamo applicazioni e sistemi terminali, raggruppati tra loro e connessi tra di loro mediante collegamenti cablati e wireless.

I sistemi terminali (o host) fanno girare diversi programmi applicativi e possono essere organizzati con architettura client/server oppure P2P.

L'accesso a internet può avvenire mediante diversi modi:

- **Accesso residenziale:** viene utilizzato un modem dial-up o DSL.
- **Accesso aziendale:** una LAN collega i sistemi terminali di aziende e università all'**edge router**, i sistemi terminali sono collegati tra loro mediante uno switch ethernet.
- **Accesso wireless:** i terminali vengono collegati mediante **access point**.
- **Reti domestiche:** sono costituite da un modem DSL o via cavo, un router/firewall/NAT, switch ethernet e accesso wireless. Spesso queste funzioni vengono raggruppate in un unico dispositivo (modem/router).

### 1.3 Il nucleo della rete

Al centro della rete si trovano invece router interconnessi tra loro, che creano quindi una rete delle reti.

I dati nel nucleo della rete vengono trasferiti con due modalità differenti:

- **Commutazione di circuito:** è presente un circuito dedicato per l'intera durata della sessione
- **Commutazione di pacchetto:** i messaggi di una sessione utilizzano le risorse su richiesta, di conseguenza potrebbero dover attendere per accedere a un collegamento.

### 1.3.1 Esempio di commutazione di circuito

Consideriamo un file L di 640.000 bit, un bitrate totale C da 1.536 Mbps, TDM con 24 slot/s, 500ms per stabilire la connessione.

Trovo inizialmente la capacità di un singolo slot:

$$C_{1slot} = \frac{C}{24} = 0.064Mbps = 64Kbps$$

Successivamente calcolo il tempo necessario alla trasmissione:

$$T_{tx} = \frac{L}{C_{1slot}} = 10s$$

Infine, calcolo il tempo totale:

$$T_{tot} = 500ms + 10s = 10,5s$$

### 1.3.2 Esempio di commutazione di circuito

I secondi necessari per trasmettere un pacchetto in uscita su un collegamento da R bps sono dati da  $L/R$ , mentre il ritardo  $3L/R$

### 1.3.3 Confronto fra commutazione di pacchetto e di circuito

La commutazione di pacchetto consente un utilizzo della rete da parte di maggiori utenti, ed è ottima per i dati a raffica.

Dal lato negativo, presenta un'eccessiva congestione, causando ritardi e perdite di pacchetti. Sono quindi necessari protocolli per il trasferimento affidabile e per il controllo della congestione.

### 1.3.4 Struttura gerarchica

La rete internet ha una struttura fondamentalmente gerarchica:

- Al centro sono presenti **ISP di livello 1**, che offrono una copertura nazionale e/o internazionale
  - Comunicano tra loro come fossero pari
- **ISP di livello 2:** ISP più piccoli, copertura nazionale/distrettuale.
  - Si può connettere solo ad alcuni ISP di livello 1 e ad altri di livello 2
  - Paga l'ISP di livello 1 che gli fornisce la connettività per il resto della rete
- **ISP di livello 3 e ISP locali (di accesso):**
  - sono le reti "ultimo salto", le più vicine agli host
  - Sono clienti degli ISP di livello superiore che li collegano all'intera internet.

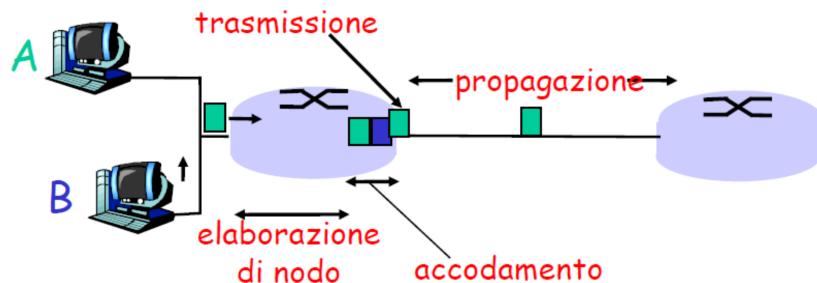
Un pacchetto attraversa un sacco di reti, dal livello più basso fino al principale e poi nuovamente a scendere.

## 1.4 Ritardi, perdite e throughput nelle reti a comunicazione di pacchetto

Nella rete si verificano dei ritardi quando il tasso di arrivo dei pacchetti eccede la capacità di evaderli, con la conseguenza che vengono accodati nei buffer del router in attesa del proprio turno. Se non ci sono buffer liberi i pacchetti vengono scartati e ritrasmessi dal nodo precedente o, in alcuni casi, non venire proprio ritrasmessi.

Quattro cause di ritardo dei pacchetti sono le seguenti:

1. Ritardo di elaborazione sul nodo
  - Controllo errori sui bit
  - Determinazione del canale di uscita
2. Ritardo di accodamento
  - Attesa di trasmissione
  - Livello di congestione del router
3. Ritardo di trasmissione (L/R)
  - $R$  = frequenza di trasmissione del collegamento
  - $L$  = lunghezza del pacchetto
4. Ritardo di propagazione ( $d/s$ )
  - $d$  = lunghezza dl collegamento fisico
  - $s$  = velocità di propagazione del collegamento ( $\sim 2 \cdot 10^8$  m/s)



### 1.4.1 Ritardo di un nodo

Il ritardo di un nodo è dato dalla seguente formula:

$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop} \quad (1.1)$$

dove:

- $d_{proc}$  = ritardo di elaborazione (processing delay)
  - in genere di pochi microsecondi, o anche meno
- $d_{queue}$  = ritardo di accodamento
  - dipende dalla congestione

- $d_{trans}$  = ritardo di trasmissione (transmission delay)
  - significativo sui collegamenti a bassa velocità
- $d_{prop}$  = ritardo di propagazione (propagation delay)
  - da pochi microsecondi a centinaia di millisecondi

### Ritardo di accodamento

- $R$  = frequenza di trasmissione (bps)
- $L$  = lunghezza del pacchetto (bit)
- $a$  = tasso medio di arrivo dei pacchetti

Se calcoliamo  $La/R$  otteniamo l'intensità di traffico:

- Vicino a 0: poco ritardo
- Minore o uguale a 1: traffico consistente
- Maggiore di 1: più lavoro in arrivo di quanto possa essere effettivamente svolto, ritardo medio infinito.

Il **throughput** viene calcolato come la frequenza (bit/unità di tempo) alla quale i bit sono trasferiti tra mittente e ricevente. Può essere istantaneo o medio (in un periodo più lungo).

In internet si considera anche il **collo di bottiglia**, ovvero un collegamento su un percorso punto-punto che vincola un throughput end to end.

## 1.5 Livelli di protocollo e i loro modelli di servizio

Si considerano principalmente 5 livelli di protocollo:

1. **Applicazione:** di supporto alle applicazioni di rete
2. **Trasporto:** trasferimento dei messaggi del livello applicazione tra modulo client e server, connessione tra processi applicativi
3. **Rete:** instradamento dei datagram dall'origine al destinatario
4. **Link (*collegamento*):** instradamento dei datagram attraverso una serie di commutatori di pacchetto
5. **Fisico:** trasferimento dei singoli bit

## 1.6 Reti sotto attacco: la sicurezza

I malintenzionati installano malware negli host attraverso internet. Il malware può raggiungere gli host attraverso virus, worm o cavalli di Troia. Un malware di spionaggio può registrare quanto viene digitato, i siti visitati e informazioni di upload. Gli host infettati possono diventare botnet e essere usati per lo spamming e attacchi DDoS. Un malware è spesso auto-replicante e da un host attaccato può passare ad altri host.

**Analisi di pacchetti** Chiamato anche packet sniffing, quando un'interfaccia di rete legge/registra tutti i pacchetti che la attraversano.

**IP spoofing** Invio di pacchetti con un indirizzo sorgente falso.

**Record-and-playback** Vengono "sniffati" dati sensibili per utilizzarli in un secondo momento.

## Capitolo 2

# Il livello Applicazione

### 2.1 Principi delle applicazioni di rete

Le applicazioni di rete sono costruite con diverse architetture

- Client-Server
  - L'host (o client) interagisce direttamente con il server
  - Più client ci sono più il servizio diminuisce
- Peer-to-Peer (P2P)
  - Non c'è un server sempre attivo
  - Coppie arbitrarie di host comunicano direttamente fra di loro
  - Facilmente scalabile ma difficile da gestire
- Architetture ibride
  - Skype, messaggistica istantanea
  - Connessione client-client, utilizzo del server per la ricerca dell'indirizzo della parte remota
- Cloud computing
  - Un insieme di tecnologie che permettono sia di archiviare dati che elaborarli tramite l'utilizzo di risorse distribuite e virtualizzate in rete
  - Creazione di copie di sicurezza preventive in modo automatico trasferendo tutta l'operatività online
  - Dati memorizzati in server farm
  - Sempre client-server ma basata sulla virtualizzazione

Un processo è un programma in esecuzione su un host. All'interno dell'host due processi comunicano utilizzando schemi di interprocesso, mentre su host differenti comunicano attraverso lo scambio di messaggi.

- **Processo client:** processo che dà inizio alla comunicazione
- **Processo server:** processo che attende di essere contattato

Un processo invia/riceve messaggi mediante la sua socket. La socket è analoga ad una porta mediante la quale un processo che vuole inviare un messaggio lo fa uscire. Questo presuppone l'esistenza di un'infrastruttura esterna che trasporterà il messaggio attraverso la rete fino alla porta del processo di destinazione.

Per il trasporto internet vengono utilizzati i servizi dei protocolli TCP e UDP.

### Servizio TCP

- **Orientato alla connessione:** è richiesto un setup fra i processi client e server
- **Trasporto affidabile** fra i processi d'invio e ricezione
- **Controllo di flusso:** il mittente non vuole sovraccaricare il destinatario
- **Controllo della congestione:** "stroppa" il processo d'invio quando la rete è sovraccaricata
- **Non offre** temporizzazione, garanzie sull'ampiezza di banda minima, sicurezza
- **Più affidabile ma si paga con ritardi di trasferimento**

### Servizio UDP

- Trasferimento dati inaffidabile fra processi d'invio e ricezione
- **Non offre** setup della connessione, affidabilità, controllo di flusso, controllo della congestione, temporizzazione né ampiezza di banda minima e sicurezza
- **Mandare i dati il più velocemente possibile ma senza completa affidabilità**, usato per gaming online e VoIP

## 2.2 Web e HTTP

L'HTTP, o HyperText Transfer Protocol, è un protocollo a livello applicazione per il web, che considera un client che richiede, riceve e visualizza gli oggetti del Web e un server, che invia gli oggetti in risposta ad una richiesta.

Viene utilizzato TCP nel seguente modo:

- Il client inizializza la connessione TCP con il server (crea una socket) tipicamente sulla porta 80
- Il server accetta la connessione TCP dal client
- Avviene lo scambio di messaggi HTTP tra browser e server web
- Chiusura della connessione TCP

HTTP è un protocollo *stateless* (senza stato), cioè il server non mantiene informazioni sulle richieste fatte dal client.

Le connessioni HTTP possono essere non persistenti o persistenti.

**RTT** Round trip time, tempo impiegato da un piccolo pacchetto per andare dal client al server e ritornare al client.

### 2.2.1 Connessioni non persistenti

Viene trasferito un solo oggetto su una singola connessione TCP. Il tempo di risposta è dato da:

- Un RTT per inizializzare la connessione TCP
- Un RTT per inviare la richiesta HTTP e i primi byte
- Il tempo necessario alla trasmissione del file

Quindi totale = 2 RTT + tempo di trasmissione. Le connessioni non persistenti presentano alcuni svantaggi:

- Richiedono 2RTT per ogni oggetto
- Overhead dell'OS per ogni connessione TCP
- I browser aprono spesso connessioni TCP parallele per caricare gli oggetti referenziati

### 2.2.2 Connessioni persistenti

Il server lascia la connessione TCP aperta dopo l'invio di una risposta, i successivi messaggi tra gli stessi client/server vengono trasmessi sulla connessione aperta, il client invia le richieste non appena incontra un oggetto referenziato, un solo RTT per tutti gli oggetti referenziati.

### 2.2.3 Messaggi HTTP

I messaggi HTTP possono essere di richiesta o di risposta.

#### Richiesta HTTP

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
User-agent: Mozilla/4.0
Connection: close Accept-language:fr
```

#### Tipi di metodi

La versione HTTP/1.0 fornisce i seguenti metodi per effettuare le richieste:

- **GET:** richiesta solamente mediante URL
- **POST:** vengono inseriti nel corpo della richiesta anche dati da inviare al server
- **HEAD:** chiede al server di escludere l'oggetto richiesto dalla risposta

Nella versione HTTP/1.1 vengono aggiunti i seguenti metodi:

- **PUT:** include il file nel corpo dell'entità e lo invia al percorso specificato nel campo URL
- **DELETE:** cancella il file specificato nel campo URL

#### Risposta HTTP

