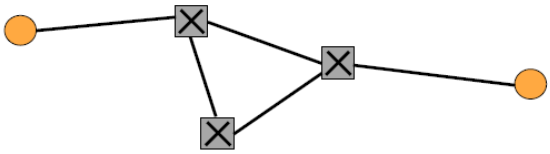
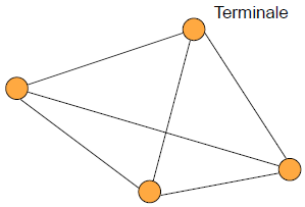
INTRODUZIONE ALLE RETI

⁑RETI
La rete è necessaria nel momento in cui una grande popolazione ha la necessità di comunicare condividendo canali di comunicazione a richiesta con possibilità di riuso. I componenti della rete sono:

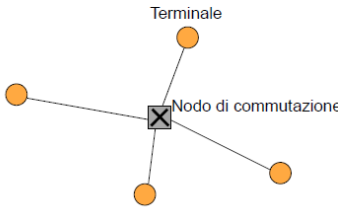
- Terminali → fungono da interfaccia con l'utente finale e codificano l'informazione
- Mezzi trasmissivi o collegamenti → permettono il trasferimento di uno o più flussi
- Nodi di commutazione → utilizzano i messi trasmissivi al fine di creare canali di comunicazione sulla base delle richieste degli utenti



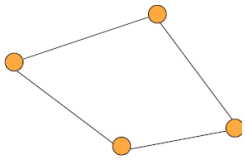
→ Topologie di rete: descrizione geometrica della rete con rami, nodi e grafi che ne identificano la forma



→ Maglia completa: ho un collegamento per ogni coppia di nodi quindi non ho alcun nodo di commutazione. N nodi implicano N(N-1)/2 collegamenti e quindi porta grande resistenza ai guasti, complessità e costo

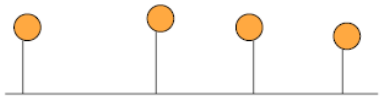


→ Maglia a stella: ho N collegamenti perché tutti i terminali sono collegati ad un nodo centrale di commutazione che deve smistare le informazioni (centro stella)

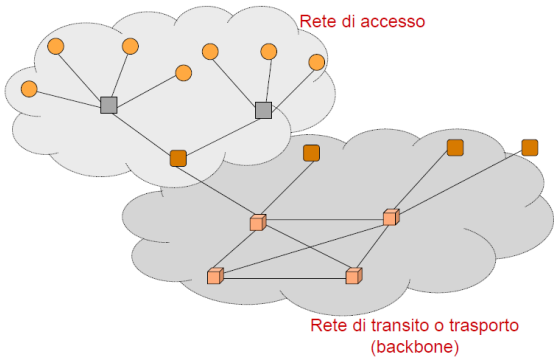


→ Anello: ho un collegamento in uscita ed uno in entrata da ogni terminale. Possono essere monodirezionali (poco resistenti ai guasti) o bidirezionali

→ Bus: unico collegamento attraverso il quale comunicano tutti i terminali. Può essere passivo o attivo oltre che mono o bidirezionale ma non è molto resistente ai guasti anche se è molto semplice ed economico. Il mezzo di trasmissione è condiviso e per questo motivo è necessario definire un protocollo di accesso MAC



_RETE GERARCHICA



La rete è solitamente organizzata su più livelli. I terminali sono connessi a nodi periferici e questi sono connessi tra loro tramite nodi intermedi. A lunga distanza si usano nodi di transito interconnessi di solito a maglia completa. Le reti più periferiche sono reti di accesso e presentano nodi di commutazione connessi variamente ai terminali. Queste reti sono connesse tra loro da reti di transito o trasporto (backbone) dove i nodi sono più complessi e costosi perché devono essere affidabili

_FUNZIONI DI RETE

- Trasmissione → trasferimento fisico del segnale
- Commutazione → instradamento delle informazioni all'interno della rete tramite nodi e collegamenti
- Segnalazione → scambio di informazioni necessarie per la gestione della comunicazione e della rete stessa
- Gestione → mantenimento delle funzioni di rete

‡RETI E SERVIZI

Le reti si sono evolute in base al servizio → diversi servizi = diverse reti

→ ITU: tassonomia dei servizi. I servizi si dividono in:

- Interattivi → esiste una interazione tra sorgente e destinazione (conversazione, messaggistica, consultazione)
- Distributivi → la sorgente diffonde informazioni in modo indipendente ad un numero imprecisato di destinazioni. Si divide in base alla presenza o meno del controllo di presentazione.

→ Flusso informativo:

- Punto-punto → trasferimento informativo uno ad uno
- Punto-multipunto (multicast) → trasferimento da uno a tanti
- Diffusivo (broadcast) → trasferimento da uno a tutti
- Monodirezionale
- Bidirezionale simmetrico → uguale capacità per ogni direzione
- Bidirezionale asimmetrico → diversa capacità per direzione

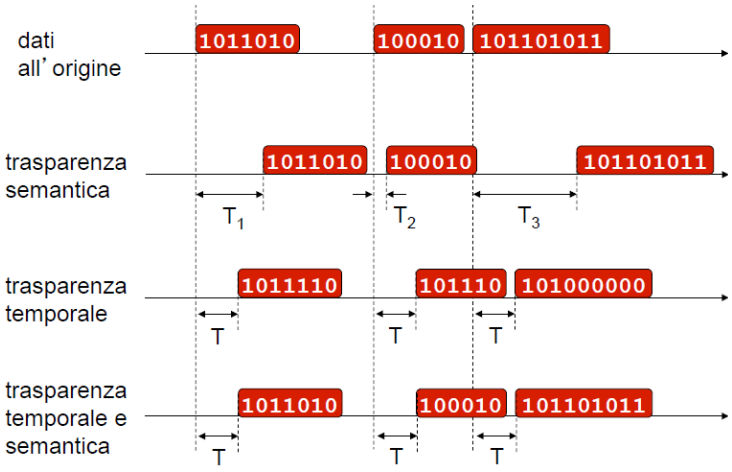
→ Servizio:

- Monomediale → trasporta informazioni di un solo tipo e quindi trasferisce più tipologie di informazione sotto forma di un solo segnale
- Multimediale → le diverse tipologie di informazioni sono trasportate dalla medesima rete con modalità distinte (diversi protocolli e di conseguenza differenti qualità di servizio)

_PROBLEMA DELLA QOS (Quality Of Service)

Il problema principale dei servizi è quello di trasparenza e del mantenimento della qualità della comunicazione percepita dall'utente del servizio. La trasparenza può essere di due tipi:

- Semantica → riguarda l'integrità delle informazioni trasportate e richiede procedure di recupero di situazioni di errore che possono incorrere nella rete
- Temporale → riguarda la variabilità dei ritardi di transito. Il ritardo non è tuttavia eliminabile del tutto a causa della presenza di un minimo ritardo di propagazione



Si possono definire degli indicatori che diano una stima sintetica della QoS come per esempio la probabilità di errore o perdita di informazioni, il ritardo nella consegna (e le sue variazioni, jitter) oppure l'uniformità delle prestazioni. La richiesta di trasparenza semantica o temporale dipende dal tipo di servizio che voglio fornire:

- Applicazioni real-time → richiede trasparenza temporale (basso ritardo e jitter). Il servizio si dice isocrono e in alcuni casi richiede anche che il valore del picco del ritardo di transito sia il minimo possibile
- Applicazioni non real-time → richiede trasparenza semantica (bassa probabilità di errore)

_OPPORTUNITA' DEL DIGITALE

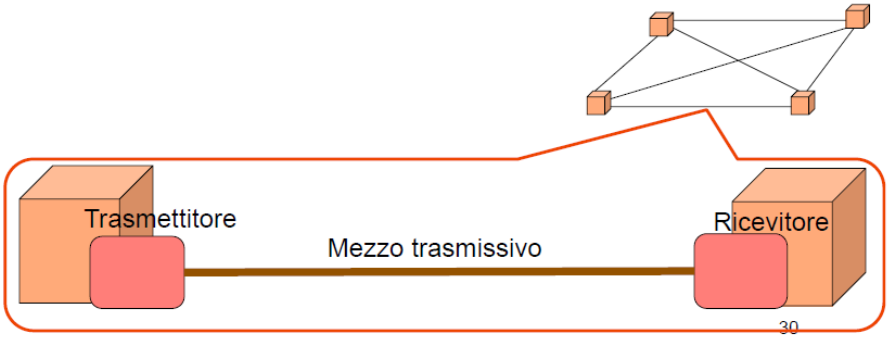
Il passaggio al digitale permette:

- Integrazione → trasporto unificato dell'informazione sotto forma di bit. Diversi servizi richiede diversi requisiti. Una rete integrata nei servizi deve essere estremamente flessibile nella allocazione della banda (distribuzione delle risorse in

conformità delle necessità del servizio) e nella gestione della QoS (le richieste dei vari servizi non devono interferire)
- Elaborazione → i segnali, una volta digitalizzati, sono trattabili con sistemi di elaborazione elettronica per combinarli, comprimerli e cifrarli volendo

⁂TRASMISSIONE: MULTIPLAZIONE E CODIFICA

I nodi di rete sono connessi tramite collegamenti caratterizzati dal tipo di mezzo trasmissivo. Con il termine canale si identifica il mezzo di supporto usato per la trasmissione informativa quindi di conseguenza il segnale stesso che si propaga → secondo la legge di Edholm ogni 18 mesi la band a disposizione dell'utente raddoppia mantenendo circa costante il costo.

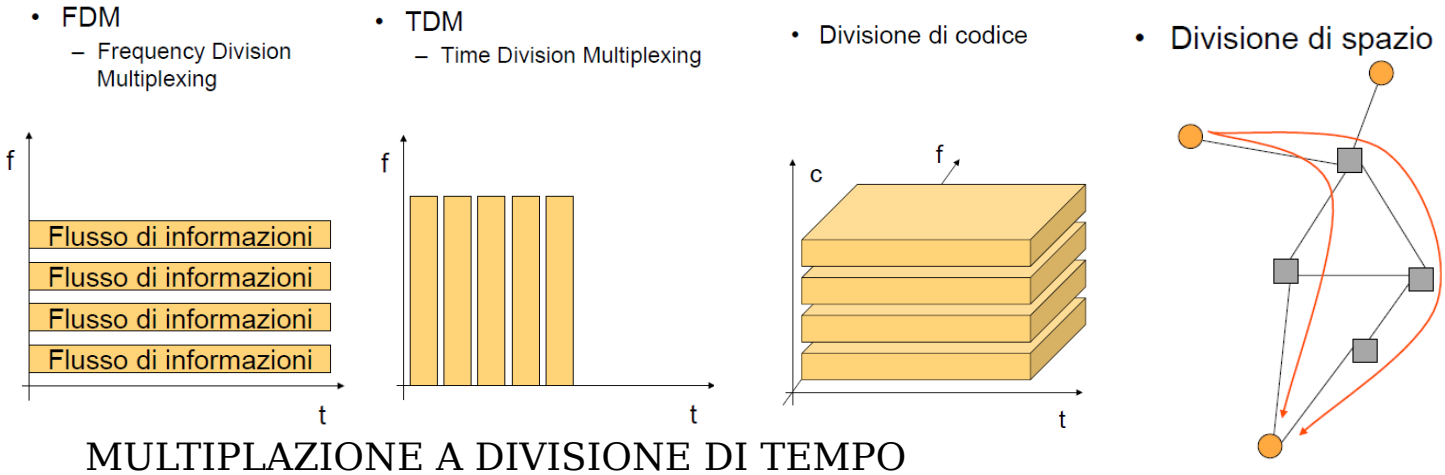


→ Codifica: il passaggio nei vari blocchi di trasmissione comporta una codifica del segnale che viene poi decodificato in ricezione. E' possibile inserire anche in diversi punti una crittografia implementata come funzionalità a parte.

- Codifica di sorgente → elimina la ridondanza intrinseca delle sorgenti diminuendo la velocità di emissione senza comprometterne la fruibilità (riduco il segnale il più possibile garantendo che rimanga utilizzabile)
- Codifica di canale → aggiunge bit di ridondanza per il controllo dell'errore
- Codifica di linea → trasforma la sequenza di bit in una di simboli (multilivello solitamente) per adattare il segnale al mezzo trasmissivo
- Codifica crittografica → trasforma la sequenza dei simboli rendendola incomprensibile a chi non possegga ulteriori specifiche informazioni (chiavi)

→ Multiplazione: più canali sono trasportati dallo stesso mezzo di trasmissione e si può realizzare in diversi modi. Dal punto di vista teorico in condizioni ideali sono tutti equivalenti e differiscono solo per modalità di implementazione:

- FDM (Frequency Division Multiplexing) → Flussi uguali coesistono nel tempo ma sono affiancati nelle frequenze lasciando bande di guardia per comodità del filtraggio finale
- TDM (Time Division Multiplexing) → Flussi uguali occupano tutti il canale delle frequenze ma sono traslati nel tempo con piccole pause gli uni dagli altri
- Divisione di codice → i canali usano tutti il tempo e la frequenza che vogliono ma l'informazione a livello di codifica di linea è modificata in modo che i flussi siano distinguibili tra loro
- Divisione di spazio → i flussi informativi usano risorse diverse della rete e hanno percorsi diversi

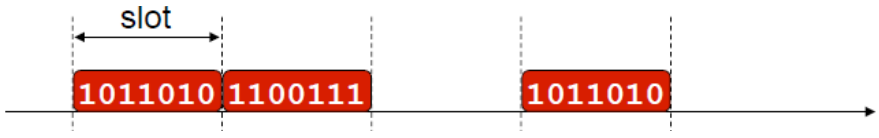


_MULTIPLAZIONE A DIVISIONE DI TEMPO

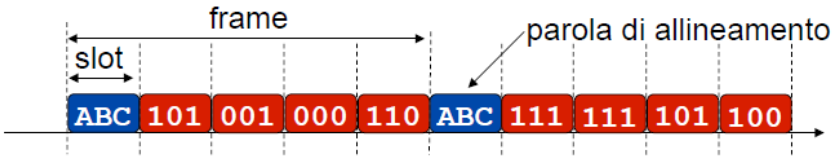
Nelle reti numeriche viene principalmente utilizzata la multiplazione a divisione di tempo la quale si può effettuare in vari modi (righe della tabella) e con varie combinazioni (colonne della tabella).

multiplazione a divisione di tempo		
slotted		unslotted
framed	unframed	
assegnazione statica della banda	assegnazione dinamica della banda	

→ TDM Slotted: l'asse dei tempi è suddiviso in intervalli di lunghezza fissa detti slot e le unità informative hanno tutte lunghezza commisurata al singolo slot

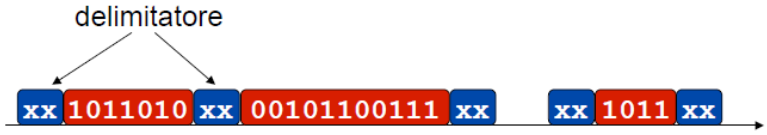


→ TDM Slotted framed: gli slot vengono strutturati in trame (frame) sincronizzati. Si definisce inoltre una regola specifica per la strutturazione della trama

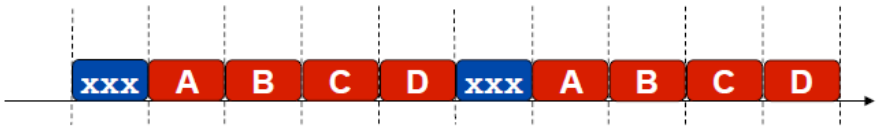


→ TDM Slotted unframed: gli slot si susseguono senza una struttura predefinita ed occorre comunque un sistema di sincronizzazione degli slot. Devo solo capire di fronte ad uno slot se ho informazione o delimitazione

→ TDM Unslotted: l'asse dei tempi non è suddiviso a priori e si possono adottare unità informative di lunghezza variabile con un sistema esplicito di delimitazione delle unità informative (combinazione in codice binario che devo essere sicuro non possa essere contenuta in nessuna possibile unità informativa)



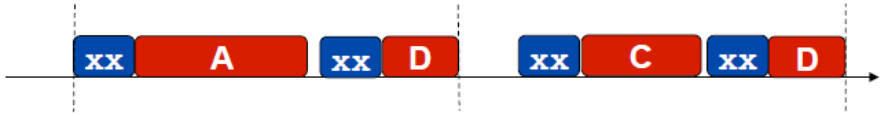
→ Assegnazione statica della banda: il flusso informativo si divide per banda dedicata (bit/sec), la quale non può cambiare a comunicazione in corso. Questo rende la richiesta complessiva di banda controllabile controllando il numero di flussi attivi. Lo slot viene assegnato all'utente per tutto il tempo per cui ha bisogno della rete garantendogli di avere a disposizione sempre la stessa bitrate. La scelta è tipicamente effettuata nel caso di soluzioni slotted-framed → all'interno di ogni trama si assegna uno slot prefissato ad una particolare coppia sorgente-destinazione (chiamata) e la durata dell'allocazione è uguale alla durata della chiamata.



C'è una associazione logica tra slot nella stessa posizione in frame diversi e la banda globale del canale è suddivisa in sotto-canali di capacità fissa → ci sono tanti sotto-canali quanti sono gli slot temporali nel frame. I fenomeni di ritardo temporale introdotti con l'assegnazione statica sono compatibili con le richieste temporali

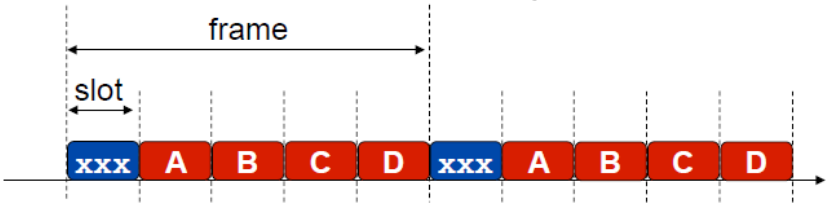
dei servizi di real-time quindi la multiplazione è completamente trasparente all'utente finale

→ Assegnazione dinamica della banda: molti flussi informativi condividono liberamente la banda in base alle necessità.



La banda può quindi cambiare a comunicazione in corso portando il rischio che la richiesta complessiva sia intollerabile

→ S-TDM Synchronous Time Division Multiplexing: le unità informative vengono trasferite periodicamente con ritardo costante (ogni periodo è uguale alla durata del frame)



⁑TECNICHE DI COMMUTAZIONE
Le due tecniche tradizionalmente utilizzate e tuttora più diffuse sono:

→ Commutazione di circuito: usata nella rete telefonica. Fa riferimento ad un circuito di commutazione (fisico o virtuale) che viene dedicato dal nodo ad un utente quanto lo richiede. La rete crea il canale di comunicazione dedicato tra due terminali e questo circuito è riservato ad esclusivo uso dei terminali chiamante e chiamato → esiste un ritardo iniziale dovuto al tempo necessario per instaurare il circuito, dopo di che la rete è trasparente temporalmente per l'utente ed equivale ad un collegamento fisico diretto.

La comunicazione avviene instaurando il circuito tramite opportuna segnalazione → segue lo scambio delle informazioni usando il circuito dedicato appena creato e infine la disconnessione del circuito che può essere utilizzato per altre chiamate da altri utenti.

- Pro →
  - Il circuito è dedicato e garantisce sicurezza ed affidabilità
  - Viene garantita la trasparenza temporale

- Contro →
    - Se le sorgenti hanno basso taso di attività il circuito viene sottoutilizzato.
    - La capacità del canale è fissata dalla capacità del circuito e non si può variare

→ Commutazione di messaggio o di pacchetto: usata nella rete telegrafica e in quella di calcolatori. Si trasportano informazioni in forma numerica e le informazioni di utente sono strutturate in messaggi insieme ad opportune informazioni di segnalazione. I messaggi vengono suddivisi in sotto-blocchi di lunghezza massima prefissata detti pacchetti, per motivi di linea (evitare frammenti troppo lunghi in relazione al rumore) o di rete (limitare i tempi medi di attesa nei nodi). Questi pacchetti vengono trasmessi da un nodo di commutazione all'altro utilizzando in tempi diversi risorse comuni. Tipologia di servizio, qualità, commutazione e multiplazione devono essere coerenti tra loro.

