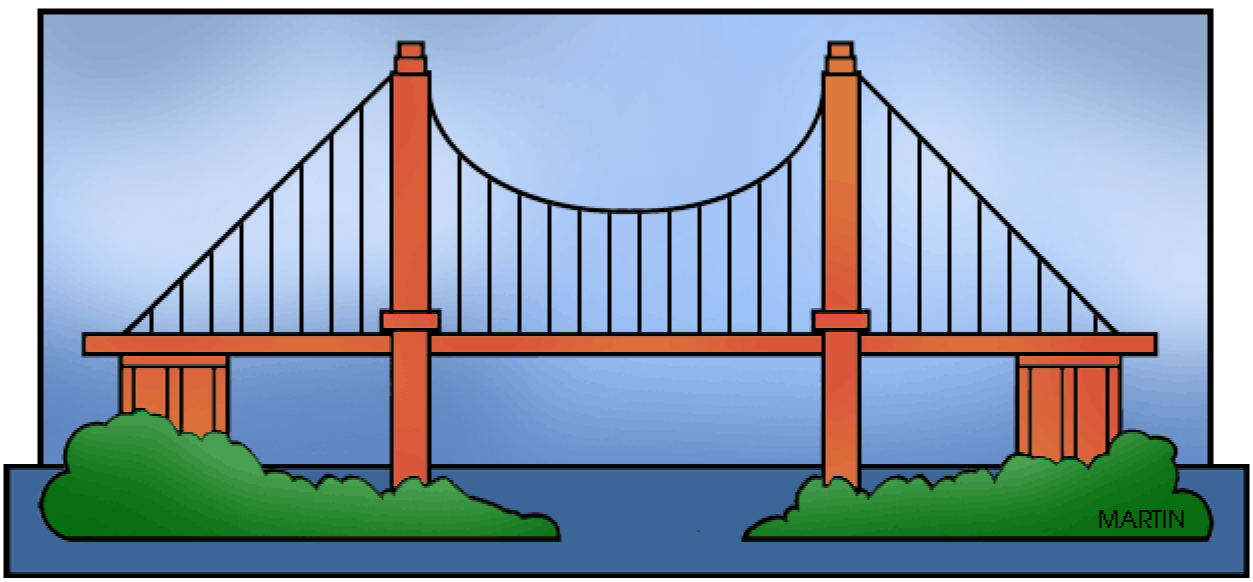
CS 118: Bridging

Interconnecting Data Links without routers



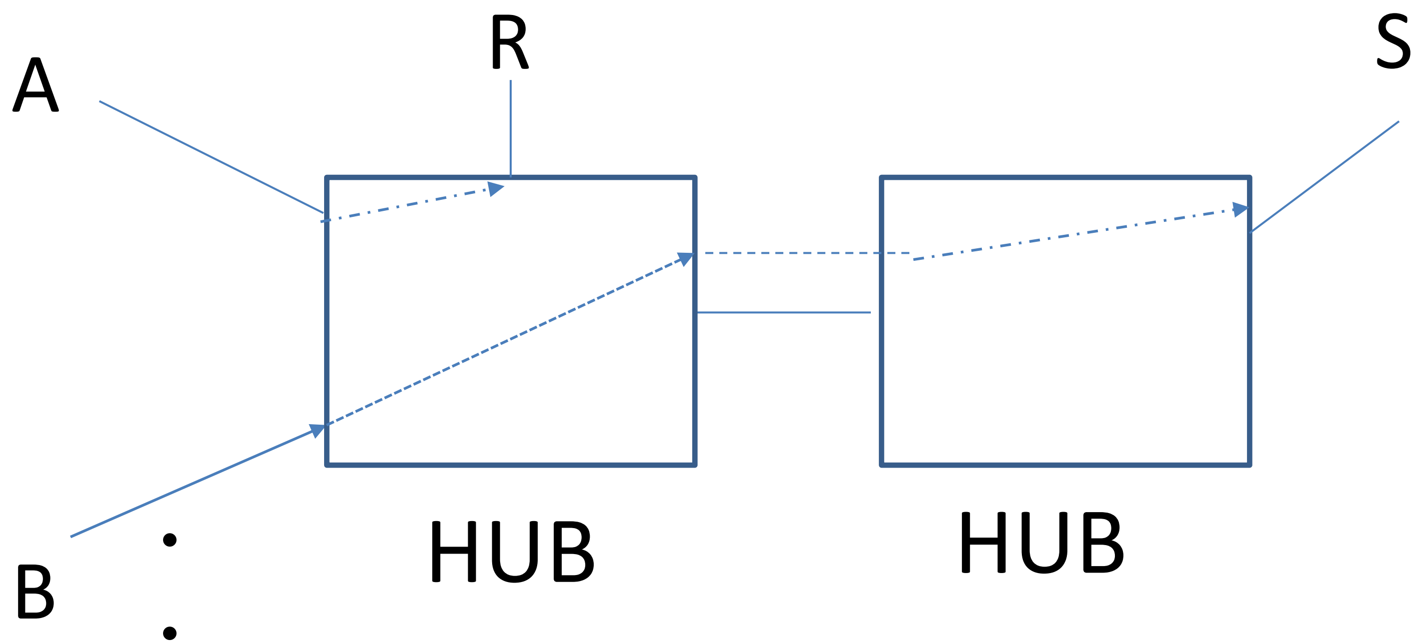
phillipmartin.com

ETHERNET 1 ETHERNET 2

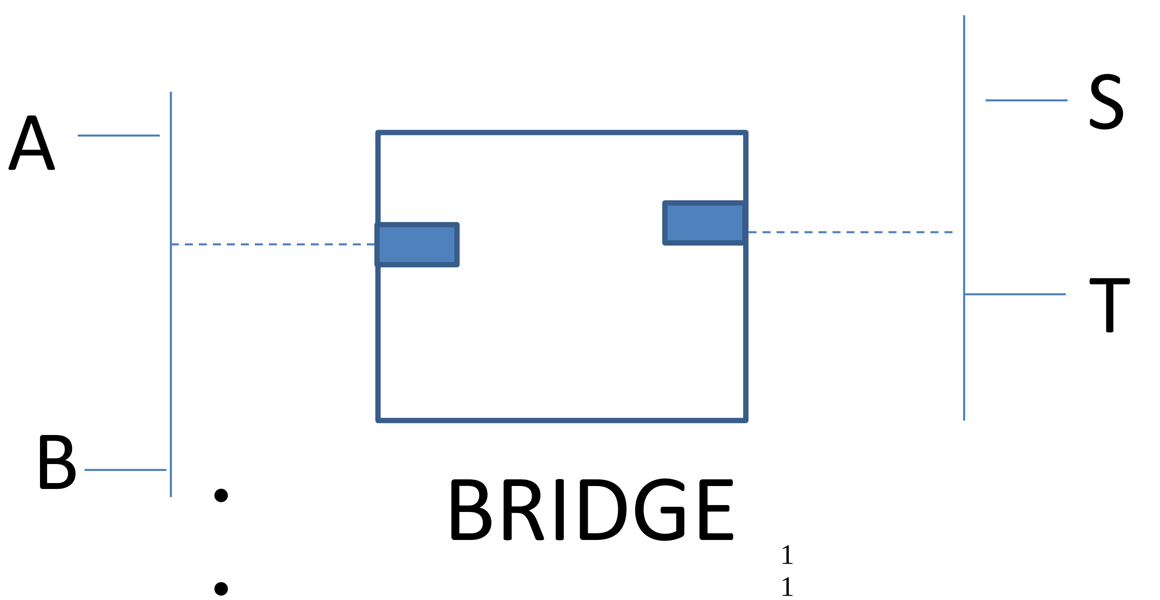
Plan of Lecture

- Review what Gigabit Ethernet is and 802.11
- Understand the structure of Ethernet addresses (srill used today when you have a MAC address)
- Then move on to how to interconnect Ethernets at the Data Link level (same as switches in Gigabit Ethernet). Not a router. Something in between! What is it?

From hubs to bridges



If A talks to R at same time as B talks to S there is a collision. True for 10 and 100 M Ethernet.

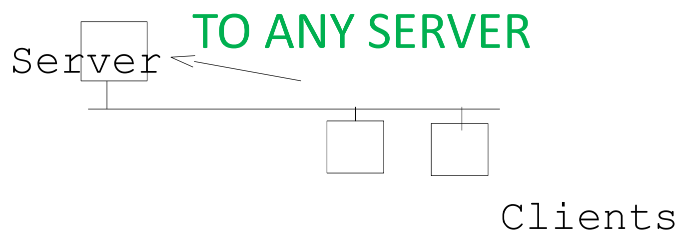


A can talk to B at same time as S talks to T with no collision. Bridge buffers and allows A to talk to S after contending on two Ethernets.

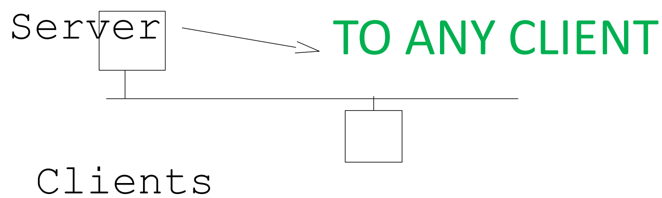
WHY MULTICAST : Autoconfiguration,Efficiency

High-order bit of DA = 1 for multicast

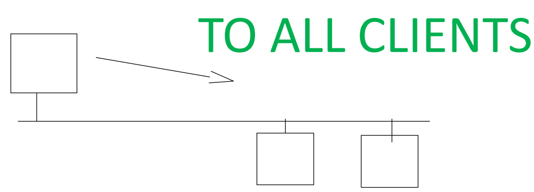
1) SOLICITATION



2)ADVERTISEMENT



3) FREE COPIES (a drug!)



Multicast generalizes broadcasting (sending to all) by sending to a *subset* of stations

Multicast Addresses

- 6 byte Ethernet Addresses assigned by 802 committee. Vendors buy a fixed 3 byte code. They can then assign remaining 3 bytes (for Dest Addresses) or 2 bytes (for 5-byte type fields). Can buy more codes.