```
HTTP/1.1 200 OK
Connection close
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998 ...
Content-Length: 6821
Content-Type: text/html
```

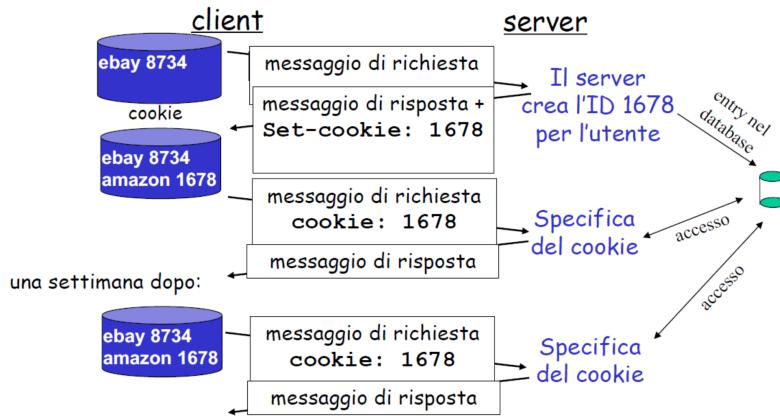
#### Codici di stato della risposta HTTP

Nella prima riga del messaggio di risposta sono inclusi codice di stato e relativa espressione in base all'esito della richiesta. Alcuni codici sono i seguenti:

- **200 OK:** La richiesta ha avuto successo, oggetto inviato nella risposta
- **301 Moved Permanently:** Oggetto trasferito nella nuova posizione indicata da Location

- **400 Bad Request:** il messaggio di richiesta non è stato compreso dal server
- **404 Not Found:** il documento richiesto non si trova su questo server
- **505 HTTP Version Not Supported:** il server non ha la versione di protocollo HTTP indicata

#### 2.2.4 Cookies



I cookies possono essere utilizzati per: autorizzazioni, dati carte per acquisti, raccomandazioni all’utente e possono mantenere lo stato del mittente e del ricevente per più transazioni; i messaggi HTTP trasportano lo stato. I cookies permettono ai siti di imparare molte cose sugli utenti.

#### 2.2.5 Cache web (server proxy)

L’obiettivo di una web cache è quello di soddisfare la richiesta del client senza coinvolgere il server d’origine. La cache opera come client e come server, e consente di ridurre i tempi di risposta alle richieste dei client, riducendo il traffico sul collegamento a internet.

**GET condizionale** Ha l’obiettivo di non inviare un oggetto se la cache ne ha una copia aggiornata. Per questo la cache specifica la data della copia dell’oggetto nella richiesta HTTP: `If-modified-since: <date>` A lato server la risposta non contiene l’oggetto se la copia nella cache è aggiornata: `HTTP/1.0 304 not modified`

#### 2.2.6 HTTP/2.0

È un’evoluzione di HTTP, che mantiene quindi i metodi HTTP, i codici di stato e la semantica migliorando però le prestazioni.

### 2.3 FTP

FTP, o File Transfer Protocol, viene utilizzato per trasferire file con un host remoto. Il server FTP opera solitamente sulla porta 21. Vengono aperte due connessioni, prima una di controllo e successivamente, se andata a buon fine la prima, una seconda per il trasferimento del file. La connessione di controllo è quindi "fuori banda" (out of band). Il server FTP mantiene lo stato

### 2.3.1 Comandi comuni

- USER `username`
- PASS `password`
- LIST elenca i file della directory corrente
- RETR `filename` recupera un file dalla directory corrente
- STOR `filename` memorizza un file nell'host remoto

### 2.3.2 Codici di ritorno comuni

- 331 Username OK, password required
- 125 data connection already open; transfer starting
- 425 Can't open data connection
- 452 Error writing file

## 2.4 Posta elettronica

La posta elettronica è costituita da tre componenti principali: agente utente, server di posta, SMTP (Simple Mail Transfer Protocol).

**Agente utente** Detto anche mail reader, si occupa della composizione, editing e lettura dei messaggi di posta elettronica (esempi: Outlook, Thunderbird). I messaggi in uscita sono memorizzati sul server.

**Server di posta** In esso è contenuta la casella di posta (mailbox) che contiene i messaggi in arrivo per l'utente e la coda dei messaggi da trasmettere.

**Protocollo SMTP** Utilizzato tra i server per inviare messaggi: il client è il server di posta trasmittente e il server il ricevente. Viene utilizzato TCP per trasferire in modo affidabile i messaggi di posta dal client al server sulla porta 25; il trasferimento è diretto. Ci sono tre fasi per il trasferimento: handshaking (saluto), trasferimento dei messaggi e chiusura. L'SMTP utilizza connessioni persistenti.

Esistono poi diversi protocolli di accesso alla posta:

- POP: Post Office Protocol. Utilizzato per autorizzazione e download
- IMAP: Internet Mail Access Protocol. Ha più funzioni e consente di manipolare i messaggi memorizzati sul server.
- HTTP: GMail, Hotmail, ecc...

**POP3** Suddiviso in due fasi: autorizzazione e transazione, è un protocollo senza stato tra le varie sessioni

**IMAP** Mantiene tutti i messaggi sul server, consente all'utente di organizzare i messaggi in cartelle e conserva lo stato tra le varie sessioni (nomi delle cartelle, associazione tra identificatori dei messaggi e nomi delle cartelle).

## 2.5 DNS

Il DNS, Domain Name System è un protocollo a livello applicazione che consente a host, router e server DNS di comunicare per risolvere i nomi dei siti web. Il DNS traduce un hostname come *www.facebook.com* in un indirizzo IP.

**Host aliasing** In alcuni casi, un host può avere più nomi, come nel caso dell'aliasing dei server mail.

Il DNS è un database distribuito implementato in una gerarchia di server DNS. Tipicamente, un server DNS radice viene contattato da un server DNS locale che non può tradurre un nome. A sua volta, il DNS radice contatta un server DNS autorizzato se non conosce la mappatura del nome, la ottiene e la restituisce al server DNS locale.

**Server TLD (top-level domain** Si occupano dei domini .com, .org, .net, .edu, ecc. e di tutti gli altri domini locali di alto livello, quali .uk, .fr, .ca e .jp.

**Server di competenza (authoritative server** Ogni organizzazione dotata di host internet pubblicamente accessibili (server web e di posta) deve fornire i record DNS di pubblico dominio che mappano i nomi di tali host in indirizzi IP; possono essere mantenuti dall'organizzazione o dal service provider.

Il server DNS locale non appartiene strettamente alla gerarchia dei server, ciascun ISP (università, società) ha un server DNS locale. Quando un host effettua una richiesta DNS, la query viene inviata al suo server DNS locale, che opera da proxy e inoltra la query in una gerarchia di server DNS.

Le query DNS possono essere effettuate in due modalità: iterativa o ricorsiva. Nel caso di query iterativa, il server contattato risponde con il nome del server da contattare, nel caso in cui non riesca a risolvere il nome; sarà quindi compito dell'host iniziale contattare quel server. Nel caso di query ricorsiva l'host affida il compito di tradurre il nome al server DNS contattato; in questo caso sarà il server stesso a contattare un secondo server per risolvere il nome, al quale può a sua volta assegnare il compito in modo iterativo.

Una volta che un server DNS impara la mappatura la mette nella cache, dove le informazioni vengono invalidate dopo un certo periodo di tempo. Tipicamente, un server DNS locale memorizza nella cache gli indirizzi IP dei server TLD, quindi i server radice non vengono visitati spesso.

Il DNS è un database distribuito che memorizza i record di risorsa (RR) che hanno il seguente formato: **(name, value, type, TTL)**

- Type = A
  - **name** è il nome dell'host (server)
  - **value** è l'indirizzo IP
- Type = NS
  - **name** è il dominio (**foo.com**)
  - **value** è il nome dell'host del server DNS di competenza di questo dominio
- Type = CNAME
  - **name** è il nome alias di qualche nome canonico (**www.ibm.com**)
  - **value** è il nome canonico (**servereast.backup2.ibm.com**)
- Type = MX
  - **value** è il nome del server di posta associato a **name**

### 2.5.1 Protocollo DNS

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

#### Intestazione del messaggio (12 byte - 6 campi)

L'intestazione è costituita da un numero da 16 bit di identificazione della domanda, utilizzato uguale dalla risposta, e da un flag indicante:

- domanda o risposta
- richiesta di ricorsione
- ricorsione disponibile
- risposta di competenza

#### Resto del messaggio

- Campi per il nome richiesto e il tipo di domanda
- RR nella risposta alla domanda
- Record per i server di competenza
- Informazioni extra che possono essere usate

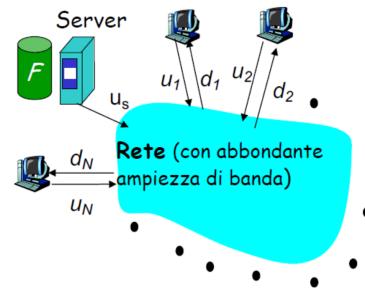
## 2.6 Condivisione di file P2P

Nell'architettura P2P pura non c'è un server sempre attivo, infatti coppie arbitrarie di host (peer) comunicano direttamente tra loro. I peer non devono essere sempre attivi e possono cambiare indirizzo IP.

### 2.6.1 Confronto tra architettura server client e P2P

#### Distribuzione di file: server-client

- Il server invia in sequenza  $N$  copie:
  - ❖  $\text{Tempo} = NF/u_s$
- Il client  $i$  impiega il tempo  $F/d_i$  per scaricare

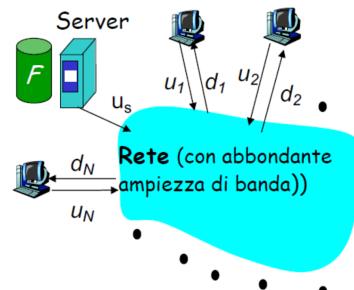


$$\text{Tempo per distribuire } F \text{ a } N \text{ client usando l'approccio client/server} = d_{cs} = \max \left\{ NF/u_s, F/\min_i(d_i) \right\}$$

aumenta linearmente con  $N$  peer

#### Distribuzione di file: P2P

- il server deve inviare una copia nel tempo  $F/u_s$
- il client  $i$  impiega il tempo  $F/d_i$  per il download
- Devono essere scaricati  $NF$  bit
- Il più veloce tasso di upload è:  $u_s + \sum u_i$



$$d_{P2P} = \max \left\{ F/u_s, F/\min_i(d_i), NF/(u_s + \sum u_i) \right\}$$

**Tracker** Il tracker tiene traccia dei peer che partecipano alla rete, dove un torrent è un gruppo di peer che si cambiano parti di un file.

Un file condiviso è diviso in più parti da 256 Kb, chiamati *chunk*. Il peer invia le sue parti a 4 vicini, quelli che gli stanno inviando alla frequenza più alta (aggiornata ogni 10 secondi). Successivamente, ogni 30 secondi viene scelto un peer a caso che riceve i chunk; oltre a questi 5 gli altri peer non riceveranno nulla.

**Query flooding** Ciascun peer indica i file che rende disponibili per la condivisione. Il messaggio di richiesta è trasmesso sulle connessioni TCP esistenti, il peer inoltra la richiesta e il messaggio di successo più trasmesso viene inviato sul percorso inverso.

## 2.7 Cloud Computing

L'architettura del cloud computing prevede uno o più server reali, generalmente in architettura ad alta affidabilità e fisicamente collocati presso i datacenter del fornitore del servizio.

### 2.7.1 CDN

Le CDN, o Content Delivery Networks, rappresentano una soluzione comune per la realizzazione di servizi su internet, la quale costruisce una rete "overlay" per la distribuzione di contenuti. Serve per memorizzare i dati il più vicino possibile ai consumatori in modo da ottimizzare le prestazioni di rete, ridurre la latenza ed evitare colli di bottiglia.

# Capitolo 3

## Il livello di Trasporto

### 3.1 Servizi a livello di trasporto

I servizi a livello di trasporto forniscono la comunicazione logica tra processi applicativi di host differenti. I protocolli di trasporto vengono eseguiti nei sistemi terminali: il lato invio scinde i messaggi in segmenti e li passa al livello di rete, il lato ricezione invece riassembra i segmenti in messaggi e li passa al livello applicazione.

- **Livello di rete:** si occupa della comunicazione logica tra host
- **Livello di trasporto:** si occupa della comunicazione logica tra processi, si basa sui servizi del livello di rete e li potenzia

**Demultiplexing** Nell'host ricevente, si occupa di consegnare i segmenti ricevuti alla socket appropriata.

**Multiplexing** Nell'host mittente, raccoglie i dati da varie socket, li incapsula con l'intestazione (usata poi nel demultiplexing).

L'host riceve i datagrammi IP; ogni datagramma ha un indirizzo IP di origine e uno di destinazione, e trasporta 1 segmento a livello di trasporto, ogni segmento ha un numero di porta di origine e un numero di porta di destinazione. L'host usa gli indirizzi IP e i numeri di porta per inviare il segmento alla socket appropriata.

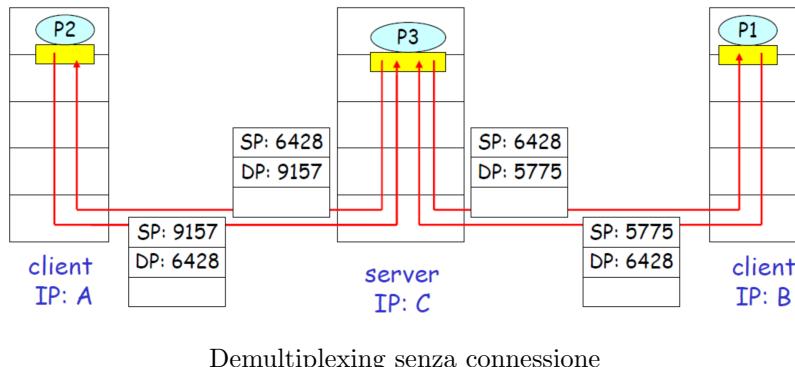
#### 3.1.1 Demultiplexing senza connessione

Il demultiplexing senza connessione utilizza una socket UDP identificata da indirizzo IP di destinazione e numero della porta di destinazione.

Quando l'host riceve il segmento UDP controlla il numero di porta nel segmento e poi lo invia alla socket correlata. Datagram IP con indirizzi IP e/o numeri di porta di origine differenti vengono inviati alla stessa socket.

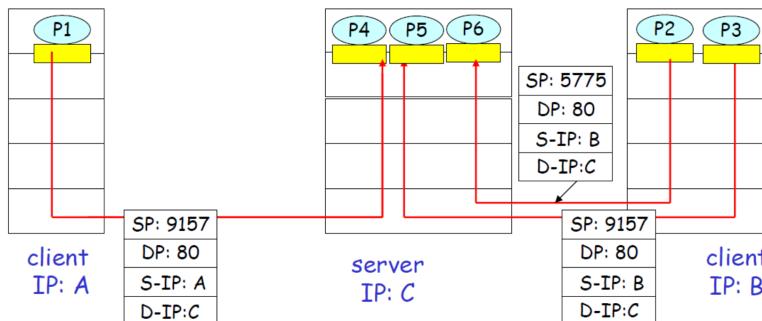


Struttura del segmento TCP/UDP



### 3.1.2 Demultiplexing orientato alla connessione

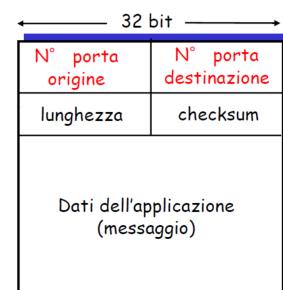
