Do the Wrong Thing!

Radia Perlman

Radia.Perlman@Dell.com

Nanog 2022

Feb 14, 2022

Overview

- A lot of history
- Two decisions
  - Assuming Ethernet was a “network” instead of a link
  - Not adopting CLNP in 1992
- Surprisingly good things
  - DHCP
  - NAT

Understanding Network Protocols

- Nobody would have designed what we have today
- No way to understand networks without looking at the history

How networking tends to be taught

- Memorize these standards documents, or the arcane details of some implementation that got deployed
- Nothing else ever existed
- Except possibly to make vague, nontechnical, snide comments about other stuff

Things are so confusing

- Comparing technology A vs B
  - Nobody seems to do that
  - Nobody knows both of them
  - Both A and B are moving targets
- Standards bodies…

So, first we need to review network “layers”

- ISO credited with naming the layers
- It’s just a way of thinking about networks

Perlman's View of ISO Layers

- 1: Physical
- 2: Data link: (neighbor to neighbor)
- 3: Network: create path, forward data (e.g., IP)
- 4: Transport: end-to end (e.g., TCP, UDP)
- 5 and above:
  - boring

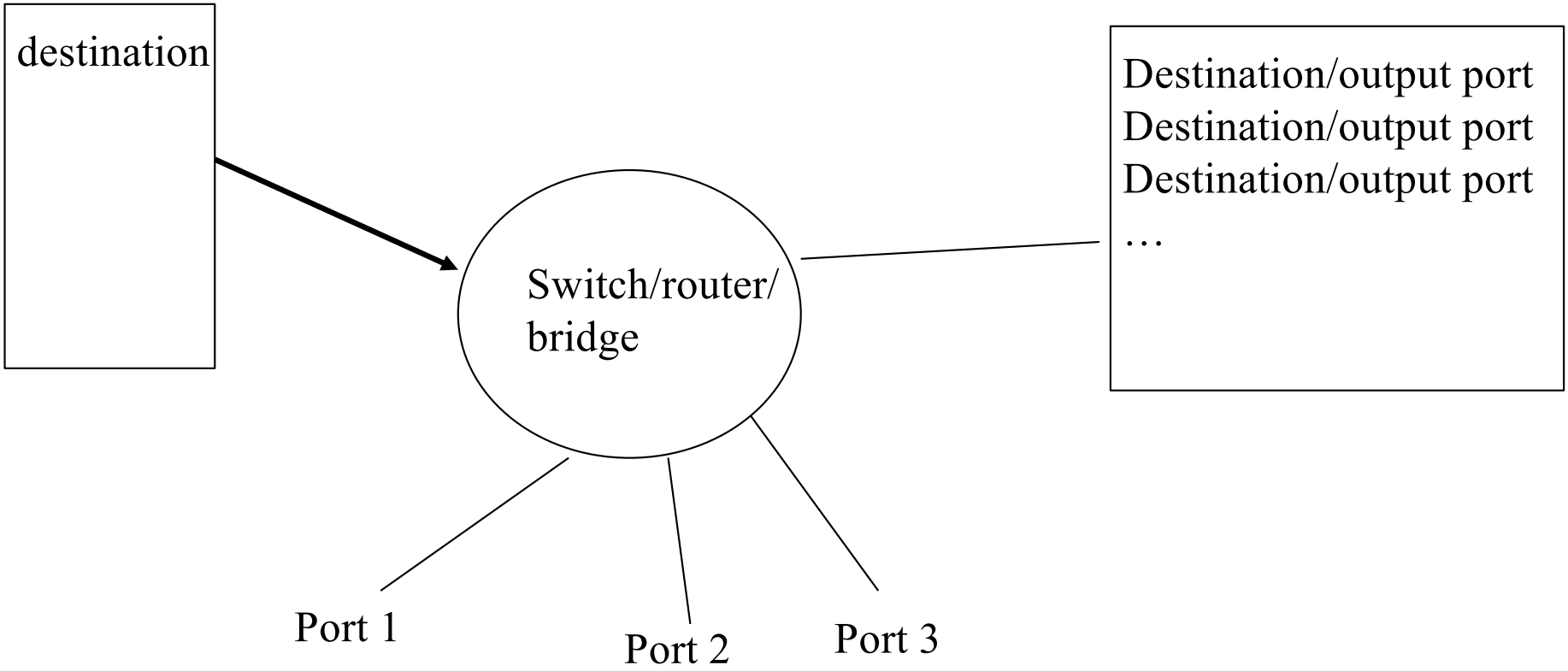
Ethernet and IP

- Ethernet is layer 2, right?
- IP is layer 3?

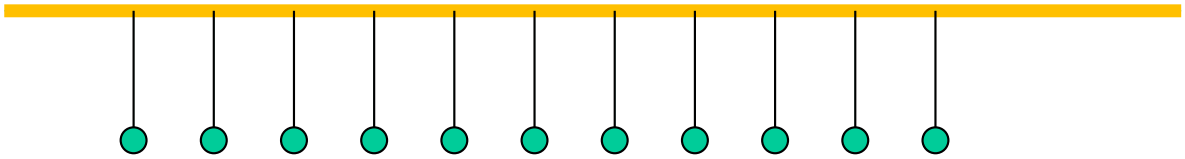
If Ethernet is layer 2….

- Why are we forwarding Ethernet packets?
- Oh, but it's a "switch" rather than a "router", so it's not really forwarding????

Forwarding Table

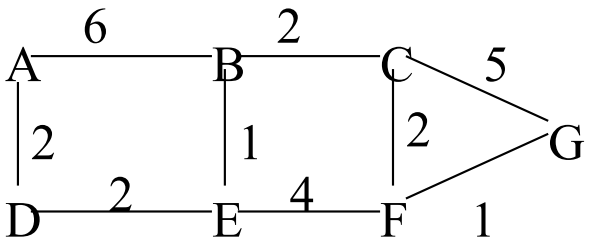


Original Ethernet: A way for a bunch of nodes to share the same wire

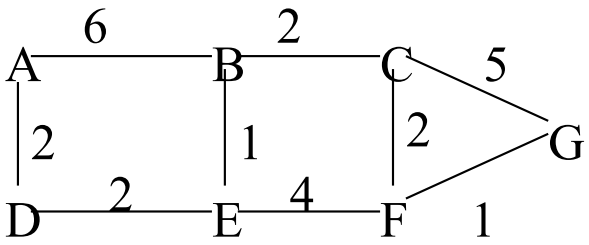


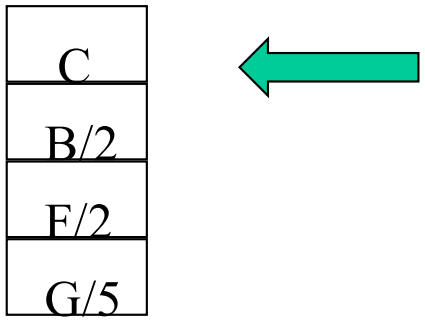
No leader to call on them…all peers
If more than one speaker, receivers get garbage
CSMA/CD (CS=Carrier sense, MA-multiple access, CD=collision detect)

I was doing layer 3 then (IS-IS)