- Multicast address denoted by Most Significant Bit. Get 2^{24} multicast and unicast address and 2^{16} types/block.
- Common multicast addresses (i.e., all IP endnodes, all IP routers) and type fields are standardized.
- Broadcast address is all 1's. Multicast better. Ethernet hardware should only pass up to software packets with DA = My address or a multicast that station listens to. Hashing or CAMs. .

Local Area Networks Review

- Till last lecture, all the physical links we studied were a single wire with one sender and one receiver. A local area network (LAN) is like a shared wire with multiple senders and multiple receivers.
- Common LANs are Ethernet and token ring. In Ethernet, stations transmit when they want to; if two or more transmit at same time, there is a collision and they retransmit. Token rings when station has token
- Ethernets were successful but had limitations. Hence the need for bridging to “extend” LANs.
- Bridges are different from hubs (they are more akin to switches) because they are data link relays, they store and buffer frames and relay frames, not bits. First a small detour into “multicasting”.

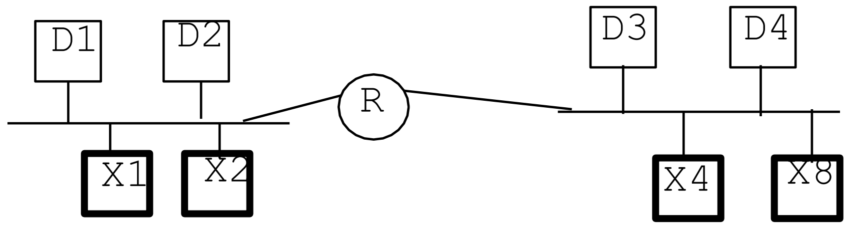
Bridging:an Exercise in Invention

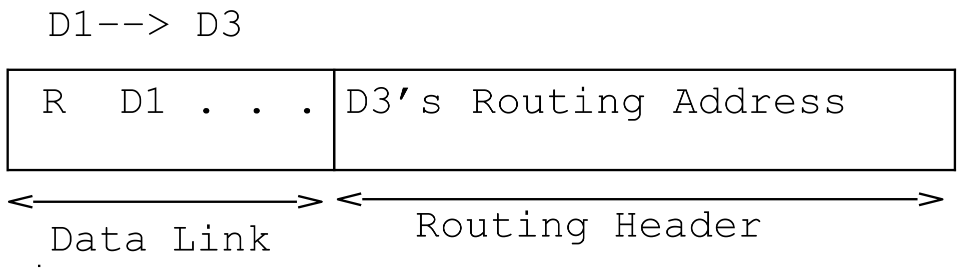
- Problem Definition
- Initial Solution
- Refinements for Efficiency and Correctness.
- Generalization.
- Realization.

