Myths and Mysteries in the Network Protocol World

Radia Perlman
Radia.Perlman@intel.com

Pet peeve

- Networking is often taught as “this is how TCP/IP works”

Pet peeve

- Networking is often taught as “this is how TCP/IP works”
- Instead, show conceptual problems
  - Show lots of solutions
  - Compare them with various metrics
  - Show examples from other protocols (Appletalk, IPX, DECnet, ATM, X.25, Infiniband, …)

Empower Students

- Which papers get accepted is somewhat random

Empower Students

- Which papers get accepted is somewhat random
- If they don’t understand that, they’ll be crushed

Empower Students

- Which papers get accepted is somewhat random
- If they don’t understand that, they’ll be crushed
- Have them read
  - Excellent papers that were rejected multiple times, perhaps with the reviews
  - Horrible papers published in good venues

Writing Books

- Incredibly educational and satisfying
- My philosophy on books
  - Describe things conceptually, with alternatives and tradeoffs
  - Describe ideas even if they are no longer deployed, or never were

Writing Books

- Incredibly educational and satisfying
- My philosophy on books
  - Describe things conceptually, with alternatives and tradeoffs
  - Describe ideas even if they are no longer deployed, or never were
- But…it is much more difficult to have successful books anymore

Network Protocols

Network Protocols

- A lot of what we all know…

Network Protocols

- A lot of what we all know…is false!

Things are so confusing

- Comparing technology A vs B
  - Nobody knows both of them
  - Somebody mumbles some vague marketing thing, and everyone repeats it
  - Both A and B are moving targets

Ways of comparing

- How about doing measurements?
  - Compare A vs B for power, latency, throughput, price….

No!

- Unless you have a sanity check about what intrinsic difference between A and B can account for the measurement…all you are testing is one implementation of A vs one implementation of B

Making Decisions

- Eliminate the “My proposal” vs “Your Proposal” complication

Making Decisions

- Instead, try to find the intrinsic differences between the technologies
- Then try to find as many arguments as possible for each variation
- Get people who disagree to have a discussion in the same room, with others who are neutral
- Write down all the arguments, and make sure everyone agrees their arguments have been captured
- Then, come up with a joint proposal

How do you compare technologies?

- Problem: technologies evolve. What exactly is “Ethernet”, for instance
- Hard to compare intrinsic differences between A and B if they can change
- Important to separate issues that are orthogonal
- If one compares existing products of A vs B, and test shows, e.g., A is lower power, then it’s good to sanity-check to see what ***intrinsic*** difference of A vs B might make that true.

Networking is really confusing

- What exactly is Ethernet?
- Why do we need both Ethernet and IP?
- What is this whole “layer 3 vs layer 2” thing about?

Perlman's View of Network Layers

- Based on OSI layers…

Perlman's View of Network Layers

- Layer 1: Physical

Perlman's View of Network Layers

- Layer 1: Physical
- Layer 2: Data Link: Neighbor-neighbor

Perlman's View of Network Layers

- Layer 1: Physical
- Layer 2: Data Link: Neighbor-neighbor
- Layer 3: Network: create path, forward

Perlman's View of Network Layers

- Layer 1: Physical
- Layer 2: Data Link: Neighbor-neighbor
- Layer 3: Network: create path, forward
- Layer 4: "Transport": end-to-end reordering, error recovery

Perlman's View of Network Layers

- Layer 1: Physical
- Layer 2: Data Link: Neighbor-neighbor
- Layer 3: Network: create path, forward
- Layer 4: "Transport": end-to-end reordering, error recovery
- Layers 5 and above:

Perlman's View of Network Layers

- Layer 1: Physical
- Layer 2: Data Link: Neighbor-neighbor
- Layer 3: Network: create path, forward
- Layer 4: "Transport": end-to-end reordering, error recovery
- Layers 5 and above: boring!

Definitions

- Repeater: layer 1 relay

Definitions

- Repeater: layer 1 relay
- Bridge: layer 2 relay

Definitions

- Repeater: layer 1 relay
- Bridge: layer 2 relay
- Router: layer 3 relay

Definitions

- Repeater: layer 1 relay
- Bridge: layer 2 relay
- Router: layer 3 relay
- OK: What is layer 2 vs layer 3?

Definitions

- Repeater: layer 1 relay
- Bridge: layer 2 relay
- Router: layer 3 relay
- OK: What is layer 2 vs layer 3?
  - The “right” definition: layer 2 is neighbor-neighbor. “Relays” should only be in layer 3!

Definitions

- Repeater: layer 1 relay
- Bridge: layer 2 relay
- Router: layer 3 relay
- OK: What is layer 2 vs layer 3?
- True definition of a layer n protocol: *Anything designed by a committee whose charter is to design a layer n protocol*

So, what’s the difference between layer 2 and layer 3?

Original Ethernet Invention

- CSMA/CD
  - CS: carrier sense
    - Don’t interrupt if someone’s talking!
  - MA: multiple access
    - You are sharing the airwaves so be polite!
  - CD: collision detect
    - If someone else starts talking while you are talking, stop talking—people can’t listen to multiple people talking at once!

