

Reti di calcolatori

Livello Rete

Prof.ssa Simonetta Balsamo
Dipartimento di Informatica
Università Ca' Foscari di Venezia
balsamo@dsi.unive.it
<http://www.dsi.unive.it/~reti>

Livello rete

S. Balsamo 2010

R8.1

Livello Network - scopo

- Spostamento dei pacchetti dal mittente al destinatario, attraverso subnet intermedi (**router**)
- Occorre:
 - Conoscere la topologia della rete
 - Scegliere di volta in volta il cammino migliore (**routing**)
 - Gestire il flusso dei dati e le congestioni (**flow control** e **congestion control**)
 - Gestire le problematiche derivanti dalla presenza di più reti diverse (**internetworking**)
- Nella realizzazione del livello network si devono prendere decisioni in merito a:
 - **Servizi offerti** al livello transport
 - **Organizzazione interna** della subnet di comunicazione

Livello rete

S. Balsamo 2010

R8.2

Livello Network - servizi

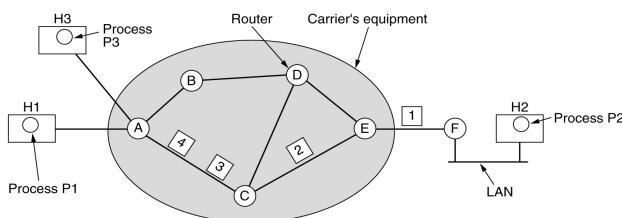
- **Orientati alla connessione:** le peer *entity* stabiliscono una connessione, negoziandone i parametri (di qualità, di costo), alla quale viene associato un identificatore ID
 - Tale ID viene inserito in ogni pacchetto che verrà inviato
 - Comunicazione è **bidirezionale**, i pacchetti viaggiano, lungo il cammino assegnato alla connessione
 - Il **controllo di flusso** è fornito attraverso alcuni dei parametri negoziati
- **Senza connessione:** la sottorete è giudicata inherentemente inaffidabile, gli host devono provvedere per conto proprio alla correzione degli errori e al controllo di flusso
 - Il servizio offerto dal livello network dev'essere datagram
 - I pacchetti viaggiano indipendentemente, devono tutti contenere un identificatore (l'indirizzo) della destinazione

Livello rete

S. Balsamo 2010

R8.3

Livello Network servizi orientati alla connessione



| A's table | | C's table | | E's table | |
|-----------|-------|-----------|-------|-----------|-------|
| In | Out | In | Out | In | Out |
| H1 : 1 | C : 1 | A : 1 | E : 1 | C : 1 | F : 1 |
| H3 : 1 | C : 2 | A : 2 | E : 2 | C : 2 | F : 2 |

| C's table | | E's table | |
|-----------|-------|-----------|-------|
| In | Out | In | Out |
| A : 1 | E : 1 | C : 1 | F : 1 |
| A : 2 | E : 2 | C : 2 | F : 2 |

| E's table | |
|-----------|-------|
| In | Out |
| C : 1 | F : 1 |
| C : 2 | F : 2 |

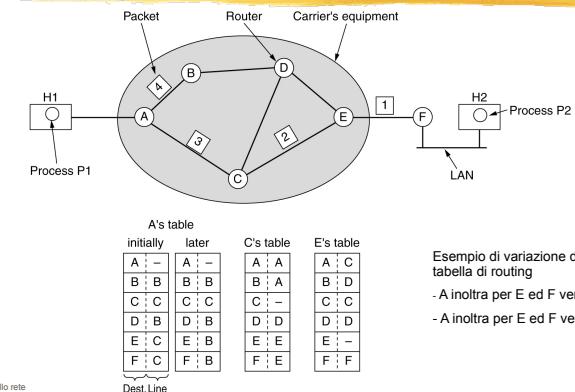
Esempio di due connessioni
1 - H1 - H2
2 - H3 - H2

Livello rete

S. Balsamo 2010

R8.4

Livello Network - servizi senza connessione



Livello rete

Dest. Line

R8.5

Livello Network - servizi

| | Servizi orientati alla connessione | Servizi senza connessione |
|-------------------------------|------------------------------------|------------------------------|
| Indirizzo di destinazione | Presente (durante l'attivazione) | Presente (in ogni pacchetto) |
| Ordine dei pacchetti | Garantito | Non garantito |
| Attivazione della connessione | Presente | Assente |
| Negoziazione di QoS | Presente | Assente |
| Identificatori di connessione | Presente | Assente |
| Controllo del flusso | Presente | Assente |

Livello rete

S. Balsamo 2010

R8.6

Livello Network - subnet

Orientata alla connessione:

- I router stabiliscono un **circuito virtuale**
- Tutti i router lungo tale cammino ricordano, in una struttura dati, la linea in entrata e quale in uscita assegnate al cammino
- Quando arrivano pacchetti che contengono l'ID di tale circuito virtuale, vengono intradati di conseguenza (tutti nello stesso modo)

Connectionless:

- I router intradano ogni pacchetto decidendo ogni volta come proseguire
- I router hanno delle tabelle di instradamento (routing table) che indicano per ogni possibile destinazione, la linea out da utilizzare
- Se offre un servizio con connessione, fa credere al livello superiore che esista una connessione, ma i pacchetti viaggiano indipendentemente e riordinati dal livello 3 solo a destinazione, prima di essere consegnati al livello 4

Livello rete

S. Balsamo 2010

R8.7

Livello Network - subnet e servizi

| | Subnet basata su connessioni | Subnet connectionless |
|------------------------|---|--|
| Banda trasmissiva | Minore (piccole ID in ogni pacchetto) | Maggiore (intero indirizzo di dest. in ogni pacchetto) |
| Spazio sui router | Maggiore (strutture dati per i circuiti virtuali) | Minore |
| Ritardo per il setup | Presente | Assente |
| Ritardo per il routing | Assente | Presente |
| Congestione | Minore (risorse allocate in anticipo) | Maggiore (possibile in ogni momento) |
| Vulnerabilità | Alta | Bassa |

- I servizi offerti sono indipendenti dalla realizzazione interna della subnet:

- Servizi **connection oriented** su circuiti virtuali
- Servizi **connectionless** su subnet datagram
- Servizi **connection oriented** su subnet datagram
- Servizi **connectionless** su circuito virtuale (IP su subnet ATM)

R8.8

Livello Network algoritmi di routing

- La funzione principale del livello network è di **intradare** i pacchetti sulla subnet, facendo fare loro molti **hop** (salti) da un router ad un altro
- Un **algoritmo di routing** è una parte del software che decide su quale linea di uscita intradare un pacchetto che è arrivato:
 - in una subnet **datagram** l'algoritmo viene applicato ex novo ad ogni pacchetto
 - in una subnet basata su **circuiti virtuali** l'algoritmo viene applicato solo nella fase di setup del circuito (**session routing**)

Livello rete

S. Balsamo 2010

R8.9

Livello Network algoritmi di routing - requisiti

- Requisiti di un algoritmo di routing:
 - Correttezza
 - Semplicità
 - Robustezza (deve funzionare anche in caso di cadute di linee e/o router e di riconfigurazioni della topologia)
 - Stabilità
 - Equità
 - Ottimalità

