



Università degli Studi di Bologna
Scuola di Ingegneria

Corso di
Reti di Calcolatori T

Reti e Routing

Antonio Corradi

Anno accademico 2023/2024

SUITE TCP/IP E INTERNET

Gli standard possono nascere da comitati o anche dal basso da esigenze di uso e con obiettivo di realizzazione immediata

Internet nasce dalla idea di potere interconnettere tutte le reti in una unica globalità (il migliore dei mondi possibili)

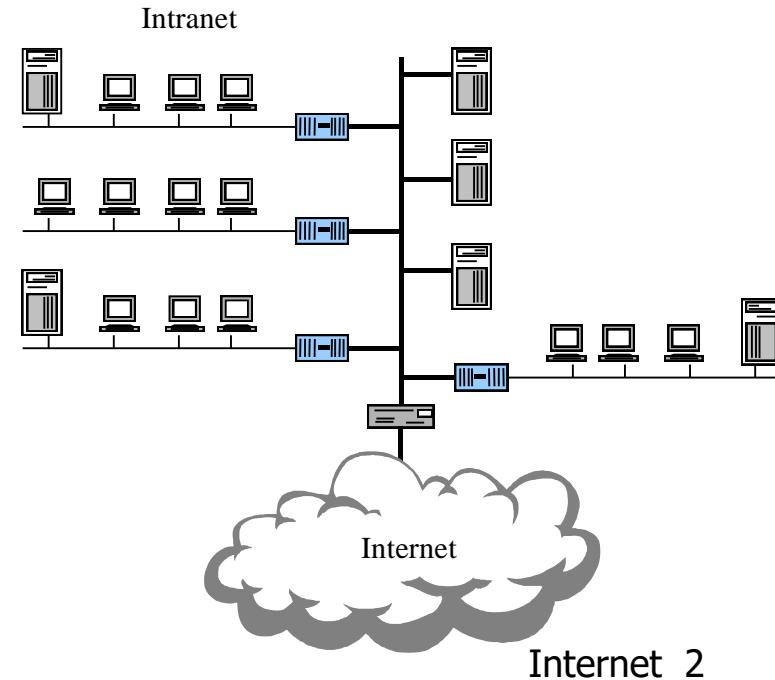
SISTEMA GLOBALE a nessun costo e per tutti

protocolli liberi, aperti, a nessun costo

Intranet come insieme di reti aziendali che adottano protocolli standard IETF

SISTEMA di RETI per scopi aziendali con problemi di sicurezza, di accesso, di controllo, di accounting, ...

Protocolli a basso costo per la comunicazione con il sistema globale



I DUE LIVELLI TCP/IP

TCP - Transmission Control Protocol

- flusso di byte bidirezionale a canale virtuale best effort, dati non duplicati, affidabili, con controllo di flusso

livello TX

UDP User Datagram Protocol

livello TX

- scambio di messaggi end-2-end

IP Internet Protocol (Routing)

livello di RETE

- scambio di datagrammi senza garanzia di consegna tra vicini

ICMP Internet Control Message Protocol

gestione RETE

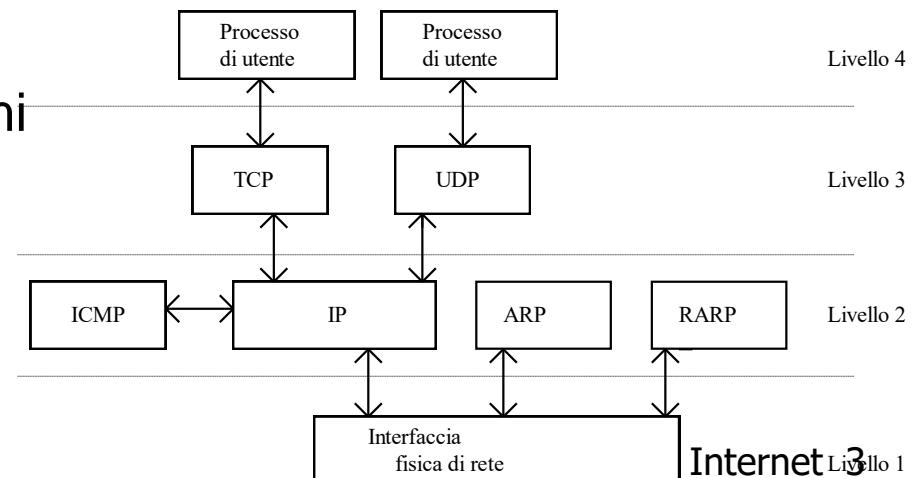
- scambio messaggi di controllo

ARP e RARP Protocol

- Interazione con livello fisico e nomi

STACK di protocolli

a basso overhead e best effort



AZIONI DI GRUPPO IN TCP/IP

Broadcast e Multicast come azioni di gruppo

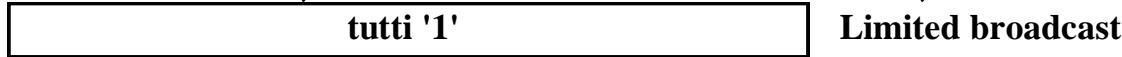
- **NON sono consentiti broadcast** a livello globale
- vista la dimensione del sistema \Rightarrow per evitare costo inaccettabile
- **Broadcast permessi solo nell'ambito di una rete locale**

BROADCAST limitato

- **per tutti gli host della rete locale indipendentemente dall'indirizzo IP** indirizzo in cui tutti i 32 bit sono a 1 (limited broadcast address)
solo intranet e non viene fatto passare da una rete ad un'altra

BROADCAST diretto

- **tutti gli host in una rete specifica**
tutti i bit di hostid a uno (broadcast direttivo o directed broadcast)
trasmesso in Internet, arrivato alla destinazione, broadcast



AZIONI DI MULTICAST

Broadcast e Multicast come azioni di gruppo

- Broadcast consentiti solo tenendo conto del costo intrinseco

Oltre ai normali indirizzamenti di classe A, B, C

Indirizzamenti multicast di Classe D



tutti gli host che si sono registrati possono ricevere messaggi e possono mandare messaggi al gruppo di multicast (vedi socket multicast)

- L'esistenza della classe implica anche il supporto per trovare il gruppo e mantenerlo
 - In Internet i protocolli hanno senso se si possono implementare
- Necessità di **infrastruttura di propagazione e di servizio** (quanto costa?)

PROBLEMA FONDAMENTALE

- I protocolli sono stati definiti molte tempo fa e solo in modo sperimentale, ma realmente implementati ampliamente in tempi recenti

RETI

Le reti permettono ed attuano le **interconnessione tra i diversi sistemi di esecuzione**

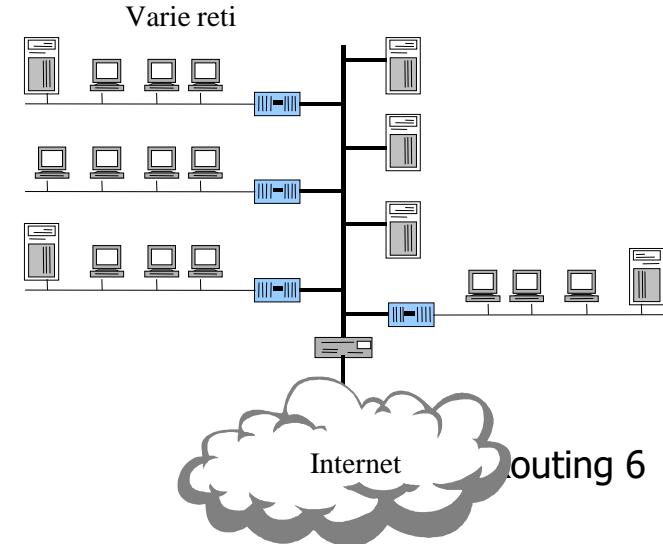
Un **sistema globale** è costituito da **una o più reti** e permette la comunicazione tra **tutti i partecipanti attraverso località interconnesse da router**

Le reti sono il mezzo di interconnessione punto a punto o via comunicazioni di gruppo che coordinano molte entità

Ragionare a livelli nasconde i dettagli e permette la astrazione

La rete è misurata da indicatori come **costo, velocità ed affidabilità** di invio dei messaggi, nell'intero **percorso** da dove provengono a dove devono essere consegnati

Sempre di più quando parliamo di reti parliamo di protocolli **Internet compatibili**



PARAMETRI E PERFORMANCE DELLE RETI

Le reti si possono misurare in molti modi e secondo molte metriche diverse e con scopi diversi

reti intese come interconnessioni punto a punto e via interconnessioni di gruppo (e.g., multicast e broadcast)

Parametri:

- **tempo di latenza** (tempo di ritardo sulla comunicazione)
- **banda** (quantità di dati trasmessi per unità di tempo)
- **connettività** (tipo di interconnessione e topologia)
- **costo apparati**
- **reliability** (affidabilità)
- **funzionalità** (ad esempio, attraverso operazioni sui messaggi come combinazione e frammentazione dei messaggi e ricomposizione)

TOPOLOGIA PERFORMANCE E COSTO

Le reti di diverse topologie hanno costi performance diverse

Quindi la scelta della topologia è particolarmente importante e deve essere pianificata attentamente

In genere, tutti devono potere comunicare con tutti

- **Interconnessione completa** (tutti possono comunicare con tutti)

Garantisce performance ottime e trasferimenti molto veloci

Ha un **costo molto elevato** e non è scalabile

Interconnessioni statiche (precisa topologia)