- Pro →
    - Efficienza nell'utilizzazione dei collegamenti poiché la stessa linea è condivisa in modo dinamico da più chiamate (posso fornire ad un utente le risorse inutilizzate di un altro)
    - La rete può sopportare diverse velocità ed effettuare anche conversioni tramite memorizzazione
    - E' facile implementare meccanismi di controllo dell'errore per garantire trasparenza semantica
- Contro →
    - Difficoltà di garantire un predeterminato tempo di transito quindi poco adatta per servizi di tipo real-time

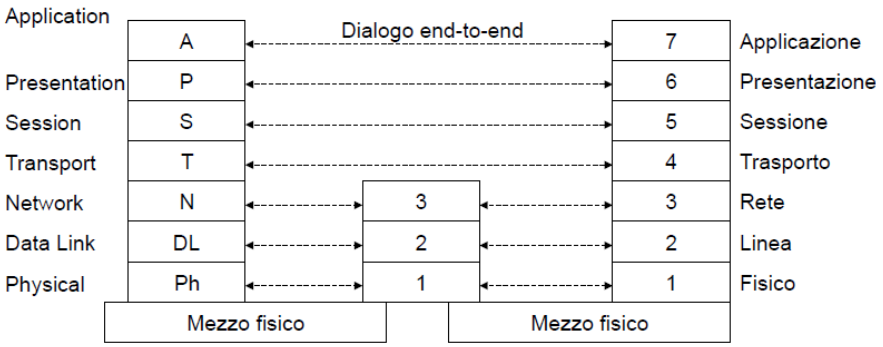
MODELLO ISO-OSI

⁑SISTEMI CHIUSI E MODELLO DI RIFERIMENTO
Tutte le reti di calcolatori della prima generazione nascono come sistemi chiusi in cui tutti i componenti della rete devono essere dello stesso costruttore → si crea incompatibilità ponendo ostacoli alla comunicazione. La ISO fornisce una serie di standard unificati per la realizzazione di reti di calcolatori aperti proponendo il modello di riferimento Open System Interconnection (OSI) che si basa sul concetto di architettura a strati:

- Scompone il problema in sottoproblemi più semplici
- Rende i vari livelli indipendenti
- Definisce solamente servizi e interfacce → i livelli diversi possono essere sviluppati da enti diversi

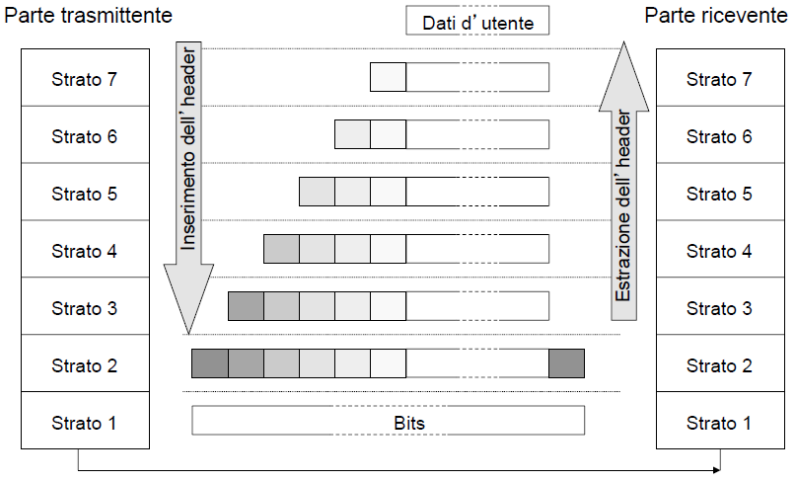
Nei fatti non venne mai applicato a causa della nascita di internet ma rimane importantissima la definizione dell'architettura che rimane il punto di riferimento per le reti di calcolatori.

→ Sistema aperto: ha l'obbiettivo di realizzare una rete di calcolatori in cui qualunque terminale comunica con qualunque fornitore di servizi mediante qualunque rete. Per realizzare un sistema di questo tipo vanno definiti degli standard che hanno in comune un'architettura a strati:



- 1,2,3 → sono detti lower o network oriented layers
- 5,6,7 → sono detti upper o application oriented layers
- 4 → funge da raccordo tra gli upper e i lower layers

Ciascun livello parla con la sua controparte in un dialogo end-to-end.
L'implementazione dei meccanismi di dialogo con la rete avviene per segnalazione.
I dati vengono impacchettati e trasferiti e nel passaggio da uno strato all'altro si aggiungono informazioni all'esterno e si trasmettono quindi sia dati utili che dati di segnalazione → in ricezioni questi dati vengono via via letti e poi tolti.

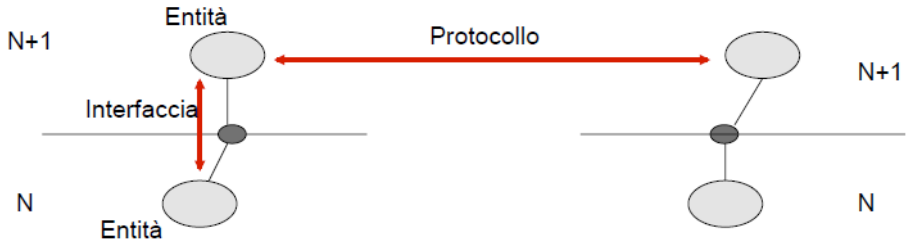


→ Strati:

1) Fisico → ha lo scopo di attivare, mantenere e disattivare la connessione fisica tra entità di stato 2 specificando le modalità di invio dei singoli bit sul mezzo fisico. Deve specificare le caratteristiche:
   - Meccaniche → forma di prese e spine
   - Elettriche → voltaggio e caratteristiche elettriche
   - Funzionali → significato dei vari segnali
   - Procedurali → combinazioni e sequenze dei segnali all'interfaccia necessarie al fine di regolarne il funzionamento
2) Linea → deve attivare, mantenere e disattivare la connessione fisica tra entità di strato 3 e rendere inoltre affidabile il collegamento tra nodi di rete (effettua la codifica di canale)
3) Rete → ha lo scopo di far giungere le unità di informazione, dette pacchetti, al destinatario scegliendo la strada attraverso la rete. Si occupa del problema della commutazione e gestisce uno schema di indirizzi che deve essere universale → sono state create delle reti parziali, dette sotto-reti, e un protocollo di interconnessione di tali reti.
4) Trasporto → riesegue quello che fa lo strato 2 ed è il primo strato che non passa per nodi intermedi. Ha lo scopo di fornire un canale sicuro end-to-end svincolando gli strati superiori da tutti i problemi di rete cercando di controllare l'aleatorietà intrinseca di possibili errori del segnale dopo i primi controlli. Una funzione tipica è adattare la dimensione dei frammenti forniti dagli strati superiori a quella richiesta dalle reti (pacchetti) → effettua il fragmenting/reassembling

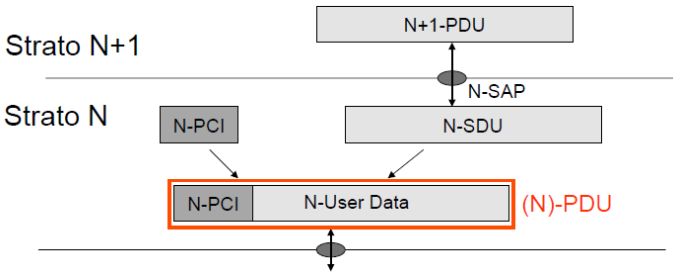
5) Sessione → suddivide il dialogo tra le applicazioni in unità logiche identificate. Permette la chiusura ordinata del dialogo e introduce punti di sincronizzazione per le sue funzioni
6) Presentazione → adatta il formato dei dati usato dagli interlocutori preservandone il significato e permette di concordare la sintassi di trasferimento visto che ogni interlocutore ha la sua sintassi locale
7) Applicazione → è l'utente della rete e non deve offrire servizi a nessuno. Rappresenta il programma applicativo che per svolgere i suoi compiti ha bisogno di comunicare con altre applicazioni remote

ENTITA', INTERFACCE E PROTOCOLLI



- Entità → ogni elemento attivo in uno strato, identificato da un nome simbolico. Nello strato n-esimo possono essere attive una o più entità
- Protocollo → regole di dialogo tra entità di uguale livello quindi a livello orizzontale
- Interfaccia → regole di dialogo tra entità di livelli adiacenti quindi a livello verticale

Anche i dati che vengono trasferiti si differenziano:

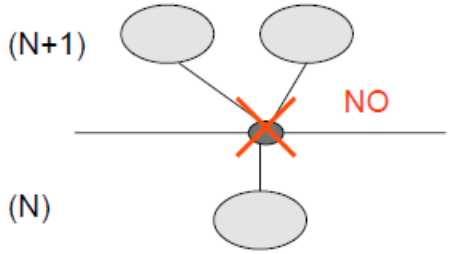


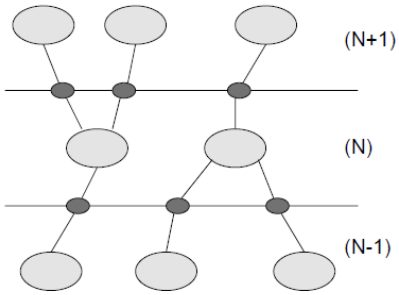
- N-PDU → Protocol Data Unit. Dati trasferiti tra entità di strato N
- N-SDU → Service Data Unit. Dati passati allo strato N dallo strato N+1
- N-PCI → informazioni aggiuntive per il controllo del dialogo a livello N

Per il collegamento tra uno strato e un altro si usa un punto di accesso N-SAP che identifica l'indirizzo del flusso dati tra N+1 e N. Il concetto di incapsulazione si realizza nel livello N dove N-PDU = N-PCI + N-SDU

_SAP

Un’entità di strato N può servire più N-SAP



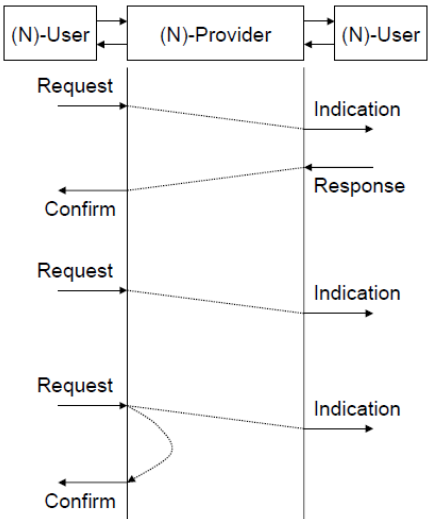


contemporaneamente ed un utilizzatore di strato N può servirsi di più N-SAP contemporaneamente. Non è tuttavia permesso connettere più user dello stesso livello allo stesso N-SAP → si genererebbe ambiguità sulla provenienza/destinazione dei dati mentre ad ogni indirizzo deve essere univocamente associato un nome.

‡SERVIZIO E DIALOGO

Ogni servizio si può fornire in due modalità:

- Connection Oriented → si stabilisce una connessione cioè una associazione logica tra due o più sistemi al fine di trasferire informazioni. Il processo di comunicazione si compone di tre fasi:
  - Instaurazione della connessione tramite lo scambio di informazioni iniziali
  - Trasferimento vero e proprio di dati
  - Chiusura della connessione
- Connectionless → non viene stabilita alcuna connessione e per ogni accesso vengono fornite tutte le informazioni necessarie per il trasferimento dati. Ogni unità dati viene trasferita in modo indipendente dalle altre.



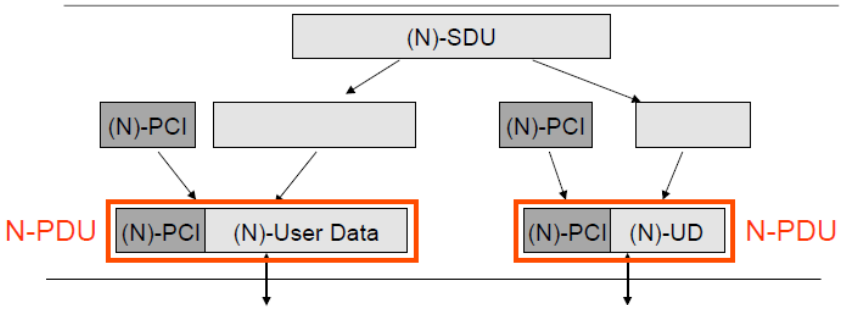
Il dialogo può essere:

- Confermato → con esplicita conferma del destinatario
- Non confermato
- Parzialmente confermato → la richiesta viene confermata dal service-provider

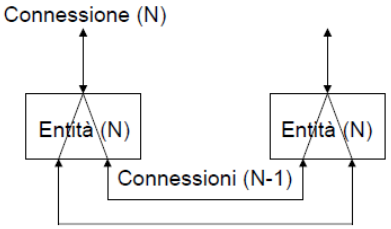
_SEGMENTAZIONE E RIASSEMBLAMENTO

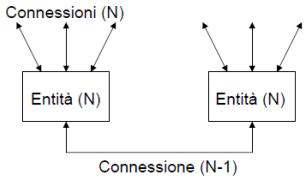
E’ possibile dividere il contenuto di un SDU in una a più PDU → la suddivisione si definisce segmentazione e serve a conformarsi alle limitazioni sulla lunghezza massima dei messaggi mentre la

ricostruzione invece è detta riassemblamento.



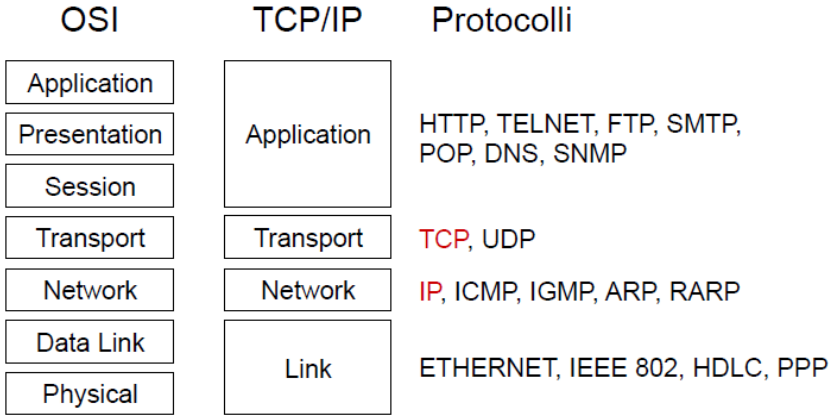
→ Multiplazione: più connessioni di strato N vengono mappate in una di strato N-1 per la condivisione delle risorse





→ Splitting: duale alla multiplazione e aumenta la flessibilità e la velocità di trasferimento dati

⁑DIFFUSIONE DI INTERNET



Mentre il modello di riferimento è stato universalmente adottato come modo di organizzare le architetture dei protocolli, il protocollo IP di OSI e quello di trasporto non hanno avuto successo → il motivo è la diffusione di internet e del protocollo TCP/IP che non usa gli strati di sessione e presentazione ma si interfaccia direttamente con l'applicazione. L'architettura rimane conforme negli strati 3,4 dell'OSI ma semplifica gli altri unendo 1 e 2 e anche 5,6,7.

ARCHITETTURA, INDIRIZZAMENTO E FUNZIONALITA' DI INTERNET

‡STANDARD
Non esistono veri e proprio enti che svolgono la funzione di gestione ma solo enti di coordinamento delle attività di ricerca → i vari protocolli sono definiti in documenti detti RFC, alcuni dei quali diventano Internet Standard.

‡INDIRIZZAMENTO
La comunicazione coinvolge sempre due o più entità e quindi tutti i protocolli contengono sorgente e destinazione. Ci sono diversi modi di indirizzare:

- Nodi di commutazione → fanno riferimento ad indirizzi numerici ben definiti e standardizzati
- Sistemi di sicurezza → fanno riferimento alle identità
- Applicazioni → fanno riferimento a identificativi opportunamente definiti

In internet tipicamente dobbiamo distinguere tra:

- Identifier → identificativo di una certa risorsa di rete
- Locator → indirizzo necessario per localizzare tale risorsa

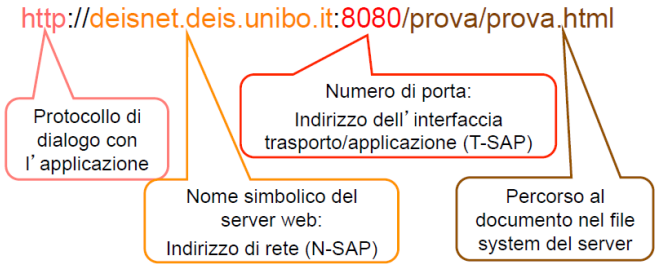
Il locator può cambiare mentre l'identifier è sempre lo stesso. Per esempio se un terminale si sposta da una rete all'altra il locator cambia oppure se un terminale è connesso con più interfacce a infrastrutture diverse presuppone la presenza di molteplici locator attivi in contemporanea

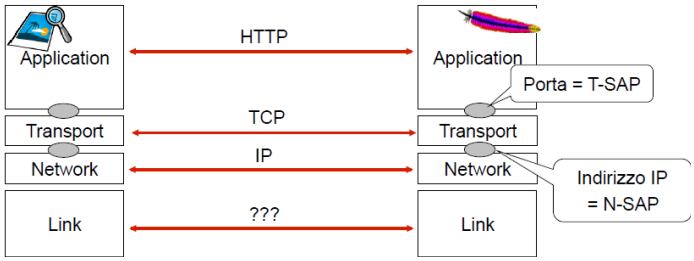
→ Indirizzo globale: valido per tutta la rete e univoco. Va assegnato seguendo una procedura di gestione globale che assicuri la non replicazione

→ Indirizzo locale: valido limitatamente ad una sotto-porzione della rete e di conseguenza replicabile in diversi domini di rete, con assegnamento puramente locale

Ogni risorsa R è univocamente identificata da un indirizzo che la localizza (locator) denominato URL (Uniform Resource Locator), che riflette l'organizzazione a livelli della rete. La stringa arancione è un indirizzo astratto nel senso che non è mappabile su alcuna struttura fisica in prima istanza ed identifica il SAP di livello

3. Questo SAP dà il collegamento (tra livello 2 e 3) alla rete e va identificato in modo univoco.





⁑HTTP
Messaggio di puro testo costruito con:

- Linea iniziale
- Una o più linee di intestazione (header)
- Linea vuota (carattere CRLF)
- Corpo del messaggio di opzionale

HTTP funziona con un meccanismo di richiesta/risposta e non viene mantenuto uno stato della comunicazione (connectionless). Tipicamente avviene una richiesta del Client a cui segue una risposta del Server → il Client manda una request GET che chiede la pagina corrispondente all'URL e il Server invia la response con il contenuto della pagina. Il testo viene inviato direttamente come sequenza di caratteri seguendo le regole del linguaggio HTML, mentre per le immagini e gli eventuali altri oggetti viene inviato l'URL dei files in cui sono contenuti.
In risposta la prima stringa porta la segnalazione della positività o della negatività.

```
GET /Didattica/CorsiCE/RetiLB/
    index.html HTTP/1.1          metodo, file, versione
Accept: */*                      contenuto accettato
Accept-Language: en-us           preferenza linguistica
If-Modified-Since:               ultima versione nella cache
  Wed, 16 Jan 2002 16:37:40 GMT
User-agent:Mozilla/4.0           tipo di browser
Host: deisnet.deis.unibo.it      host
Connection: Keep-Alive           connessione permanente
```

```
HTTP/1.1 404 Not Found
Date: Wed, 03 Mar 2004 17:38:37 GMT
Content-Length: 1067
Content-Type: text/html
Server: Apache/2.0.40 (Red Hat Linux)

<html>
...
Object not found
...
Error 404
...
</html>
```

```
HTTP/1.1 200 OK
Date: Wed, 03 Mar 2004 17:37:44 GMT
Content-Length: 19692
Content-Type: text/html
Server: Apache/2.0.40 (Red Hat Linux)
Last-Modified: Mon, 01 Mar 2004 16:02:27 GMT

<html>
...
qui c'è il testo HTML della pagina richiesta (19692 byte)
...
</html>
```

⁑PROTOCOLLO DI TRASPORTO
Si occupa del trasporto dei dati end-to-end e può trasportare i dati pertinenti ad una qualunque applicazione → i flussi dati di diverse applicazioni sono distinguibili sulla base del numero di porta. Esistono vari protocolli di trasporto scelti in funzione delle caratteristiche che il trasporto deve avere:

- UDP
- TCP
- RTP

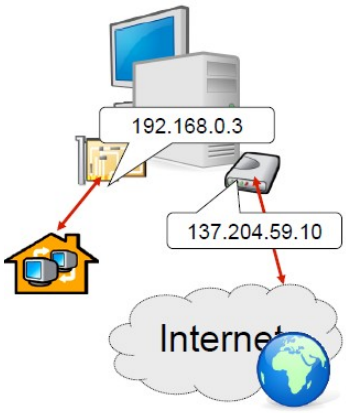
Il numero di porta è un indirizzo di 16 bit in valori decimali da 0 a 65536 che è locale al calcolatore e viene condiviso tra tutti i protocolli di trasporto.

