

# ARCHITETTURA DELLE RETI DI CALCOLATORI 1

FROSIO IURI

LAB. MAVR

frosio@dsi.unimi.it

## Cosa è Internet?

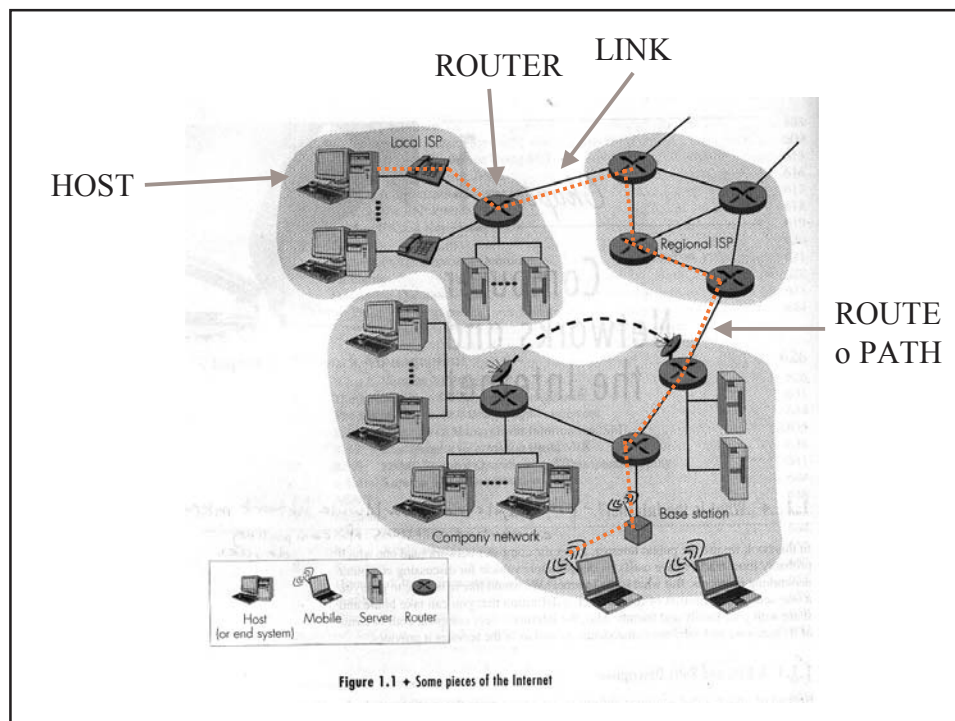
A SPECIFIC COMPUTER NETWORK... Non l'unica rete, ma la più importante!!!

INTERNET:

- NUTS & BOLTS: componenti HW e SW della rete;
- NETWORKING INFRASTRUCTURE: fornitura di servizi per applicazioni distribuite;

# Internet: Nuts & Bolts 1

- HOST o END SYSTEMS → La rete connette milioni di macchine (PC, UNIX Workstations, Servers, Mobile, ..., Toaster (!)) nel mondo;
- NETWORK APPLICATION PROGRAMS (Web browser, E-Mail programs, ...) → Girano sugli HOSTS;
- PROTOCOLLI (TCP, IP) → Codificano lo scambio di informazioni tra componenti diversi;
- COMMUNICATION LINKS → cavi coassiali, doppino di rame, ..., differenti bande di trasmissione (bandwidth);
- ROUTERS (o switching devices) → Connessione indiretta tra HOSTS;



## Internet: Nuts & Bolts 3

- Protocollo IP: Specifica il formato delle informazioni tra ROUTER e END SYSTEM;
- INTERNET = RETE DI RETI
  - Protocollo IP in ogni rete (pubblica, privata, ...);
  - Naming e Addressing secondo IP all'interno di ogni rete;
  - IP = INTERNET DIAL TONE;
  - Configurazione di ogni rete a scelta dell'amministratore (Naming e Addressing esclusi).

## Internet: Nuts & Bolts 4

### GERARCHIA:

END SYSTEMS

access networks (LAN, phone line & modem, ...)

LOCAL INTERNET SERVICE PROVIDER (ISPs)

REGIONAL INTERNET SERVICE PROVIDER

NATIONAL INTERNET SERVICE PROVIDER

INTERNATIONAL INTERNET SERVICE PROVIDER

### TOPOLOGIA:

MOLTO (!) COMPLESSA

## Internet: Nuts & Bolts 5

### INTERNET:

- Tutto ciò che è accessibile pubblicamente;

### RETI PRIVATE:

- Hosts non accessibili pubblicamente;
- Hosts visibili solo dall'interno della rete;
- Vengono dette INTRANETS;
- Usano (non sempre) gli stessi protocolli di Internet public!