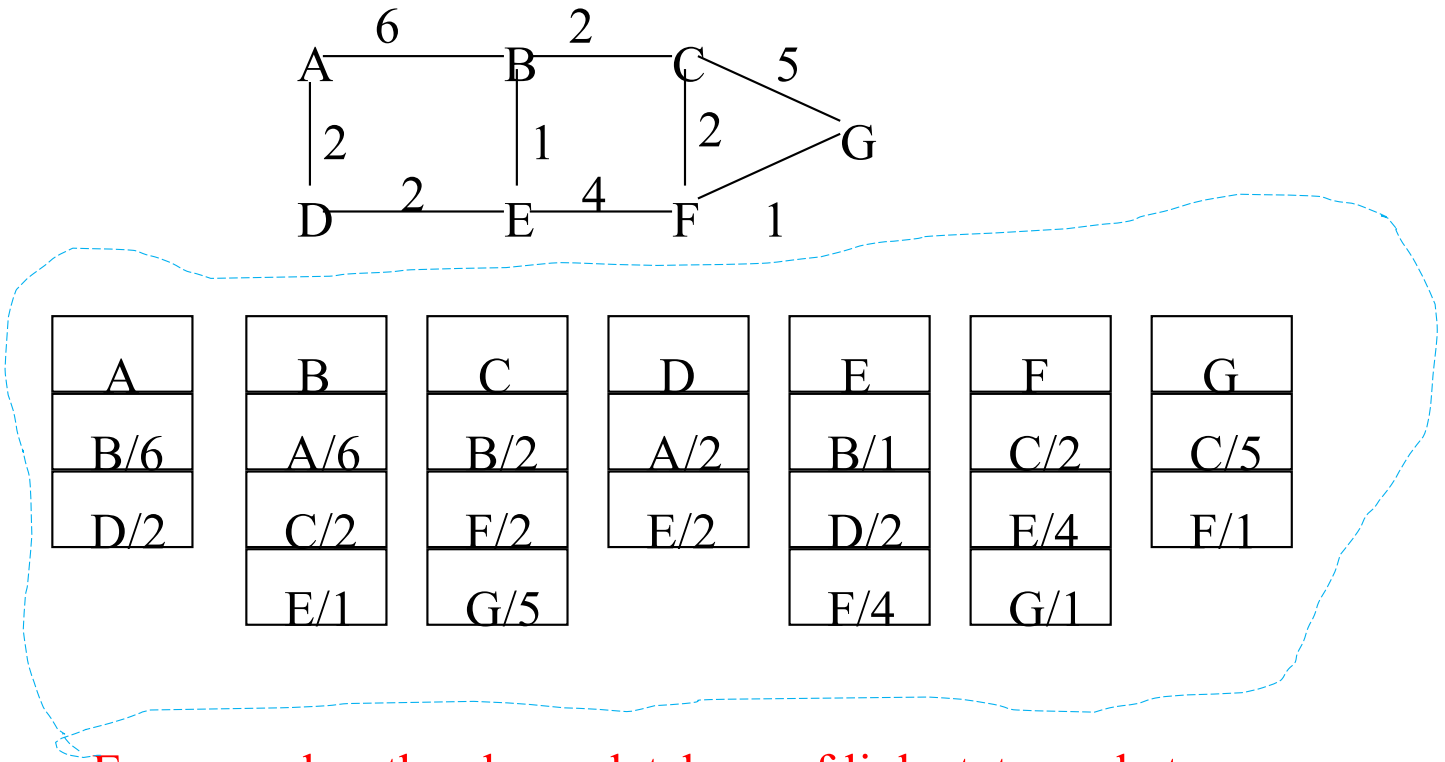
A network



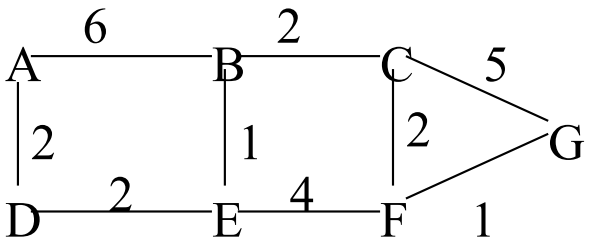


Each node creates a “link state packet” saying who it is, and who its neighbors are. This is C’s link state packet.

It gets sent to all the other nodes



Everyone has the above database of link state packets



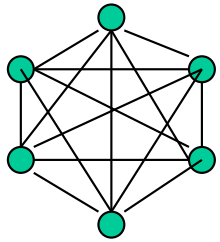
A	B	C	D	E	F	G
B/6	A/6	B/2	A/2	B/1	C/2	C/5
D/2	C/2	F/2	E/2	D/2	E/4	F/1
	E/1	G/5		F/4	G/1	

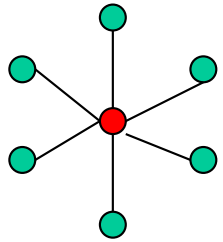
Enough information to calculate paths

Enter Ethernet!

I saw Ethernet as a new type of link

- I modified IS-IS a little to accommodate this type of link
- For instance, the concept of “pseudonodes” so that instead of n^2 links, it’s n links with n+1 nodes





But Ethernet was a link in a network, not a network

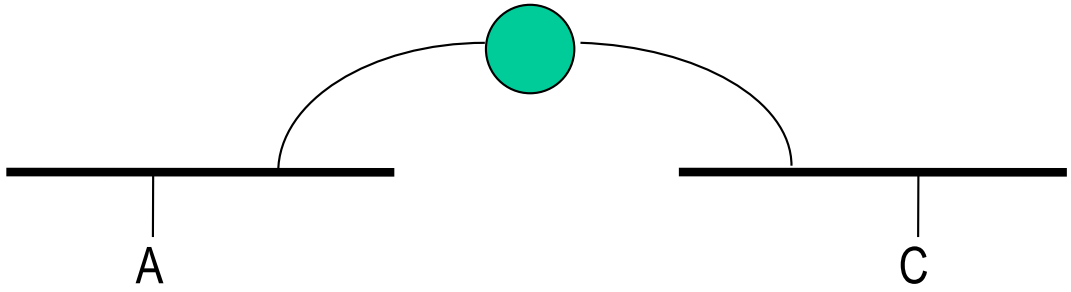
- I wish they'd called it "Etherlink"

How Ethernet evolved from CSMA/CD to spanning tree

- People built apps on Ethernet, with no layer 3
- (Layer 3) router can't forward without the right envelope
- I tried to argue…
- The applications were good, and the implementers were considered heroes
- The applications would have been just as good if they'd done it correctly…on top of layer 3

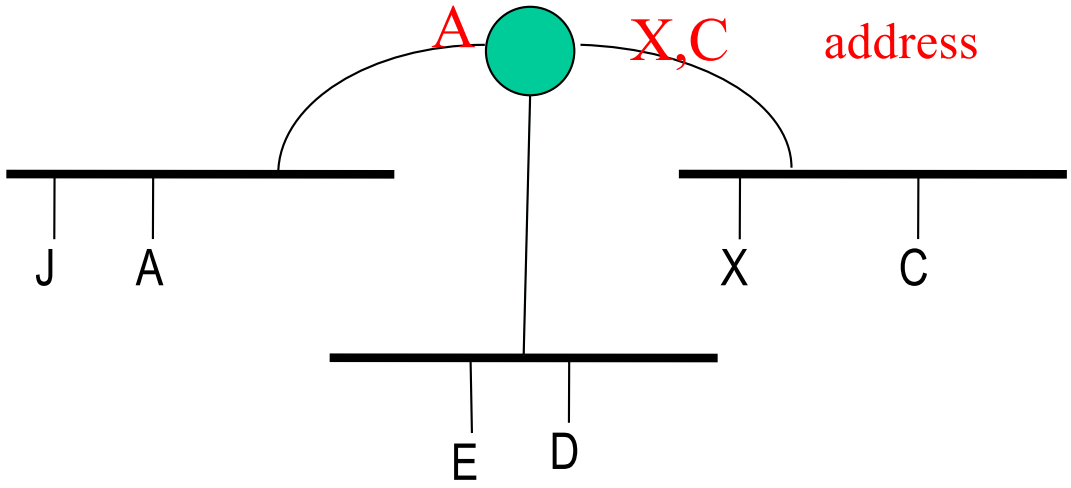
Problem Statement (from about 1983)

Need something that will sit between two Ethernets, and let a station on one Ethernet talk to another



Without modifying the endnode, or Ethernet packet, in any way!