Livello rete

S. Balsamo 2010

R8.10

Livello Network algoritmi di routing - criteri

- Definizione di criteri, ottimalità delle prestazioni:
 - Minimizzare il **ritardo** medio pacchetti
 - Massimizzare il **throughput** totale dei pacchetti
- Gli algoritmi di routing si dividono in:
 - **Algoritmi non adattivi (static routing)**: le decisioni di routing sono prese in anticipo, all'avvio della rete, e sono comunicate ai router che poi si attengono sempre a quelle
 - **Algoritmi adattivi (dynamic routing)**: le decisioni di routing sono riformulate molto spesso in base allo stato della rete
 - sulla base del traffico
 - della topologia della rete
 - altre informazioni

Livello rete

S. Balsamo 2010

R8.11

Livello Network algoritmi di routing - classificazione

- Gli algoritmi adattivi differiscono fra loro per:
 - **Come ricevono le informazioni**
 - Localmente
 - Dai router adiacenti
 - Da tutti i router
 - **Quanto spesso rivedono le decisioni**
 - A intervalli di tempo prefissati
 - Quando il carico cambia
 - Quando la topologia cambia
 - **Quale metrica di valutazione adottano**
 - Distanza
 - Numero di hop
 - Tempo di transito stimato

Livello rete

S. Balsamo 2010

R8.12

Livello Network algoritmi di routing - classificazione

Algoritmi (o politiche)

isolate

- routing calcolato con sole informazioni locali, indipendentemente dal resto (stato degli altri nodi e stato della rete)

es. SNA/IBM

distribuite

- i nodi cooperano e comunicano frequentemente il proprio stato e quello della rete

es. Internet

centralizzate

- un centro di controllo conosce lo stato globale e calcola il cammino ottimo per ogni coppia (mittente,destinatario) e dirama le tabelle

es. TYMNET

miste

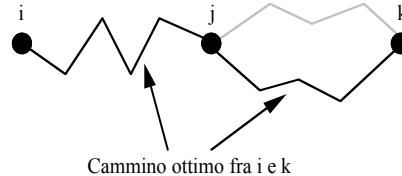
- combinazione di politiche isolate e centralizzate

es. TRANSPAC

Es. isolata: algoritmo *hot potato*

Algoritmi di routing: principio di ottimalità

- Se il router j è nel cammino ottimo fra i e k, allora anche il cammino ottimo fra j e k è sulla stessa strada:



- L'insieme dei cammini ottimi da tutti i router a uno specifico router di destinazione costituiscono un **albero**, detto **sink tree** per quel router

Livello rete

S. Balsamo 2010

R8.14

Algoritmi di routing statici

- Algoritmi che definiscono le regole di instradamento a priori e senza informazione di stato

- shortest path routing
- flooding
- flow-based routing

Livello rete

S. Balsamo 2010

R8.15

Algoritmi di routing statici shortest path routing

1/3

- Un host di gestione della rete mantiene un **grafo** che rappresenta la subnet:

- I nodi rappresentano i router
- Gli archi rappresentano le linee punto-punto

- All'avvio della rete (o quando ci sono variazioni permanenti della topologia) l'algoritmo:

- Applica al grafo un algoritmo per il calcolo del **cammino minimo** fra ogni coppia di nodi; ad esempio, il noto algoritmo di Dijkstra ('59) può essere usato
- Invia tali informazioni a tutti i router

Livello rete

S. Balsamo 2010

R8.16

Algoritmi di routing statici shortest path routing

2/3

Misure da minimizzare, tipicamente:

- Numero di hop, cioè di archi, da attraversare
- Lunghezza dei collegamenti
- Tempo medio di accodamento e trasmissione
- Una combinazione di lunghezza, banda trasmissiva, traffico medio, o altro

Livello rete

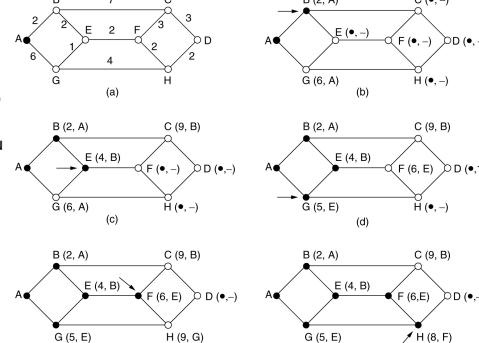
S. Balsamo 2010

R8.17

Algoritmi di routing statici shortest path routing

3/3

Calcolo dei cammino minimo da A a D.
In nero il nodo su cui si opera.



Livello rete

Algoritmo dei cammini minimi

```
#define MAX_NODES 1024           /* maximum number of nodes */
#define INFINITY 1000000000        /* a number larger than every maximum path */
int n, dist[MAX_NODES][MAX_NODES];/* dist[i][j] is the distance from i to j */
void shortest_path(int s, int t, int path[])
{
    struct state {
        /* the path being worked on */
        int predecessor;          /* previous node */
        int length;                /* length from source to this node */
        enum {permanent, tentative} label; /* label state */
    } state[MAX_NODES];
    int i, k, min;
    struct state *p;
    for (p = &state[0]; p < &state[n]; p++) { /* initialize state */
        p->predecessor = -1;
        p->length = INFINITY;
        p->label = tentative;
    }
    state[t].length = 0; state[t].label = permanent;
    k = t;                                /* k is the initial working node */
}
```

Livello rete

S. Balsamo 2010

R8.19

Algoritmo dei cammini minimi

```
do {
    for (i = 0; i < n; i++)
        /* Is there a better path from k? */
        /* this graph has n nodes */
        if (dist[k][i] != 0 && state[i].label == tentative) {
            if (state[k].length + dist[k][i] < state[i].length) {
                state[i].predecessor = k;
                state[i].length = state[k].length + dist[k][i];
            }
        }
    /* Find the tentatively labeled node with the smallest label. */
    k = 0; min = INFINITY;
    for (i = 0; i < n; i++)
        if (state[i].label == tentative && state[i].length < min) {
            min = state[i].length;
            k = i;
        }
    state[k].label = permanent;
} while (k != s);

/* Copy the path into the output array. */
i = 0; k = s;
do {path[i++] = k; k = state[k].predecessor;} while (k >= 0);
}
```

Livello rete

S. Balsamo 2010

R8.20

Algoritmi di routing statici flooding

1/2

- Invio a tutte le linee eccetto a quella da cui il pacchetto è arrivato tecniche per limitare il traffico generato:
 - Inserire in ogni pacchetto un **contatore** che viene decrementato ad ogni hop. Se il contatore arriva a zero, il pacchetto viene scartato
 - Inserire la coppia (**source router ID, sequence number**) in ogni pacchetto. Ogni router esamina tali informazioni e ne tiene traccia, e se le vede per la seconda volta scarta il pacchetto
 - **Selective flooding:** i pacchetti vengono duplicati solo sulle linee che vanno all'incirca nella giusta direzione (tabelle)

Livello rete

S. Balsamo 2010

R8.21

Algoritmi di routing statici flooding

2/2

- Il flooding non è utilizzabile in generale come algoritmo di routing, però:
 - E' utile in campo militare (offre la massima affidabilità e robustezza)
 - E' utile per l'aggiornamento contemporaneo di informazioni distribuite
 - E' utile come strumento di paragone per altri algoritmi, visto che trova sempre, fra gli altri, il cammino minimo

