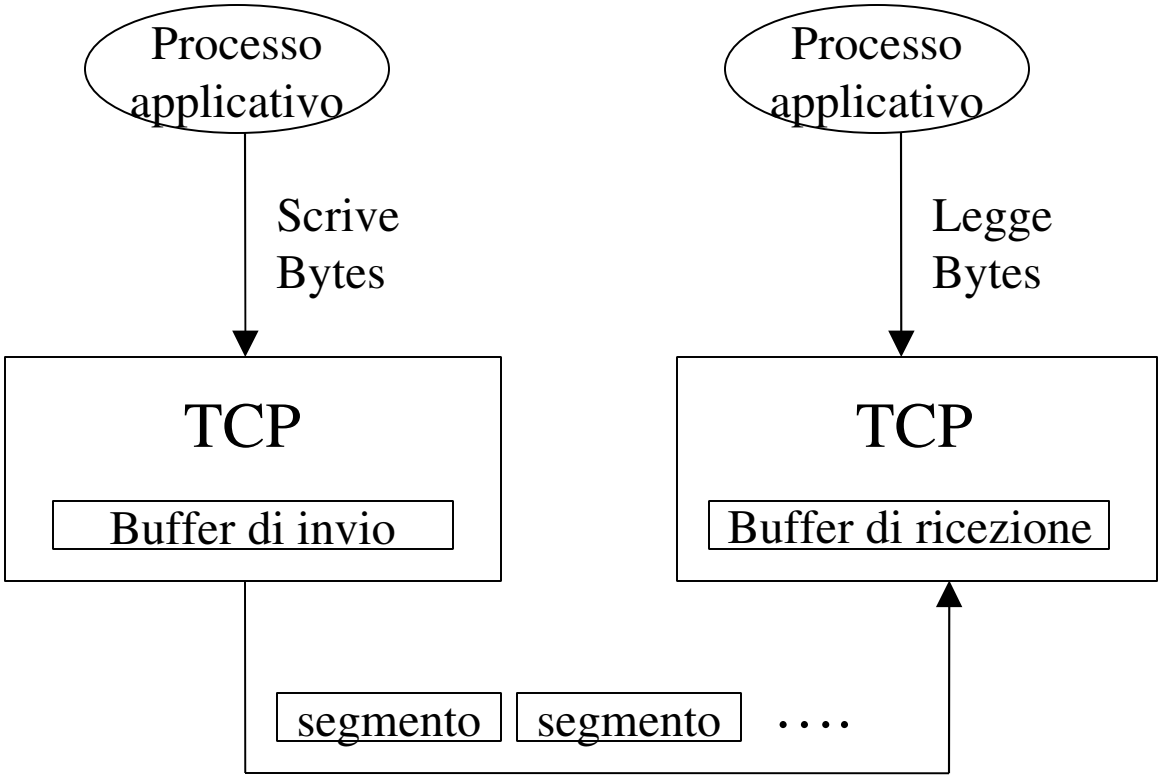
CORSO DI RETI SSIS

Lezione n.4

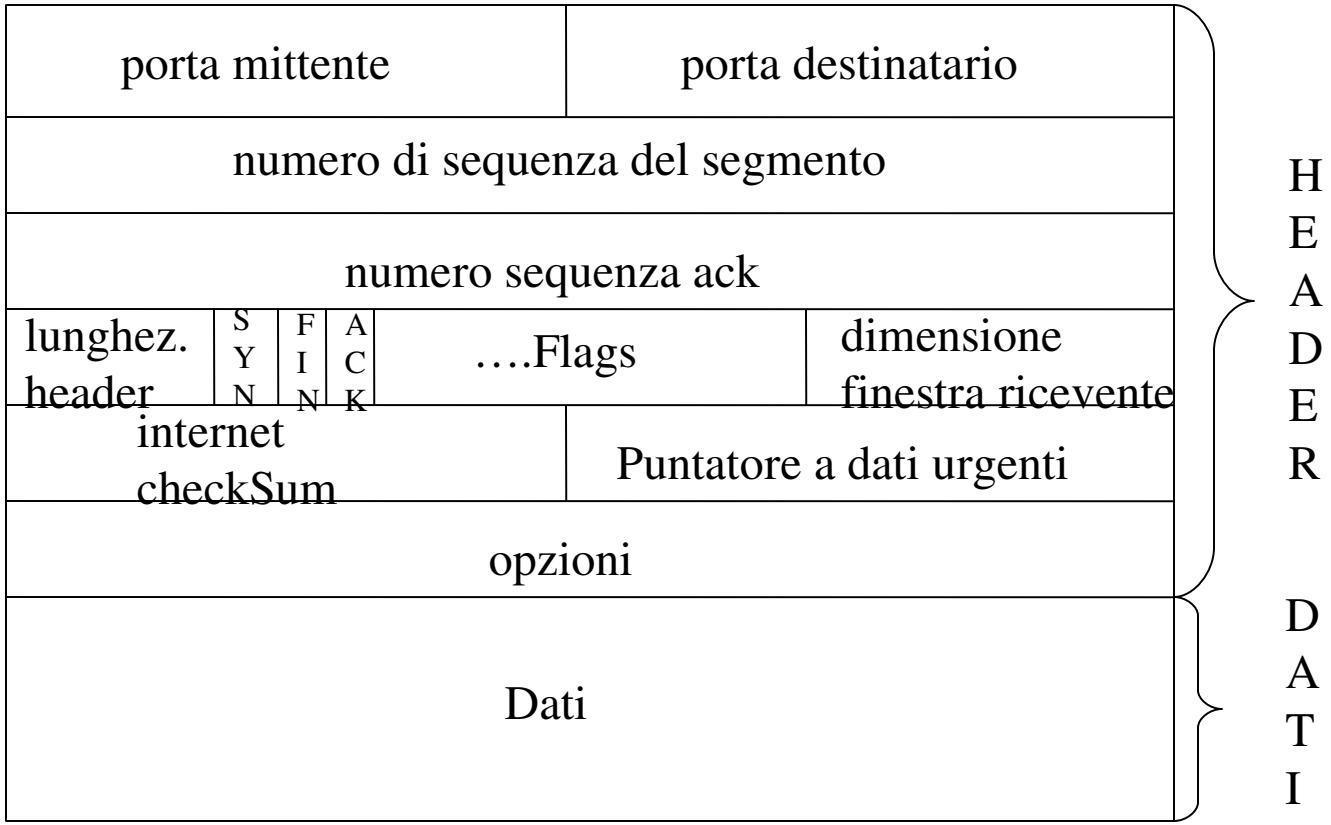
16 novembre 2005

Laura Ricci

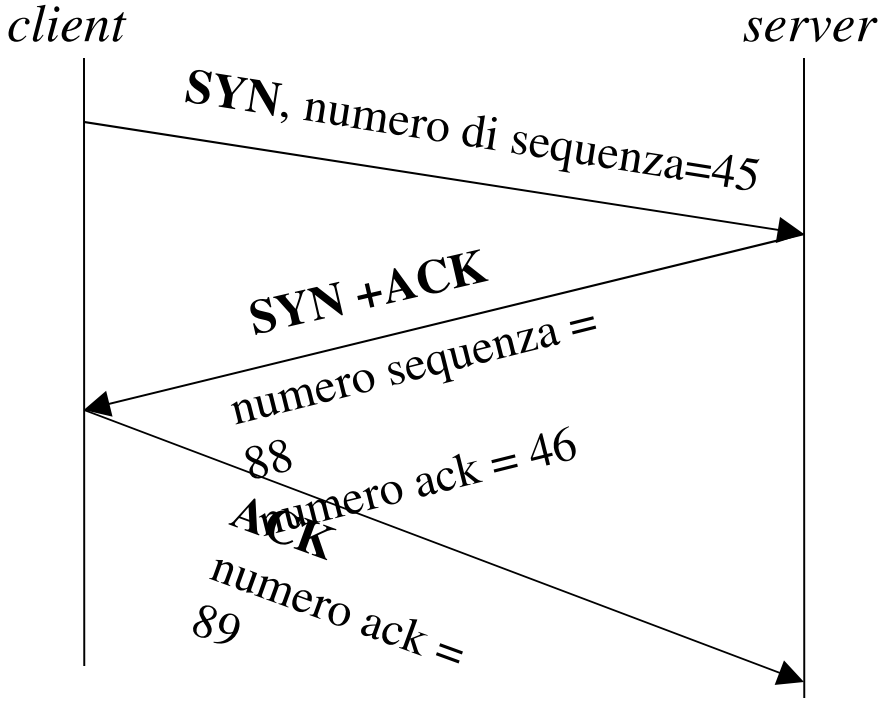
TCP: INVIO DI UN FLUSSO DI BYTES



SEGMENTIO TCP: STRUTTURA

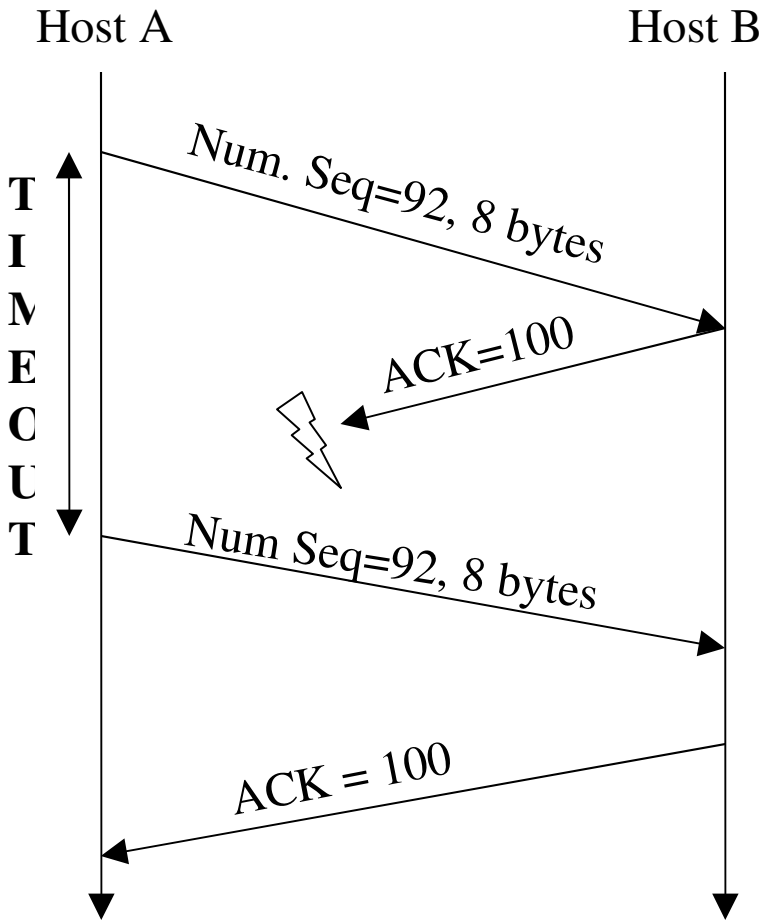


TCP: THREE WAY HANDSHAKE

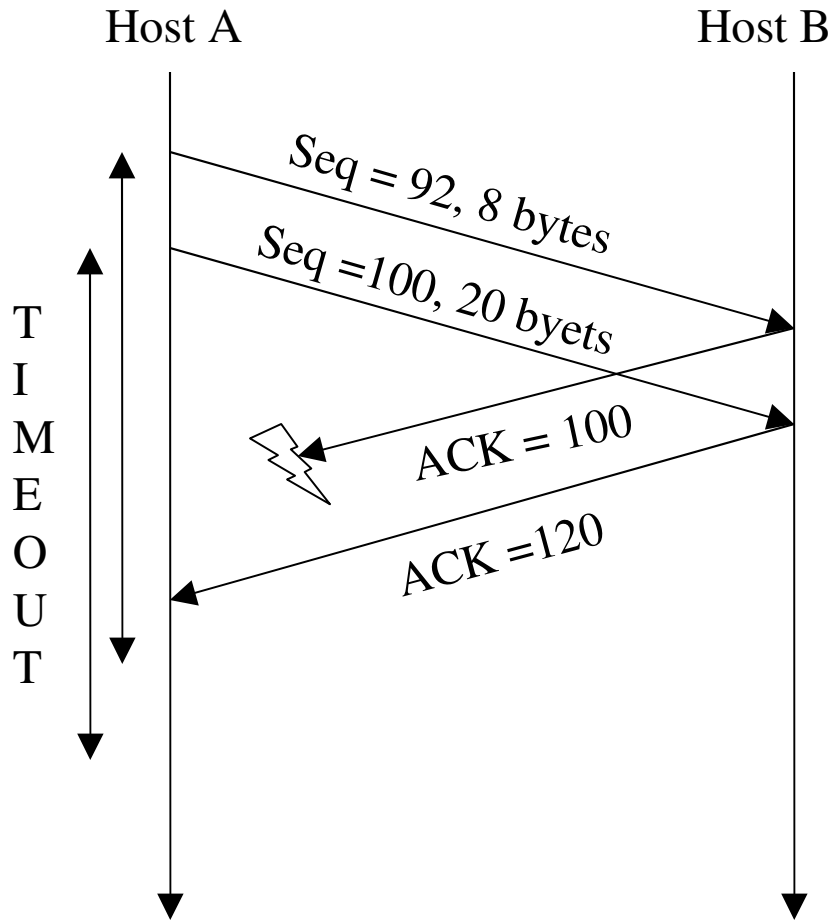


• *segmento 1:* il client comunica al server il numero di sequenza che attribuirà al primo segmento dati (generato casualmente, 45)
• *segmento 2:* il server conferma la ricezione dal client ecomunica il proprio numero iniziale di sequenza (generato casualmente, 88)
• *segmento 3:* il client conferma il numero di sequenza ricevuto dal server

TCP: TRASMISSIONE DI DATI



TCP: TRASMISSIONE DI DATI



Cumulative acknowledgement

TCP: CONTROLLO DELLA CONGESTIONE

- *Problema:* i datagram (pacchetti) possono essere persi durante il loro percorso sulla rete IP.

- la perdita di pacchetti, in generale, deriva dalla *congestione* dei routers, che comporta, in generale, il traboccamento (overflow) dei buffers da essi gestiti

- Senza alcun mccanismo di controllo, il fenomeno della congestione è *autoalimentante*:
  - un router elimina uno o più pacchetti
  - l'host che ha inviato i pacchetti, quando vede scadere i loro timeout, li rinvia
  - il router continua ad essere congestionato per il quantità dei pacchetti in arrivo

  $\Rightarrow$

  è necessario un meccanismo per il controllo della congestione

TCP: CONTROLLO DELLA CONGESTIONE

Approcci per il controllo della congestione:

- *controllo della congestione supportato dai routers:* i routers notificanom esplicitamente il loro stato di congestione agli end systems (hosts). Gli hosts regolano di conseguenza il flusso dei segmenti spediti sulla rete

- *controllo della congestione end to end*:

  - i routers non forniscono alcun supporto per il controllo della congestione.
  - gli host decidono autonomamente quando la rete risulta congestionata in base all'osservazione del *comportamento della rete* in seguito all'invio di segmenti TCP

TCP: CONTROLLO DELLA CONGESTIONE

Controllo della congestione in TCP è basato principalmente sulla seguente osservazione:

- se la rete è congestionata, inizierà a scartare dei pacchetti

- l'host non riceve l'ack per il pacchetto scartato

  $\Rightarrow$

  interpreta lo *scadere del timeout* associato ad un pacchetto inviato come indicazione dello stato di congestione della rete.

  $\Rightarrow$

  rallenta la spedizione dei pacchetti sulla rete in modo da diminuire lo stato di congestione della rete

TCP: CONTROLLO DELLA CONGESTIONE

Variabili necessarie per il *controllo della congestione*

Finestra di Congestione (FG)= numero di dati inviati e non confermati
Soglia (S) = determina come variare il valore della finestra di congestione

Il valore della finestra di congestione viene modificato a seconda del livello di congestione percepito dall' host

- si parte con un valore molto basso di FG e si aumenta rapidamente (*crescita esponenziale)*
- quando la dimensione di FG eccede il valore di S, la crescita rallenta (*crescita lineare)*
- quando scade un time-out di un pacchetto non confermato, il valore di FG viene *diminuito* (dimezzato).