Basic concept

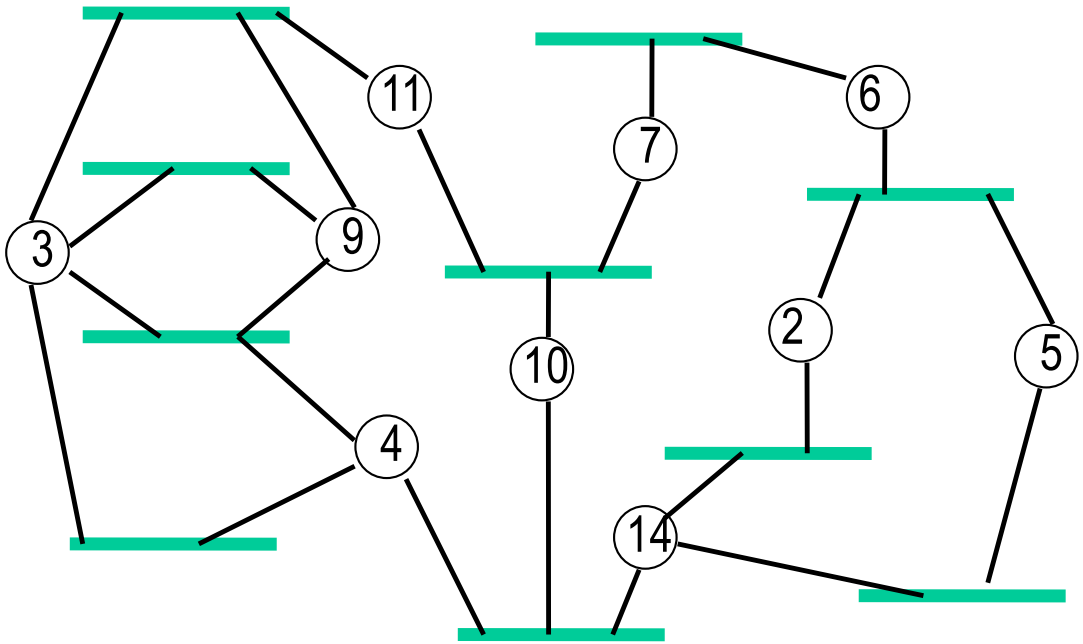


Optimization: Learning location based on source address

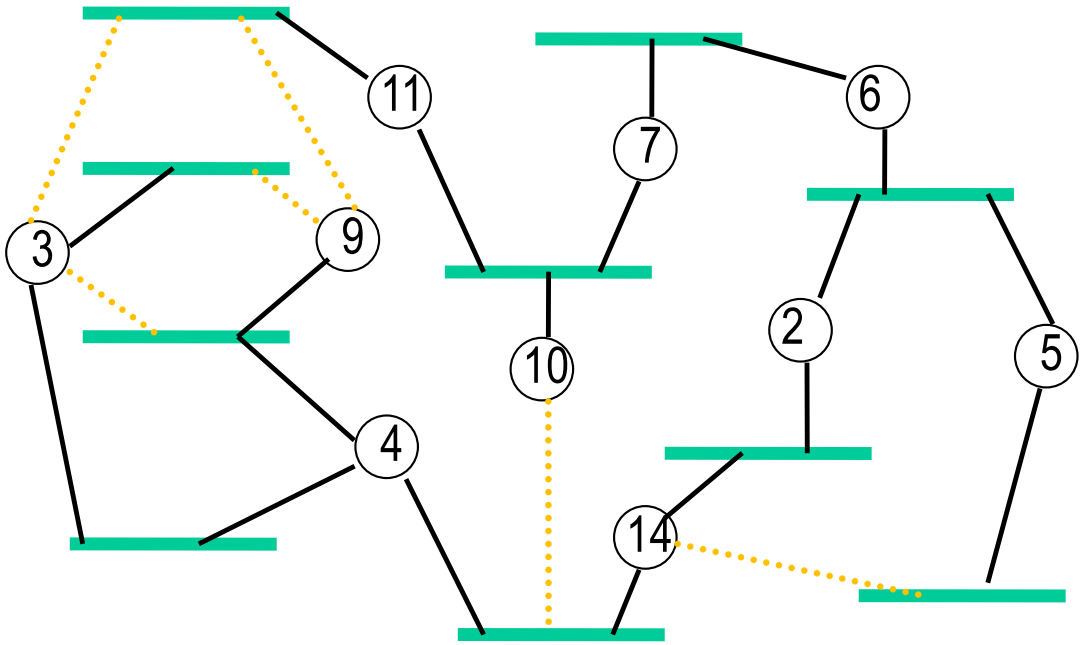
What to do about loops?

- Just tell customers “no loops”?
  - What about miscabling?
  - What about backup paths?
- So…desire for loop-pruning algorithm
  - Allowing any physical topology
  - Pruning to a loop-free topology for sending data
  - So…the birth of the spanning tree algorithm (to be described)

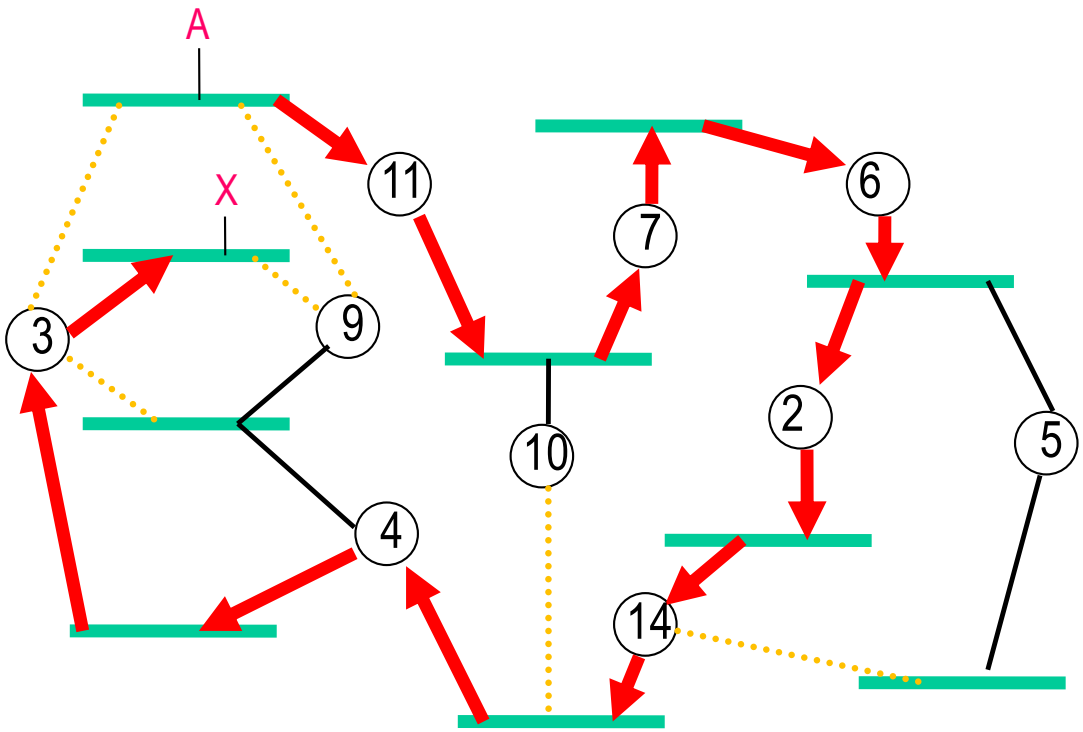
Physical Topology



Pruned by Spanning Tree



Path from A to X



CSMA/CD died long ago

- A variant is used on wireless links
- But wired Ethernet quickly became spanning tree with point-to-point links
- So “Ethernet” today has nothing to do with the original CSMA/CD Ethernet invention

Algorhyme

I think that I shall never see
A graph more lovely than a tree.

A tree whose crucial property
Is loop-free connectivity.

A tree which must be sure to span
So packets can reach every LAN.

First the root must be selected,
By ID it is elected.

Least cost paths from root are traced,
In the tree these paths are placed.