Livello rete

S. Balsamo 2010

R8.22

Algoritmi di routing statici flow-based routing

1/3

- E' basato sull'idea di
 - **Calcolare in anticipo** il traffico atteso su ogni linea
 - Da questi calcoli derivare una **stima del ritardo medio atteso** per ciascuna linea
 - Basare su tali informazioni le decisioni di routing
- Le informazioni necessarie per poter applicarlo sono:
 - La topologia della rete
 - La matrice delle quantità di traffico $T(i,j)$ stimate fra ogni coppia (i,j) di router
 - Le capacità (in bps) delle linee point to point

Livello rete

S. Balsamo 2010

R8.23

Algoritmi di routing statici flow-based routing

2/3

- Vengono fatte le assunzioni
 - Il traffico è **stabile** nel tempo e **noto in anticipo**
 - **ritardo \propto traffico** Il ritardo su ciascuna linea aumenta all'aumentare del traffico sulla linea e diminuisce all'aumentare della velocità della linea secondo le leggi della teoria delle code
- **Ritardo medio della rete:** somma pesata dei ritardi delle singole linee
- **peso della linea \propto traffico**
peso della linea = traffico sulla linea / traffico totale sulla rete

Livello rete

S. Balsamo 2010

R8.24

Algoritmi di routing statici flow-based routing

3/3

Algoritmo di routing basato sul flusso

- Si sceglie un algoritmo di routing
- Sulla base di tale algoritmo si determinano i percorsi che verranno seguiti per il collegamento fra ogni coppia di router
- Si calcola il traffico che incide su ogni linea (uguale alla somma di tutti i $T(i,j)$ instradati su quella linea)
- Si calcola il ritardo di ogni linea
- Si calcola il ritardo medio della rete
- Si ripete il procedimento con vari algoritmi di routing, scegliendo alla fine quello che minimizza il ritardo medio dell'intera rete

Livello rete

S. Balsamo 2010

R8.25

Algoritmi di routing dinamici

- Algoritmi che definiscono le regole di instradamento dinamicamente sulla base di informazione di stato

- **distance vector routing**
- **link state routing**

Livello rete

S. Balsamo 2010

R8.26

Algoritmi di routing dinamici distance vector routing

1/4

- In ogni router mantiene una tabella (**vector**) con, per ogni router:
 - La distanza (numero di hop, ritardo, ecc.) che lo separa dal router in oggetto
 - La linea in uscita da usare per arrivarci
- Per i suoi vicini immediati il router stima direttamente la distanza dei collegamenti corrispondenti, mandando speciali **pacchetti ECHO** e misurando quanto tempo ci mette la risposta a tornare
- A intervalli regolari ogni router manda la sua tabella a tutti i vicini, e riceve quelle dei vicini

Livello rete

S. Balsamo 2010

R8.27

Algoritmi di routing dinamici distance vector routing

2/4

- Quando un router riceve le nuove informazioni, calcola una **nuova tabella** scegliendo, fra tutte, la concatenazione migliore fra
 - **distanza** fra il router stesso ed un suo **vicino immediato** (viene dalla misurazione diretta)
 - **distanza** fra quel vicino immediato ed il **router remoto** di destinazione (viene dalla tabella ricevuta dal vicino immediato)
- Algoritmo buono ma molto lento nell'adattamento (topologia non nota)
- Algoritmo di routing di ARPANET
- usato anche in Internet col nome di **RIP** (**Routing Internet Protocol**)
- nelle prime versioni di DECnet e IPX

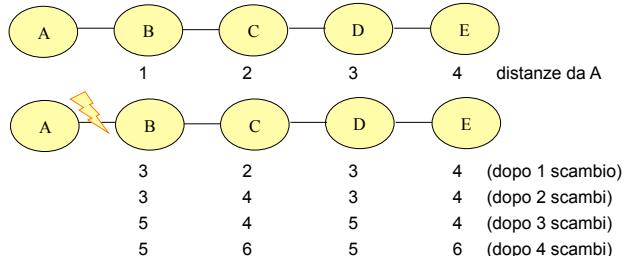
Livello rete

S. Balsamo 2010

R8.28

Algoritmi di routing dinamici distance vector routing 3/4

Esempio



- A lungo andare, tutti i router vedono lentamente aumentare sempre più la distanza per arrivare ad A.
- Problema del [count-to-infinity](#)

Algoritmi di routing dinamici distance vector routing 4/4

count-to-infinity

- Se distanza = # hop, limite diametro della rete
- se distanza = ritardo, l'upper bound molto alto per non interrompere cammini
- Problema non risolto efficacemente

Livello rete

S. Balsamo 2010

R8.30

Algoritmi di routing dinamici link state routing 1/5

- Ogni router mantiene informazioni sullo stato dei collegamenti con i vicini immediati (misurando il ritardo di ogni linea) e distribuisce tali informazioni a tutti
- Sulla base di tali informazioni, ogni router ricostruisce localmente la topologia completa della rete e calcola il cammino minimo fra se e tutti gli altri

Livello rete

S. Balsamo 2010

R8.31

Algoritmi di routing dinamici link state routing 2/5

I passi da seguire sono:

- Scoprire i vicini e identificarli
- Misurare il costo (ritardo o altro) delle relative linee
- Costruire un pacchetto con tali informazioni
- Mandare il pacchetto a tutti gli altri router
- Previa ricezione degli analoghi pacchetti che arrivano dagli altri router, costruire la topologia dell'intera rete
- Calcolare il cammino più breve a tutti gli altri router

- Invia un [pacchetto HELLO](#) su tutte le linee in uscita. In risposta riceve dai vicini i loro indirizzi (univoci in tutta la rete)

Livello rete

S. Balsamo 2010

R8.32

Algoritmi di routing dinamici link state routing

3/5

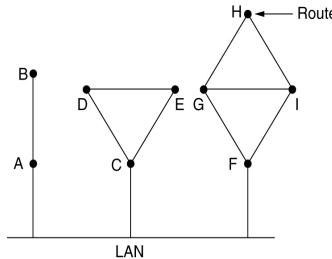
- Si costruisce un pacchetto con:
 - Identità del mittente
 - Numero di sequenza del pacchetto
 - Età del pacchetto
 - Lista dei vicini con i relativi ritardi
- La costruzione e l'invio di tali pacchetti si verifica tipicamente a intervalli regolari o quando accade un evento significativo
- Il link state routing è molto usato attualmente:
 - In Internet OSPF (Open Shortest Path First) è basato su tale principio e si avvia ad essere l'algoritmo più utilizzato
 - IS-IS (Intermediate System-Intermediate System), progettato per DECnet. Gestisce indirizzi di diverse architetture (OSI, IP, IPX) per cui può essere usato in reti miste o multiprotocollo