CSMA/CD

- Lots of papers about limited “goodput” due to collisions
- Limited scalability (distance, number of stations)

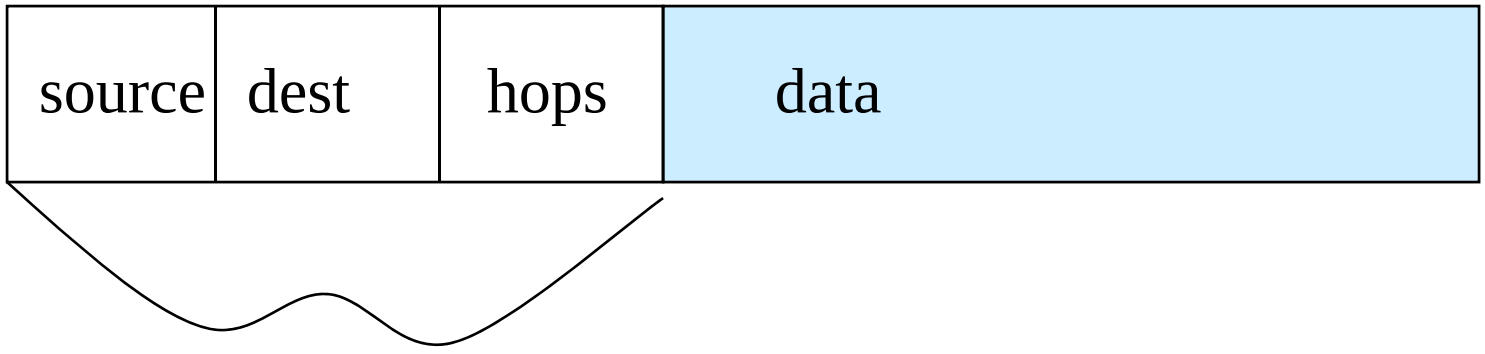
CSMA/CD

- Lots of papers about limited “goodput” due to collisions
- Limited scalability (distance, number of stations)
- But Ethernet hasn’t been CSMA/CD for years!

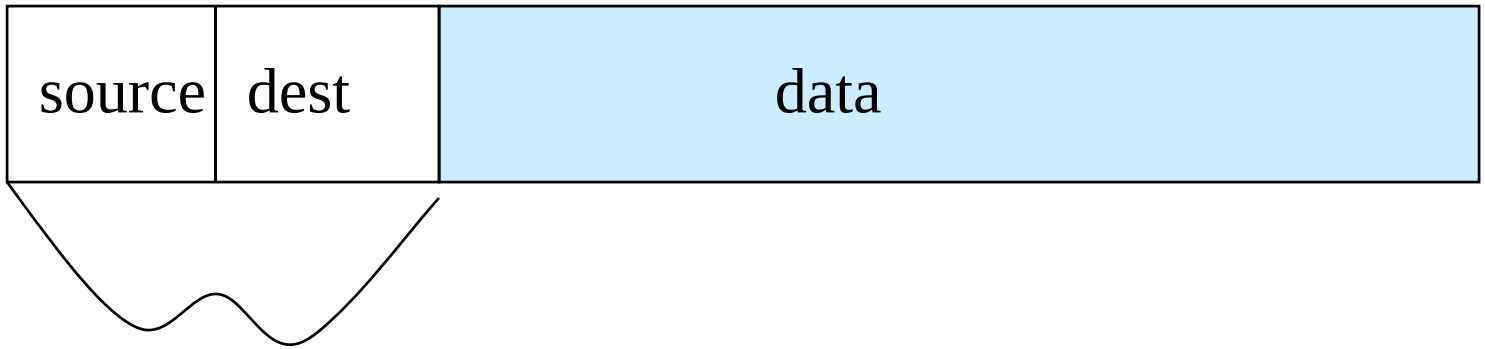
Layer 3 (e.g., IPv4, IPv6, DECnet, Appletalk, IPX, etc.)

- Put source, destination, hop count on packet
- Then along came “the EtherNET”
  - rethink routing algorithm a bit, but it’s a link not a NET!
- The world got confused. Built on layer 2
- I tried to argue: “*But you might want to talk from one Ethernet to another*!”
- “*Which will win? Ethernet or DECnet*?”

Layer 3 packet



Ethernet packet



Autoconfiguration

- Ethernet philosophy: plug and play
- Worst part of configuration: addresses
- They wanted each device to have its own address
- Decided on 6 byte addresses, even though the technology as originally invented was only for connecting, say, 1000 nodes

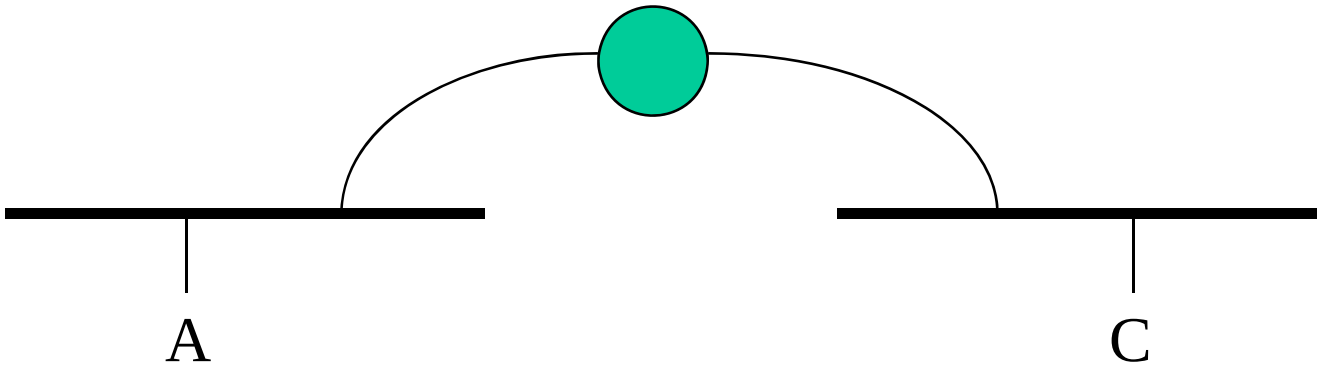
Unique addresses

- Two proposals
  - Pick an address at random
  - Administer them centrally (now done by IEEE) and have manufacturer created devices with permanent addresses in ROM

So where did Ethernet bridges come from?

Problem Statement

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



Why routers won’t work

- Router knows about one layer 3 protocol
- And the endnode has to implement that!