Il demultiplexing orientato alla connessione utilizza una socket TCP identificata da quattro parametri: indirizzo IP e porta d'origine e indirizzo IP e porta di destinazione. L'host ricevente usa tutti e quattro i parametri per inviare il segmento alla socket appropriata. Un host server può supportare più socket TCP contemporanee. I server web hanno socket differenti per ogni connessione client, con HTTP non-persistente si avrà invece una socket differente per ogni richiesta.



## 3.2 Trasporto senza connessione: UDP

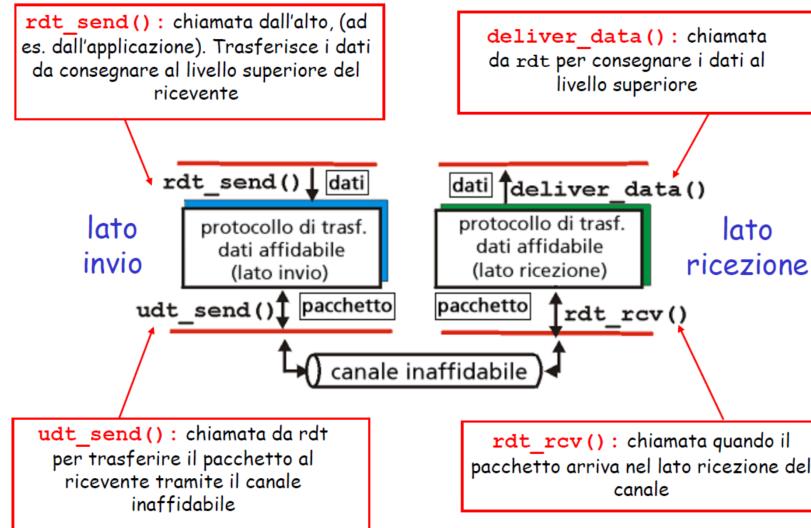
UDP, User Datagram Protocol, è un protocollo di trasporto "senza fronzoli", infatti ha un servizio di consegna best effort (miglior sforzo). Per questo i segmenti UDP possono essere perduti o consegnati fuori sequenza all'applicazione.

Essendo senza connessione non c'è handshaking tra mittente e destinatario, quindi ogni segmento UDP è gestito indipendentemente dagli altri.

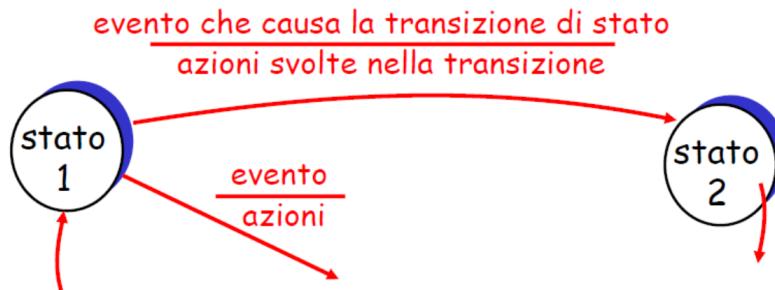


**Checksum UDP** Il checksum UDP serve per rilevare gli "errori" (bit alternati) nel segmento trasmesso, il segmento viene trattato come una sequenza di interi da 16 bit.

### 3.3 Principi del trasferimento dati affidabile



**Stato di Rdt** Lo stato successivo a quello corrente è determinato unicamente dall'evento successivo



#### 3.3.1 Rdt 1.0: trasferimento affidabile su canale affidabile

Il canale sottostante è perfettamente affidabile e è presente un automa distinto per mittente e ricevente.

#### 3.3.2 Rdt 2.0: canale con errori nei bit

Il canale sottostante potrebbe confondere i bit nei pacchetti, si utilizza quindi il checksum per rilevare gli errori. Una volta ricevuto, con la notifica positiva **ACK** il ricevente comunica esplicitamente al mittente che il pacchetto ricevuto è corretto mentre con la notifica negativa **NAK** comunica che il pacchetto contiene errori. Se il mittente riceve un **NAK** verrà ritrasmesso il pacchetto.

Con Rdt 2.0 sono stati introdotti nuovi meccanismi tra cui il rilevamento di errore e il feedback del destinatario (**ACK** e **NAK**).

Rdt 2.0 però ha un difetto fatale: se i pacchetti **NAK** e **ACK** sono danneggiati il mittente non saprà cos'è successo; non basta però ritrasmettere la notifica perché sono possibili duplicati. Il mittente ritrasmette quindi il pacchetto aggiungendo un numero di sequenza, e il ricevente lo scarterà se duplicato. Una volta inviato, il mittente aspetta la risposta del destinatario (*stop and wait*).

### 3.3.3 Rdt 2.1

Il mittente aggiunge un numero di sequenza al pacchetto, saranno sufficienti due numeri (0 e 1). Dovrà poi controllare se gli ACK/NAK sono danneggiati e ricordarsi se il pacchetto corrente ha numero di sequenza 0 o 1.

Il ricevente deve invece controllare se il pacchetto ricevuto è duplicato, lo stato indicherà se il numero di sequenza atteso è 0 o 1. Il ricevente non potrà però sapere se il suo ultimo ACK/NAK è stato ricevuto correttamente dal mittente.

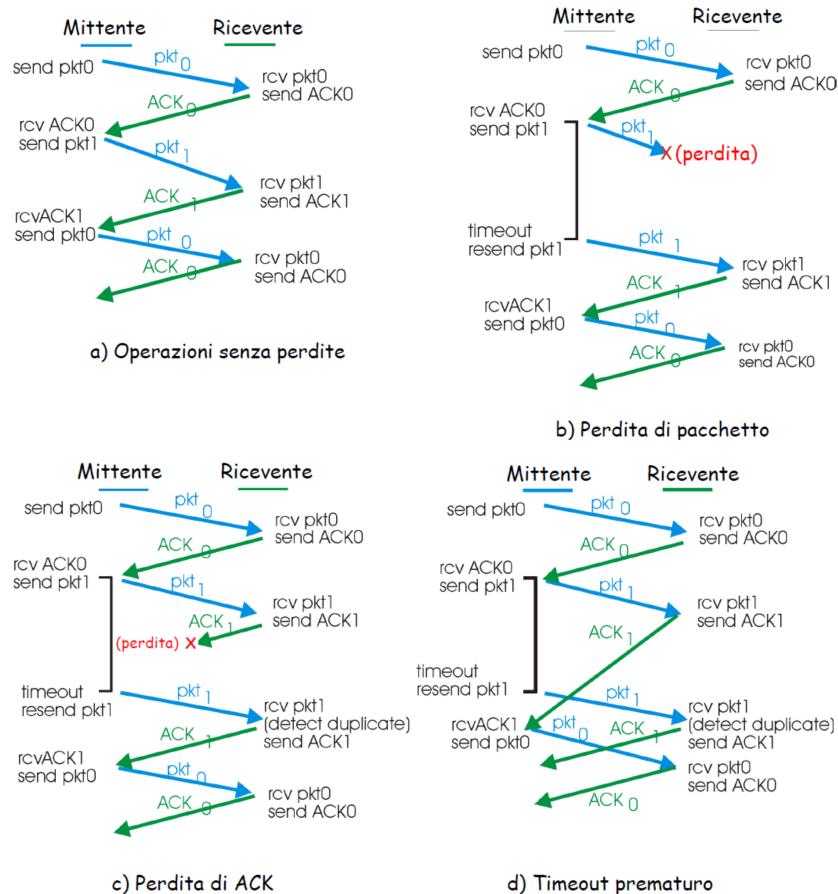
### 3.3.4 Rdt 2.2: un protocollo senza NAK

Ha le stesse funzionalità di Rdt 2.1, utilizzando solamente gli ACK. In sostituzione al NAK il destinatario invierà un ACK per l'ultimo pacchetto ricevuto correttamente, il destinatario invece deve includere esplicitamente il numero di sequenza con l'ACK. Un ACK duplicato presso il mittente determina la stessa azione del NAK, cioè ritrasmettere il pacchetto corrente.

### 3.3.5 Rdt 3.0: canali con errori e perdite

Può succedere che il canale sottostante smarrisca i pacchetti (dati o ACK). Per ovviare a ciò il mittente attende un ACK per un tempo ragionevole, dopodiché, se non ricevuto, ritrasmetterà il pacchetto.

Se il pacchetto (o l'ACK) è solo in ritardo la trasmissione sarà duplicata, ma il problema è già gestita dai numeri di sequenza, che il destinatario specificherà anche nei pacchetti da riscontrare; è necessaria l'introduzione di un contatore.



### 3.3.6 Pipelining

Il mittente ammette più pacchetti in transito ancora da notificare, il loro numero di sequenza deve essere incrementale ed è presente un buffering dei pacchetti presso il mittente e il ricevente. Ci sono due forme generiche di protocolli con pipeline: *Go-Back-N* e *Ripetizione selettiva*.

#### Go-Back-N

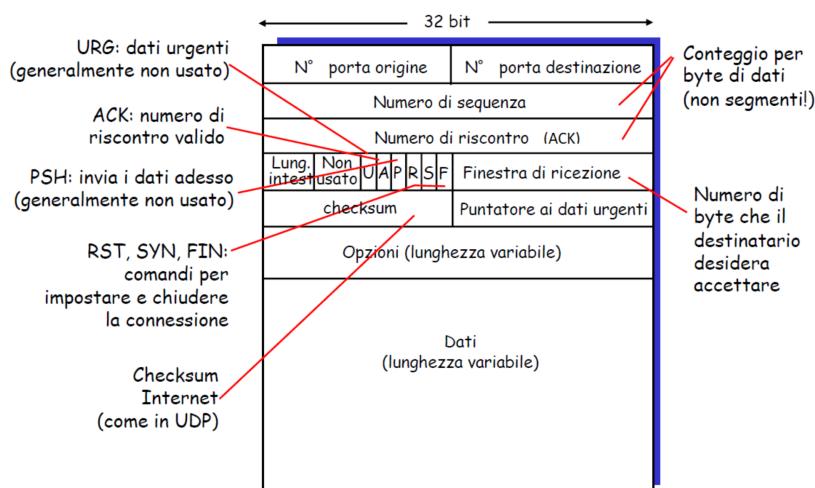
- Il mittente può avere fino a N pacchetti senza ACK in pipeline
- Il ricevente invia solo ACK cumulativi, non dà quindi l'ACK di un pacchetto se c'è un gap
- Il mittente ha un timer per il pacchetto più vecchio senza ACK.

#### Ripetizione selettiva

- Il mittente può avere fino a N pacchetti senza ACK in pipeline
- Il ricevente trasmette gli ACK solo sui singoli pacchetti.
- Il mittente mantiene un timer per ciascun pacchetto che non ha ancora ricevuto un ACK, che alla scadenza farà ritrasmettere solo i pacchetti senza ACK.
- Il ricevente accusa la ricevuta di ciascun singolo pacchetto.

## 3.4 Trasporto orientato alla connessione: TCP

- Implementa l'**Rdt 3.0** con pipelining
- **Connessione punto-punto:** un mittente e un destinatario
- Il flusso di byte è **affidabile** e in sequenza
- Il **controllo di flusso e di congestione** del TCP definiscono la dimensione della finestra di **pipelining**
- **Full-Duplex:** il flusso dei dati è bidirezionale nella stessa connessione e viene definita la dimensione massima del segmento (*MSS*)
- **Orientato alla connessione:** l'handshaking inizializza lo stato di mittente e destinatario prima di scambiare i dati
- **Flusso controllato:** il mittente non sovraccarica il destinatario



Alcuni dettagli sul segmento TCP:

- **Lunghezza dell'intestazione:** serve per sapere se ci sono informazioni nella parte opzionale, in caso negativo sarà di default 20 ( $5 \text{ righe} \cdot 4 \text{ byte} = 20$ )
- **Numero di sequenza:** numero del primo byte del segmento nel flusso di byte
- **ACK:** numero di sequenza del prossimo byte atteso
- Per calcolare il **timeout** si usa una media esponenziale ponderata

Il trasporto TCP crea un servizio di trasferimento dati affidabile sul servizio inaffidabile di IP. Le ritrasmissioni sono avviate da eventi di timeout e ACK duplicati.

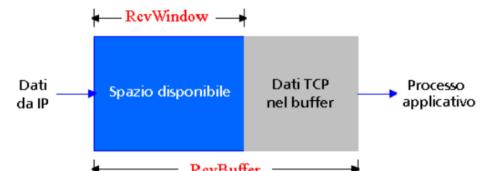
Evento presso il destinatario	Azione del ricevente TCP
Arrivo ordinato di un segmento con numero di sequenza atteso. Tutti i dati fino al numero di sequenza atteso sono già stati riscontrati.	ACK ritardato. Attende fino a 500 ms l'arrivo del prossimo segmento. Se il segmento non arriva, invia un ACK.
Arrivo ordinato di un segmento con numero di sequenza atteso. Un altro segmento è in attesa di trasmissione dell'ACK.	Invia immediatamente un singolo ACK cumulativo, riscontrando entrambi i segmenti ordinati.
Arrivo non ordinato di un segmento con numero di sequenza superiore a quello atteso. Viene rilevato un buco.	Invia immediatamente un ACK duplicato, indicando il numero di sequenza del prossimo byte atteso.
Arrivo di un segmento che colma parzialmente o completamente il buco.	Invia immediatamente un ACK, ammesso che il segmento cominci all'estremità inferiore del buco.

Il timeout spesso è relativamente lungo, si ha quindi un lungo ritardo prima che venga ritrasmesso il pacchetto perduto.

Gli ACK perduti vengono rilevati tramite gli ACK duplicati: il mittente spesso invia molti segmenti, se un segmento viene smarrito è probabile che ci saranno molti ACK duplicati. Se il mittente riceve 3 ACK duplicati per lo stesso dato si suppone che il segmento che segue il dato riscontrato è andato perduto e rispedirà quindi il pacchetto prima che scada il timer (metodo di **trasmissione rapida**).

### 3.4.1 TCP: controllo di flusso

Il lato ricevente della connessione TCP ha un buffer di ricezione, quindi il processo applicativo potrebbe essere rallentato dalla lettura del buffer. Con il controllo di flusso il mittente non vuole sovraccaricare il buffer del destinatario, trasmettendo troppi dati troppo velocemente. Il servizio di corrispondenza delle velocità indica che la frequenza di invio deve corrispondere alla frequenza di lettura dell'applicazione ricevente.



Il destinatario comunica lo spazio disponibile includendo il valore **RcvWindow** nei segmenti, il mittente limita i dati non riscontrati a **RcvWindow** e garantisce che il buffer di ricezione non vada in overflow.

### 3.4.2 Gestione della connessione

La connessione viene gestita mediante un Handshake a tre vie:

1. Il client invia un segmento SYN al server che specifica il numero di sequenza iniziale, non viene inviato nessun dato
2. Il server riceve il SYN e risponde con un segmento SYNACK e successivamente alloca i buffer
3. Il client riceve SYNACK e risponde con un ACK che può anche contenere dati

Per chiudere una connessione:

1. Il client invia un segmento di controllo FIN al server
2. Il server riceve il segmento FIN e risponde con un ACK, chiude la connessione e invia un FIN
3. Il client riceve il FIN e risponde con un ACK
4. Il server riceve l'ACK e chiude la connessione

## 3.5 Principi del controllo di congestione

Per congestione si intende quando troppe sorgenti trasmettono troppi dati a una velocità talmente elevata che la rete non è in grado di gestirli. I sintomi della congestione possono essere pacchetti smarriti (causati da overflow nei buffer dei router) o lunghi ritardi (accodamento nei buffer).

I due principali approcci al controllo di congestione sono:

- **Controllo di congestione punto-punto**
  - Nessun supporto esplicito dalla rete
  - La congestione è dedotta osservando le perdite e i ritardi nei sistemi terminali
  - Metodo adottato da TCP
- **Controllo di congestione assistito dalla rete**
  - I router forniscono un feedback ai sistemi terminali
  - Utilizzato un singolo bit per indicare la congestione
  - Viene comunicata in modo esplicito al mittente la frequenza trasmissiva

## 3.6 Controllo di congestione in TCP (AIMD)

Il controllo di congestione in TCP viene effettuato mediante l'approccio AIMD; ovvero incremento additivo e decremento moltiplicativo.

Consiste nell'aumentare il tasso trasmissivo sondando la rete fino a quando non si verifica una perdita. Secondo l'**incremento adattivo** fa aumentare la CongWin di 1 MSS a ogni RTT in assenza di perdita, mentre secondo il **decremento moltiplicativo** riduce a metà CongWin dopo un evento di perdita. La formula diventa quindi approssimativamente:

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

CongWin è una funzione dinamica della congestione percepita. Il mittente percepisce la congestione dopo un evento di perdita, quindi un timeout o una ricezione di 3 ACK duplicati, e riduce di conseguenza la frequenza di invio (CongWin).

Vengono utilizzati tre meccanismi: AIMD, partenza lenta e reazione agli eventi di timeout.

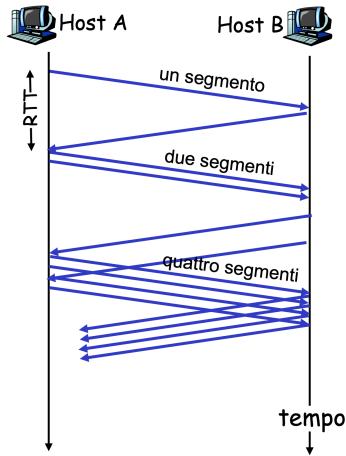
### 3.6.1 Partenza lenta

Quando si stabilisce una connessione ( $CongWin = 1$  MSS) la frequenza aumenta in modo esponenziale fino a quando non si verifica una perdita: infatti  $CongWin$  raddoppia a ogni RTT incrementandolo per ogni ACK ricevuto.

Dopo 3 ACK duplicati (perdita),  $CongWin$  è ridotto a metà e la finestra cresce linearmente. Se si verifica però un timeout,  $CongWin$  viene impostato a 1 MSS crescendo in modo esponenziale fino a una soglia, dopo la quale crescerà linearmente.

La filosofia seguita è quella che 3 ACK duplicati indicano la capacità della rete di consegnare qualche segmento, mentre un timeout prima di 3 ACK duplicati è "più allarmante".

La soglia impostata è variabile, ma in caso di perdita la soglia diventa  $\frac{1}{2}$  di  $CongWin$  appena prima dell'evento.



### 3.6.2 Riassunto: controllo di congestione

- Quando  $CongWin$  è sotto la soglia, il mittente è nella fase di **partenza lenta**; la finestra cresce in modo esponenziale.
- Quando  $CongWin$  è sopra la soglia, il mittente p nella fase di **congestion avoidance**; la finestra cresce in modo lineare.
- Quando si verificano **tre ACK duplicati**, il valore della soglia viene impostato a  $CongWin/2$  e  $CongWin$  viene impostata al valore della soglia.