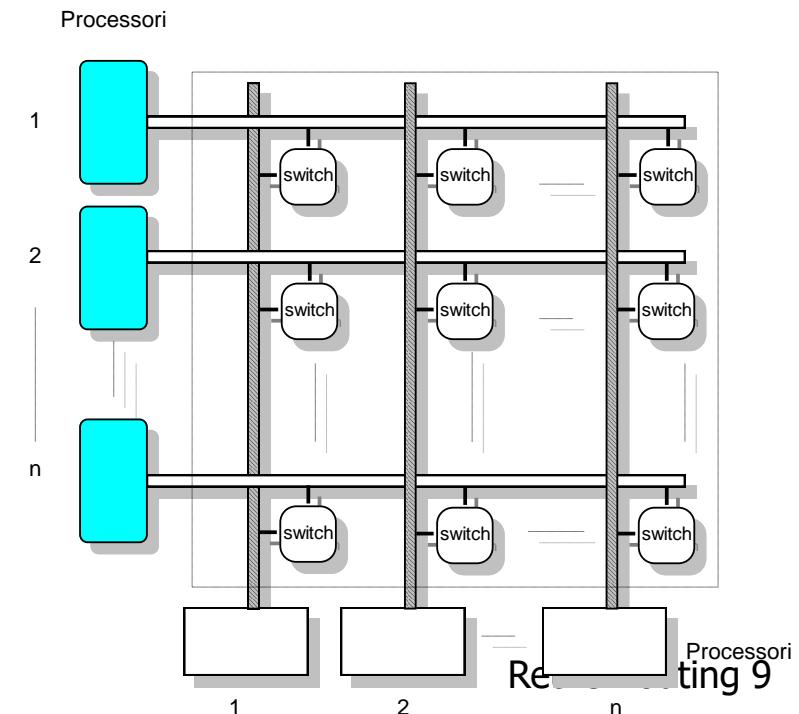
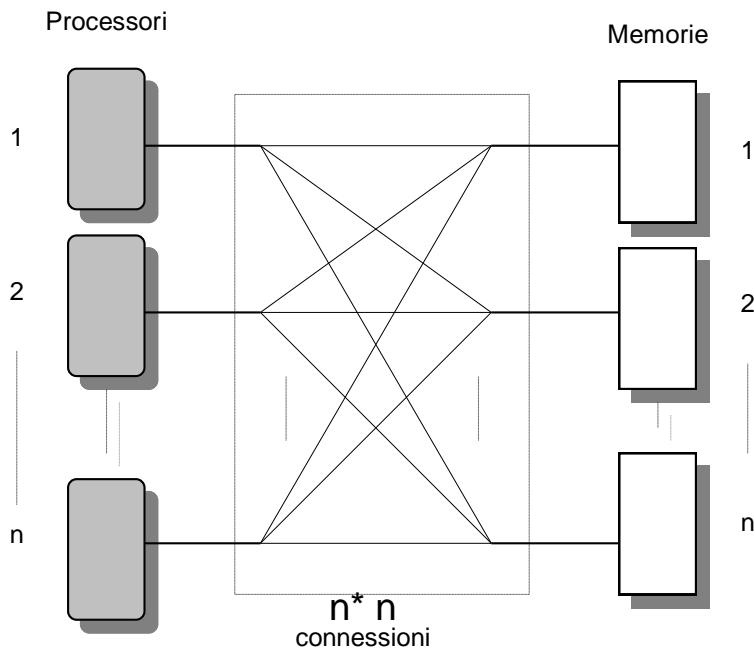
A seconda della topologia decisa ha performance e costi diversi

Interconnessioni dinamiche (topologia variabile)

A seconda della topologia, risparmia sulle interconnessioni predefinite, ma ha costi di switching e limiti sui temi che devono essere tenuti in conto

INTERCONNESSIONE COMPLETA

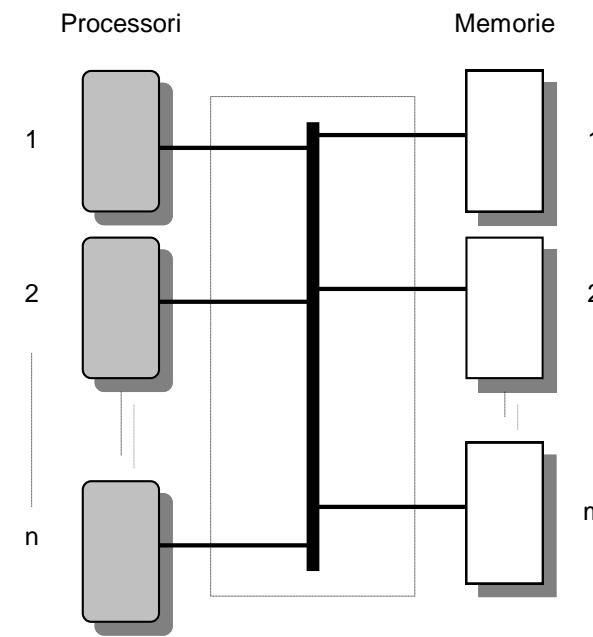
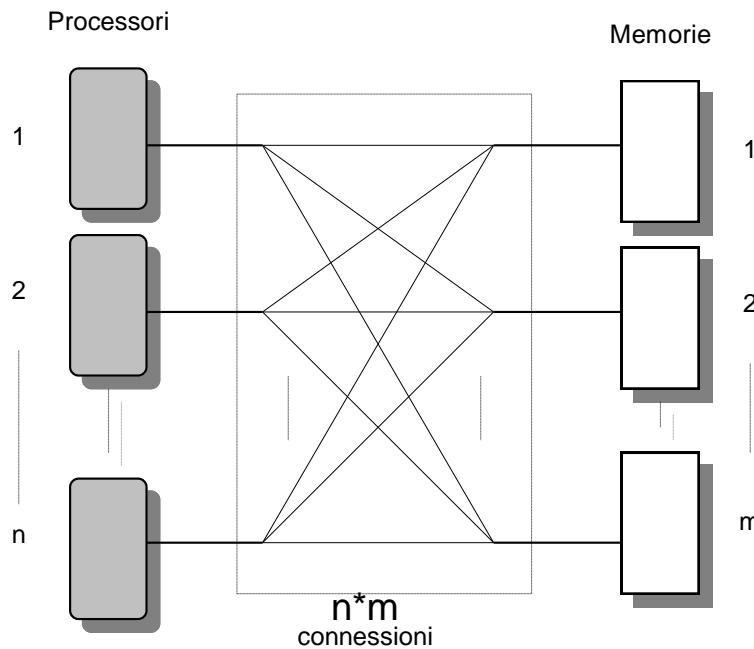
Possiamo anche pensare di interconnettere tutti con tutti
Interconnessione completa, ossia tutti con tutti
se n elementi, n^2 connessioni
con il costo conseguente di una serie di switch hardware



INTERCONNESSIONE VIA BUS (DINAMICA)

Per interconnettere **tutti con tutti**, possiamo usare **un unico mezzo di interconnessione**, cioè **un bus dinamico**

Bus unico: se qualcuno lo occupa, gli altri devono aspettare
(il mezzo è unico e va usato bene in accesso)



SWITCHING E CONNESSIONI DINAMICHE

A volte, interconnettere di diverse entità richiede di **ottimizzare l'uso delle risorse** e metterle a **disposizione in modo dinamico** di chi manifesta la **necessità di uso (vedi bus)**

Lo **switching** permette di dedicare le risorse a più **richieste in tempi diversi** e non mantenerle allocate ad un unico obiettivo di connessione

Nel caso del bus, un unico **collegamento e banda** viene ad essere usato per veicolare comunicazioni diverse (evitando interferenza) usando la dimensione tempo per sfasare le interferenze

Lo **switching** permette di prevedere un **impegno anche condiviso delle risorse** per consentire di passare i dati in caso di nodi non in visibilità in modi diversi anche ottimizzando

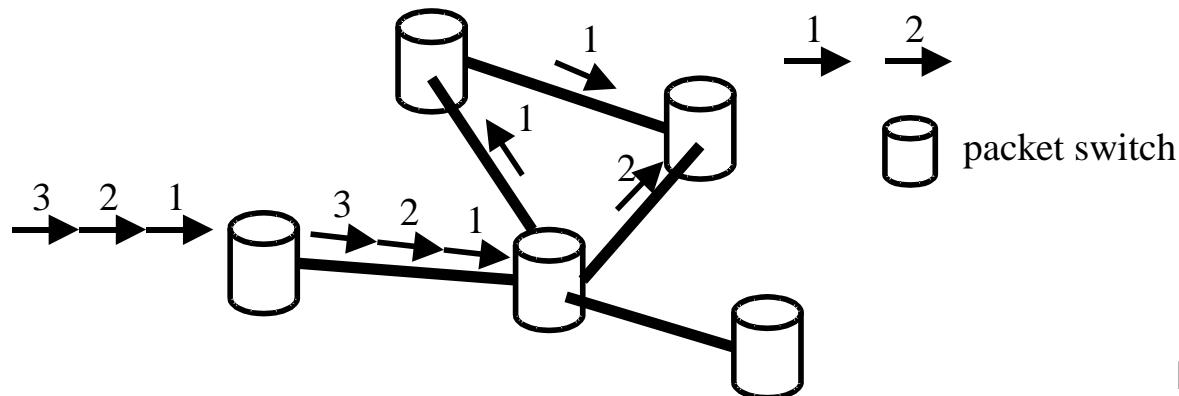
Nel caso di messaggi diversi, possono essere frammentati in pezzi corti che possono viaggiare insieme su tratte di reti disponibili (**caso di IP Internet e dei datagrammi**)

SWITCHING A DATAGRAMMI (DINAMICO)

Non si prevede alcun **circuito o canale**, ma solo **datagrammi**, come unità di trasmissione, che possono essere mandate tra nodi vicini ed inoltrati via **routing**, mirando ad ottenere il migliore uso del sistema di comunicazione con una politica reattiva

Comunicazione a datagrammi (tecnica ottimista)

Con i **datagrammi**, non si garantisce nessuna connessione end-to-end, e quindi nessun controllo flusso, nessuna garanzia (no ordine, no QoS) ogni entità da inviare (**datagramma**) porta indirizzo del ricevente e viene smistato in modo indipendente; si possono introdurre **molti ritardi e jitter**