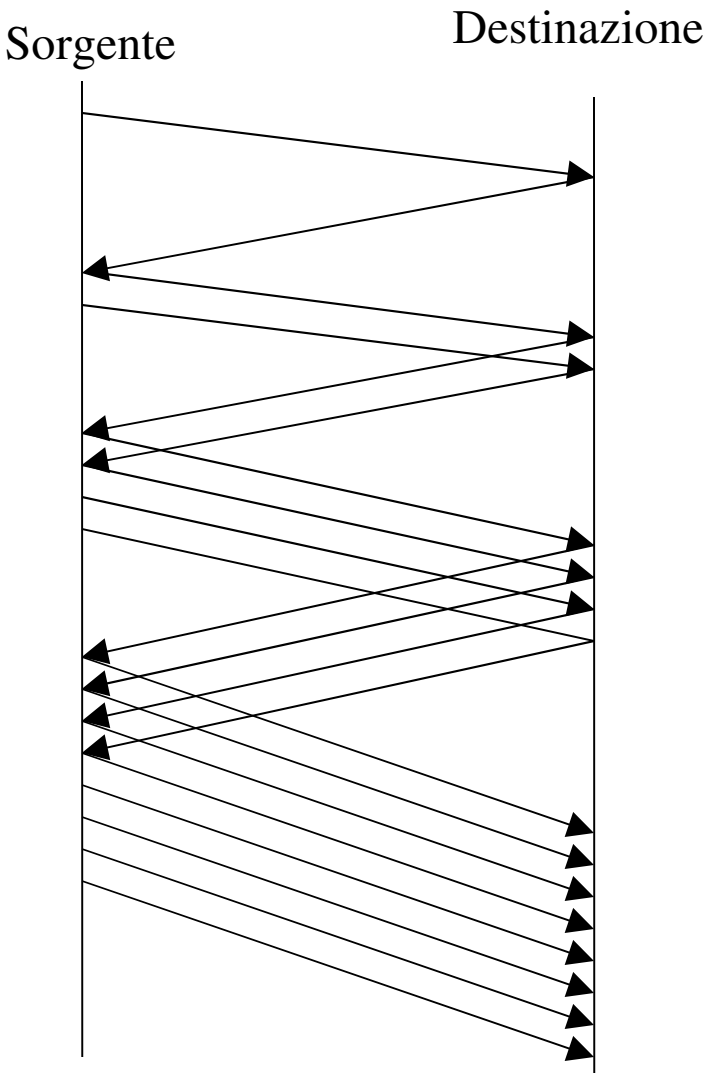
TCP: SLOW START

- all'inizio FG=1 (1 solo segmento in transito, senza essere confermato)
- se il segmento viene confermato, prima dello scadere del time-out, FG=2

  $\Rightarrow$

  si possono inviare due segmenti prima di aspettare la conferma
- Se i due segmanti vengono confermati prime dei rispettivi time-out, si aumento FG di una unità per ogni ack ricevuto, FG=4
- E cosi' via…..

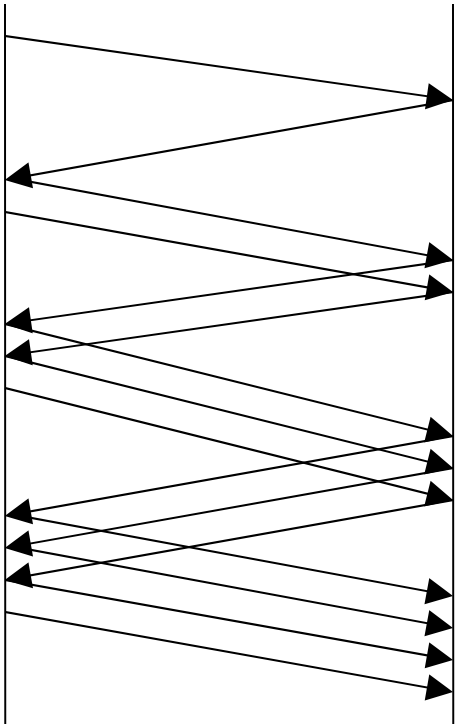
TCP: SLOW START



TCP: CONTROLLO DELLA CONGESTIONE

Quando il valore di FG supera il valore di S (soglia):

- la fase di slow-start terminaed inizia la fase di *aumento additivo* della finestra di congestione
- si inviano FG segmenti. Se vengono tutti confermati prima dello scadere di un time out, si aumenta la dimensione di FG di una unità



TCP: CONTROLLO DELLA CONGESTIONE

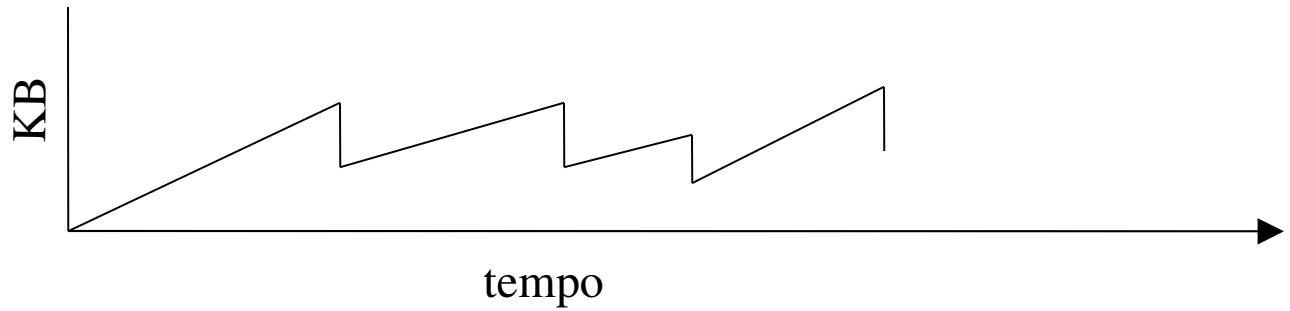
Nel caso in cui venga rilevata la perdita di un segmento (il segmanto non viene confermato prima dello scadere del time-out)

$\Rightarrow$

Il valore di FG viene dimezzato

Strategia complessiva, se non si considera la fase di slow start:

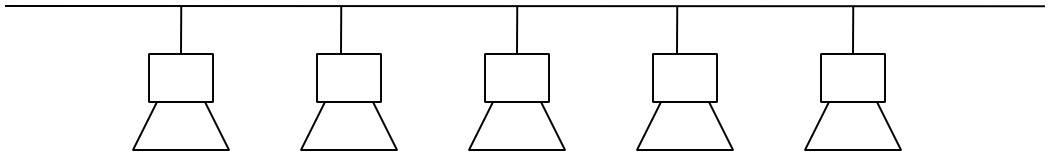
AIMD = Additive Increase, Multiplicative Decrease



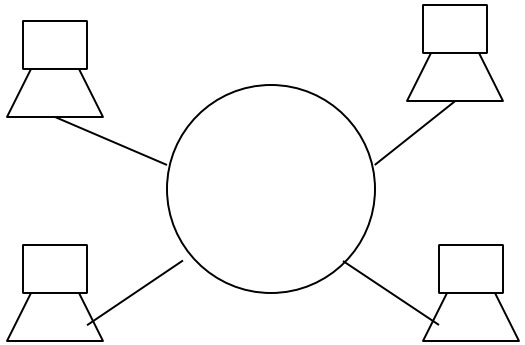
INTERCONNESSIONE DI RETI: IL LIVELLO IP

- *Rete elementare* = gli host sono connessi direttamente mediante un mezzo fisico, ad esempio un cavo o una fibra.

- Tecnologie disponibili:
  - *Ethernet* (cavo coassiale condiviso, bus)

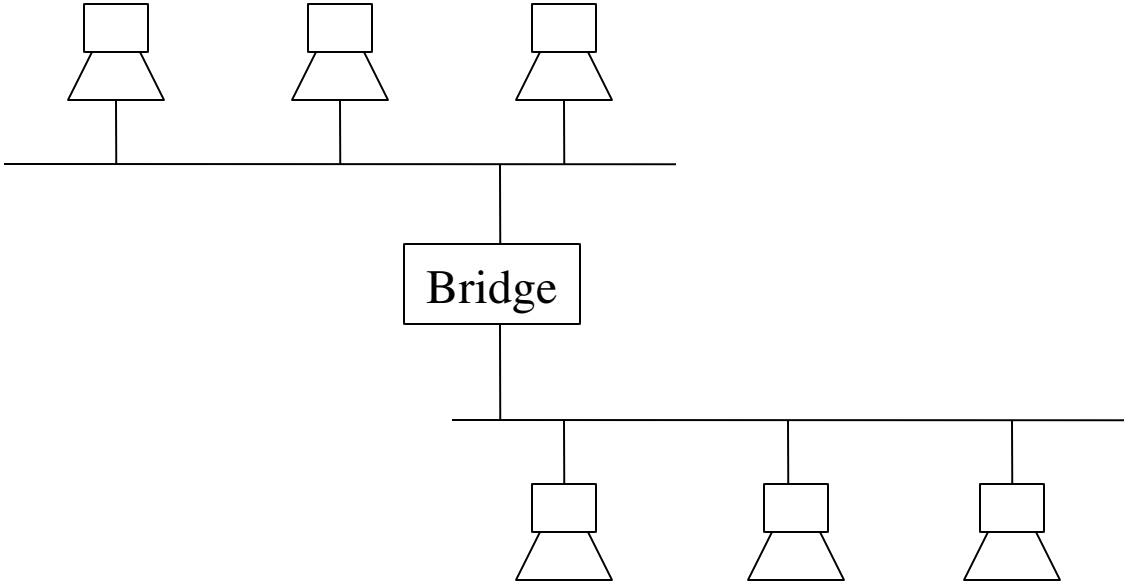


  - *Token Ring* (FDDI, anello condiviso)



INTERCONNESSIONE DI RETI: IL LIVELLO IP

Rete locale estesa = insieme di reti elementari interconnesso mediante un insieme di commutatori (*bridge*).