- Quando si verifica un **timeout**, il valore della soglia viene impostato a  $CongWin/2$  e  $CongWin$  è impostata a 1 MSS.

### 3.6.3 Throughput TCP

Il throughput medio di TCP varia in funzione della dimensione della finestra e di RTT. Se, dopo una perdita, la finestra è  $W$ , il throughput sarà  $W/RTT$ . Dopo la perdita la finestra diventerà  $W/2$  e il throughput di conseguenza diventa  $W/2RTT$ . Il throughput medio è quindi  $0.75 W/RTT$ .

### 3.6.4 Equità di TCP

**Equità** se  $K$  sessioni TCP condividono lo stesso collegamento con ampiezza di banda  $R$  (collo di bottiglia), ogni sessione dovrà avere una frequenza trasmisiva media pari a  $R/K$ .

TCP infatti è equo perché con due connessioni in concorrenza l'incremento additivo ha una pendenza pari a 1, mentre il decremento moltiplicativo riduce il throughput in modo proporzionale.

Proprio per questo le applicazioni multimediali spesso usano UDP, poiché non vogliono che il loro tasso trasmisivo venga ridotto ma tollerano la perdita di pacchetti.

# Capitolo 4

## Il livello di Rete

### 4.1 Introduzione

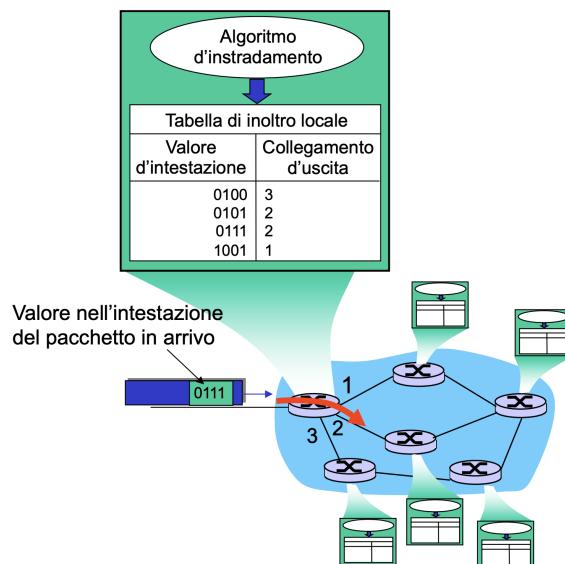
#### 4.1.1 Funzioni chiave del livello di rete

Il livello di rete svolge due funzioni principali:

- **inoltro (o forwarding):** trasferimento dei pacchetti dall'input di un router all'output appropriato
- **instradamento (o routing):** determinare il percorso seguito dai pacchetti da origine a destinazione

Possiamo, per analogia, considerare l'inoltro come l'attraversamento di uno svincolo durante un viaggio e l'instradamento come la pianificazione dell'intero viaggio.

Per effettuare queste funzioni, viene inserito nell'header del pacchetto un'indicazione per l'instradamento, che verrà utilizzata dal router per fare un match con la sua tabella di routing.



Una terza funzione importante del livello di rete è **l'impostazione della connessione**, viene utilizzata però solo in alcune reti (come ATM o X.25). Prima della trasmissione dei dati viene stabilita una connessione virtuale tra i due host.

### 4.1.2 Modelli di servizio

Il livello di rete offre diversi modelli di servizio: possiamo ricordare il servizio **best effort** utilizzato sulla rete internet attuale e i modelli **CBR (Constant)**, **VBR (Variable)**, **ABR (Available)**, **UBR** utilizzati in ATM, per diversi anni utilizzata alternativa a Internet.

## 4.2 Reti a circuito virtuale e datagramma

Le reti a datagramma offrono solo il servizio senza connessione, mentre le reti a **circuito virtuale** offrono il servizio con connessione. Questa è realizzata da host a host e non si può scegliere (il livello di rete non può fornirli entrambi contemporaneamente).

### 4.2.1 Reti a circuito virtuale

Le reti a circuito virtuale (come ATM) sono reti analoghe ai circuiti telefonici che consentono buone prestazioni e vedono il coinvolgimento della rete: il pacchetto ha un numero di VC nell'header utilizzato e sostituito dai router, questo numero consente di effettuare multiplexing.

I router di queste reti mantengono le informazioni sullo stato delle connessioni, infatti ogni router ha una tabella di inoltro con tutti i VC utilizzati.

### Tabella d'inoltro

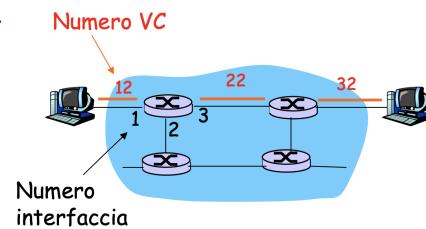


Tabella d'inoltro:

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

### Protocolli di segnalazione

I protocolli di segnalazione, non utilizzati sulla rete Internet, sono dei messaggi utilizzati per gestire la connessione sui VC quindi avvio, mantenimento e chiusura dei circuiti.

### 4.2.2 Reti a datagramma

Nelle reti a datagramma non avviene l'impostazione di chiamata: i dati vengono inviati quando si vuole e quando la rete può li inoltrerà. In aggiunta, la rete è stateless (senza connessione) e i pacchetti vengono inoltrati utilizzando l'indirizzo di destinazione.

Le tabelle di inoltro sono costituite da 4 miliardi di indirizzi possibili (32 bit), queste tabelle spesso sono realizzate ad intervalli. Per semplificazione si preferisce effettuare il matching confrontando i prefissi tra datagram e tabella, nel caso in cui i bit siano uguali si aumenta il numero di bit confrontati.

### 4.2.3 Confronto tra le due tipologie

- Circuiti virtuali (ATM)

- eredita dalla telefonia
- requisiti più stringenti in termini di tempo e affidabilità, utilizzate per servizi garantiti
- i terminali sono stupidi: la complessità è interna alla rete (es. controllo di flusso)

- Datagrammi (internet)

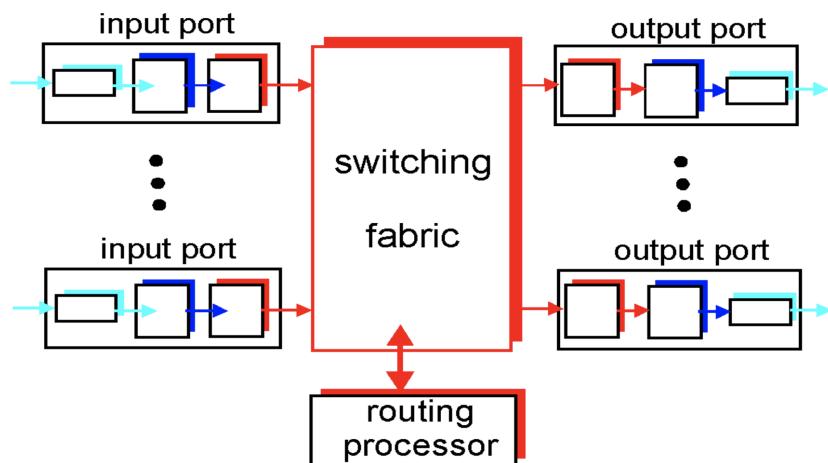
- scambio di dati tra calcolatori
- i servizi sono elasticci: pochi requisiti di tempo
- L'interconnessione è semplice: adattabile, controllo degli errori, rete interna semplice ed ottimizzata alla velocità
- svariati tipi di link, servizio non uniforme

## 4.3 Cosa si trova all'interno dei router?

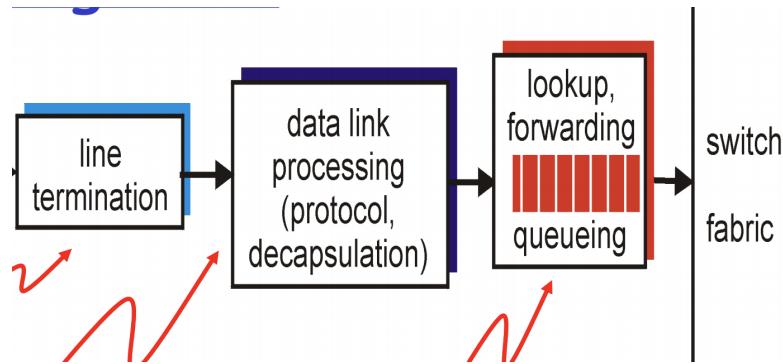
### 4.3.1 Architettura

I router svolgono due funzioni principali:

- far girare i protocolli e gli algoritmi di instradamento (RIP, OSPF, BGP)
- inoltrare i datagrammi dagli input agli output



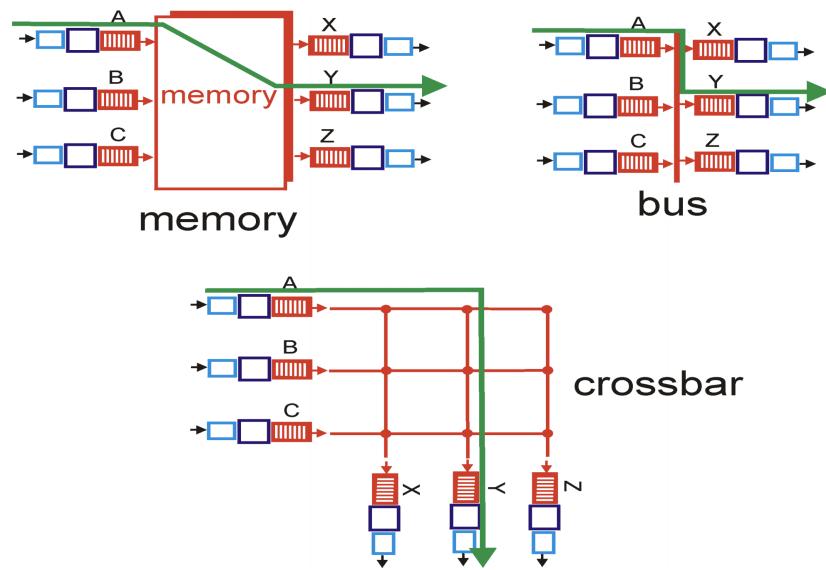
### 4.3.2 Porte d'ingresso



Le porte d'ingresso dei router vengono utilizzate per effettuare una commutazione decentralizzata:

- determinare la porta d'uscita dei pacchetti utilizzando le informazioni della tabella d'inoltro
- obiettivo: completare l'elaborazione allo stesso tasso della linea
- si presenta un **accodamento** se il tasso d'arrivo dei pacchetti è maggiore del tasso d'inoltro

### 4.3.3 Tecniche di commutazione



La tecnica di commutazione **in memoria** è la più semplice ma la meno performante, la tecnica **a crossbar** è invece la più efficace.

#### Commutazione in memoria

La tecnica di commutazione in memoria veniva utilizzata nelle prime generazioni di router, i quali erano tradizionali calcolatori. La commutazione veniva controllata dalla CPU: il pacchetto veniva copiato in memoria e successivamente mandato in uscita con una frequenza totale di **B/2**.

### Commutazione tramite bus

La commutazione tramite bus si basa sull'utilizzo di un bus condiviso tra tutte le porte: siamo in presenza di una **contesa per il bus**, la cui banda limita di conseguenza la banda della commutazione.

Un esempio di router è il Cisco 5600, che opera con un bus da 32 Gbps, sufficiente per reti d'accesso o aziendali.

### Commutazione attraverso rete d'interconnessione

Questa tecnica supera il limite di banda di un bus condiviso. Viene utilizzato un **crossbar switch**, ovvero una rete d'interconnessione di  $2n$  bus che collegano  $n$  porte d'entrata a  $n$  porte d'uscita.

Un esempio come il Cisco 12000 raggiunge i 60 Gbps nella struttura di commutazione.

#### 4.3.4 Porte d'uscita

Le porte d'uscita implementano le funzioni di accodamento (se la frequenza di pacchetti in arrivo è superiore a quella del collegamento uscente) e di schedulatore di pacchetti (stabilire l'ordine di trasmissione dei pacchetti).

#### 4.3.5 Quale deve essere la capacità dei buffer?

Secondo la regola spannometrica della RFC 3439 la capacità deve essere

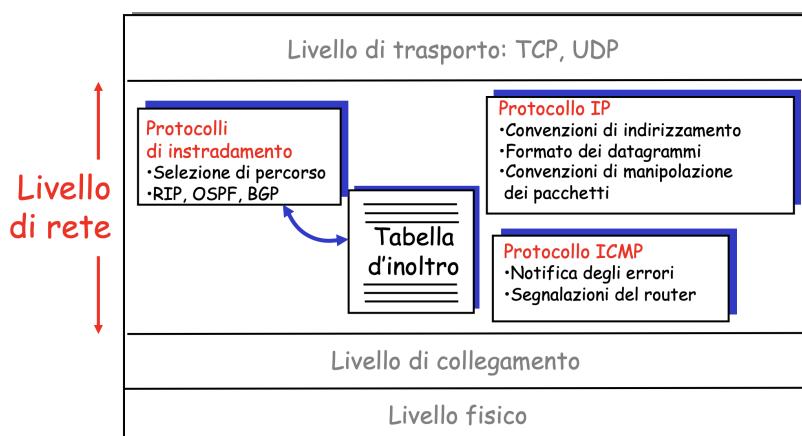
$$\text{mediaRTT} * \text{capacitaC}$$

dove C è la capacità del collegamento.

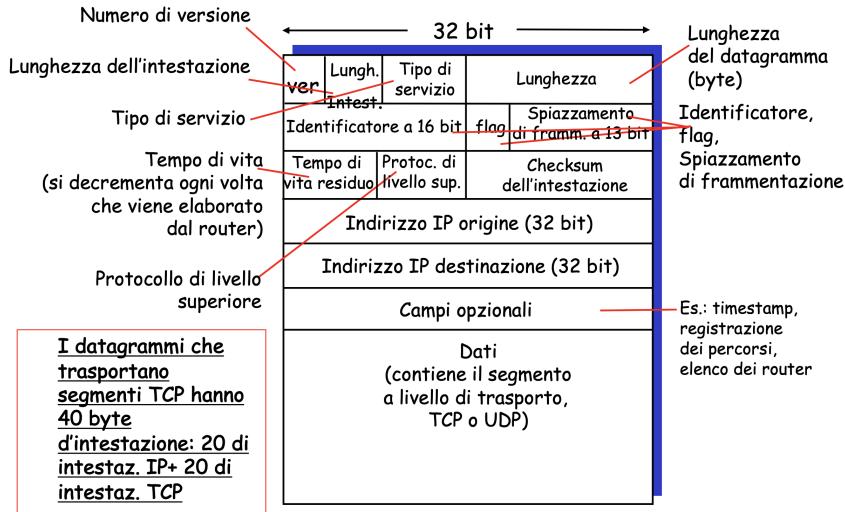
Secondo attuali raccomandazioni, dati N flussi la capacità dei buffer deve essere

$$\frac{\text{RTT} * C}{\sqrt{N}}$$

## 4.4 Protocollo Internet (IP)



#### 4.4.1 Formato dei datagrammi



I datagrammi IP utilizzano 16 byte, hanno quindi una dimensione massima di 64 KB. Il campo del protocollo di livello superiore indica il protocollo utilizzato dal pacchetto trasmesso, quindi TCP/UDP. Avendo a disposizione 32 bit per gli indirizzi abbiamo in totale poco più di 4 miliardi di indirizzi.

#### Frammentazione dei datagrammi

La frammentazione dei datagrammi è una funzione importante poiché consente il trasporto da parte del livello sottostante (collegamento) di datagrammi più grandi di quanto ne potrebbe portare.

Se la **MTU** (Maximum Transmission Unit, quantità massima di dati che un frame a livello di collegamento può portare) è più bassa della dimensione del datagramma IP, il livello 3 frammenta i datagrammi troppo grossi in altri datagrammi, che verranno riassemblati solo a destinazione. Ogni router della rete può frammentare in base alle sue esigenze.

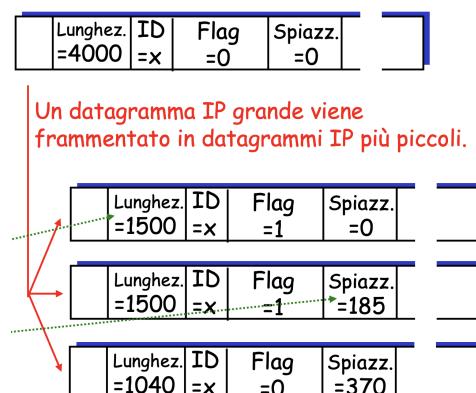
Quando viene effettuata la frammentazione viene settato a 1 il flag relativo nell'header di ogni pacchetto inviato (tranne l'ultimo) e in ognuno viene indicato lo spiazzamento (utilizzato per riposizionare il datagramma nel giusto payload e in posizione corretta).

Il destinatario ricostruirà il datagramma originale riconoscendo gli ID comuni, sapendo che se lo spiazzamento è uguale a 0 sarà il primo, mentre se il flag sarà uguale a 0 sarà l'ultimo.

#### 4.4.2 Indirizzamento IPv4

L'indirizzamento IPv4 consente il posizionamento dei dispositivi nella rete globale: infatti, ogni **interfaccia** di ciascun host collegato su Internet ha un IP globalmente univoco. Questo indirizzo è costituito da 32 bit.

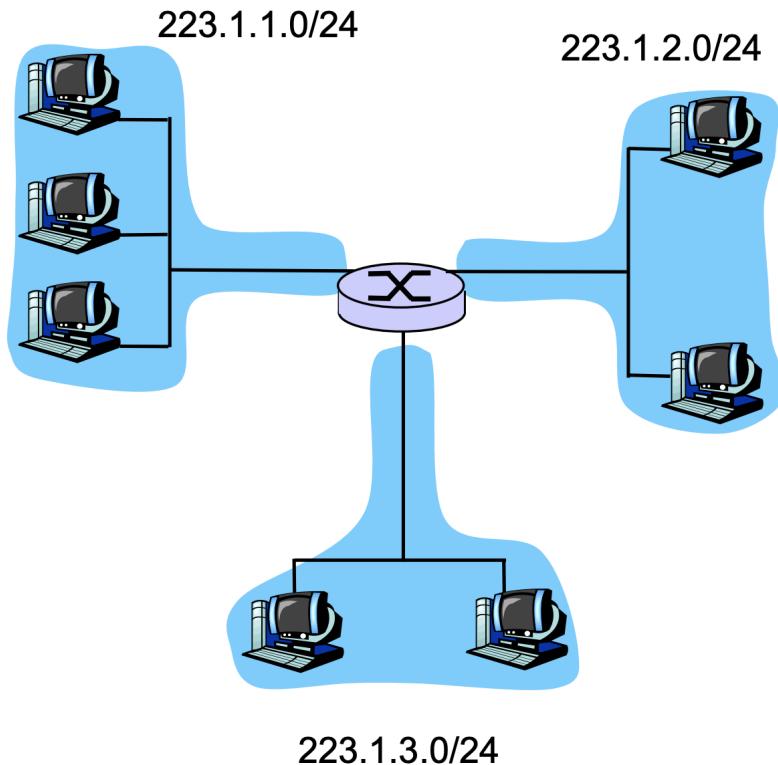
L'interfaccia è il confine tra host e collegamento fisico: i router devono avere almeno due interfacce (2 indirizzi IP) mentre gli host, in genere, hanno una sola interfaccia.



#### Sottoreti

Le sottoreti consentono di dividere l'indirizzamento in Internet in parti a sé stanti. Per dare una definizione appropriata possiamo considerare le sottoreti come delle reti isolate i cui punti terminali sono collegati all'interfaccia di un host o un router.

Questo è possibile suddividendo gli indirizzi IP in due parti: una parte di **sottorete**, condivisa fino a un certo punto, e una parte di **host**.



## Maschera di sottorete: /24

La **maschera di rete** viene utilizzata per indicare quanti bit da sinistra contengono il prefisso condiviso tra tutti gli host della sottorete.

**CIDR** CIDR, o (Classless Interdomain Routing), è una strategia di assegnazione degli indirizzi che definisce il prefisso comune agli apparati in una rete.

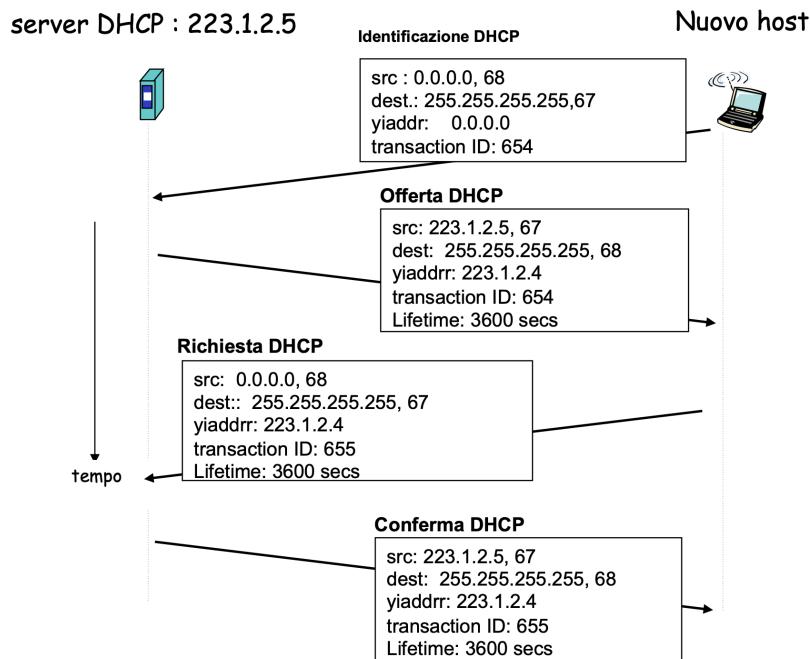
Ha una struttura del tipo  $a.b.c.d/x$  dove x indica il numero di bit della prima parte dell'indirizzo (parte subnet).

←	Parte di sottorete	→	Parte host →
11001000 00010111 00010000 00000000			
<b>Indirizzi della sottorete: 200.23.16.0/23</b>			
<b>Netmask della sottorete: 255.255.254.0</b>			

La strategia CIDR facilita ai router la costruzione delle tabelle di inoltro e consente a un PC di sapere se un host si trova nella sua stessa sottorete (facendo l'AND tra indirizzo e maschera di rete troverà il prefisso che confronterà).

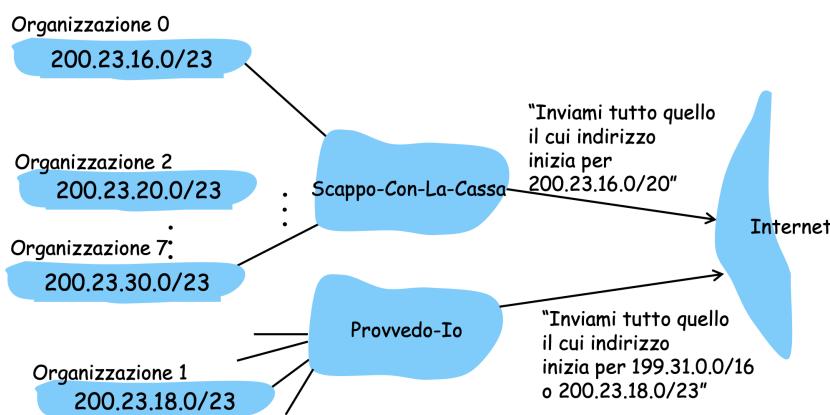
### Assegnazione degli indirizzi

L'assegnazione degli indirizzi può avvenire in modalità manuale o mediante **DHCP**. Il DHCP (Dynamic Host Configuration Protocol) consente di ottenere dinamicamente un IP da un server (plug-and-play), l'IP avrà una scadenza e potrà essere rinnovato, consente il riuso e il risparmio di indirizzi.



L'amministratore di rete o l'ISP ha a disposizione un blocco di IP da utilizzare nelle subnet: se sono privati non si pone alcun problema, mentre se sono pubblici questi dovranno essere comprati da altri ISP. A sua volta, un ISP o un amministratore può dividere il blocco a sua disposizione in altri sottoblocchi contigui.

### Indirizzamento gerarchico



**Indirizzi IP alla fonte** Come fa un ISP a ottenere indirizzi IP? Si rivolge all'ICANN (Internet Corporation for Assigned Names and Numbers), ente che si occupa di allocare i blocchi di indirizzi, gestire i server DNS radice, assegnare e risolvere dispute sui domini.

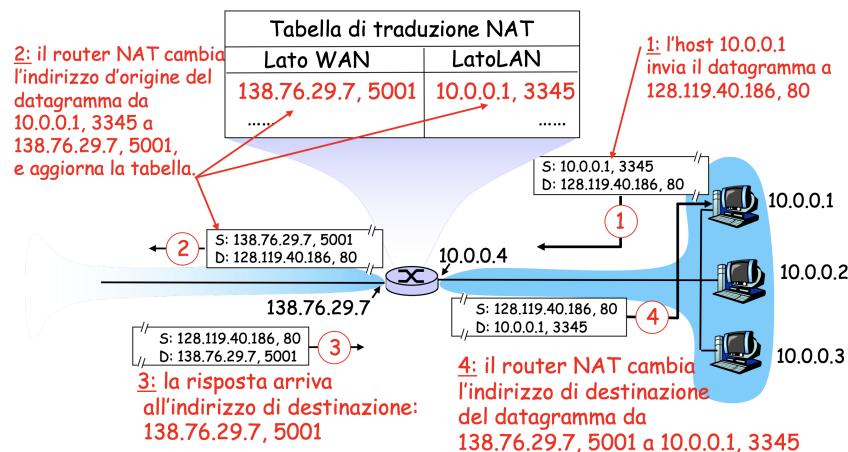
### Traduzione degli indirizzi di rete (NAT)

La traduzione degli indirizzi di rete consente ai router di apparire a Internet con un unico indirizzo IP, quindi tutto il traffico della sottorete avrà lo stesso indirizzo. Si utilizzano all'interno classi di IP private come  $10.0.0.0/8$  e  $192.168.0.0/16$ .