## Internet: Nuts & Bolts 6

### INTERNET STANDARDS

- Creati, testati e implementati dalla Internet Engineering Task Force (IETF);
- Definiscono i protocolli:
  - TCP
  - IP
  - HTTP (web)
  - SMTP (e-mail)
  - ...

# Internet: Networking Infrastructure 1

INTERNET PERMETTE AD APPLICAZIONI DISTRIBUITE CHE GIRANO SUGLI END SYSTEMS DI SCAMBIARE DATI.

Applicazioni:

- Remote Login, file transfer, e-mail, www, ...

WEB: Non una rete separata, bensì un'applicazione distribuita che utilizza i servizi di comunicazione offerti da Internet!

WWW potrebbe "girare" su una rete diversa da Internet...Ma:

Internet  $\Leftrightarrow$  >43M PC connessi, >100M utenti (nel 1999)

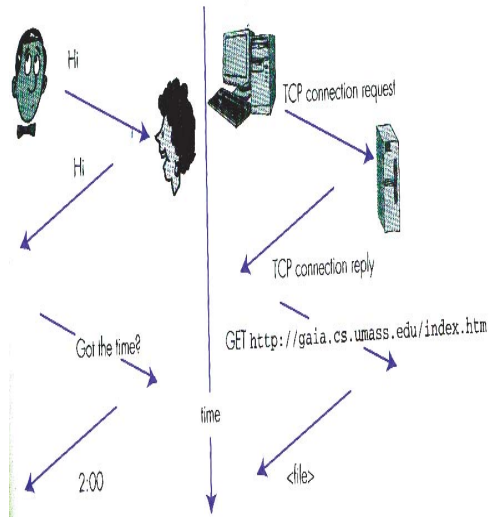
# Internet: Networking Infrastructure 2

SERVIZI OFFERTI:

- 1) CONNECTION ORIENTED SERVICES: garantiscono l'arrivo dei dati interamente e nel giusto ordine;
- 2) CONNECTIONLESS SERVICES: non garantiscono l'arrivo dei dati interamente e nel giusto ordine;

Le applicazioni vengono progettate per 1) o per 2), ma... Nessuno assicura un tempo max di trasferimento dei dati!!!

## Cosa è un protocollo? 1



### PROTOCOLLO

“BUONE MANIERE”

- 1) Saluta prima di chiedere l'ora;
- 2) Chiedi l'ora sola dopo aver ricevuto un saluto di risposta;
- 3) ...

## Cosa è un protocollo? 2

NETWORK PROTOCOLS: Versioni HW e SW

Tutte le comunicazioni in rete avvengono attraverso un protocollo!

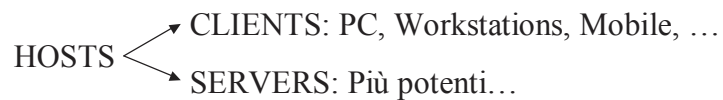
UN PROTOCOLLO DEFINISCE IL FORMATO E L'ORDINE DEI MESSAGGI SCAMBIATI TRA DUE O PIU' ENTITA' COMUNICANTI, NONCHE' LE AZIONI INTRAPRESE AL MOMENTO DELLA TRASMISSIONE DI UN MESSAGGIO O DI ALTRI EVENTI.

## Cosa è un protocollo? 3

ESEMPI:

- PROTOCOLLO “BUONE MANIERE”;
- Protocollo HTTP: Definisce il modo in cui vengono trasferite le pagine web;
- Nei ROUTERS: Determinazione del path tra Source Host e Destination Host;
- Protocolli HW: Codifica dei bit sulla scheda di rete;
- Congestion & Control Protocols: Gestione del rate di trasmissione dei dati sulla base del traffico in rete;

## Network Edge 1



Caso più frequente: CLIENT PROGRAM su un host scambia info con un altro host (SERVER) → CLIENT/SERVER MODEL.

Routers, links, ... (l'intera rete) → BLACK BOX!

