

## Riassunto livello MAC

- Come gestire accesso a mezzo condiviso?
  - Partizionando il mezzo in canali logici
    - Time Division, Code Division, Frequency Division
- Per ogni canale logico, gestisco accessi in vari modi:
  - Accesso casuale (dinamico)
    - ALOHA, S-ALOHA, CSMA, CSMA/CD
    - carrier sensing: fattibile in tecnologie cablate e non fattibile in tecnologie wireless
    - CSMA/CD usato in Ethernet
  - Protocolli a turni
    - Polling da entità centrale
    - Token passing

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

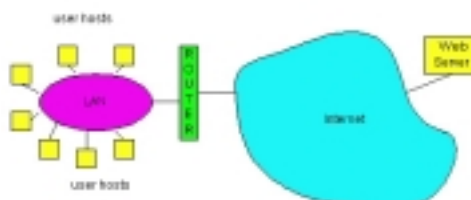
## Tecnologia di rete locale: LAN

Il livello Data Link visto finora gestisce:

- servizi, error detection/correction, accessi multipli

Ora vediamo meglio:

- Indirizzamento LAN
- Ethernet
- hubs, bridges, switches
- 802.11
- PPP
- ATM



Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## LAN Addresses and ARP

### 32-bit IP address:

- *network-layer* address
- used to get datagram to destination network (recall IP network definition)

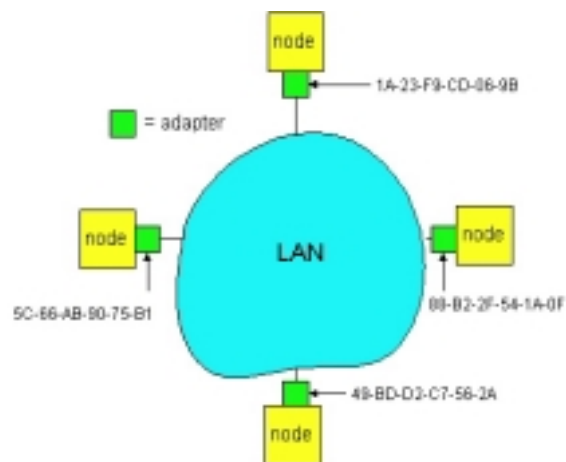
### LAN (or MAC or physical) address:

- used to get datagram from one interface to another physically-connected interface (same network)
- 48 bit MAC address (for most LANs) burned in the adapter ROM

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## LAN Addresses and ARP

Each adapter on LAN has unique LAN address



Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## LAN Address (more)

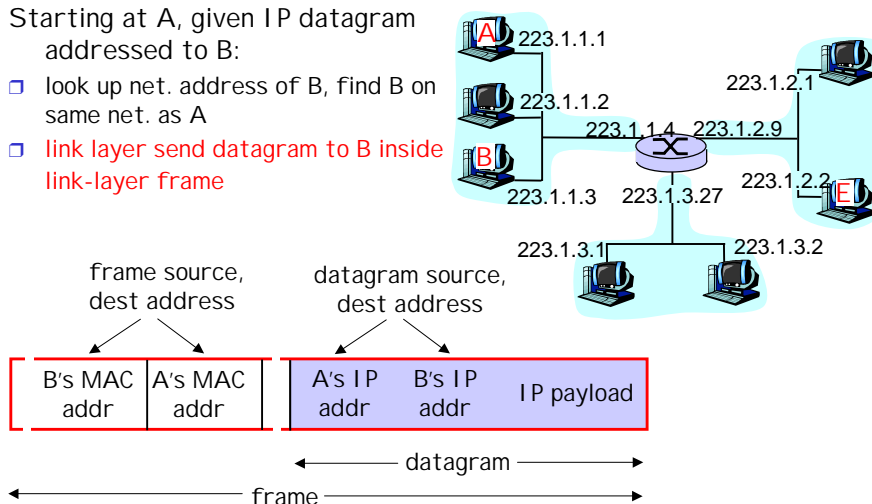
- ❑ MAC address allocation administered by IEEE
- ❑ manufacturer buys portion of MAC address space (to assure uniqueness)
- ❑ Analogy:
  - (a) MAC address: like Social Security Number
  - (b) IP address: like postal address
- ❑ MAC flat address => portability
  - can move LAN card from one LAN to another
- ❑ IP hierarchical address NOT portable
  - depends on network to which one attaches

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Recall earlier routing discussion

Starting at A, given IP datagram addressed to B:

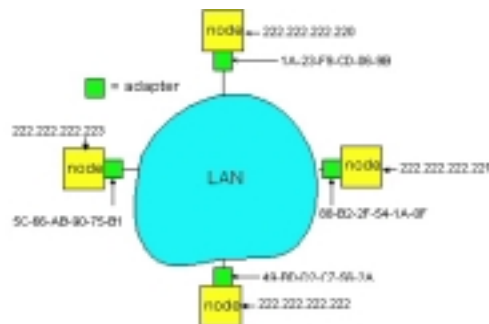
- ❑ look up net. address of B, find B on same net. as A
- ❑ link layer send datagram to B inside link-layer frame



Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## ARP: Address Resolution Protocol

Question: how to determine MAC address of B given B's IP address?



Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

- Each IP node (Host, Router) on LAN has **ARP** module, table
- ARP Table: IP/MAC address mappings for some LAN nodes
  - < IP address; MAC address; TTL >
  - < ..... >
  - TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)

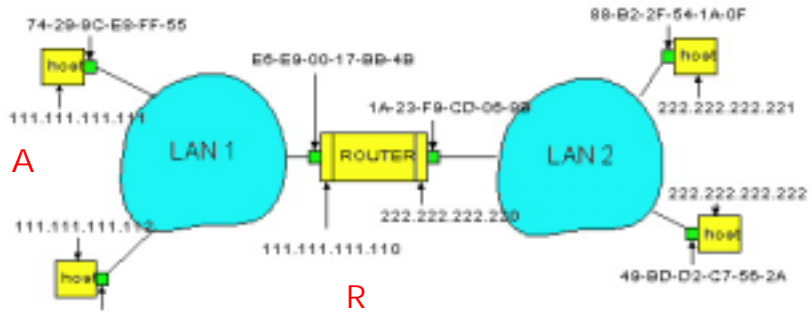
## ARP protocol

- A knows B's IP address, wants to learn physical address of B
- A **broadcasts** ARP query pkt, containing B's IP address
  - all machines on LAN receive ARP query
- B receives ARP packet, replies to A with its (B's) physical layer address
- A caches (saves) IP-to-physical address pairs until information becomes old (times out)
  - soft state: information that times out (goes away) unless refreshed