Utilizzando il NAT il router abilitato nasconde i dettagli della rete domestica al mondo esterno, portando alcuni vantaggi:

- non serve allocare intervalli di indirizzi da un ISP
- è possibile riconfigurare la rete privata senza comunicarla a Internet
- è possibile cambiare ISP senza modificare la configurazione della rete privata
- i dispositivi interni non sono indirizzabili e visibili dall'esterno (sicurezza)

Per implementare il NAT il router, all'arrivo di un datagramma, genera una nuova porta d'origine e sostituisce l'IP di origine con il proprio IP sul lato WAN e la porta di origine iniziale con il nuovo numero.

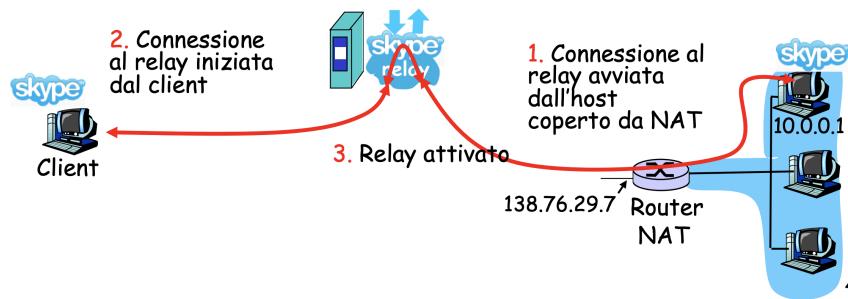


Il numero di porta è costituito da 16 bit, quindi sono possibili più di 60000 connessioni simultanee. NAT è contestato per diversi motivi:

- i router dovrebbero lavorare solo a livello 3+
- viola il punto-punto, causando interferenze a P2P (serve una specifica configurazione)
- si dovrebbe usare alternativamente IPv6

**Collegamenti dall'esterno** Un problema importante del NAT è l'impossibilità di effettuare collegamenti dall'esterno a un host interno alla rete (come un server web). Per effettuare questo si possono implementare alcune soluzioni

- impostare delle configurazioni statiche per inoltrare le richieste entranti a determinate porte dell'host (tabelle di forwarding)
- utilizzare **UPnP** (Universal Plug n Play), parte integrante di IGD (Internet Gateway Device protocol), che consente agli host nascosti da un NAT di chiedere in automatico di scrivere una riga nella tabella di forwarding
- **relay** (utilizzato da Skype), prevede un punto di riferimento a cui entrambi i client si collegano



#### 4.4.3 ICMP

ICMP (Internet Control Message Protocol) consente lo scambio di informazioni relative al controllo della rete, come errori o ping. ICMP è considerato parte di IP, i messaggi hanno un campo tipo e un campo codice mentre l'intestazione e i primi 8 byte sono uguali al datagram IP.

##### Traceroute e ICMP

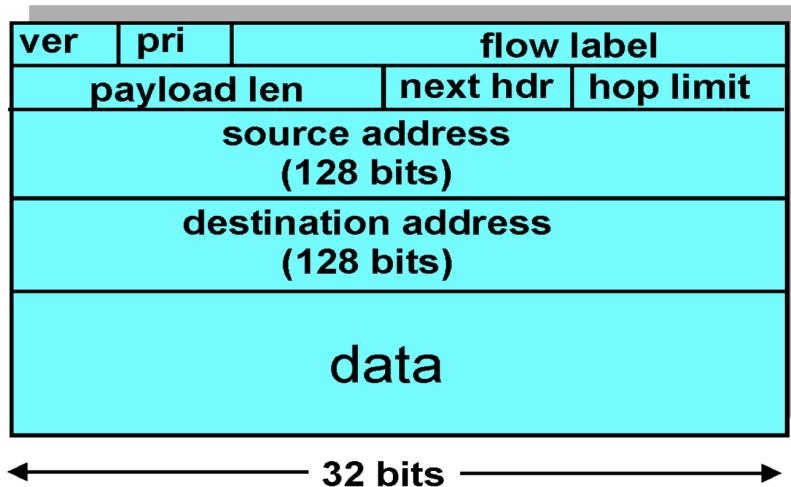
Il programma invia più datagrammi con un TTL incrementale, ogni router scarterà il datagram e invierà all'origine un'allerta ICMP la quale calcolerà il RTT. Ogni router viene calcolato per 3 volte per avere una media e il programma si fermerà una volta arrivato a destinazione, ovvero quando l'host invierà un pacchetto ICMP di porta non raggiungibile.

#### 4.4.4 IPv6

L'esigenza principale che ha portato allo sviluppo di IPv6 è stato lo spazio di indirizzamento IP a 32 bit che stava cominciando ad esaurirsi. Oltre a questo, altre motivazioni erano un header più leggero per velocizzare elaborazione e inoltro, e un agevolazione del QoS.

##### Formato dei datagrammi

I datagram IP sono costituiti da 40 byte di header a lunghezza fissa e non è consentita la frammentazione.

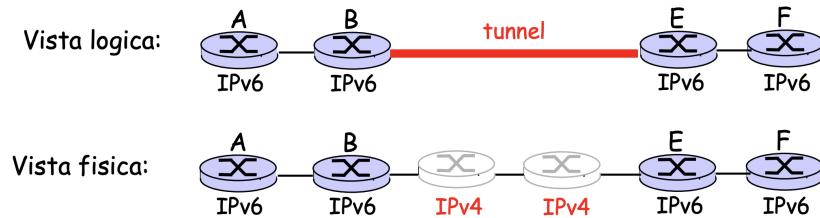


Alcuni campi nuovi sono la **priorità di flusso**, l'**etichetta di flusso** che identifica flussi particolari (non è chiaro il concetto di flusso) e l'**intestazione successiva** che identifica il protocollo di destinazione dei contenuti.

Inoltre, è stato rimosso il checksum poiché ridondante, il campo opzioni non è più parte dell'intestazione ma può venire indicato in "intestazioni successive" ed è stato introdotto **ICMPv6** con nuovi codici che assume le funzionalità di IGMP (gestisce l'ingresso e l'uscita di host dai gruppi multicast).

### Passaggio a IPv6

Non è possibile aggiornare simultaneamente tutti i router, poiché servirebbe stabilire una giornata "di passaggio", attualmente impossibile. Per questo si utilizza il **tunneling**: IPv6 viene trasportato come payload in datagram IPv4.



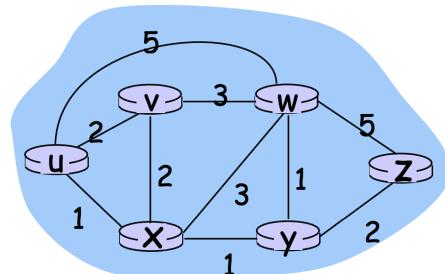
## 4.5 Algoritmi di instradamento

Gli algoritmi di instradamento sono implementati nelle reti a commutazione di pacchetto, grazie all'inserimento di un valore nell'header di livello 3, che verrà letto e confrontato con le varie tabelle di instradamento.

Possiamo assumere una rete come un grafo dove i nodi sono i router e gli archi sono i collegamenti. Gli algoritmi di instradamento si occupano di calcolare il cammino a costo minimo tra due router.

Questi algoritmi possono essere classificati in più modalità: globali o decentralizzati e statici o dinamici:

- un algoritmo **globale** riceve in ingresso tutti i collegamenti tra nodi e loro costi, un esempio sono gli algoritmi **link-state**
- in un algoritmo **decentralizzato** ogni nodo elabora un vettore di stima dei costi verso tutti gli altri nodi, quindi il cammino a costo minimo viene calcolato in modo distribuito e iterativo. Un esempio sono gli algoritmi **distance-vector**
- in un algoritmo **statico** i cammini cambiano molto raramente
- gli algoritmi **dinamici** determinano gli instradamenti in base al traffico e alla topologia della rete



### 4.5.1 Stato del collegamento (link state)

Un esempio di algoritmo link state è l'**algoritmo di Dijkstra** il quale prevede che la topologia di rete e i costi siano noti a tutti i nodi mediante il "link-state broadcast" e che tutti i nodi dispongano delle stesse informazioni.

L'algoritmo calcola il cammino a costo minimo dall'origine a tutti gli altri nodi, creando una **tabella d'inoltro** per quel nodo. Essendo iterativo, dopo la k-esima iterazione i cammini a costo minimo sono noti a k nodi di destinazione.

I costi di ogni collegamento possono essere:

- predefiniti dal provider
- tutti uguali
- costi effettivi (satellite)
- definiti dal ritardo di collegamento

La regola è evitare che i costi dipendano dal routing.

La sua complessità con n nodi è data da

$$O(n \log n)$$

Può presentare delle oscillazioni ad esempio nel costo del collegamento in base alla quantità di traffico trasportato.

#### 4.5.2 Algoritmo con vettore distanza

Un esempio di algoritmo con vettore distanza è la **formula di Bellman-Ford** (o a programmazione dinamica) che definisce il costo del percorso a costo minimo da x a y

$$d_x(y) = \min_v \{c(x, v) + d_v(y)\}$$

dove  $\min_v$  riguarda tutti i vicini di x,  $c(x, v)$  comprende il costo di tutti i collegamenti diretti da x a v e  $d_v(y)$  è il percorso a costo minimo da v a y.

L'algoritmo viene eseguito nel modo seguente: il nodo x contatta tutti i vicini v e si fa dare da ognuno di essi ogni costo per andare a y, successivamente valuterà tutte le alternative.

L'idea di base è che ogni nodo invia una copia del proprio vettore distanza a ogni vicino, quando un nodo riceve un nuovo vettore distanza da un vicinolo salva e usa la formula B-F per aggiornare il proprio DV. Finché tutti i nodi continuano a cambiare i propri DV in maniera asincrona, ciascuna stima  $D_x(y)$  converge a  $d_x(y)$ .

L'algoritmo con vettore distanza è **iterativo ed asincrono**, poiché ogni iterazione locale è causata dal cambio del costo di un link locale o dalla ricezione di un DV aggiornato; è **distribuito** poiché ogni nodo aggiorna i suoi vicini solo quando il suo DV cambia.

#### Modifica dei costi

Quando un nodo rileva un cambiamento nel costo dei collegamenti allora aggiorna il proprio vettore distanza e lo trasmetterà ai suoi vicini, secondo il principio che "le buone notizie viaggiano in fretta".

#### Confronto tra algoritmi LS e DV

- Complessità dei messaggi
  - LS con n nodi, E collegamenti implica l'invio di  $O(nE)$  messaggi
  - DV richiede scambi tra nodi adiacenti, quindi il tempo di convergenza può variare
- Velocità di convergenza
  - LS: l'algoritmo  $O(n^2)$  richiede  $O(nE)$  messaggi, ci possono essere oscillazioni di velocità

- DV può convergere lentamente, può presentare cicli d'instradamento e può presentare il problema del conteggio all'infinito
- **Robustezza:** cosa avviene se un router funziona male
  - LS: un router può comunicare via broadcast un costo sbagliato per uno dei suoi collegamenti (non per altri), i nodi si occupano di calcolare solo le proprie tabelle
  - DV: un nodo può comunicare cammini a costo minimo errati a tutte le destinazioni, la tabella di ogni nodo può essere usata da altri quindi un calcolo errato si diffonde per l'intera rete

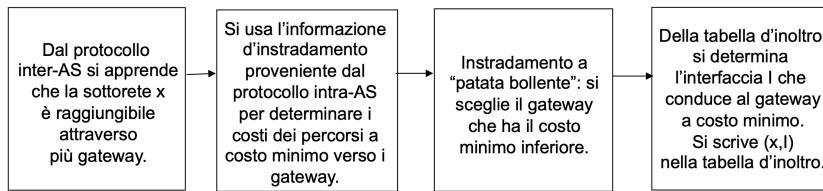
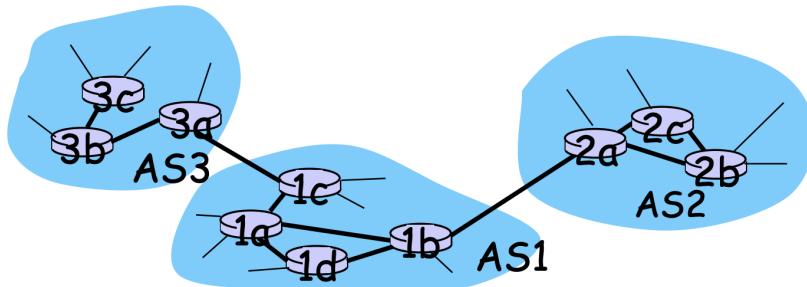
#### 4.5.3 Instradamento gerarchico

Fino ad ora abbiamo visto la rete come un insieme di router interconnessi, con una visione omogenea, ma nella pratica le cose non sono così semplici. Nella realtà ci sono 200 milioni di destinazioni, quindi archiviare tutte le informazioni di instradamento su ogni host sarebbe impossibile data l'enorme quantità di memoria necessaria e l'elevato traffico (bloccherebbe il resto) che si creerebbe.

Per questo, conviene impostare la rete Internet con una **autonomia amministrativa**, secondo la quale idealmente ciascuno sarebbe in grado di amministrare la propria rete connettendola alle altre.

Nella realtà è possibile organizzare i router in **sistemi autonomi** (AS), dove in ogni gruppo autonomo i router eseguono lo stesso algoritmo di instradamento (protocollo **intra-AS**) mentre i **router gateway** hanno il compito di inoltrare i pacchetti a destinazioni esterne.

Ogni sistema autonomo sa come inoltrare i pacchetti lungo il percorso ottimo verso qualsiasi destinazione interna al gruppo, mentre per trasferire dati tra sistemi autonomi differenti (i quali potrebbero usare protocolli differenti) si utilizza l'instradamento **inter-AS**.



#### 4.6 Instradamento in Internet

I protocolli d'instradamento *intra-AS* sono noti come protocolli gateway interni (**IGP**), i più comuni sono:

- **RIP:** Routing Information Protocol
- **OSPF:** Open Shortest Path First
- **IGRP:** Interior Gateway Routing Protocol (proprietario Cisco)

### 4.6.1 RIP (Routing Information Protocol)

RIP è un protocollo a vettore distanza incluso in UNIX BSD dagli anni '80. Effettua un conteggio degli hop come metrica di costo con un massimo di 15 hop.

I router adiacenti si scambiano aggiornamenti ogni 30 secondi mediante l'annuncio RIP (*RIP advertisement*), contenente un elenco di fino a 25 sottoreti di destinazione interne all'AS, insieme alla distanza tra il mittente ed esse.

Se un router non riceve notizie dal vicino per più di 180 secondi il nodo viene considerato spento, con il ricalcolo della tabella e la propagazione dell'informazione agli altri vicini. L'utilizzo dell'*inversione avvelenata* evita i loop (16 hop).

RIP viene implementato a livello applicazione con messaggi su socket standard e protocollo di trasporto standard.

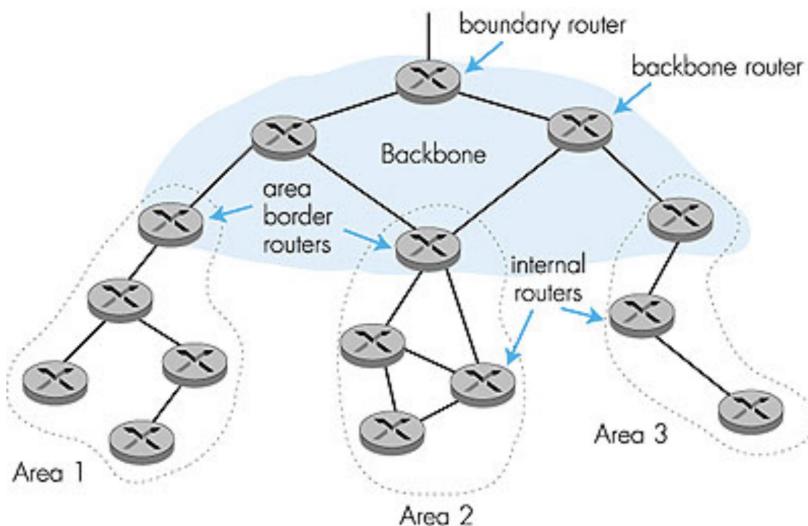
### 4.6.2 OSPF (Open Shortest Path First)

Essendo open, le specifiche del protocollo sono pubblicamente disponibili. Protocollo a link-state, utilizza il flooding di informazioni di stato del link e l'algoritmo di Dijkstra per determinare il percorso a costo minimo. Ogni volta che si verifica un cambiamento su un link il router inoltra l'informazione a tutti i router (all'intero AS) utilizzando il flooding, i messaggi sono trasportati da IP.

OSPF presenta alcuni vantaggi rispetto a RIP:

- **sicurezza**: scambi tra router autenticati
- **multipath**: consente l'utilizzo di più percorsi con uguale costo
- per ogni link possono esserci più costi in base al servizio (es. satellite costo elevato)
- **supporto unicast e multicast**: viene utilizzato OSPF Multicast
- **supporto alle gerarchie** in un dominio d'instradamento

#### OSPF strutturato gerarchicamente



La struttura gerarchica in OSPF consente di impostare due livelli: area locale e dorsale, nelle quali i messaggi LS sono solo all'interno dell'area e ogni nodo conosce la direzione verso le reti nelle altre aree.

I **router di confine d'area** appartengono sia alla dorsale che a un'area generica, i **router di dorsale** effettuano l'instradamento interno alla dorsale, i **router di confine** scambiano informazioni con router di altri AS.

### 4.6.3 BGP: instradamento inter-AS

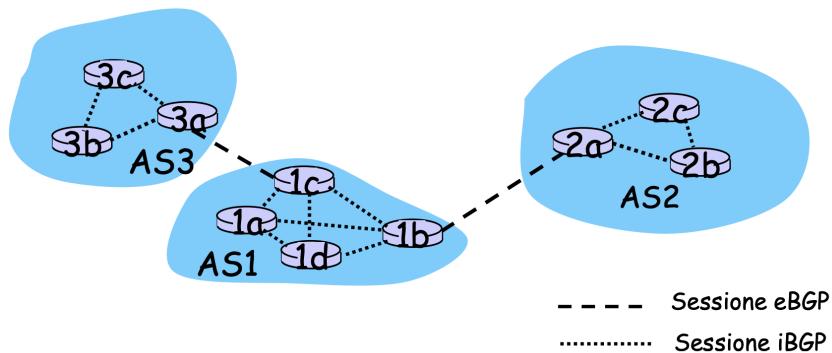