Applicazione: Connection oriented vs. Connectionless Service

# Connection Oriented Service 1

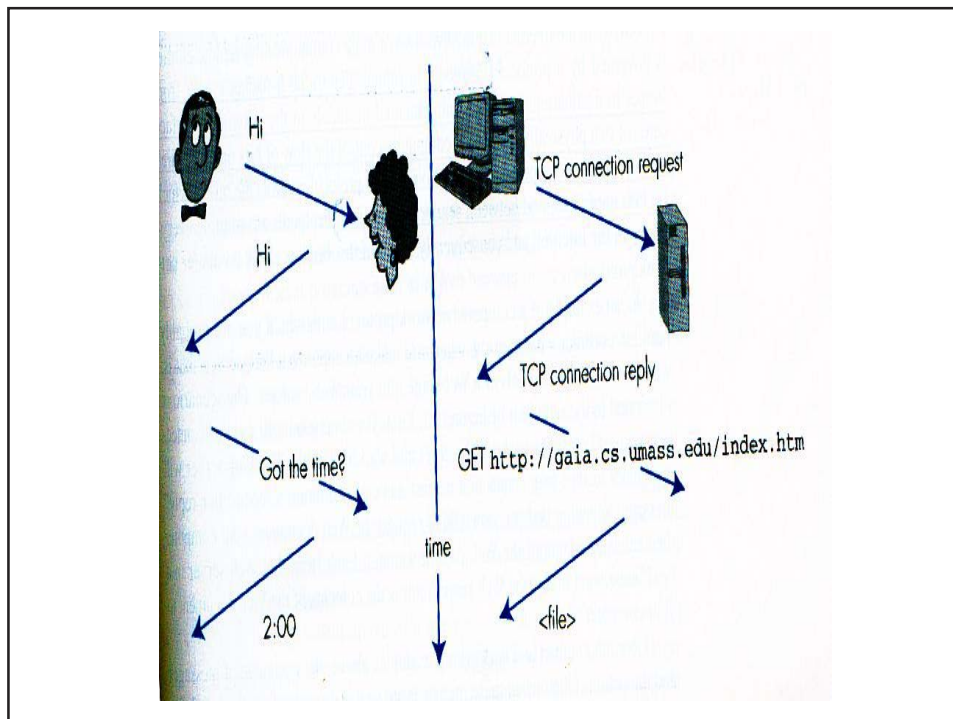
- Client e Server si scambiano control packets prima dell'invio dei dati reali (HANDSHAKING PROCEDURE);

- Protocollo usato: Transmission Control Protocol TCP:

- Da Client a Server: TCP Connection Request;

- Da Server a Client: TCP Connection Reply;

- Da Client e Server: Get http://... (Conferma)





## Connection Oriented Service 3

Connessione "DEBOLE":

- I due end systems sanno di essere connessi;
- Tutti i routers lungo il path non lo sanno!!!

La connessione TCP alloca risorse (buffers) e variabili di stato nei due end systems. I packet switches (=routers) non possiedono informazioni riguardanti la connessione.

## Connection Oriented Service 4

L'utilizzo del connection oriented service permette l'implementazione dei seguenti servizi:

- RELIABLE DATA TRANSFER: trasferimento di TUTTI i dati, nel giusto ORDINE → affidabilità!

1) Handshaking procedure tra A e B:  $A \leftarrow \rightarrow B$ .

2) Real data da A a B:  $A \rightarrow B$ .

3) Acknowledgement da B ad A:  $A \leftarrow B$ .

Se il passo 3) non avviene, A trasmette nuovamente i dati a B.

## Connection Oriented Service 5

- FLOW CONTROL: Un end system non può spedire dati troppo “velocemente” all’altro. Il transmission rate (bps) viene controllato, anche attraverso l’utilizzo di buffer.
- CONGESTION CONTROL: Quando il traffico in rete è molto intenso, gli host vengono forzati a trasmettere ad un rate inferiore.

TCP: è l’Internet Connection Oriented Service, si occupa di reliable transport, flow control, congestion control.

## Connectionless Service 1

- Non c’è Handshaking procedure;
- Trasmissione più veloce dei dati;
- Non c’è garanzia sull’arrivo dei dati e sul loro ordine;
- Non ci sono Flow Control e Congestion Control;

Internet Connectionless Service: UDP (User Datagram Protocol)

TCP: Telnet (Remote login), SMTP (e-mail), FTP (file transfer), ...

UPD: Internet Phone, Video Conferencing, ...

# Network Core 1

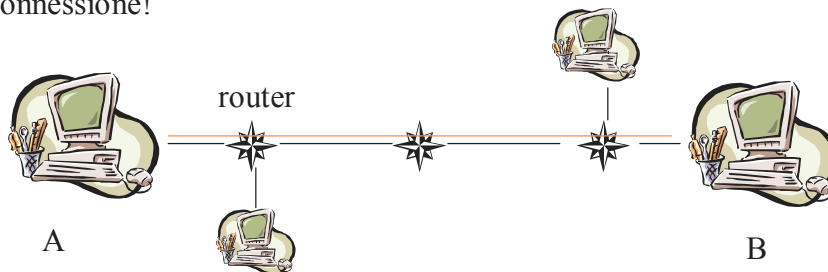
Network core = Mesh di routers che connettono gli end systems;

Gestione del Core:

- 1) **CIRCUIT SWITCHING**: Le risorse (buffers, links, bandwidth) lungo un path sono **RISERVATE** per tutta la durata della sessione; es. reti telefoniche: connessione → creazione circuito riservato → garanzia rate costante.
- 2) **PACKET SWITCHING**: Si accede alle risorse su domanda, si formano code (**QUEUE**) per l'accesso ad un communication link; es. Internet: dopo la connessione, un messaggio inviato può restare nel buffer di un router in attesa che si liberi un link.
- 3) Reti ibride: es. ATM technology

# Circuit Switching 1

- Host A vuole comunicare con Host B;
- Si costruisce un circuito dedicato tra A e B;
- Il link tra A e B ha n canali (multiplexing);
- $1/n$  della larghezza di banda del link è occupato per tutta la durata della connessione!