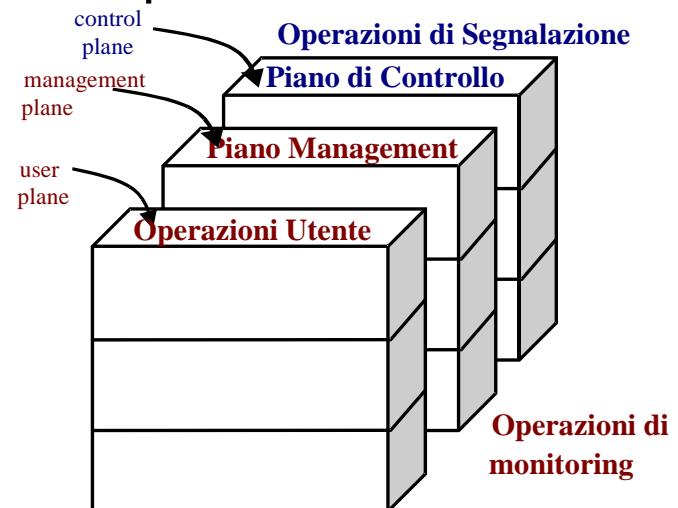


SEGNALAZIONE E CONTROLLO COMUNICAZIONE

Se dobbiamo **gestire connessioni in modo dinamico** (come nella rete telefonica), abbiamo necessità di operazioni per **preparare la comunicazione** (stabilire la connessione), **mantenerla** e **garantirla**, e **chiuderla** al termine

Questo introduce nuovi piani di specifica oltre a quello utente

- **Piano User**
per i protocolli utente
- **Piano di Management**
stabilire e gestire il canale
- **Piano di Controllo (segnalazione)**
gestione **qualità della connessione**



Possibilità di Segnalazione (o Controllo) e gestione tabelle

- **in-band**: usando gli stessi cammini e risorse per i dati
- **out-of-band**: cammini separati per il controllo e il signaling

RETI? DI CHE TIPO?

Spesso i sistemi di interesse sono costituiti da reti molto differenziate
... (ossia reti destinate a interconnettere entità geograficamente
molto eterogenee)

- **Wide Area Network - WAN**

per reti geografiche anche a copertura molto ampia (globale)

- **Metropolitan Area Network - MAN**

area di copertura di una città

- **Local Area Network - LAN**

reti di dimensione limitata e con forte limitazione sui partecipanti

- **Personal Area Network - PAN**

reti di dimensione veramente limitata e ad uso di utenti singoli
con tecnologie ad hoc e propagazione ancora più limitata (reti
wireless)

RETI LOCALI

Le **reti locali**, o **LAN**, sono state usate molto come **banco di prova** e di esperienza dei **protocolli**, oltre che ampia palestra di uso

In una rete locale ogni messaggio è mandato broadcast

LAN caratterizzate da

- **alta velocità ed ampia banda di trasmissione**
- **facilità di broadcast**
- **bassa probabilità di errori**

A livello di analisi possiamo partire ad analizzare le LAN,

- **topologie**
- **mezzo trasmissivo** (wired e wireless)
- **controllo di accesso**

Anche se ovviamente molto dell'interesse industriale è non solo sulle LAN (e PAN), ma sulla loro interconnessione in modo efficace

INTERCONNESSIONE TRA RETI

Possiamo avere molti apparati per interconnettere (e separare!) entità a diversi livelli OSI

Ripetitori (o hub): rigeneratori di segnale a livello fisico, superando e rimediando ad un'eventuale livello di attenuazione (livello fisico OSI)
Un ripetitore non effettua alcuna separazione

Bridge (o switch multiporta): apparati che collegano reti diverse, con capacità di separazione e maggiore intelligenza (**livello data link**)

Router (o gateway): apparati e sistemi per il passaggio da una rete ad un'altra con obiettivo di supportare la comunicazione dei messaggi ed il routing (**livello network**)

Protocol converter: sistemi che collegano reti diverse a più alto livello con protocolli diversi di interconnessione (operazioni dal livello di **trasporto in su**)

BRIDGE (LIVELLO OSI 2)

Un **bridge** collega e separa due (o più) reti a livello di **data link**, controllando il passaggio e la separazione delle due reti, ad esempio passando e bufferizzando i frame dall'una all'altra, solo se necessario, e trattando anche situazioni di errore (solo livello 2)

Un bridge mantiene il traffico di una parte di e non fa entrare traffico non necessario ottenendo una separazione a livello 2

Vantaggi

- **separazione effettiva delle reti**
- **bufferizzazione dei frame** (trattando il caso di overflow)
- capacità di gestire **controlli di accesso diversi**

Svantaggi

- ritardo di bufferizzazione
- bufferizzazione limitata e non infinita
- trasformazione dei frame (con controllo)

TIPI DI BRIDGE (SPEZZONI A LIVELLO 2)

I **bridge** possono lavorare in molti modi diversi
Bridge **multiporta**, che collegano più segmenti di rete diversi
Bridge **trasparenti**, che operano in modo invisibile agli utenti
che devono **collegare solo le entità** interessate, ossia **bloccare i pacchetti** che devono rimanere **locali** e fare **passare solo** ciò che **deve transitare**

Si parla di **routing isolato**, anche se siamo a livello di frame, e
dobbiamo avere previsto un **database di forwarding**

1. informazioni **della tabella** come database in PROM
2. bridge con **capacità di apprendimento** osservando il traffico
Il bridge impara la allocatione vedendo il traffico della rete e dai vicini (il tutto è ripetuto quando è necessario coordinarsi)

BRIDGE VARI (AZIONI A LIVELLO 2)

Un bridge può fare **learning delle diverse configurazioni**

FASE di LEARNING iniziale - Alla inizializzazione (ossia all'inserimento su una rete), il bridge comincia a controllare le esigenze di comunicazioni nel sistema corrente e si adegua, prima facendo inizialmente passare tutto, poi cominciando a filtrare secondo la topologia appresa

Possibilità di conflitti per situazioni diverse e con molti bridge

In caso di più bridge, con interconnessioni varie, si tenta il coordinamento dei diversi bridge

Algoritmo spanning tree - i bridge si scambiano messaggi per trovare i costi più bassi di collegamento e costruire un albero unico che percorre tutta la topologia di collegamenti

Si sceglie un **bridge radice** tra tutti e ognuno degli altri trova il cammino minimo per collegarsi a quello (passi e velocità)

ROUTE E ROUTING (LIVELLO OSI 3)

Il problema del **cammino dal sorgente al destinatario** in caso di **comunicazione** viene affrontato e risolto dal **routing**, che coordina e permette di attraversare un insieme di nomi intermedi (**dinamico**)
Necessità di mapping efficiente

ROUTING come la **identificazione** e l'**uso di algoritmi di routing**

Proprietà del routing

- **correttezza**
- **semplicità**
- **robustezza** (tolleranza ai guasti e variazioni)
- **stabilità** della soluzione
- **ottimalità**
- **fairness** (giustizia)

PROBLEMI NEL ROUTING

Alcune situazioni sono critiche per il supporto al routing:
Congestione, Deadlock, Livelock

Congestione

Le entità diverse devono predisporre risorse per evitare i problemi
C'è necessità di controllare gli asincronismi che possono impegnare le risorse e mantenerle

È opportuno un **buon controllo dei buffer** per evitare il problema

- si inviano **indicazioni al mittente** (messaggi di choke)
- si scartano tutti i **messaggi successivi** (tecnica reattiva)
- si prevede un **numero massimo fisso di messaggi circolanti** (tecnica proattiva)

ANCORA SITUAZIONI DA EVITARE

Deadlock impegno totale dei buffer con blocco critico

Avoidance si numerano i buffer e si **acquisiscono** in ordine

Prevention si mantengono buffer per fare **scambi in caso di saturazione**

Recovery si tratta il problema quando rilevato

Livelock con **messaggi che continuano a permanere** nel sistema **senza** giungere a **destinazione** (persi in cicli o altro)

Soluzioni a priori

si mantiene il **cammino percorso** e si evitano i loop

Soluzioni a posteriori

si elimina il messaggio oltre un certo **numero di passi** (time-to-live del messaggio)

CLASSIFICAZIONE ROUTING

Classificazione delle strategie di ROUTING

- **GLOBALE** / **LOCALE (ISOLATO)**
- **STATICO** / **DINAMICO**
- **NON ADATTATIVO** / **ADATTATIVO**
anche **non deterministico / deterministico**

In caso di **routing locale**, si attuano decisioni a basso costo ma al limitata visibilità; il **globale** richiede tabelle per mostrare lo stato di tutte le interconnessioni possibili

In caso di **routing statico** si usano cammini fissi e globali di propagazione, che invece possono variare nel **routing dinamico**

In caso di **routing non adattativo** i cammini sono fissi e non variano, in caso **adattativo** si sfruttano anche le **risorse libere in modo dinamico**

FAMIGLIE DI ALGORITMI DI ROUTING

In generale per fare istradamento, funzione fondamentale del livello OSI 3, si possono scegliere molte strategie ed implementazioni diverse

Facendo riferimento ad Internet, come caso globale, e non solo... abbiamo la scelta tra molte famiglie:

Algoritmi isolati, senza tabelle e coordinamento tra i router Random, Patata bollente, ...

Algoritmi globali, con tabelle che ogni router mantiene e che prevedono forme di coordinamento tra i router

Distance Vector, Link State

Queste due famiglie, in ordine storico, prevedono **tabelle su ogni router** e qualche forma di **propagazione di informazioni**, in caso di variazione

ALGORITMI ISOLATI DISTRIBUITI

Strategie **dinamiche** indipendenti dalla topologia di interconessione che si basano su informazioni solo **locali (isolati)** o solo di **vicinato (distribuiti e locali)** possono essere molto efficaci

Mancano del tutto i costi delle fasi di coordinamento e si limita fortemente overhead in caso di variazioni, e di cammini diversi

Si possono introdurre problemi per la perdita della visibilità della topologia e del coordinamento: sono possibili cicli o livelock

Algoritmo Random, scelta **casuale** dell'uscita

Algoritmo Patata Bollente

Un messaggio viene smistato (se non a destinazione) attraverso la **coda di uscita più scarica** del nodo

Non si può predire il numero di passi per arrivare a destinazione: il numero di hop ed i cammini dipendono dal traffico

ALGORITMI ISOLATI

Nel caso si conosca la topologia, si possono sovrapporre anche informazioni di direzione per raggiungere il destinatario (ad es. su una **mesh**)

Si noti che un algoritmo isolato adattativo trova il ricevente anche se questo si muove!