Livello rete

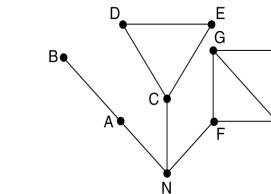
R8.33

Algoritmo link state routing esempio

4/5



Nove routers e una LAN

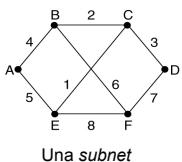


Un modello a grafo della rete

R8.34

Algoritmo link state routing esempio

5/5



Una subnet

| | Link | State | Packets |
|---|------|-------|---------|
| A | Seq. | Seq. | E |
| B | Seq. | Age | Seq. |
| C | Age | Age | Age |
| D | Age | Age | Age |
| E | Age | Age | Age |
| F | Age | Age | Age |

I pacchetti link state per la subnet

| Source | Seq. | Age | Send flags | | | ACK flags | | | Data |
|--------|------|-----|------------|---|---|-----------|---|---|------|
| | | | A | C | F | A | C | F | |
| A | 21 | 60 | 0 | 1 | 1 | 1 | 0 | 0 | |
| F | 21 | 60 | 1 | 1 | 0 | 0 | 0 | 1 | |
| E | 21 | 59 | 0 | 1 | 0 | 1 | 0 | 1 | |
| C | 20 | 60 | 1 | 0 | 1 | 0 | 1 | 0 | |
| D | 21 | 59 | 1 | 0 | 0 | 0 | 1 | 1 | |

Il buffer di pacchetti per il router B

Livello rete

S. Balsamo 2010

R8.35

Algoritmi di routing dinamici routing gerarchico

1/4

- La rete viene divisa in zone (dette **regioni**):

- All'interno di una regione ogni router (detti **router interni**) applica un algoritmo di routing verso tutti gli altri router della regione
- Quando un router interno deve spedire un pacchetto a un router di un'altra regione lo invia ad un particolare router della propria regione, detto **router di confine**
- Il router di confine conosce gli altri router di confine e conosce a quale inviare il pacchetto perché arrivi alla regione di destinazione
- **Due livelli** di routing:
 - un primo livello di routing all'interno di ogni regione
 - un secondo livello di routing fra tutti i router di confine

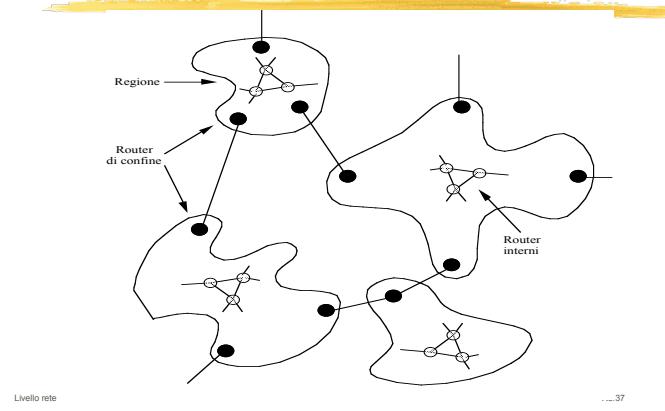
Livello rete

S. Balsamo 2010

R8.36

Algoritmi di routing dinamici routing gerarchico

2/4



... 37

Algoritmi di routing dinamici routing gerarchico

3/4

- I router interni mantengono delle loro tabelle di routing
 - Una riga per ogni altro router interno (**linea**)
 - Una riga per ogni altra regione (**router di confine**)
- I router di confine mantengono
 - Una riga per ogni altra regione, con l'indicazione del prossimo router di confine da contattare e della linea da usare per raggiungerlo

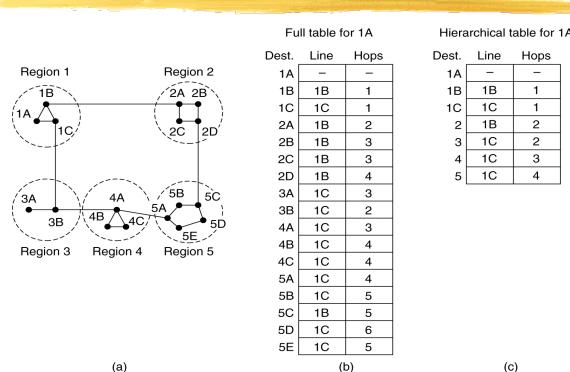
Livello rete

S. Balsamo 2010

R8.38

Algoritmi di routing dinamici routing gerarchico

4/4



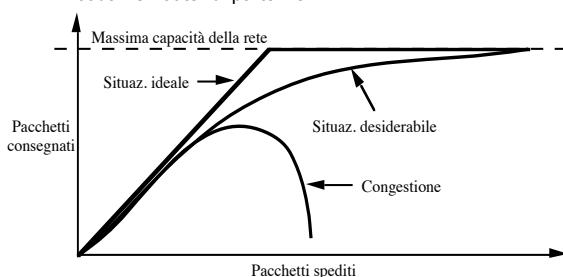
Livello rete

S. Balsamo 2010

R8.39

Livello Network controllo della congestione

- La congestione in un router può derivare da diversi fattori
 - Buffer** a capacità limitata nel router
 - Processore** troppo lento nel router
 - Linea** di trasmissione troppo lenta, che causa congestione della coda nel router di partenza



R8.40

Livello Network controllo della congestione

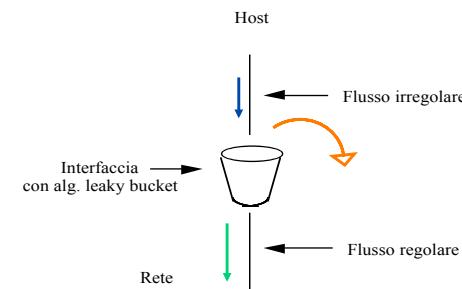
- Il controllo della congestione è un problema **globale** di tutta la rete, ed è diverso dal problema del controllo di flusso
- Due approcci al problema della congestione
 - **Open loop** (senza contro-reazione): cerca di **prevenire** la congestione, ma se si verifica non effettua azioni correttive
 - *Leaky Bucket*
 - *Token Bucket*
 - **Closed loop** (con contro-reazione): cerca di **evitare** la congestione, controllando lo stato della rete e intraprendendo le azioni opportune quando necessario

Livello rete

S. Balsamo 2010

R8.41

Controllo della congestione algoritmo leaky bucket 1/2



Livello rete

S. Balsamo 2010

R8.42

Controllo della congestione algoritmo leaky bucket 2/2

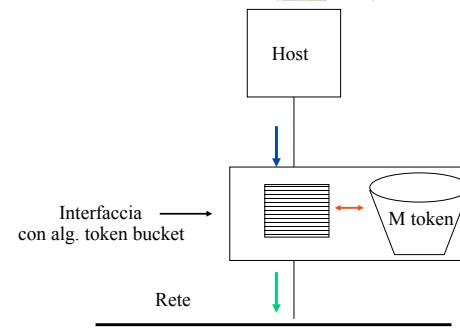
- **Prevenzione** della congestione (open loop)
- Metafora del "secchio bucato"
- Se viene immessa troppa acqua, fuoriesce dal bordo superiore del secchio e si perde
- Sull'host si realizza (nell'interfaccia di rete o in software) un buffer (leaky bucket) che
 - immette i pacchetti sulla rete ad un tasso costante (max data rate) b bps
 - mantiene i pacchetti accodati per la trasmissione
- Se l'host genera più pacchetti di quelli contenibili dal buffer, ovvero se il $data\ rate > b \Rightarrow$ si ha **perdita di pacchetti**
- L'host può anche produrre un traffico **bursty** senza che ciò provochi perdita di pacchetti finché $data\ rate < b$ bps

