Myths, Missteps, and Folklore in Network Protocols

Radia Perlman
Sun Microsystems Laboratories

Messages

- Dispel myths and “religion”
  - *“It’s not what you don’t know that’ll get you. It’s what you do know that ain’t true”*
    Mark Twain
- Learn from mistakes
- Learn from cool ideas
- Be provocative. Start lively discussion

Bridges, Routers, and Switches! Oh my!

- This discussion sheds light on how/why things work today
- Need the background for some other examples

Why this whole layer 2/3 thing?

- Myth: bridges/switches simpler devices, designed before routers
- OSI Layers
  - 1: physical

Why this whole layer 2/3 thing?

- Myth: bridges/switches simpler devices, designed before routers
- OSI Layers
  - 1: physical
  - 2: data link (nbr-nbr)

Why this whole layer 2/3 thing?

- Myth: bridges/switches simpler devices, designed before routers
- OSI Layers
  - 1: physical
  - 2: data link (nbr-nbr)
  - 3: network (create entire path)

Why this whole layer 2/3 thing?

- Myth: bridges/switches simpler devices, designed before routers
- OSI Layers
  - 1: physical
  - 2: data link (nbr-nbr)
  - 3: network (create entire path)
  - 4 end-to-end

Why this whole layer 2/3 thing?

- Myth: bridges/switches simpler devices, designed before routers
- OSI Layers
  - 1: physical
  - 2: data link (nbr-nbr)
  - 3: network (create entire path)
  - 4 end-to-end
  - 5 and above: boring

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?
- True definition of a layer n protocol: *Anything designed by a committee whose charter is to design a layer n protocol*

Layer 3 (DECnet, IP)

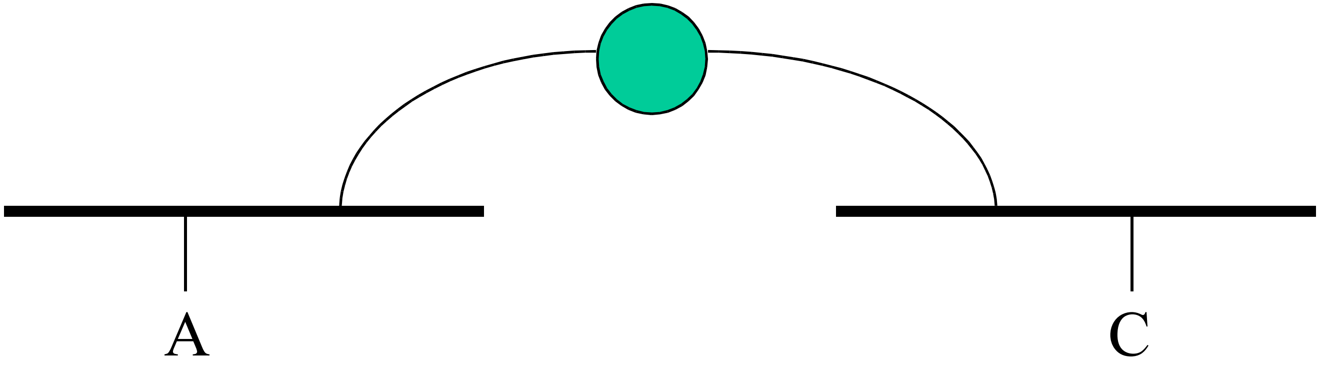
- Put source, destination, hop count on packet
- At the time DECnet was more prevalent, but it’s logically equivalent to IP
- Then along came “the EtherNET”
  - rethink routing algorithm a bit, but it’s a link!
- The world got confused. Built on layer 2
- I tried to argue: “*But you might want to talk from one Ethernet to another*!”
- “*Which will win? Ethernet or DECnet*?”