Algoritmo FLOODING

Un messaggio viene smistato (se non è arrivato a destinazione) attraverso **tutte le code di uscita del nodo** (nella direzione giusta)

Uso di contatori per limitare i passi di un messaggio

Uso di identificatori per evitare generazione senza fine
(per quanto tempo si mantiene lo stato sui nodi?)

ALGORITMI ULTERIORI ISOLATI

Algoritmo RANDOM Ogni messaggio viene smistato, se non a destinazione, usando una coda di uscita scelta a caso (non input)

Teorema per sistemi ideali di interconnessione (a messaggi)

Algoritmo di **routing ottimale** in un sistema dinamico con un **numero infinito** di nodi è una combinazione del **routing random**
se M mittente e D destinatario, si determina in modo **random un nodo R** (diverso da M e D) e si manda il messaggio in due hop: la **prima fase da M ad R e la seconda da R a D**

In sistemi globali le tabelle devono essere aggiornate spesso: algoritmi senza tabelle limitano l'overhead e possono consentire di raggiungere destinatari anche in movimento

In sistemi molto dinamici e a rapida variazione, **ma con indicazioni di località per le entità da raggiungere**, si cercano di determinare **direzioni di orientamento o polarizzazione** che possano orientare le decisioni non informate attuate da algoritmi locali

ALGORITMI TIPICI DI ROUTING

Si adottano tabelle di configurazione (tabelle di routing locali ai gateway). Algoritmi **GLOBALI** propagano globalmente informazioni in caso di modifica delle tabelle (scarsa scalabilità in caso di variazione)

Algoritmo **SHORTEST PATH FIRST** di Dijkstra

Ogni **nodo** possiede un **grafo completo** dell'**intera interconnessione**, stabilendo una **metrica di distanza** in base a pesi

con successive iterazioni, si calcolano le distanze minime per ogni nodo e il routing relativo

Il traffico di routing segue il **cammino più corto** determinato, a parte **casi particolari e situazioni di congestione**

Svantaggi / Vantaggi

- Costo della **propagazione** delle **valori** in caso di **variazione** (e in caso iniziale di creazione delle stesse)
- Possibilità di usare al meglio **tutte le risorse** esistenti

ALTRI ALGORITMI A CAMMINI MULTIPLI

Sono interessanti ed usati anche **algoritmi MULTIPATH** ossia strategie che permettano di utilizzare anche **più possibili percorsi verso uno stesso destinatario, sfruttando meglio le risorse**

Ogni nodo mantiene una tabella propria, con più possibilità per ogni destinazione, considerando più cammini possibili per il routing tra due nodi

Scelta random (probabilistica) del cammino, partendo dai cammini determinati per primi (anche con bilanciamento di carico)

Vantaggi

- Bilanciamento del traffico di routing
- Affidabilità in caso di guasti (anche multipli)

ALGORITMI ADATTATIVI

Algoritmi BACKWARD LEARNING

Ogni messaggio porta l'indicazione del **mittente** e, quindi, consente di inferire la **posizione del mittente stesso** ad ogni messaggio

I nodi intermedi possono stimare la distanze e la topologia

La fase iniziale di **apprendimento** dell'algoritmo deve lavorare in base ad una **politica**, da cui dipendono le stime successive

La conoscenza della topologia della **interconnessione completa** (globale) \Rightarrow permette di evitare situazioni di **ciclo** o **livelock** in cui un messaggio si perde in passaggi inutili ripetuti

Algoritmi globali costosi in ambiente dinamico

FAMIGLIE DI ALGORITMI DI ROUTING

Internet ha scelto due famiglie di Routing con caratteristiche diverse e in evoluzione storica (dinamicità ed efficienza)

Le famiglie sono il **Distance Vector** e il **Link state**

Il **Distance Vector** è la prima vera proposta Internet (a parte il random o flooding o simili per piccole reti)

Consente un routing dinamico ottenuto con un **protocollo distribuito tra router che si scambiano informazioni locali (con conoscenza crescente)** per costruire dei percorsi condivisi, producendo tabelle di routing locali

Il **Link State** ovvia ai problemi della famiglia precedente ed è il più usato adesso

Il protocollo **scambia informazioni tra i router in modo globale con un broadcast**, distribuendo informazioni consistenti in modo veloce, in caso di configurazione o variazione

IMPLEMENTAZIONI DEGLI ALGORITMI

Distance Vector

dinamico

single-path

Per ogni gateway, la tabella mantiene **la sola distanza in passi** e il **primo passo di uscita** per il routing. Le tabelle sono minime e consentono un istradamento statico facile

Problemi nei cambiamenti e nella propagazione delle variazioni

Link State

dinamico

multi-path

Ogni nodo **mantiene tutto il grafo** e tende a limitare la propagazione delle informazioni, rendendo facili anche cammini multipli

Variazioni sono propagate in broadcast, spesso uso di spanning tree

Algoritmi Spanning tree

statico

single-path

Si tende ad identificare un **albero di interconnessione** per ogni nodo della rete, consentendo la eliminazione dei cicli e determinando cammini senza problemi in modo globale

FAMIGLIE DI ALGORITMI DI ROUTING

Due famiglie principali globali, non adattative, basate su tabelle, e statiche (le tabelle possono anche cambiare)

Distance Vector	dinamico con overhead
Per ogni gateway, la tabella mantiene la sola distanza in passi e il primo passo di uscita per il routing . Le tabelle sono minime e consentono un istradamento statico facile	
Problemi nella costruzione delle tabelle, nei cambiamenti e nella propagazione delle variazioni	
Link State	dinamico con overhead

CREAZIONE TABELLE DISTANCE VECTOR

FASE di PROPAGAZIONE

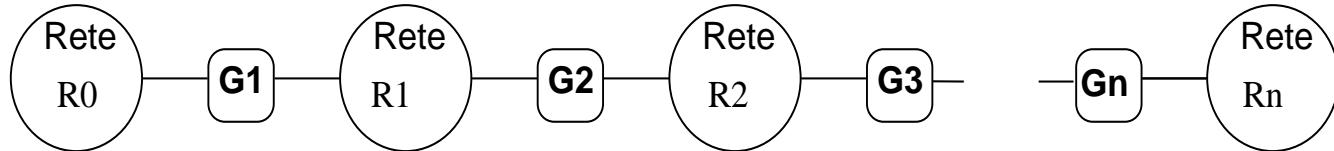


Tabelle al primo passo

R0 0	R1 0	R2 0	Rn-1 0
R1 0	R2 0	R3 0	Rn 0

Tabelle al secondo scambio

R0 0	R1 0	R2 0	Rn 0
R1 0	R2 0	R3 0	Rn-1 0
R2 1 G2	R0 1 G1	R1 1 G2	Rn-2 1 Gn-1
R3 1 G3		R4 1 G4	

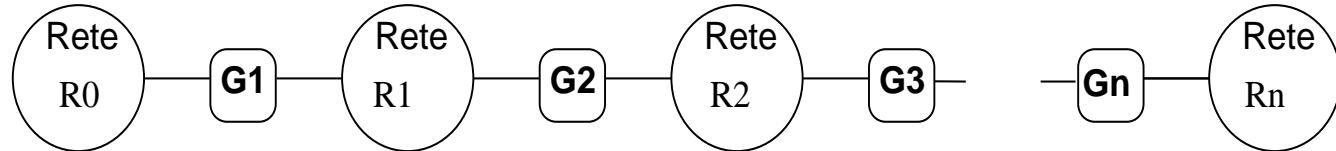
A regime, ogni gateway contiene la distanza di ogni rete

G1	G2	G3	Gn
R0 0	R1 0	R2 0	Rn 0
R1 0	R2 0	R3 0	Rn-1 0
R2 1 G2	R0 1 G1	R1 1 G2	Rn-2 1 Gn-1
R3 2 G2	R3 1 G3	R4 1 G4	Rn-3 2 Gn-1
...	Rn-4 3 Gn-1

PROPAGAZIONE DISTANCE VECTOR

**FASE di
PROPAGAZIONE**

Molto lenta



G1

R0	0
R1	0
R2	1 G2
R3	2 G2

G2

R1	0
R2	0
R0	1 G1
R3	1 G3
R4	2 G1

G3

R2	0
R3	0
R1	1 G2
R4	1 G4
R0	2 G2
R5	2 G4

Gn

Rn	0
Rn-1	0
Rn-2	1 Gn-1
Rn-3	2 Gn-1
Rn-4	3 Gn-1
Rn-5	4 Gn-1

G1

R0	0
R1	0
R2	1 G2
R3	2 G2
R4	3 G2
R5	4 G2

G2

R1	0
R2	0
R0	1 G1
R3	1 G3
R4	2 G1
R5	3 G3
R6	4 G1

G3

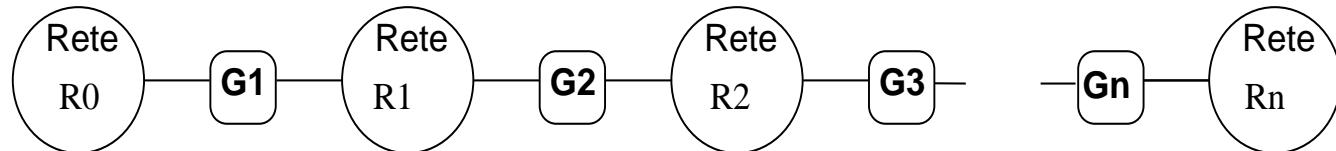
R2	0
R3	0
R1	1 G2
R4	1 G4
R0	2 G2
R5	2 G4
R6	3 G2
R7	4 G4

Gn

Rn	0
Rn-1	0
Rn-2	1 Gn-1
Rn-3	2 Gn-1
Rn-4	3 Gn-1
Rn-5	4 Gn-1
Rn-6	4 Gn-1

FORMAZIONE DELLE TABELLE DISTANCE VECTOR

FASE di
PROPAGAZIONE



Molto lenta (esponenziale nel numero dei nodi)

