

# Come se ne esce?

- ◆ Una delle due cose andrà sacrificata...
- ◆ Visto che alla banda non possiamo rinunciare → essenzialmente serve una rete che abbia lunghezza massima limitata



Passiamo ora ai supereroi...



# Il problema delle reti...



- ◆ ... è farle grandi!
- ◆ Ma ce ne sono molte, e diverse, ed anche reti dello stesso tipo hanno limiti...
- ◆ Ecco quindi che arrivano i supereroi alla riscossa, salvando la situazione

# Colla per reti!

Application layer

Application gateway

Transport layer

Transport gateway

Network layer

Router

Data link layer

Bridge, switch

Physical layer

Repeater, hub



# Repeaters e Hubs

- ◆ Il **Ripetitore**, come già visto, ripete (tipicamente, ***amplificandolo*** in potenza) il segnale.
- ◆ Lo **Hub** (che può essere anche repeater) è l'equivalente di una connessione fisica, cioè propaga il segnale da una porta a tutte le altre





# Il Bridge/Switch

- ◆ E' un dispositivo di livello più alto (strato **data link**), e quindi interagisce con la struttura dei frame, cosa che repeaters e hubs non fanno



# Il bridge/switch



- ◆ Crea una rete più grande ma a livello logico (data link) e non fisico
- ◆ In altre parole, non abbiamo più i limiti di ***grandezza*** della rete, ed abbiamo ***domini di collisione*** diversi
- ◆ Questo si ottiene ispezionando i pacchetti, e filtrando il traffico

# Learning



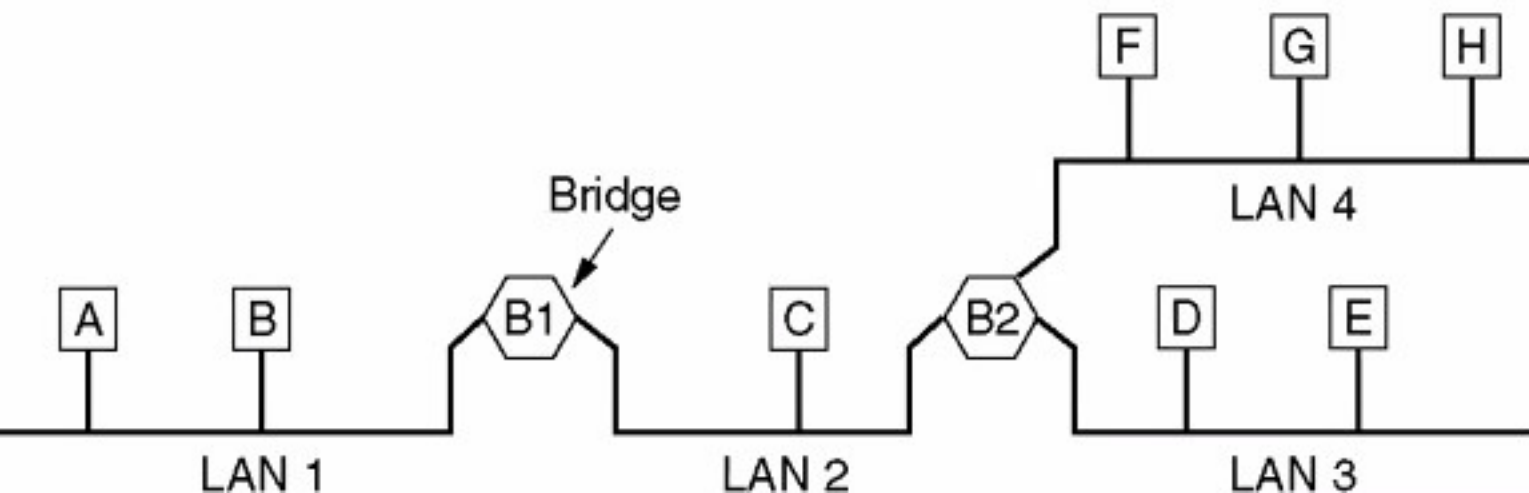
- ◆ C'è quindi una fase in cui il bridge/switch impara la configurazione di rete, e gestisce il traffico corrispondentemente
- ◆ Essendo a livello data link, controlla mittente e destinazione dei frame (→ tipicamente, i **MAC address**)