A mesh is made by folks like me.
Then bridges find a spanning tree.

Radia Perlman

Why Bridging was so popular

- At the time there were lots of layer 3 protocols (IP, IPX, DECnet, Appletalk)
- There were multiprotocol routers, but they were slow and expensive and complex to configure
- Bridges just worked, and were cheap and fast, and autoconfiguring

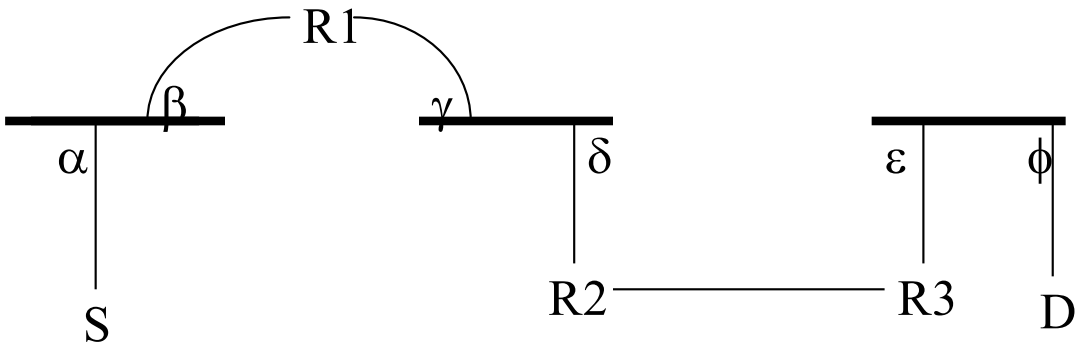
Why spanning tree Ethernet is a kludge

- You don’t get optimal paths
- Unused links (those not selected to be in tree) underutilized, other links overutilized
- Temporary loops really dangerous (header has no hop count)

Why not get rid of Ethernet and use only IP?

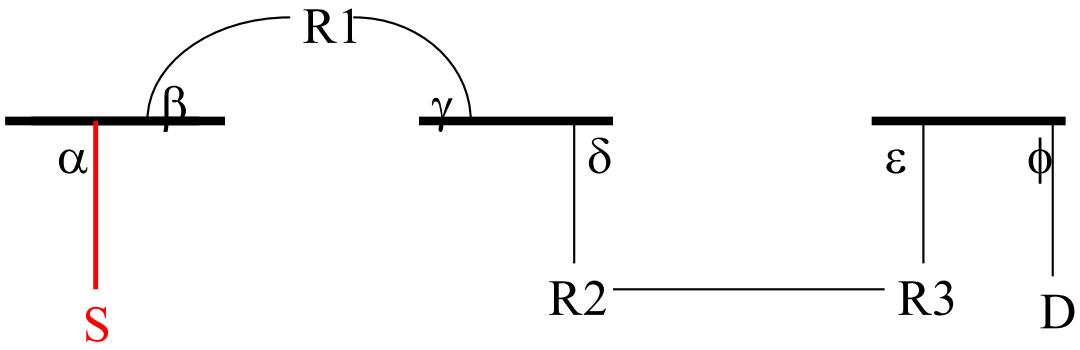
- World has converged to IP as layer 3, and it's in the network stacks – so original reasons for needing bridged Ethernet is gone
- On a link with just 2 nodes, why do you need a 6-byte source and 6-byte destination address just to talk on that one link??
- Why can't you just forward things with layer 3?
- If IP were designed differently, we wouldn't need Ethernet header anymore!

Today: Two headers: “layer 3”, and “layer 2”



As transmitted by S? (L2 hdr, L3 hdr)
As transmitted by R1?
As received by D?

Hdrs inside hdrs

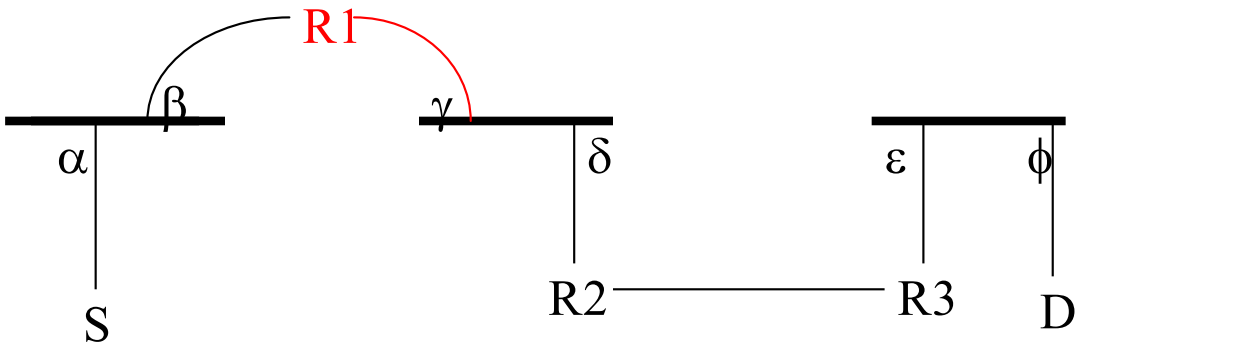


S:

Dest=β Source=α	Dest=D Source=S	

Layer 2 hdr Layer 3 hdr

Hdrs inside hdrs

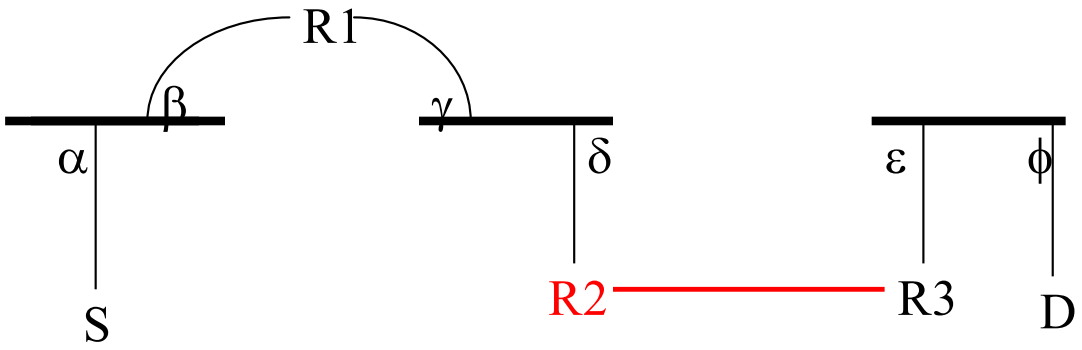


R1:

Dest=δ Source=γ	Dest=D Source=S	

Layer 2 hdr Layer 3 hdr

Hdrs inside hdrs

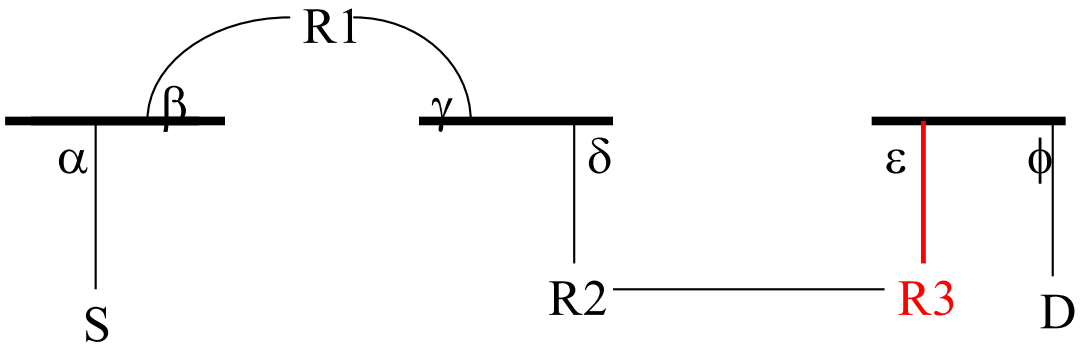


R2:

No hdr needed	Dest=D Source=S	

Layer 2 hdr Layer 3 hdr

Hdrs inside hdrs

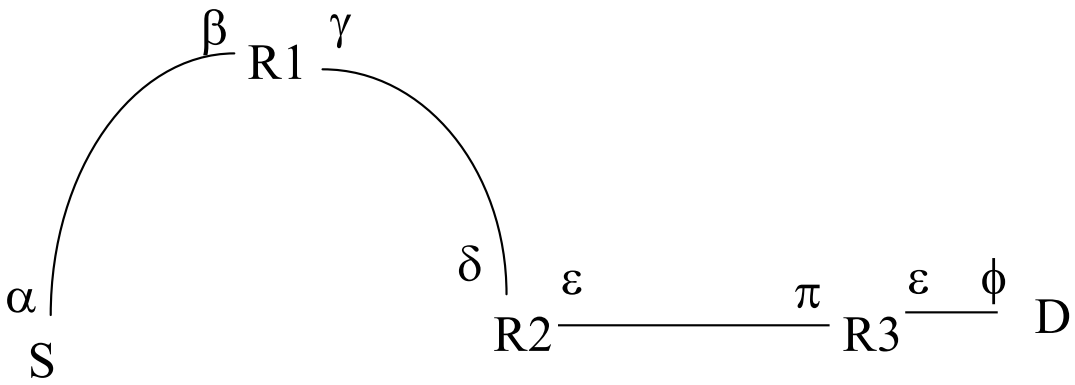


R3:

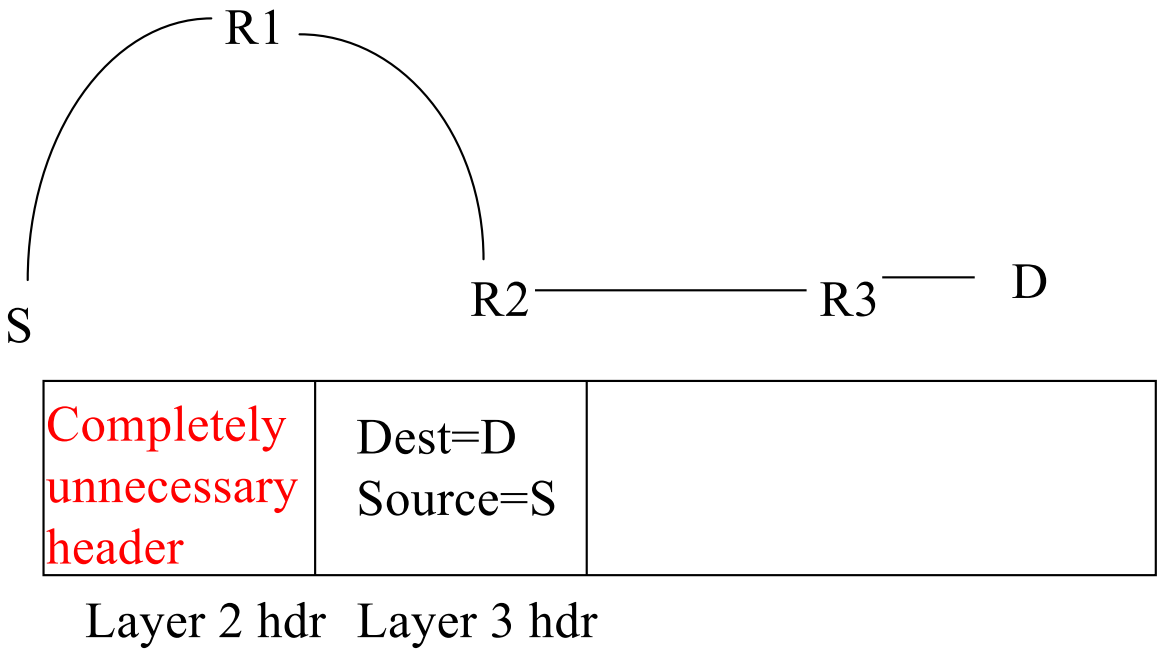
Dest=ϕ Source=ε	Dest=D Source=S	

Layer 2 hdr Layer 3 hdr

With all pt-to-pt links



With all pt-to-pt links



What's wrong with IP?

- IP is configuration intensive, moving VMs disruptive
  - Every link must have a unique block of addresses
  - Routers need to be configured with which addresses are on which ports
- If something moves, its address changes
  - If you move from one side of an IP router to another, your layer 3 address has to change
  - You can't have a cloud with a flat address space, where nodes can move around without changing IP address

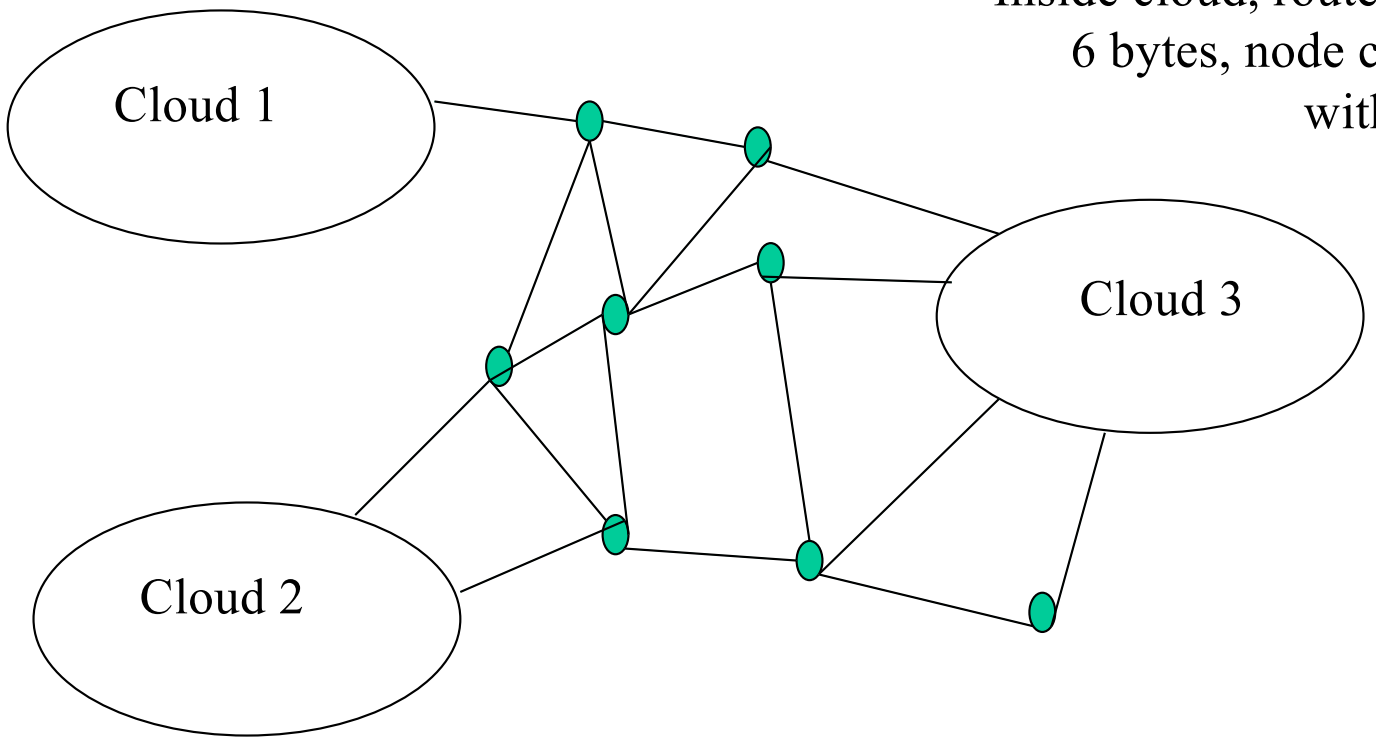
Layer 3 doesn't have to work that way!