Constraint at that time for “magic box at layer 2”

- Must not modify Ethernet packet in any way
- Hard limit on size of packet

Basic idea

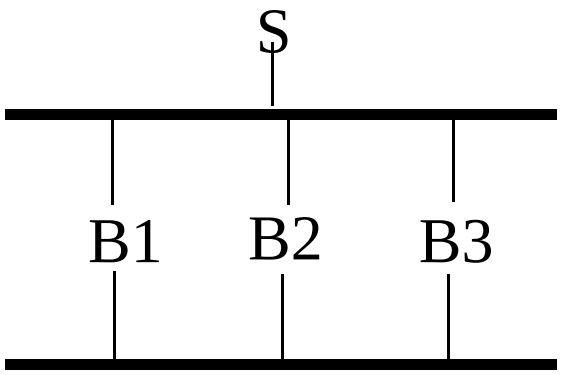
- Listen promiscuously
- Learn location of source address based on source address in packet and port from which packet received
- Forward based on learned location of destination

What's different between this and a repeater?

- no collisions
- with learning, can use more aggregate bandwidth than on any one link
  - Repeater forwards immediately…can't look at destination address before forwarding
- no artifacts of LAN technology (# of stations in ring, distance of CSMA/CD)

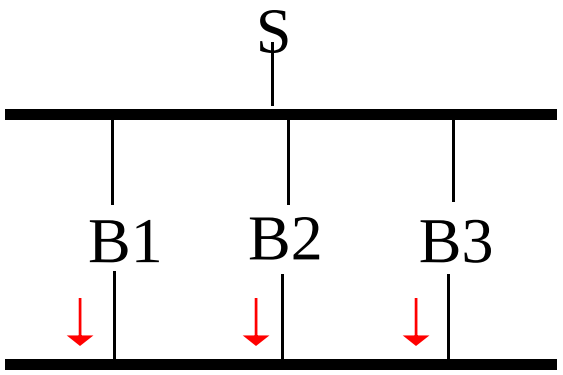
But loops are a disaster

- No hop count
- Exponential proliferation



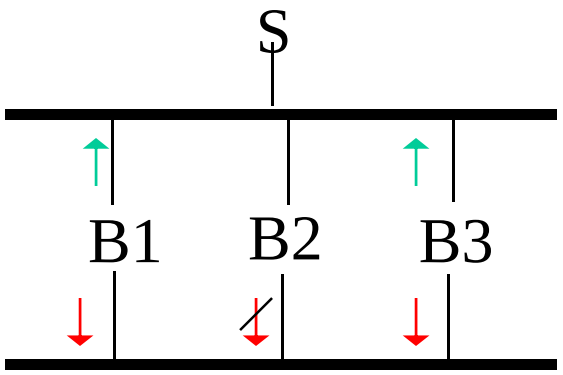
But loops are a disaster

- No hop count
- Exponential proliferation



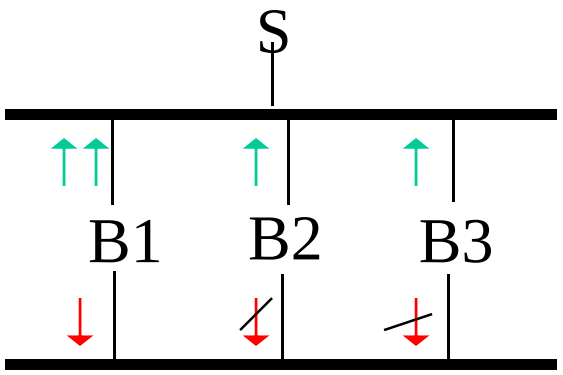
But loops are a disaster

- No hop count
- Exponential proliferation



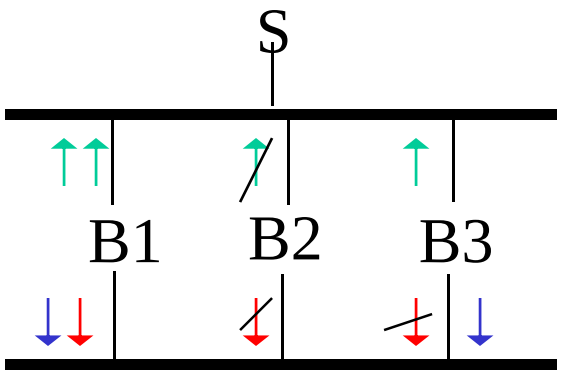
But loops are a disaster

- No hop count
- Exponential proliferation



But loops are a disaster

- No hop count
- Exponential proliferation



What to do about loops?

- Just say “don’t do that”
- Or, spanning tree algorithm
  - Bridges gossip amongst themselves
  - Compute loop-free subset
  - Forward data on the spanning tree
  - Other links are backups

Algorhyme

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

A tree whose crucial property
Is loop-free connectivity.