G1		
R0	0	
R1	0	
R2	1	G2
R3	2	G2
R4	3	G2
R5	4	G2
...		
...		
...		
Rn	n-1	G2

G2		
R1	0	
R2	0	
R0	1	G1
R3	1	G3
R4	2	G1
R5	3	G3
R6	4	G1
...		
...		
...		
Rn	n-2	G3

G3		
R2	0	
R3	0	
R1	1	G2
R4	1	G4
R0	2	G2
R5	2	G4
R6	3	G2
R7	4	G4
...		
Rn	n-3	G4

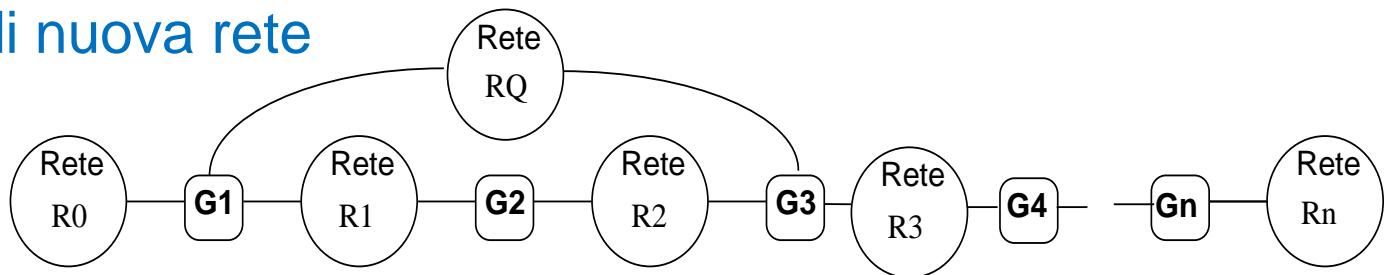
Gn		
Rn	0	
Rn-1	0	
Rn-2	1	Gn-1
Rn-3	2	Gn-1
Rn-4	3	Gn-1
Rn-5	4	Gn-1
Rn-6	4	Gn-1
...		
R1	n-2	Gn-1
R0	n-1	Gn-1

Intanto dobbiamo fare **routing**

PROPAGAZIONE VARIAZIONE

Variazione tabelle per una variazione di configurazione

Inserimento di nuova rete
e possibili
shortcut



G1

R0 0
R1 0
RQ 0
R2 1 G2
R3 2 G2
...

G2

R1 0
R2 0
RQ 0
R0 1 G1
R3 1 G3
R4 2 G3
...

G3

R2 0
R3 0
RQ 0
R1 1 G2
R0 2 G2
...

G4

R3 0
R4 0
R2 1 G3
R5 1 G5
R1 2 G3
R0 3 G3

Gn

Rn-1 0
Rn-2 0
Rn-3 1 Gn-1
Rn-4 2 Gn-1
Rn-5 3 Gn-1
...

G1

R0 0
R1 0
RQ 0
R2 1 G2
R3 1 G3
...

G2

R1 0
R2 0
RQ 0
R0 1 G1
R3 1 G3
R4 2 G3
...

G3

R2 0
R3 0
RQ 0
R1 1 G2
R0 1 G1
...

G4

R3 0
R4 0 ...
R2 1 G3
R1 2 G3
RQ 1 G3
R0 2 G3

Gn

Rn-1 0
Rn-2 0
Rn-3 1 Gn-1
Rn-4 2 Gn-1
Rn-5 3 Gn-1
...

PROPAGAZIONE CAMBIAMENTI IN DV

Ogni gateway decide la **politica di routing** in base alla **tabella di routing locale che costruisce interagendo con altri**

Propagazione **locale delle tabelle di routing** ad ogni vicino a tempi interni: le tabelle sono disseminate in modo **asincrono**

In genere, **si dissemina solo la rete e la propria distanza**

Chi riceve **una offerta** aggiorna **la propria tabella se la proposta è conveniente** in base alla metrica (ad esempio **distanza in hop**)

In caso di **CAMBIAMENTO** (crash o di aggiunta di nuovi gateway)

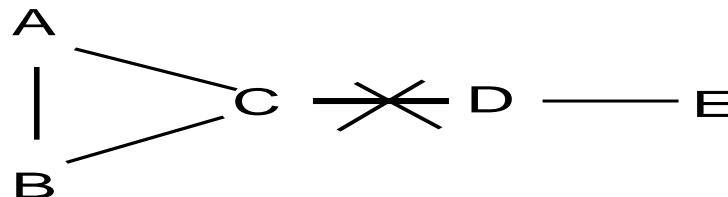
Possibilità di problemi (come cicli) e non convergenza delle tabelle che possono portare a situazioni non stabili

SVANTAGGI ULTERIORI

- tutti i messaggi seguono gli stessi cammini
- messaggi di aggiornamento con propagazione lunga e lenta
convergenza

PROBLEMI DEL DV: COUNTING-TO-INFINITY

Problemi in Riconfigurazione



In caso di guasto del link da D verso C, E ha un valore precedente e lo manda a D, poi ottiene il valore da D ed incrementa il valore locale, etc.

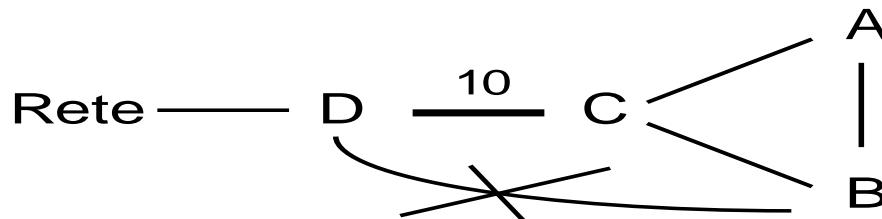
Producendo un **lento aumento fino all'infinito:**
counting-to-infinity

Problema generale dovuto al **non tenere traccia di chi fornisce una distanza da un nodo** (e cammino relativo) che rende possibile utilizzare l'offerta anche se non rilevante o affidabile

Spesso si usa la limitazione dell'infinito a 16

PROBLEMI DEL DV: INSTABILITÀ

Ancora problemi in Riconfigurazione



Per ogni nodo, si riporta la riga di tabella per raggiungere la rete Rete dopo il guasto del link BD (in figura la entry per Rete per ogni gateway)

prima Tabelle dopo

Rete 1	D	Rete 1
D 2	B	no
B 3	C	B 3
B 3	A	B 3

Rete 1
C 4
A 4
C 4

Rete 1
A 5
A 5
C 5

Tabelle fine

Rete 1
C 6
A 6
C 6

Rete 1
A 11
A 11
C 11

A , B e C si danno informazioni sbagliate l'uno all'altro

STRATEGIE MIGLIORATIVE NEL DV

Split Horizon

per evitare di passare informazioni sbagliate, **non si offrono cammini ai nodi da cui le abbiamo ottenute** (necessarie maggiori informazioni sulla topologia delle reti)

Si noti la lenta convergenza del sistema

le cattive notizie si propagano con lunghi intervalli di time-out

Si limitano i cicli di due nodi e non ulteriori

Hold-down

dopo una notifica di un **problema**, si **ignorano** le informazioni di cammino per un certo intervallo: tutti hanno modo di accorgersi del problema e non ci sono propagazioni errate

Il problema dei loop che si sono già creati non si risolve e vengono mantenuti durante l'hold-down

STRATEGIE MIGLIORATIVE NEL DV

Split Horizon con poisoned reverse e triggered broadcast

In caso di variazione, ogni nodo invia immediatamente un broadcast con la indicazione del problema ed il cammino

si usano evoluzioni dello split horizon con conoscenza di cammini (limite di 16)

A invia a C un messaggio di non raggiungibilità se crede di raggiungere D via C

C non può più rifarsi ad A (che pure non raggiungeva D)

Ulteriori problemi

Queste politiche generano fasi di broadcast in caso di variazioni

Evoluzione della famiglia di algoritmi per privilegiare efficienza nelle variazioni

FAMIGLIA DEI PROTOCOLLI LINK STATE (LS)

Algoritmi link-state

Ogni gateway mantiene una conoscenza completa della topologia di interconnessione (grafo completo)

Tabelle di routing calcolate da ogni nodo per tutte le reti, e basate sulla conoscenza dell'intero cammino: tipicamente percorsi unici da ogni nodo ad ogni rete

Il grafo di interconnessione, per evitare cicli, viene gestito con algoritmi che possono favorire decisioni locali ottime (routing dinamico)

Dijkstra shortest-path-first costruisce in ogni nodo un albero di comunicazione per ogni altra rete (come radice)

Il LS permette intrinsecamente la possibilità di fare source routing e anche di spedire messaggi diversi su cammini diversi (routing dinamico) e di utilizzare tutte le risorse di interconnessione

STRATEGIE LINK STATE (LS)

A REGIME, **ogni gateway tiene sotto controllo** le proprie connessioni e le verifica periodicamente

- **invio periodico di un messaggio ai vicini** per controllo della correttezza delle risorse locali
- **identificazione del guasto** (uso di messaggi specifici per evitare transitori in caso di variazione)

Non appena si verifica un **problema**, chi ha rilevato il problema invia il messaggio di variazione a tutti i partecipanti (**broadcast o flooding** del messaggio)

In sostanza le variazioni non sono dipendenti da possibili intermediari
I messaggi sono gli stessi qualunque sia la dimensione del sistema
(velocità di propagazione con triggering)

PROPRIETÀ DEGLI ALGORITMI LINK STATE (LS)

Vantaggi

- si controlla solo il **vicinato**
- azioni di **variazione** propagate **rapidamente** (senza ambiguità)
- possibilità di scelte **differenziate** dei cammini nella topologia
- conoscenza dei cammini **completi** e **source routing**

Svantaggi

- necessità di **mantenere tutta** la topologia
- azioni **costose** (broadcast) in caso di **variazione**

In generale, necessità di limitare i domini di conoscenza reciproca

Conclusione

Tutti i **protocolli dinamici** sono **poco scalabili** in caso di variazioni
(causa broadcast)