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Routing to another LAN

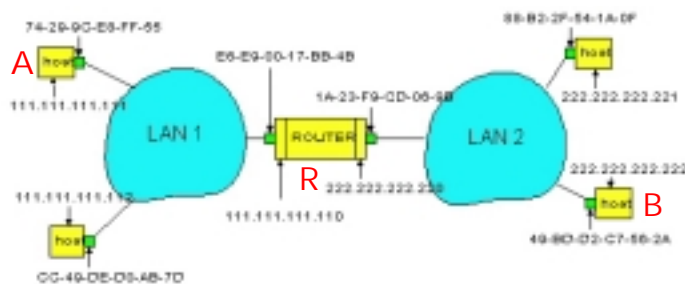
walkthrough: routing from A to B via R



- In routing table at source Host, find router 111.111.111.110
- In ARP table at source, find MAC address E6-E9-00-17-BB-4B, etc

Copyright © Luciano Bonori 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

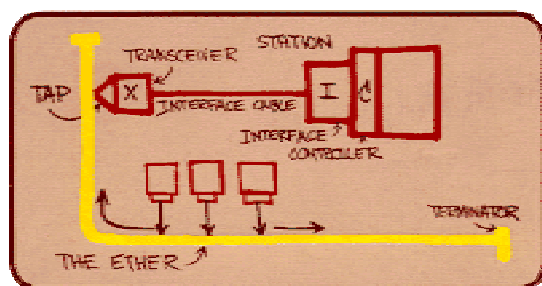
- A creates IP packet with source A, destination B
- A uses ARP to get R's physical layer address for 111.111.111.110
- A creates Ethernet frame with R's physical address as dest, Ethernet frame contains A-to-B IP datagram
- A's data link layer sends Ethernet frame
- R's data link layer receives Ethernet frame
- R removes IP datagram from Ethernet frame, sees its destined to B
- R uses ARP to get B's physical layer address
- R creates frame containing A-to-B IP datagram sends to B



## Ethernet

"dominant" LAN technology:

- ❑ cheap \$20 for 100Mbps!
- ❑ first widely used LAN technology
- ❑ Simpler, cheaper than token LANs and ATM
- ❑ Kept up with speed race: 10, 100, 1000 Mbps

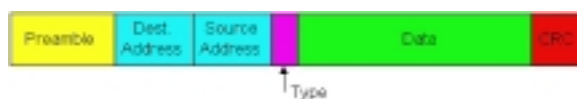


Metcalfe's Ethernet sketch

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Ethernet Frame Structure

Sending adapter encapsulates IP datagram (or other network layer protocol packet) in **Ethernet frame**



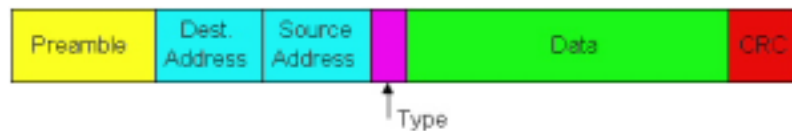
**Preamble:**

- ❑ 7 bytes with pattern 10101010 followed by one byte with pattern 10101011
- ❑ used to synchronize receiver, sender clock rates

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Ethernet Frame Structure (more)

- **Addresses:** 6 bytes, frame is received by all adapters on a LAN and dropped if address does not match
- **Type:** indicates the higher layer protocol, mostly IP but others may be supported such as Novell IPX and AppleTalk)
- **CRC:** checked at receiver, if error is detected, the frame is simply dropped



Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Ethernet: uses CSMA/CD

```
A: sense channel, if idle
  then {
    transmit and monitor the channel;
    If detect another transmission
      then {
        abort and send jam signal;
        update # collisions;
        delay as required by exponential backoff algorithm;
        goto A
      }
    else {done with the frame; set collisions to zero}
  }
else {wait until ongoing transmission is over and goto A}
```

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Ethernet's CSMA/CD (more)

**Jam Signal:** make sure all other transmitters are aware of collision; 48 bits;

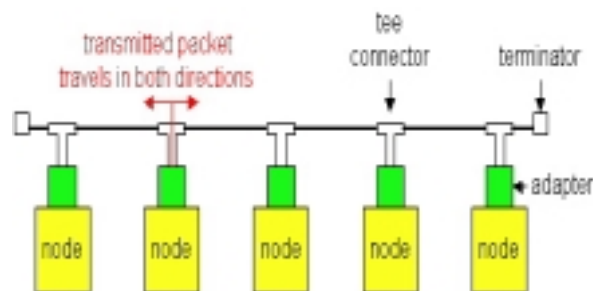
**Exponential Backoff:**

- ❑ **Goal:** adapt retransmission attempts to estimated current load
  - heavy load: random wait will be longer
- ❑ first collision: choose  $K$  from  $\{0,1\}$ ; delay is  $K \times 512$  bit transmission times
- ❑ after second collision: choose  $K$  from  $\{0,1,2,3\}$ ...
- ❑ after ten or more collisions, choose  $K$  from  $\{0,1,2,3,4,\dots,1023\}$

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Ethernet Technologies: 10Base2

- ❑ 10: 10Mbps; 2: under 200 meters max cable length
- ❑ thin coaxial cable in a bus topology



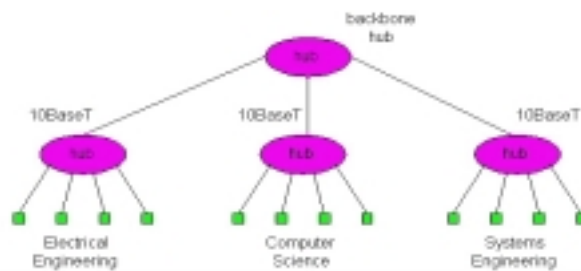
- ❑ repeaters used to connect up to multiple segments
- ❑ repeater repeats bits it hears on one interface to its other interfaces: physical layer device only!

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)



## 10BaseT and 100BaseT

- ❑ 10/100 Mbps rate; latter called “fast ethernet”
- ❑ T stands for Twisted Pair
- ❑ Hub to which nodes are connected by twisted pair, thus “star topology”
- ❑ CSMA/CD implemented at hub



Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## 10BaseT and 100BaseT (more)

- ❑ Max distance from node to Hub is 100 meters
- ❑ Hub can disconnect “jabbering adapter”
- ❑ Hub can gather monitoring information, statistics for display to LAN administrators

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

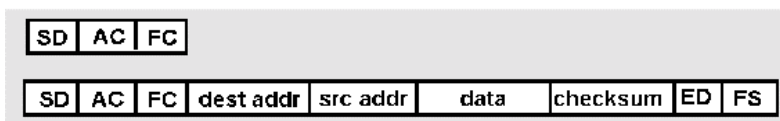
## Gbit Ethernet

- ❑ use standard Ethernet frame format
- ❑ allows for point-to-point links and shared broadcast channels
- ❑ in shared mode, CSMA/CD is used; short distances between nodes to be efficient
- ❑ uses hubs, called here "Buffered Distributors"
- ❑ Full-Duplex at 1 Gbps for point-to-point links

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Token Passing: IEEE802.5 standard

- ❑ 4 Mbps
- ❑ max token holding time: 10 ms, limiting frame length



- ❑ SD, ED mark start, end of packet
- ❑ AC: access control byte:
  - token bit: value 0 means token can be seized, value 1 means data follows FC
  - priority bits: priority of packet
  - reservation bits: station can write these bits to prevent stations with lower priority packet from seizing token after token becomes free