_NUMERI DI PORTA
I numeri di porta si dividono in tre tipologie e il livello 4 deve decidere quali usare rispettando i vincoli:

- Riservati (Well Known Ports) → da 1 a 1023. Possono essere usati solo dai server per scopi predeterminati
- Registrati → da 1024 a 49151. Sono usati da alcuni servizi ma anche da client
- Ad uso dei client → da 49152 a 65536. Possono essere usati liberamente

⁑INDIRIZZO IP
Sono indirizzi di lunghezza fissa pari a 32 bit scritti convenzionalmente come sequenza di 4 numeri decimali da 0 a 255 in rappresentazione dotted decimal. L'indirizzo tuttavia non viene presentato come numeri ma come stringa di caratteri → la stringa è correlabile semanticamente a ciò che voglio cercare e viene costruito un sistema di riconoscimento dei nomi simbolici. Il client deve conoscere indirizzo Ip e numero di porta del server destinazione e li ricava dall'URL dove l'indirizzo viene ricavato dal nome simbolico tramite il server DNS → quando chiamo l'IP viene richiesto al DNS a quale numero IP corrisponde l'indirizzo fornito.

10001001.11001100.11010100.00000001
137.204.212.1



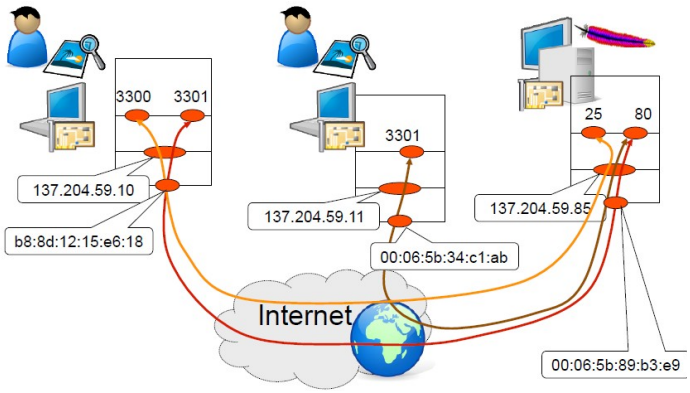
L'indirizzo identifica i punti di interconnessione di un host con la rete e quindi non un host individuale ma una delle sue interfacce di

rete. Un host multi-homed è un host che presenta due o più interfacce di rete (tipo un nodo di commutazione).

⁑INTERFACCIAMENTO LAN
Tipicamente un calcolatore si connette alla rete tramite una rete di accesso locale detta LAN (Local Area Network) che implementa strato 2 e 1 secondo il modello ISO-OSI.
Gli indirizzi LAN, detti MAC address, sono composti da 48bit e sono cablati nella scheda di rete. Questi numeri sono univoci a livello mondiale e per questo motivo i primi 3 byte identificano il costruttore mentre i secondi 3 numerano progressivamente le schede.

⁑FLUSSI DI COMUNICAZIONE
I flussi di comunicazione sono identificati da:

- IP sorgente
- IP destinazione
- Porta sorgente
- Porta destinazione



La coppia numero_IP : numero_porta si definisce end-point. Due flussi che hanno 1 o più numeri diversi sono sempre flussi diversi. Il MAC identifica il SAP di accesso alla rete

⁑IMPLEMENTAZIONE DEI SERVIZI INTERNET
Le comunicazioni tra calcolatori prevedono lo scambio di messaggi tra processi applicativi perché un messaggio diventa utilizzabile se è in esecuzione un processo applicativo che legge il messaggio e sa cosa farne. I server sono sempre attivi perché devono essere chiamati e di conseguenza sono i client che effettuano la chiamata e possono essere attivati solo quando serve.

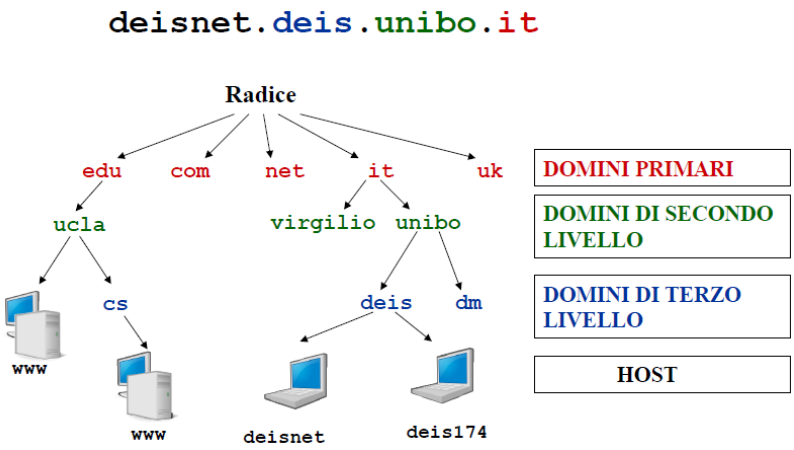
- Modalità Client-Server → gli host in rete vengono suddivisi in Client (ospitano le applicazioni che connettono ai server per ottenere risorse e informazioni) e Server (mettono a disposizione risorse di elaborazione e dati)
- Modalità Peer-to-peer → gli host di rete sono tutti equivalenti e fungono alternativamente sia da client che da server verso altri nodi mettendo a disposizione e utilizzando contemporaneamente risorse e informazioni di rete

→ Processo Server-Client: si predispone una apertura passiva del Server che apre l’end-point e aspetta di essere chiamato e quindi di ricevere una connessione. Il Client successivamente esegue una apertura attiva tentando di collegarsi al processo server di destinazione (quindi crea il suo end-point solo nel momento in cui devo fare la chiamta)

NOMI ED INDIRIZZI DI RETE: DNS

‡NOMI ED INDIRIZZI

Per comodità di utente ai numeri IP sono associati simbolici nomi come stringhe alfanumeriche separate da punti. Le stringhe non sono arbitrarie e le componenti del nome riflettono un'organizzazione gerarchica in Domini a cui vengono associati nomi convenzionali (per esempio .it è la stringa identificativa del dominio Italia) → i domini a loro volta sono suddivisi in sottodomini che non vengono registrati poiché sono i domini stessi responsabili della gestione di questi. La sequenza dei nomi di dominio di scrive a partire dal più esteso a destra. Il nome specifico dell'host è arbitrario mentre i nomi del dominio sono assegnati da IANA per evitare che vi siano due nomi uguali per host diversi. L'architettura è di tipo gerarchico e i vari domini sono stabiliti in modo standard con il country-code ISO e con altri metodi che distinguono ambiti applicativi (edu, gov, com, mil, org, net oppure contry-code it, fr, uk, ecc…).



_REGISTRO.IT

Il registro è l'anagrafe dei domini internet .it e soltanto qui è possibile chiedere, modificare o cancellare uno o più domini di questo tipo. Su richiesta è inoltre possibile associare un gruppo di indirizzi numerici ad un solo nome.

_SERVIZIO WHOIS

Con questo servizio è possibile verificare se ed a chi è assegnato un determinato nome di dominio

‡DNS

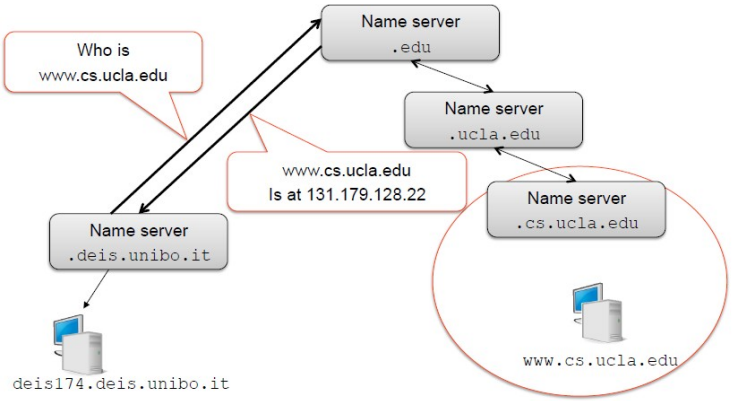
Per eseguire la ricerca degli indirizzi a partire dai nomi si utilizza il servizio automatico che è un database distribuito il quale associa

ad ogni nome il relativo indirizzo di rete. La consultazione del DNS avviene tramite opportuni server tipicamente trasparenti per l'utente. Il browser sa sempre cosa fare senza doverlo chiedere all'utente finale → in ricerca mando sempre pacchetti DNS per scoprire il numero che corrisponde al nome dato.
Poiché un database con tutti i nomi degli host sarebbe troppo complicato da gestire si opta per un database distribuito → lo spazio dei nomi è suddiviso in zone non sovrapposte che contengono uno o più sottodomini e ciascuna zona prevede un name server principale ed uno o più secondari. Ogni name server è a conoscenza degli indirizzi IP corrispondenti ai nomi degli host contenuti nella sua zona, di cui è responsabile.

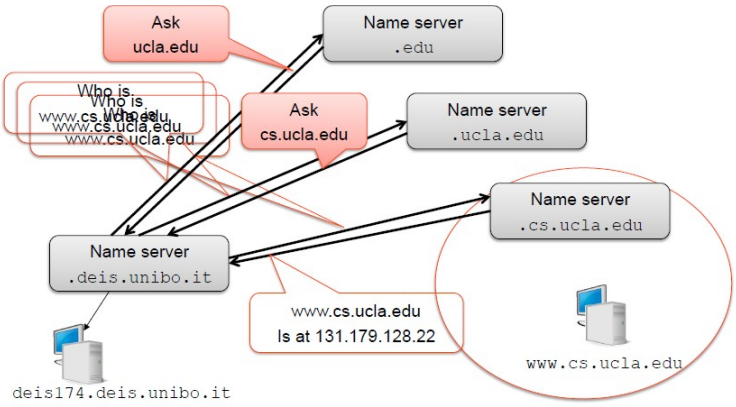
_NAME RESOLVER
Per convertire un nome in numero IP l'host deve essere equipaggiato con un programma specifico detto name resolver e deve essere configurato l'indirizzo IP del server DNS della zona di appartenenza.
Nell'host possono essere pre-configurate alcune corrispondenze nomi-numeri in un archivio locale e se l'applicazione deve risolvere un nome invoca il name resolver. Si possono verificare i seguenti casi:

- Il name resolver può risolvere il nome localmente grazie all'archivio → lo comunica quindi direttamente al numero IP
- Il name resolver non può risolvere il nome localmente → interroga il name server della zona che risolve cooperando con server DNS di altre zone

→ Risposta ricorsiva: il name server interrogato si preoccupa di risolvere il nome interrogando eventuali server di sotto-dominio e risponde alla richiesta



→ Risposta iterativa: il name server interrogato risponde indicando un name server di sotto-dominio a cui delega la risoluzione della richiesta



_DIG

Il comando dig ‘indirizzo come stringa’ analizza una richiesta DNS. Il server può rispondere in diversi modi in base alla tipologia di risorsa che viene inviata come risposta:

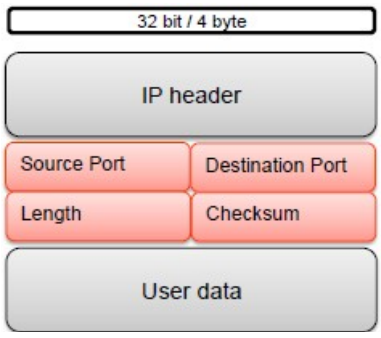
- A → modalità classica. Viene fornito il numero corrispondente al nome cercato
- NS → do il nome del dominio e nella risposta ho il server che ha la competenza per questo dominio
- CNAME → do un alias che corrisponde ad un valore canonico e nella risposta viene fornita l’associazione tra CNAME e nome vero e poi il numero
- MX → do il nome del dominio nella domanda e nella risposta ho il nome del server di posta del dominio richiesta (è un servizio significativo che non ha sempre lo stesso nome)

STATO DI TRASPORTO IN INTERNET

⁑TRASPORTO IN INTERNET
Esistono due protocolli di trasporto in internet usati in base alle necessità:
- TCP
- UDP

Lo strato di trasporto ha l'importante funzione di consentire la multiplazione, cioè di permettere a più processi applicativi di usare le funzioni di comunicazione in contemporanea. I protocolli usano il numero di porta per distinguere flussi dati di applicazioni diverse e controlla il comportamento del canale di comunicazione end-to-end → ha l'obbiettivo di garantire la qualità del trasferimento dati a livello del trasporto.
Le sue altre funzioni sono:
- Controllo dell'errore → gestisce gli errori di trasmissione e il problema della possibile perdita di unità informative
- Controllo di sequenza → controlla che l'ordine di consegna non sia stato alterato a causa di perdite o dei tempi di propagazione variabili
- Controllo di flusso e di congestione → controlla la compatibilità delle velocità di canale, ricevitore e trasmettitore

⁑UDP User Datagram Protocol
E' un protocollo di tipo connectionless per cui ogni messaggio è indipendente dagli altri. Non esiste conferma di ricezione dei pacchetti ed è quindi pensato per invio di blocchi di limitate dimensioni e comunicazione tra applicazioni per le quali non è richiesto controllo della qualità del trasporto. Il pacchetto UDP ha al suo interno indicate:



- Porta sorgente
- Porta destinazione
- Lunghezza del dato → permette di capire se ci sono degli errori nel caso non conoscessimo inizio e fine del dato

⁑TCP Transmission Control Protocol
Ha come obbiettivo il controllo della comunicazione end-to-end e full-duplex tra processi applicativi → garantisce affidabilità del trasporto attraverso la conferma della ricezione. Per il TCP si

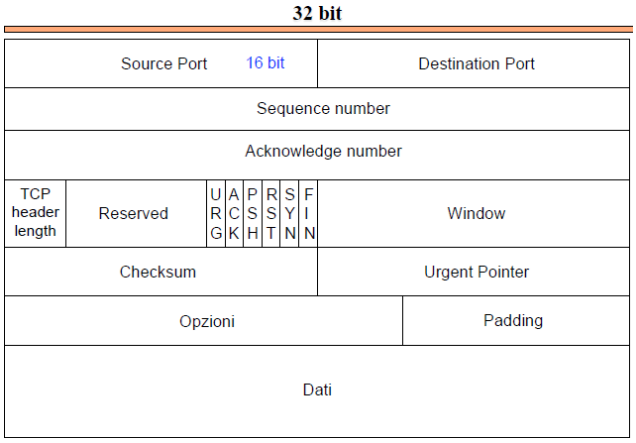
assume che lo strato di rete fornisca solamente un semplice ed inaffidabile servizio di trasferimento dei pacchetti connectionless. I pacchetti sono di maggior dimensione rispetto a quelli dell'UDP e non hanno dimensione fissa ma dipendente dal calcolatore. Il segmento TCP ha tuttavia dimensione massima detta MSS (Maximum Segment Size)

_SEGMENTI TCP

Il TCP incapsula i dati delle applicazioni in pacchetti detti segmenti che prevedono:

- Header standard di 20 byte
- Header variabile per negoziare delle opzioni
- Payload di dimensione variabile contenente i dati di applicazione

⁑FORMATO DEL SEGMENTO TCP

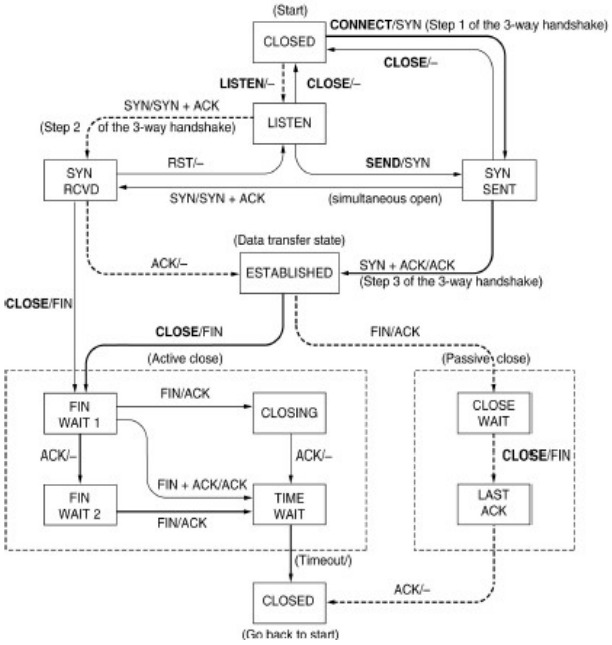


- Source/Destination port → numero della porta sorgente/destinazione
- Sequenze number → numero di sequenza del primo byte del pacchetto. Se è presente il bit SYN questo è anche il numero di sequenza inziale su cui sincronizzarsi. Permette di ordinare i pacchetti → per convenzione sappiamo che se riceviamo la risposta di ricezione di un pacchetto viene confermata automaticamente anche la ricezione dei precedenti. Il TCP infatti numera i byte di ogni pacchetto (se ho un pacchetto lungo 668 byte allora il pacchetto successivo sarà riconosciuto con il numero 669)
- Acknowledge number → se il bit ACK è ad 1 questo numero contiene il numero di sequenza del blocco di dati che il ricevitore si aspetta di ricevere (nella logica dell'esempio con 668 questo sarà il numero 669 che indica la risposta di corretta ricezione)
- TCP Header length → indica il numero di parole di 32 bit dell'intestazione TCP e quindi dice implicitamente dove iniziano i dati
- Reserved → campi vuoti per uso futuro
- Control bit → sono 6 bit di controllo:

    - URG → indica se si deve considerare il campo Urgent Pointer
    - ACK → posto ad 1 se si deve considerare il campo Acknowledge
    - PSH → ad 1 per la funzione di push
    - RST → ad 1 per resettare la connessione
    - SYN → ad 1 per sincronizzare i numeri di sequenza
    - FIN → ad 1 per indicare la fine dei dati
- Window → finestra del ricevitore quindi numero di byte che il ricevitore è disposto a ricevere partendo dal numero di sequenza contenuto nel campo acknowledge
- Checksum → ridondanza per la rilevazione degli errori
- Urgent Pointer → contiene un puntatore a dati urgenti eventualmente presenti nel pacchetto
- Options → contiene opzioni per la connessione
- Padding → sono bit aggiuntivi per fare in modo che l'intestazione sia multipla di 32 bit

⁑STATI DI CONNESSIONE TCP

La macchina a stati finiti del TCP è molto complessa:

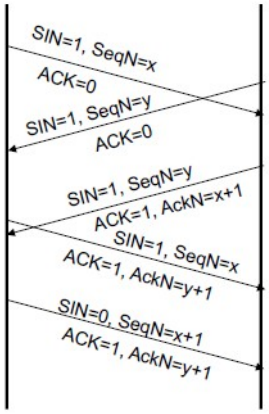


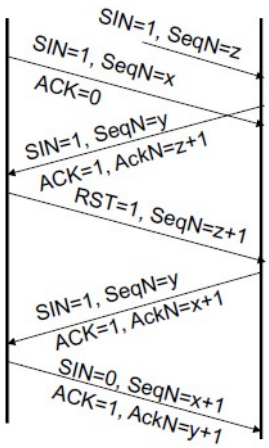
- LISTEN → il TCP apre la porta di comunicazione indicata e nel passaggio non vengono generati pacchetti (operazione tipo server)
- SYN RCVD → secondo step del TWH quindi passaggio in cui avviene la risposta di sincronizzazione con l'invio del SYN+ACK dal server
- SYN SENT → primo step del TWH quindi apertura lato client
- ESTABLISHED → step 3 del TWH. Da qui l'unità è in grado di trasmettere e ricevere dati
- CLOSE WAIT → ci si sposta con il flag FIN segnalando l'intenzione di chiudere la comunicazione

_APERTURA DELLA CONNESSIONE TCP

E' un momento critico poiché i segmenti di segnalazione possono essere persi, duplicati o ritardati. Si usa la tecnica del Three Ways

Handshake che usa i bit di flag e di numerazione in modo sinergico. I pacchetti di apertura della connessione preparano la comunicazione. Viene impostato il numero di partenza (indispensabile per sapere inizio e fine del pacchetto). Il server che riceve il dato capisce che è presente richiesta di comunicazione SYN=1 e risponde con un SYN,ACK dove dice che è pronto a ricevere informazione e indica il numero di inizio da cui verranno contati i byte dei pacchetti. Il client invia dunque una conferma di comprensione ACK=1 nella quale viene inserito il numero di inizio + 1. Questo tipo di transazione è indispensabile per capire se i pacchetti sono ricevuti in maniera ordinata anche se fa perdere tempo inizialmente (quindi se non ho bisogno di affidabilità conviene l'UDP).