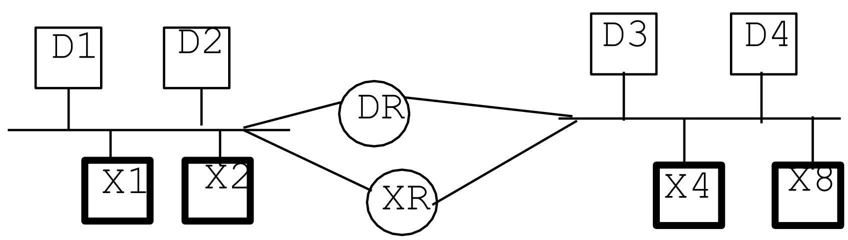
Problem Statement

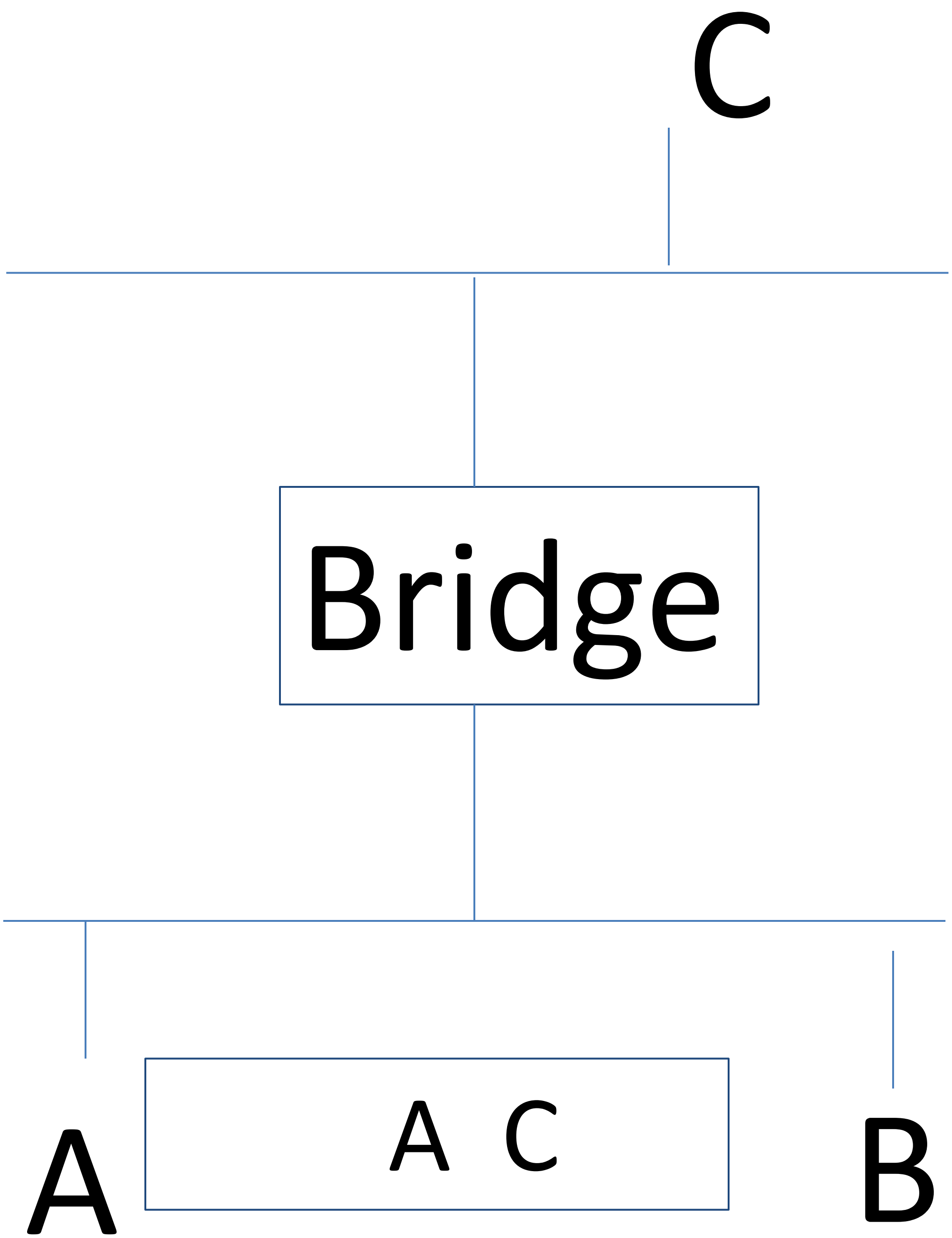
Circa 1980 at Digital: Ethernets under attack

- Ethernet had limite distance (2.5 km) and stations (8000). Also perception that Ethernet collapsed at high loads. Token ring emerging.
- Question: how can we extend 2 Ethernets to make a larger Ethernet that has twice the distance, twice the bandwidth and twice the number of stations?
- Repeaters don't work as they repeat all bits everywhere. So bandwidth will not be twice. Routers work, but are expensive because we need different routers for each high-level routing protocol on the Ethernet (In next page figure, DR = DEC Router, XR = Xerox Router. D1, D2, D3, D4 are DEC stations. X1, X2, X3, X4 are Xerox).
- How can we extend LANs ***transparently***: without end-stations knowing that they are on an extended LAN.