Il protocollo BGP (Border Gateway Protocol) è l'attuale standard *de facto* per l'instradamento inter-AS, che mette a disposizione di ciascun AS le seguenti funzionalità:

1. ottenere informazioni sulla raggiungibilità delle sottoreti da parte di AS confinanti
2. propagare le informazioni di raggiungibilità a tutti i router interni di un AS
3. determinare percorsi "buoni" verso le sottoreti sulla base delle informazioni di raggiungibilità e delle politiche dell'AS

In breve, BGP consente alle sottoreti di comunicare la propria esistenza alla rete Internet.

#### Fondamenti di BGP

Due router che si scambiano messaggi BGP sono chiamati **peer BGP** mentre la connessione TCP è detta **sessione BGP**. Quando un AS annuncia un prefisso ad un altro AS, sta in realtà "promettendo" di inoltrare i datagrammi sul prefisso stabilito, un AS può aggregare più prefissi in un annuncio.

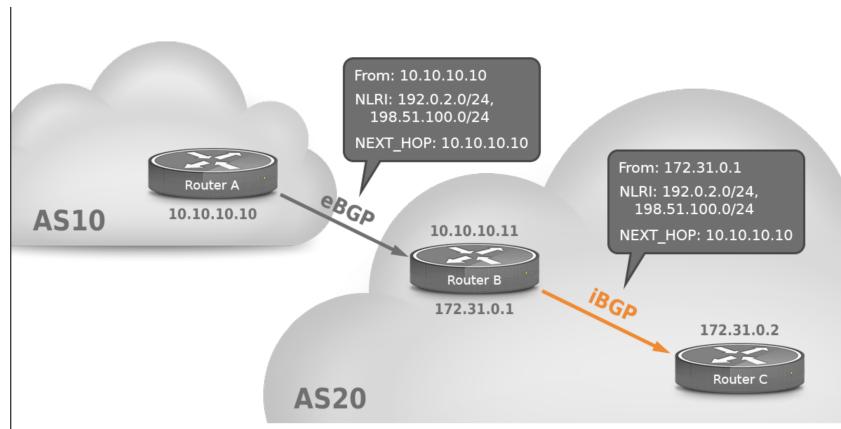


Le sessioni BGP interne sono utilizzate per distribuire i prefissi a tutti i router del AS, mentre le sessioni esterne scambiano informazioni sulla raggiungibilità dei prefissi.

**Attributi del percorso** Quando viene annunciato un prefisso, nel messaggio vengono aggiunti anche degli attributi BGP come:

- **AS-PATH** che elenca gli AS che ha attraversato l'annuncio
- **NEXT-HOP** che indica l'eventuale collegamento fisico su cui viene inoltrato il pacchetto

Ogni router gateway ha delle **politiche di importazione** per decidere se accettare o filtrare la rottura.



**Selezione dei percorsi** Nel caso in cui siano presenti più rotte si seguono alcune regole di eliminazione:

1. si preferiscono le rotte con dei valori di preferenza locale più alti
2. si seleziona la rota con AS-PATH più breve
3. si seleziona la rota con il router NEXT-HOP più vicino, instradamento a *patata bollente*
4. se avanzano più rotte, ci si basa sugli identificatori BGP

### Messaggi BGP

- **OPEN**: apre la connessione TCP e autentica il mittente
- **UPDATE**: annuncia il nuovo percorso
- **KEEPALIVE**: mantiene la connessione attiva in mancanza di UPDATE
- **NOTIFICATION**: riporta gli errori del precedente messaggio; usato anche per chiudere il collegamento.

### Differenze tra i protocolli inter-AS e intra-AS

- **Politiche**
  - Inter-AS: l'amministrazione vuole controllare l'instradamento del traffico e chi instrada attraverso le sue reti
  - Intra-AS: un solo controllo amministrativo, rotte interne scelta senza questioni di politica importanti
- **Scala**: l'instradamento gerarchico fa risparmiare sulle tabelle d'instradamento riducendo il traffico di aggiornamento
- **Prestazioni**: Intra-AS orientato alle prestazioni, Inter-AS le politiche possono prevalere sulle prestazioni

# Capitolo 5

## Il livello di collegamento

### 5.1 Livello di collegamento: introduzione e servizi

A livello di collegamento definiamo come **nodi** host e router, che vengono collegati tra loro da **collegamenti (link)** di tipologia cablata, wireless o LAN. Le unità di dati scambiate a livello link sono chiamate **frame**.

In breve, i protocolli a livello di collegamento si occupano del trasporto di datagram lungo un singolo canale di comunicazione. Questi protocolli possono essere differenti sui vari collegamenti che seguirà il datagram e i servizi erogati possono essere differenti: non tutti i protocolli, ad esempio, forniscono consegna affidabile.

#### 5.1.1 Servizi offerti a livello di link

- **Framing**
  - I protocolli encapsulano i datagram del livello di rete in frame a livello di link
  - Il protocollo MAC controlla l'accesso al mezzo trasmissivo (il collegamento)
  - Viene utilizzato l'indirizzo MAC (diverso dall'IP) per identificare i nodi di origine e destinazione
- **Consegna affidabile**
  - Non necessaria sui collegamenti a basso numero di errori sui bit (fibra ottica, cavo coassiale, doppino intrecciato)
  - Utilizzata nei collegamenti soggetti a elevato tasso di errori (wireless)
- **Controllo di flusso:** non saturare il nodo ricevente
- **Rilevazione degli errori:** causati dall'attenuazione del segnale e dal rumore elettromagnetico, il ricevente individua gli errori grazie a un bit di controllo inserito nel frame
- **Correzione degli errori:** il ricevente determina l'errore e lo corregge
- **Half-duplex e full-duplex**

### 5.1.2 Dov'è implementato il livello link

Viene implementato in tutti gli host grazie ad un adattatore (o **NIC**, network interface card) che implementa il livello di collegamento e fisico ed è una combinazione di hardware, software e firmware.

L'adattatore ha il ruolo in trasmissione di incapsulare i datagram nei frame impostando bit di controllo errori, trasferimento affidabile, controllo di flusso, ecc e dall'altro lato di individuare errori ed estrarre i datagram passandoli al nodo.

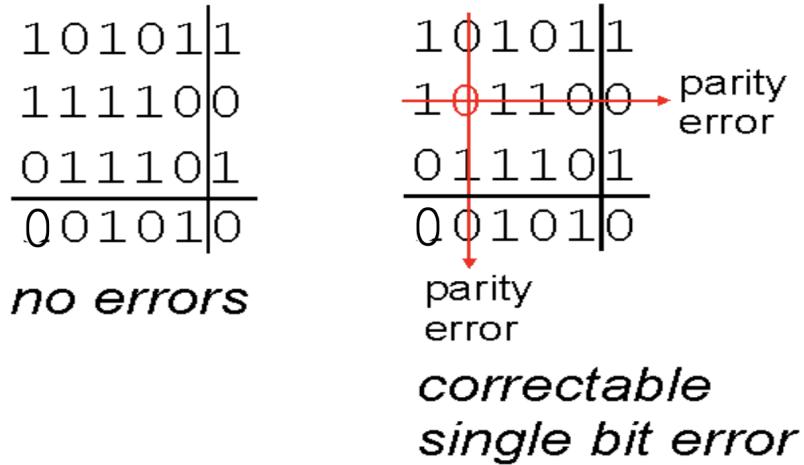
## 5.2 Tecniche di rilevazione e correzione degli errori

La rilevazione degli errori non è attendibile al 100%, infatti è possibile che ci siano errori non rilevati e per ridurre che questo accada le tecniche più sofisticate prevedono un'elevata ridondanza.

### 5.2.1 Controllo di parità

Nel caso di **unico bit di parità** è presente un solo bit che consente di riconoscere che si è verificato almeno un errore in un bit.

Nel caso della **parità bidimensionale** è possibile individuare e correggere il bit alterato.



## 5.3 Protocolli di accesso multiplo

Esistono due tipi di collegamenti di rete:

- collegamento **punto-punto** (PPP), utilizzato per connessioni telefoniche o collegamenti punto-punto tra Ethernet e host
- collegamento **broadcast** (cavo o canale condiviso) come l'Ethernet tradizionale o wireless

Nel caso di una connessione a un canale broadcast condiviso è consentita la connessione anche a migliaia di nodi: si genera quindi una **collisione** quando i nodi ricevono più di un frame contemporaneamente.

I protocolli ad accesso multiplo consentono quindi di definire le modalità con cui i nodi regolano le loro trasmissioni sul canale, la cui comunicazione utilizza il canale stesso (non è presente un canale fuori banda).

Il protocollo di accesso multiplo ideale sarebbe un protocollo decentralizzato e semplice che suddivide il tasso trasmissivo tra il numero di nodi che devono inviare dati. Ovviamemente tale protocollo non è possibile

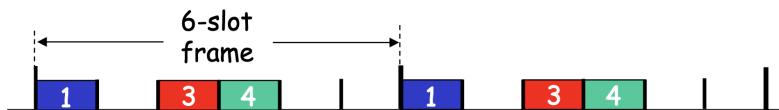
I protocolli di accesso multiplo esistenti vengono classificati come segue:

- Protocolli a **suddivisione del canale**: il canale è suddiviso in parti (slot di tempo, frequenza, codice) ed allocate a uno specifico nodo per utilizzo esclusivo
- Protocolli ad **accesso casuale**: nessuna divisione, si possono verificare collisioni, i nodi ritrasmettono ripetutamente i pacchetti
- Protocolli a **rotazione**: ogni nodo ha il suo turno di trasmissione, ma quelli che hanno molto da trasmettere potrebbero avere turni più lunghi

### 5.3.1 Protocolli a suddivisione del canale

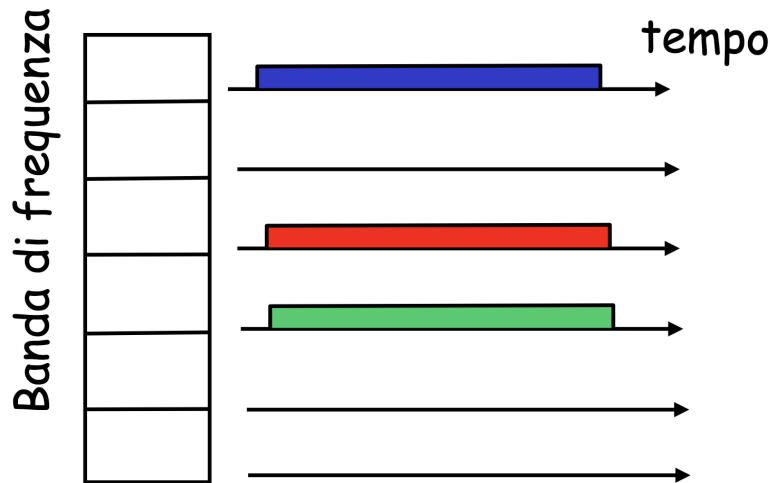
#### TDMA: accesso multiplo a divisione di tempo

Il protocollo TDMA consiste in turni per accedere al canale, suddividendolo in intervalli di tempo. Gli slot non usati rimangono inattivi.



#### FDMA: accesso multiplo a divisione di frequenza

Il protocollo FDMA suddivide il canale in bande di frequenza e ad ogni nodo viene assegnata una banda di frequenza prefissata.



### 5.3.2 Protocolli ad accesso casuale

Nei protocolli ad accesso casuale ogni nodo che deve inviare dati trasmette alla massima velocità consentita dal canale, senza coordinazione tra i nodi. Se più nodi stanno trasmettendo si verifica una collisione. Il protocollo definisce quindi come rilevare le collisioni e come ritrasmettere in caso di avvenuta collisione.

Alcuni protocolli ad accesso casuale sono slotted ALOHA, ALOHA, CSMA, CSMA/CD, CSMA/CA.

### Slotted ALOHA

Nel protocollo slotted ALOHA si assume che tutti i pacchetti abbiano la stessa dimensione e il tempo sia suddiviso in slot, equivalenti al tempo di trasmissione di un singolo pacchetto.

Quando un nodo deve spedire, esso attenderà fino all'inizio dello slot successivo. Se nel frattempo non si verifica una collisione il nodo potrà trasmettere il pacchetto nello slot successivo, altrimenti se avviene la collisione, essa verrà rilevata prima della fine dello slot e ritrasmetterà il pacchetto con probabilità  $p$  durante gli slot successivi.

Questo protocollo consente ai nodi di trasmettere continuamente alla massima velocità decidendo indipendentemente quando ritrasmettere (decentralizzazione). Dall'altro lato alcuni slot presenteranno collisioni andando sprecati ed altri rimarranno vuoti (inattivi)

### Efficienza di Slotted ALOHA

**Efficienza** definita come la frazione di slot vincenti in presenza di un elevato numero di nodi attivi. Nel caso migliore solo il 37% degli slot compie lavoro utile.

### ALOHA Puro

ALOHA puro è più semplice e non sincronizzato: quando arriva il primo pacchetto lo trasmette immediatamente e integralmente nel canale broadcast. Ci sono però elevate probabilità di collisione (i pacchetti si sovrappongono tra loro).

L'efficienza di ALOHA puro (18%) è peggio dello slotted.

### Accesso multiplo a rilevazione della portante (CSMA)

CSMA si pone in ascolto prima di trasmettere: se il canale è libero trasmette l'intero pacchetto, se sta già trasmettendo aspetta un altro intervallo di tempo.

Possono ancora verificarsi collisioni: il ritardo di propagazione fa sì che due nodi non rilevino la reciproca trasmissione. Se avviene una collisione, non appena rilevata il nodo cessa immediatamente la trasmissione. La distanza e il ritardo di propagazione sono fondamentali per calcolare la probabilità di collisione.

**CSMA/CD (collision detection)** Rilevamento della portante differito, come in CSMA: rileva la collisione in poco tempo e annulla la trasmissione non appena si accorge che c'è un'altra trasmissione in corso. La collisione è di facile rilevazione nelle LAN cablate e difficile nelle LAN wireless.

### 5.3.3 Protocolli MAC a rotazione

Prendono il meglio dai protocolli precedenti, cercando di ereditare dalla suddivisione di canale la condivisione equa del canale evitando congestione, mentre dai protocolli ad accesso casuale ereditano l'efficacia con carichi non elevati.

Si basano sul protocollo **polling**: un nodo principale sonda a turno gli altri. Vengono eliminate collisioni e slot vuoti, si introduce però il ritardo di polling e il problema che se il nodo master si guasta il canale resta inattivo.

### Protocollo token-passing

Un messaggio di controllo circola fra i nodi con un ordine prefissato e chi ne è in possesso può trasmettere. Questo protocollo è decentralizzato, altamente efficiente ma il guasto di un nodo può mettere fuori uso l'intero canale.

### 5.3.4 Riepilogo dei protocolli

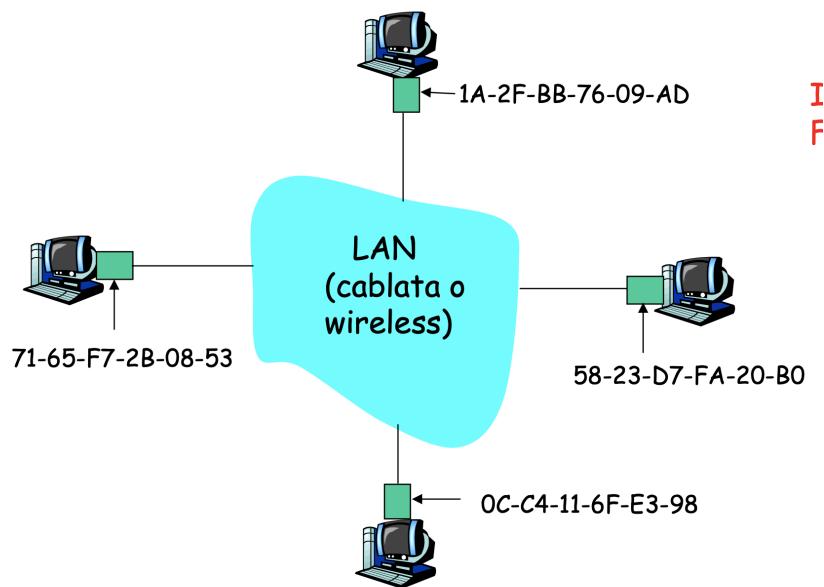
Cosa si può fare con un canale condiviso?

- **Suddivisione del canale:** per tempo, frequenza o codice (TDM, FDM)
- **Suddivisione casuale o dinamica:**
  - ALOHA, S-ALOHA, CSMA, CSMA/CD (Ethernet)
  - Rilevamento della portante: facile per tecnologie cablate, difficile con wireless
  - 802.11 utilizza la variante CSMA/CA
- **A rotazione**
  - Polling di un nodo principale o passaggio di un token
  - Completa decentralizzazione ed elevata efficienza
  - Usati in Bluetooth, FDDI, IBM Token Ring

## 5.4 Indirizzi a livello di collegamento

### 5.4.1 Indirizzi MAC e ARP

L'indirizzo MAC (o fisico o Ethernet), è analogo al numero di codice fiscale di una persona: ha una struttura piatta e non dipende dalla rete in cui si è collegati. Dipende dal produttore della scheda di rete e solitamente ha 48 bit. Viene scritto in esadecimale usando 6 coppie di cifre esadecimale. L'indirizzo broadcast di livello 2 ha tutti 1 (FF-FF-FF-FF-FF-FF).



Sono gestiti dalla **IEEE** che vende alle società costruttrici di adattatori i blocchi di spazio di indirizzi. La società dovrà garantire l'unicità degli indirizzi.

Il vantaggio dell'orizzontalità del MAC è la portabilità delle schede da una rete all'altra (cambierà solo l'IP)