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Token Passing: IEEE802.5 standard



- ❑ **FC**: frame control used for monitoring and maintenance
- ❑ **source, destination address**: 48 bit physical address, as in Ethernet
- ❑ **data**: packet from network layer
- ❑ **checksum**: CRC
- ❑ **FS**: frame status: set by dest., read by sender
  - set to indicate destination up, frame copied OK from ring
  - DLC-level ACKing

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Interconnecting LANs

**Q:** Why not just one big LAN?

- ❑ Limited amount of supportable traffic: on single LAN, all stations must share bandwidth
- ❑ limited length: 802.3 specifies maximum cable length
- ❑ large "collision domain" (can collide with many stations)
- ❑ limited number of stations: 802.5 have token passing delays at each station

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Hubs

- ❑ Physical Layer devices: essentially repeaters operating at bit levels: repeat received bits on one interface to all other interfaces
- ❑ Hubs can be arranged in a **hierarchy** (or multi-tier design), with **backbone** hub at its top



Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Hubs (more)

- ❑ Each connected LAN referred to as LAN **segment**
- ❑ Hubs **do not isolate** collision domains: node may collide with any node residing at any segment in LAN
- ❑ Hub Advantages:
  - simple, inexpensive device
  - Multi-tier provides graceful degradation: portions of the LAN continue to operate if one hub malfunctions
  - extends maximum distance between node pairs (100m per Hub)

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Hub limitations

- ❑ single collision domain results in no increase in max throughput
  - multi-tier throughput same as single segment throughput
- ❑ individual LAN restrictions pose limits on number of nodes in same collision domain and on total allowed geographical coverage
- ❑ cannot connect different Ethernet types (e.g., 10BaseT and 100baseT)

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Bridges

- ❑ **Link Layer devices**: operate on Ethernet frames, examining frame header and selectively forwarding frame based on its destination
- ❑ Bridge **isolates collision** domains since it buffers frames
- ❑ When frame is to be forwarded on segment, bridge uses CSMA/CD to access segment and transmit

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Bridges (more)

- Bridge advantages:
  - Isolates collision domains resulting in higher total max throughput, and does not limit the number of nodes nor geographical coverage
  - Can connect different type Ethernet since it is a **store and forward** device
  - Transparent: no need for any change to hosts LAN adapters

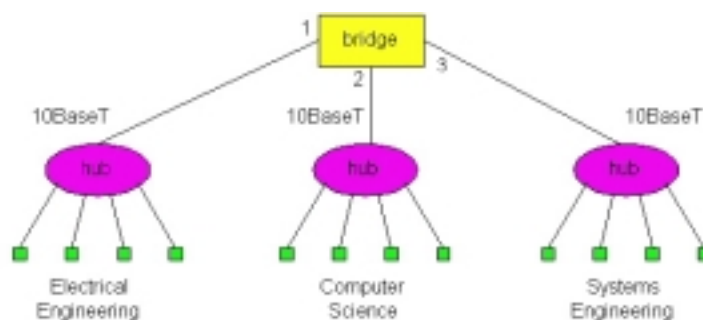
Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Bridges: frame filtering, forwarding

- bridges filter packets
  - same-LAN -segment frames not forwarded onto other LAN segments
- forwarding:
  - how to know which LAN segment on which to forward frame?
  - looks like a routing problem (more shortly!)

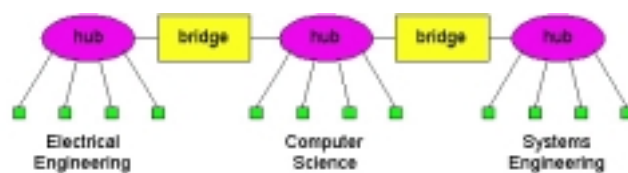
Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Backbone Bridge



Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Interconnection Without Backbone



- Not recommended for two reasons:
  - single point of failure at Computer Science hub
  - all traffic between EE and SE must path over CS segment

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Bridge Filtering

- bridges *learn* which hosts can be reached through which interfaces: maintain filtering tables
  - when frame received, bridge “learns” location of sender: incoming LAN segment
  - records sender location in filtering table
- filtering table entry:
  - (Node LAN Address, Bridge Interface, Time Stamp)
  - stale entries in Filtering Table dropped (TTL can be 60 minutes)

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Bridge Filtering

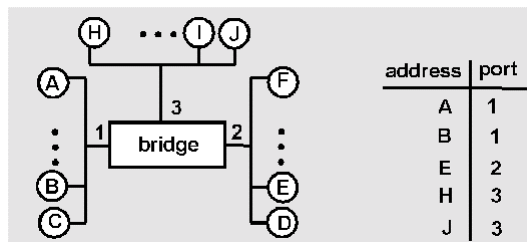
- filtering procedure:
  - if** destination is on LAN on which frame was received
    - then** drop the frame
    - else** { lookup filtering table
      - if** entry found for destination
        - then** forward the frame on interface indicated;
        - else** flood; */\* forward on all but the interface on which the frame arrived\*/*

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)



## Bridge Learning: example

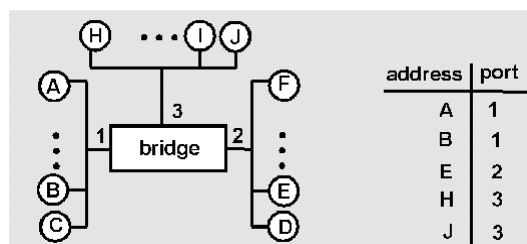
Suppose C sends frame to D and D replies back with frame to C



- C sends frame, bridge has no info about D, so floods to both LANs
  - bridge notes that C is on port 1
  - frame ignored on upper LAN
  - frame received by D

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Bridge Learning: example

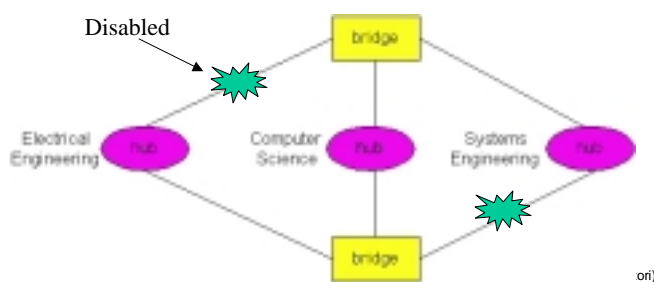


- D generates reply to C, sends
  - bridge sees frame from D
  - bridge notes that D is on interface 2
  - bridge knows C on interface 1, so *selectively* forwards frame out via interface 1

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

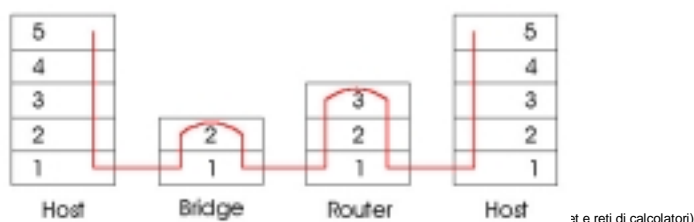
## Bridges Spanning Tree

- for increased reliability, desirable to have redundant, alternate paths from source to dest
- with multiple simultaneous paths, cycles result - bridges may multiply and forward frame forever
- solution: organize bridges in a spanning tree by disabling subset of interfaces