- CLNP / DECnet...20 byte address
  - Top 14 bytes shared by all nodes within cloud
  - Inside the cloud, route to 6 byte ID -- nodes can move within the cloud without changing their address
  - Enabled by "ES-IS" protocol, where endnodes periodically announce themselves to the routers

CLNP

14 byte prefix routes to a cloud

Inside cloud, route based on last 6 bytes, node can move within cloud

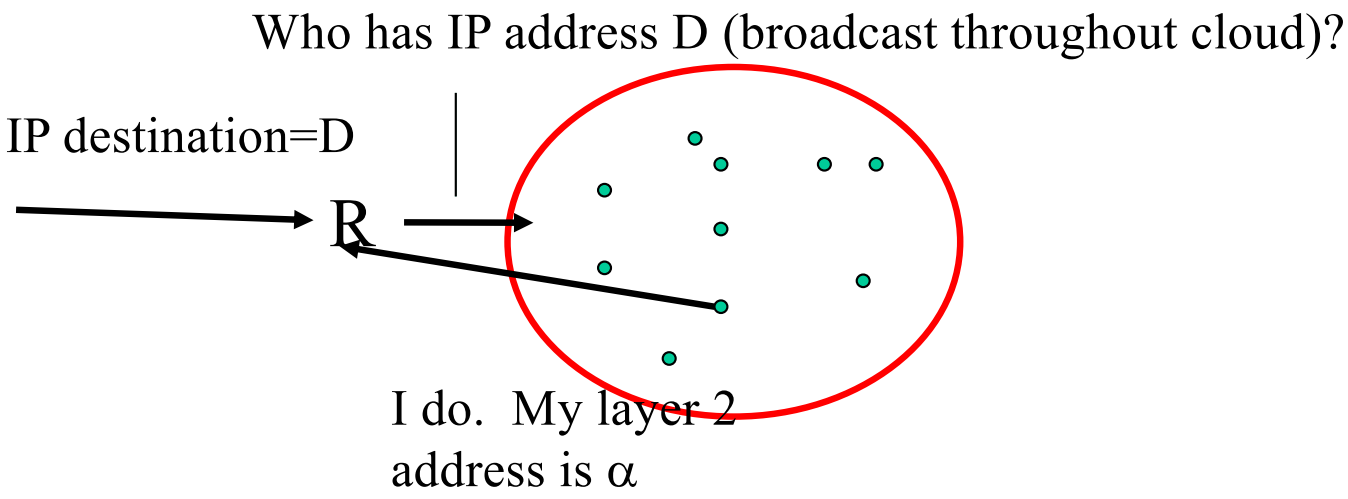


CLNP also had

- Autoconfiguration (put your configured MAC address into bottom 6 bytes)
- Plenty of addresses
  - (20 bytes vs 16 bytes for IPv6)
  - 14 byte “high order part” vs 8 for IPv6
- Widely deployed, supporting large customer networks

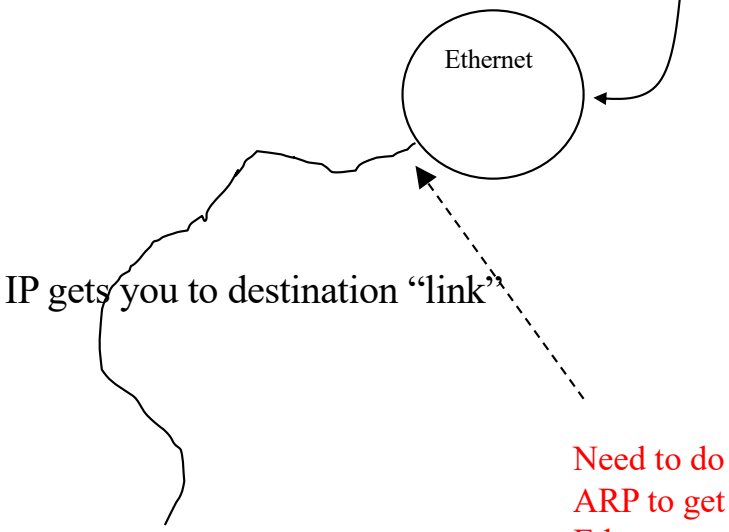
IP depends on “something else” creating a cloud with a flat address space

ARP/ND…finding destination's "layer 2" address



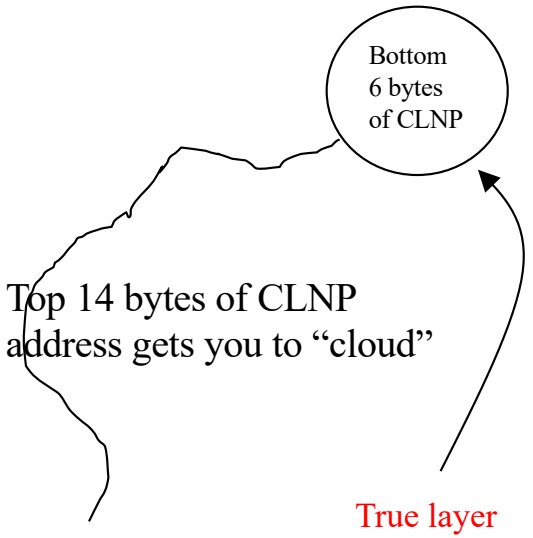
IP Plus Ethernet

CSMA/CD?
Spanning tree?
TRILL?
VXLAN?



Need to do
ARP to get
Ethernet
address

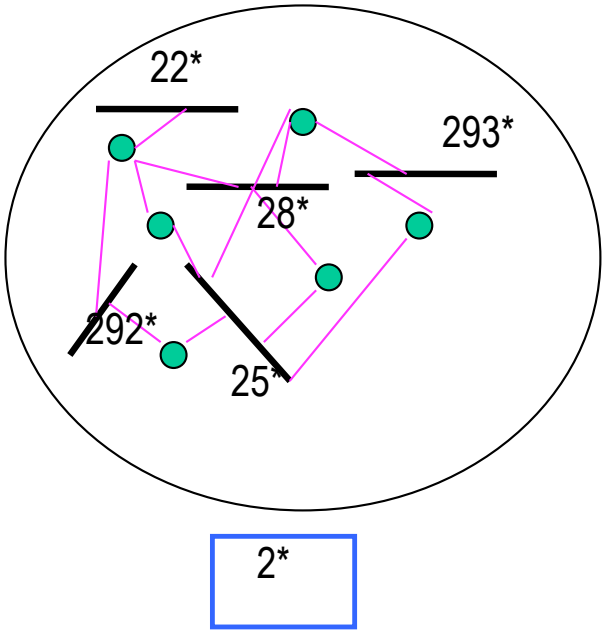
CLNP



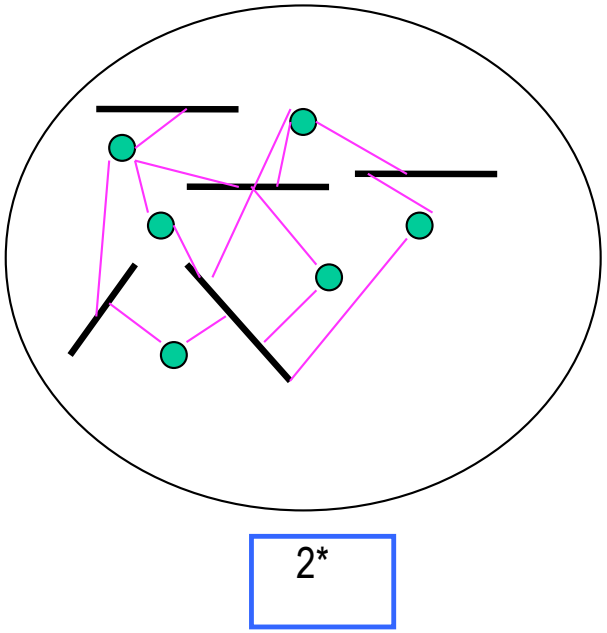
True layer
3 routing inside
final circle.
No need for layer 2 header

