

Obiettivi – 1

- Ricevere pacchetti dal livello Trasporto del nodo **M** e farli pervenire allo stesso livello del nodo **D**
- Conoscere la topologia dei *router* presenti nella rete
 - Determinando per quali vie raggiungerli
- Superare ogni eventuale disomogeneità tra gli ambienti di rete di **M** e **D**

Livello rete (IP) - parte 1

Architettura degli Elaboratori 2 - T. Vardanega

Pagina 356

Obiettivi – 2

- Offrire servizi **indipendenti** dalla tecnologia usata per realizzare le *subnet* nascondendone la complessità al livello Trasporto
- Garantire l'**uniformità** del sistema di indirizzamento superando le diverse tipologie delle reti fisiche collegate
 - *Inter-networking*

Livello rete (IP) - parte 1

Architettura degli Elaboratori 2 - T. Vardanega

Pagina 357

Scelte progettuali – 1

- Servizi privi di garanzie di connessione (**connectionless**)
 - La scelta progettuale di **Internet**
 - Essendo le *subnet* **intrinsecamente inaffidabili** (introducono ritardi, causano perdite) conviene delegare interamente al livello Trasporto la responsabilità di effettuare i controlli necessari su eventuali errori trasmissivi
 - Nessun percorso fisso e nessuna informazione di stato sulla conversazione in corso tra **M** e **D**
 - Commutazione di pacchetto invece che di circuito
 - Modello analogo al **servizio postale**
 - Due pacchetti inviati dallo stesso nodo **M** allo stesso nodo **D** possono seguire due percorsi **totalmente diversi**

Livello rete (IP) - parte 1

Architettura degli Elaboratori 2 - T. Vardanega

Pagina 358

Scelte progettuali – 2

- I pacchetti (**datagram**) circolanti in questa modalità devono designare gli indirizzi dei nodi **D** ed **M**
 - **D** ed **M** devono possedere un indirizzo **unico** nell'intera rete
- I *router* usano questa informazione per determinare il balzo (*hop*) successivo da far compiere al pacchetto entro la *subnet*
 - Quale linea in uscita, verso quale prossimo *router*

Livello rete (IP) - parte 1

Architettura degli Elaboratori 2 - T. Vardanega

Pagina 359

Scelte progettuali – 3

- Servizi con garanzia di connessione (**connection-oriented**)
 - Per ogni richiesta di comunicazione si configura un circuito virtuale (*virtual circuit*) dedicato
 - Il circuito virtuale è bidirezionale e tutti i pacchetti vi viaggiano in modo ordinato e controllato
 - Informazione di stato associa i pacchetti di una connessione **M-D** al relativo percorso (circuito)
 - Modello analogo al **servizio telefonico**: i pacchetti scambiati in una stessa conversazione percorrono sempre lo stesso percorso

Livello rete (IP) - parte 1

Architettura degli Elaboratori 2 - T. Vardanega

Pagina 360

Scelte progettuali – 4

- Un circuito virtuale è un tragitto tra nodo **M** e nodo **D** che, una volta configurato (*set up*) è usato per tutto il traffico specifico di quella particolare comunicazione
- Ogni *router* deve poter mantenere **più** circuiti virtuali aperti **simultaneamente** e avviare ciascun pacchetto sul proprio circuito
- Ogni pacchetto deve perciò identificare il proprio circuito virtuale di appartenenza

Livello rete (IP) - parte 1

Architettura degli Elaboratori 2 - T. Vardanega

Pagina 361

Algoritmi di instradamento – 1

- Usati per decidere verso quale linea in uscita e quale successivo *router* avviare pacchetti
 - Circuito virtuale:** la scelta viene fatta una volta per tutte nell'inizializzare il circuito
 - Datagram:** la scelta viene fatta per ogni pacchetto in arrivo e può pertanto variare nel tempo
- 2 classi di algoritmi**
 - Statici:** ogni *router* è dotato di tabelle predefinite che associano percorsi a destinazioni
 - Dinamici:** tengono conto di eventuali modifiche (anche temporanee) della topologia delle *subnet* e del volume di traffico sui collegamenti disponibili