## Bridges vs. Routers

- both store-and-forward devices
  - routers: network layer devices (examine network layer headers)
  - bridges are Link Layer devices
- routers maintain routing tables, implement routing algorithms
- bridges maintain filtering tables, implement filtering, learning and spanning tree algorithms



## Routers vs. Bridges

### Bridges + and -

- + Bridge operation is simpler requiring less processing bandwidth
- Topologies are restricted with bridges: a spanning tree must be built to avoid cycles
- Bridges do not offer protection from broadcast storms (endless broadcasting by a host will be forwarded by a bridge)

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Routers vs. Bridges

### Routers + and -

- + arbitrary topologies can be supported, cycling is limited by TTL counters (and good routing protocols)
  - + provide firewall protection against broadcast storms
  - require IP address configuration (not plug and play)
  - require higher processing bandwidth
- bridges do well in small (few hundred hosts) while routers used in large networks (thousands of hosts)

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Protocolli Data Link: tipo punto-a-punto

- ❑ Un mittente, un destinatario e un link (non broadcast)
  - Non c'è Media Access Control
  - Non necessari indirizzi MAC
  - e.g., connessione modem, linea I SDN
- ❑ Due esempi di protocolli LLC di tipo punto-a-punto:
  - PPP (point-to-point protocol)
  - HDLC: High level data link control (Data link è il sottolivello superiore al MAC nel livello 2)

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## PPP: point to point protocol [RFC 1557]

- ❑ **packet framing**: incapsulamento di un pacchetto (datagram) di livello Rete in un frame di livello Data Link
  - Può essere usato per trasportare dati di livello rete (non solo IP)
  - Richiede abilità di demultiplexing verso i protocolli opportuni del livello rete
- ❑ **bit transparency**: deve essere consentito di trasportare qualsiasi sequenza di bit (arbitraria)
- ❑ **error detection** (non esegue nessuna error correction)
- ❑ **connection liveness**: è in grado di rilevare e segnalare la caduta del link al livello Rete
- ❑ **network layer address negotiation**: gli estremi possono apprendere e configurare i rispettivi indirizzi di Rete

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## PPP: cosa non prevede?

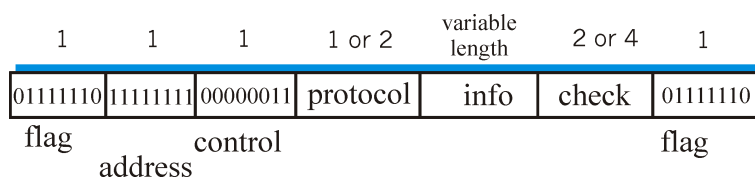
- ❑ Non esegue error correction/recovery (solo detection con CRC)
- ❑ Non esegue flow control
- ❑ Ammette consegna di frame fuori ordine
- ❑ Non supporta collegamenti multipunto (e.g., polling)

Error recovery, flow control, ordinamento  
Sono tutti aspetti lasciati ai livelli superiori

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## PPP: struttura del Frame dati

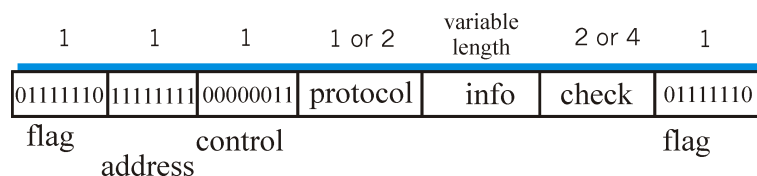
- ❑ **Flag**: delimitatore (cornice=frame)
- ❑ **Address**: è solo un campo opzionale (non usato)
- ❑ **Control**: è solo un campo opzionale (non usato)
- ❑ **Protocol**: contiene il codice del livello superiore al quale va passato il frame (eg. IP)



Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## PPP Data Frame

- **info**: sono i dati (bit) da passare al livello superiore
- **check**: cyclic redundancy check (per error detection)



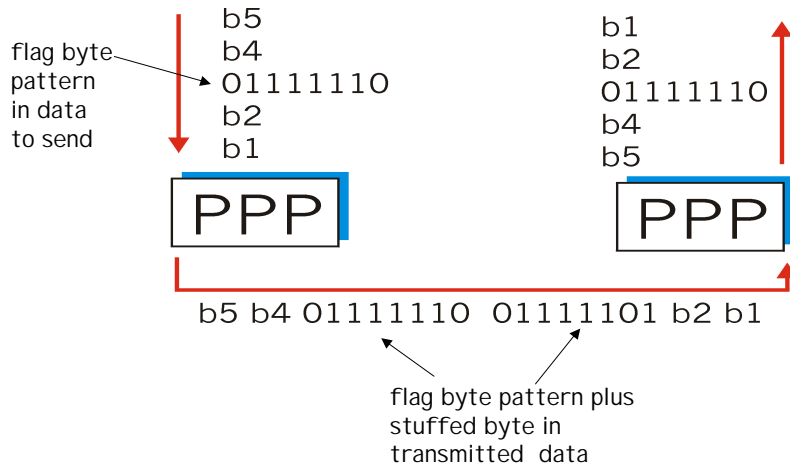
Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Trasparenza e Byte Stuffing

- Necessità di "trasparenza" dei dati: il campo dati potrebbe dover contenere il pattern <01111110>
  - **Problema** se ricevo <01111110> si tratta di dati o del delimitatore finale (flag)?
  - **Soluzione**: byte stuffing
- **Sender**: aggiunge ("stuffs") un byte extra < 01111110> dopo ogni byte < 01111110> che va *interpretato come byte di dati*
- **Receiver**:
  - Se riceve due byte 01111110 consecutivi: dimentica il primo e considera il secondo come dati
  - Se riceve solo un byte 01111110: si tratta della fine del frame (flag).

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Byte Stuffing

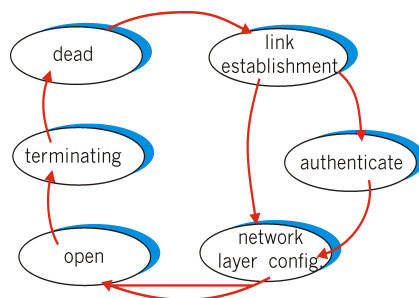


Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## PPP e il Data Control Protocol

Prima di avviare lo scambio di frame con dati del livello rete i due device devono

- **configurare il link PPP** (max. frame length, autenticazione...)
- **Configurare il livello rete**
  - Es. IP: con messaggi IP Control Protocol (IPCP=codice 8021) si configurano indirizzi IP



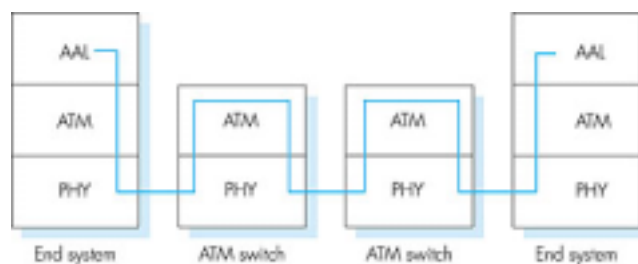
Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Asynchronous Transfer Mode: ATM