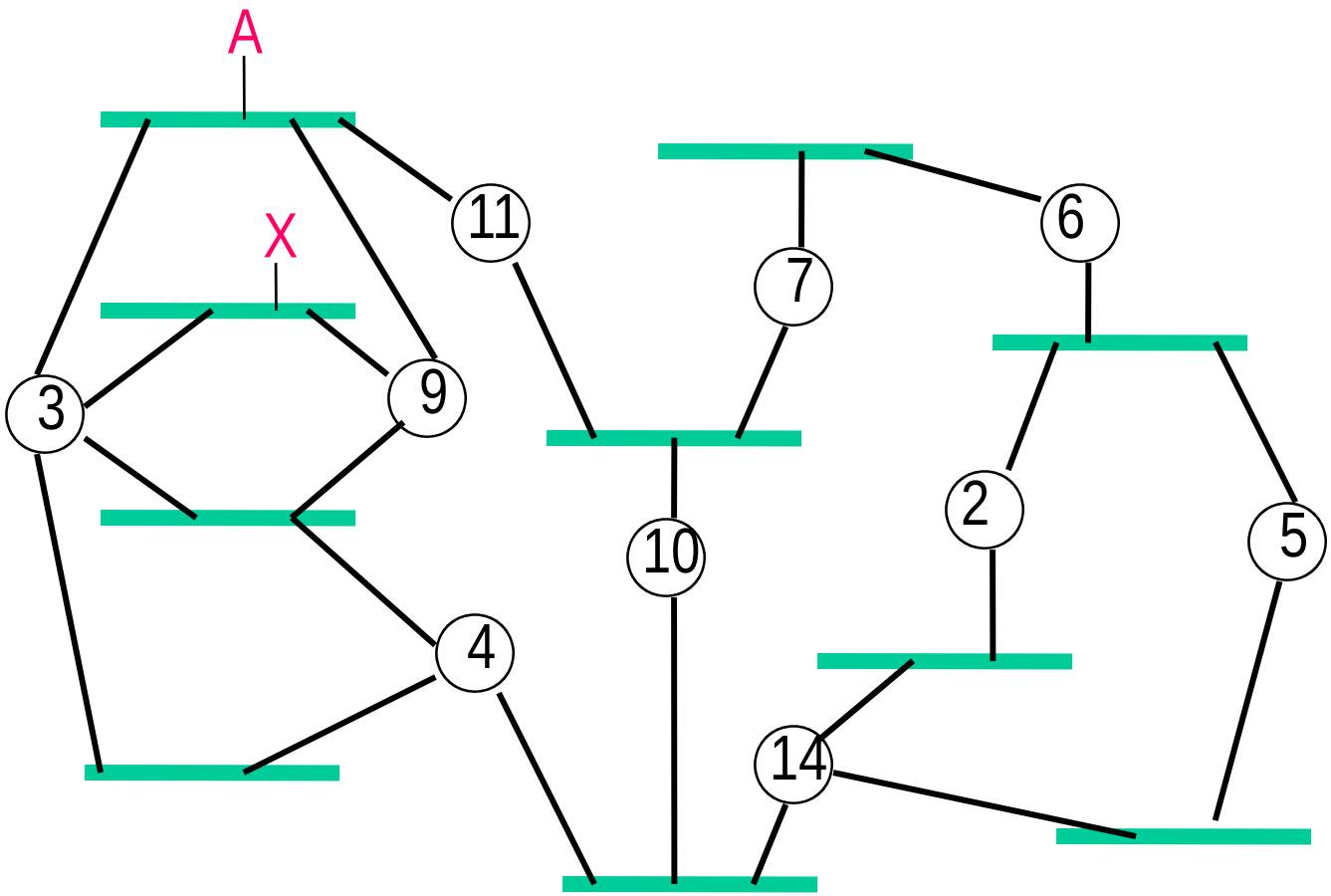
A tree which must be sure to span
So packets can reach every LAN.

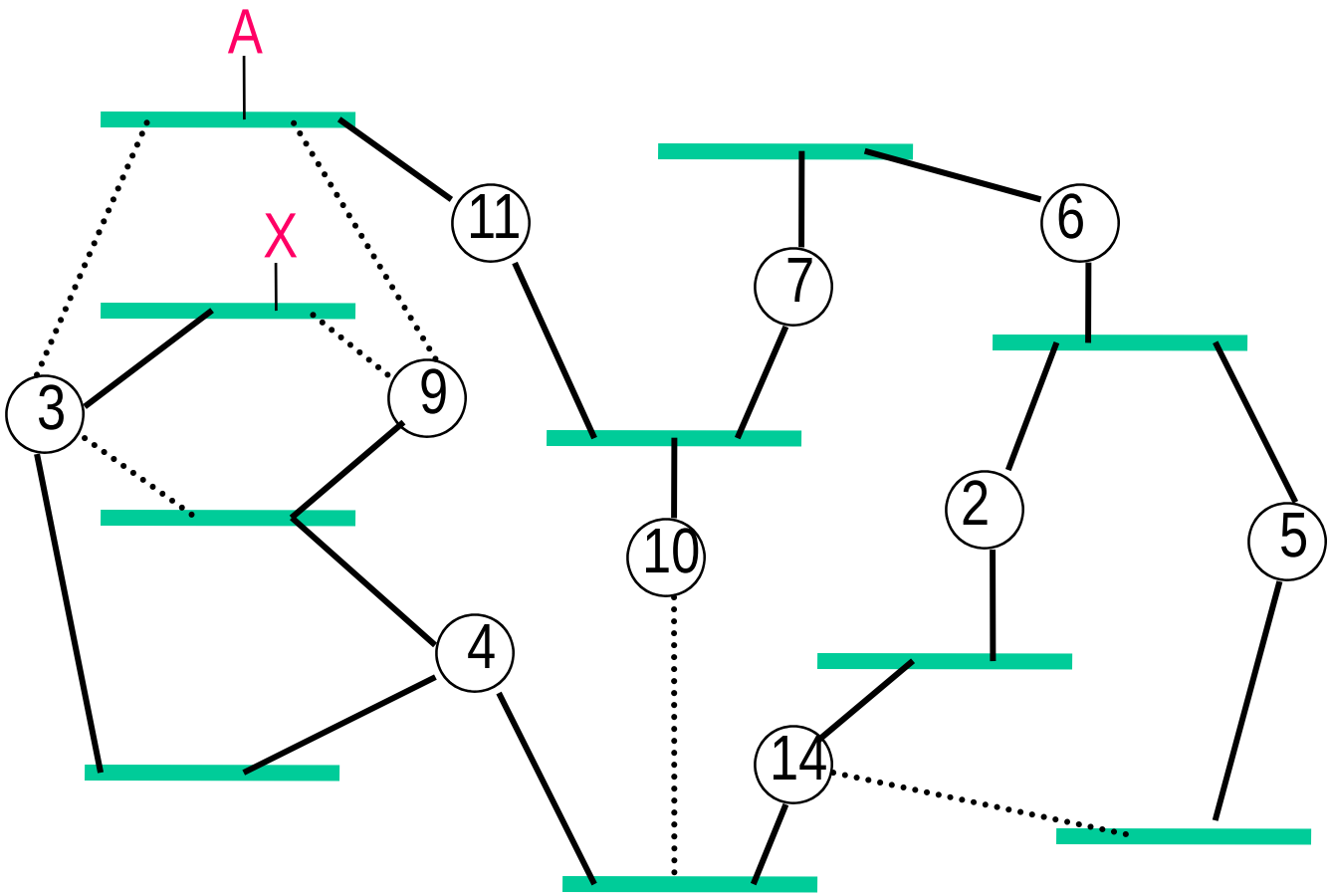
First the Root must be selected
By ID it is elected.