SHORTEST PATH FIRST SPF

Protocollo usato in sistemi Link State per determinare i percorsi da usare nel routing, univocamente e senza cicli (non multipath)

Algoritmo Shortest Path First (OSPF) Dijkstra

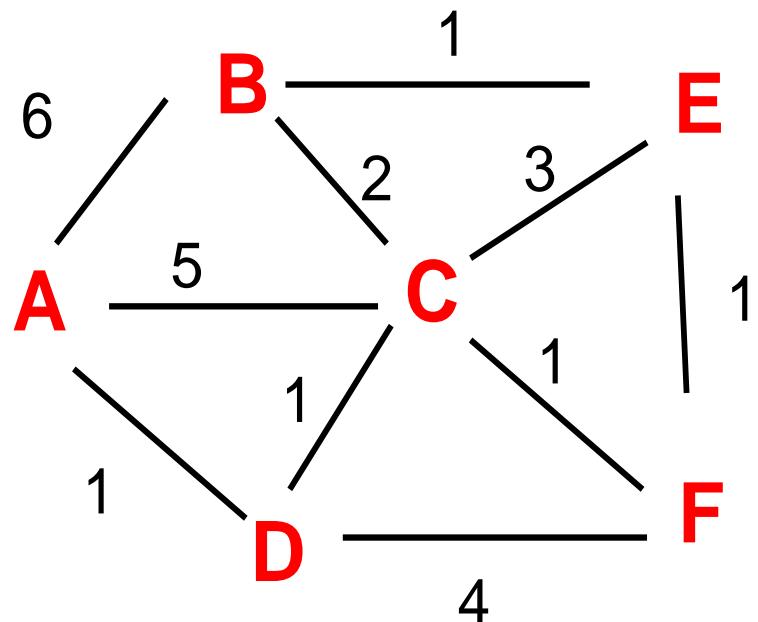
Ogni nodo costruisce una **propria tabella di cammini** in modo iterativo e la usa per il **normale routing**

Per ogni nodo, si deve memorizzare **il cammino** che permette di raggiungerlo, tenendo come base il **grafo di interconnessione** (che viene mantenuto aggiornato)

L'algoritmo permette di calcolare il **routing di tutti i cammini minimi** considerando l'albero che parte dal nodo stesso (spanning tree)

L'algoritmo funziona in **modo iterativo per passi**, e comincia a calcolare le distanze dai vicini adiacenti, poi i nodi di distanza due hop, poi fino a trovare un **albero completo di cammini minimi**

SHORTEST PATH FIRST DI DIJKSTRA



Notazioni

- **$c(i,j)$: costo collegamento** dal nodo i a j:
infinito se non c'è collegamento
- **$D(x)$: costo corrente del percorso**, dalla sorgente al nodo x
- **$p(x)$: precedente** (collegato a x) nel cammino dal sorgente a x
- **N : insieme di nodi** già verificati come cammino

ALGORITMO SPF

<Inizializzazione> $N = \{A\}$

for tutti i nodi x

do if (x è adiacente a A)

then $D(x) = c(A,x)$ else $D(x) = \text{infinito}$

<Ciclo principale>

do <consideriamo un **nodo** alla volta>

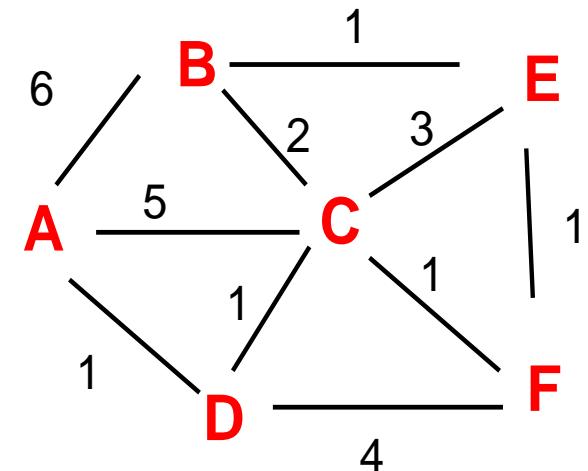
scelto un nodo y tale che $D(y)$ sia minimo, aggiungi y a N

aggiorna $D(x)$ per ogni x adiacente a y e non ancora in N :

$D(x) = \min(D(x), D(y) + c(y,x))$

< il nuovo costo fino a x è o il vecchio costo, oppure il costo del cammino più breve fino a y più il costo da y a x >

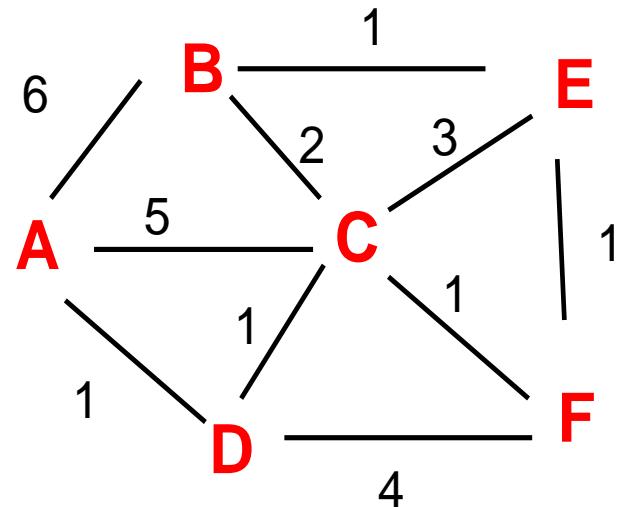
until tutti i nodi sono in N



PASSI PER ALGORITMO SPF

Si calcola a partire dal nodo A per ogni nodo possibile quali sono i percorsi minimi di raggiungibilità

Dopo N-1 cicli abbiamo aggiunto tutti i nodi e trovato tutti i percorsi minimi



Passo	Nodi Considerati	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
0	A	6,A	5,A	1,A	infinito	infinito
1	AB	6,A	5,A	1,A	7,B	infinito
2	ABC	6,A	5,A	1,A	7,B	6,C
3	ABCD	4,C	2,D	1,A	5,C	3,C
4	ABCDE	4,C	2,D	1,A	5,C	3,C
5	ABCDEF	4,C	2,D	1,A	4,F	3,C

IMPLEMENTAZIONE ROUTING INTERNET

Internet prevede una strategia precisa per **raggiungere tutti i possibili nodi** che possono intervenire **in una comunicazione** basata sulla **separazione delle reti** e sulla loro **interconnessione**

- **Ogni connessione appartiene ad una rete ed una sola**
per connessioni **punto-a-punto**
- **Ogni connessione è libera di indirizzare nella rete facendo operazioni solo locali e a basso costo**
per comunicazioni **punto-a-punto o broadcast**
- **Ogni connessione può indirizzare in Internet (in modo globale) ma deve usare opportuni intermediari**
routing previsto per Internet basato su **router** responsabili della comunicazione globale a costo più elevato

L'insieme delle reti Internet tende a minimizzare il costo di supporto del routing

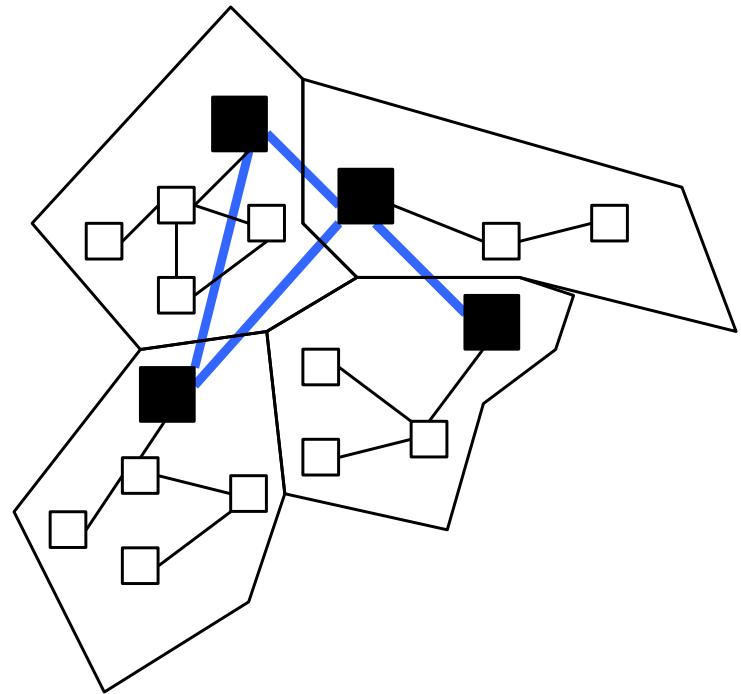
ROUTING INTERNET

In Internet si attua un Routing Gerarchico

per aree distinte di gestione
e con domini amministrativi diversi
unico protocollo di routing per area

La connessione tra le aree avviene attraverso gerarchie di router

**Routing per livelli
le informazioni di routing possono essere aggregate**



- level 1 router
- level 2 router

ROUTING GLOBALE INTERNET

Distinzione tra sistemi **core e noncore (ARPANET)**

core insieme di gateway chiave con **tabelle complete (e replicate)**

non core con informazioni di routing **solo parziali (e locali)**

i nodi CORE si scambiano tutte le informazioni di routing

(algoritmo **Distance Vector e Link State**)

Nella prima Internet, alcuni core e tutti gli altri non core in modo piatto

Scalabilità? problemi aumentando il numero delle reti

necessità di routing con astrazione e gerarchia

Introduzione dei **sottosistemi autonomi**

insieme di reti e gateway controllati da una autorità unica centrale, con proprie politiche di routing interne e non visibili

alcuni **gateway di controllo** provvedono al protocollo verso l'esterno

i **sistemi AUTONOMI** devono scambiarsi informazioni di routing e coordinamento solo intra-sistema

ROUTING GLOBALE INTERNET

Necessità di routing con astrazione e gerarchia

**Ogni sistema autonomo deve provvedere alla
comunicazione con l'esterno in modo predeterminato
comunicazione con l'interno in modo libero**