Horrible terminology

- Local area net
- Subnet
- Ethernet
- Internet

Problem Statement

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



Basic idea

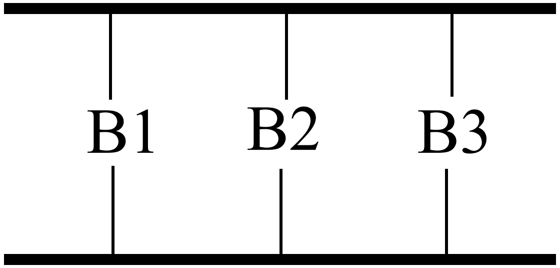
- Listen promiscuously
- Learn location of source address based on source address in packet and port from which packet received
- Forward based on learned location of destination

What’s different between this and a repeater?

- no collisions
- with learning, can use more aggregate bandwidth than on any one link
- no artifacts of LAN technology (# of stations in ring, distance of CSMA/CD)

But loops are a disaster

- No hop count
- Exponential proliferation



Thus the Spanning Tree Algorithm

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.

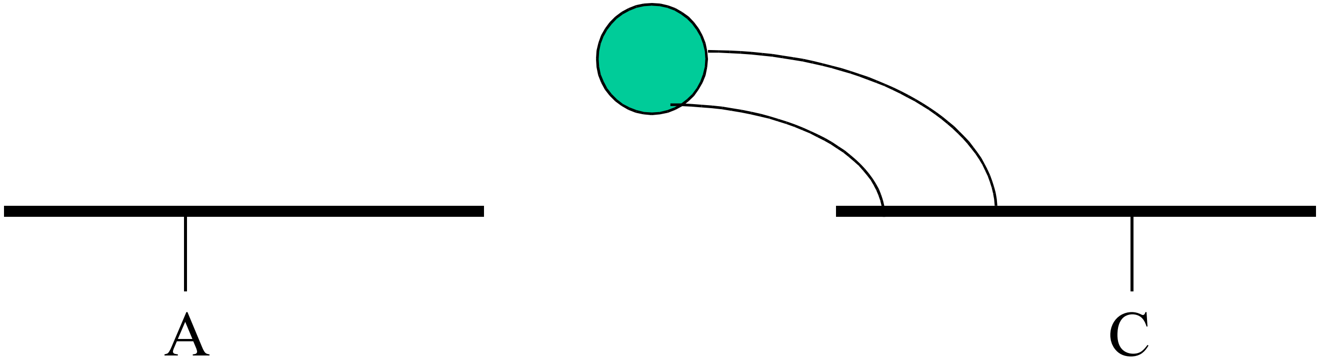
Least cost paths from Root are traced
In the tree these paths are placed.

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

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

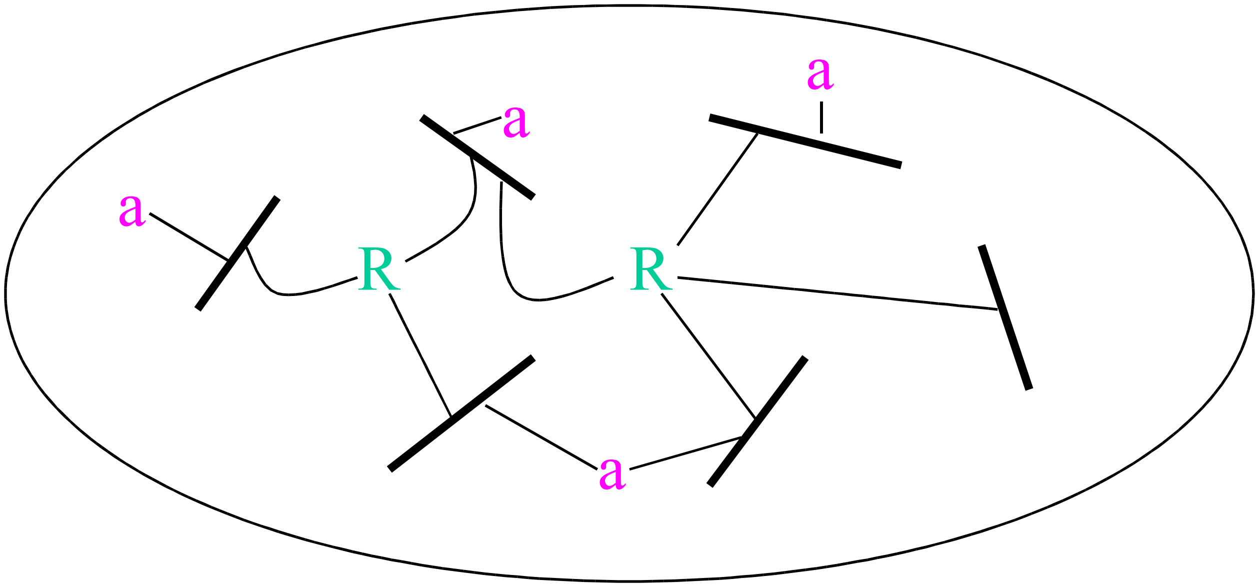
- Why are they so popular?
  - – plug and play
  - – simplicity
  - – high performance
- Will they go away?
  - – because of idiosyncracy of IP, need it for lower layer. Wouldn’t have needed that for CLNP