Hierarchy

One prefix per link (like IP)



One prefix per cloud



Worst decision ever

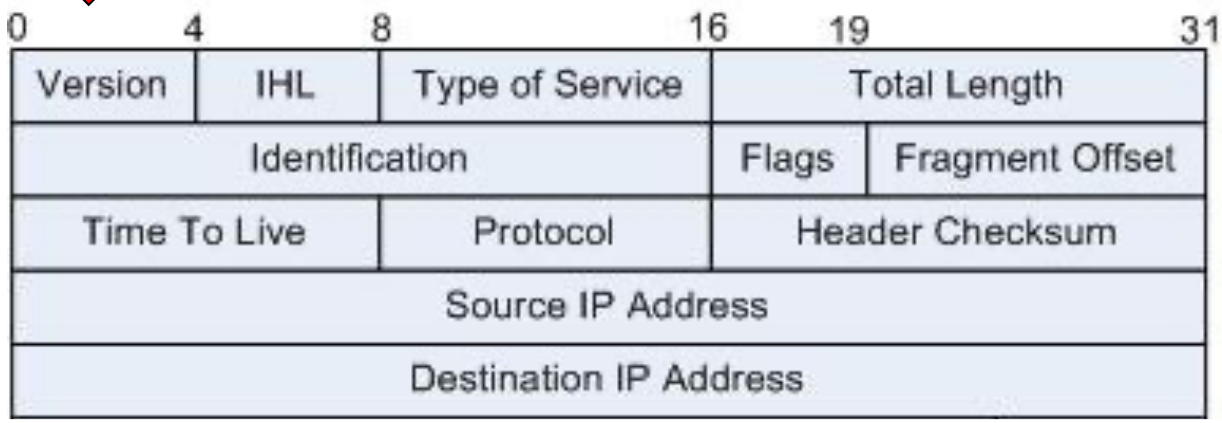
- 1992…Internet could have adopted CLNP
- Someone modified TCP to work on top of CLNP…and all Internet applications therefore would work with CLNP
- Easier to move to a new layer 3 back then
  - Internet smaller
  - Not so mission critical
  - IP hadn't yet (out of necessity) invented DHCP, NAT (explained soon), so CLNP gave understandable advantages
- CLNP much cleaner than IP
  - No ARP
  - "Level 1" is cloud with flat address, and true layer 3 routing (hop count, best paths)
- Decision: Let's invent something new. We'll call it IPv6

IPv6

- Just a 16-byte version of IP, with 8 bytes for "prefix" and 8 bytes for "node ID on last link"
  - 8 bytes on top is inconveniently small for administering addresses
  - 8 bytes on bottom is way too much (e.g., DHCP, to be discussed soon)
- Technically inferior to CLNP!
  - IPv6 (like IPv4) needs a different prefix on every "link"
  - So IPv6 depends on "something else" to create a cloud with a flat address space
- No more "compatible with IPv4" than CLNP is
- Silly hype
  - like that it has 2^{128} addresses (not true with 8-8 split, and in general with hierarchy)
  - Or that it has "security built in" vs IPv4

Is IPv6 just a “new version” of IPv4?

Version Number



What's a Version Number?

- Version number
- What is the purpose?
- Philosophical question:
  - what is "new version" vs "new protocol"?

What makes sense

- Envelope says what the protocol is (which upper layer process should receive the packet)
  - Ethernet: Ethertype
  - IP: Protocol Type
  - TCP/UDP: port

Different Protocol vs Different Version

- Protocol type says which process to give the packet to
- If differentiate based on protocol type, then it's a different protocol
- If share a protocol type, and differentiate based on version number, then it's a new version of the same protocol

If differentiate based on version number

- You can't just say "write this value into this field"
- You have to say "Look at the version number, and if it's not your version, then drop the packet"!
- IPv4 never said that…so need a new Ethertype for IPv6
- So "IPv6" is a new protocol…not a "new version" of IPv4

Reasons given at the time not to adopt CLNP

- “It would be ripping out the heart of the Internet and putting in a foreign thing”
- “We don’t like the ISO layer 6”
- “We’re not immediately out of IPv4 addresses, and I’m sure we could invent something way more brilliant”
- And (unsaid) “But it’s a different sports team!

Alternate History

Stuff that would never have been invented, without these “bad decisions”

If we had treated Ethernet like a link rather than a network

- Endnodes would have used layer 3 (with IS-IS) to forward between links
- The Ethernet header would have died out
- Spanning tree Ethernet wouldn't have been invented
- But the spanning tree algorithm is actually a nice simple thing that (I've been told) is being used in other contexts
- And the "baked in" MAC address is useful for lots of things
- Though privacy advocates get upset about a baked-in unique ID

If Ethernet remained CSMA/CD

- We couldn't have had fast Ethernet
- CSMA/CD requires detecting collisions within minimum packet size
- So if the speed is x times faster, the maximum cable length has to be decreased by a factor of x (or increase min pkt sz)
- Original 10 Megabit Ethernet…about .5 Kilometer
  - 10 Gig CSMA/CD would have been maximum length .5 Meter

Spanning Tree Ethernet allows IPv4 a much longer life

- “Links” in IP would have been limited to a single LAN (hundreds of nodes)
- No ability to have a multilink cloud look to IP like a single IP link with a flat address space
- Spanning tree Ethernet widely deployed by 1992…maybe without spanning tree Ethernet, this would have caused the world to have had the good sense to go to CLNP in 1992?

If we had adopted CLNP

If we had converted to CLNP in 1992

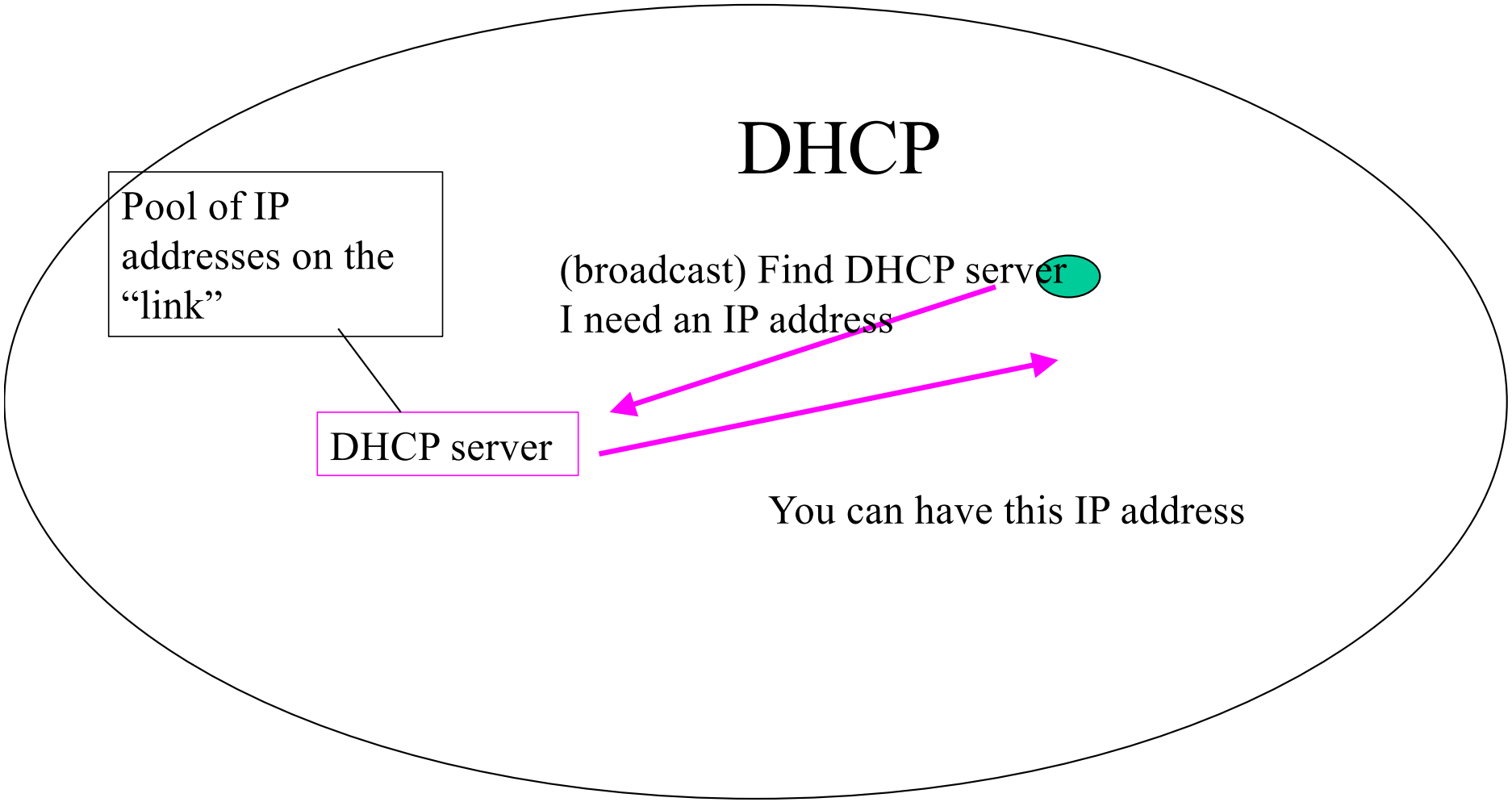
- What we think of as “layer 2” would have just been the level 1 routing of IS-IS (the bottom 6 bytes of the CLNP address)
- Flat cloud with optimal paths, hop counts
- Ethernet header would have been unnecessary