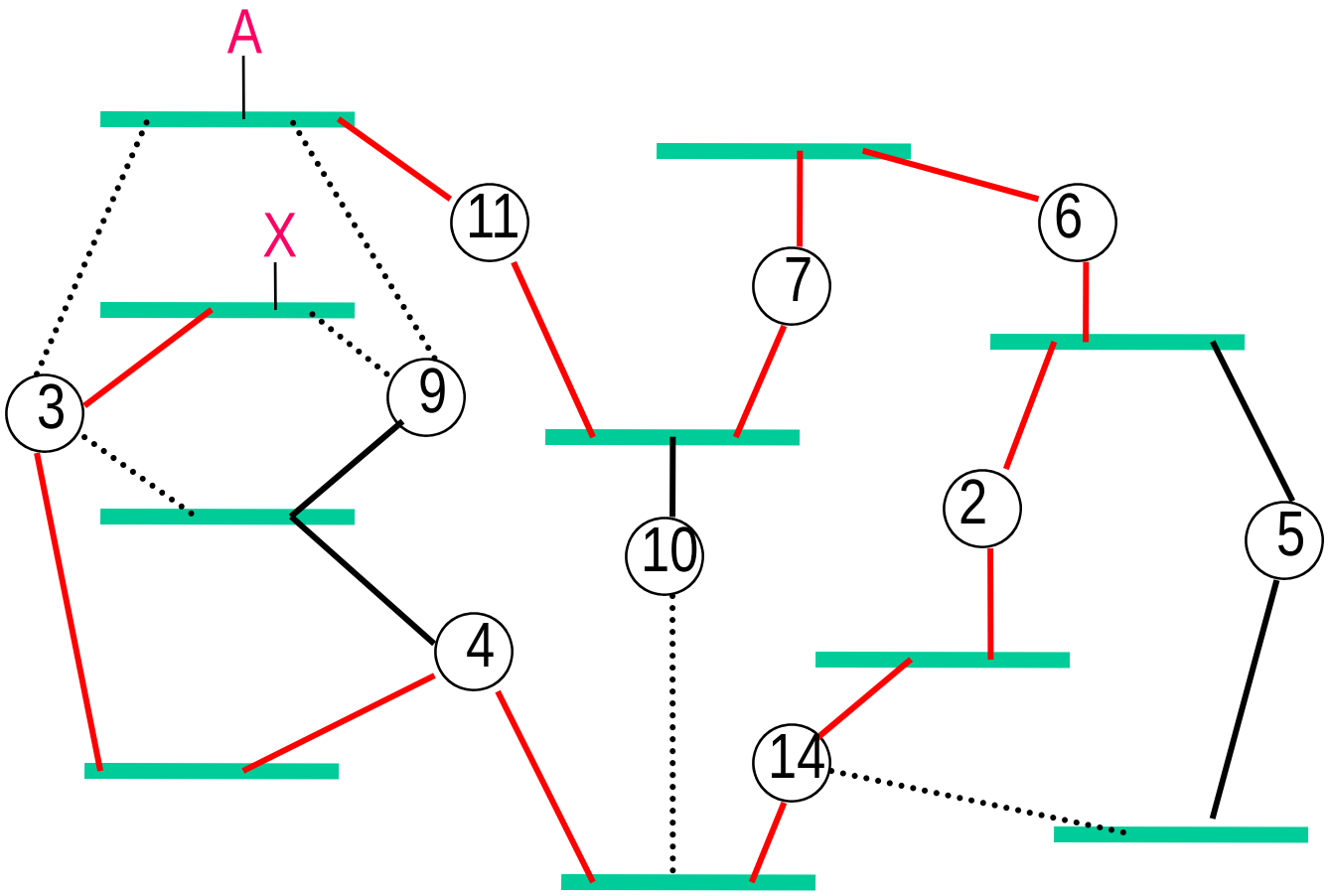
Least cost paths from Root are traced
In the tree these paths are placed.

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

Radia Perlman


A
X
11
7
6
3
9
2
5
10
4
14

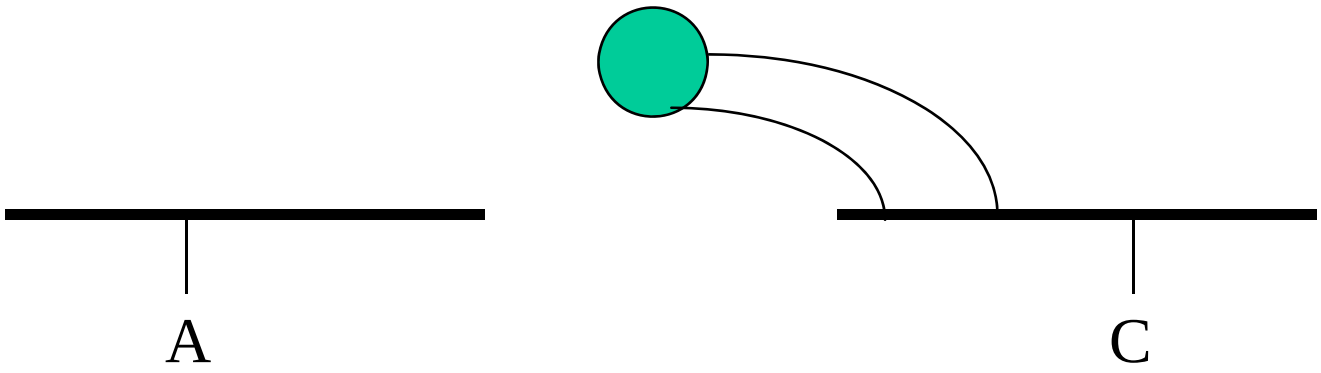

A
X
11
6
7
3
9
2
5
10
4
14


A
X
11
7
6
3
9
10
2
5
4
14

Bother with spanning tree?