- **Livello Data link? Forse sì, forse no...**
- **1980s/1990's standard per alte prestazioni** (155Mbps fino a oltre 622 Mbps) in architetture *Broadband Integrated Service Digital Network*
- **Obiettivo: trasporto end-to-end integrato di voce, video, dati (orientato alla connessione)**
  - Soddisfa richieste di qualità del servizio (QoS) di voce, video (rispetto al servizio best-effort di Internet)
  - Telefonia a pacchetto?
  - Circuiti virtuali realizzati su canali condivisi usando commutazione di pacchetto (detti celle di dimensione costante=53 bytes)

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## ATM architecture



- **Livello adaptation (AAL): agli estremi della rete ATM**
  - Segmenta e riassume i dati delle celle
  - Rozzamente simile al livello trasporto di TCP
- **Livello ATM:** analogo al livello rete
  - Gestisce il routing e lo smistamento di celle sui virtual circuits (VCs)
- **Livello fisico (PHY)**

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)



## Livello ATM: circuiti virtuali (VC)

- **VC:** cammini per celle dal mittente al destinatario
  - call setup: fase di creazione (e negoziazione) del VC
  - Ogni cella riporta l'indirizzo del VC (e non indirizzo del destinatario)
  - *Ogni ATM switch* sul cammino di un VC mantiene lo stato della connessione
  - Ogni ATM switch gestisce e alloca le risorse del canale per garantire le prestazioni richieste (VC come canali dedicati)
- **VC permanenti (PVCs)**
  - Durano per molto tempo
  - Caso tipico: connessioni permanenti tra router IP sulle dorsali
- **VC commutati (Switched)= (SVC):**
  - Stabili al volo per connessioni dati non permanenti

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

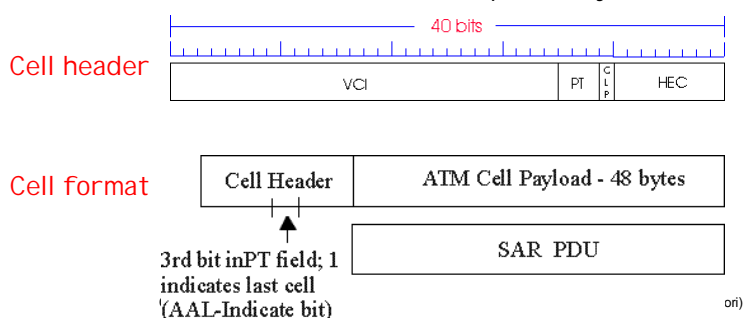
## ATM VCs

- **Vantaggi dei VC ATM:**
  - Prestazioni e QoS garantite per connessioni realizzate tramite VC (es. bitrate, ritardo, jitter)
- **Svantaggi dei VC ATM:**
  - inefficiente supporto per pacchetti IP
  - Non posso sopporre di avere un PVC per ogni coppia mittente/destinatario
  - SVC richiedono ritardo di call setup, e overhead di protocollo in ogni switch ATM che non vale la pena per brevi connessioni.

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## ATM Layer: la cella ATM

- 5-byte: ATM cell header
- 48-byte payload
  - perchè?: piccolo payload -> piccolo ritardo per "chiudere" la cella e spedire la voce
  - EU propose 32, USA propose 64 byte di payload
    - Salomonicamente si decise per 48 byte!



## ARP in reti ATM?

- Una rete ATM ha bisogno di avere indirizzi ATM
  - Come Ethernet ha bisogno di indirizzi MAC
- Traduzione di indirizzi IP in ATM: ATM ARP (Address Resolution Protocol)
  - ARP server della rete ATM manda in broadcast una richiesta ATM ARP a tutti i dispositivi ATM connessi
  - Inoltre gli host ATM possono registrarsi presso il server specificando il loro indirizzo

## Ultimi esempi: X.25 and Frame Relay

### Come ATM:

- ❑ Sono tecnologie di livello Data link concepite per reti Wide Area (WAN)
- ❑ Orientate alla connessione (circuito virtuale)
- ❑ Storicamente nate e legate alla telefonia
- ❑ Possono essere usate per trasportare pacchetti IP
  - ...e quindi, come ATM, possono essere interpretate come tecnologie di livello Data Link da parte dei protocolli di livello Rete (IP).

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## X.25

- ❑ X.25 realizza VC per ogni connessione
- ❑ **Controllo hop-by-hop lungo il cammino**
  - controllo di errore eseguito in ogni hop
    - variante del protocollo HDLC
  - Controllo di flusso eseguito in ogni hop
    - La congestione di un hop si propaga verso il mittente rallentandolo (back pressure)

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

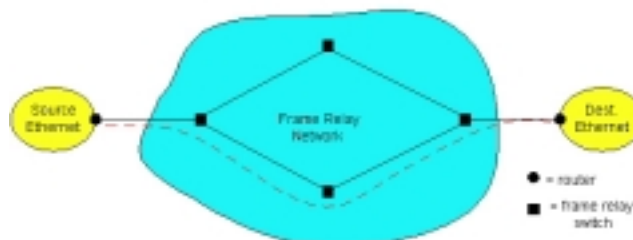
## IP versus X.25

- ❑ X.25: consegna affidabile e in sequenza end-to-end
  - “intelligenza” è sparsa nella rete
- ❑ IP: consegna inaffidabile e fuori sequenza end-to-end
  - “intelligenza” è solo sugli estremi.
- ❑ IP è la tecnologia vincente e consolidata!

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Frame Relay

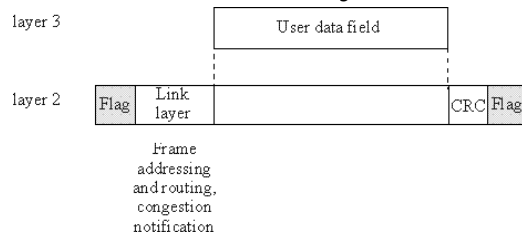
- ❑ Progettata negli anni '80 e usata negli anni '90 in USA
- ❑ Servizio frame relay:
  - Non fa controllo errore
  - Eseguce controllo della congestione end-to-end



Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Frame Relay

- Progettata per connettere LAN distanti, usa:
  - (tipicamente) circuito virtuale permanente PVC:  
"tubi" che trasportano traffico di vari utenti tra due router
  - switched VC's: come in ATM
- Gli acquirenti interessati possono affittare i servizi di connessione da Frame Relay network (eg, Sprint, ATT)