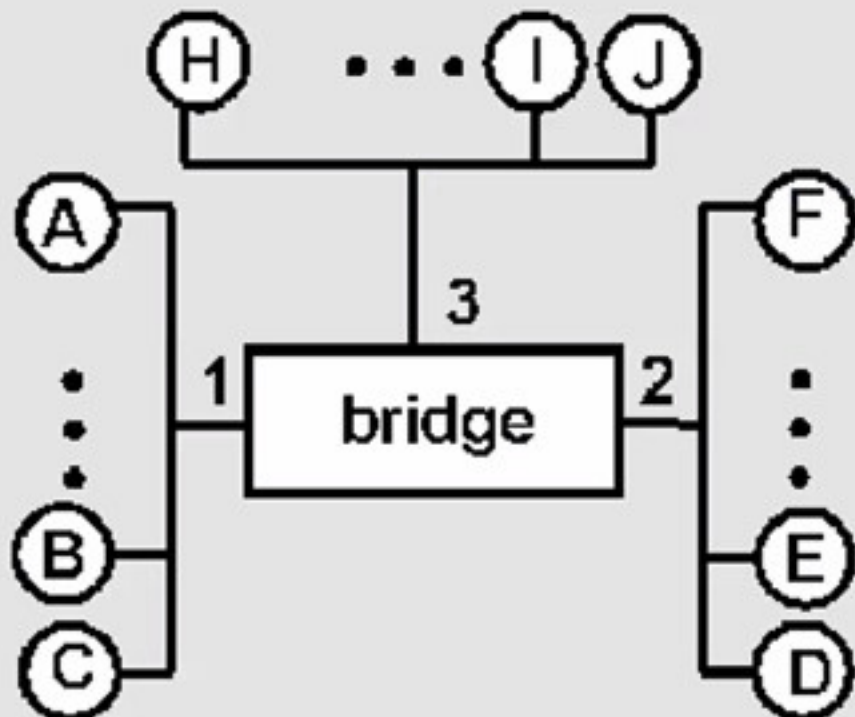
# Hash tables



- ◆ Si usa il backward learning: le hash table vengono costruite analizzando il flusso di dati e risposte, e si fa ***broadcasting*** nella fase intermedia



# Esempio



address	port
A	1
B	1
E	2
H	3
J	3

# Timeouts...



- ◆ C'è un timeout di ***fading*** per ovviare ai cambiamenti nella topologia di rete


# Situazione simile...

- ◆ Ai motori di ricerca... pensate a come funziona ad esempio Google!





## Ora andiamo più in alto...

Application layer	Application gateway
Transport layer	Transport gateway
Network layer	Router
 Data link layer	Bridge, switch
Physical layer	Repeater, hub

# Percorso migliore?

- ◆ Ci serve un concetto di distanza
- ◆ Una prima approssimazione è contare il numero di “**hops**” (balzi), cioè di stazioni incontrate nel cammino



GASTRONOMIE  
THEATERPLATZ  
GMBH

# Il flooding



- ◆ Il flooding ("alluvione") è un mezzo potentissimo per fare il routing
- ◆ L'idea: ogni pacchetto viene ***ritrasmesso a tutte le linee*** (tranne quella da cui è arrivato)

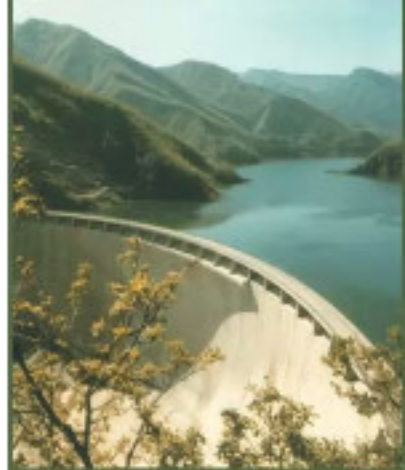


# Il flooding: il male



- ◆ Cominciamo dai lati negativi: ovviamente la rete col flooding semplice verrebbe sommersa
- ◆ Occorrono quindi dei metodi per il “controllo delle acque” per così dire

# Controllo del flooding



- ◆ Una tecnica è l'***hop counting***: si associa un numero massimo di hop (→ un'età massima) ad ogni pacchetto, dopo i quali il pacchetto muore

# Controllo del flooding



- ◆ Un'altra tecnica alternativa è il ***tracking*** : tenere traccia dei pacchetti che sono stati già trasmessi, e non ritrasmetterli
- ◆ Invece di tenere tutta la lista dei pacchetti, si possono tenere liste separate per ogni router, tenendo un contatore speciale per la lista dei pacchetti **O-N** che sono stati già ricevuti