D1
D2
R
D3
D4
X1
X2
X4
X8

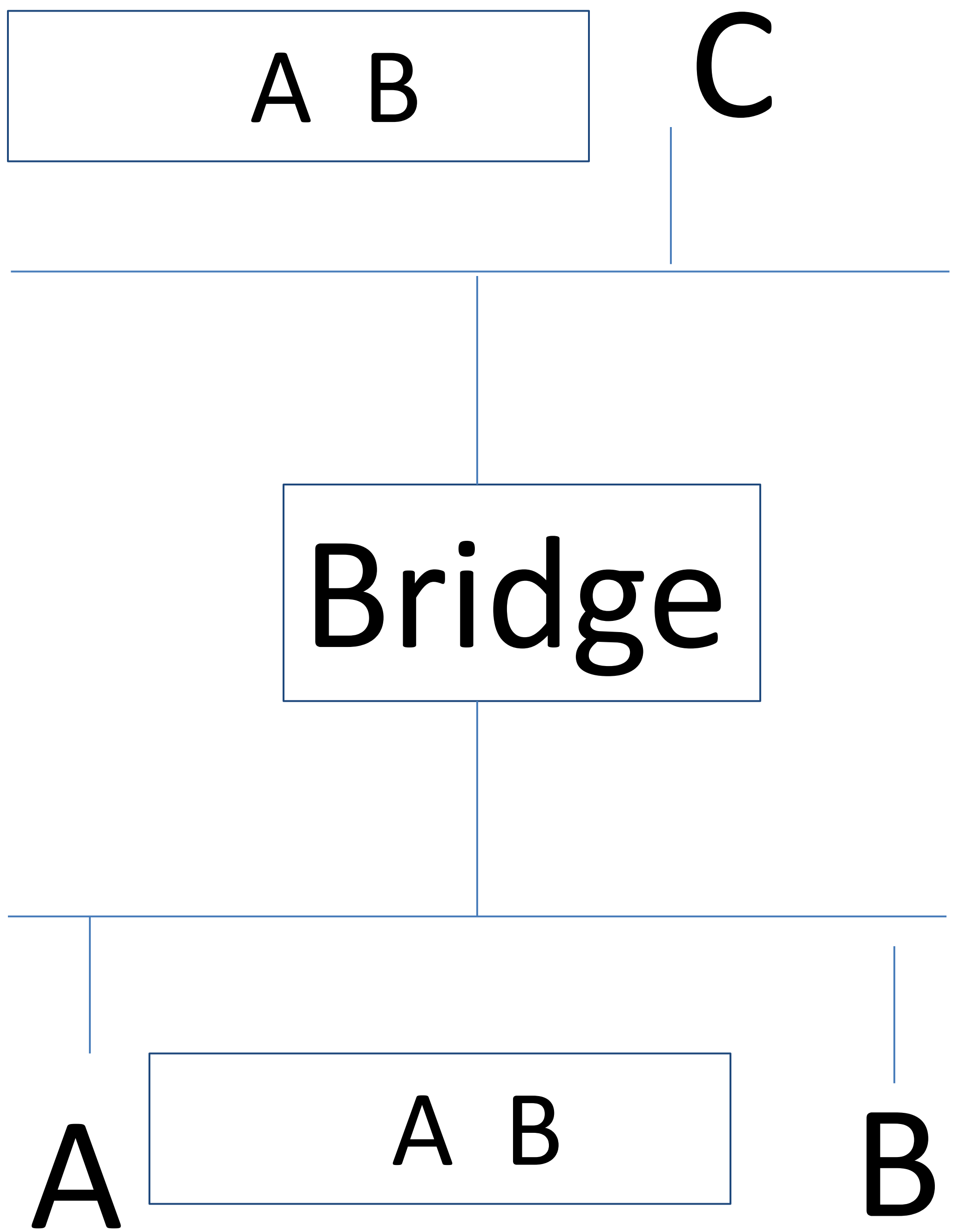

D1--> D3
R D1 . . .
D3's Routing Address
Data Link
Routing Header


D1
D2
DR
XR
D3
D4
X1
X2
X4
X8

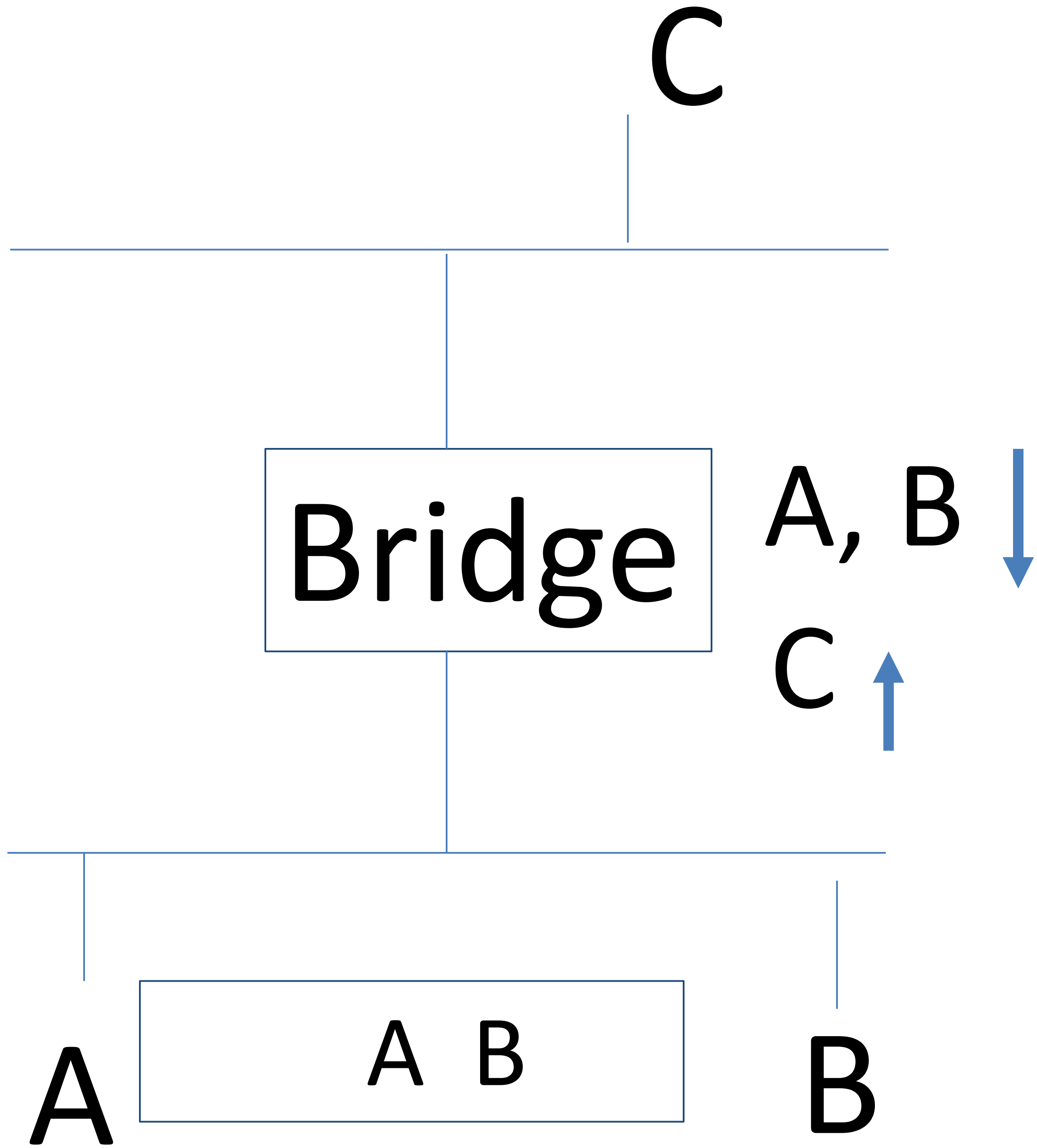