Layer 3 Hierarchy

- In IP, each link has a prefix
  - If you have multiple links, you have multiple IP addresses
  - If you move to a different link, your IP address changes
- In CLNP, “area” has prefix
  - within area, can move, and have multiple links, and routing will take best path to you
  - “level 2 routing”: like IP, longest prefix match
  - “level 1 routing”: to specific node within area
- bridges serve as “level 1 routing” for IP

CLNP "level one"

One prefix for entire campus



Inside campus, route directly to endnode's unique address
Endnodes announce location periodically
Routers tell each other which endnodes they connect to

Plug for RBridges

- New WG in IETF: TRILL
  - TRansparent Interconnection of Lots of Links
- Similar to level 1 routing for CLNP, but without help from the endnodes
- Will combine best features of bridges and routers
- Join mailing list www.postel.org/rbridge

Myth

- Ethernet continues to be a successful technology

So what is Ethernet?

- CSMA/CD, right? Not any more, really...
- source, destination (and no hop count)
- limited distance, scalability (not any more, really)

Switches

- Ethernet used to be bus
- Easier to wire, more robust if star (one huge multiport repeater with pt-to-pt links
- If store and forward rather than repeater, and with learning, more aggregate bandwidth
- Can cascade devices…do spanning tree
- We're reinvented the bridge!

Stuff too obvious to say

- What’s a version number?
- Coordinating parameter settings

What’s a version?

- What’s the difference between a “new protocol” and a “new version” of an existing protocol?

What’s a version?

- What’s the difference between a “new protocol” and a “new version” of an existing protocol?
- Is IPv6 a “new version” of IP?

What’s a version?

- What’s the difference between a “new protocol” and a “new version” of an existing protocol?
- Is IPv6 a “new version” of IP?
- Would CLNP have been a “replacement” of IP?

My definition

- Same protocol – same layer n protocol type (e.g., Ethertype)
- New version – incompatible with current version

But what if you want to add compatible changes?

- Major/minor version number
- Use reserved fields properly
- TLV encoding (type/length/value)
  - Skip over unknown T’s
- So – only increment version number if incompatible

Logical conclusion

- Have to specify more than “set this field to 4”
- You need to say “throw away the packet if it’s not 4”
- And future versions must leave that one field (version number) in the same place

Do they do this?

- IPv4
  - Just says “set this to 4”
  - So implementations ignore it if it’s 6
  - So IPv6 can’t use same Ethertype
  - So…IPv6 is not a “new version” of IPv4

Do they do this?

- IPv4
  - Just says “set this to 4”
  - So implementations ignore it if it’s 6
  - So IPv6 can’t use same Ethertype
  - So…IPv6 is not a “new version” of IPv4
- IPv6
  - They must have learned their lesson, right?

Do they do this?