### Protocollo per la risoluzione degli indirizzi (ARP)

Ogni nodo IP nella LAN ha una tabella ARP, la quale contiene la corrispondenza tra indirizzi IP e MAC.

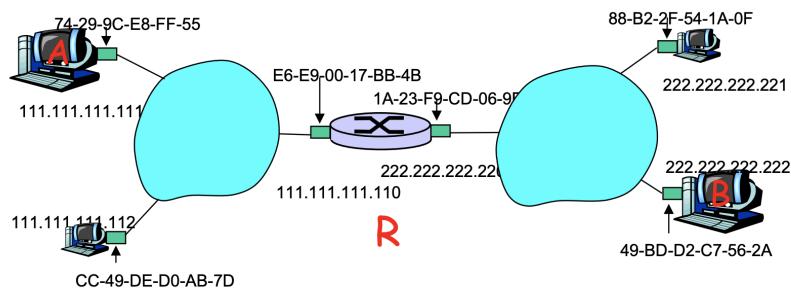
<Indirizzo IP, Indirizzo MAC, TTL>

Il TTL indica quando bisognerà eliminare una voce nella tabella (tipicamente 20 minuti).

**ARP nella stessa sottorete** ARP è plug-and-play: la tabella si costruisce automaticamente, non è necessario l'intervento dell'amministratore di rete. Un nodo trasmette in broadcast il messaggio di richiesta ARP, richiedendo l'indirizzo di un secondo nodo. Quando quest'ultimo riceve il pacchetto ARP risponderà comunicando il proprio indirizzo MAC.

**Invio verso un nodo esterno alla sottorete** Se si vuole inviare un pacchetto da A a B attraverso un router R, il router stesso avrà due tabelle ARP, una per ciascuna rete IP (LAN) a cui è collegato.

1. Il nodo A controlla qual'è la destinazione del suo datagram, se è nella sua stessa sottorete
2. A deve inoltrare il datagram al router R, deve quindi trovare il MAC dell'interfaccia del router utilizzando ARP
3. A quindi incapsulerà il datagram in un frame di livello 2 indirizzato a R, il quale toglierà l'header di livello 2, leggerà l'IP di destinazione e cercherà il percorso nella sua tabella di routing.
4. Il router R dovrà poi cercare l'indirizzo MAC del destinatario sulla seconda sottorete sempre con ARP
5. Infine il router R incapsulerà a sua volta il datagram in un frame di livello 2 inviandolo al destinatario.



## 5.5 Ethernet

Ethernet detiene una posizione dominante nel mercato delle LAN cablate:

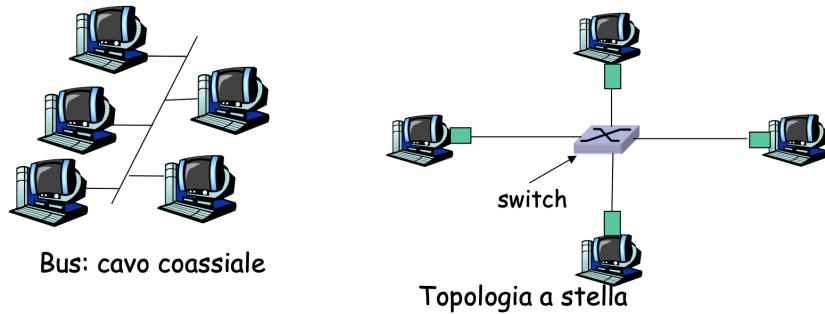
- è stata la prima LAN ad alta velocità con vasta diffusione
- Più semplice e meno costosa di token ring, FDDI e ATM
- Al passo coi tempi con il tasso trasmisivo: da 10 Mbps fino a 10 Gbps

Il progetto originale di Ethernet fu ideato da Bob Metcalfe.

### 5.5.1 Topologia

La topologia originale era quella a Bus con cavo coassiale, diffusa fino alla metà degli anni 90.

Le reti odierne seguono la topologia a stella, ogni nodo è collegato a un hub o commutatore (*switch*) permettendo di eseguire in ogni nodo un protocollo Ethernet separato non entrando in collisione con altri.

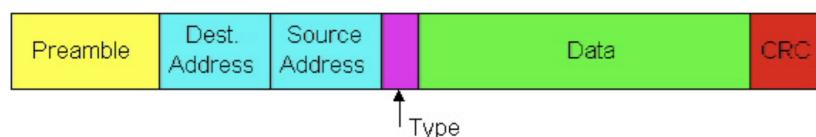


5

### 5.5.2 Struttura dei pacchetti Ethernet

Preamble, MAC di destinazione, MAC di origine, tipo, payload dati e CRC (correzione degli errori).

- Il **preamble** in specifico è costituito da 8 byte, i primi sette sono 10101010 e l'ultimo 10101011, e viene utilizzato per "attivare" il ricevitore quando viene inviato un pacchetto e per sincronizzare l'orologio con quello del trasmittente.
- Gli **indirizzi** sono costituiti da 6 byte. Se è presente l'indirizzo di destinazione o il broadcast il payload viene trasferito direttamente al livello di rete altrimenti il pacchetto viene ignorato.
- Il campo **tipo** consente a Ethernet di supportare i vari protocolli di rete (multiplexing).
- Il controllo **CRC** consente all'adattatore ricevente di rilevare la presenza di un errore nei bit del pacchetto.



### 5.5.3 Servizio senza connessione non affidabile

- **Senza connessione:** non è prevista nessuna forma di handshake preventiva prima di inviare un pacchetto.
- **Non affidabile:** non esiste riscontro, il flusso dei datagram non è garantito poiché il compito viene delegato ai protocolli di livello superiore.

### 5.5.4 Fasi operative del protocollo CSMA/CD

1. L'adattatore che riceve un datagram da livello 3 prepara il frame e ascolta il canale.
2. Se è inattivo inizia la trasmissione, altrimenti resta in attesa.
3. Durante la trasmissione, verifica se ci sono altri segnali provenienti da altri host.

4. Se rileva altri segnali interrompe la trasmissione e invia un segnale di disturbo (*jam*) per avvisare della collisione.
5. L'adattatore si mette in attesa e viene definito uno slot di attesa pari a 512 bit; se viene messo in attesa successivamente l'intervallo raddoppia ogni volta. Dopo 10 volte raggiungerà il valore massimo, quando scade l'attesa se il canale è inattivo si potrà trasmettere nuovamente.

Il segnale *jam* viene trasmesso a un voltaggio più elevato ed è lungo 48 bit. L'attesa è esponenziale con l'obiettivo di stimare quanti sono i nodi in attesa coinvolti.

### 5.5.5 Efficienza di Ethernet

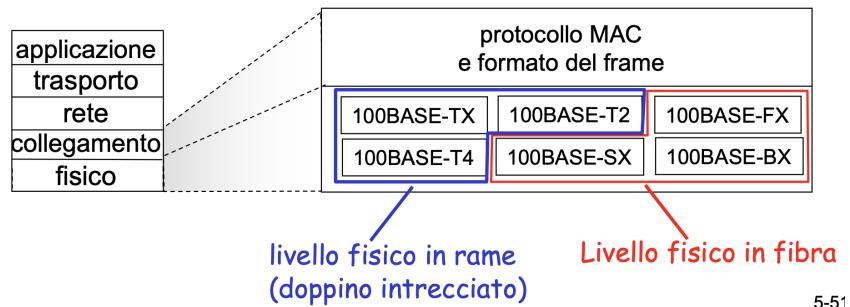
- Tempo di propagazione: tempo massimo che occorre al segnale per propagarsi tra due host
- Tempo di trasmissione: tempo necessario per trasmettere un pacchetto della maggior dimensione possibile

$$\text{efficienza} = \frac{1}{1 + 5t_{prop}/t_{trasm}}$$

- Se il tempo di propagazione tende a 0, l'efficienza tenderà a 1 (100%, efficienza massima)
- Per aumentare l'efficienza, in alternativa, si può aumentare il tempo di trasmissione.
- Molto meglio di ALOHA: decentralizzato, semplice e poco costoso.

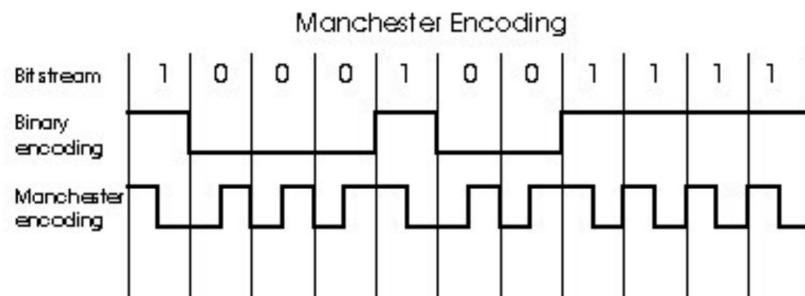
### 5.5.6 Ethernet 802.3

Sono presenti diversi standard Ethernet: il MAC e il frame sono solitamente standard. Abbiamo però differenti velocità e differenti mezzi trasmittivi (fibra o cavo).



5-51

### 5.5.7 Codifica Manchester



La codifica Manchester veniva utilizzata in 10BaseT: durante la ricezione di ciascun bit si verifica una transizione, permettendo di sincronizzare gli orologi di trasmittenti e riceventi. L'operazione veniva effettuata a livello fisico.

## 5.6 Switch a livello di collegamento

### 5.6.1 Hub

L'hub è un dispositivo "stupido" che opera sui singoli bit:

- riproduce un bit incrementandone l'energia trasmettendolo su tutte le interfacce, anche se su alcune c'è un segnale (avverrà una collisione)
- non implementa rilevazione di portante né CSMA/CD

### 5.6.2 Switch

Lo switch è un dispositivo a livello di link, è più intelligente di un hub e svolge un ruolo attivo:

- filtra e inoltra i pacchetti
- ha una tabella e sa a quale porta inoltrare un pacchetto
- è trasparente agli host
- è un componente plug-and-play con autoapprendimento, non richiede l'intervento salvo configurazioni particolari

Lo switch consente più trasmissioni simultanee: i pacchetti vengono infatti bufferizzati e il protocollo Ethernet viene implementato su ciascun collegamento in entrata, evitando collisioni ma consentendo il full-duplex. Grazie allo switching avvengono quindi più trasmissioni simultaneamente.

#### Tabella di commutazione

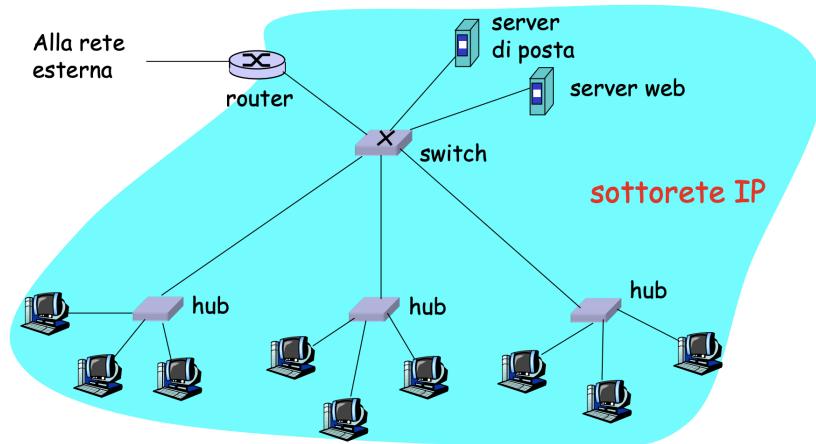
Componente software utilizzato dallo switch per sapere dove inoltrare i pacchetti ricevuti. Contiene il MAC del nodo di destinazione e l'interfaccia associata, assomiglia alle tabelle di instradamento con la differenza che può essere generata automaticamente dallo switch.

**Autoapprendimento** Quando riceve un pacchetto, lo switch impara l'indirizzo del mittente: registrerà quindi nella tabella di switching la coppia indirizzo/porta.

#### Collegare gli switch

Gli switch possono essere interconnessi tra loro: gli host avranno la sensazione di essere sulla stessa rete di livello 2 ma saranno su differenti domini di collisione.

### Esempio di rete di un'istituzione



### Switch e router a confronto

- Entrambi i dispositivi sono store-and-forward
  - Il router lavora a livello di rete (dominio di broadcast)
  - Lo switch lavora a livello di collegamento (dominio di collisione)
- I router mantengono tabelle d'inoltro e implementano algoritmi d'instradamento
- Gli switch mantengono tabelle di commutazione e implementano il filtraggio e algoritmi di autoapprendimento

## 5.7 PPP: protocollo punto-punto

Protocollo estremamente semplice, prevede un solo collegamento tra mittente e destinatario. Non è necessario un protocollo di accesso al mezzo, non occorre un indirizzo MAC esplicito, veniva usato soprattutto nei collegamenti ISDN.

Sono presenti protocolli più complessi come **HDLC** (High-level Data Link Control).

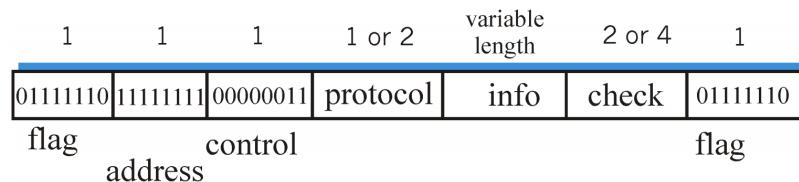
### 5.7.1 Requisiti di IETF (RFC 1547)

- **Framming dei pacchetti:** incapsulare il pacchetto di rete dentro una struttura riconoscibile a livello di link e a livello fisico
- **Trasparenza:** non porre nessuna restrizione alla configurazione dei dati
- **Rilevazione di errori**
- **Disponibilità della connessione:** rilevare eventuali guasti
- **Negoziazione degli indirizzi di rete:** necessario un protocollo per interagire con la rete per ottenere ad esempio un indirizzo IP

Le altre funzioni come correzione degli errori, controllo di flusso, riordinamento dei pacchetti **sono delegati ai livelli superiori**.

### 5.7.2 Formato dei pacchetti dati PPP

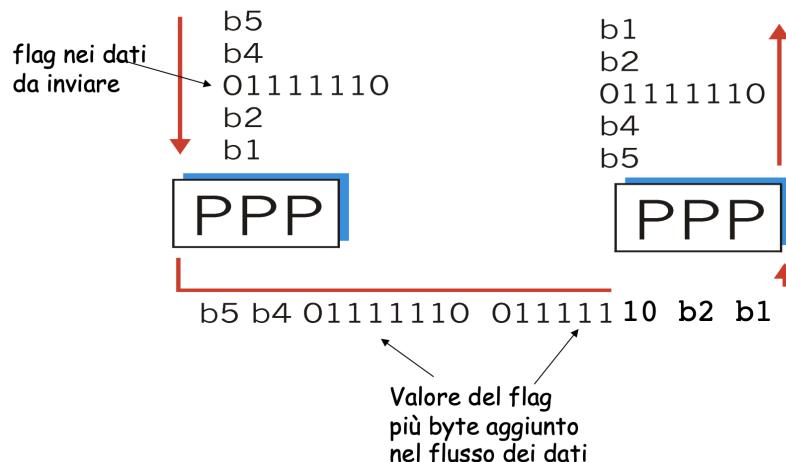
- **Flag:** ogni pacchetto inizia e termina con un byte di valore **01111110**
- **Indirizzo:** unico valore **11111111** broadcast (solo due entità che comunicano)
- **Controllo:** unico valore, tutti i frame sono dello stesso tipo
- **Protocollo:** qual è il protocollo di livello superiore cui appartengono i dati encapsulati



#### Riempimento dei byte

Per il requisito di trasparenza non dev'essere possibile inserire nel campo informazioni la stringa flag **01111110**, poiché non sarebbe riconoscibile la fine del frame PPP.

Per ovviare a questo problema si ricorre alla tecnica del byte stuffing: si aggiunge un byte di controllo pari al flag prima di ogni byte di dati. In questo modo il destinatario riconoscerà la presenza di dati se sono presenti due byte **01111110** consecutivi.



## 5.8 Canali virtuali: una rete come un livello di link

### 5.8.1 Il concetto di virtualizzazione

Per **virtualizzazione** si intende il processo di sostituzione di una versione fisica con una rappresentazione software. Consente di spezzare la dipendenza tra hardware e funzionalità software e una veloce innovazione, è più facile testare e progettare i servizi senza il bisogno dell'ambiente fisico. La virtualizzazione consente di isolare e partizionare le risorse disponibili, oltre alla possibilità di maggior controllo.

### Tipi di virtualizzazione

- **Virtualizzazione hardware:** astrarre la funzionalità logica dall'infrastruttura fisica (ad esempio programmazione con compilatore)
- **Virtualizzazione di rete:** reti virtuali basate su dispositivi virtuali, successivamente mappate su risorse fisiche
- **Cloud:** gestire al meglio grosse quantità di CPU, storage e rete fornendo un pool di risorse aggregate, con lo scopo di fornire le risorse in maniera scalabile (illusione di risorse infinite)

### Storia della virtualizzazione

La virtualizzazione inizia nell'era dei mainframe negli anni '60 dove le risorse di computazione venivano condivise tra molti utenti. Successivamente la virtualizzazione venne utilizzata nei datacenter, consentendo la suddivisione delle risorse disponibili, arrivando al giorno d'oggi con la virtualizzazione delle reti.

#### 5.8.2 Internet: virtualizzazione delle reti

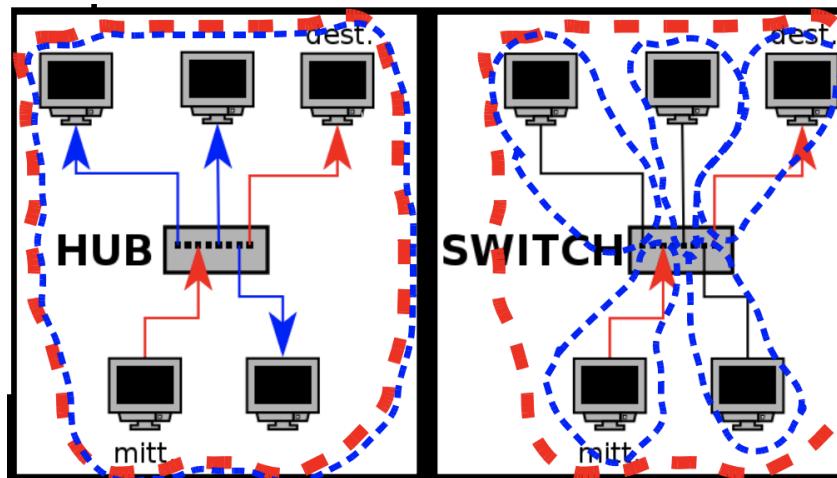
Grazie all'indirizzamento IP e ai gateway divenne possibile collegare tra loro reti diverse per configurazione tra loro che prima non potevano comunicare. IP consente quindi di virtualizzare le reti.

### Architettura di Cerf e Kahn

Propone un primo tipo di virtualizzazione, dove il livello 3 (IP) rende tutto omogeneo, infatti il livello è necessariamente standard. La virtualizzazione ha consentito di spezzare l'indirizzamento a livello globale (livello 3) da quello locale (livello 2), virtualizzando le tecnologie di livello 2 (cavo, satellite, 56K).

### Ethernet e domini

- **Dominio di "collisione":** determinato dall'insieme degli host che possono risentire di una collisione generata da due postazioni arbitrarie
- **Dominio di "broadcast":** insieme degli host che ricevono i pacchetti in broadcast



### 5.8.3 LAN virtuali

Consente di definire più reti locali virtuali distinte utilizzando una stessa infrastruttura fisica. Ogni VLAN si comporta come una rete locale separata dalle altre:

- i pacchetti di broadcast sono confinati all'interno della VLAN
- la comunicazione a livello 2 è confinata all'interno della VLAN