## Circuit Switching 2: Multiplexing

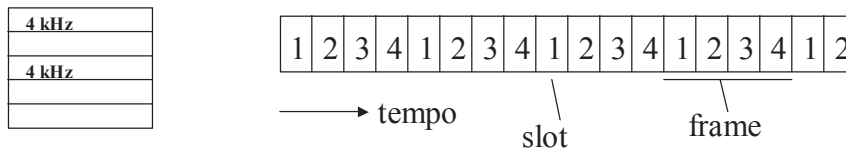
1) Frequency Division Multiplexing FDM: lo spettro di frequenza è diviso tra le diverse comunicazioni (es. bande di 4KHz per le linee telefoniche);

2) Time Division Multiplexing TDM (standard attuale): il tempo è diviso in frame di durata fissa; ogni frame è diviso in slot dedicati a connessioni diverse.

$$\text{Transmission Rate} = \text{frame rate} * (\text{\#bit} / \text{slot})$$

Esempio: link che trasmette 8,000 frame / s. Ciascuno slot è di 8 bit. La velocità di trasmissione sarà:

$$\text{Transmission Rate} = 8,000 * 8 \approx 64\text{kbs}$$



## Circuit Switching 3

Devo determinare quanto tempo mi viene richiesto per trasferire un file su un link TDM.

File: 640 Kbit.

Link: TDM con 24 slot.

Bit Rate: 1536 Mbp/s.

Per stabilire la connessione: 500ms.

Per ogni circuito:  $\text{Transmission Rate} = 1536\text{Mbp/s} / 24 = 64\text{kbp/s}$

$T_{\text{tot}} = 640\text{kb} / 64\text{kbp/s} + 0.5\text{s} = 10\text{s} + 500\text{ms} = 10.5\text{ s}$

Problemi nella modalità Circuit switching: circuiti silenti (esempio, occupo il link anche se, all'interno di una telefonata, non parlo).

Vantaggio: una volta ricevuto lo slot all'interno del link, viene mantenuto fino alla fine della trasmissione. Posso avere una stima ragionevole della durata del trasferimento.

## Spiegazione esercizio

Dobbiamo trasmettere un file delle dimensioni di 640Kbit, su di un link TDM con 24 slot. Ciò significa che, in un secondo di trasmissione, 1/24 s viene dedicato alla connessione n°1, 1/24 s alla connessione n°2 e così via (Time Division Multiplexing); l'utente che trasmette il file attiva una sola connessione, dunque ha a disposizione 1/24 della larghezza di banda del link, cioè 1/24 del transmission rate. Essendo il transmission rate di 1536Kbp/s, il singolo utente può trasmettere il file al rate di  $1536\text{Kbp/s}/24 = 64\text{ kbp/s}$ . Il tempo totale per trasmettere il file è allora dato da  $640\text{kbit}/64\text{kbp/s} = 10\text{s}$ . A questi 10 secondi aggiungiamo 500ms per stabilire la connessione, per un totale di 10,5s.

## Packet Switching 1

- I Sources spezzano i messaggi in pacchetti (packets) più piccoli; vengono trasmessi i packets!
- I pacchetti sono trasmessi al full transmission rate di ogni link;
- STORE & FORWARD TRANSMISSION: prima di spedire un pacchetto i packet switches (routers) aspettano di averlo ricevuto per intero; ciò genera un ritardo (STORE & FORWARD DELAY) proporzionale alla lunghezza del pacchetto in bit;
- Se un link è occupato, il packet viene messo nell'output buffer del router (QUEUEING); si genera un ulteriore ritardo proporzionale al traffico in rete (QUEUEING DELAY);
- buffer pieni → PACKET LOSS!

## Packet Switching 2 - Esempio

Calcoliamo quanto tempo ci vuole per trasferire un pacchetto di  $L$  bit da un host A ad un host B.

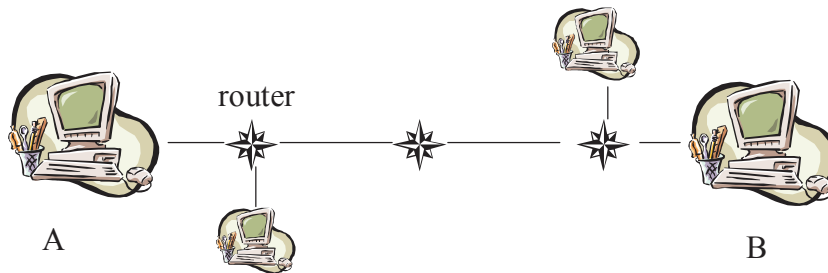
Supponiamo che ci siano un numero pari a  $Q$  link consecutivi tra l'host A e l'host B, tutti con la stessa velocità di trasmissione.