Bridge = è in grado di eseguire solo il *livello link* ed il *livello fisico* dello stack TCP-IP

INTERCONNESSIONE DI RETI: IL LIVELLO IP

La costruzione di reti estese mediante bridges e commutatori è limitata da diversi fattori

- *scalabilità:* una rete estesa non può contenere un numero troppo elevato di hosts
- *eterogeneità limitata:* una rete estesa utilizza una unica tecnologia

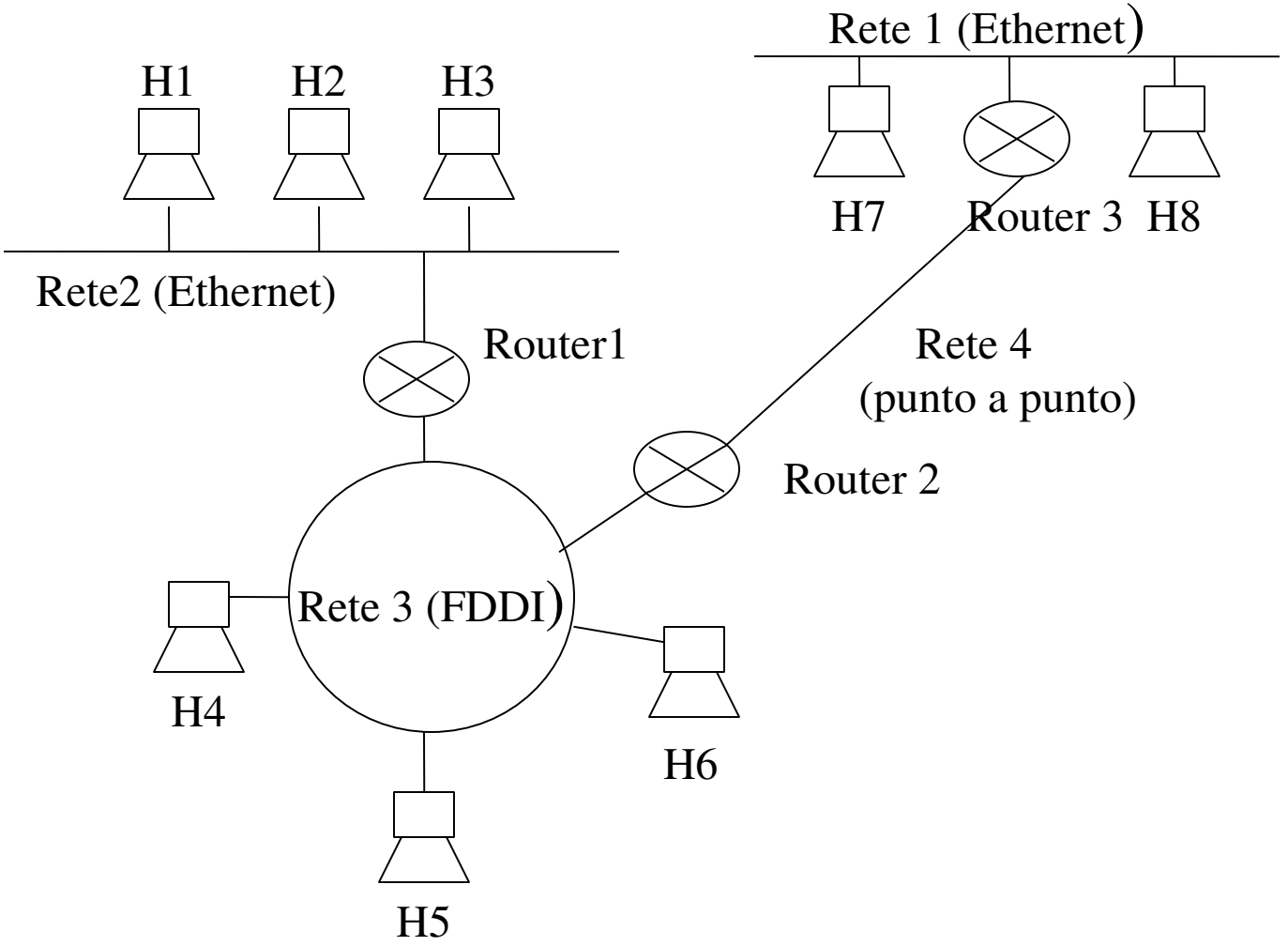
$\Rightarrow$

la costruzione di una rete eterogenea di grosse dimensioni (internet) richiede l'utilizzo di diverse tecnologie

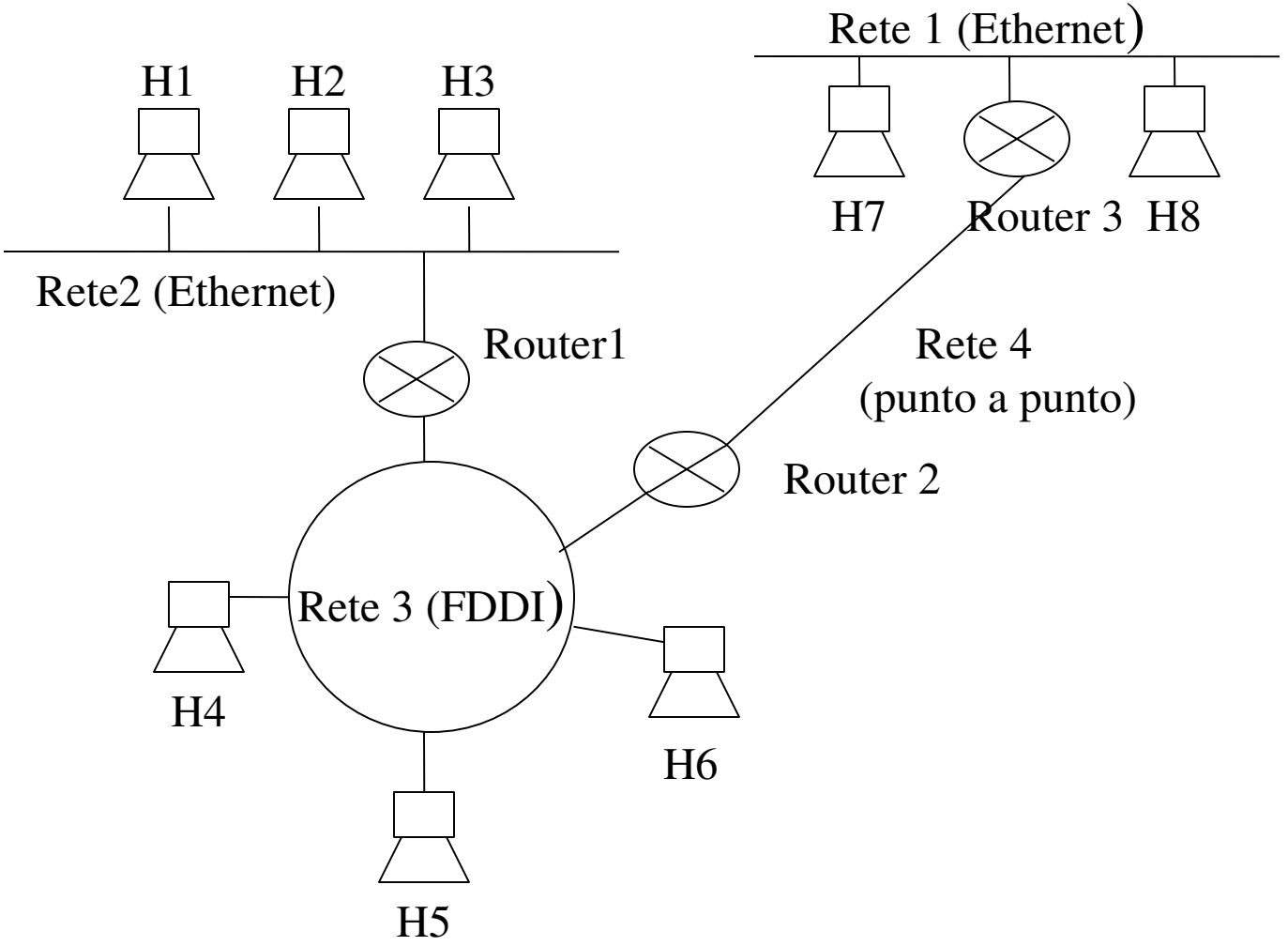
$\Rightarrow$

protocollo IP(internet protocol): permette l'instradamento di messaggi tra reti estese diverse utilizzando componenti specializzate (*routers o gateway*)