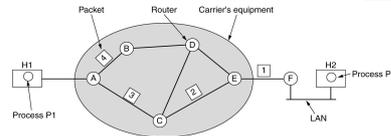
Livello rete (IP) - parte 1

Architettura degli Elaboratori 2 - T. Vardanega

Pagina 362

Algoritmi di instradamento – 2

Modalità **datagram**: i pacchetti viaggiano su percorsi indipendenti



A's table		C's table		E's table	
initially	later	initially	later	initially	later
A: -	A: -	A: A	A: A	A: C	A: C
B: B	B: B	B: A	B: A	B: D	B: D
C: C	C: C	C: -	C: -	C: C	C: C
D: B	D: B	D: D	D: D	D: D	D: D
E: C	E: B	E: E	E: E	E: -	E: -
F: C	F: B	F: E	F: E	F: F	F: F
Dest. Line					

Ciascuna tabella di routing indica il passo successivo (1 balzo) verso ciascuna delle destinazioni note

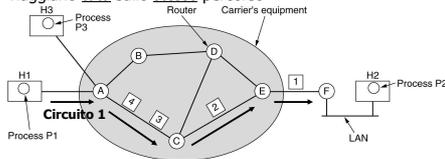
Livello rete (IP) - parte 1

Architettura degli Elaboratori 2 - T. Vardanega

Pagina 363

Algoritmi di instradamento – 3

Circuito virtuale: i pacchetti di una stessa conversazione viaggiano tutti sullo stesso percorso



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
H2: 1	C: 2	A: 2	E: 2	C: 2	F: 2

Ciascuna tabella di routing fissa il percorso di ogni singolo circuito

Livello rete (IP) - parte 1

Architettura degli Elaboratori 2 - T. Vardanega

Pagina 364

Algoritmi di instradamento – 4

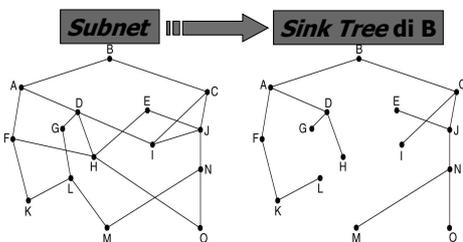
- Principio di ottimalità:** se J si trova sul **cammino ottimale** da I a K allora quello è anche il cammino ottimale da J a K
 - Metrica di ottimalità** è la lunghezza del percorso espressa, p.es., in numero di balzi
- L'insieme dei cammini ottimali da ogni possibile sorgente a una stessa destinazione è un albero con radice nella destinazione (**sink tree**)
 - Questa proprietà garantisce l'arrivo a destinazione di ogni pacchetto inviato in un tempo finito
 - Ma solo se i collegamenti non subiscono modifiche
 - Se l'albero resta albero

Livello rete (IP) - parte 1

Architettura degli Elaboratori 2 - T. Vardanega

Pagina 365

Algoritmi di instradamento – 5



Le connessioni fornite da una *subnet* danno origine a ≥ 1 *sink tree* per ogni possibile destinazione (p.es. B)

Livello rete (IP) - parte 1

Architettura degli Elaboratori 2 - T. Vardanega

Pagina 366

Algoritmi di instradamento – 6

- Varie metriche possono essere usate per determinare la lunghezza di un cammino
 - Numero di balzi**
 - # archi attraversati sul grafo
 - Distanza geografica**
 - Σ distanze espresse come attributo di ciascun arco
 - Tempo medio di attesa** presso un certo *router*
 - Attributo di ciascun nodo
 - Capacità** (portanza) del collegamento
 - Attributo di arco
 - ...