Il TWH resiste all'instaurazione contemporanea di due connessioni ed ignora i pacchetti di apertura ritardati → se vengono ricevuti pacchetti di una sequenza non voluta mentre aspettiamo pacchetti della sequenza giusta viene mandato un reset per eliminare tutto.

_CHIUSURA DELLA CONNESSIONE TCP

La chiusura avviene in modalità soft-release cioè si cerca di realizzare la chiusura ordinata della connessione garantendo che non vadano persi dati → TCP sceglie di realizzare la chiusura con modalità simplex:

- Le due direzioni vengono rilasciate in modo indipendente
- Il TCP che intende terminare la trasmissione emette un segmento FIN=1:
    - Quando l'entità riceve l'Ack la direzione si considera chiusa
    - Se dopo un certo tempo non arriva alcun Ack il mittente del FIN rilascia comunque la connessione
- L'altra direzione può continuare a trasmettere dati finchè non decide di chciudere

‡AFFIDABILITA': ARQ DEL TCP

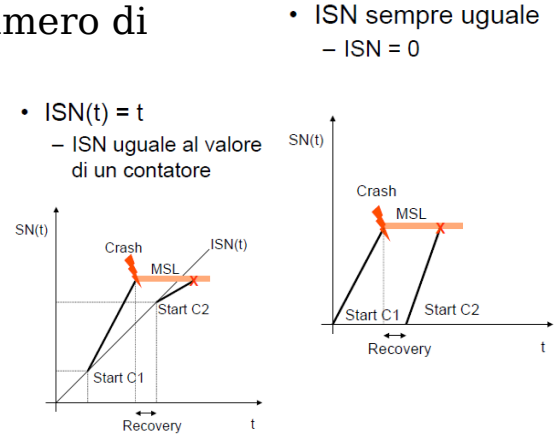
Il TCP cerca di garantire affidabilità del canale end-to-end usando:

- Numerazione sequenziale dei dati
- Conferma della ricezione di ogni byte

- Ritrasmissione dei dati

Tutto questo viene implementato con un protocollo di tipo Automatic Repeat Request. La numerazio delle sequenze è un problema potenziale per l'affidabilità poiché i numeri devono poter essere duplicati ma solo se si è sicuri che non esistano più in rete vecchi segmenti numerati con tali numeri → lo spazio di numerazione rende necessario limitare il tempo di vita dei segmenti e il Maximum Segment Lifetime o MSL è l'indicazione del tempo di vita massimo di un segmento (deve essere noto a priori).
All'apertura della connessione si sceglie il numero di sequenza iniziale ISN:

- Numero prefissato uguale per tutti → se replico il numero di sequenza ossia riparto da zero dopo il problema del crash avrò pacchetti con lo stesso numero perciò problemi di identificazione.
- Numero puramente casuale
- Numero legato al valore di un contatore



Il numero deve garantire che non ci sia duplicazione nell'uso dei numeri di sequenza e qualora non sia prefissato e costante deve essere concordato tra i due
host che aprono la connessione.

‡MESSAGGIO ACK DI CONFERMA

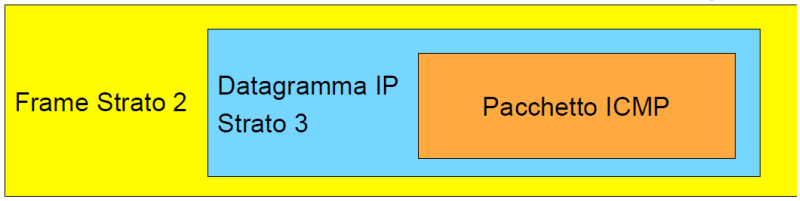
Gli ACK sono cumulativi e la conferma ha la forma di un normale messaggio TCP con il flag ACK=1 e può contenere dati oppure no (se non li contiene ho un datagramma IP di 40byte). Di default la conferma è esplicita e il ricevitore trasmette un ACK per ogni segmento ricevuto → al momento della ricezione corretta di un segmento il ricevitore può subito inviare un ACK oppure ritardarlo per minimizzare il numero di ACK, sempre rimanendo entro un tempo che eviti di fare scattare i time-out. Normalmente si produce un ACK ogni due segmenti.

_RTT Round Trip Time

Tempo necessario per effettuare un'andata e un ritorno sul canale e quindi tempo intercorso tra la partenza dell'ultimo bit di una trama e la ricezione dell'ACK relativo.

PROTOCOLLI ACCESSORI ALL'IP

⁑PROTOCOLLO ICMP
Il protocollo serve ad implementare le funzioni di gestione per la verifica e la notifica delle componenti funzionali della rete. E' un protocollo di controllo che segnala errori e malfunzionamenti ma non corregge nulla quindi non rende affidabile l'IP → gli offre solo un servizio in quanto IP lo usa per gestire situazioni anomale, incapsulando pacchetti ICMP in datagrammi IP. Formato pacchetti:



- Ip Header → 20-60 byte
- Message Type → definisce il tipo di messaggio, errore o richiesta di informazioni
- Message Code → descrive il tipo di messaggio
- Checksum → controlla i bit errati
- Additional Fields
- Data → intestazione e parte dei dati del datagramma che ha creato l'errore

→ Messaggi di errore:
- Destination Unreachable (Type=3) generato da un gateway quando la rete o l'host non sono raggiungibili oppure quando si presenta un errore sull'indirizzo dell'entità di livello superiore a cui trasferire il datagramma. Il messaggio viene rimandato al sorgente del pacchetto tramite il livello IP
- Time Exceeded (Type=11) generato dal router quando trova un datagramma con il TTL a 0 o da un host quando un timer si azzera in attesa dei frammenti per riassemblare il datagramma
- Redirect (Type=5) generato da un router per indicare all'host sorgente una strada più conveniente per la destinazione

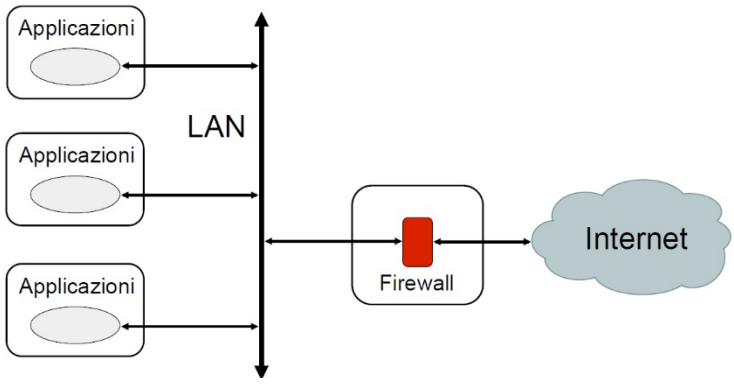
→ Messaggi di richiesta informazioni:
- Echo
- Echo Reply (Type=0). L'host sorgente invia la richiesta ad un altro host o gateway che deve rispondere immediatamente. Viene usato per determinare lo stato di una rete e dei suoi host, la raggiungibilità e il tempo di transito in rete
- Timestamp Request e Reply (Type=13,14). L'host sorgente invia alla destinazione un originate timestamp che contiene

l’istante in cui la richiesta è partita e l’host destinazione risponde inviando un receive timestamp (istante in cui è stata ricevuta la richiesta) e un transit timestamp (istante in cui la risposta è stata inviata) → serve a calcolare il tempo di transito della rete

_COMANDO PING: ping DEST
Permette di controllare se l’host DEST è raggiungibile dalla sorgente. SORG invia un pacchetto echo e se DEST è raggiungibile risponde con un echo reply.

_COMANDO TRACEROUTE: tracert DEST
Permette di conosce il percorso seguito dai pacchetti inviati da SORG e diretti a DEST. Vengono mandati pacchetti echo con un TTL progressivo da 1 a 30 e ciascun nodo intermedio decrementa TTL. Quando un nodo riceve TTL=0 invia a SORG un pacchetto time exceeded → SORG di conseguenza costruisce la lista dei nodi attraversati fino a DEST. Come output mostra il TTL, il nome DNS e l’indirizzo IP dei nodi intermedi e il Round Trip Time (RTT).

⁑PACKET FILTER, FIREWALL, PROXY

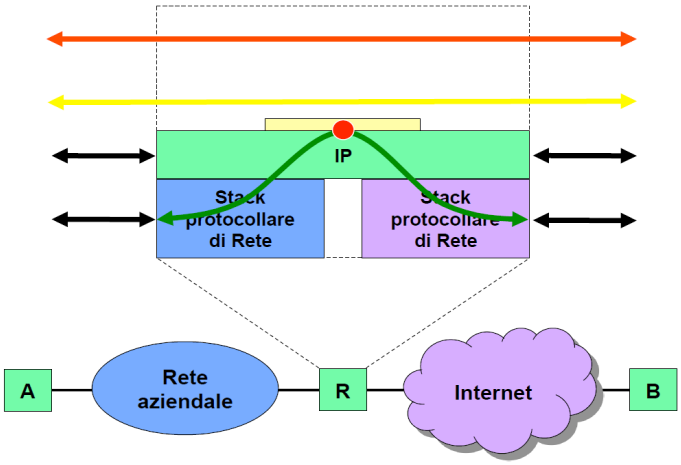


Ogni firewall può essere implementato come packet filter o come proxy server a livello di applicazione o di circuito. Un firewall è un filtro software/hardware che serve per proteggersi da attacchi indesiderati provenienti dall’esterno della rete. Può essere semplicemente un programma installato nel PC oppure una vera e propria macchina dedicata che filtra il traffico da e per una rete locale. Per ogni strato dello stack possono essere applicate politiche differenti. Tutto il traffico tra la rete locale e internet passa dal firewall ma solo il traffico autorizzato lo attraversa, garantendo comunque i servizi necessari.

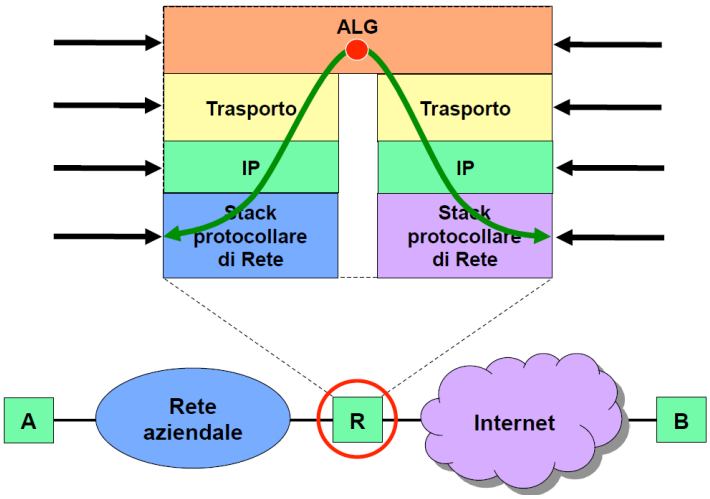
→ Politica di default:

- Default deny: tutti i servizi non esplicitamente permessi sono negati
- Default permit: tutti i servizi non esplicitamente negati sono permessi

_PACKET FILTER



Filtra i pacchetti seguendo politiche stabilite generalmente tramite filtri staticamente configurati. Si interpone tra rete locale e router e su di esso si configura appunto come un filtro sui datagrammi IP che vengono scartati sulla base degli indirizzi sorgente e destinazione, del tipo di servizio e dell'interfaccia di provenienza o destinazione.

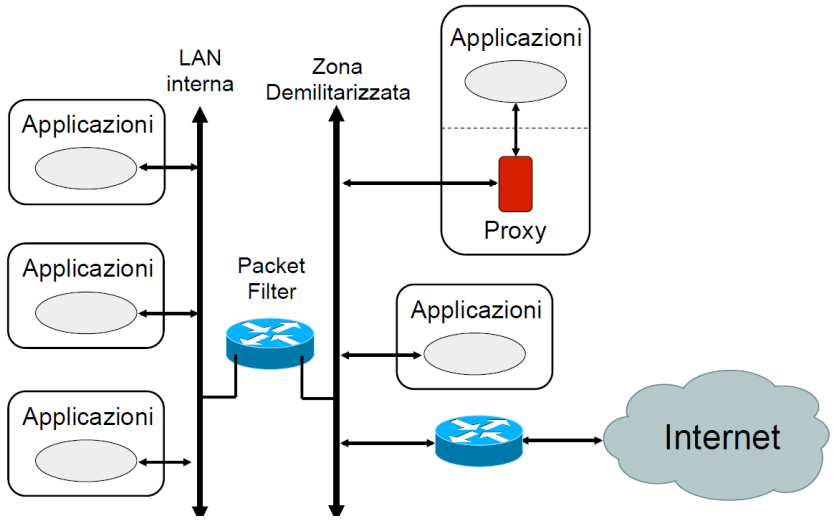


_PROXY SERVER

Realizza un server apposito detto proxy che serve a realizzare la comunicazione per tutti gli host. Tale serve intermedio serve ad evitare un flusso diretto di datagrammi tra internet e le macchine della rete locale e può essere applicato in due livelli:

- Livello di applicazione: proxy dedicato per ogni servizio
- Livello di circuito: proxy generico

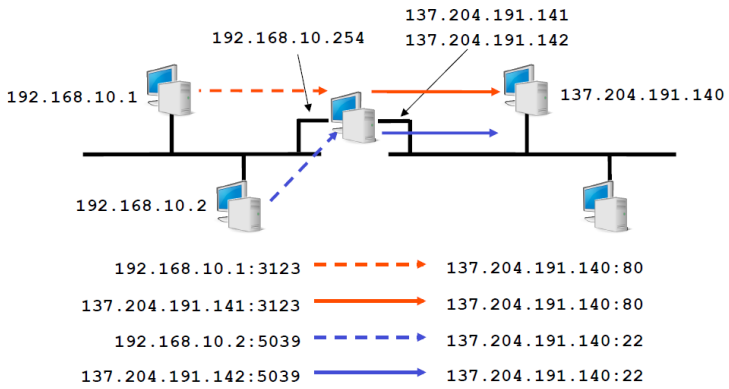
→ Configurazione packet filter e proxy combinata



Nella zona demilitarizzata divido le applicazioni con accesso relativamente facile e quelle più interne da ricevere. Possono essere inserite nella prima zona delle sonde che si accorgano se qualcosa va storto e se c'è un tentativo di accesso dando il tempo al packet filter di gestire la protezione. Alcune applicazioni hanno invece diretto accesso alla rete internet perché garantiscono le funzionalità base.

⁑NAT: Network Address Translation
Nella struttura della rete ci sono grandi zone a numerazione privata collegate alla rete pubblica. I gateway di bordo di queste aree hanno delle regole per tradurre il numero privato in uno pubblico quando un pacchetto deve uscire e per fare il contrario quando deve entrare. Il NAT non è un protocollo ma una funzionalità di rete che viene espletata dentro ad un nodo.
Con questa tecnica risparmio indirizzi IP pubblici e riutilizzo indirizzi privati → posso eseguire una multiplazione a divisione di tempo in cui un gruppo di numeri pubblici vengono condivisi tra un gruppo di numeri privati che aspettano ce ne sia uno libero. In questo modo ho la sicurezza degli host interni che non sono accessibili dall'esterno e possono nascondere la struttura interna della rete. Il NAT include un packet filter configurato dinamicamente.
Il NAT è trasparente per l'applicazione poiché viene modificata solo l'intestazione IP e TCP/UDP ma non il payload → può creare problemi perché alcune applicazioni non vi sono trasparenti: se le applicazioni contengono nel payload indirizzi IP e numeri di porta si presenta il problema che il server chiamato cerca di parlare con un client che non vede a causa del blocco del NAT (applicazioni FTP).

→ Conversione
Il NAT può convertire indirizzo IP soltanto oppure la coppia indirizzo IP, porta TCP/UDP. Questo mi permette di dare a due



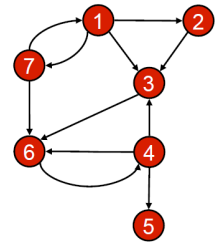
pacchetti in arrivo lo stesso numero IP ma due numeri di porta diversi mandandoli sullo stesso calcolatore. Tipicamente la conversione avviene dalla rete privata alla pubblica e le corrispondenze vengono via via registrate in una tabella → non è di solito possibile dall'esterno contattare per primi un host all'interno della rete privata poiché l'assegnazione avviene solo quando un host interno fa la richiesta di uscita. Se ho la necessità di farlo posso configurare nel NAT (aperto) associazioni statiche che permettono il contatto. Si può configurare il NAT in vari modi decidendo quante informazioni inserire nel collegamento → viene limitato il numero di calcolatori che possono parlare con il gateway e passare alla rete privata e quante informazioni possono essere controllate

‡ARP: Address Resolution Protocol
E' un meccanismo che permette di ricavare l'indirizzo fisico di un host sulla stessa rete mediante una richiesta che contiene l'indirizzo IP del destinatario. Per comunicare sulla stessa rete LAN infatti si utilizzano i numeri MAC e se un host deve comunicare con un altro ha la necessità di conoscere questo numero → va ricavato conoscendo l'indirizzo IP.
Viene mandata una trama broadcast ARP request che contiene l'indirizzo IP del nodo destinazione e viene letto da tutte le macchine della rete locale. Il destinatario risponde con un messaggio ARP reply che contiene il proprio indirizzo fisico permettendo all'host sorgente di associare l'appropriato indirizzo fisico all'indirizzo IP destinazione → ogni host mantiene una tabella cache ARP con le corrispondenze fra indirizzi logici e fisici

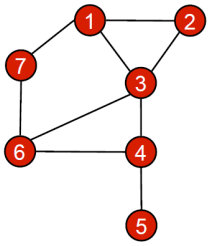
ALGORITMI DI ROUTING

⁑APPLICAZIONE DELLA TEORIA DEI GRAFI
Ogni rete è un insieme di nodi di commutazione e viene rappresentata con i modelli matematici della teoria dei grafi:

- Insieme V finito di nodi
- Un arco si definisce come coppia di nodi (i,j) appartenenti a V → in questo caso i e j sono adiacenti
- Un grafo è la coppia (V,E) dove E è l'insieme degli archi. Il grafo può essere:
  - Orientato
  - Non orientato



Grafo orientato



Grafo non orientato

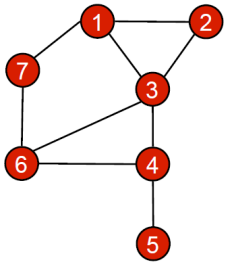
→ Grafo pesato: grafo su cui in ogni arco è presente un numero chiamato peso o costo. In generale quindi sul grafo orientato i pesi sono fondamentali perché variano anche in base alla direzione. Per due nodi non collegati si assume che il peso sia $+\infty$.