Exterior Gateway Protocol (EGP)

protocollo del **gateway di controllo** per trovare il percorso fino ai core
struttura ad albero con i core come radice

Interior Gateway Protocol (IGP)

protocollo per trovare il percorso all'**interno** di un **sistema autonomo**
(intra-sistema)

politica che consente percorsi multipli e con possibilità di tollerare i
guasti (algoritmi multipath IGRP CISCO)

ROUTING LOCALE INTERNET

Routing Information Protocol (RIP) implementato in routed UNIX
nodi attivi e passivi

ATTIVI partecipano a determinare i percorsi

PASSIVI restano ad ascoltare le decisioni degli altri

si manda un messaggio ogni 30 secondi nel vicinato con la tabella di routing locale

Si aggiornano le tabelle in base ai messaggi ricevuti: se i messaggi rilevano cammini più brevi di quelli noti si stabiliscono nuovi cammini

Un cammino ha un time-out associato e scade dopo un certo intervallo

Ogni nodo viene dichiarato guasto se non ha mandato un messaggio per un certo intervallo (180 sec)

Metrica senza costi di link e valore massimo a 10

Introduciamo una limitata capacità di riconfigurazione

Adatto solo per reti di piccole dimensioni

ROUTING

Routing Information Protocol ([RIP](#)) (con [split horizon triggered update poisoned reverse](#))

ispirato a Distance-Vector (con modifiche) basato su

- ruoli attivi e passivi
- broadcast (30 secondi) di messaggi di cambiamento
- mantiene vecchi cammini
- elimina problemi di non convergenza

Open SPF Protocol ([Link-State o Shortest Path First](#)) con servizi ulteriori

- cammini multipli e load balancing, cammini specifici
- introduzione di aree auto-contenute
- autenticazione
- definizione di risorse virtuali
- ottimizzazione delle risorse (broadcast)

ROUTING E RETI

Internet prevede una separazione tra reti

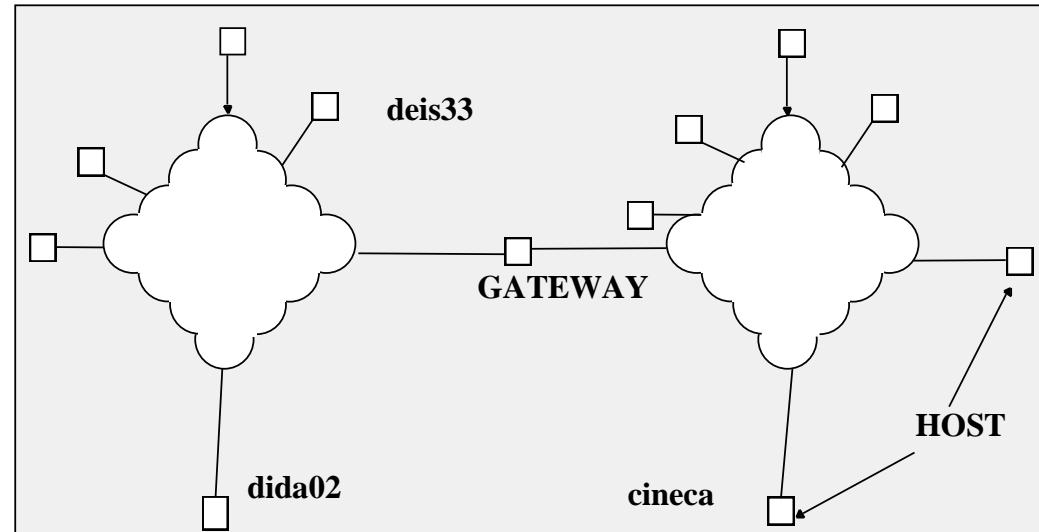
RETI uniche logicamente connesse

RETI fisiche separate

Indirizzamento diretto solo nell'ambito di nodi della stessa rete, altrimenti si devono usare **gateway**, ossia nodi intermediari con **almeno due indirizzi IP**, ossia connessioni su reti diverse che tendono a connettere

dida02 e deis33 si vedono perché in classe B stessa rete

dida02 e cineca non possono intervento di un router



SUBNETTING o SOTTORETI

Ulteriore protezione di restrizione su una rete e politica locale

Una rete può **essere divisa in sottoreti al suo interno introducendo maggiore granularità nella località** (all'esterno la suddivisione non è visibile in alcun modo)

La **sottrete** è rispettata **riconoscendola solo nella sottrete stessa**, anche rafforzandola tramite gateway interni

Ogni connessione su una sottrete

- **può comunicare** direttamente con ogni nodo della sola sottrete
- **non può comunicare** direttamente con nodi di altre sottoreti

In modo operativo, ogni nodo divide il **campo host in subnet e host**

In reti di classe B, subnet host suddiviso in 8 bit e 8 bit

dida01 137.204.56 subnet 56 deis33 137.204.57 subnet 57

e riconosce le proprie limitazioni, usando i router anche per comunicazioni all'interno della sua rete stessa

SUBNETTING: MASCHERE

Meccanismo: maschere di chiusura e protezione che segnalano le capacità locali alla driver di comunicazione

NETMASK come maschera di specifica di subnet per reti delle varie classi che determina le sottoreti stesse

NETMASK ad esempio maschera in classe B (per 3 byte)

11111111 11111111 11111111 0000000 o 255.255.255.000

La MASCHERA come insieme di bit a livello di rete che determina quali siano i limiti di comunicazione che richiedono un router apposito per uscire FUORI dalla SOTTORETE

La decisione di **mascherare è locale** ad ogni connessione e si potrebbe anche non rispettare (?)

DALL'ESTERNO DELLA RETE \Rightarrow il subnetting è invisibile e non produce **alcuna differenza sui nomi globali**

SUBNETTING: MASCHERE

DALL'INTERNO DELLA RETE ⇒ organizzazione diversa

quando il messaggio è arrivato alla rete, il **routing locale** deve **coordinarsi** per rendere attiva la **suddivisione**, usando un router per portare il messaggio alla corretta sottorete fino alla destinazione con un coordinamento di tabelle di routing

All'interno della rete, si devono individuare tutti i router per le altre sottoreti

network logica	network IP	gateway di routing
cineca	default	137.204.57.253
didalan	137.204.56	137.204.57.33
deislan	137.204.57	137.204.57.33
cciblan	137.204.58	137.204.57.33

Il **subnetting** rende possibili ulteriori suddivisioni dello spazio dei nomi IP (non deducibili automaticamente dal nome IP)

IP

SUCCESSIVAMENTE TROVATE QUELLO CHE AVETE GIÀ VISTO IN ALTRI CORSI

IP PROTOCOLLO E SPECIFICHE

LIVELLO NETWORK: INDIRIZZI IP

Per considerare un livello, dobbiamo partire dal sistema dei nomi

INDIRIZZAMENTO GERARCHICO a livello di IP

Ogni connessione di un host a una rete ha un indirizzo IP
unico di 32 bit, costituito di due parti, NETID, HOSTID

NETID **un identificatore di rete e**

HOSTID **un identificatore di host**

La distinzione nelle due parti, in gerarchia, facilita il routing
l'indirizzo individua le connessioni nella rete virtuale

- se un host usa una connessione diversa nella stessa rete, cambia **il suo IP, in particolare hostid** (non in dipendenza dalla locazione di accesso)
- se un host si collega in una rete diversa, anche con la stessa connessione, **cambia il suo IP, in particolare il netid, ma può mantenere hostid**
- host con diverse connessioni hanno più indirizzi (multiporta per gateway)

LIVELLO NETWORK: INDIRIZZI IP

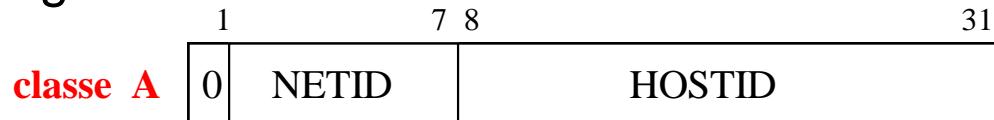
STANDARD

I **nomi di IP** sono dati di **autorità** dal Network Information Center (NIC) che assegna i numeri di rete, cioè informazione usata nei gateway per routing

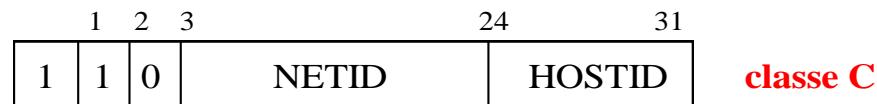
La parte di rete quindi è assegnata di autorità

La parte di nodo in IP è soggetta a **3 classi primarie** (in base al numero di reti e al numero di host collegabili) e differisce per **numero di bit** delle **singole parti**: analizzando un indirizzo IP si può distinguere la classe in modo automatico

Le WAN hanno generalmente un IP di classe A



Le LAN hanno un IP di classe B o C



LIVELLO NETWORK: INDIRIZZI IP

Per considerare un livello, dobbiamo partire dal sistema dei nomi

Parliamo di **protocollo IPv4 e nomi IP relativi** (a 32 bit): ogni protocollo deve definire i propri nomi

Un nodo è qualificato come Rete e Host, per un totale di 32 bit

Tre classi di indirizzi fisici (a byte) suddivisi in parte Network e Host

	0	1	2	3	4	8	16	24	31
classe A	0	netid					hostid		
classe B	1	0	netid				hostid		
classe C	1	1	0	netid			hostid		
classe D	1	1	1	0	indirizzo multicast				
classe E	1	1	1	1	0	indirizzi riservati ad usi futuri			

LIVELLO NETWORK: INDIRIZZI IP

Formati delle tre classi di indirizzi fisici (a byte): Network e Host

classe A:	Network	Host	es:	arpa	10
	0 7 bit	24 bit			
1 . # . # . # ..	126 . # . # . #		(127 riservato per usi locali)		
classe B:	Network	Host	almanet	137.204.#.#	
	10 14 bit	16 bit			
128 . 0 . # . # ..	191 . 255 . # . #				
classe C:	Network	Host	crbologna	192.94.70.#	
	110 21 bit	8 bit			
192 . 0 . 0 . # ..	223 . 255 . 255 . #				
classe D:	1110 28 bit	multicast			
224 . # . # . # ..	239 . # . # . #				

LIVELLO NETWORK

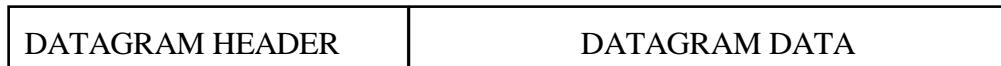
Cominciamo a considerare la struttura del **datagramma IP**

Un **DATAGRAMMA IP** è la **unità base di informazione** che viaggia in Internet

Suddiviso in due parti principali:

parte **intestazione**

parte **dati**



da **20 byte** ...