Livello rete

S. Balsamo 2010

R8.43

Controllo della congestione algoritmo token bucket 1/2



Livello rete

S. Balsamo 2010

R8.44

Controllo della congestione algoritmo token bucket 2/2

- Metafora del "secchio di gettoni" (open loop)
- Consente **irregolarità** controllate nel flusso in uscita sulla rete
 - Quando non trasmette la stazione accumula un **credito trasmisivo** con un certo data rate (fino ad un massimo consentito)
 - Il **secchio** contiene dei **token**, creati con cadenza prefissata fino a quando il loro numero raggiunge un valore massimo M
 - Se si devono trasmettere h pacchetti e il secchio ha k token
 - se $h > k \Rightarrow$ i primi k sono trasmessi subito e gli altri aspettano nuovi token
 - Per trasmettere si sfrutta **tutto** il credito disponibile, se occorre, e si invia alla massima velocità consentita dalla linea
 - I pacchetti non vengono mai scartati (il secchio contiene token, non pacchetti). Se necessario, si avverte il livello superiore (4), per fermare la produzione dei dati
- **Leaky buket e token bucket** possono essere usati per regolare il traffico **host-router e router-router**
- Se il router sorgente deve fermarsi e non ha spazio sufficiente, si può aver perdita

Livello rete

S. Balsamo 2010

R8.45

Controllo della congestione flow specification

- **Evitare** della congestione (**closed loop**)
 - È molto efficace se tutti (sorgente, subnet e destinazione) si accordano in merito **prima** di trasmettere
- Per ottenere tale accordo bisogna **specificare**:
 - Le caratteristiche del **traffico** che si vuole **inviare**
 - data rate
 - grado di burstiness
 - altro
 - La **qualità del servizio**
 - ritardo massimo
 - frazione di pacchetti che si può perdere
 - altro
- **Admission control**:
 - politica di ammissione per attivare nuovi circuiti virtuali,
 - per evitare la congestione si può negare una attivazione se non vi sono sufficienti risorse per gestirli

Livello rete

S. Balsamo 2010

R8.46

Controllo della congestione choke packet e hop by hop 1/3

- Il router controlla il grado di utilizzo delle sue linee di uscita, misurando per ciascuna linea, l'utilizzo istantaneo U
- accumula, entro una media esponenziale M, la storia passata:

$$M_{\text{nuovo}} = a \cdot M_{\text{vecchio}} + (1 - a) U$$
 - dove:
 - a peso dato alla storia passata, $0 \leq a \leq 1$
 - $(1 - a)$ peso dato all'informazione più recente

Livello rete

S. Balsamo 2010

R8.47

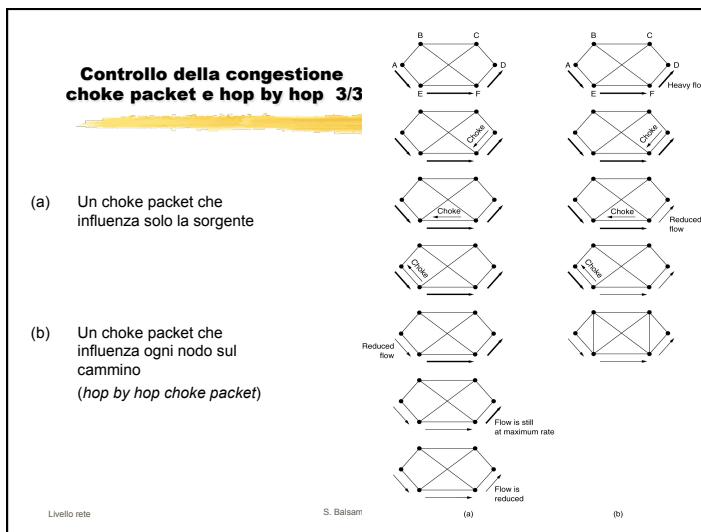
Controllo della congestione choke packet e hop by hop 2/3

- Quando M è vicina ad una soglia di pericolo prefissata il router
 - esamina i pacchetti in ingresso
 - invia all'host di origine del pacchetto un **choke packet** per diminuire il flusso
- Quando l'host sorgente riceve il choke packet
 - diminuisce il flusso (es. lo dimezza)
 - ignora i successivi choke packet per un tempo prefissato
 - successivamente si rimette in attesa di altri choke packet
 - se ne arrivano altri, riduce ancora il flusso, altrimenti lo aumenta di nuovo
- **La reazione è molto lenta**
- **Miglioramento:** ogni router sul percorso, appena riceve i **choke** rallenta subito (**hop by hop choke packet**)
 - E' più veloce, ma richiede più spazio nei router sul percorso.

Livello rete

S. Balsamo 2010

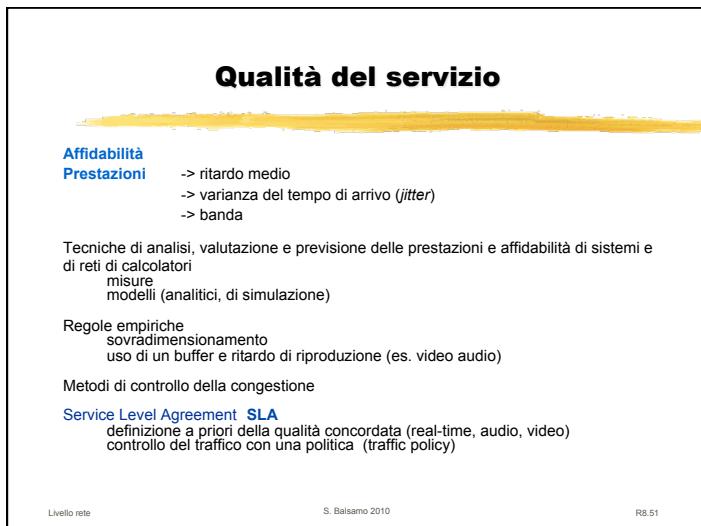
R8.48



Qualità del servizio - requisiti

| Application | Reliability | Delay | Jitter | Bandwidth |
|-------------------|-------------|--------|--------|-----------|
| E-mail | High | Low | Low | Low |
| File transfer | High | Low | Low | Medium |
| Web access | High | Medium | Low | Medium |
| Remote login | High | Medium | Medium | Low |
| Audio on demand | Low | Low | High | Medium |
| Video on demand | Low | Low | High | High |
| Telephony | Low | High | High | Low |
| Videoconferencing | Low | High | High | High |