Want to let A talk to C. What Should bridge do?

SIMPLEST: FRAME REPEATER

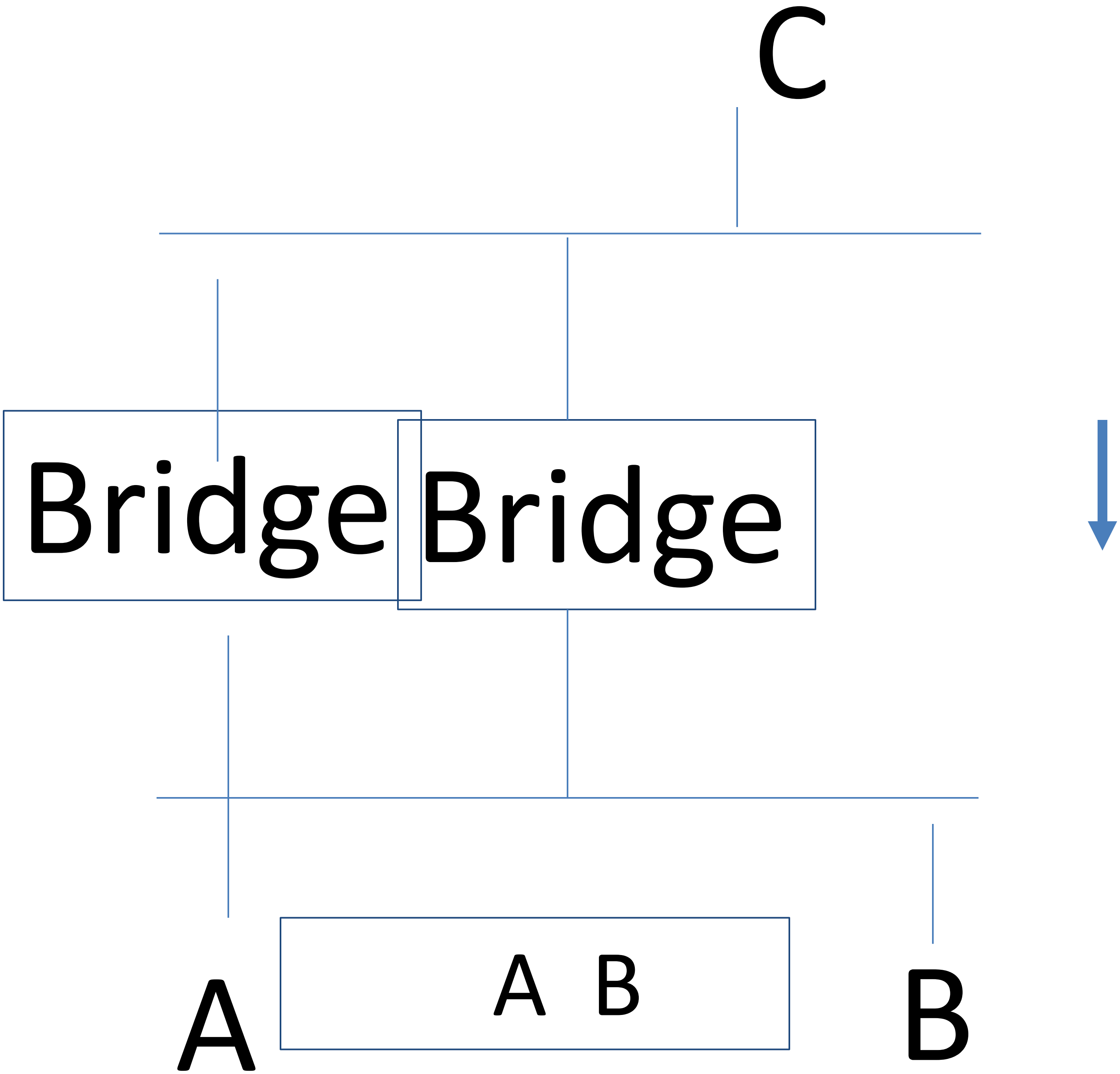


Want to let A talk to B. What Should bridge do? Can we do Better?



Want to let A talk to B. What Should bridge do?