- IPv4
  - Just says “set this to 4”
  - So implementations ignore it if it’s 6
  - So IPv6 can’t use same Ethertype
  - So…IPv6 is not a “new version” of IPv4
- IPv6
  - They must have learned their lesson, right?
  - No…IPv6 says “set this field to 6”

SSL

- Version 3 totally moved all the fields around from version 2
- And wanted to use the same ports

SSL

- Version 3 totally moved all the fields around from version 2
- And wanted to use the same ports
- Version 2 just says “set this to 2”

SSL

- Version 3 totally moved all the fields around from version 2
- And wanted to use the same ports
- Version 2 just says “set this to 2”
- And….version 3 even moved the version number field!
- And they use the same ports

So how does it work?

- First pkt in v2 format, setting version to “3”

So how does it work?

- First pkt in v2 format, setting version to “3”
- And just for a final irony:
  - V2 is specified as 0.2
  - V3 is specified as 3.0

So how does it work?

- First pkt in v2 format, setting version to “3”
- And just for a final irony:
  - V2 is specified as 0.2
  - V3 is specified as 3.0
- So version 2 node receives what it thinks is a version 768 packet, and doesn’t even blink

Next obvious thing: Parameters

- It is nice to avoid parameters
  - – Have to be documented
  - – Customer has to be intimidated
  - – Can be set wrong
- How to avoid
  - – Self-configuring nets
  - – Architectural constants

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

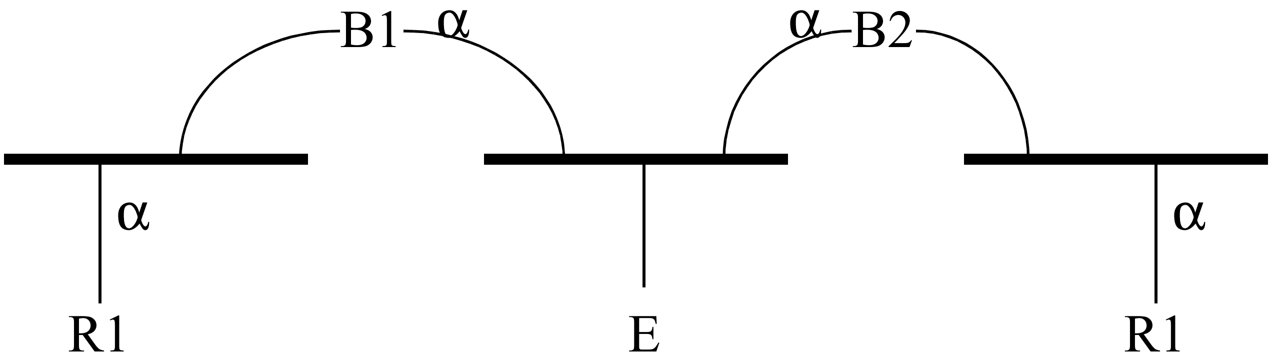
Parameter tricks

- IS-IS
  - pairwise parameters reported in “hellos”
  - area-wide parameters reported in LSPs
- OSPF
  - copied most of IS-IS, but got this wrong. Use field in hello to refuse to talk if not identical!
- Bridges
  - Use Root’s values, sent in spanning tree msgs

VRRP

- VRRP is a new protocol, and it makes the same mistake
- VRRP has an election among routers to choose who will be (layer 3 “R1”, layer 2 “x”)
- Bad for two routers to both think they are master

VRRP/Bridges/Multiple R1's



state if both R1's send msg at about the same time

VRRP

- Message says “this is my hello timer”
- Spec says “throw away message if the hello timer doesn’t agree with your configured value”