IP non specifica il **formato dell'area dati**, ma prescrive in modo preciso **la parte di header**, che ovviamente viene a contenere le parti dei protocolli superiori e ad essere incapsulata in un header (e footer) dei frame di più basso livello

PROTOCOLLO IP

Il protocollo IPv4 prescrive come si deve realmente implementare l'instradamento dei datagrammi, obiettivo fondamentale del livello

Il PROTOCOLLO prescrive per ogni nodo che deve comportarsi come un **router**, ossia fare **routing**, due funzioni principali da svolgere

- **elaborazione del messaggio** del livello superiore nel formato per la trasmissione

- encapsulamento / frammentazione

il datagramma deve contenere le informazioni di livello superiore, eventualmente i dati devono essere frammentati (se possibile)

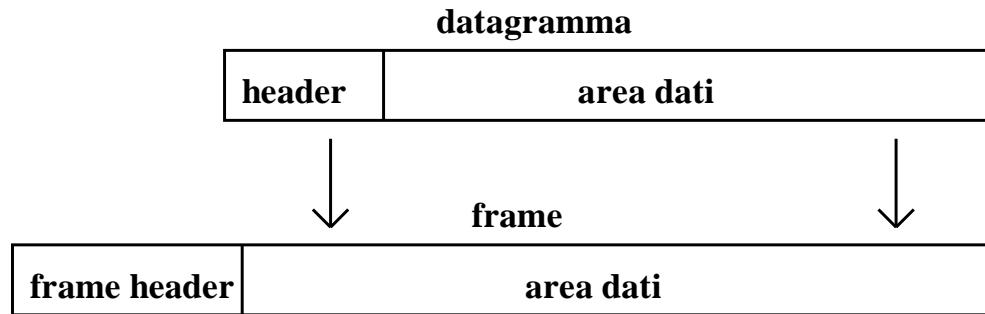
- **intradamento (routing)** cioè:

- traduzione da indirizzo logico IP a indirizzo fisico di frame (ARP)
- scelta della porta di uscita (in base al percorso)

INVIO DI DATAGRAMMA IP

Un datagramma deve essere **smistato** da un nodo ad un nodo successivo secondo le normali regole di encapsulamento

Ogni nodo deve **incapsulare** ogni **datagramma** in un frame di livello data link



Ovviamente questo avviene anche per ogni singolo datagramma, e senza legami tra i diversi frammenti e datagrammi, che vengono trattati in modo indipendente dalla driver di routing (FIFO)

FORMATO DELL'HEADER E DATI IP

In un datagramma: Header (minimo 20 byte, max 64), oltre ai Dati

0	4	8	16	19	24	31	Parole
VERS	HLEN	SRV TP		TOTAL LENGTH			1
				FLAGS	FRAGMENT OFFSET		2
TIME TO LIVE	PROTCL			HEADER CHECKSUM			3
			SOURCE IP ADDRESS				4
			DESTINATION IP ADDRESS				5
IP OPTIONS	(if any)			PADDING			(6..16)
		DATA					6
		...					

0	3	SeRvice TyPe
PRECEDENCE	D	T R C UNUSED

FLAGS		
Do not fragment	More fragments	UNUSED

CAMPI DELL'HEADER IP

5 parole di 32 bit (o più, se attive opzioni)

1) **versione** del protocollo, **lunghezza header e totale**, service **type**

- precedenza (0-7)
- tipo di trasporto desiderato (bit di throughput T, di affidabilità R, di ritardo D, di costo C)

2) **identificazione** del datagramma (usato per ricomporre i frammenti)

- flag (3 bit: non frammentare e altri frammenti) e frammentazione (13 bit moltiplicato x8)

3) **time to live**, tempo di permanenza del datagramma

- tipo di protocollo superiore (TCP 6, UDP 17, ICMP 1, ...)
- checksum per il controllo (complemento a 1 a 16 bit)

4-5) indirizzo **IP sorgente e destinazione**

oltre) **opzioni**: monitoraggio e controllo rete

HEADER IP

In generale, IP protocollo senza qualità

I datagrammi viaggiano in modo indipendente e autonomo (anche come sottoparti o frammenti)

IP il protocollo di base di Internet specifica anche il routing

Service type, diviso in:

- **precedenza** (primi tre bit service type)
- indicazioni **di preferenza di scelta** tra route
 - **D** minimo delay
 - **T** massimo throughput
 - **R** massima reliability
 - **C** minimo costo
- un bit non usato (1 bit non usato, 7 utili)

IP E QoS (QUALITY OF SERVICE)

In generale, IP protocollo base di Internet senza qualità

IP è un protocollo best-effort a basso costo e non garantisce Quality of Service **QoS**

Ma ci sono evoluzioni in atto per garantire QoS ...

Service type, anche come **codepoint**, ossia possibilità di identificare raggruppamenti del traffico e di dividere in classi il traffico

- **tipo di traffico** (ultimi due bit del service type)
- **indicazioni di precedenza** tra classi di traffico
 - tre classi diverse distinte dai bit finali (0 11 01)
 - relativa precedenza

Esempio: **x x x x x 0**

Le classi differenziate permettono di dividere, distinguere e classificare il traffico

LIVELLO IP

IPv4 specifica il **servizio accoppiato al relativo protocollo**

Il **servizio** è quello che ci impegniamo a fornire ai livelli superiori

Il **protocollo** è la specifica di come si deve lavorare nella realizzazione

Per il SERVIZIO si specifica che il tutto è:

connectionless: ciascun pacchetto è trattato **indipendentemente** dagli altri. Diversi pacchetti possono seguire percorsi diversi ed essere consegnati fuori ordine

unreliable: la consegna **non è garantita**, cioè non si effettua alcun controllo sull'avvenuta ricezione di un pacchetto

best-effort: l'inaffidabilità del trasferimento è dovuta a **cause esterne** e non al software di rete; nessun messaggio di errore è previsto per il richiedente in caso di non inoltro o di perdita

DATAGRAMMA IP

Un datagramma deve essere mosso da un nodo ad un altro e **ogni intermediario può operare sul messaggio**, a partire dal mittente

Ogni nodo può frammentare il datagramma

- **DECOMPOSIZIONE** al mittente
- **DECOMPOSIZIONE** ad ogni intermedio
- **RICOMPOSIZIONE** solo al destinatario

header datagramma	data 1	data 2	data 3
----------------------	--------	--------	--------

header frammento	data 1	frammento 1
---------------------	--------	-------------

header frammento	data 2	frammento 2
---------------------	--------	-------------

header frammento	data 3	frammento 3
---------------------	--------	-------------

FRAMMENTAZIONE DATAGRAMMA IP

I datagrammi devono essere incapsulati nei frame di livello data link su cui transitano in base alla **MTU** (maximum transfer unit), ossia la lunghezza **massima dei frame** a livello **fisico** (di Internet cioè 2 OSI)

Come si può determinare la **dimensione massima** del datagramma dalla partenza all'arrivo?

1° possibilità: **calcolo statico da parte del mittente**

il datagramma incapsulato nel singolo frame fisico (dimensioni del datagramma minore o uguale alla più piccola MTU presente in Internet)

- a livello utente trasmissione con tempi molto lunghi per il trasferimento di un messaggio (se MTU molto piccole)

- **efficiente solo per reti fisiche con MTU a lunghezza elevata ed omogenea**

FRAMMENTAZIONE DATAGRAMMA IP

2° possibilità: decisione passo passo (QUELLA USATA)

MTU scelta **indipendente dalle tecnologie sottostanti** per rendere efficiente la comunicazione a livello utente (fissata tipicamente a 64Kbyte)

Il pacchetto originale viene suddiviso in frammenti su reti con MTU a dimensione inferiore (a 64Kbyte)

La frammentazione del pacchetto può avvenire **ad ogni passo nelle reti intermedie** e potendo fare solo al **destinatario la ricomposizione o riassemblaggio** dei diversi frammenti che possono essere stati attuati durante l'instradamento

Il destinatario riceve i diversi **frammenti**, li identifica in base **allo stesso ID** e li mette insieme in base **all'offset**: se l'intero **datagramma** è stato ricevuto, allora viene considerato; altrimenti, il tutto viene eliminato

OPZIONI: MONITORAGGIO E CONTROLLO RETE

Alcune opzioni interessanti, con informazioni diverse memorizzate sulla parte di opzioni, sono:

record route: genera una **lista degli indirizzi IP** dei gateway che il frame ha attraversato (al massimo 9 intermedi)

⇒ **otteniamo una indicazione dei gateway intermedi attraversati dal datagramma**

timestamp: genera una **lista dei tempi di attraversamento** sugli intermedi

⇒ **possiamo ottenere una indicazione dei tempi di passaggio e della permanenza del datagramma nei gateway intermedi (vedi mail)**

per attuare anche azioni correttive e cercare strade diverse

OPZIONI: SOURCE ROUTE

source route o instradamento al sorgente: il sorgente fornisce indicazioni sul cammino da seguire nel routing del frame

- **strict source:** il datagramma porta nella parte opzione una indicazione di tutti i gateway intermedi da attraversare
- **loose source:** indicazione di un insieme di percorsi da attraversare non in modo contiguo ed unico

Numero massimo di parole nella parte opzione del datagramma
**limite al controllo del percorso:
al massimo spazio per registrare 9 passi**