Supponiamo che la velocità di trasmissione (Transmission Rate) sia di  $R$  bps.

Supponiamo trascurabili il tempo di queuing, ed il tempo di propagazione end-to-end trascurabili. Supponiamo che non serva la procedura di hand-shaking (e.g. UDP).

*Store&Forward-Delay =  $L/R$  per 1 link.*

*Store&Forward-Delay totale =  $QL/R$ .*



## Packet vs. Circuit Switching 1

+ Packet switching: migliore gestione Bandwidth, più semplice ed efficiente, meno costoso;

- Packet switching: no real time (es. videoconferenze), ritardi variabili ed imprevedibili.

## Packet vs. Circuit Switching - Esempio

### *ESEMPIO*

link: 1Mbps  
 dati: 100kbps intervallati con 0kbps  
 user: attivi in media il 10% del tempo.

### *CIRCUIT SWITCHING*

- Max 10 utenti (1Mbp/s/100kbp/s); Una volta connessi 10 utenti, la larghezza di banda del link viene completamente saturata, sia che gli utenti stiano trasmettendo sia che non lo stiano facendo.

### *PACKET SWITCHING*

- Supponiamo di avere 35 utenti attivi al 10% →  $P < 0.04\%$  di avere più di 10 utenti che trasmettono contemporaneamente (dalla distribuzione binomiale);  
 - Allora, se ci sono meno di 10 utenti in trasmissione, la larghezza di banda utilizzata è inferiore a 1Mbps → non ci sono ritardi!  
 - Se ho più di 10 utenti in trasmissione ( $P < 0.04\%$ ) → queuing delays.

## Packet vs. Message switching 1

Perché perdere tempo per spezzare e poi riassemblare un messaggio?

MESSAGE SWITCHING: packets = messages

Il router deve aspettare l'arrivo dell'intero messaggio prima di spedirlo sul link successivo!

M=PPPPP

t	Source	Router1	Router2	Destination
1	M			
1	PPPPP			
2		M		
2		P	P	PPP
3			M	
3				PPPPP

## Packet vs. Message Switching - esempio

Devo trasmettere un messaggio lungo 7.5 Mbits.  
Supponiamo che tra la sorgente e la destinazione ci siano due router che operano packet switch e 3 link. Supponiamo che ciascun link abbia una velocità di trasferimento di 1.5Mb/s. Supponiamo che non ci sia congestione sul link.

S → link 1.5Mbps → R1 → link 1.5Mbps → R2 → link 1.5 Mbps → D

### **Message switching:**

Per un link  $7.5 \text{ Mb} / 1.5 \text{ Mbps} = 5 \text{ s}$   
 $T_{\text{tot}} = 5 \text{ s} + 5 \text{ s} + 5 \text{ s} = 15 \text{ s}$

### **Packet switching:**

Supponiamo che il messaggio sia spezzettato in 5000 packets: P1,P2, ...

Ciascun pacchetto sarà di  $7.5 \text{ Mb} / 5000 = 1.5 \text{ kb}$

Per il singolo packet  $1.5 \text{ kb} / 1.5 \text{ Mbps} = 1 \text{ ms}$

Mentre P1 va da R1 a R2, P2 va da S a R1, ...

$T_{\text{tot}} = 5000 * 1 \text{ ms} + 2 \text{ ms} = 5.002 \text{ s}$

Con packet switching si ha una pipe-line della trasmissione.

## Packet vs. Message Switching 3

MESSAGE SWITCHING → processo seriale, alcuni nodi “dormono” mentre gli altri lavorano;

PACKET SWITCHING → processo parallelo, tutti i nodi vengono utilizzati contemporaneamente;

- Se vengono riscontrati errori di trasmissione, l'intero pacchetto viene eliminato;

- Ogni pacchetto ha uno HEADER (info di controllo, source, destination, ...) di dimensione ~ costante. Con il packet switching → overhead di dati maggiore!



# Packet Switched Networks 1

PACKET SWITCHED NETWORKS:

## 1) VIRTUAL CIRCUIT NETWORKS:

Instradamento dei messaggi secondo virtual circuit numbers; in questo tipo di reti i routers devono mantenere informazioni di stato sulle connessioni!