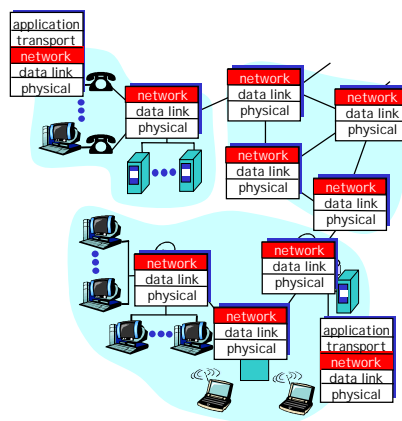


Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Livello 3: Rete

### Principi del livello 3:

- servizi:
  - Indirizzamento IPv4
  - routing e forwarding
    - Gerarchico
    - Intra-, inter-dominio
    - multicast
  - Scalabilità rete
  - IPv6, multicast
- Implementazione
  - Formato pacchetto
  - Protocolli di routing
  - Come lavora un router



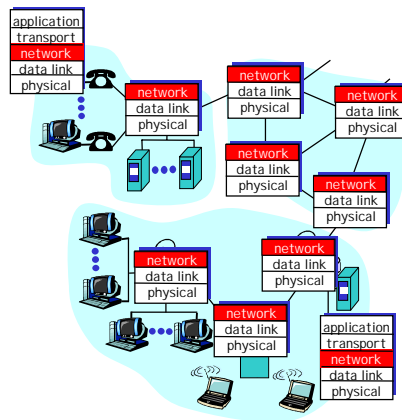
Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Funzioni del Network layer

- ❑ Trasporto pacchetti da mittente a ricevente finale
- ❑ Protocolli di livello rete implementati su **ogni host da mittente a destinatario**

### Funzioni principali:

- ❑ *Trovare cammino*: funzione degli **algoritmi di routing**
- ❑ *Instradamento*: inoltra dei pacchetti da un input a un output di un router
- ❑ *call setup*: previsto da alcune tecnologie per instaurare preventivamente il cammino mittente-destinatario (VC)



Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Modello del servizio di rete

**Domanda:** quale *modello del servizio* vale per il canale che trasporta i dati da mittente a destinatario?

- Astrazione del Servizio**
- ❑ Banda garantita?
  - ❑ Ritardo tra pacchetti costante (jitter costante)?
  - ❑ Consegna affidabile?
  - ❑ Consegna ordinata?
  - ❑ Notifica di congestione verso il mittente?

**Al livello più alto l'astrazione del servizio è:**

Circuito Virtuale (VC)  
o  
Rete a pacchetto (datagram)?

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Circuito Virtuale

“il cammino mittente-destinatario funziona come una rete telefonica”

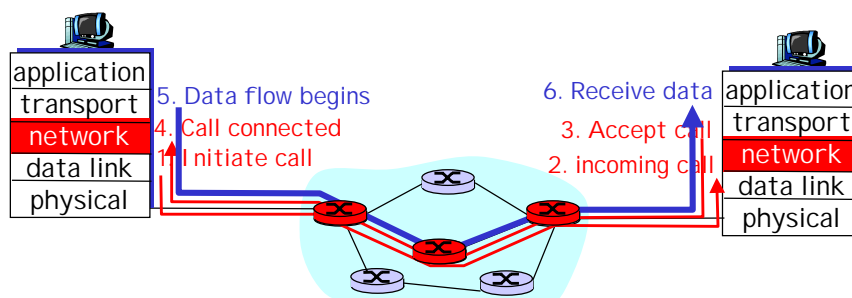
- Prestazioni minime garantite
- configurazione preventiva tra mittente e destinatario

- “call setup”, e rilascio *prima che i dati siano trasmessi*
- Ogni pacchetto riporta identificatore del VC e non del destinatario
- *Ogni router lungo il cammino del VC mantiene lo stato di tutte le connessioni passanti*
  - Diverso rispetto al livello trasporto, curato ai soli estremi
- Le risorse del collegamento e del router sono allocate e amministrare preventivamente
  - Prestazioni: basate su banda, buffer e link riservati

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Circuiti virtuali: protocolli di segnalazione

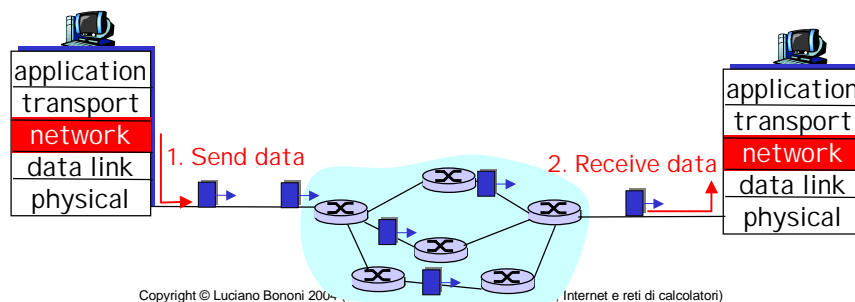
- Usati per instaurare, mantenere e chiudere VC
  - Es. Tecnologie ATM, frame-relay, X.25
- ...MA non usati in Internet!
  - Qualcosa di simile usato in Internet2



Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Reti a pacchetto:

- il modello di servizio del livello rete di Internet
- Non c'è "call setup"
- routers: non mantengono stato delle connessioni end-to-end
  - Non esiste il concetto di "connessione di livello rete"
- Ogni pacchetto è inoltrato usando indirizzo destinatario
  - Pacchetti diversi tra mittente-destinatario possono seguire strade diverse



## Perchè 2 modelli di servizio a livello rete?

### Internet

- Trasferimento dati tra PC
  - Servizio "elastico", non costante
  - Possono adattarsi ai problemi
  - Controllo e correzione dell'errore
  - Modello semplice NELLA rete, complicato agli estremi
- Esistono tanti tipi di collegamento
  - Caratteristiche diverse
  - Difficile rendere servizio uniforme

### ATM

- Evoluzione da telefonia
- Trasferimento dati tra "umani"
  - Poco tolleranti ai problemi
  - Interattività e affidabilità
  - Servizi garantiti
- La rete agli estremi è banale
  - La complessità è tutta interna alla rete

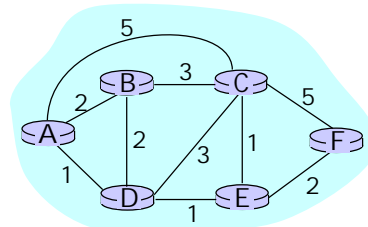
Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)



## I I Routing

### Protocollo di Routing

**obiettivo:** trovare un buon cammino tra i router della rete tra mittente e destinatario



Rete astratta: un grafo

- Nodi=router
- Archi= collegamenti
  - Costo del collegamento: ritardo, congestione, costo economico
- Cos'è un "buon" cammino?:
  - Cammino di costo minimo
  - quale è il costo rilevante?