Good Stuff we (Probably) wouldn't have thought of

Story: The invention of DHCP

How do endnodes get their address?

- CLNP (and IPX): stick 6-byte Ethernet address into layer 3 address
- Appletalk: incredibly cute hack with only 3 byte address!
- IPv4: originally manual configuration, then DHCP (Dynamic Host Configuration Protocol)


DHCP
Pool of IP addresses on the "link"
(broadcast) Find DHCP server
I need an IP address
DHCP server
You can have this IP address

DHCP

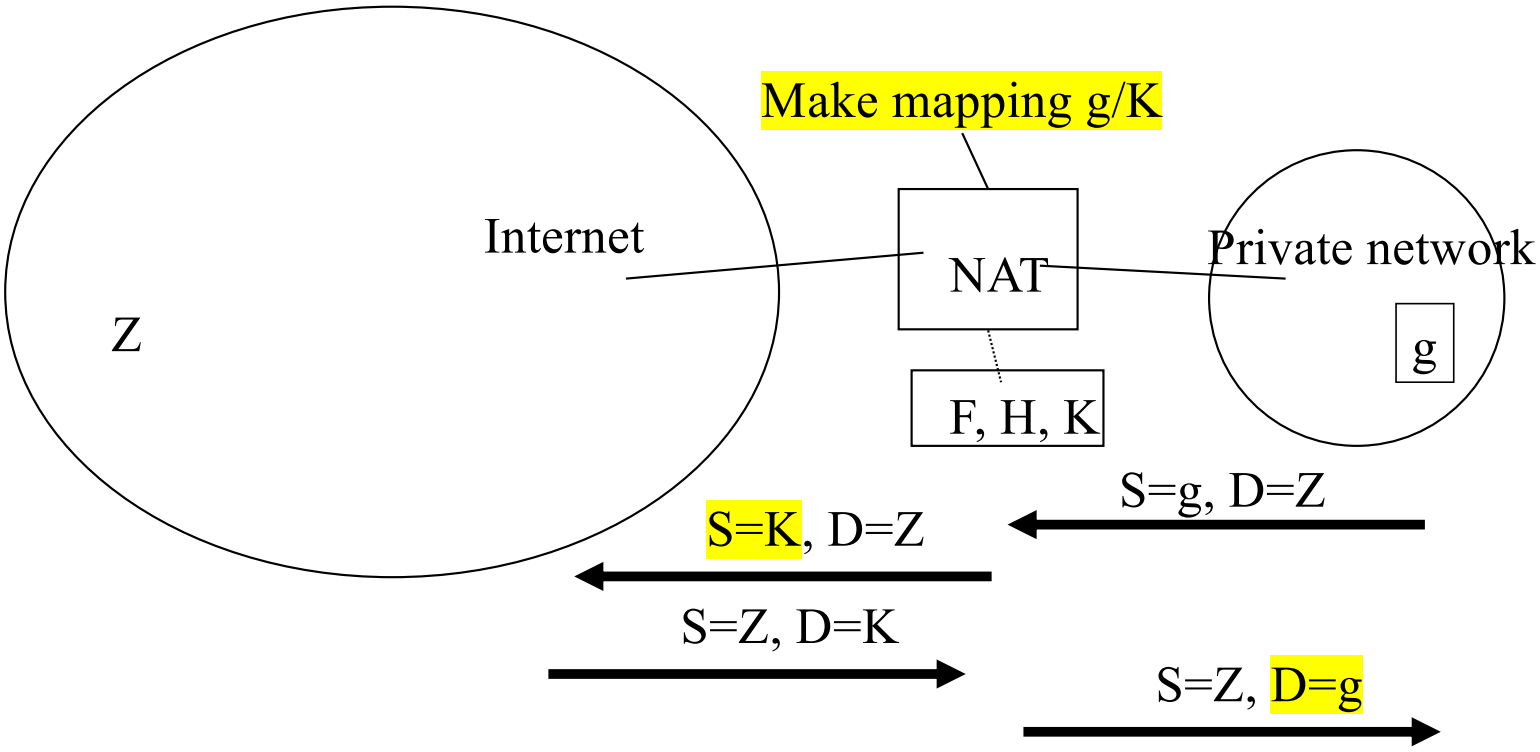
- Probably wouldn't have been invented if CLNP adopted in 1992
  - CLNP already had autoconfiguration of endnodes – stick 6 byte Ethernet address into 20-byte layer 3 address
- But DHCP has some important advantages

DHCP is really good!

- Don’t need to use 6 bytes for link-local ID…3 bytes enough
- So bigger prefix, or smaller total address
- The bigger the “prefix”, the easier to administer addresses
- “Privacy”: If my permanent Ethernet address were embedded in my layer 3 address, no matter where I connected, it would be known that it was my machine – with DHCP, more privacy

NAT: Another cool thing

NAT


Make mapping g/K
Internet
NAT
Private network
Z
g
F, H, K
S=g, D=Z
S=K, D=Z
S=Z, D=K
S=Z, D=g

NAT properties

- Nodes inside private network cannot be contacted from outside, unless the inner node makes contact first
- Lets us live with IPv4, perhaps indefinitely
- A case can be made that IPv6 (16 byte addresses) doesn't give you anything you can't get with IPv4+NAT
  - Geoff Huston "In Defense of NATs"
    https://www.potaroo.net/ispcol/2017-09/natdefence.html

Hatred of NATs

- Don't "help NATs" by documenting how they work
- Purposely design IPsec's AH header to "break NATs"
  - Type different headers: AH and ESP
  - AH not only protected the data, but also so the IP header
  - No security reason to do so
  - Nobody needs AH anyway. ESP works

But NATs are good!

- Even if we have infinite addresses
  - NAT enhances security
  - It allows different types of addresses in different portions of the net

Summary

- CSMA/CD vs token ring vs token bus irrelevant
- Both DHCP and NAT work just fine with any layer 3 protocol, including CLNP
- Both DHCP and NAT are good things, no matter what the layer 3 protocol is
- But would the world have thought of inventing them if we had CLNP?

Closing Thoughts

- Tribalism bad
- Technology that wins isn’t necessarily “best”
- Autoconfiguration good
- Know what problem you are solving

Thank you!