Filter based on database

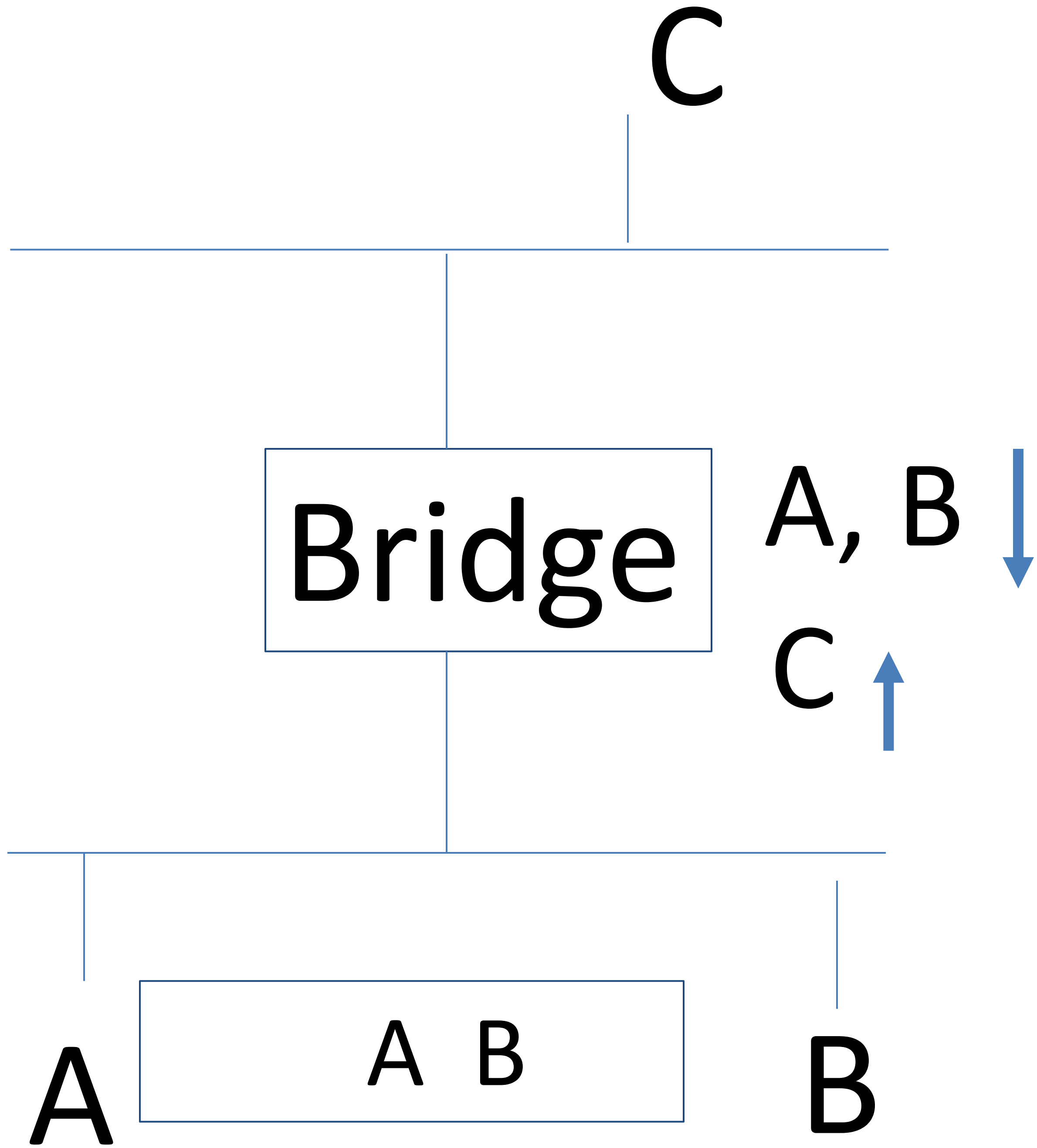


What if A sends to C?

Manual is error prone

How to build database?

Manually?



How to build database *automatically?*

BREAKOUT

Learn based on source Address and forward Based on destination Address and send to other Interface when no info is Known (flooding)

Generalizations

- Any LAN or Data Link that puts both source and destination addresses in frames (includes all 802 LANs)
- Any topology without cycles.
- More than two bridge ports.

Code

ReceiveFrame F on port X

```
AddTable(F.Source, X (* learn source, refresh timer *)
Y = Lookup (F.Dest) (* lookup destination *)
If (Y = Nil) then (* unknown destination *)
    Forward Frame F on All Ports Y!= X; (*flood*)
Else if (Y != X) then
    Forward Frame F on Port Y

Else if Y = X Drop frame F (* filtering *)
(* received on same port as source *)
```

TimerExpiry (E) (* timer for entry E fires *)

```
E.port = Nil;    (* reenable flooding *)
```

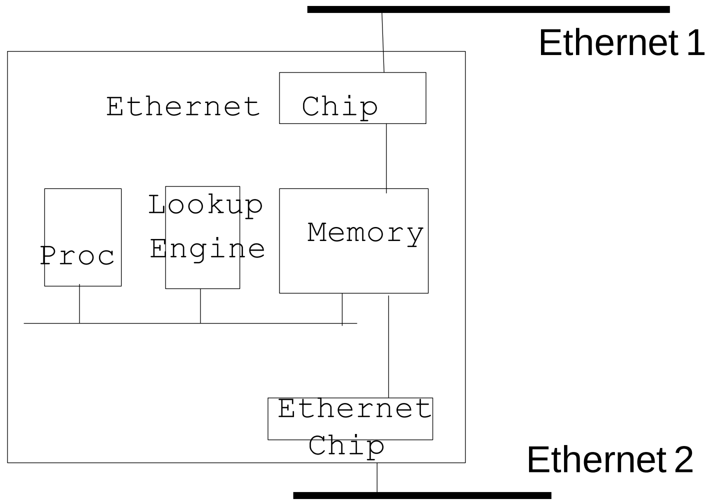
Bridge Terminology and Summary

- Transparency, Promiscuous Receive, Flooding, Filtering
- Main idea: Learn based on source addresses and forward based on destination addresses. Using flooding when there is no info, and timeout to handle stale learning information.

Realization

- Need much higher performance than router which handles all frames addressed to it.
- Need to decide whether to drop or forward frames in minimum interframe time on Ethernet, 51.2 usec on each port. Otherwise could drop some frames that need to be forwarded while examining others that have to be filtered.
- First DEC implementation by Mark Kempf in 1984 technology. Achieved forwarding in min frame time with low cost (1000 dollars).