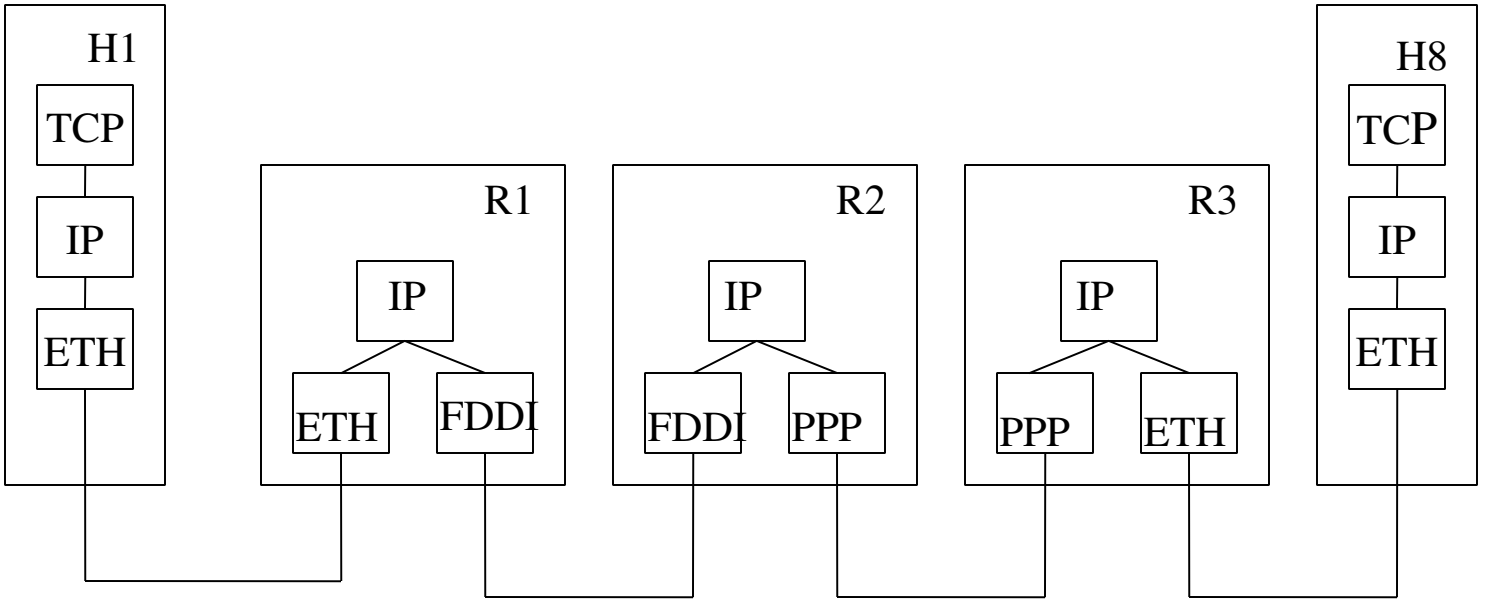
INTERNETWORKS (INTERNETS)



INTERNETWORKS (INTERNETS)



INTERNETWORKS (INTERNETS)



INTERNETWORKS (INTERNETS)

- internetwork(internet) = insieme di reti qualsiasi interconnesse che utilizzano un servizio di consegna di pacchetti tra hosts

- *Esempi:*

  una grossa azienda con molte sedi interconnette le LAN presenti nelle diverse sedi e costruisce una propria *internet privata*

  Internet (con l'iniziale maiuscola): la rete mondiale a cui sono interconnesse quasi tutte le reti

- *Terminologia:*

  rete fisica= rete a connessione diretta oppure rete commutata che utilizza una unica tecnologia

  rete logica = rete (internet) costruita interconnettendo un insieme di reti fisiche

LIVELLO IP: CARATTERISTICHE GNERALI

- Modello di servizio di IP: invio di datagrams secondo la filosofia *best effort.*

- *datagram (*pacchetto IP) : contiene informazioni sufficienti perché la rete (i routers) possa inoltrarlo verso la destinazione

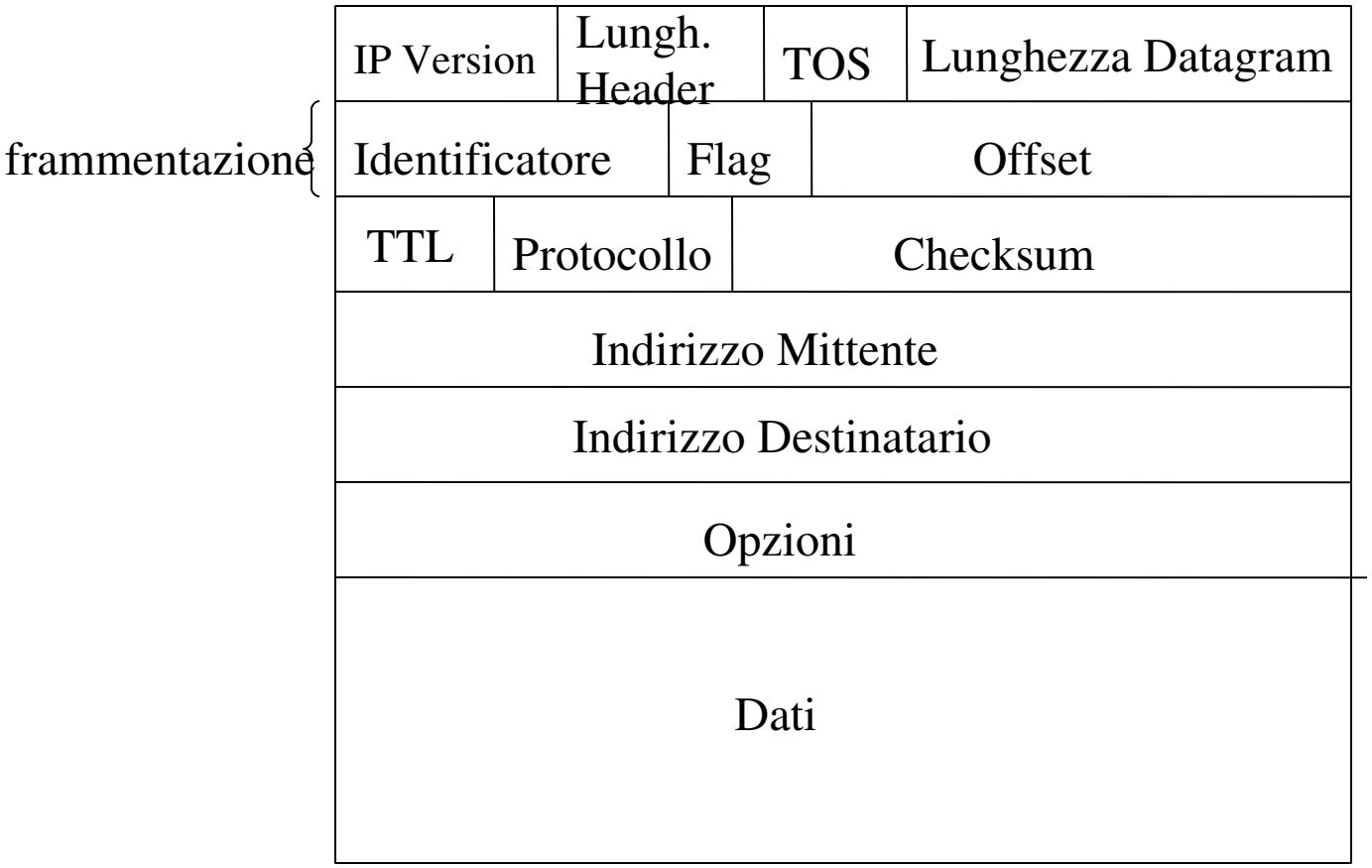
- *Best effort:* Ip fornisce un servizio *minimale*. Motivazione: possibilità di eseguire il protocollo su reti che utilizzano diverse tecnologie