Livello rete (IP) - parte 1

Architettura degli Elaboratori 2 - T. Vardanega

Pagina 367

Algoritmi di instradamento – 7

- **Algoritmo di Dijkstra (statico)**
 - Ogni arco ha un attributo iniziale che ne indica la lunghezza
 - Ogni nodo raggiungibile da **A** in 1 balzo riceve un'etichetta indicante la distanza da **A** e il nodo adiacente di provenienza nel cammino
 - Ogni nodo raggiunto che risulta più vicino nel cammino da **A** viene marcato come "permanente"
 - Gli altri vengono marcati come "tentativi"
 - I nodi non ancora toccati sono considerati a distanza infinita
 - Si visita ogni nodo adiacente a ogni nodo marcato "permanente" e si aggiorna la distanza da **A** sommandovi la lunghezza percorsa sul relativo arco

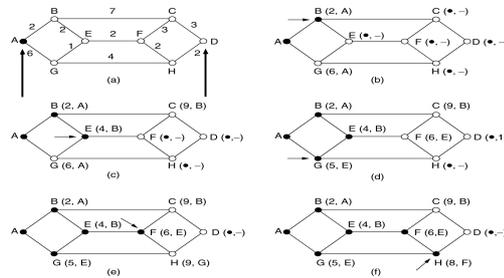
Livello rete (IP) - parte 1

Architettura degli Elaboratori 2 - T. Vardanega

Pagina 368

Algoritmi di instradamento – 8

Determinazione del percorso più breve tra **A** e **D** con l'algoritmo di Dijkstra



Livello rete (IP) - parte 1

Architettura degli Elaboratori 2 - T. Vardanega

Pagina 369

Note all'algoritmo di Dijkstra

- In figura
 - I nodi rappresentati in nero sono stati marcati come "permanenti"
 - I nodi non ancora raggiunti da un cammino con origine in **A** sono marcati come a distanza infinita
 - Quando più cammini raggiungono lo stesso nodo l'etichetta che viene assegnata al nodo è ovviamente quella del cammino minore

Livello rete (IP) - parte 1

Architettura degli Elaboratori 2 - T. Vardanega

Pagina 370

Algoritmi di instradamento – 9

- **Alluvione, flooding (statico)**
 - Ogni pacchetto in ingresso è emesso **in copia** su ciascuna linea in uscita
 - Per limitare le emissioni inutili ci si può limitare ai soli collegamenti nella "direzione giusta"
 - Ogni pacchetto può percorrere al più un **numero di balzi** prefissato
 - Un apposito attributo contatore viene decrementato ad ogni balzo
 - Quando il contatore giunge a 0 quella copia del pacchetto viene distrutta

Livello rete (IP) - parte 1

Architettura degli Elaboratori 2 - T. Vardanega

Pagina 371

Algoritmi di instradamento – 10

- **Stato del collegamento, link state (dinamico)**
 - Ogni **router** si aggiorna sull'indirizzo e la distanza dei propri vicini e determina il cammino più breve verso di essi tramite l'invio di **pacchetti di prova**
 - I nomi dei **router** devono essere **unici**
 - L'aggiornamento deve essere ripetuto **periodicamente**
 - I **router** si scambiano pacchetti contenenti la propria informazione topologica
 - L'aggiornamento asincrono delle tabelle dei **router** può rendere instabile la determinazione dei cammini
 - Ottenuta tutta l'informazione necessaria ciascun **router** usa l'algoritmo di Dijkstra per i cammini minimi

Livello rete (IP) - parte 1

Architettura degli Elaboratori 2 - T. Vardanega

Pagina 372

Nodi mobili – 1

- 2 tipi di utenti (= nodi) **mobili** in una rete
 - Utenti che si connettono in tempi diversi da diverse **postazioni fisse**
 - Utenti di **postazioni mobili (roaming)**
- Ogni utente ha una base **HB** di riferimento fissa
- La rete globale è suddivisa in aree ciascuna delle quali possiede
 - ≥ 1 agenti **FA** degli utenti mobili (**foreign** **connessi**)

Livello rete (IP) - parte 1

Architettura degli Elaboratori 2 - T. Vardanega

Pagina 373

Nodi mobili – 2

- Ogni utente mobile **MU** si registra nell'area ove si trova al momento
 - Ogni **FA** effettua *broadcast* periodico a tutti i possibili **MU** nell'area
 - Ogni **MU** notifica al **FA** di contatto la propria **HB** e il proprio **accredito**
 - Il **FA** comunica a **HA** di **MU** il proprio indirizzo **FA** e **accredito**
 - L'**HA** di **MU** invia a **FA** l'autorizzazione a registrare **MU**
 - Il **FA** registra **MU** nella sua anagrafe locale
- Ogni comunicazione destinata a **MU** va prima alla sua **HB** il cui **HA** la invia al **FA** presso il quale esso è registrato
 - I pacchetti originari vengono inviati come parte dati dei pacchetti inviati da **HA** a **FA** (*tunneling*)