Livello rete S. Balsamo 2010 R8.50

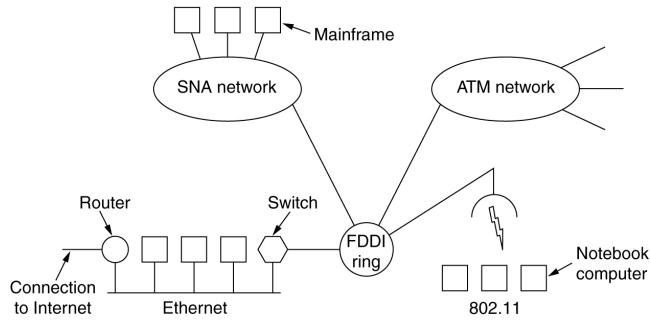


Livello Network - internetworking

- Per connettere fra loro reti **eterogenee** si devono superare problemi non banali, tra i quali:
 - Difformità nei **formati** dei pacchetti e degli indirizzi
 - Difformità, nelle **subnet**, dei **meccanismi di controllo dell'errore e della congestione**
 - **Diverse architetture** di rete
 - **servizi** offerti dai vari livelli
 - modalità di **indirizzamento**
 - **max dimensione dei pacchetti**
- Tecniche
 - **Bridge** - livello *data link*
 - **router multiprotocollo**: gestione contemporanea di più pile di protocolli
 - **tunneling**
 - **frammentazione**
 - **circuiti virtuali concatenati**

Livello rete S. Balsamo 2010 R8.52

Esempio di interconnessione di reti



Esempi di differenze nello strato rete

| Item | Some Possibilities |
|--------------------|--|
| Service offered | Connection oriented versus connectionless |
| Protocols | IP, IPX, SNA, ATM, MPLS, AppleTalk, etc. |
| Addressing | Flat (802) versus hierarchical (IP) |
| Multicasting | Present or absent (also broadcasting) |
| Packet size | Every network has its own maximum |
| Quality of service | Present or absent; many different kinds |
| Error handling | Reliable, ordered, and unordered delivery |
| Flow control | Sliding window, rate control, other, or none |
| Congestion control | Leaky bucket, token bucket, RED, choke packets, etc. |
| Security | Privacy rules, encryption, etc. |
| Parameters | Different timeouts, flow specifications, etc. |
| Accounting | By connect time, by packet, by byte, or not at all |

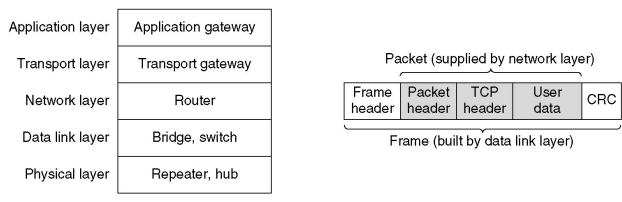
Livello rete

S. Balsamo 2010

R8.54

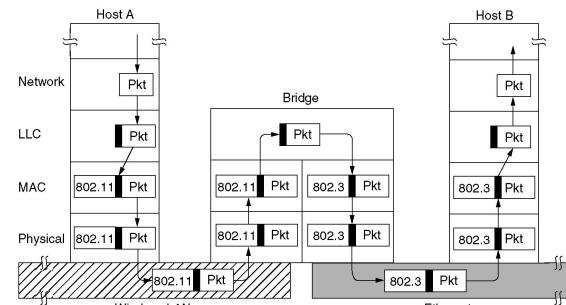
Livello Data Link - internetworking

- Elementi di interconnessione a livello data link: **bridge**
- Controllano l'indirizzo ed eseguono l'instradamento, ma non esaminano il contenuto dei pacchetti
- Confronta l'indirizzo destinazione contenuto nel frame con il contenuto di una tabella (hash) che riporta la corrispondente linea di uscita



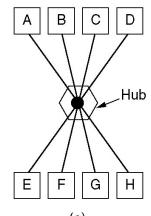
Livello MAC/LLC - internetworking

- Esempio di bridge fra rete wireless e rete Ethernet

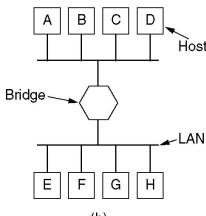


Livello fisico e Data Link - internetworking

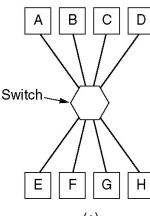
- Collegamenti con **HUB** (a) a livello fisico
- **BRIDGE** (b) e **SWITCH** (c) a livello data link



Livello rete



(b)



R8.57

Livello fisico e Data Link - internetworking

- **HUB** connette un insieme di linee
 - riceve da una linea e trasmette su tutte le altre
 - possibili collisioni (hub è un dominio di collisione)
 - linee alla stessa velocità
 - non amplificano il segnale e non filtrano l'indirizzo
- **BRIDGE** connette un insieme di LAN
 - estrae dal frame l'indirizzo e lo confronta con valori in tabelle per determinare la destinazione
 - può supportare linee a diverse velocità
 - ogni linea è un dominio di collisione
- **SWITCH** connette un insieme di computer
 - Similmente al Bridge estrae dal frame l'indirizzo e lo instrada a destinazione
 - Ha solitamente più schede di linea del bridge, con relativo buffer sulle porte
 - ogni porta è un dominio di collisione

Livello rete

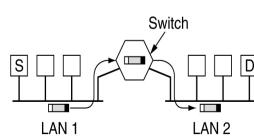
S. Balsamo 2010

R8.58

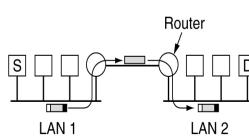
Livello Data Link e rete - internetworking

- Collegamenti di due reti con **SWITCH** (a) a livello data link
- **ROUTER** (b) a livello rete

Legend
 Header
 Packet
 Trailer



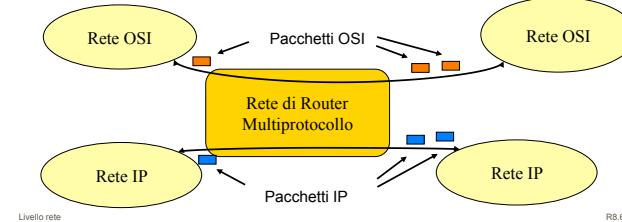
Livello rete



R8.59

Livello Network - internetworking reti di router multiprotocollo

- Ogni rete può comunicare con le altre reti a lei conformi attraverso una porzione di rete costituita di router multiprotocollo (es. reti OSI possono comunicare fra loro, e così quelle IP)
- Nella subnet costituita dai router multiprotocollo circolano pacchetti di entrambe le architetture, che vengono instradati secondo le regole di competenza dell'architettura di cui fanno parte

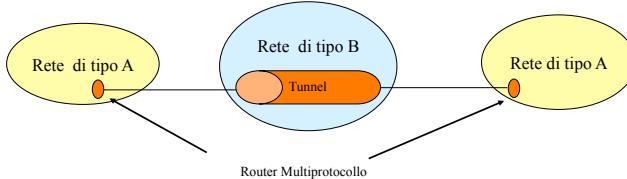


Livello rete

R8.60

Livello Network - internetworking tunelling 1/3

- Si utilizza per mettere in comunicazione due reti uguali attraverso una rete diversa



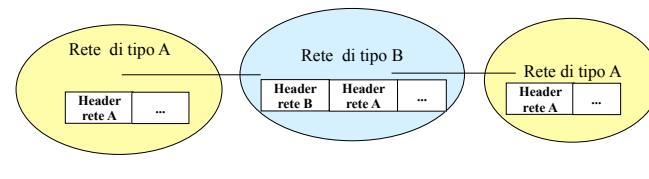
Livello rete

S. Balsamo 2010

R8.61

Livello Network - internetworking tunelling 2/3

- B non ha router multiprotocollo
- un router designato nelle due reti A è multiprotocollo e incapsula i pacchetti delle reti di tipo A dentro pacchetti di tipo B, consegnandoli alla rete di tipo B