  $\Rightarrow$

  IP attualmente può essere eseguito su qualsiasi rete, anche su reti con tecnologie non esistenti al momento della sua definizione

  IP può operare su una rete che trasporta i messaggi mediante piccioni viaggiatori

LIVELLO IP: FORMATO DEL PACCHETTO



LIVELLO IP: FORMATO DEL PACCHETTO

IP Version: IPV4 / IPV6

TOS (*Type of Service*) Consente un trattamento differenziato dei pacchetti.
Esempio: un particolare valore di TOS indica che il pacchetto ha una priorità maggiore rispetto agli altri.
Utile per distinguere per distinguere tipi diversi di traffico (traffico real time, messaggi per la gestione della rete,..)

TTL – Time to Live Consente di limitare la diffusione del pacchetto sulla rete

- valore iniziale impostato dal mittente
- quando il pacchetto attraversa un router, il valore viene decrementato
- quando il valore diventa 0, il pacchetto viene scartato

Introdotto per evitare *percorsi circolari* infiniti del pacchetto. Utilizzato anche per limitare la diffusione del pacchetto nel multicast

FRAMMENTAZIONE E RICOSTRUZIONE DI PACCHETTI IP

Trasmissione di pacchetti IP su reti fisiche diverse *ed eterogenee*:

- quando un pacchetto P viene spedito su una rete fisica R, P deve essere *incapsulato* in un pacchetto P_{link}.

- la dimensione di P_{link} dipende dal *livello link* della rete fisica R, in particolare dall' *MTU* (*Maximum Transfer Unit)* di quella rete

  Reti fisiche diverse $\Rightarrow$ MTU diversi

  Ethernet: MTU = 1500 bytes

  FDDI : MTU = 45500 bytes

- la dimensione di P_{link} in genere è diversa da quella di P

FRAMMENTAZIONE E RICOSTRUZIONE DI PACCHETTI IP

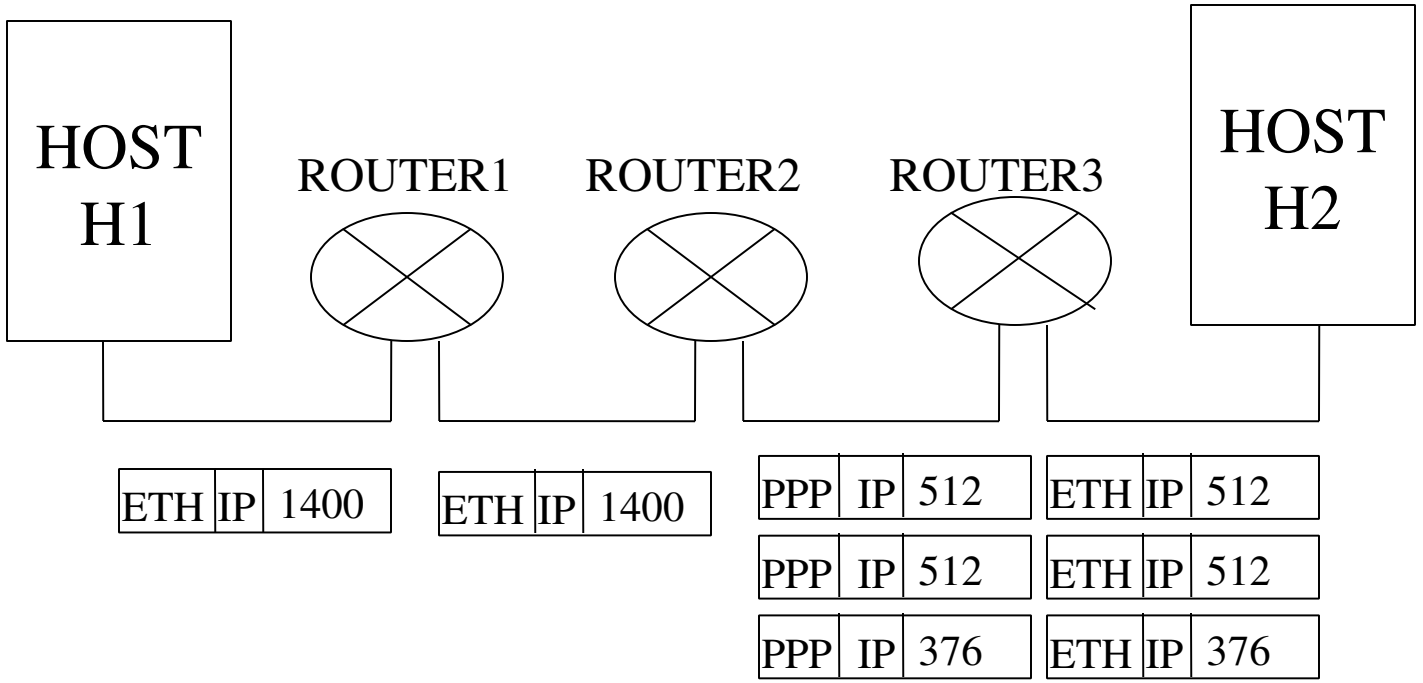
Approcci per la gestione di links eterogenei

- definire la dimensione massima di un pacchetto IP come la minima dimensione dei pacchetti supportata alivello link dalle rete fisiche

  svantaggi:
  - difficoltà relativa alla definizione a priori di un limite inferiore alla dimensione dei pacchetti a livello link (specie per reti in espansione come Internet)
  - spreco di risorse

- *fragmentazione:* dividere un datagram IP in una o più parti (*fragmenti*) nel caso in cui l'MTU di una rete fisica sia inferiore alla dimensione del datagram.

FRAMMENTAZIONE E RICOSTRUZIONE DI PACCHETTI IP



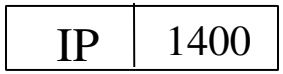
FRAMMENTAZIONE E RICOSTRUZIONE DI PACCHETTI IP

- la frammentazione del pacchetto viene effettuata *dai routers*

- la ricostruzione del pacchetto avviene *nell'host finale* (un pacchetto una volta frammentato viene ricostruito solamente quando arriva a destinazione)

- *ricostruzione* del pacchetto frammentato:
  - utilizza i campi *identificatore*, *flag*, *offset* contenuti nel pacchetto IP
  - *identificatore:* identifica in modo univoco i pacchetti generati da un host. Quando un router fragmenta un pacchetto assegna a tutti i segmenti generati l'identificatore del pacchetto da cui provengono
  - *offset*: identifica la posizione del fragmento all'interno del datagram originale
  - Flag M (more): indica se questo è l'ultimo fragmento del pacchetto IP

FRAMMENTAZIONE E RICOSTRUZIONE DI PACCHETTI IP



Pacchetto IP, identificatore 233

Fragmento 1, identificatore 233, offset 0, M=1

Fragmento 2, identificatore 233, offset 512, M=1

Fragmento 3, identificatore 233, offset 1024, M=0

INDIRIZZAMENTO IP

Schemi di indirizzamento IP

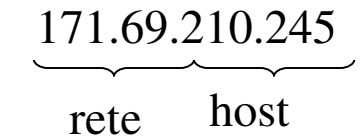
- Classful Addressing
- Subnetting
- Classless Addressing *CIDR*