# Il flooding: lati negativi



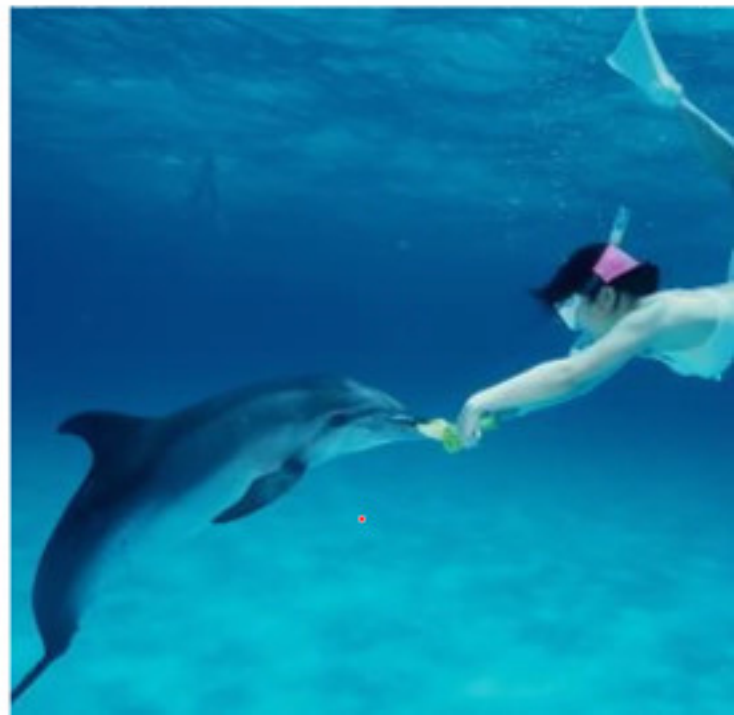


# Lati positivi?

- ◆ A prima vista il flooding può sembrare una tecnica grossolana e di dubbia utilità
- ◆ Finchè non se ne comprendono i vantaggi

# Vantaggi del flooding

- ◆ Il flooding ***sceglie sempre la via migliore (!)***



# Vantaggi del flooding

- ◆ Ed infine, uno dei più grandi vantaggi del flooding: è robustissimo rispetto alle modifiche della rete
- ◆ In realtà, è facile dimostrare che è *il più robusto* sistema di routing possibile, cioè, nessun altro sistema è migliore del flooding in questo senso

## Quindi...

- ◆ ... utilissimo in tutti quei casi quando il carico di rete non è molto alto, ma o c'è topologia di rete estremamente variabile, o è critico che un messaggio arrivi sempre nel minor tempo possibile...

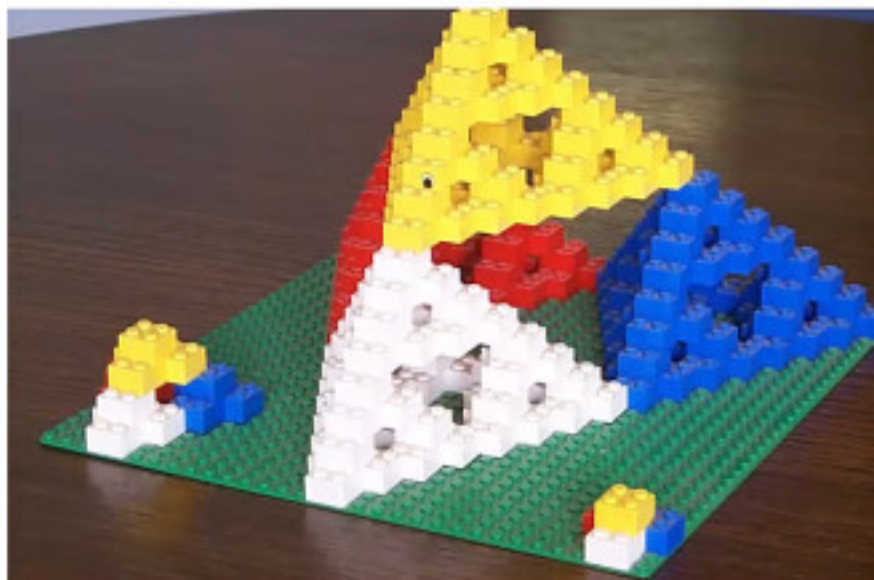
Ad esempio, ambito militare...!