Livello rete (IP) - parte 1

Architettura degli Elaboratori 2 - T. Vardanega

Pagina 374

Controllo di congestione – 1

- Si ha congestione quando troppi pacchetti si trovano entro una stessa *subnet*
 - I pacchetti in ingresso a un *router* le cui code di lavoro sono piene vanno persi
 - Code più ampie peggiorano il problema
 - I pacchetti stanno più a lungo in coda, **M** li considera persi e li replica, aumentando così il traffico!
- Il **controllo di congestione** garantisce che la *subnet* sia sempre capace di trattare tutto il suo traffico interno
- Il **controllo di flusso** evita che **M** possa mai congestionare **D**

Livello rete (IP) - parte 1

Architettura degli Elaboratori 2 - T. Vardanega

Pagina 375

Controllo di congestione – 2

- Le congestioni possono essere prevenute per via strutturale o per controllo d'esecuzione
- Dimensionamento strutturale di traffico**
 - Leaky bucket**: fissa il flusso massimo di uscita, vi dimensiona la coda di ingresso e scarta tutto il traffico in eccesso
 - Il flusso d'uscita non può avere picchi
 - Il "secchio" è la coda e perde pacchetti se si riempie
 - Token bucket**: il controllore produce gettoni con periodo T e ogni pacchetto in arrivo viene accettato se vi è un gettone per lui altrimenti resta in coda
 - I gettoni possono essere accumulati fino alla capacità massima del "secchio" → il flusso d'uscita può avere picchi
 - Il "secchio" accoda (e perde) gettoni e non pacchetti
 - Naturalmente fino alla capienza massima della coda dei pacchetti

Livello rete (IP) - parte 1

Architettura degli Elaboratori 2 - T. Vardanega

Pagina 376

Controllo di congestione – 3

- Soluzioni dinamiche a controllo di traffico**
 - Soffocamento di linea**: se il flusso in ingresso da una linea supera una data soglia si invia una notifica (**choke packet**) a **M**, marcando il suo ultimo pacchetto accettato come "avvisato" e facendolo proseguire
 - Ricevuta la notifica **M** riduce il flusso in uscita
 - Metodo rischioso perché si basa sulla buona fede di **M**
 - Load shedding**: il *router* in sovraccarico scarta pacchetti secondo criteri arbitrari e/o secondo l'importanza loro attribuita da **M**

Livello rete (IP) - parte 1

Architettura degli Elaboratori 2 - T. Vardanega

Pagina 377

Internet : IP – 1

- Il termine **internet**, originariamente coniato per denominare una **singola** interconnessione tra reti locali, denota ora un insieme di reti connesse da un **protocollo comune** in modo da formare una singola rete virtuale
- Il termine **Internet** denota la confederazione di reti **eterogenee** connesse da **IP** per formare una singola rete virtuale **globale**

Livello rete (IP) - parte 1

Architettura degli Elaboratori 2 - T. Vardanega

Pagina 378

Internet : IP – 2

- IP** si propone 5 obiettivi
 - Definire l'unità di comunicazione su rete
 - Datagram**
 - Definire lo schema di indirizzamento di **tutti** i nodi della rete (non solo i *router*!)
 - Offrire i servizi attesi dal livello Trasporto (*host-to-host*) utilizzando i servizi forniti dal livello inferiore
 - Instradare i **datagram** verso le rispettive **D**
 - Frammentare i **datagram** all'invio e ricostituirli alla ricezione
 - Come misura necessaria per l'invio a livelli 2 che accettino unità trasmissive di dimensione inferiore al **datagram**