→ Cammino: sequenza di nodi che parte dal nodo u al nodo v. Si definisce semplice se non passa mai più di una volta sullo stesso nodo, non generando quindi dei cicli. Il problema dei cicli è fondamentale per i meccanismi di rete e viene gestito proprio dal TTL.

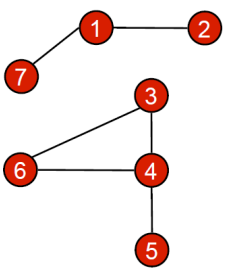
→ Ciclo: cammino in cui il nodo di partenza e di fine è lo stesso. Un grafico si definisce aciclico se non contiene cicli. In un grafo aciclico se tolgo un arco divido il grafo in due sottografi. La struttura a grafo aciclico sarebbe ottima per i sistemi di rete ma è molto poco resistente ai guasti.

→ Connettività dei grafi: più un grafo è connesso più è resistente ai guasti, ma resta anche allo stesso rischioso per la possibile creazione di cicli:

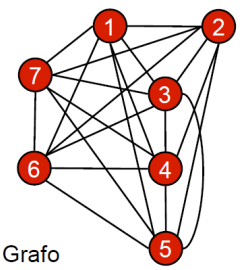
- Grafo connesso: per ogni coppia di nodi c'è sempre un cammino
- Grafo completamente connesso: tutti i nodi sono adiacenti tra di loro



Grafo connesso



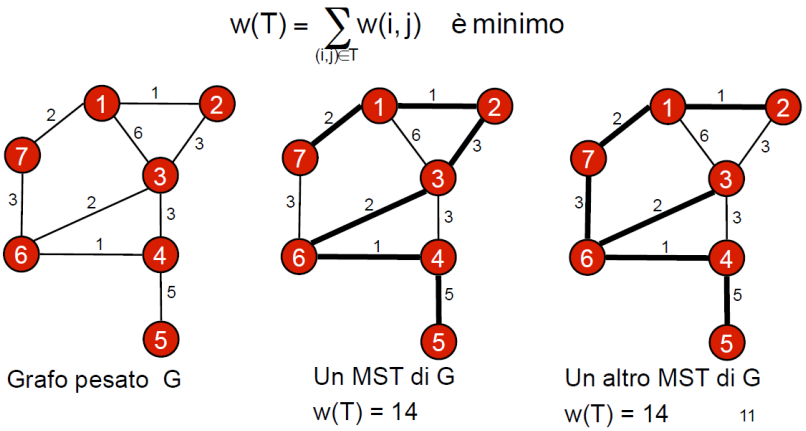
Grafo non connesso



Grafo completamente connesso

→ Albero: grafo connesso e aciclico (se rimuovo qualsiasi arco il grafo smette di essere connesso). Ogni coppia di nodi è connessa da un unico cammino semplice

→ Spanning tree: è un sotto-grafo che si può trovare in un grafo connesso non orientato ed è l'albero che contiene tutti i nodi del grafo. Si può utilizzare per creare una rete molto connessa e fare in modo che i nodi ne vedano solo una sotto-porzione. In questo modo prevengo i cicli il più possibile ma ho anche resistenza ai guasti perché posso sostituire archi non funzionanti. Lo spanning tree è il modo in cui viene vista la struttura per l'instradamento.



Dato un grafo pesato connesso e non orientato si può definire un minimum spanning tree che è l'albero di ricoprimento di peso minimo.

‡ALGORITMI DI CALCOLO DEL MST E SP

Gli algoritmi sono di tipo greedy, cioè inseriscono ad ogni passo un arco ad un sotto-grafo che è un sotto-grafo anche del MST. I principali algoritmi sono:

- Kruskal
- Prim

Poiché nei nodi si cerca di creare tabelle di instradamento che cerchino i percorsi a lunghezza minima si usando gli shortest path che sono i cammini a peso minimo tra due nodi. Il percorso a lunghezza minima trovato sarà poi intrinsecamente ottimale e senza cicli a patto che i pesi siano tutti positivi

→ Principio di ottimalità: dato un grafo pesato orientato e un SP va v a k qualsiasi cammino (i,…,j) contenuto in p è anche esso un SP tra i e j.

Esistono algoritmi che servono a trovare un SP da s verso ogni altro nodo v usando la tecnica del rilassamento dell'arco. Questi algoritmi si differenziano per il modo con cui effettuano il rilassamento:

- Bellman-Ford → popola la tabella di instradamento. Ha la funzionalità che termina sempre in un numeri di passi massimo pari al numero di archi presenti.
- Dijkstra

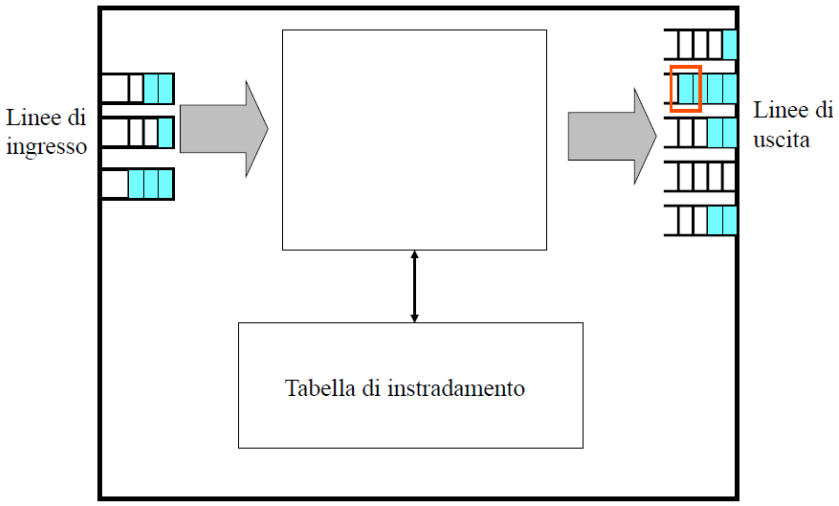
_BELLMAN-FORD DISTRIBUITO
Viene eseguito da tutti i nodi che devono essere consci dei valori da usare durante l'algoritmo. Per fare ciò viene usato un protocollo specifico per la comunicazione. Ogni router esegue i calcoli sul costo dell'instradamento e quindi determina il percorso sulla base dei pesi dei collegamenti. Poiché i nodi conoscono solo le proprie interfacce e le proprie caratteristiche si procede per soluzioni parziali ad ogni passo.
L'algoritmo viene iniziato e dovrebbe essere portato avanti in modo sincrono → ogni volta che si ha un cambiamento nella lunghezza degli archi o nella tipologia della è previsto un meccanismo per resettare i nodi e riavviare la procedura per calcolare nuove tabelle.
Un problema che non è risolvibile è quello di sincronizzazione poiché esiste il tempo di propagazione che non è eliminabile → l'algoritmo viene quindi applicato in modo asincrono: il nodo aggiorna periodicamente la sua distanza minima da quello iniziale e trasmette il valore stimato a tutti gli altri nodi.
Si dimostra che se:
- I nodi non interrompono mai
- I messaggi di trasmissione delle stime delle distanze minime sono dimenticati dalla rete dopo un certo tempo
- Non esistono cicli di lunghezza negativa o nulla

Allora l'algoritmo converge sempre ai valori giusti dopo un tempo sufficientemente lungo anche se le lunghezze di uno o più archi vengono cambiate

_DIJKSTRA
L'algoritmo crea un insieme vuoto S e ad ogni passo vi inserisce un nodo che ha distanza minima e rilassa tutti gli archi uscenti da esso e diretti verso altri elementi del grafo iniziale T=I. Si toglie quindi il nodo dal grafo T e si inserisce in S → l'algoritmo termina quando T=0 e S=I. E' un algoritmo che computazionalmente è più performante ma non è possibile risolverlo in maniera distribuita e per questo motivo non è implementabile se la topologia non è del tutto nota a priori.

ROUTING NELLE RETI IP

⁑INSTRADAMENTO E ROUTER



Ad una generica rete di telecomunicazioni si può sempre associare un grafo orientato e pesato → i nodi rappresentano i terminali e nodi di commutazione mentre gli archi i collegamenti con il loro relativo costo. Ogni router ha sempre varie linee di ingresso e di uscita, in numero non necessariamente uguale tra loro, e si configura come un nodo di commutazione a pacchetto specializzato per l'uso del protocollo IP.

La funzione di instradamento gestisce il movimento dei pacchetti e si basa sulla tabella di instradamento per farlo nel modo ottimale → mentre la funzione di instradamento è modificabile dal costruttore del router, la tabella contiene sempre le stesse informazioni anche se viene usata da apparati diversi. Con instradamento si intende il processo con cui si prende un pacchetto da una linea di ingresso e lo si porta su una linea di uscita. Vi sono poi anche particolari meccanismi di accordamento per permettere la gestione di due pacchetti contemporanei diretti alla stessa uscita.

→ Store-and-forward: ogni pacchetto che entra viene verificato e memorizzato estraendone le informazioni di instradamento e confrontandole con la tabella del router. Il pacchetto viene dunque successivamente inserito nella linea di uscita in attesa dell'effettiva ritrasmissione.

→ Classificazione dei router:

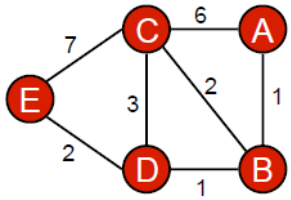
- SOHO (Small Office and HOme): utilizzati in domestico o in piccoli uffici hanno poche porte e una interfaccia LAN
- Router di accesso: hanno relativamente poche interfacce fisiche ma ben strutturate e divise in LAN e WAN. Presentano abbastanza porte a velocità media
- Enterprise router: servono per l'interconnessione tra LAN ed hanno poche porte ad alta velocità

- Backbone router: usati per le reti di trasporto tra domini. Hanno poche porte ad alta velocità e sistemi di garanzia dell'affidabilità

→ Funzioni dei router:
- Routing: scambio di informazioni tra router, elaborazione locale e popolazione delle tabelle di routing
- Forwarding: indirizzamento dei pacchetti IP con table lookup ed header update.
- Switching: trasferimento da interfaccia di input a interfaccia di output
- Trasmissione: trasmissione effettiva del datagramma sul mezzo fisico

⁑PROTOCOLLI DISTANCE VECTOR
Si basano sull'algoritmo Bellman-Ford in versione dinamica e distribuita (versione Ford-Fulkerson). Ogni nodo dialoga e scopre i suoi vicini (interfaccia attiva=router vicino) per poi calcolare la propria distanza da essi, valutando anche il peso che può essere impostato secondo due modalità:
- Sempre ad 1 → prediligo alla fine percorsi con minor numero di archi
- Assegnato in base alla preferenza che si ha nell'uso di una interfaccia piuttosto che un'altra

Ad ogni passo dell'algoritmo il nodo invia ai propri vicini un vettore contenente la stima della sua distanza dagli altri nodi della rete che conosce → il nodo può così eseguire una operazione di rilassamento verso ogni altro nodo ed aggiornare la stima delle distanze e il next-hop verso le varie destinazioni.



Distance Vector iniziali: DV(i)= {(i,0)}, per i = A,B,C,D,E

Distance Vector dopo la scoperta dei vicini:
DV(A) = {(A,0), (B,1), (C,6)}
DV(B) = {(A,1), (B,0), (C,2), (D,1)}
DV(C) = {(A,6), (B,2), (C,0), (D,3), (E,7)}
DV(D) = {(B,1), (C,3), (D,0), (E,2)}
DV(E) = {(C,7), (D,2), (E,0)}

1. A riceve DV(B)

dest	Costo, next hop
A	0
B	1, B
C	3, B
D	2, B

Tabella di A

2. A riceve DV(C)

dest	Costo, next hop
A	0
B	1, B
C	3, B
D	2, B
E	10, B

Tabella di A

3. B riceve DV(D)

dest	Costo, next hop
A	1, A
B	0
C	2, C
D	1, D
E	3, D

Tabella di B

4. A riceve DV(B)

dest	Costo, next hop
A	0
B	1, B
C	3, B
D	2, B
E	4, B

Tabella di A

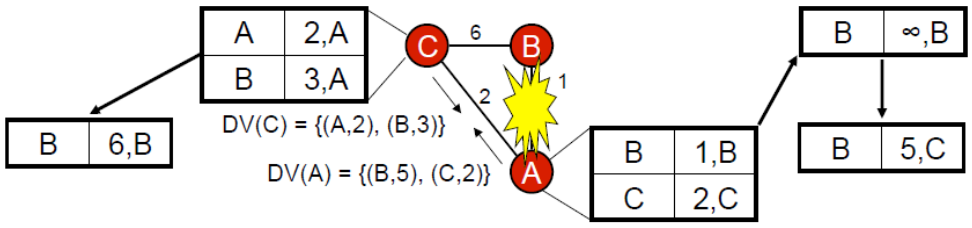
Le tabelle utilizzate nell'algoritmo di routing sono riconducibili a quelle di instradamento:

- La destinazione è l’indirizzo IP comprensivo di netmask
- Il next-hop è l’indirizzo dell’intefaccia e quello di gateway

‡PROBLEMI DEI PROTOCOLLI DV

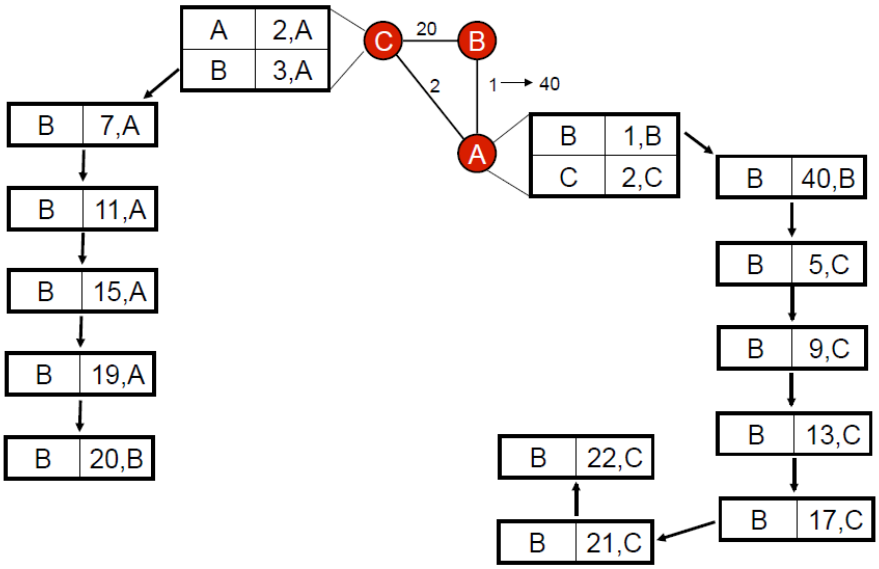
_COLD START E TEMPO DI CONVERGENZA
Allo start le tabelle dei singoli nodi contengono solo l’indicazione del nodo stesso a distanza 0 → i primi distance vector scambiati contengono solo questa informazione. Successivamente con i vector successivi si creano tabelle sempre più complete e al più dopo un numero di passi pari al numero di nodi della rete si ha convergenza dell’algoritmo → se lo stato della rete cambia in un tempo inferiore a quello di convergenza l’algoritmo prosegue e si ritarda solamente il tempo di convergenza

_BOUNCING EFFECT



Si ha quando il link tra due nodi A e B cade. A e B si accorgono che il collegamento non funziona e immediatamente pongono ad infinito la sua lunghezza. Se alti nodi nel frattempo hanno inviato anche i loro vettori delle distanze ci possono essere delle temporanee incongruenze che durano in base alla complessità della rete. Queste incongruenze possono dare luogo a dei cicli per cui uno o più nodi si scambiano datagrammi fino a che non esaurisce il TTL o finchè non si ha di nuovo convergenza.

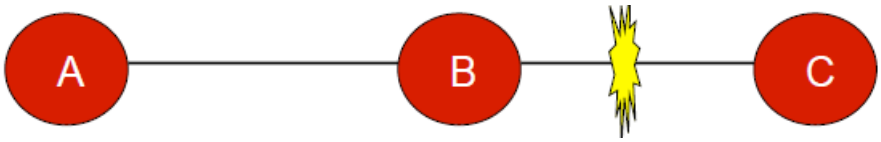
_CONVERGENZA LENTA



In alcune situazioni il cambiamento del peso di un arco può portare a convergere in un tempo molto più lungo del necessario.

_COUNT TO INFINITY

Se inizialmente tutti i collegamenti valgono 1 e si rompe il collegamento BC il conteggio delle distanze va avanti fino all'infinito. B riceve il DV da A con l'informazione $D_{AC}=2$ per cui calcola che la sua distanza da C è quella di A sommata a quella che c'è tra lui ed A e imposta questa come distanza reale da C perché non riceve il DV da C con l'informazione che la sua distanza è pari ad 1 poiché il collegamento non esiste più. Si va quindi avanti con B che è convito che per raggiungere C debba passare da A ed entrambi i nodi che ad ogni passo aumentano di 1 la loro distanza da C passando l'uno dall'altro. Poiché la cosa teoricamente potrebbe non terminare si impone un valore massimo di distanza → nel momento in cui si raggiunge si suppone che il nodo non sia raggiungibile.



→ Risoluzione con split-horizon: è una risoluzione solo parziale. Se A instrada pacchetti verso X passando da B allora non ha senso che B conosca la distanza di A da X poiché non avrebbe senso per lui cercare di raggiungere X tramite A. Un router deve quindi inviare informazioni diverse a vicini diversi. Ci sono due modi di implementarlo:

- Semplice → A omette la sua distanza da X nel DV per B

- Poisonus reverse→ A inserisce la sua distanza da X ma la pone uguale a ∞

→ Risoluzione con trigger update: è relativa alla tempistica con cui inviare i DV ai vicini. Si impone che ad ogni cambiamento della tabella di instradamento di un nodo questo mandi a tutti i suoi vicini le nuove informazioni. In questo modo si evita il problema che inviando i dati periodicamente un DV contenente un cambiamento della topologia parta in ritardo e venga sopravanzato da informazioni vecchie di altri nodi

Tutti questi rimedi non sono risolutivi del tutto perché vi sono ancora situazioni patologiche dove la convergenza è troppo lenta o non avviene per niente.

⁑PROTOCOLLI EVOLUTI

_PATH VECTOR
Per evitare il problema occorre evitare di creare cicli nel percorso dei pacchetti. I protocolli più evoluti funzionano a path vector → il vettore che ogni router manda ai vicini contiene, oltre alle distanze dagli altri nodi, anche l'intero cammino che il pacchetto deve seguire per raggiungerli. Il router così ignora i cammini in cui compare lui stesso e si evitano a prescindere cicli anche se sono di più le informazioni da scambiare.