Bridge Implementation

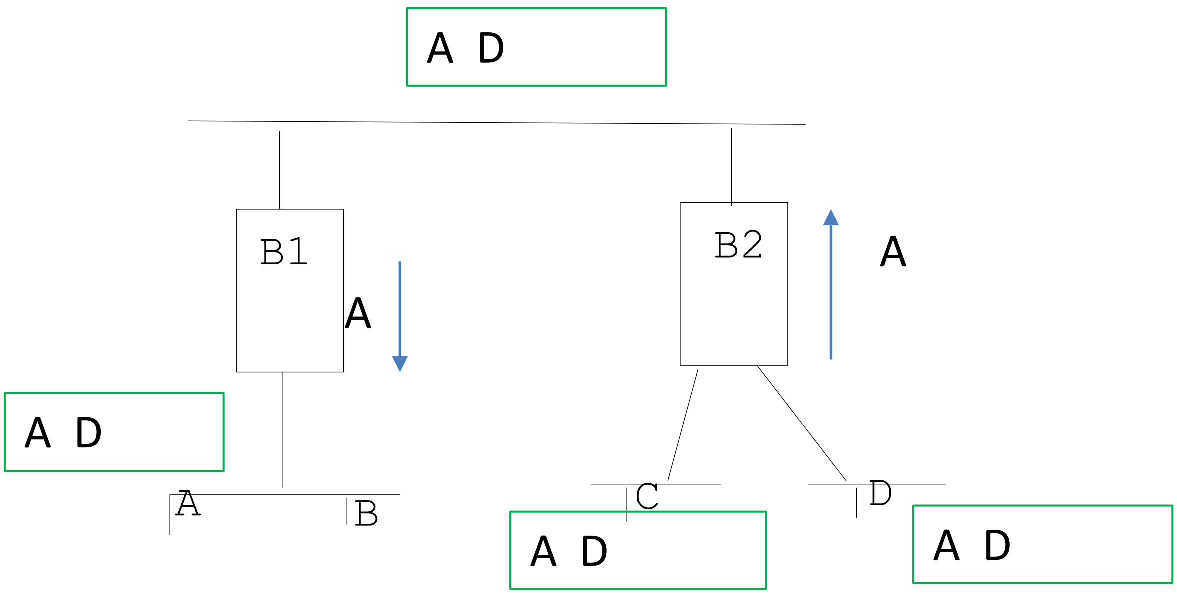


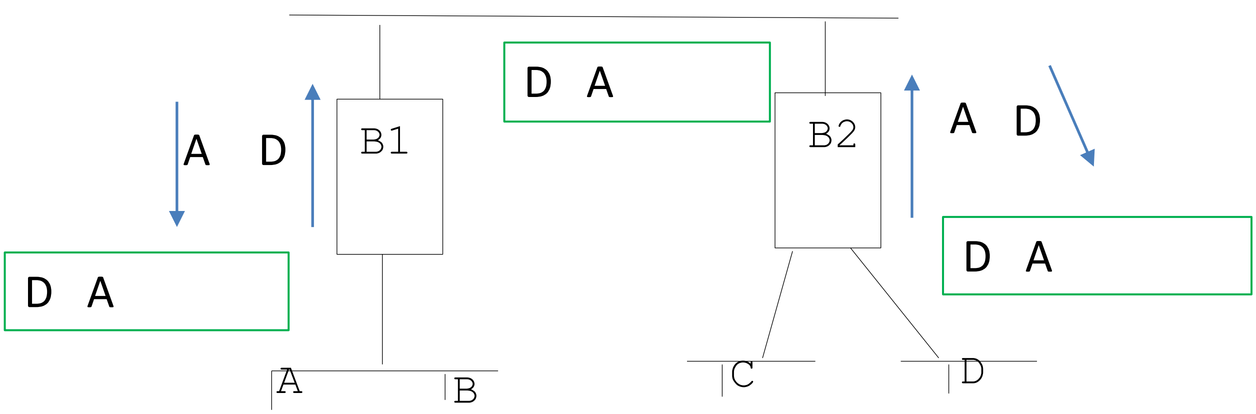
1) 3 bus design to avoid congestion on processor bus

2) 4 ported memory design.

3) Hardware binary search lookup engine. Takes log(8000) = 13 memory accesses of 100 nsec each = 1.3 usec

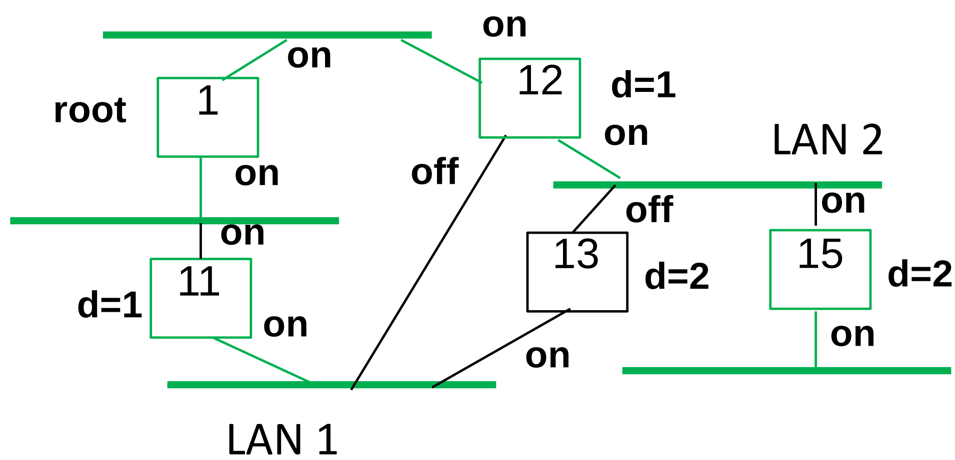
4) Processor stays in loop after a packet interrupt servicing as many packets as arriive to reduce context switching overhead.