- Maybe just tell customers “don’t do loops”
- First bridge sold...

First Bridge Sold



So Bridges were a kludge, digging out of a bad decision

- Why are they so popular?
  - plug and play
  - simplicity
  - high performance
- Will they go away?
  - because of idiosyncracy of IP, need it for lower layer.

Note some things about bridges

- Certainly don’t get optimal source/destination paths
- Temporary loops are a disaster
  - No hop count
  - Exponential proliferation
- Inherently unstable
- But they are wonderfully plug-and-play

Why not just use IP routers?

- World has converged to IP as layer 3, and it’s in the network stacks

Why not just use IP routers?

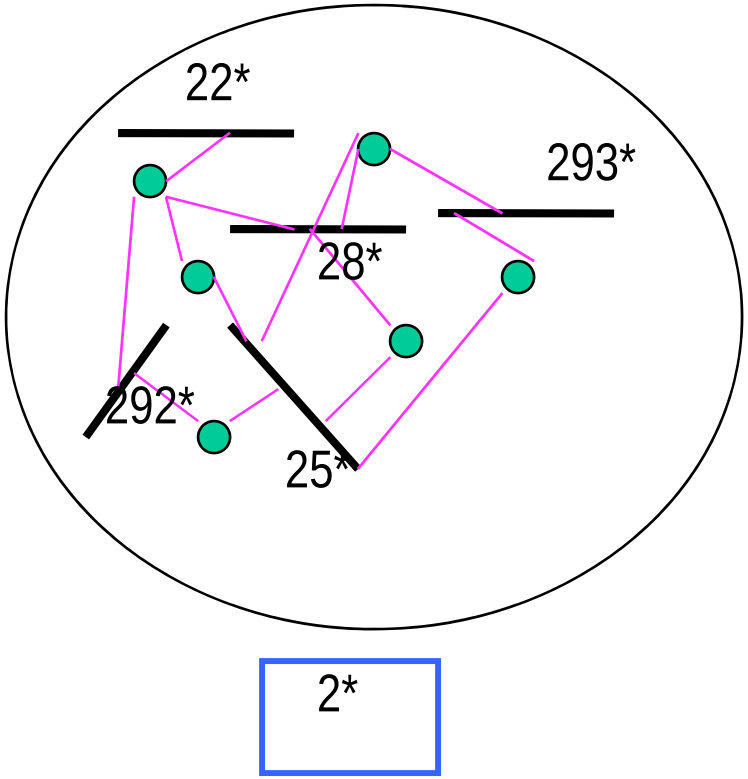
- IP is configuration intensive, moving VMs disruptive
  - IP protocol requires every link to have a unique block of addresses
  - Routers need to be configured with which addresses are on which ports
  - If something moves, its address changes

Layer 3 doesn’t have to work that way!

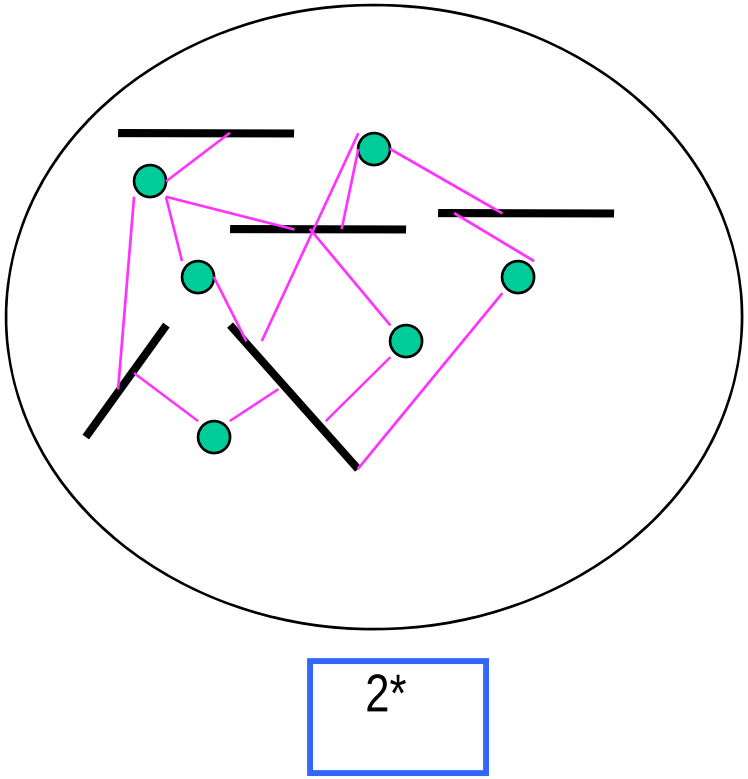
- CLNP / DECnet
  - Bottom level of routing is a whole cloud with the same prefix
  - Routing is to endnodes inside the cloud
  - Enabled by “ES-IS” protocol, where endnodes periodically announce themselves to the routers
  - Also in ES-IS: routers announce themselves to endnodes…

Hierarchy

One prefix per link



One prefix per campus



Worst decision ever

- 1992…Internet could have adopted 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, so CLNP gave understandable advantages
- CLNP still has advantages over IPv6 (e.g., large multilink level 1 clouds)

Ethernet looks like a single IP link

- So Ethernet provides a large cloud in which switches can autoconfigure, and nodes (e.g., VMs) can move around transparently
- But don’t want limitations of spanning tree

Next step in evolution: TRILL

TRILL

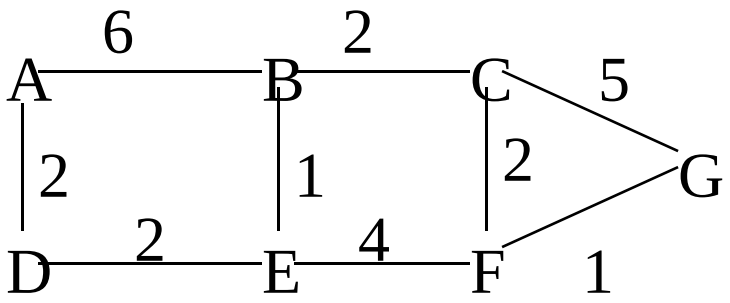
- TRansparent Interconnection of Lots of Links
- Basic idea: Put Ethernet in another envelope that acts more like a layer 3 envelope, and can be routed