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Classificazione algoritmi di Routing

### Basati su informazioni

#### Globali

- Tutti i router hanno tutte le info sui cammini della rete (topologia e costi)

- ...detti algoritmi "link state"

#### Distribuite (decentralizzate)

- Ogni router conosce solo i suoi router vicini e i collegamenti verso di essi
- Processi iterativi di calcolo dei cammini, scambio di info tra vicini
- ...algoritmi "distance vector"

### Statici o dinamici?

#### Statici:

- I cammini sono fissi o cambiano lentamente nel tempo

#### Dinamici:

- Cammini cambiano in fretta
  - Aggiornamenti periodici
  - Es. Reti wireless

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Es. algoritmo Link-State: Dijkstra

### Algoritmo di Dijkstra

- Topologia e costi dei link conosciuti a tutti i nodi
- Ottenuta attraverso broadcast "link state"
  - Tutti i nodi hanno stessa informazione
  - Calcola cammini di costo minimo verso ogni destinatario
    - ottenendo tabella di routing per ogni nodo
  - Metodo iterativo: dopo k passi ha ottenuto cammino minimo verso k destinazioni

### Notazione usata:

- $c(i,j)$ : costo cammino tra nodo  $i$  e nodo  $j$  (infinito se non sono collegati)
- $D(v)$ : valore attuale del cammino minimo da sorgente a nodo  $v$
- $p(v)$ : nodo precedente a  $v$  lungo il cammino da sorgente a  $v$
- $N$ : insieme di nodi il cui cammino minimo è stato individuato

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

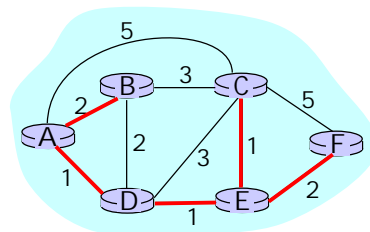
## Dijkstra's Algorithm

- 1 **Inizializzazione:** (quando eseguito su un nodo A)
- 2  $N = \{A=(mittente)\}$
- 3 Per ogni nodo  $v$
- 4 se  $v$  è adiacente ad A
- 5 Allora poni  $D(v) = c(A,v)$
- 6 altrimenti  $D(v) = \text{infinito}$
- 7
- 8 **Ciclo da eseguire finchè  $N$  non è completo**
- 9 trova  $w$  non in  $N$  tale che  $D(w)$  è minimo
- 10 aggiungi  $w$  all'insieme  $N$
- 11 aggiorna  $D(v)$  per ogni  $v$  non in  $N$ , e adiacente a  $w$ :
- 12  $D(v) = \min( D(v), D(w) + c(w,v) )$
- 13 /\* il nuovo costo da A a  $v$  è: il costo precedente, oppure
- 14 Il costo minore da A a  $w$  più il costo da  $w$  a  $v$ \*/
- 15 **Ripeti finchè  $N$  non contiene tutti i nodi della rete**

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Esempio: algoritmo di Dijkstra

Step	start N	D(B),p(B)	D(C),p(C)	D(D),p(D)	D(E),p(E)	D(F),p(F)
→0	A	2,A	5,A	1,A	infinito	infinito
→1	AD	2,A	4,D		2,D	infinito
→2	ADE	2,A	3,E			4,E
→3	ADEB		3,E			4,E
→4	ADEBC					4,E
5	ADEBCF					



Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Algoritmo di routing: Distance Vector

### iterativo:

- Continua finchè nessun nodo scambia informazione
- *Termina da solo!*

### asincrono

- Non si richiede scambio di info in passi fissi e sincroni

### Distribuito:

- Ogni nodo comunica solo con i suoi vicini

### Tabella delle Distanze

- Ogni nodo ha una riga verso ogni possibile destinazione, e una colonna per ogni nodo adiacente.
- riporta quale sia il costo del cammino verso la destinazione passando per il prossimo destinatario (nodo adiacente)
- Es.: nel nodo X, per la destinazione Y, attraverso il vicino Z

$$\begin{aligned}
 D^X(Y,Z) &= \text{distanza da X a } Y, \text{ passando per } Z \\
 &= c(X,Z) + \min_w \{D^Z(Y,w)\}
 \end{aligned}$$

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Distance Vector Routing

### Iterativo e asincrono:

Ogni modifica locale è scatenata da

- Modifica costo link
- Notifica di modifica costo link di un vicino

### distribuito

- Ogni nodo comunica ai vicini solo quando si cambia il cammino minimo verso un destinatario
  - I vicini propagano il messaggio a loro volta (effetto domino)

### Ogni nodo:

*attende* notifica di una modifica del costo di un link da parte dei nodi vicini

*ricalcola* la tabella delle distanze

Se si sono avute modifiche nella tabella locale, *notifica* tali modifiche ai suoi vicini

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Confronto tra algoritmi LS and DV

### Complessità dei messaggi

- LS: con  $n$  nodi,  $E$  link,  $O(nE)$  messaggi spediti da ogni nodo
- DV: scambio solo tra vicini

### Velocità di aggiornamento

- LS: algoritmo  $O(n^2)$  richiede  $O(nE)$  messaggi
  - Può oscillare
- DV: i tempi dipendono
  - Possono dare cicli

**Robustezza:** cosa succede se un router si rompe?

### LS:

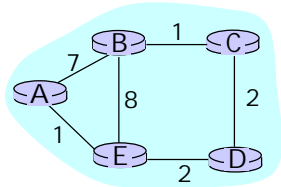
- Se Router diffonde costi di link sbagliati: Ogni nodo calcola solo la propria tabella di routing che rimane corretta in gran parte

### DV:

- Se router diffonde costo dei cammini sbagliati
- Errore si propaga a tutti i nodi della rete

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Tabella delle distanze: esempio



$$D^E(C,D) = c(E,D) + \min_w \{D^D(C,w)\} = 2+2 = 4$$

$$D^E(A,D) = c(E,D) + \min_w \{D^D(A,w)\} = 2+3 = 5 \text{ loop!}$$

$$D^E(A,B) = c(E,B) + \min_w \{D^B(A,w)\} = 8+6 = 14 \text{ loop!}$$

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

Costo verso destinazione via

$D^E()$	A	B	D
A	1	14	5
B	7	8	5
C	6	9	4
D	4	11	2

destinazione

## Distance table gives routing table

Costo verso la destinazione via

$D^E()$	A	B	D
A	1	14	5
B	7	8	5
C	6	9	4
D	4	11	2

destinazione

Prossimo router, costo

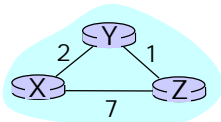
A	A,1
B	D,5
C	D,4
D	D,4

destinazione

Tabella delle distanze → Tabella di instradamento

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Distance Vector: esempio



		cost via	
		Y	Z
D <sup>X</sup> d e s t	Y	2	∞
	Z	∞	7

		cost via	
		X	Z
D <sup>Y</sup> d e s t	X	2	∞
	Z	∞	1

		cost via	
		X	Y
D <sup>Z</sup> d e s t	X	7	∞
	Y	∞	1

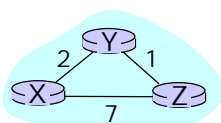
		cost via	
		Y	Z
D <sup>X</sup> d e s t	Y	2	8
	Z	3	7

		cost via	
		X	Z
D <sup>Y</sup> d e s t	X	2	8
	Z	9	1

		cost via	
		X	Y
D <sup>Z</sup> d e s t	X	7	3
	Y	9	1

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Distance Vector Algorithm: example