_LINK STATE
Ogni nodo si procura una immagine della topologia completa della rete e sulla base di questa calcola le tabelle di routing con un determinato algoritmo. Lo scopo del protocollo è quello di permettere la creazione dell'immagine di rete tramite la scoperta dei nodi vicini, la raccolta di informazioni e la diffusione agli altri nodi. Per fare ciò si usano due tipi di pacchetti:
- Hello packet → usato per comunicare con i vicini e imparare gli indirizzi
- Echo packet → usato per conoscere la distanza dai vicini

In seguito ogni router crea un LSP (Link State Packet) che contiene la lista dei suoi vicini con le lunghezze dei collegamenti per raggiungerli

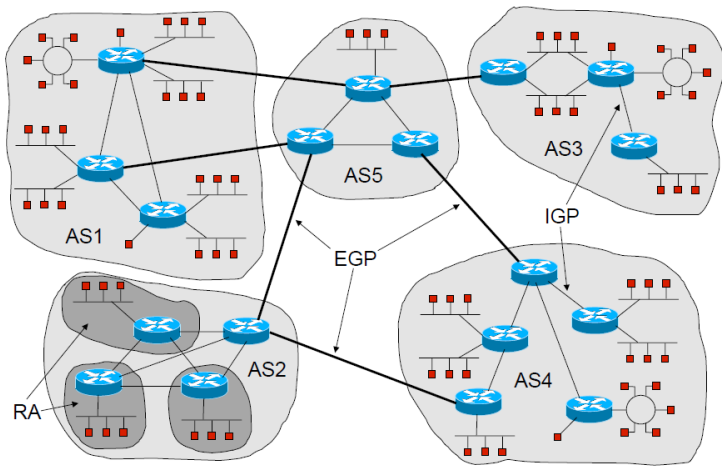
_ROUTING GERARCHICO

Nel caso di reti di grandi dimensioni non è possibile gestire tabelle di routing complete. Per questo motivo il routing avviene in maniera gerarchica:

- La rete viene divisa in sottoporzioni
- I router all'interno di ognuna effettuano l'instradamento relativo alla sola area
- Si inseriscono router di bordo a cui vengono inviati i pacchetti che devono uscire e solo loro conoscono la topologia fuori dall'area → si occupano del passaggio dei pacchetti da un'area all'altra

AUTONOMOUS SYSTEM

‡ROUTING GERARCHICO IN INTERNET



In internet le aree di routing sono suddivise tra di loro per garantire maggiore efficienza. La suddivisione crea gli Autonomous System, gruppi connessi di una o più reti IP (classless) gestite da uno o più operatori ma con identiche e ben definite politiche di routing. Ogni area viene poi divisa a sua volta in porzioni dette Routing Area (RA) interconnesse da zone di backbone → ogni network IP è contenuto in un AS o in una RA secondo la classe o il CIDR. Al loro interno i protocolli adottati e le politiche di routing sono decise autonomamente dagli AS e i vari enti di gestione devono accordarsi su quali protocolli usare per router che collegano i vari AS:

- IGP (Interior Gateway Protocol) → protocolli di routing all'interno degli AS
- EGP (Exterior Gateway Protocol) → protocolli di routing tra i vari AS. Essi vedono una complessità minore poiché ignorano tutta la struttura interna ai vari AS ma si occupano solo della loro interconnessione.

Nella realtà sulla stessa area geografica ho più di un AS poiché ogni operatore ha il proprio. I vari AS sono su piani paralleli e sono presenti degli internet exchange (si chiamano NAP, neutral access point), enti che si preoccupano di collegare i vari piani dei diversi operatori. Ogni livello si collega ad essi ed è quindi possibile la comunicazione attraverso i vari piani.

_WHOIS

Le politiche di routing degli AS sono contenute in registri consultabili con il comando whois → viene visualizzata come risposta l'AS a cui appartiene la route specificata e il registro che la contiene. Poiché le politiche di sicurezza per gli AS sono più vincolanti spesso vengono anche indicati gli AS da cui la destinazione può accettare route e quelli verso cui può annunciarne

ROUTING IN INTERNET: EGP, BGP

⁑EGP (EXTERIOR GATEWAY PROTOCOL)
Sono i protocolli che si usano per la comunicazione tra gli AS e al contrario dei protocolli per l'interno, dove si persegue l'ottimizzazione dei percorsi, nel routing tra diversi AS si deve tenere conto delle politiche di instradamento → ogni AS vuole rimanere indipendente dagli altri e alcuni non vogliono permettere ad altri di instradare il traffico attraverso le loro reti. Il protocollo ha tre funzionalità principali:

- Neighbor acquisition → verifica se esiste un accordo per diventare vicini
- Neighbor reachability → monitora le connessioni con i vicini
- Network reachability → scambia informazioni sulle reti raggiungibili da ciascun vicino

Il protocollo è simile ad un DV poiché le informazioni inviate ai vicini sono di raggiungibilità, anche se non sono specificate regole per definire le distanze per cui non viene usato il criterio della distanza minima come indice di ottimalità.
Essendo progettato sulla base di una specifica topologia funziona bene solo per le topologie ad albero e non per quelle a maglia complessa a causa della presenza di cicli → la convergenza del protocollo può essere quindi molto lenta e non si ha veloce adattamento alle modifiche della topologia stessa. Non sono inoltre presenti meccanismi di sicurezza.

⁑BGP (BORDER GATEWAY PROTOCOL)
E' stato concepito come sostituto di EGP e i router BGP scambiano informazioni attraverso connessioni TCP (porta 179) chiamate sessioni → le connessioni sono affidabili e sempre aperte. Esistono due tipi di sessioni:

- Esterne → instaurate tra router BGP di AS diversi
- Interne → instaurate tra router BGP appartenenti allo stesso AS

Le informazioni scambiate riguardano la raggiungibilità di reti IP secondo lo schema classless CIDR.
BGP è un protocollo di tipo path vector quindi nel vettore che viene inviato si elencano tutti gli AS da attraversare per raggiungere la destinazione → si evitano i cicli perché quando un router di bordo di un AS riceve un path vector controlla se il suo AS è già elencato all'interno e in tal caso non considera il vettore per evitare di creare cicli.

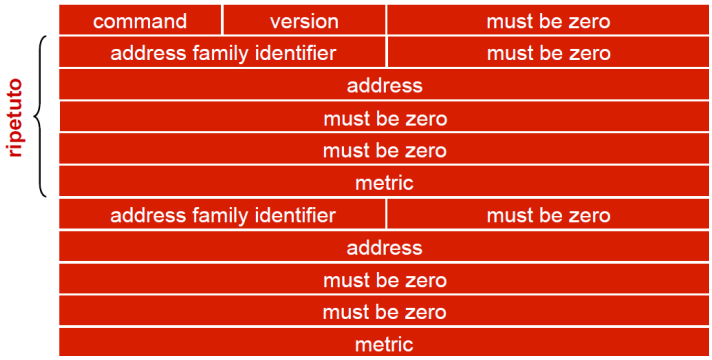
ROUTING IN INTERNET: IGP

‡ROUTING INFORMATION PROTOCOL: RIP
E’ un protocollo distance vector di vecchia implementazione che utilizza due tipi di messaggi:

- Request: chiede informazioni ai nodi vicini
- Response: serve per inviare informazioni di routing (DV)

I messaggi RIP sono trasportati da UDP e usano la porta 520 in trasmissione e ricezione.

→ Response: invia informazioni di routing periodicamente (ogni 30 secondi con scarto da 1 a 5 secondi), come risposta ad una richiesta esplicita oppure quando una informazione di routing cambia



La struttura del pacchetto è basata su parole di 32bit.
Nel campo command viene indicato response/request.
Nel campo address family identifier viene indicato il codice dell’IP.
Nel campo address viene invece indicato l’indirizzo IP.

_TABELLA DI ROUTING
Ogni riga della tabella di routing contiene:

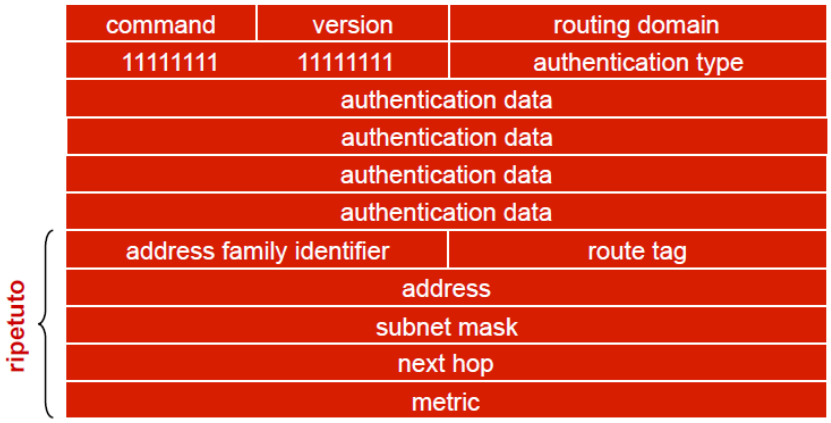
- Indirizzo di destinazione a 32bit
- Distanza dalla destinazione in termini di hop-count (peso=1). La distanza massima per il RIP è 16 e questo dimostra che è un protocollo per reti relativamente piccole
- Next-hop sul percorso verso la destinazione → router a cui inviare i datagrammi per la destinazione
- Due contatori:
  - Timeout → se una route non viene aggiornata dopo TO secondi la sua distanza è posta infinito (si ipotizza perdita di connettività)
  - Garbage-collector timer → dopo altri GC secondi la route viene eliminata del tutto dalla tabella

L’aggiornamento avviene in questo modo:

- A riceve da B un response, controlla la correttezza dei dati considerando solo le voci con distanze < ∞ e calcola la distanza come d+1
- Si procede aggiornando la tabella:
    - Se è gia presente la destinazione si modifica nel caso la nuova distanza sia inferiore a quella già presente e si setta B come next-hop
    - Se non è presente la destinazione si crea un nuova entry della tabella

Il RIP fa uso di split horizon e trigger update ma non supporta il CIDR (viene indicato indirizzo IP ma non la netmask) quindi non riesce a vedere aggregazioni di numeri diversi in punti diversi della rete. E' un protocollo insicuro poiché ogni trasmissione dalla porta UDP 520 viene considerata come trasmissione autorizzata

⁑MIGLIORAMENTO DEL RIP: RIP2

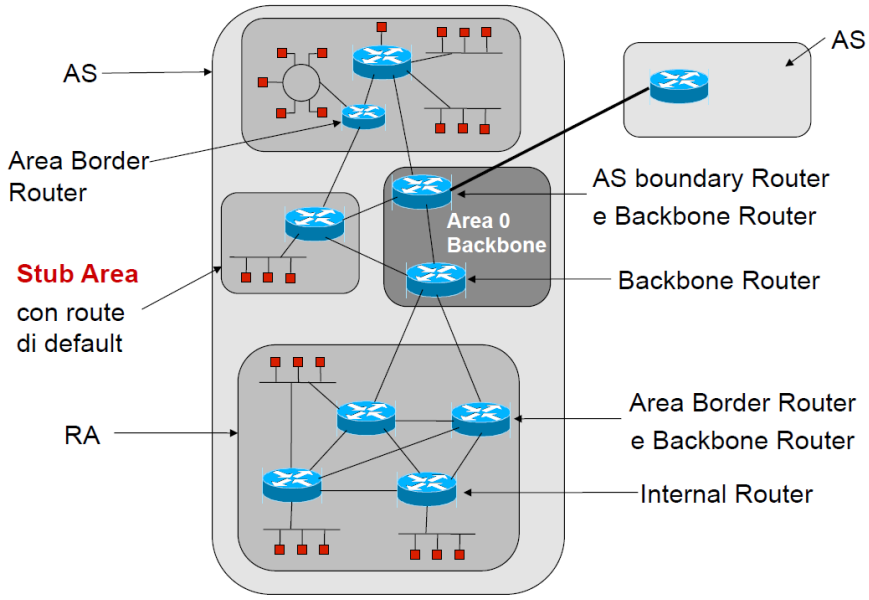


Viene introdotto un meccanismo per il subnetting e il CIDR, unitamente ad uno di autenticazione. La nuova versione ha compatibilità verso il basso poiché RIP1 ignora tutte le entry con i campi riservati diversi da zero. Il campo subnet mask permette di indicare sottoreti o indirizzamento CIDR e sono inoltre presenti campi per l'autenticazione.

⁑OPEN SHORTEST PATH FIRST: OSPF

IGP più diffuso. Protocollo di tipo link-state che invia dei link-state advertisement a tutti i router della rete. Viene incapsulato direttamente nell'IP utilizzando il campo protocol della sua intestazione.

Nelle aree di routing ogni AS viene diviso in routing area interconnesse da un backbone. Ogni area risulta separata dalle altre e si comporta come entità indipendente, collegandosi alle altre tramite router apposta. Secondo OSPF i router vengono divisi in:



- Internal router → router interni alle aree
- Area border router → scambiano informazioni tra aree
- Backbone router → scambiano informazioni con il backbone
- AS boundary router → scambiano informazioni con altri AS con il protocollo EGP

La topologia dell'area è ad albero con l'area di backbone (ha solo router e nessun calcolatore) in alto e tutte le altre a lei collegate (albero puro se ogni RA connette il backbone con una sola connessione e quindi non ho rischio di cicli almeno a questo livello). Si sconsiglia di creare collegamenti tra aree senza passare dal backbone. Le informazioni che permettono il collegamento intra-area vengono scambiate solo internamente mentre quelle che occorrono per le route inter-area vengono mandate anche oltre il confine degli RA → in questo modo si rimane a metà tra il link-state e il distance-vector. Il protocollo link-state dunque crea una ripartizione della topologia e gerarchia delle aree → distinguo le rotte e semplifico gli scambi informativi tra i router ai due lati di un bordo. Rimane comunque il problema che ogni link-state packet deve girare tutta la rete e se essa risulta complessa aumenta la quantità di dati generati periodicamente.

→ Tipologie di route: le route che possono crearsi sono:

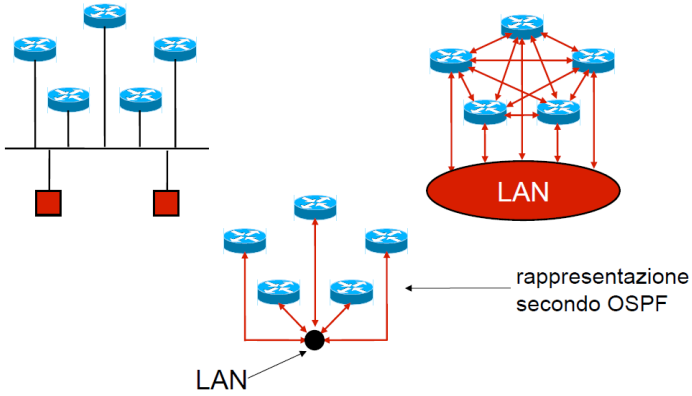
- Intra-area → aggiornamento delle informazioni di routing pertinenti all'area
- Inter-area → aggiornamento delle informazioni di routing pertinenti ad aree diverse da quella considerata. Ci sono situazioni nelle quali devo richiedere ai router di fare passare dentro un'area anche le informazioni inter-area, per esempio

nel caso in cui un'area sia collegata al backbone con un solo collegamento e quello cada.
- Esterne → aggiornamento delle informazioni di route provenienti da altri protocolli al di fuori del dominio OSPF

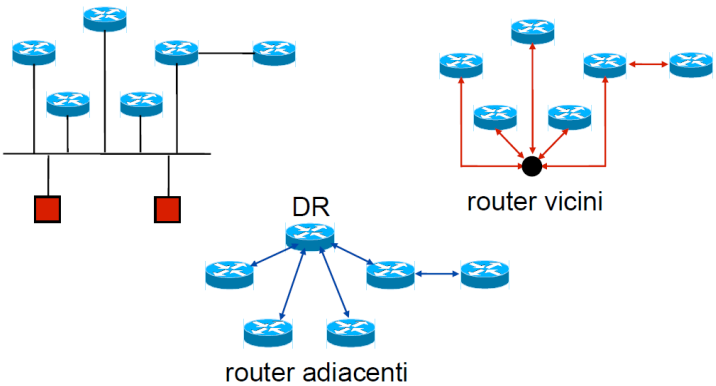
→ Ulteriori caratteristiche:
- Bilanciamento del carico → se un router ha più percorsi verso una destinazione ed essi sono di uguale lunghezza allora vi ripartisce i dati equamente
- Autenticazione → è prevista autenticazione crittografica e con password
- Routing dipendente dal grado di servizio → i router scelgono il percorso sulla base dell'indirizzo e del campo Type of Service tenendo conto che diversi percorsi significano diversi gradi di servizio

_VICINANZA E ADIACENZA TRA ROUTER

Due router si definiscono:
- Vicini → sono connessi alla stessa rete e possono comunicare direttamente punto-punto o punto-multipunto
- Adiacenti → si scambiano informazioni di routing



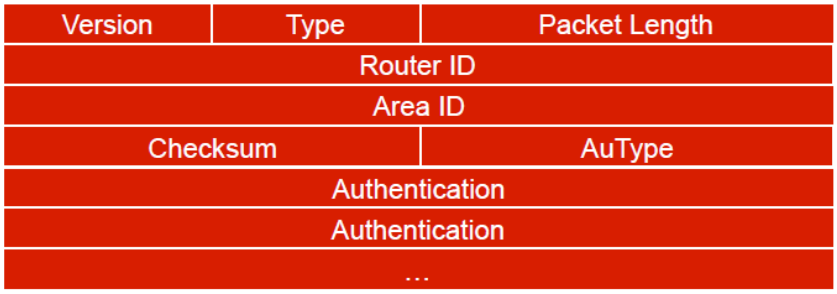
Nelle reti ad accesso multiplo, per evitare di mandare a tutti le informazioni OSPF di routing si sceglie un Designated Router tra quelli vicini (e uno di backup per sicurezza) → ogni router della LAN risulterà adiacente solo al DR e lo scambio delle informazioni di routing avviene solo tra router adiacenti (quindi DR fa sempre da tramite). Il router manda sempre le informazioni al DR che poi le gira a tutti gli altri con l'indirizzo di multicast della rete. Inoltre è il DR che comunica la raggiungibilità di router e LAN al mondo esterno.



_IDENTIFICAZIONE E PRIORITA'

Ogni router di un AS ha un identificativo univoco e di default si prende l'indirizzo IP più alto fra quelli delle sue interfacce. Ai singoli router possono quindi essere associate delle priorità (0-255) che di default sono settate a 0. Queste priorità sono utili quando viene eletto il DR → ogni router della rete esamina i suoi vicini ed elimina tutti quelli non eleggibili dalla sua lista (priorità=0). Successivamente seleziona tra quelli rimasti quello con la priorità maggiore e lo elegge a DR, poi riesegue il procedimento ed elegge il BDR

_SOTTOPROTOCOLLI
OSPF usa il protocollo IP ma ha 3 sotto-protocolli che hanno tutti intestazione comune e informazioni particolari aggiunte in base allo scopo del protocollo.



- Version indica la versione di OSPF
- Type indica il tipo di pacchetto
- Router ID è l'indirizzo IP del mittente
- Area ID identifica l'area di appartenenza (0.0.0.0 è quella di backbone)
- Checksum calcolata sul pacchetto togliendo gli 8 bit dell'authentication
- AuType indica il tipo di autenticazione (0 nessuna, 1 semplice con password in authentication,2 crittografia con dati in authentication)

→ Type 1: Hello protocol. Usato per controllare se il link funziona e per scoprire e mantenere relazioni fra vicini. E' tramite questo protocollo che viene eletto il DR e il DBR. Questi pacchetti sono inviati periodicamente tramite il parametro HelloInterval ed includono la lista dei vicini dai quali è stato recentemente ricevuto un pacchetto HELLO (recentemente = non più vecchio del RouterDeadInterval), per capire se c'è collegamento bidirezionale coi vicini.

→ Type 2: Exchange protocol. Usato una volta stabilite le adiacenze per sincronizzare i link-state database. La procedura avviene in maniera asimmetrica stabilendo master e slave. Il master invia pacchetti Database Description contenenti l'elenco degli LSA del suo database e lo slave risponde con il suo elenco. Durante lo

scambio si confrontano le informazioni e se in un database ci sono LSA meno recenti rispetto a quelli dell’altro vengono richiesti con un pacchetto successivo di tipo 3. Questo avviene perché router adiacenti devono avere topologia coerente e sincronizzata, nonché consistente, per evitare di far girare informazioni sbagliate.

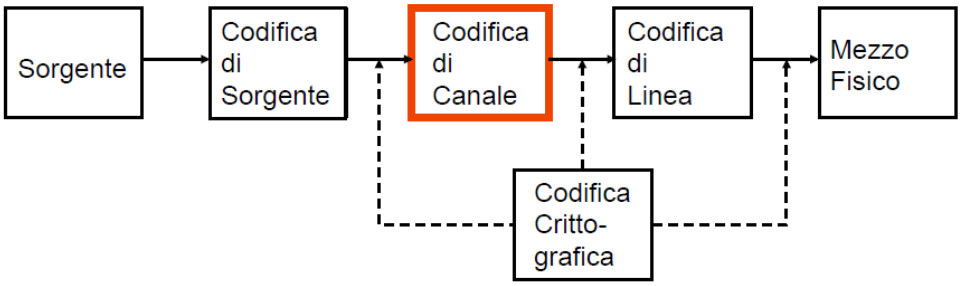
→ Type 3,4,5: Flooding protocol. Controlla i messaggi che diffondono gli LSA (Link State Update, Type 4) e vengono inviati:

- A fronte di un cambiamento nello stato di un collegamento
- A fronte di una link-state request
- Periodicamente ogni 30 minuti. Di solito a meno di richiesta esplicita oppure cambiamenti della rete si vuole che i link state girino molto di rado perché sono molto pesanti → per questo si usa un periodo così lungo di rinvio automatico

Lo stesso update viene mandato finchè tutti gli adiacenti non hanno confermato la sua ricezione tramite il pacchetto Link State Acknowledgement (Type 5).