Form network of TRILL switches

- TRILL switches find each other if:
  - Directly connected with pt-to-pt
  - Both connected to same Ethernet island
- Do “link state protocol” among TRILL switches to calculate paths to other TRILL switches

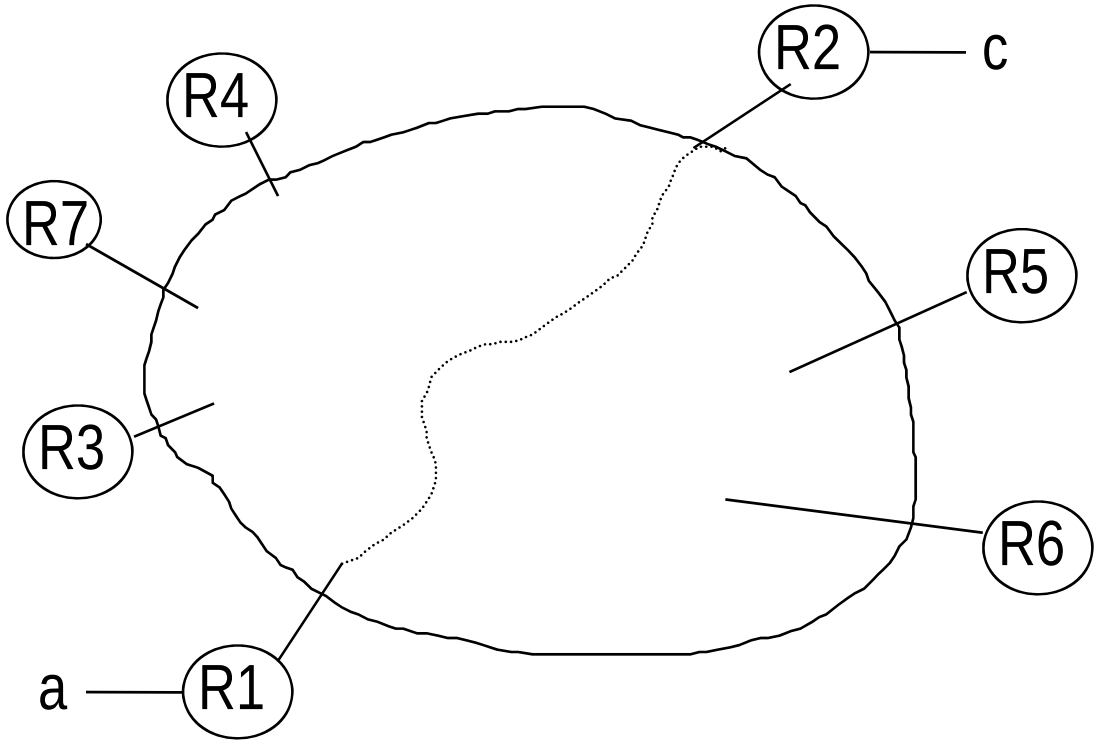
Link State Routing

- meet nbrs
- Construct Link State Packet (LSP)
  - who you are
  - list of (nbr, cost) pairs
- Broadcast LSPs to all rtrs (“a miracle occurs”)
- Store latest LSP from each rtr
- Compute Routes (breadth first, i.e., “shortest path” first—well known and efficient algorithm)

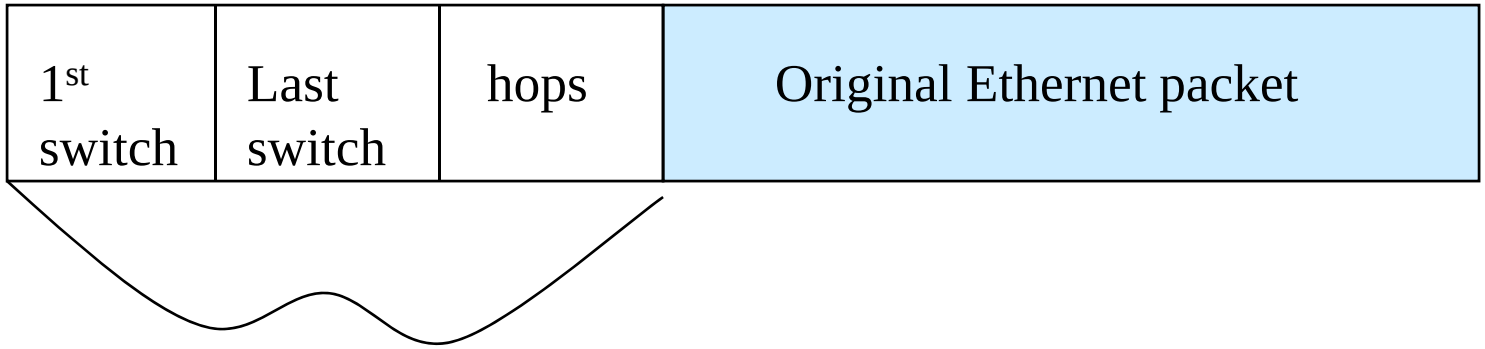


A	B	C	D	E	F	G
B/6	A/6	B/2	A/2	B/1	C/2	C/5
D/2	C/2	F/2	E/2	D/2	E/4	F/1
	E/1	G/5		F/4	G/1	

TRILL



TRILL packet



TRILL header
Switch addresses are 16 bits

Advantage of extra header

- Switches inside cloud don't need to know about all the endnodes…just how to reach all the other switches
- The outer header is like a layer 3 header, and can use all the layer 3 techniques, e.g.,
  - Shortest paths
  - Multiple paths (exploit parallelism)
  - Traffic engineering