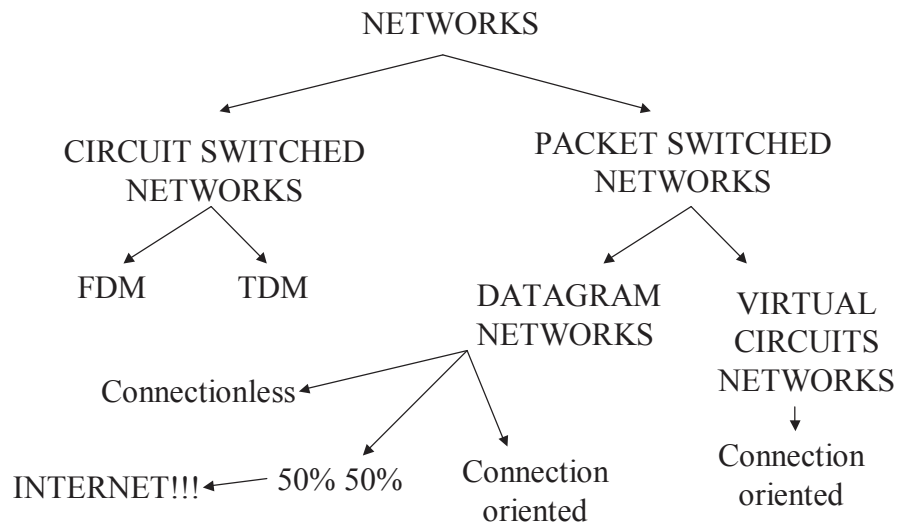
## 2) DATAGRAM NETWORKS:

Instradamento dei messaggi secondo l'host di destinazione (es. protocollo IP); i routers non devono mantenere informazioni di stato sulle connessioni!

# Datagram Networks 1

- Ogni packet contiene nello header la destinazione finale, organizzato in maniera GERARCHICA (es. indirizzo postale);
- Ogni routers ha una routing map che assegna ad ogni destinazione un certo link in uscita;
- Non è necessario mantenere informazioni di stato nel router riguardo la connessione;
- Packet ~ automobilista che ad ogni bivio chiede la direzione da prendere;
- Router ~ passante (con mappa stradale in tasca).

# Internet Taxonomy 1



## Reti di accesso e mezzi fisici 1

ACCESS NETWORK: link fisico tra host e edge router

1) Residenziali

PC + modem over a POTS (plain old telephone system)

PC → digital → modem → analog → phoneline → modem → digital → ISP

Dialup line verso un Internet Server Provider (ISP)

Modem: 56kbps

Efficienza complessiva del sistema: bassa!

## Reti di accesso e mezzi fisici 2

ISDN ~ miglior modem (128kbps), l'host trasmette direttamente in digitale verso l'ufficio centrale di una compagnia telefonica;

ASDL (Asymmetrical Digital Subscriber Line) → funziona sulle vecchie linee telefoniche, 8Mbps router→host, 1Mbps host→router. Usa frequency division multiplexing e modem dedicati:

High Speed Downstream Channel: 50kHz...1MHz

Medium Speed Upstream Channel: 4kHz...50kHz

Ordinary POTS Telephone Channel: 0kHz...4kHz

Il rate reale dipende dalla distanza host/ISP:

<=3000m      <=8Mbps

<=6000m      <=2Mbps

## Reti di accesso e mezzi fisici 3

HFC: fibre ottiche fino ad un punto di smistamento; fibre coassiali fino all'host di casa. Richiede modem dedicati.

Upstream: fino a 10Mbps      Downstream: fino a 768kbps

Canali: shared

2) Company Access Network

LAN (Local Area Network) per la connessione degli end systems con l'edge router; tecnologia LAN: Ethernet (10...100Mbps) con doppini di rame e cavi coassiali (condivisi).

3) Mobile Access Network

Onde radio, decine di kbps, banda shared, protocolli MAP e IP.

## Reti di accesso e mezzi fisici 4

BIT = (assenza di) impulso elettromagnetico attraverso un trasmettitore.

Mezzi fisici:

- guided media → fibre ottiche, cavi coassiali, doppino di rame, ...
  - unguided media → atmosfera, spazio, ...
- 
- Il mezzo di trasmissione è economico;
  - La sua installazione è molto dispendiosa!

## Reti di accesso e mezzo fisici 5

*Doppino di rame:* 2 fili avvolti a spirale per evitare interferenze e.m.. Fino a 100Mbps per distanze inferiori a 200m. Economico. Ideale per high speed LAN.

*Cavo coassiale:* 2 conduttori concentrici, minori interferenze, maggior velocità.

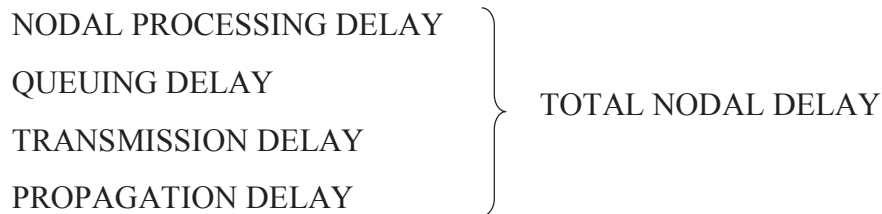
BaseBand Coaxial Cable: bit non shiftati in banda prima della trasmissione su cavo. Necessario HUB.