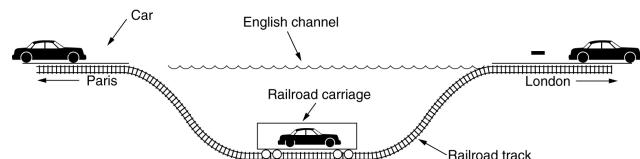
Livello rete

S. Balsamo 2010

R8.62

Livello Network - internetworking tunelling 3/3

- il router designato nelle due reti A è multiprotocollo -> imbarco/sbarco
- incapsula i pacchetti delle reti di tipo A -> auto
- dentro pacchetti di tipo B-> treno
- consegnandoli alla rete di tipo B -> rotaie nel tunnel



Livello rete

S. Balsamo 2010

R8.63

Livello Network - internetworking frammentazione 1/2

- E' necessario che il livello network della rete di origine (e di quella di destinazione) prevedano meccanismi di **spezzettamento** del **pacchetto** in **frammenti** prima di consegnarli alla rete di transito, e di **ricomposizione** dei frammenti appena essi giungono dalla rete di transito in quella di destinazione (es: il protocollo IP è fornito di questa funzionalità)

- Limiti alla dimensione dovuti a

- Hardware
- Protocolli
- Sistema operativo
- Standard
- Limiti agli errori o al tempo (affidabilità e prestazioni)

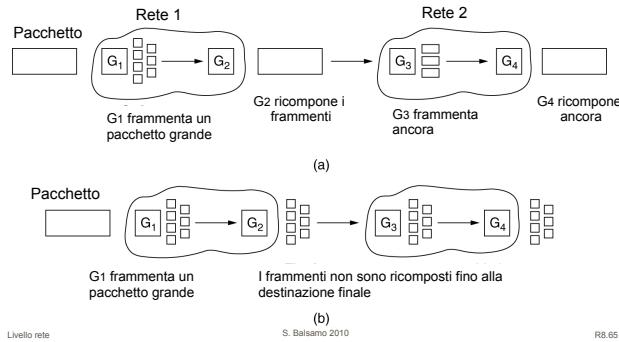
Livello rete

S. Balsamo 2010

R8.64

Livello Network - internetworking frammentazione 2/2

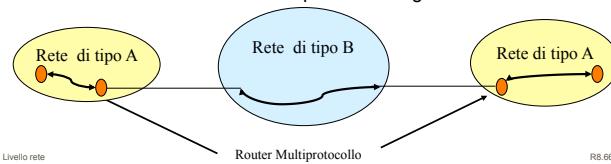
- Frammentazione trasparente (a) e non trasparente (b)



Livello Network - internetworking circuiti virtuali concatenati

Se tutte le reti interessate offrono servizi connessi nel livello network, è possibile costruire un circuito virtuale che si estende attraverso più reti eterogenee come **concatenazione di circuiti virtuali** che attraversano ciascuno una delle reti. Ai confini fra ogni rete ci sono dei router multiprotocollo che:

- Creano la porzione di circuito virtuale che attraversa la rete di competenza, arrivando fino ad un router multiprotocollo situato all'altra estremità della rete
- Instradano successivamente i pacchetti lungo tale circuito virtuale



Livello Network - internetworking transport gateway

- E' piuttosto diffuso a **livello transport** (4)
- Offre **servizi connessi** in quasi tutte le architetture
- Poiché operano a livello transport, essi ricevono i dati dal livello application di partenza e li consegnano al livello application di arrivo
- Possono interfacciare connessioni di trasporto con reti aventi diversi protocolli

Livello rete

S. Balsamo 2010

R8.67

Livello Network - internetworking application gateway

- Effettuano una conversione di dati a **livello application** (5)
- Traducono la semantica dei messaggi connettendo diversi portocolli di livello applicazione
- Per mandare da un host Internet un messaggio di posta elettronica a un utente OSI:
 - Si compone il messaggio secondo il formato Internet; l'indirizzo del destinatario è un indirizzo OSI, ma viene espresso secondo regole valide per Internet
 - Si invia il messaggio, che viene consegnato a un **mail gateway**
 - Il mail gateway estrae il testo dal messaggio, lo inserisce in un nuovo messaggio in formato OSI (X.400) e lo invia sulla rete OSI

Livello rete

S. Balsamo 2010

R8.68

Livello Network - internetwork routing

- In una internetwork le singole reti componenti sono entità autonome e vengono chiamate **AS (Autonomous System)**
- Il routing complessivo è a due livelli:
 - Un primo livello è costituito dall'**Interior Gateway Protocol (IGP)** che identifica l'algoritmo di routing usato da un AS al proprio interno.
 - Diversi AS possono utilizzare diversi IGP
 - Un secondo livello è dato dall'**Exterior Gateway Protocol (EGP)**, che è l'algoritmo che si usa per gestire il routing fra diversi AS
 - Deve tener conto di specifiche leggi nazionali (divieto di far transitare dati sul suolo di una nazione ostile), per cui le decisioni di routing devono adattarsi a tali direttive

Livello rete

S. Balsamo 2010

R8.69

Livello Network in Internet

■ Internet: collezione di AS interconnessi

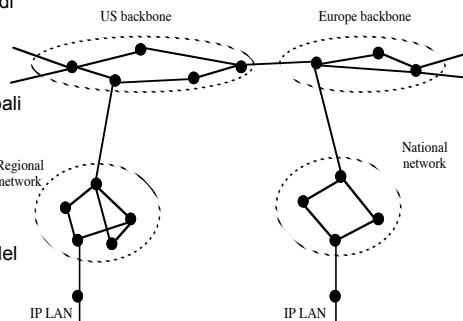
■ struttura flessibile, componenti:

- Backbone principali (linee ad alta velocità)
- Reti regionali (USA)
- Reti nazionali (Europa e resto del mondo)
- Reti locali

Livello rete

S. Balsamo 2010

R8.70



Livello Network - IP

1/12

- IP è un protocollo **datagram**
- Riceve i dati dal livello transport e li **incapsula** in pacchetti di dimensione massima pari a 64 Kbyte
- **Instrada** i pacchetti sulla subnet, eventualmente **frammentandoli** lungo il viaggio
- **A destinazione**:
 - **Riassembra** i frammenti in pacchetti
 - **Estrae** i dati del livello transport
 - **Consegna** al livello transport i dati nell'ordine in cui sono arrivati
- Un pacchetto IP è costituito da un **header** e da una parte dati
- L'header ha una parte fissa di 20 byte e una parte, opzionale, di lunghezza variabile

Livello rete

S. Balsamo 2010

R8.71

Livello Network - IP Principi di progetto 2/12

- Correttezza di funzionamento
- Semplicità
- Chiarezza
- Modularità
- Possibile eterogeneità
- Non avere opzioni e parametri statici
- Buon progetto, non necessariamente omnicomprensivo e perfetto
- Invio rigoroso e ricezione tollerante
- Prevedere la scalabilità
- Considerare le prestazioni e i costi