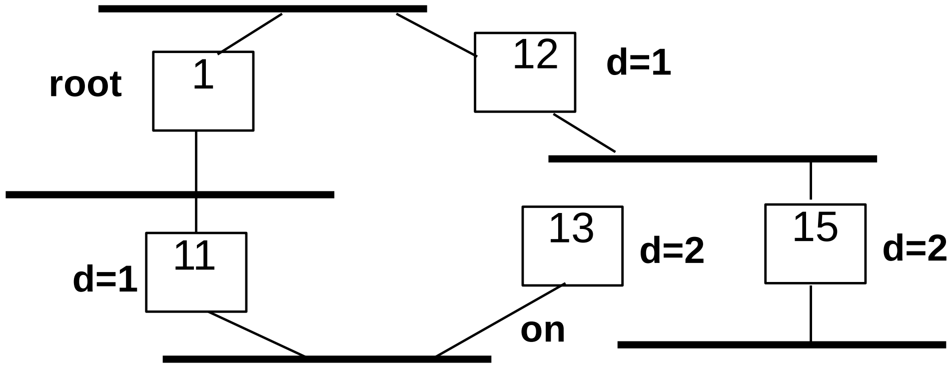
D to A packet is not sent on the LAN to C on the return because now bridges know where A is. No flooding on all subsequent frames

Centralized Algorithm



- Root is Min ID node (in this case Bridge 1)
- Other bridges finds Min port, port through which it has shortest path to root (parent), For 11 it is upper port.
- Each bridge also finds the ports for which this bridge is on the shortest path between root and corresponding LAN: Designated Ports. For example, 11 and 12 have d= 2 for LAN 1, so we pick shorter ID as tiebreaker, Bridge 11 is designated bridge for LAN 1, 12 for LAN 2
- Each bridge turns ON Min port and all Designated Ports. ON,OFF are software states: always receive hello and management messages on all ports. Drop data packets to/from OFF port.

Final Tree



- To be a tree, each LAN must have a unique path to every other LAN.
- Algorithm guarantees that each LAN can get to root only through designated bridge for LAN and designated bridges have unique path through their parent LAN.

Distributed Algorithm

- All nodes sends hello message saying ID and current distance to root (r, d). Every node N keeps an estimate r(N), d(N) of its current estimate of root and distance

- On hearing a smaller root, adopt that root and update your distance as d + 1:

  On receipt of (r,d) at Node N

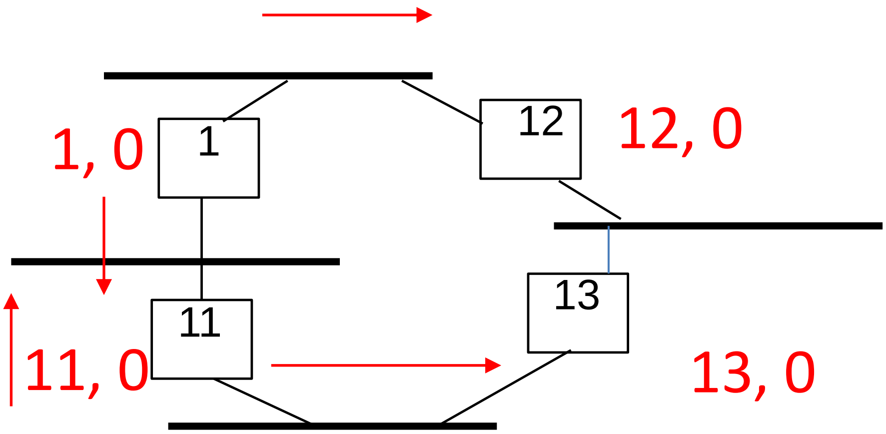
  If $(r < r(N))$ then $r(N) = r$; $d(N) = d+1$;

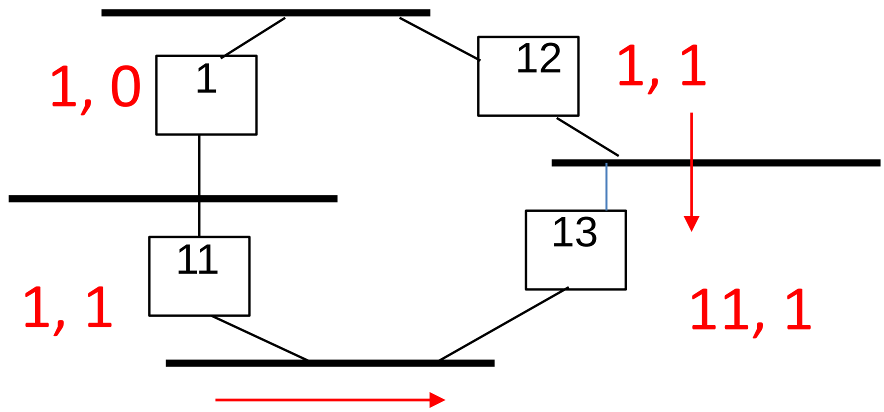
  Else if $(r = r(N))$ and $(d < d(N))$ then

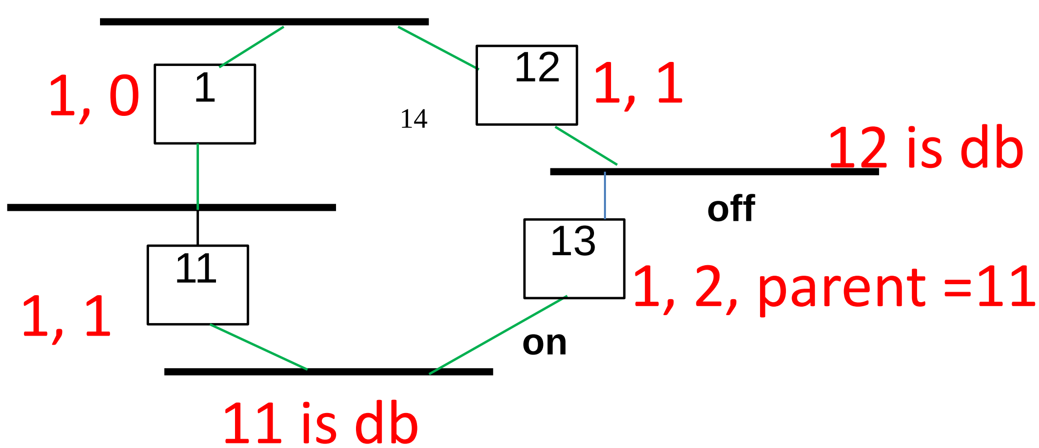
  $d(N) = d + 1$;

- Bridges must wait some some time after the estimates stabilize to turn ON ports and start forwarding to avoid loops

Spanning Tree Computation







Conclusion

- Necessity is the mother of invention.
- Find new problems that matter and find creative solutions. Challenge the dogma (wire-speed)
- The big ideas: auto-configuration, filtering, flooding, wire speed forwarding, spanning tree, designated bridges
- Link layer relays versus physical layer relays. Next lecture: routing layer relays aka Internet routers! Why so many concepts. Will discuss next time.