BroadBand Coaxial Cable: bit shiftati in banda prima della trasmissione su cavo.

*Fibra ottica:* fino a 100Gbps. Ideali per lunghe distanze, dispositivi ottici costosi.

*Onde radio:* ideali per lunghe distanze e Mobile, 50kbps. Problemi per interferenze e zone d'ombra.

## Packet Switched Networks: Delays 1



NODAL PROCESSING DELAY: Il router deve leggere lo header del packet per determinare dove è diretto ( $\mu\text{s}$ );

QUEUING DELAY: Pacchetto nell'output buffer del router in attesa che il link successivo si liberi. Dipende dal numero di pacchetti arrivati prima nel router, cioè dal traffico in rete ( $\mu\text{s} \dots \text{ms}$ );

## Packet Switched Networks: Delays 2

TRANSMISSION DELAY: per un pacchetto di  $L$  bits, Rate transmission sul link di  $R$  bit/s, è dato da:

$$\textit{Transmission delay} = L/R \text{ (}\mu\text{s)}.$$

Dipende anche dal mezzo fisico di cui è composto il link.

PROPAGATION DELAY: dipende dal mezzo fisico del link e dalla distanza tra i due routers; la velocità di propagazione è nel range di  $2 \cdot 10^8 \dots 3 \cdot 10^8$  m/s; si ha:

$$\textit{Propagation delay} = (\textit{dist})/(\textit{propagation speed}) \text{ (ms)}.$$

## Packet Switched Networks: Delays 3

$$D_{\text{nodal}} = D_{\text{proc}} + D_{\text{queue}} + D_{\text{trans}} + D_{\text{prop}}$$

$D_{\text{queue}}$ : dipende dal traffico in rete, cambia da pacchetto a pacchetto, grande variabilità; va caratterizzato statisticamente!

Quando  $D_{\text{queue}}$  diventa significativa (con buffer infinito)?

$a$  = rate medio di arrivo dei pacchetti       $R$  = transmission rate

$L$  = lunghezza in bit di ogni pacchetto

In coda  $\rightarrow$  arrivano  $aL$  bits al secondo

Definiamo **Traffic Intensity** =  $L a / R$

Diventa significativa quando  $\# \text{bit}_{\text{ingresso}} > \# \text{bit}_{\text{trasmessi}}$ .

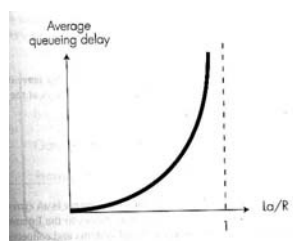
## Packet Switched Networks: Delays 4

$L a / R > 1 \rightarrow$  arrivano più bit di  $R \rightarrow$  la coda cresce!

$L a / R < 1 \rightarrow$  non si forma la coda...

Regola per il disegno di una rete:  $L a / R < 1$

In realtà: l'arrivo dei pacchetti è un processo stocastico!!!  $L a / R$  non caratterizza del tutto la situazione...



Se  $L a / R \rightarrow 1$ , un piccolo incremento nel traffico di rete comporta un grande incremento di  $D_{\text{queue}}$  medio.

Regola per il disegno di una rete:  **$L a / R \ll 1$**

## Packet Switched Networks: Delays 5

- QUEEUING BUFFERS: hanno dimensione finita!
- E' dunque impossibile avere  $\dim(\text{queue}) \rightarrow \infty$ ;
- Se  $L\lambda/R \rightarrow 1$ , è alta la possibilità che il pacchetto in arrivo al router trovi la coda piena  $\rightarrow$  il pacchetto viene cancellato (PACKET LOSS);
- La frazione di PACKET LOSS è proporzionale al traffico in rete;

Performance di un nodo:     1) Queuing delay  
   2) % Packet Loss

## Packet Switched Networks: Delays 6

Trascuriamo il delay introdotto dal queuing (rete non congestionata).

- Definiamo:  $D_{end\ to\ end} = D_{proc} + D_{trans} + D_{prop}$ ;
- (Q-1) routers tra Source e Destination, rete non congestionata;
- Per ogni router  $\rightarrow D_{proc}$
- Ogni router trasmette a R bit/s  $\rightarrow D_{trans} = L/R$ ;
- Per ogni link ritardo di propagazione:  $D_{prop}$ ;

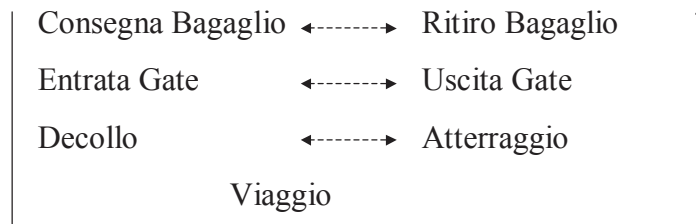
$$D_{end\ to\ end} = Q(D_{proc} + D_{trans} + D_{prop})$$

$$D_{tot} = D_{end\ to\ end} + \sum D_{queue}$$

# Protocol layers 1

Abbiamo visto come un pacchetto si sposta nella rete... Ma come è fatta l'architettura della rete?