GESTIONE DEGLI ERRORI

⁂CODIFICA DI CANALE



Ad ogni blocco in cui avviene una codifica in trasmissione corrisponde una decodifica in ricezione. Le operazioni di codifica possono essere applicate in vari modi e in alcuni punti può essere anche inserita una crittografia. I codici si dividono in:

- Codici a blocco → viene applicata la codifica a blocchi di k bit di informazione e per effettuare la rilevazione e la correzione dell'errore vengono inseriti altri bit generati algoritmicamente secondo una funzione combinatoria a partire dai k iniziali. Vengono quindi generati r bit di ridondanza e trasmessi n=k+r bit
- Codici a convoluzione → vengono calcolati r bit di ridondanza ogni k di informazione mediante reti logiche sequenziali e si tiene conto anche di variabili di stato dipendenti dalle operazioni passate (codifica variabile nel tempo)

La codifica di canale si applica a blocchi codificando k bit in parole di n bit aggiungendo quindi r=n-k bit di ridondanza. Essendo disponibili 2^n parole di codice posso trasportare 2^k messaggi (quindi parole valide) andando a creare inevitabilmente 2^n-2^k parole non ammesse.

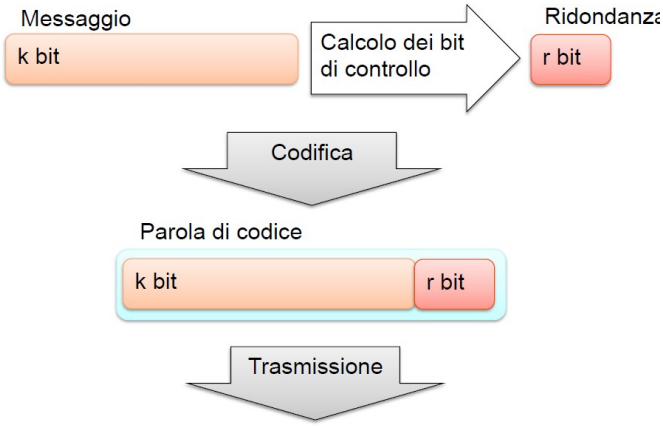
→ Rivelazione di errore: non si può dire quali siano i bit errati ma si dà solo l'informazione che alcuni bit sono stati sbagliati, chiedendo la ritrasmissione dei dati errati

→ Correzione di errore: permette di individuare la parola di codice valida che invece si voleva trasmettere e garantisce trasparenza semantica in tutti i casi in cui l'errore è correggibile

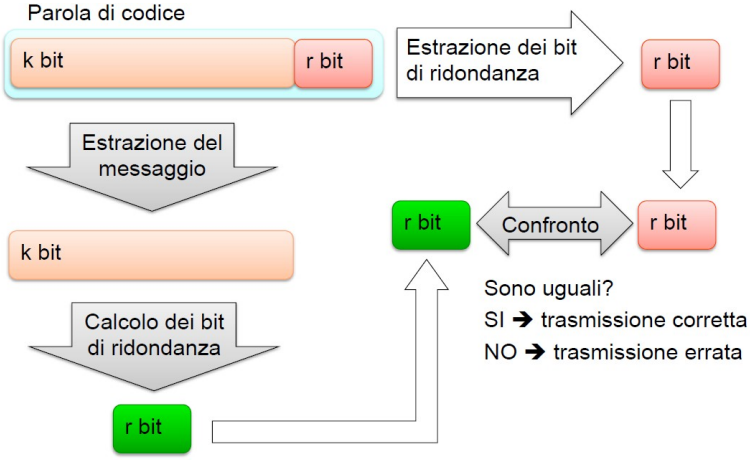
→ Errori a burst: tipicamente nelle reti non ho distribuzione uniforme degli errori ma c'è concentrazione in alcuni intervalli. I bit intermedi del flusso di solito infatti risultano essere quelli meno affidabili soprattutto nei canali di rame che non sono schermati.

La scelta del tipo di codifica dipende dalla situazione → esistono alcuni casi in cui conviene la correzione, per esempio nel caso in cui io abbia molta probabilità di errore. Infatti la codifica a correzione necessita l'aggiunta di molti più bit che può non essere compensata nel caso in cui gli errori siano molto rari. Nel caso quindi in cui il canale sia affidabile conviene sempre la rivelazione. Di solito nello strato fisico si usa la correzione mentre in quello di linea e trasporto la rivelazione

→ Trasmissione: codice a blocco sistematico. Ogni messaggio di k bit viene analizzato e vengono calcolati gli r bit di ridondanza. Questi finiscono nelle PCI al livello protocollare a cui mi trovo e vengono aggiunti alla fine dei primi k bit prima di trasmettere



→ Ricezione: viene usato lo stesso algoritmo del trasmettitore per calcolare quali dovrebbero essere gli r bit di ridondanza per la parola di k bit che ha ricevuto → se gli r bit sono uguali a quelli arrivati si dà per scontato che la parola sia giusta. Il meccanismo funziona solo se assicuro che un cambiamento dei k bit produca sempre un cambiamento degli r bit. Si deve quindi garantire che da sequenze diverse si abbiano bit di ridondanza sempre diversi e che qualora la sequenza k cambi allora si produca sempre un cambiamento degli r bit in modo che un errore produca una parola di codice che non sia valida (r bit non riconoscibili)



_BIT DI PARITA'

Dati k bit di informazione si inserisce un ulteriore bit che serve a rendere il numero totale degli 1 sempre pari (se ho già 1 pari allora p=0, altrimenti p=1). In questo modo se ho errore in trasmissione il ricevitore quando conta gli 1 nel pacchetto trova che non è coerente con il valore di p → ha il problema che rileva solo se ho numero dispari di errori

_CHECKSUM INTERNET
Estensione del bit di parità che si applica su parole di 16 bit indipendentemente dalla lunghezza complessiva del blocco dati. L'IP ha il suo checksum solo sull'header per controllare che il numero sia corretto. Viene effettuata la somma complemento ad 1 dei bit di cui voglio calcolare il checksum. Il risultato viene negato (in modo che la sequenza risultante dia tutti 1 se sommata al risultato vero) e poi viene creata la parola di codice aggiungendo alla fine la sequenza calcolata. Alla ricezione si riesegue lo stesso procedimento solo che all'ultima somma vado a sommare il risultato dei bit inviati con il checksum → se ho trasmesso bene il risultato dei bit iniziali dovrebbe essere la versione negata del checksum (che viene sommato per ultimo essendo alla fine) e quindi il nuovo risultato sarà tutti 1. In caso la somma non produca tutti 1 ho ricezione errata sicuramente.

PROTOCOLLI ARQ

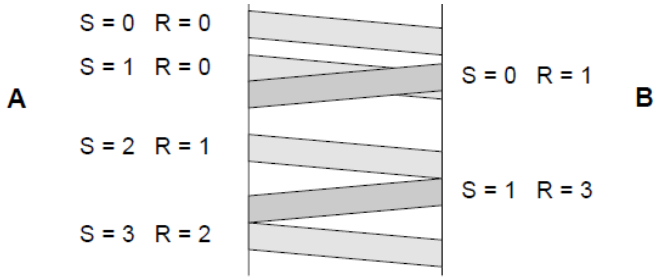
‡ARQ: AUTHOMATIC REPEAT REQUEST
Sono protocolli usati nello strato di linea e di trasporto con l'obbiettivo di rendere affidabile il canale di comunicazione. Il canale può essere:

- Link by link → singolo collegamento seriale nello strato di linea con banda circa costante
- End to end → collegamento a livello di trasporto che non garantisce flusso seriale di bit (i pacchetti possono arrivare fuori ordine) e ha banda variabile nel tempo.

Ogni flusso informativo diviso in PDU porta delle PCI che contengono informazioni relative al protocollo ARQ → ci sono poi anche PDU speciali destinati solo alla segnalazione interna del protocollo (acknowledge e altri).
La garanzia dell'affidabilità si ha controllando i dati prima di consegnarli allo strato superiore:

- Errori di trasmissione → controllati con i codici a rivelazione di errore e la conferma esplicita della ricezione
- Sequenzialità dei dati → numerazione delle unità informative e conferma esplicita di ricezione
- Flusso dei dati → finestra scorrevole e conferma dei dati. E' necessario accordare le velocità perché le tecnologie ai due lati del collegamento potrebbero non riuscire a gestire lo stesso tipo di velocità di flusso (magari per mancanza di memoria o insufficienza di memoria)

_NUMERAZIONE
Le unità informative sono numerate sequenzialmente prima di essere consegnate ai protocolli superiori → TX e RX mantengono due contatori inseriti nelle PCI:

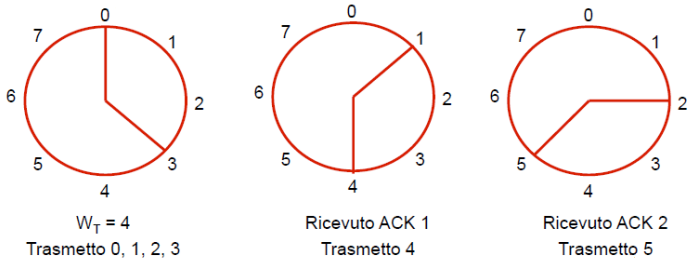


- S → conta le unità inviate. Permette il posizionamento nel flusso informativo
- R → conta le unità ricevute. Permette di confermare la ricezione

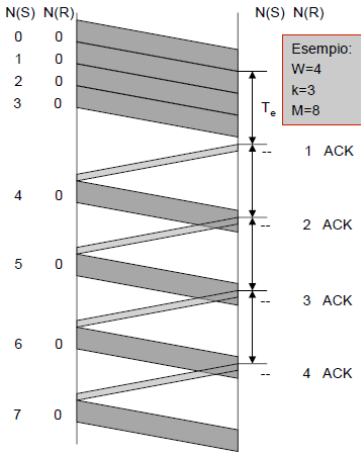
→ Acknowledge: trame speciali che portano il valore di R. Vengono inviate dal ricevitore a conferma della ricezione poiché R aumenta solo se la PDU viene ricevuta correttamente (altrimenti viene

ignorata ed R rimane non modificato). La conferma è quindi implicita → R = n conferma che ho ricevuto tutte le trame fino ad n-1. Per il protocollo non è necessario numerare gli ACK e quindi non c'è conferma della ricezione di questi pacchetti (altrimenti la cosa sarebbe ridondante).

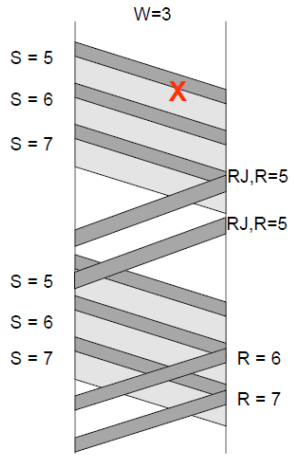
_FINESTRA SCORREVOLE



Con la numerazione a finestra scorrevole limito a W_T il numero massimo di trame che TX può inviare senza riceve alcuna conferma. Effettuo dunque la numerazione modulo $M = 2^n$ dove n è il numero di bit utilizzati per la numerazione. Trametto nuove trame solo al ricevimento delle conferme → la numerazione scorre nel tempo. Per garantire unicità di numerazione lo spazio dipende dal numero di bit che posso dedicare alla numerazione nell'intestazione del pacchetto; posso usare W_T che sia al massimo M-1 dove $M=2^k$ sono i numeri di sequenza → la finestra deve sempre essere più piccola dello spazio di numerazione di almeno 1.

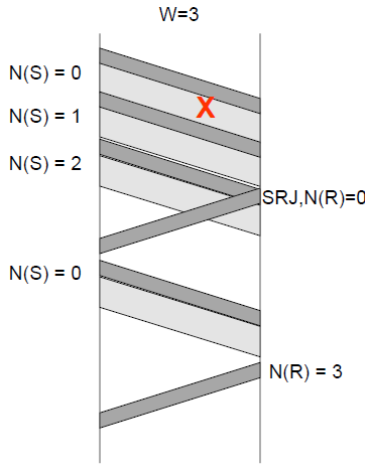


→ Efficacia: si gestisce il controllo del flusso in maniera automatica e il ricevitore deve solo poter riceve, memorizzare ed elaborare un'intera finestra, dopodichè sarà lui a decidere implicitamente quando ricevere nuove trame inviando gli ACK → così accordo le velocità poiché a regime mando una trama ogni T (tempo che RX impiega per elaborarne una). Se RX vede una trama fuori sequenza si ricostruisce in ricezione la corretta sequenza chiedendo il rinvio.



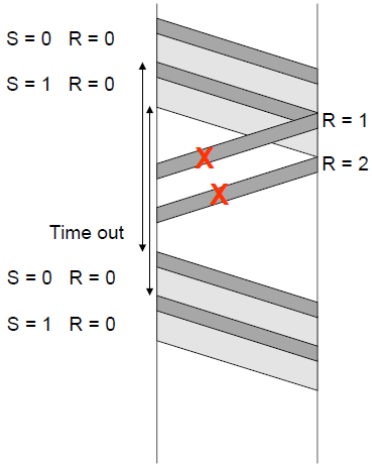
→ Recupero dell'errore: se viene persa una trama il ricevitore scarta le successive e segnala la mancata ricezione mandando un RJ (acknowledge negativo), dopo di che rimane in silenzio senza mandare alcuna segnalazione. TX appena riceve RJ rimanda tutte le trame a partire da quella persa numerata N (metodo go-back-N). Con questa soluzione ho poca complessità nel ricevitore che deve solo controllare la sequenza ma ho una situazione di inefficienza.

_SELECTIVE REPEAT ARQ



Se viene persa la trama N si può anche procedere in maniera più efficiente a ritrasmettere solo quella tenendo le altre ricevute in memoria in RX, per poi ricostruire la sequenza. RX manda la segnalazione con un SRJ e appena TX la riceve rimanda solo la trama N persa → ho maggiore efficienza ma in questo modo devo tenere conto di un accesso non sequenziale in memoria nel ricevitore. Quando vengono inviati gli ACK dopo la ritrasmissione viene inviato solo l'ACK dell'ultima trama che conferma implicitamente anche le precedenti. In questo caso ho il problema della finestra anche in ricezione. Ho W_R che è il massimo numero di PDU che RX può memorizzare prima di consegnarle allo strato superiore → ho allora come vincolo $W_T + W_R \leq M$

_TIME OUT



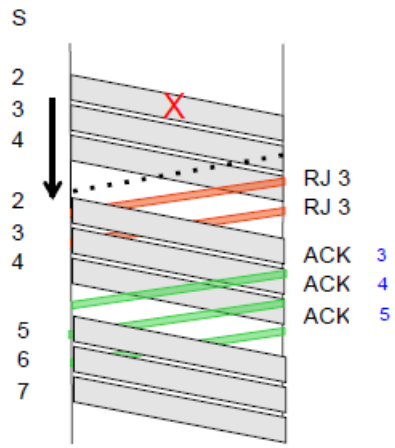
Il protocollo può entrare in stallo se le trame informative o gli ACK sono perduti → si imposta un time out per riprendere il dialogo creando un orologio che parte al termine di ogni trasmissione. Se si raggiunge il valore di time out senza conferma si ritrasmette a prescindere la trama. Nel caso in cui il time out venga mal dimensionato è fondamentale la presenza dei RJ per evitare di perdere tempo e reagire in tempo utile.

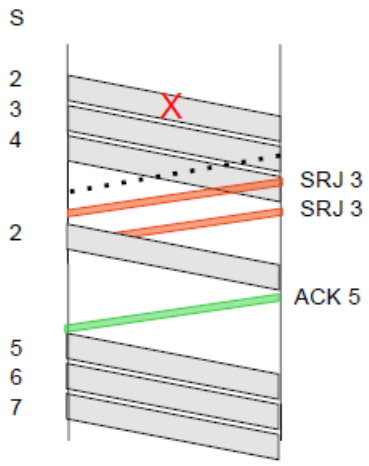
_ROUND TRIP TIME

E' il tempo necessario a fare una andata e un ritorno sul canale quindi quello che passa dalla trasmissione dell'ultimo bit alla ricezione dell'ACK della trama → il time out deve essere relazionato al RTT poiché un time out che scada prima del RTT

porta a ritrasmettere trame che magari non erano state perse mentre un time out che scade troppo più tardi del RTT fa perdere tempo → il time out ideale scade un secondo dopo l'istante in cui ipoteticamente sarebbe dovuto arrivare l'ACK (quindi un secondo dopo il RTT)

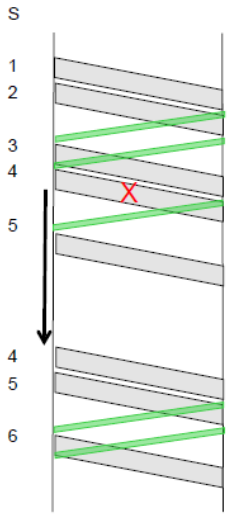
_CONFRONTO GO-BACK-N E SELECTIVE REPEAT

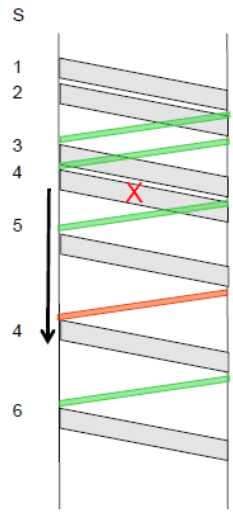




La dimensione della finestra ($W_T = 3$) è prossima al RTT.
In questo caso non c'è particolare differenza nel caso di una singola perdita.

In questo caso avendo time out sovradimensionato ho un vantaggio con selective repeat poiché non devo aspettare ma ritrasmetto appena arriva RJ.

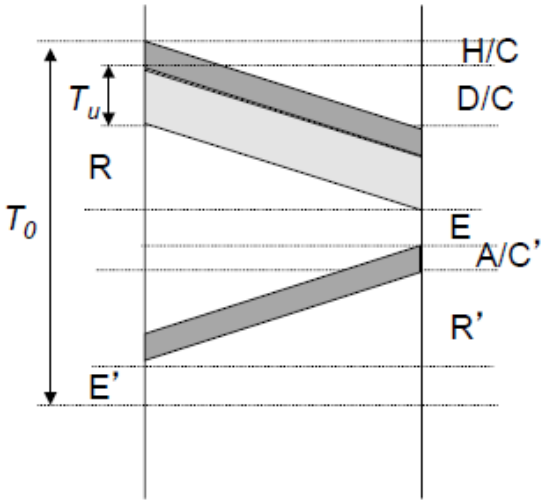




PRESTAZIONI ED EFFICIENZA DEI PROTOCOLLI DI STRATO 2

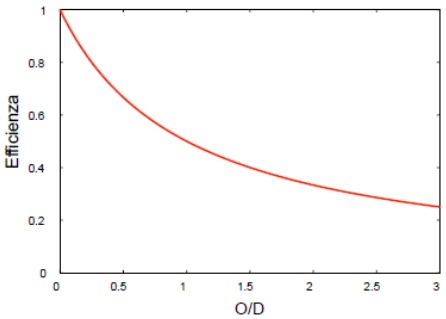
‡PRESTAZIONI ARQ STOP AND WAIT
Equivale ad un protocollo a finestra scorrevole con finestra unitaria. Misuro vari indici:

- D → dimensione del campo dati (in bit)
- H → dimensione dell'header (PCI) in bit
- F=D+H → lunghezza totale del frame
- A → lunghezza dell'ACK
- E,E' → tempi di elaborazione per il controllo del frame in arrivo e per la preparazione del frame in partenza
- E → tempo di propagazione del segnale da un capo all'altro
- I=E+R, I'=E'+R' → tempo totale tra arrivo ed elaborazione (di solito E=E' e R=R')
- C,C' → velocità dei canali di trasmissione
- H/C → misura di tempo su quanto impiego a trasmettere l'header
- D/C → misura di tempo su quanto impiego a trasmettere il dato



Il tempo tra due invii di frame successivi è dato da T_0 = F/C+I+A/C' + I'.