# Oltre il flooding?

- ◆ Gli altri sistemi si basano tutti sull'idea che si raccolga informazione globale assommando varie informazioni locali, come i mattoncini lego



# Distance vector routing



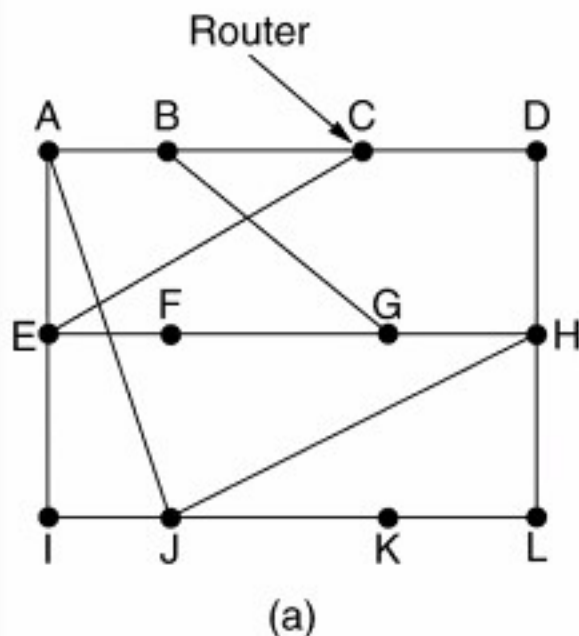
- ◆ Era il routing usato dalla prima versione di Internet (**ARPANET**)
- ◆ L'idea è che ogni router ha una tabella di routing che contiene informazioni su quanto veloce è la connessione ad un altro router, e qual è la via migliore (tra i primi vicini) per raggiungerlo

# Distance value routing



- ◆ Funziona in modo molto semplice:
- ◆ Ogni router chiede ai suoi vicini la loro tabella
- ◆ Usa poi le loro tabelle, ed il tempo che c'è voluto per averle, per costruire la sua tabella selezionando i percorsi migliori

## Distance Vector Routing: esempio



New estimated  
delay from J

To	A	I	H	K
A	0	24	20	21
B	12	36	31	28
C	25	18	19	36
D	40	27	8	24
E	14	7	30	22
F	23	20	19	40
G	18	31	6	31
H	17	20	0	19
I	21	0	14	22
J	9	11	7	10
K	24	22	22	0
L	29	33	9	9

	Line
8	A
20	A
28	I
20	H
17	I
30	I
18	H
12	H
10	I
0	–
6	K
15	K

New routing table for J

JA delay is 8

JI delay is 10

JH delay is 12

JK delay is 6

Vectors received from J's four neighbors



# Pro e contro Distance Vector Routing



◆ Pro: è veloce a recepire le "buone notizie"





# Contro...



- ◆ Tanto questo routing si comporta bene rispetto alle "***buone notizie***" quanto, dualmente, si comporta male con le "***cattive notizie***"

# Routing fase 2: il Link State Routing

- ◆ Per questi problemi, il routing su Internet, dopo il **1979**, è stato sostituito da un altro algoritmo, il cosiddetto ***link state routing***



# Link State Routing



- ◆ Quando ha informazione completa sui suoi vicini, ogni nodo costruisce un pacchetto che contiene tutta questa informazione, più altra informazione che vedremo fra poco...
- ◆ ... e la manda a tutti gli altri (***broadcast***)

# Link State Routing



- ◆ L'idea è quindi che ogni nodo riceverà i mattoncini lego corrispondenti alle informazioni locali di ***ogni altro nodo***
- ◆ In tal modo, potrà ricostruire una mappa completa della rete, e quindi calcolare i percorsi migliori

# Link State Routing (cont.)



◆ Restano da sistemare varie cosette



# Link State Routing

- ◆ Notare la differenza con il Distance Vector Routing:
- ◆ Stiamo sprecando più banda (per via del broadcasting), però, proprio come nel caso del flooding, quest'uso aggiuntivo della banda (a livello **globale**) ci permette di essere molto più robusti (evitiamo il problema della **località** presente nell'altro algoritmo)



# Visione più alta: Quality of Service

- ◆ La qualità del servizio (**QoS**) è una serie di parametri che dettano la qualità del servizio offerto
- ◆ Nell'ambito delle reti, tipicamente la QoS è data da  
***4 parametri principali***