- *Classfull Addressing:*

  ogni indirizzo IP rappresenta una gerarchia a due livelli

  la prima parte dell'indirizzo *identifica la rete fisica* a cui appartiene l'host individuato da quell'indirizzzo

  la seconda parte individua *l'host*



CLASSFULL ADDRESSING

Classfull Addressing

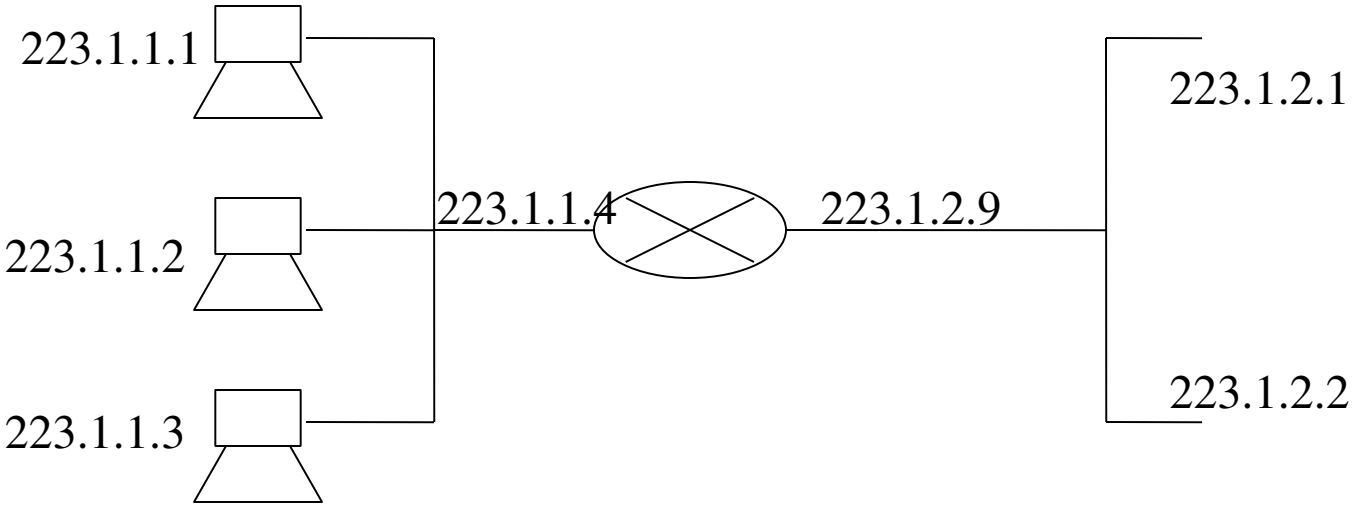
Classe A

Classe B

Classe C

CLASSFULL ADDRESSING

- Tutti gli hosts ed i routers che condividono lo stesso indirizzo di rete sono connessi alla stessa rete fisica



- ogni rete fisica connessa ad Internet ha almeno un router che è connesso ad un'altra rete fisica
- un router possiede un insieme di *interfacce di rete*

INOLTRO DI PACCHETTI

- *inoltro* = meccanismo utilizzato da routers ed hosts per decidere su quale porta di uscita inviare un pacchetto ricevuto su una porta di ingresso

- il meccanismo di inoltro sfrutta un insieme di *tabelle di inoltro* che contengono un insieme di coppie

  (numero di rete, interfaccia di uscita, prossimo router)

- *indirizzamento gerarchico* consente di
  - nelle tabelle dei routers vengono memorizzati solo gli indirizzi delle reti, piuttosto che gli indirizzi dei singoli hosts della rete.
  - Il router associato alla rete destinataria invia il messaggio all'host selezionato.

- *indirizzamento gerarchico*= aumento della *scalabilità del sistema*

INTERNETWORKS (INTERNETS)

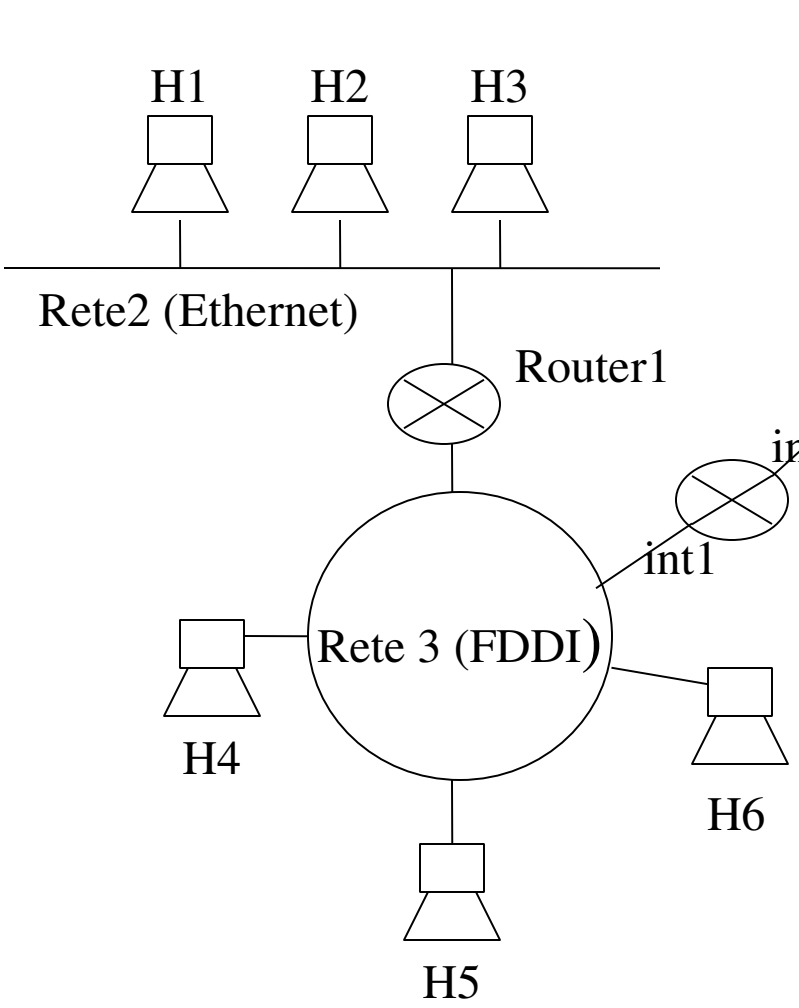


Tabella di inoltro del router2

N. di rete	*Interfaccia di uscita*	*Next router*
1	0	Router3
2	1	Router1

INOLTRO DEI PACCHETTI

Algoritmo di inoltro eseguito dal nodo N (host o router):

NetworkAdd = indirizzo di rete contenuto nel pacchetto da inoltrare

- *Networkadd è uguale all' indirizzo di rete di una delle interfacce di N*

  $\Rightarrow$ il destinatario si trova sulla stessa rete del mittente, il pacchetto può essere consegnato direttamente dal livello link

- *Networkadd è contenuto nella tabella di inoltro di N*

  $\Rightarrow$ il pacchetto va consegnato al router *next router*

- *altrimenti*

  il pacchetto va consegnato ad un *default router*

  (ad esempio un host può avere un default router a cui consegnare tutti i pacchetti non destinati alla rete loacle su cui tale host è connesso

INOLTRO DEI PACCHETTI

- la trasmissione di un pacchetto IP su una rete fisica avviene mediante il livello link
- ogni nodo connesso ad una rete fisica possiede un indirizzo detto indirizzo fisico o *MAC (media acces control)*
- quando un pacchetto IP viene spedito su una rete fisica va utilizzato il MAC del nodo destinazione
- traduzione indirizzo IP–MAC address avviene mediante
  ARP (Address Resolution Protocol)
- Pacchetto IP incapsulato in un frame a livello link che contiene il MAC address del prossimo router o dell'host destinatario

INSTRADAMENTO (ROUTING)

- *Algoritmi di routing*: permettono la costruzione delle tabelle di inoltro

- Dominio di instradamento (o *sistemi autonomi di instradamento*): internetwork in cui tutti i routers sono gestiti dalla stessa entità amministartiva (es: la rete di una università o la rete di una grossa azienda)

- Classificazione:

  - *algoritmi intradominio (interior gateway protocols)*: utilizzati all'interno di un unico *dominio di instradamento,*

  - *algorimi interdominio*

- Domini di instradamento diversi possono utilizzare algoritmi di routing diversi