Es. Sistema trasporto aereo



———— Spostamento all'interno della rete

←-----→ Servizi svolti da ogni nodo (architettura – layers structure)

La struttura a layers è gerarchica (stack).

# Protocol layers 2

Struttura a livelli → è possibile cambiare l'implementazione di un layer senza cambiare tutto il resto!

## NETWORK PROTOCOLS

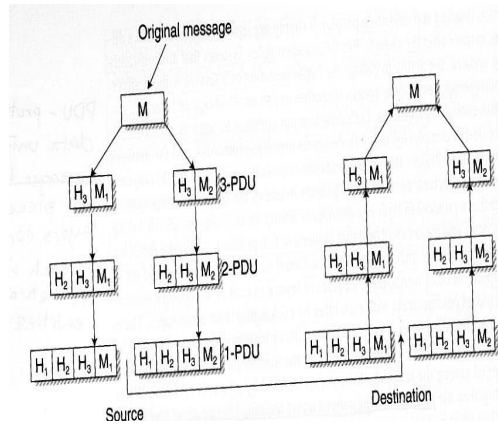
- I disegnatori di reti organizzano i protocolli a strati gerarchici (layers);
- Ogni protocollo appartiene ad uno degli strati;
- Il protocollo nello strato n è DISTRIBUITO tra le entità della rete che implementano il layer n;
- Entità della rete che implementano il layer n possono comunicare tra di loro attraverso il protocollo n. I messaggi scambiati sono detti n-PDUs (layer n Protocol Data Units).



## Protocol layers 3

SERVICE MODEL: lo strato n-1 offre un servizio allo strato n.

### PROTOCOL LAYERING



Ogni layer aggiunge degli header!

Es. servizio postale USA aggiunge codici a barre sulle buste. Tali codici (header) sono usati solo dai layer più bassi, non dal postino che usa solo H3 (via e numero civico)

## Internet Protocol Stack 1

L'Internet stack è organizzata in 5 layers:

Layer Name:	PDU-s Name:	Implementation
Physical	--	HW
Data Link	Frame	HW
Network	Datagram	HW & SW
Transport	Segment	SW
Application layer	Message	SW

Funzioni: Connection setup, multiplexing, segmentation & reassembly, error functions, flow control.

## Internet Protocol Stack 2

- 1) APPLICATION LAYER: Supporto alle applicazioni di rete. Include molti protocolli (HTTP, SMTP, FTP, ...).

 MESSAGE

- 2) TRANSPORT LAYER: Trasporto dei messaggi application layer tra client e server. Ha due protocolli: TCP (Connection-oriented service) e UDP (connectionless service).

 SEGMENT

- 3) NETWORK (IP) LAYER: Si occupa del routing diagram tra un host e l'altro. Il protocollo IP definisce i campi dell'IP datagram; il routing protocol stabilisce il cammino tra due host.

## Internet Protocol Stack 3

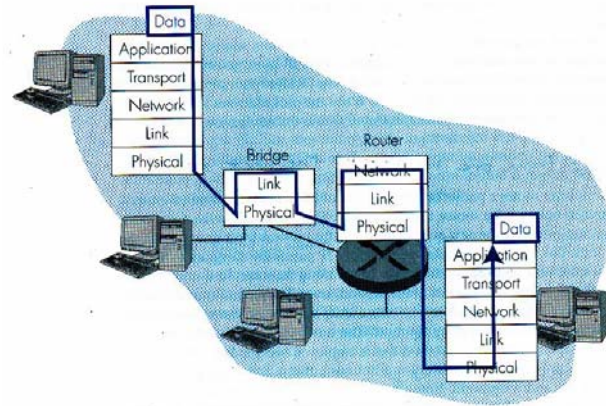
 DATAGRAM

- 4) LINK LAYER: L'IP layer chiede i servizi del link layer per muovere un datagram da un nodo al successivo. Su link diversi, si possono avere protocolli diversi!

 FRAME

- 5) PHYSICAL LAYER: sposta i singoli bit tra segmenti adiacenti della rete. I protocolli dipendono dal mezzo di trasmissione del link.

# Internet Protocol Stack 4



Non tutti gli elementi della rete implementano tutti e 5 i layer!!!