- l'interconnessione tra più VLAN viene effettuata a livello 3 (è necessario un router)

Le VLAN sono definite nello standard 802.1q e nel 802.1d che riguarda la comunicazione tra diversi standard 802 attraverso bridge.

#### Scopo delle VLAN

L'utilizzo delle VLAN consente:

- **risparmio:** definire più topologie virtuali utilizzando la stessa infrastruttura fisica. Non sono necessarie modifiche all'hardware
- **aumento di prestazioni:** costruire la rete in base alle esigenze del momento, limitando il traffico broadcast
- **aumento della sicurezza:** suddividere il traffico e isolarlo nelle varie VLAN
- **flessibilità:** più facile spostare un utente dal punto di vista logico da una VLAN a un'altra

#### Requisiti sui bridge

Per implementare le VLAN è necessario che gli apparati supportino lo standard 802.1q, con il quale è possibile definire due tipologie di VLAN: la port based (privata) e quella tagged (802.1q).

Bisogna anche istruire bridge e switch perché riconoscano le VLAN, non è possibile farlo in autoapprendimento ma vanno preprogrammati.

#### Funzioni del bridge in 802.1q

Per supportare le VLAN è necessario che i bridge svolgano le seguenti tre funzioni:

- **ingress:** l'apparato deve capire a quale VLAN appartiene il frame in ingresso
- **forwarding:** effettuare l'inoltro in base alla VLAN di appartenenza
- **egress:** deve poter trasmettere il frame in uscita in modo che la VLAN sia interpretabile dagli altri bridge

#### Port based VLAN (untagged)

Tecnica abbastanza semplice, assegna in maniera statica ciascuna porta del bridge a una VLAN definita con la configurazione del bridge. Permette di costruire su un singolo apparato 2 o più bridge logici.

Le funzioni del bridge sono semplici:

- **ingress:** un frame in ingresso appartiene alla VLAN a cui è assegnata la porta
- **forwarding:** frame inoltrato solamente verso le porte appartenenti alla stessa VLAN (forwarding database distinto per ogni VLAN)
- **egress:** determinata la porta il frame viene trasmesso così com'è

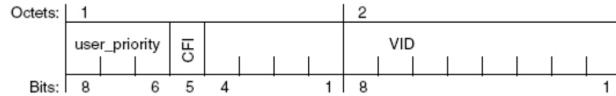
Le VLAN untagged non richiedono di modificare i pacchetti Ethernet poiché viene definito tutto sul bridge (che dev'essere compatibile con lo standard 802.1q).

### VLAN 802.1q (tagged)

Con questo standard è possibile far condividere lo stesso link fisico da tra VLAN differenti, stampando nel pacchetto di livello 2 la VLAN di appartenenza, aggiungendo nel frame Ethernet 4 byte che trasportano le informazioni sulla VLAN. Questo identificativo (VLAN tag) deve essere uguale per tutti i bridge che saranno tutti programmati per riconoscere tale VLAN.

**Frame Ethernet 802.1q** Vengono aggiunti 4 byte dopo gli indirizzi di sorgente e destinazione i quali conterranno il **TPI** (Tag Protocol Identifier), che specifica che il frame è aderente a 802.1q e il **TCI** (Tag Control Information) che trasporta informazioni relative alla VLAN (priorità, interconnessione e VLAN tag).

(6 bytes)	(6 bytes)	(2 bytes)	(2 bytes)	(2 bytes)	(1500 bytes)	(4 bytes)
Source Addr	Dest Addr	TPI	TCI	Type	Data	CRC



**Considerazioni sul frame** La modifica proposta da 802.1q richiede anche la modifica della dimensione massima di 1518 bytes con l'aggiunta di due byte. Inoltre, il campo TPI ha un valore non utilizzato come "protocol type" nei frame Ethernet ordinari in modo che sia riconoscibile che il frame è di tipo 802.1q ma una scheda non compatibile non scarti il frame.

### VLAN con switch/router

Molti produttori consentono di implementare all'interno degli switch funzionalità di routing (livello 3), permettendo di interconnettere tra loro diverse VLAN mantenendo la separazione dei domini di broadcast.

Utilizzando porte configurate in modalità TRUNK è possibile trasportare su un unico cavo i dati di più VLAN, lasciando agli switch il compito di inoltrare i pacchetti correttamente.

### Porte tagged e untagged

Negli switch 802.1q tutte le porte devono essere associate a una o più VLAN: se la porta è associata a una VLAN untagged i frame ricevuti non trasporteranno tag ne lo trasporteranno i frame in uscita, il link su tali porte si chiama *access link*.

Se la porta è in modalità tagged, il link si chiamerà *trunk link* e la VLAN di appartenenza del frame sarà definita dal valore nel tag.

### Protocol based VLAN

Esiste la possibilità di assegnare un frame a una VLAN in maniera dinamica, sulla base di diversi parametri opportunamente configurati negli apparati (richiesti particolarmente evoluti).

Viene effettuato il packet filtering in base a delle regole come IP del mittente, protocol type, indirizzo Ethernet. Può essere definita una associazione statica, che avrà precedenza sulle altre regole.

Alcuni protocolli proprietari consentono di configurare le regole dinamiche in maniera centrale, importando le configurazioni tramite la rete, ad esempio non consentire l'accesso alle VLAN se il MAC address dell'host non è stato registrato dall'amministratore di rete.

### Default VLAN

Gli switch 802.1q sono preconfigurati con una default VLAN assegnata col tag 1 e tutte le porte assegnate ad essa in modalità untagged, permettendo al primo accesso di tale switch un funzionamento tradizionale. Per poter modificare il VLAN ID associato a ciascuna porta bisognerà eliminare tale porta dalla VLAN per poi poterla riconfigurare.

### VLAN di management

In ogni rete IP sono presenti due piani di funzionalità: un piano dati (o data plane) dove circola il traffico degli utenti e un piano di controllo (control plane) relativo al traffico di controllo della rete (BGP).

Usando le VLAN è interessante poter creare una VLAN relativa alla gestione della rete, utilizzando la stessa infrastruttura.

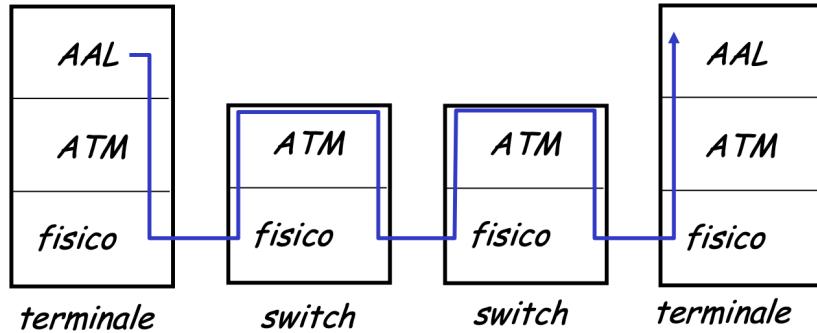
#### 5.8.4 ATM e MPLS

ATM e MPLS sono delle soluzioni che consentono di generare circuiti virtuali utilizzando l'approccio a commutazione di pacchetto, consentono l'allocazione delle risorse per i flussi e la gestione della qualità del servizio. Queste tecnologie possono essere integrate nelle reti IP, attualmente sono utilizzate per interconnettere alcune zone della rete e non sono visibili all'utente.

#### Trasferimento asincrono (ATM)

ATM è nato verso metà anni 80 con l'obiettivo di estendere la tecnologia delle reti telefoniche in modo tale da essere utilizzata per reti dati, progettando reti in grado di trasportare file audio e video in tempo reale e supportare file di testo e immagini. Può essere considerata la tecnologia telefonica di ultima generazione e viene tutt'ora usata nelle reti ADSL, consente la realizzazione di circuiti virtuali e l'implementazione del QoS.

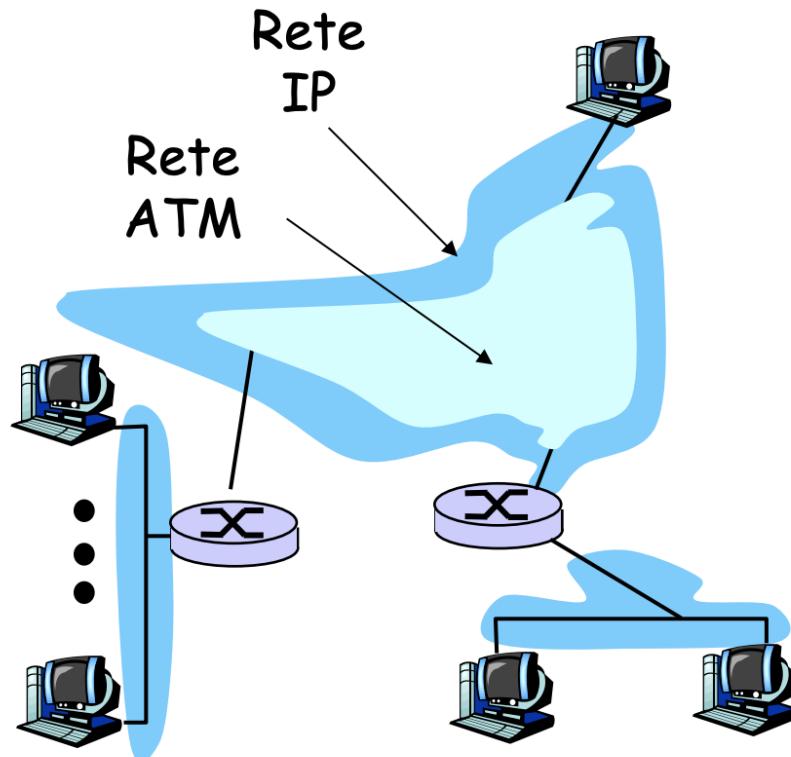
#### Architettura ATM



Segue la struttura TCP/IP, i terminali hanno 3 livelli mentre gli switch 2. I 3 livelli sono:

- **AAL (ATM adaptation layer):** presente nei dispositivi alla periferia della rete, svolge una funzione analoga al livello di Trasporto quindi segmentazione e riassemblaggio dei pacchetti e mappatura dei flussi nelle tipologie di QoS
- **ATM:** fulcro dell'architettura, considerato "livello di rete", definisce la struttura della cella ATM (pacchetto) e tutti i suoi campi
- **Livello fisico**

La rete ATM inizialmente era concepita come una rete stand-alone, che fosse in grado di trasportare dati "da una scrivania a un'altra", venendo considerata una tecnologia di rete. Dopo l'affermazione dello stack TCP/IP come standard di Internet, per far sì che ATM si potesse integrare nella reti IP venne messo al suo di sotto, diventando un livello 2, o livello di link commutato (utilizzato solo in alcune reti al di sotto di IP).



**AAL: ATM Adaptation Layer** Presente solo negli host terminali, adatta i livelli superiori al livello ATM sottostante frammentandoli adeguatamente (come nella segmentazione TCP in pacchetti IP).

Ci sono diverse tipologie (o profili) AAL che suggeriscono il QoS richiesto:

- **AAL1:** servizio a tasso costante, CBR, come nei servizi tradizionali per il traffico telefonico
- **AAL2:** servizio a tasso variabile, VBR, adeguato per la trasmissione di video MPEG
- **AAL5:** servizio dati (datagram IP)

**Livello ATM** Offre il servizio di trasporto di celle attraverso la rete ATM, analogo al livello di rete IP con servizi però molto differenti.

Architettura della rete	Modello di servizio	Larghezza di banda	Garanzie			Feedback congestione
			Perdita	Ordine	Timing	
Internet	best effort	nessuna	no	no	no	no (dedotta se c'è perdita)
ATM	CBR	Tasso costante	sì	sì	sì	non c'è congestione
ATM	VBR	Tasso garantito	sì	sì	sì	non c'è congestione
ATM	ABR	Minimo garantito	no	sì	no	sì
ATM	UBR	nessuna	no	sì	no	no

Il livello ATM implementa una rete a pacchetto a circuiti virtuali, chiamati **canali virtuali (VC)**, percorsi con un collegamento diretto fra sorgente e destinazione. Ciascun pacchetto viene marchiato con l'indicatore **VCI** cosicché ogni switch saprà come interpretarlo. Al canale virtuale possono essere riservate **risorse dedicate** come banda e buffer per garantire il QoS.

I canali virtuali possono essere **permanenti**, per connessioni di lunga durata, utilizzati tra zone della rete lontane tra loro, oppure **dinamici** (creati su richiesta).

L'utilizzo di canali virtuali ha il vantaggio di poter controllare prestazioni e QoS, controllando la congestione della rete, ma ha gli svantaggi che potrebbe non esserci un profilo per ogni tipologia di dato da trasportare, e avendo un numero limitato di canali virtuali non è del tutto scalabile. Nel caso di connessioni di breve durata la costruzione del VC richiede tempo riducendo le prestazioni percepite.

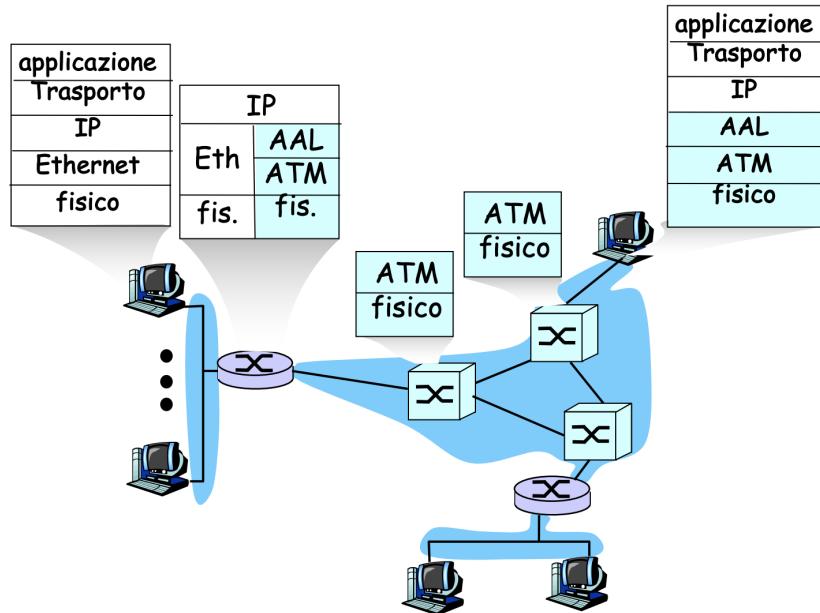
La **cella ATM** è costituita da un'intestazione da 5 byte (VCI, Payload Type, Priorità sulla perdita di cella, e byte di controllo errore) e un carico utile da 48 byte, che consente una trasmissione più efficace avendo dimensione fissa e un ritardo minore data la piccola dimensione.

**Livello fisico ATM** Suddiviso in due parti:

- **Transmission Convergence Sublayer (TCS):** adatta la cella creata da ATM al livello fisico sottostante creando il checksum, effettuando la delineazione della cella e consentendo la strutturazione del canale trasmettendo celle inattive se non ci sono dati da inviare (il canale fisico è come un nastro trasportatore di celle)
- **Physical Medium Dependent:** dipende dal mezzo fisico utilizzato

Il livello fisico di ATM consente il funzionamento con diverse tecnologie come SONET/SDH (reti ottiche sincrone), T1/T3 (fibra, microonde e cavo) o mappare le celle ATM su canali non strutturati (grazie alla delineazione di ATM).

## IP su ATM



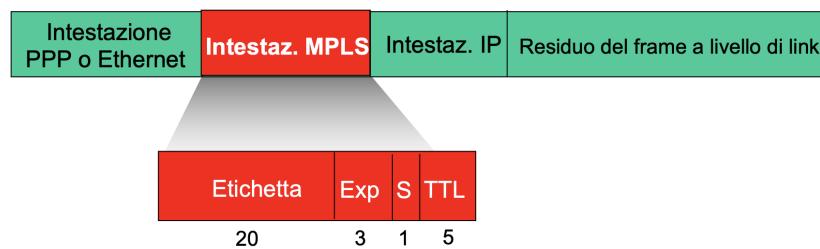
Quando si raggiunge un router di ingresso nella rete ATM, questo router dovrà determinare come trasferire il datagramma attraverso la rete, utilizzando l'indirizzo IP di destinazione per capire dove inoltrarlo e utilizzando ARP per chiedere alla rete ATM di costruire il percorso relativo.

Raggiunto il router di uscita verrà risalita la pila protocollare, incontrando AAL5 che permetterà di ricostruire il datagramma IP e di conseguenza la risalita del pacchetto IP e sua consegna.

Per il corretto funzionamento di IP su ATM sarà necessario quindi un protocollo ARP dedicato ad ATM e dei router dotati di uno stack protocollare e un'interfaccia di rete compatibili con lo standard.

### Multiprotocol Label Switching (MPLS)

L'obiettivo iniziale del protocollo MPLS è quello di velocizzare l'inoltro IP usando un'etichetta di lunghezza stabilita (invece dell'indirizzo di destinazione IP).



I router a commutazione di etichetta (router MPLS) inviano i pacchetti analizzando l'etichetta MPLS nella tabella d'instradamento, passando il datagramma all'interfaccia corretta. Viene utilizzato un particolare protocollo RSVP-TE (estensione di RSVP) per distribuire etichette tra router, che consente l'invio di pacchetti lungo reti non utilizzabili con IP standard. Questi router coesistono coi router "solo-IP".