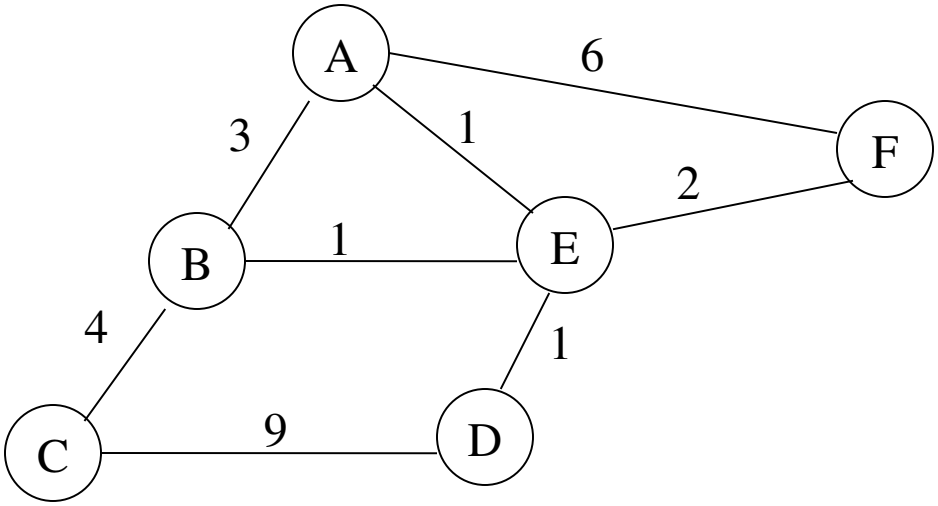
ALGORITMI DI ROUTING INTRA DOMINIO

- *assunzione base*: la rete è rappresentata come *un grafo, in cui i nodi rappresentano routers,* gli *archi* linee fisiche di collegamento

- Ad ogni arco è associato un valore che indica il *costo* dell'invio di un pacchetto su quel link

- Assegnazione dei costi ai link
  - *costi unitari* $\Rightarrow$ il percorso di costo minimo è quello che attraversa il minor numero di routers
  - Traffico sulla linea
  - costi assegnati in base *alla latenza* delle linee
  - costi assegnati in base alla *banda* delle linee

ALGORITMI DI ROUTING INTRA DOMINIO



Problema: dato un nodo mittente N1 ed un nodo destinatario N1, trovare il cammino di costo minimo tra N1 ed N2
(costo di un cammino= somma dei costi associati agli archi che definiscono il cammino)

ALGORITMI DI ROUTING INTRA DOMINIO

Algoritmi di routing intradominio: tutti gli algoritmi vengono eseguiti in modo

Distribuito = ogni nodo esegue lo stesso algoritmo, i nodi cooperano scambiandosi

informazioni sulla topologia della rete e sui costi associai ai links

- *Algoritmi di routing globale*: ogni nodo deve avere *conoscenza* del grafo dell'intera rete e dei costi associati ai links.

  Algoritmi basati sullo Stato delle Linee (OSPF)

- *Algoritmi di routing decentralizzati*: ogni nodo ha solo conoscenza dei costi dei costi relativi ai links che lo collegano ai nodi vicini

  Algoritmi basatisu vettori distanza (RIP)

STATO DELLE LINEE (OSPF)

- ogni router conosce lo stato delle linee (costi) che lo collegano ai nodi vicini
- ogni router invia in *broadcast* a tutti gli altri router le informazioni relative ai links ad esso connessi
- quando un router ha ricevuto lo stato dell'intera rete, crea una mappa completa della topologia della rete
- Ogni router applica alla mappa della rete *l'algoritmo di Dijkstra* per il calcolo dei *cammini minimi* (noto algoritmo di teoria dei grafi)

L'ALGORITMO DI DIJKSTRA

- A =nodo sorgente
- l(i,j) = costo associato al link esistente tra i e j, se i nodi i e j non sono connessi dierettamente l(i,j) =∞
- D(v) costo del cammino minimo dal nodo A a v, ad una certa iterazione dell'algoritmo
- N insieme di nodi per cui il cammino minimo e' stato individuato definitivamente

Fase di inizializzazione:

N={A};

per ogni nodo v

se v è *adiacente ad* A

allora D(v)=L(A,v)

altrimenti D(v)= ∞

L'ALGORITMO DI DIJKSTRA

- A = nodo sorgente
- l(i,j) = costo associato al link esistente tra i e j, se i nodi i e j non sono connessi dierettamente l(i,j) =∞
- D(v) costo del cammino minimo dal nodo A a v, ad una certa iterazione dell'algoritmo
- N insieme di nodi per cui il cammino minimo è stato individuato definitivamente

repeat

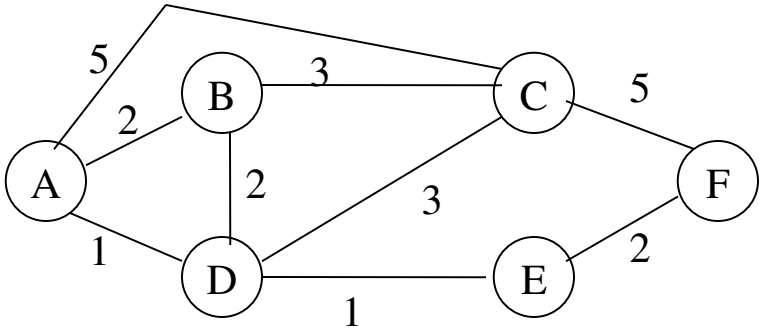
N = N ∪ {w} tale che D(w) è minimo tra tutti i w ∉ N

per ogni v ∉ N

D(v)=min{ D(v), D(w) +l(w,v)}

finchè N contiene tutti i nodi del grafo

L'ALGORITMO DI DIJKSTRA



passo	N	D(B)	D(C)	D(D)	D(E)	D(F)
0	A	2	5	1	∞	∞
1	A,D	2	4	1	2	∞
2	A,D,E	2	4	1	2	4
3	A,D,E,B	2	4	1	2	4
4	A,D,E,B,C	2	4	1	2	4
5	A,D,E,B,C,F	2	4	1	2	4

OSPF: OPEN SHORTEST PATH FIRST

- basato sull'algoritmo di Dijkstra
- utilizza *reliable flooding* (inondazione affidabile) :ogni nodo conosce lo stato delle linee che lo collegano ai nodi vicini
- *reliable flooding*: tutti i nodi che partecipano al protocollo ricevono da tutti gli altri nodi una copia delle informazioni relative allo stato delle linee.
- ogni nodo riceve dai propri vicini le informazioni e le invia a sua volta ai propri vicini (inondazione)

VETTORI DISTANZA(RIP)

- *Vettori distanza:* ciascun nodo costruisce un vettore monodimensionale che contiene le distanze che lo separano dagli altri nodi della rete

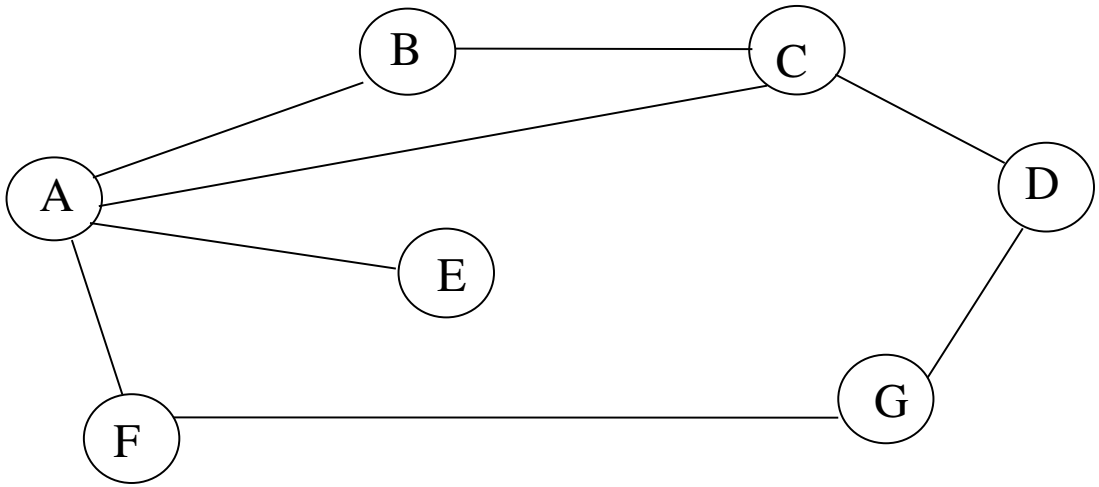
- ciascun nodo inizialmente conosce *solamente il costo delle linee* che lo collegano ai suoi vicini immediati

- inizialmente sono significative solo le entrate relative ai nodi vicini

- ogni nodo invia il proprio vettore di distanze ai nodi vicini

- Quando un nodo riceve un vettore di distanze da un vicino, aggiorna il proprio vettore in base alle informazioni ricevute

VETTORI DISTANZA (RIP)



A	B	C	D	E	F	G
0	1	1	inf	1	1	inf

Vettore distanza iniziale del nodo A

VETTORI DISTANZA (RIP)