		cost via	
		Y	Z
D <sup>X</sup> d e s t	Y	2	∞
	Z	∞	7

		cost via	
		X	Z
D <sup>Y</sup> d e s t	X	2	∞
	Z	∞	1

		cost via	
		X	Y
D <sup>Z</sup> d e s t	X	7	∞
	Y	∞	1

$$D^X(Y,Z) = c(X,Z) + \min_w \{D^Z(Y,w)\}$$

$$= 7 + 1 = 8$$

$$D^X(Z,Y) = c(X,Y) + \min_w \{D^Y(Z,w)\}$$

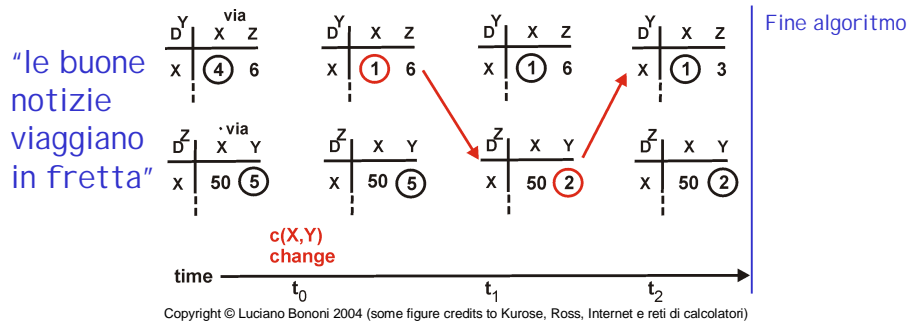
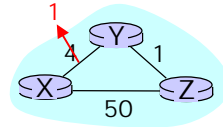
$$= 2 + 1 = 3$$

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Distance Vector: se cambia il costo di un link

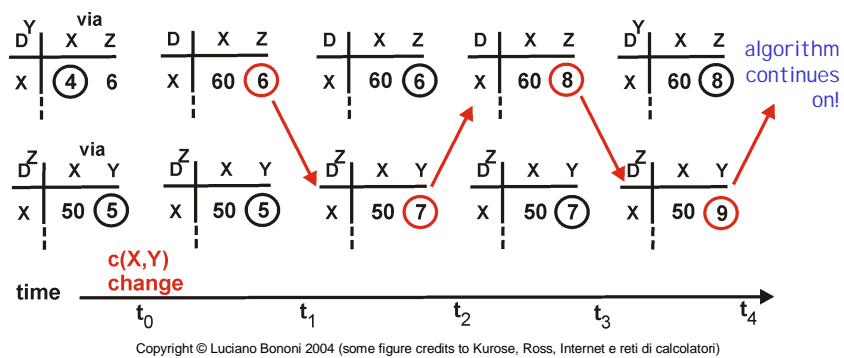
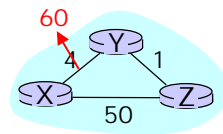
### Se cambia il costo di un link

- Il nodo nota il cambiamento
- Aggiorna la tabella delle distanze
- Se ciò determina una modifica di un costo minimo di un cammino, allora notifica ai nodi vicini il cambiamento



## Distance Vector: se cambia il costo di un link

- Le cattive notizie viaggiano lentamente



## Routing Gerarchico: motivazioni

Idealmente, nel routing

- Tutti i router sono identici
- Tutti i router usano stesso algoritmo
- La rete è "piatta"

*Ma ciò non è vero in realtà!*

**Problema di scala:** con  
200 milioni di IP:

- Non posso inserire tutti in una routing table
- Non posso scambiare routing table così grandi coi vicini

**Autonomia amministrativa**

- internet = rete di reti
- Ogni amministratore di rete vuole controllare le politiche di routing

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)

## Routing Gerarchico

- Si aggregano gruppi di router in regioni dette, "autonomous systems" (AS)
- I router di un AS eseguono lo stesso protocollo
  - Protocollo "intra-AS" routing
- Routers di diversi AS possono eseguire diversi protocolli di routing intra-AS

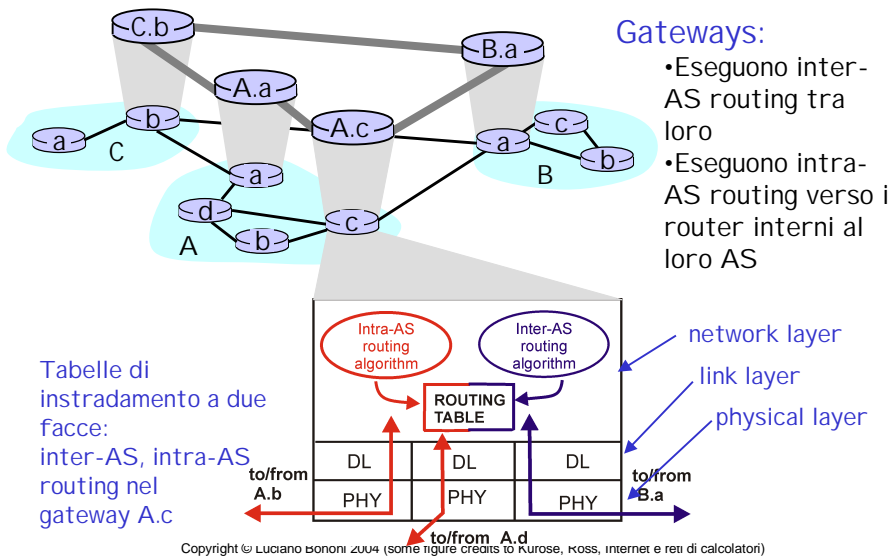
gateway routers

- Router speciali negli AS
- Fanno da tramite tra AS e AS
  - eseguono il protocollo di routing intra-AS con tutti gli altri router intra-AS
  - *Inoltre:* eseguono il routing verso altri AS esterni, mediante protocolli di routing *inter-AS routing*, verso altri gateway routers

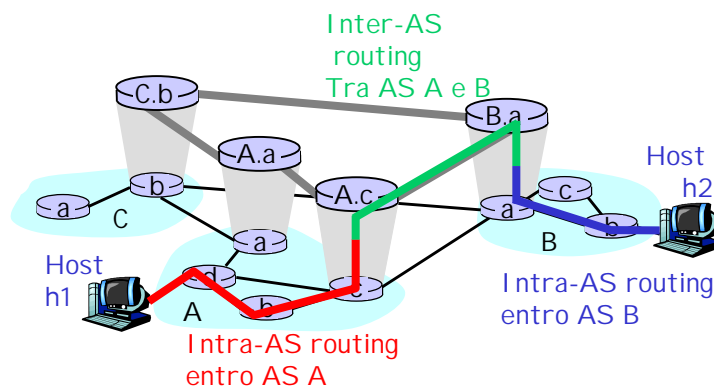
Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)



## Intra-AS and Inter-AS routing



## Intra-AS and Inter-AS routing



- Vedremo esempi di protocolli di routing inter-AS e intra-AS in seguito

Copyright © Luciano Bononi 2004 (some figure credits to Kurose, Ross, Internet e reti di calcolatori)