_EFFICIENZA E OVERHEAD
Definisco l'efficienza (considero I=I' e C=C', inoltre A=H poiché l'ACK è quasi solo PCI) come $\eta = (D/C)/T_0 = D/(D+H+A+2IC) = D/(D+2H+2IC) = D/(D+O)$. La quantità O=2H+2IC si definisce overhead e rappresenta la quantità di dati aggiunti dal protocollo → è una grandezza in bit. L'efficienza diminuisce:

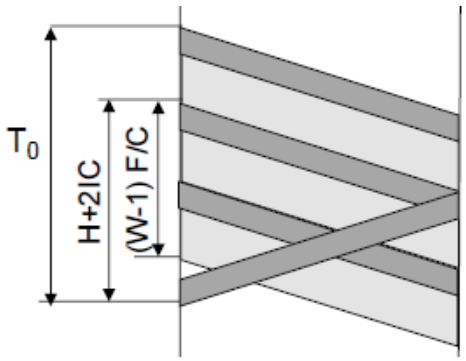


- Al crescere di H → ho molti bit per PCI
- Al crescere di C → la linea è molto veloce
- All'aumentare di I → dovendo coprire molte distanze ho tempi lunghi

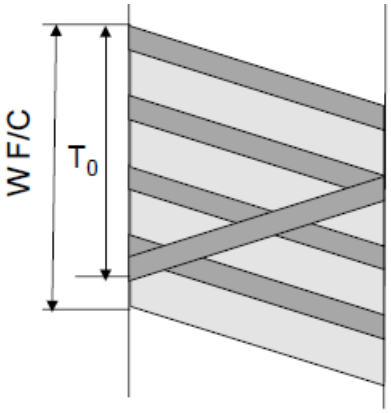
⁑FINESTRA W>1 CON ASSENZA DI ERRORI

Bisogna distinguere due casi:

- (W-1)F ≤ H + 2IC e quindi WF ≤ CT_0 → trasmetto W trame in un tempo pari a T_0 e non vado a riempire con le trame trasmesse in più il tempo vuoto prima dell'ACK → η = WD/(D+2H+2IC)



- WF ≥ CT_0 → non interrompo mai la trasmissione per cui η = D/(D+H)



Il secondo caso è il migliore per l'efficienza poiché non ho tempi morti in trasmissione. Le trame trasmesse in più vanno anche oltre il tempo in cui l'ACK ritorna quindi il tempo di propagazione non conta più e l'unico tempo perso è quello di trasmissione delle PCI → se gestisco la lunghezza di queste massimizzo l'efficienza.

Ho un trade off perché vorrei H il più piccolo possibile ma riducendolo riduco la finestra di TX e rischio di aumentare i tempi persi e perdere il vantaggio dato dal fatto che l'efficienza dipenda solo dalle PCI

RETI LOCALI LAN

⁑LAN
Infrastruttura di telecomunicazioni che consente ad apparati indipendenti di comunicare in un'area limitata attraverso un canale fisico condiviso (oppure il canale radio che è intrinsecamente condiviso) ad elevata bit rate con bassi tassi di errore. Le reti essendo di dimensioni limitate rendono convenienti soluzioni particolari per gli strati 1 e 2 ed occorre scegliere mezzo trasmissivo, topologia e protocollo di accesso.

→ Mezzo trasmissivo: in generale nelle reti moderne le fibre ottiche stanno progressivamente sostituendo il rame mentre nelle LAN, poiché ci sono dimensioni limitate, il costo del mezzo incide di meno rispetto al costo dell'attacco per le stazioni (prese e spine per la fibra sono molto costose e complicate). Il mezzo radio, per motivi di affidabilità e di costi, non è stato usato fino alla fine degli anni 90 ma acquista importanza crescente sempre.

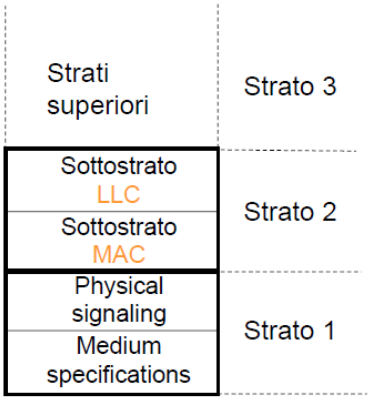
→ Topologia punto-multipunto: mezzo di trasmissione condiviso con caratteristiche broadcast native e con problemi di collisione intrinseca dovuto al fatto che il mezzo condiviso crea la possibilità che più utenti inviino informazioni contemporaneamente.

_MAC ADDRESS
Sono indirizzi composti da 48bit cablati nella scheda di rete e univoci a livello mondiale. I primi 3 byte individuano il costruttore e i secondi 3 numerano progressivamente le schede. E' possibile specificare:
- Singolo destinatario (unicast)
- Gruppo di destinatari (multicast) → il primo bit ad 1
- Invio a tutti (broadcast) → ff-ff-ff-ff-ff-ff

_CONTROLLO E PROTOCOLLO DI ACCESSO
Nelle LAN si preferisce l'accesso multiplo a divisione di tempo.
- Controllo centralizzato → è presente un coordinatore o stazione primaria che coordina tutti gli altri
- Controllo distribuito → ogni terminale decide per se eseguendo il protocollo MAC
    - Assegnazione statica → per ogni connessione si assegna un canale a priori
    - Assegnazione dinamica → la stazione impegna il mezzo solo quando ne ha bisogno

⁑PROGETTO IEEE 802
Serviva a cercare di definire uno standard per le LAN. Lo strato 2 viene diviso in due sottostrati:

- LLC → Logical Link Control. Indipendente dal mezzo fisico, dalla topologia e dal protocollo di accesso
- MAC → Medium Access Control

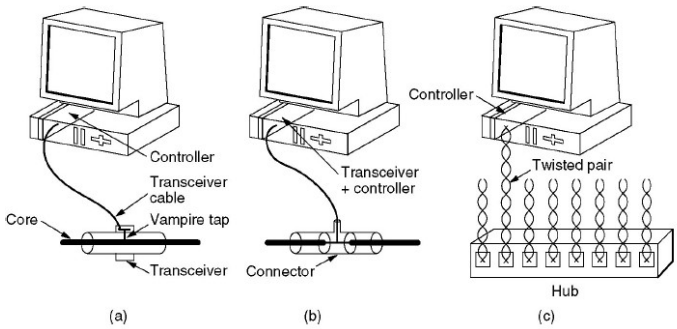


Anche lo strato fisico 1 viene diviso in due parti.

⁑SOLUZIONI PER LO STRATO FISICO ETHERNET
La proposta iniziale era con un cavo coassiale a 50Ω (poi valutato troppo rigido e quindi inadatto al cablaggio di un edificio) che trasmetteva in 10base5:

- 10 → velocità 10Mbit/s con codifica Manchester
- Base → trasmissione in banda base
- 5 → segmenti fino a 500 metri

(a) 10Base5, (b) 10Base2, (c) 10Base-T.



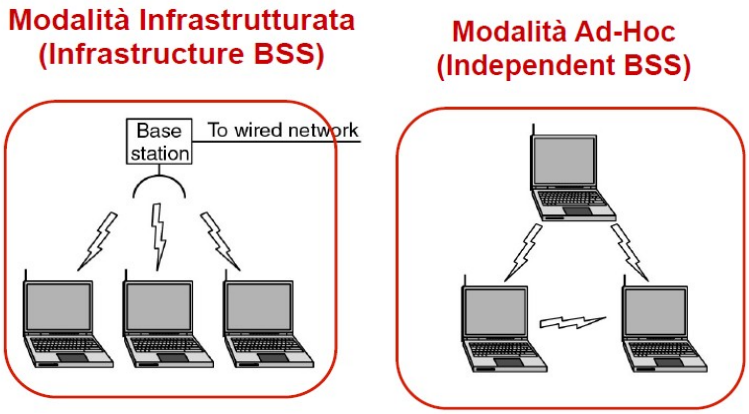
Si passa poi al 10base2 con cavo coassiale sottile e segmenti fino a 180m → il transceiver è integrato nella scheda a bordo del computer. Il collegamento avviene con connettori BNC e computer in seri sul segmento con connettori a T.

La versione successiva è il 10baseT che usa coppie simmetriche intrecciate (non schermate UTP) e ogni stazione ha una UTP a disposizione collegata alla porta di un hub che funziona da multiport repeater.

→ Cablaggio: organizzato in maniera gerachica

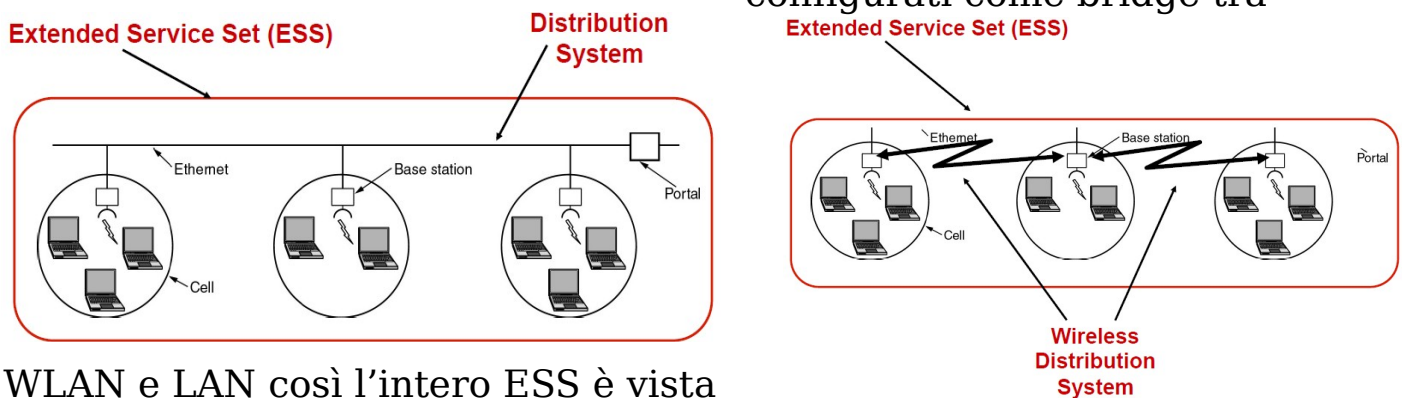
⁑WIRELESS LAN (Wi-Fi)
Nei primi standard 802 viene ignorato il mezzo radio che viene inserito solo successivamente. Nello standard 802.11 si definisce la trasmissione fisica delle trame usando la banda ISM a 2.4GHz.
Nella versione 802.11a si implementa il Wi-Fi a banda larga → usa la banda ISM a 5GHz in 12 canali da 20MHz ciascuno ottenuti con diverse codifiche e modulazioni scelte in base alla distanza da coprire.
Come ultimo viene definito 802.11b che usa la banda ISM a 2.4GHz e 14 canali da 5MHz ciascuno

_ARCHITETTURA DI RETE 802.11



- Modalità infrastrutturata (infrastructure BSS) → le stazioni comunicano attraverso l'AP anche se non si vedono direttamente.
- Modalità Ad-Hoc (indipendent BSS) → le stazioni comunicano in modalità peer-to-peer e solo se si vedono direttamente.

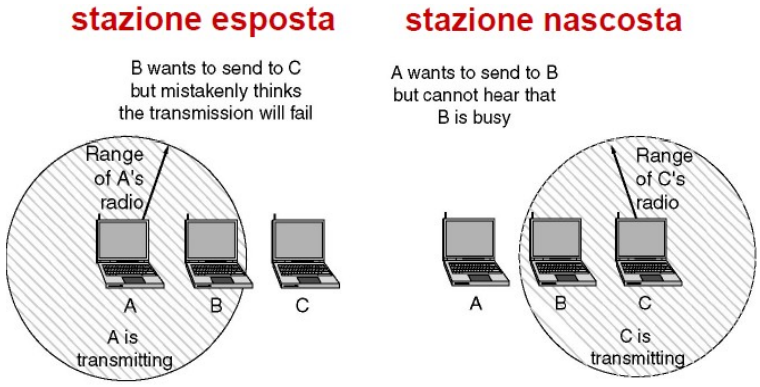
Occorre gestire l'associazione tra le stazioni e gli AP e permette la mobilità delle stazioni trasparente agli stati superiori. Gli AP sono configurati come bridge tra



WLAN e LAN così l'intero ESS è vista come unica LAN (quindi unico dominio di broadcast).

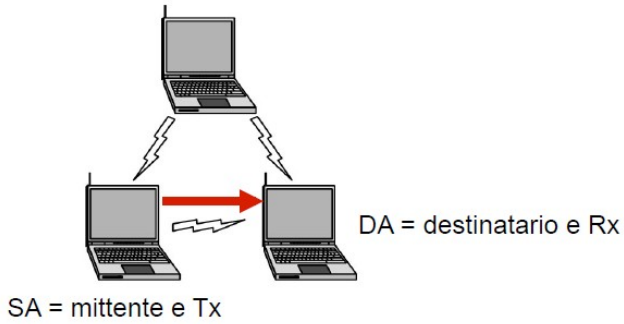
_PROBLEMI DI ACCESSO MULTIPLO

Al contrario delle LAN cablate nelle WLAN ho problemi specifici:



- Stazione nascosta → A vuole mandare a B ma non riesce a vedere che B è occupato
- Stazione esposta → B vuole mandare a C ma pensa che la trasmissione non funzionerà

_INDIRIZZAMENTO 802.11



Nei protocolli Wi-Fi di tipo IBSS non ho indirizzo solo indirizzo MAC sorgente e destinazione ma ne ho 4:

- Address 1 → DA. Indirizzo del destinatario del pacchetto
- Address 2 → SA, id della rete. Serve a capire, qualora ci siano altre reti ad-hoc conviventi, quali pacchetti vanno letti
- Address 3 → BSSID. Generato casualmente da una delle stazioni dell'IBSS
- Address 4 → N/A

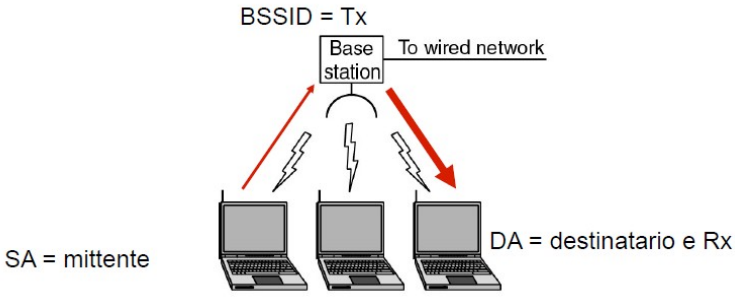
Nei protocolli Wi-Fi di tipo BSS/ESS ho invece sempre due canali attivi:

- Uplink dal computer verso la stazione base
- Downlink dalla stazione base al computer

Entrambi i canali sono broadcast ma l'uplink al contrario del downlink è condiviso. Il protocollo coordina il numero dei nodi periferici per evitare collisioni → quando ho troppi nodi attaccati la collisione crea problemi molto gravi. Anche in questo caso ho 4 indirizzi:

→ Uplink:

- Address 1 → BSSID (MAC address dell'AP) cioè indirizzo reale destinazione del pacchetto (stazione base nel caso dell'uplink)

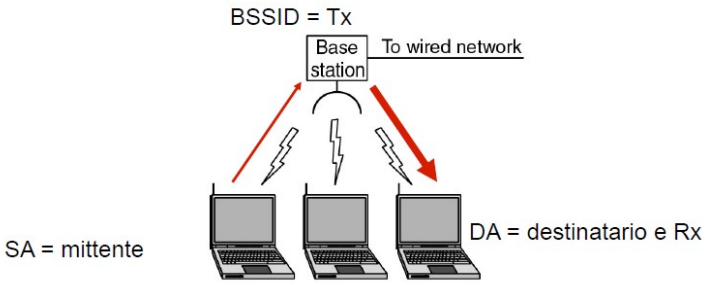


- Address 2 → SA. Indirizzo chi manda il pacchetto
- Address 3 → DA. Indirizzo destinazione finale del pacchetto
- Address 4 → N/A

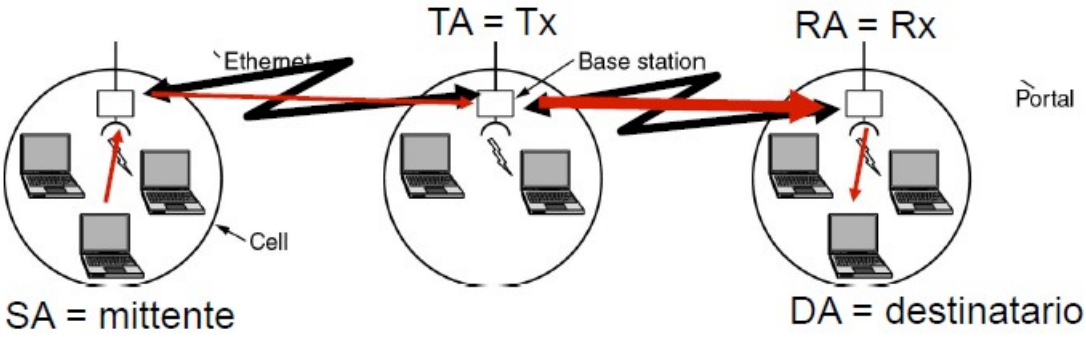
Quando il pacchetto arriva alla stazione AP si cambiano i posti degli address.

→ Downlink:

- Address 1 → DA. Indirizzo destinazione finale del pacchetto
- Address 2 → BSSID (MAC address dell'AP)
- Address 3 → SA. Indirizzo chi manda il pacchetto
- Address 4 → N/A



→ ESS con Wireless Distribution System

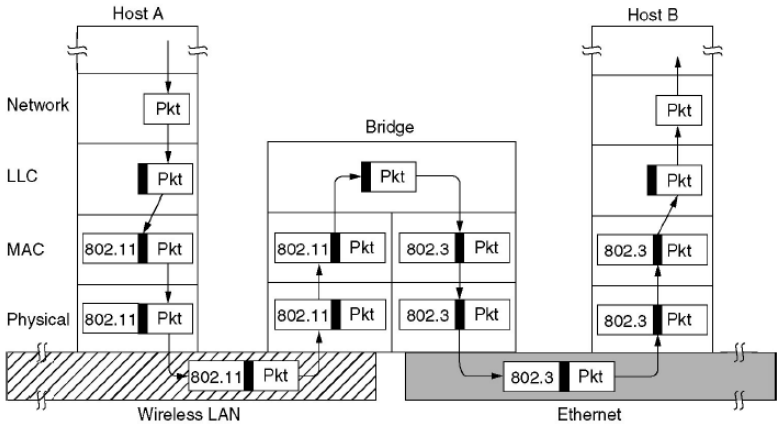


‡INTERCONNESSIONE DI LAN E VLAN

→ Repeater: apparato attivo che collega 2 o più mezzi di trasmissione e opera a livello dello strato 1 OSI. Permette l'estensione del mezzo di trasmissione e amplifica il segnale rigenerando i bit entranti e sincronizzandoli.

→ Bridge: opera a livello dello strato 2 OSI e può interconnettere LAN di tipo diverso. Nel caso di reti Ethernet separa i domini di collisione.

Operation of a LAN bridge from 802.11 to 802.3.



Da ogni lato il bridge parte di livello 1 e 2 di ogni zona e deve essere in grado di arrivare al livello 3, anche se non legge il pacchetto. Inoltre deve essere in grado di inserire il pacchetto dalla busta del primo lato ad una del secondo. Il base all'IP il bridge può decidere di fare anche da gateway verso la rete geografica.

→ Switch: è un bridge ad alta densità di porte. Ad ogni porta è connessa una sola stazione e tipicamente le porte sono tutte di uguale tecnologia o comunque di sole 2/3 tecnologie diverse. E' in grado di trasferire contemporaneamente trame da più porte di ingresso a più di uscita.