Livello rete (IP) - parte 1

Architettura degli Elaboratori 2 - T. Vardanega

Pagina 379

Internet : IP – 3

- Esiste una pluralità di protocolli a livello > 3
- Esiste una pluralità di protocolli a livello < 3
- Vi è un solo protocollo a livello IP

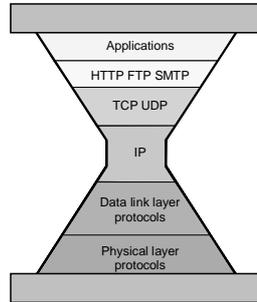


Immagine tratta da:
Internet Engineering
www.cs.virginia.edu/~cs458/

Livello rete (IP) - parte 1

Architettura degli Elaboratori 2 - T. Vardanega

Pagina 380

Internet : IP – 4

- IP è il protocollo di livello più alto che si trova realizzato sia su nodi *host* che su nodi *router*

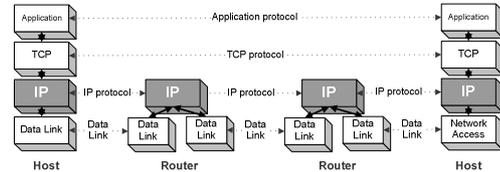


Immagine tratta da: Internet Engineering (www.cs.virginia.edu/~cs458/)

Livello rete (IP) - parte 1

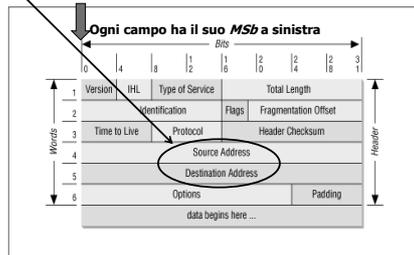
Architettura degli Elaboratori 2 - T. Vardanega

Pagina 381

Internet – 3

Indirizzi IP

Il formato di un datagram



Livello rete (IP) - parte 1

Architettura degli Elaboratori 2 - T. Vardanega

Pagina 382

Internet – 4

- L'intestazione (*header*) occupa sempre le prime 5 parole da 32 bit del datagram
 - Anche la 6ª parola, se così specificato nel campo **IHL** (*Internet Header Length*)
- L'intestazione fornisce le informazioni che permettono la consegna a destinazione
- Il datagram viene trattato e trasmesso in ordine *Big-Endian* (*network-byte order*)
 - *Big-Endian*: la parte più significativa di un dato viene posta in memoria all'indirizzo superiore
 - *Network-byte order*: riga per riga da sx a dx

Livello rete (IP) - parte 1

Architettura degli Elaboratori 2 - T. Vardanega

Pagina 383

Internet – 5

- **Versione**: versione del protocollo cui appartiene il datagram
- **Tipo di servizio**: caratteristiche richieste alla trasmissione (veloce, sicura)
 - Spesso ignorato
- **Lunghezza totale**: specifica la dimensione del datagram, e dunque della sua area dati ($\leq 64k-1$ B)
- **Identificativo**: flusso di comunicazione cui appartengono i dati del datagram
- **Flags e Fragmentation Offset**: indicano se il datagram possa essere frammentato e la posizione relativa del frammento nella sequenza
 - Reti fisiche diverse impongono limiti diversi sulla massima dimensione delle unità trasportate

Livello rete (IP) - parte 1

Architettura degli Elaboratori 2 - T. Vardanega

Pagina 384

Internet – 6

- **Tempo di vita**: numero di prossimi balzi (*hop*) ancora consentiti al datagram
 - Decrementato a ogni balzo: quando va a 0, causa la distruzione del datagram e l'invio di una notifica a M
- **Protocollo**: tipo di trattamento richiesto dal datagram a livello trasporto (TCP o UDP)
- **Controllo (checksum) di intestazione**: per verifica di integrità dell'intestazione
 - Deve essere calcolato ogni volta che cambi il contenuto dell'intestazione e dunque ad ogni balzo
- **Opzioni**: riservato per sperimentazioni o future estensioni del protocollo

Livello rete (IP) - parte 1

Architettura degli Elaboratori 2 - T. Vardanega

Pagina 385