Forwarding within TRILL cloud

- Switches autoconfigure their own 16-bit nickname, and routes to all other switches
- Forwarding tables O(#switches)
- Simple forwarding lookup (16 bits)
- 16 bits = 64000 switches
  - And of course orders of magnitude more endnodes (VMs) than switches
  - So it easily scales to a data center

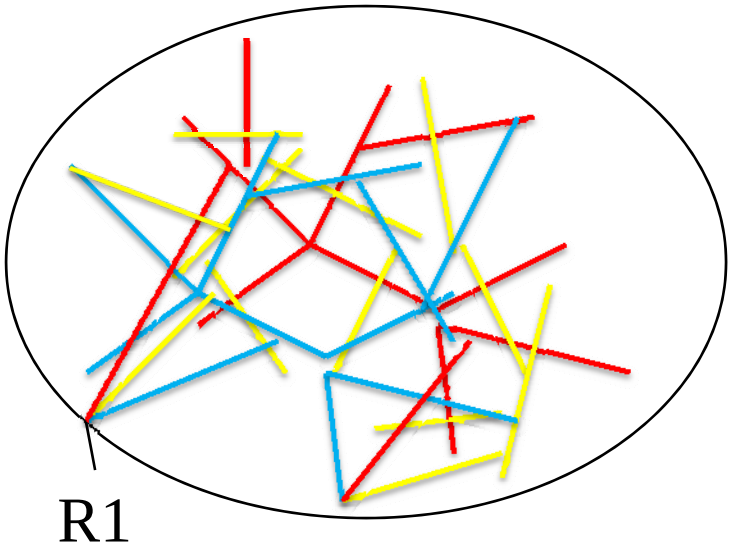
How did R1 know R2 was correct “last switch”?

- Autoconfiguring method:
  - If you don’t know, send through a tree to all edges (in that VLAN)
  - When decapsulating, remember (source MAC, ingress switch)
- Other methods
  - Configuration
  - Directory
  - Location-dependent MACs

TRILL and Multicast

- For spreading multicast traffic around, campus computes several trees
- “Last TRILL switch” field in TRILL header specifies which tree to send on
- Traffic filtered in the core based on VLAN, and IP multicast addresses

Multiple trees for multicast



R1 specifies which tree (yellow, red, or blue)

Note: TRILL is evolutionary

- Endnodes just think it's Ethernet…no changes
- Even interworks with existing spanning tree switches
- The more switches you upgrade to TRILL, the better the bandwidth utilization
- This could have been implemented by a single vendor, without standardizing

Algorhyme v2

I hope that we shall one day see
A graph more lovely than a tree.

A graph to boost efficiency
While still configuration-free.

A network where RBridges can
Route packets to their target LAN.

The paths they find, to our elation,
Are least cost paths to destination.

With packet hop counts we now see,
The network need not be loop-free.

RBridges work transparently.
Without a common spanning tree.

Ray Perlner

TRILL “Cousins”

- Recently a bunch of other encapsulation formats have been proposed
  - NVGRE, VXLAN, “Shortest Paths Bridging”, “Software Defined Networking”
- All accomplish basically the same thing
- Some can’t do multicast, headers are bigger

Change of Topic: Heresy

Network Heresy

- TCP model of congestion is wrong
- We'll have better performance if we allow a fabric to:
  - misorder packets
  - drop packets due to congestion
- We should redesign endnodes to cope
- And ultimately get much better performance

What's wrong with TCP?

- Years of research assuming flows really long-lived
- Exponential decrease/additive increase of window size to settle into having n flows share one bottleneck equally
- Conservative toe-in-water when start so as not to take more than your share
- If this was ever true, no longer at these speeds

Wrap-up

- folklore of protocol design
- things too obvious to say, but everyone gets them wrong

Forward Compatibility

- Reserved fields
  - spare bits
  - ignore them on receipt, set them to zero. Can maybe be used for something in the future
- TLV encoding
  - type, length, value
  - so can skip new TLVs
  - maybe have range of T's to ignore if unknown, others to drop packet

Forward Compability

- Make fields large enough
  - IP address, packet identifier, TCP sequence #
- Version number
  - what is “new version” vs “new protocol”?
    - same lower layer multiplex info
  - therefore, must always be in same place!
  - drop if version # bigger

Fancy version # variants

- Might be security threat to trick two Vn nodes into talk V(n-1)
- So maybe have “highest version I support” in addition to “version of this packet”
- Or just a bit “I can support higher” (we did this for IKEv2)
- Maybe have “minor version #”, for compatible changes. Old node ignores it

Version

- Nobody seems to do this right
- IP, IKEv1, SSL unspecified what to do if version # different. Most implementations ignore it.
- SSL v3 moved version field!
  - v2 sets it to 0.2. v3 sets (different field) to 3.0.
  - v2 node will ignore version number field, and happily parse the rest of the packet

Parameters

- Minimize these:
  - someone has to document it
  - customer has to read documentation and understand it
- How to avoid
  - architectural constants if possible
  - automatically configure if possible

Settable Parameters

- Make sure they can’t be set incompatibly across nodes, across layers, etc. (e.g., hello time and dead timer)
- Make sure they can be set at nodes one at a time and the net can stay running

Parameter tricks

- IS-IS
  - pairwise parameters reported in “hellos”

Parameter tricks

- IS-IS
  - pairwise parameters reported in “hellos”
- Kind of obvious…but OSPF copied IS-IS and what they did is:

Parameter tricks

- IS-IS
  - pairwise parameters reported in “hellos”
- Kind of obvious…but OSPF copied IS-IS and what they did is:
  - If neighbor’s Hello value is not identical to yours…refuse to talk!!

Summary

- If things aren’t simple, they won’t work
- Good engineering requires understanding tradeoffs and previous approaches.
- It’s never a “waste of time” to answer “why is something that way”
- Don’t believe everything you hear
- Know the problem you’re solving before you try to solve it!