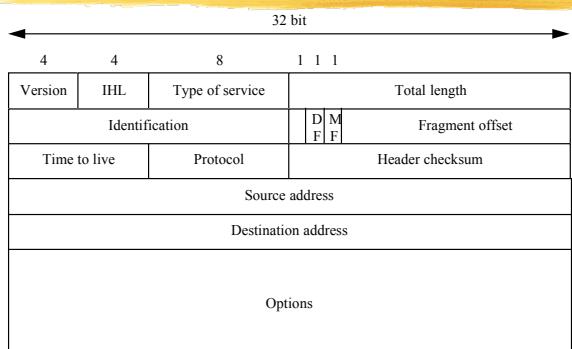
Livello rete

S. Balsamo 2010

R8.72

Livello Network - IP

3/12



Livello rete

S. Balsamo 2010

R8.73

Livello Network - IP

4/12

- **Version** numero di versione del protocollo (4)
- **IHL** lunghezza header in parole di 32 bit [5,15]
- **Type of service** affidabilità e velocità richieste - ignorato dai router;
- **Total length** lunghezza del pacchetto (inclusi dati), massimo 65.535 byte
- **Identification** i frammenti di uno stesso pacchetto hanno lo stesso valore
- **DF don't fragment** =1, non frammentare anche scegliendola via più lenta
- **MF more fragment** =1, il pacchetto non è ancora finito
- **Fragment offset** indice del frammento nel pacchetto
- **Time to live** contatore (inizializzato a 255) decrementato ad ogni hop (o sec) se=0 il pacchetto viene scartato

Livello rete

S. Balsamo 2010

R8.74

Livello Network - IP

5/12

- **Protocol** codice del protocollo di livello transport per consegna dei dati
- **Header checksum** somma le parole dell'header e si complementa ad uno si ricalcola ad ogni hop (time to live cambia)
- **Source e destination address**
 - **Options** oggi ne sono definite solo cinque
 - **Security** quanto è segreto il pacchetto
 - **Strict source routing** cammino da seguire
 - **Loose source routing** lista di router da non mancare
 - **Record route** ogni router appende il suo indirizzo
 - **Time stamp** marca temporale

Livello rete

S. Balsamo 2010

R8.75

Livello Network - IP

6/2

- Un indirizzo IP è formato da 32 bit e codifica due cose:
 - **network number** numero assegnato alla rete IP (detta **network**) su cui si trova l'elaboratore
 - **host number** numero assegnato all'elaboratore
- La combinazione è unica: non possono esistere nell'intera rete Internet due indirizzi IP uguali
- Gli indirizzi sono assegnati alle interfacce di rete:
 - Se un host ha un'unica interfaccia di rete allora ha un unico indirizzo IP
 - Se un host ha X interfacce di rete ha X indirizzi
- Gli indirizzi IP sono assegnati da autorità nazionali (**NIC**, **Network Information Center**) coordinate a livello mondiale

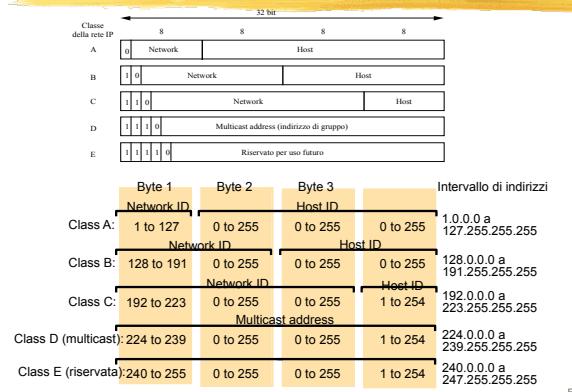
Livello rete

S. Balsamo 2010

R8.76

Livello Network - IP: formati

7/12



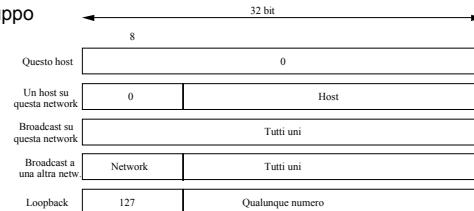
Livello Network - IP

8/12

Classe A: 126 reti di 16 milioni di host
 Classe B: 16.382 reti di ca. 64.000 host
 Classe C: 2 milioni reti di 254 host

Es. indirizzo IP Rete . Host
 157.138.20.17

- **Loopback:** il pacchetto non viene inviato sulla rete ma viene elaborato come se fosse in arrivo
- Serve per effettuare localmente dei test su un software di rete in fase di sviluppo



Livello Network - IP

9/12

- **ICMP (Internet Control Message Protocol):** Controlla l'operatività delle subnet, tipi di messaggi:
 - Destination unreachable
 - Time exceeded - time to live ha raggiunto 0
 - Redirect - routing
 - Echo request, reply
 - Time stamp request, reply
- **ARP (Address Resolution Protocol):** deriva dall'[indirizzo IP](#) dell'host di destinazione, l'[indirizzo di livello data link](#) necessario per inviare il frame
- **RARP (Reverse Address Resolution Protocol):** Trovare quale indirizzo IP corrisponda a un dato indirizzo data link

Livello rete

S. Balsamo 2010

R8.79

Livello Network - IP

10/12

- **Interior Gateway Protocol (IGP):**
 - **RIP (Routing Information Protocol):** Di tipo distance vector
 - **OSPF (Open Shortest Path First):** Di tipo link state o routing gerarchico
- **Exterior Gateway Protocol (EGP)**
 - **BGP (Border Gateway Protocol):** Di tipo distance vector

Livello rete

S. Balsamo 2010

R8.80

Livello Network - IP

11/12

- **Versione 6:** Successore di IP versione 4 ([IPv6](#))
- Principali requisiti di progetto:
 - Aumento del numero di indirizzi
 - Maggiore efficienza nei router (tavole più piccole, routing più veloce)
 - Miglior supporto del traffico **real time**
 - Maggiore **sicurezza** ai dati riservati
- Principali differenze rispetto a IPv4:
 - **Indirizzi di 16 byte:** 2^{128} ($\approx 3 \times 10^{38}$) indirizzi IP, o 7×10^{23} ind. IP per mq
 - **Header semplificato:** 7 campi contro 13
 - Funzioni di **autenticazione** e **privacy**, basate su crittografia
 - Gestione della QoS attraverso un campo **flow label**, per istituire pseudo connessioni negoziate in anticipo

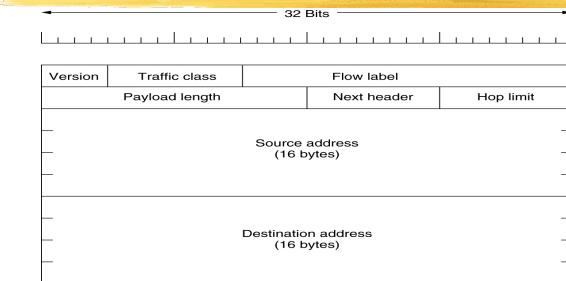
Livello rete

S. Balsamo 2010

R8.81

Intestazione base di IPv6

12/12



Traffic class (8 bit), Flow label (20 bit) nuovi campi

New header (8 bit) tipo del prossimo header

Hop limit (8 bit) corrisponde al Time to live

Livello rete

S. Balsamo 2010

R8.82