Random comments people make

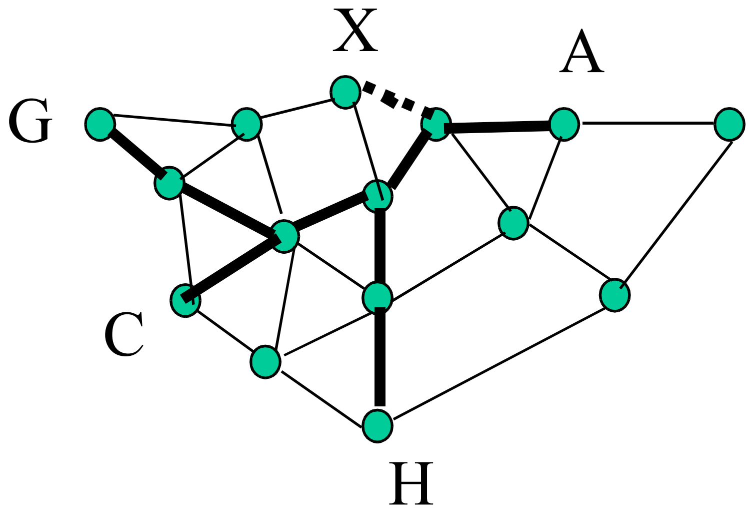
- PKI is dead
- Security is built into IPv6, but is just an add-on to IPv4
- If things are encoded in XML, everything will be interoperable

Things to rant about

- IP multicast
- BGP
- IPv6
- X.509

Multicast

- Ethernet: falls out of technology
- ATM: create VC. “Add member”



IP Multicast

- Idea: make it look “just like Ethernet”
  - globally unique multicast addresses
    - IP address 32 bits, top 4 bits=1110
  - anyone can request to listen. anyone can send without being a member
- So, start out with unchangeable “model”
  - signalling protocol to inform local rtr to send G

Problem: Can’t be implemented

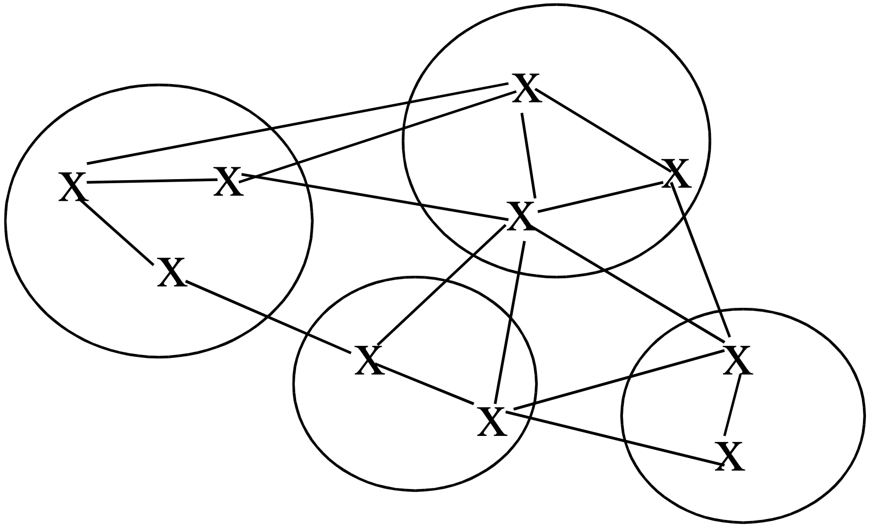
- various attempts:
  - flood and prune
    - send all data everywhere, in case someone in Albania wants to listen
    - if not interested, send “prune”
    - keep track of all (S,G) pairs nbr NOT interested in
  - MOSPF
    - routers keep track of all listeners for all groups

IP Multicast attempts

- Tree building like with ATM
    - send join towards Root
    - create tree
- Problems:
    - who is Root for G?
        - unscalable intradomain protocol to select a Root-candidate for G
    - how to administer addresses

IP Multicast

- So, came up with unscalable complex intradomain
- Then MSDP to piece domains together



How IP Multicast should look

- Two types
  - finding something (low bandwidth, can’t set up tree). Just flood with RPF
  - conference call, etc. Find host H. Build tree to H. Have address of group be (H,G), where G only has to be unique to H

BGP

- It’s an interdomain protocol

BGP

- It’s an interdomain protocol
- OK, what’s an interdomain protocol?

BGP

- It’s an interdomain protocol
- OK, what’s an interdomain protocol?
  - Interdomain: between domains
  - Intradomain: within a domain

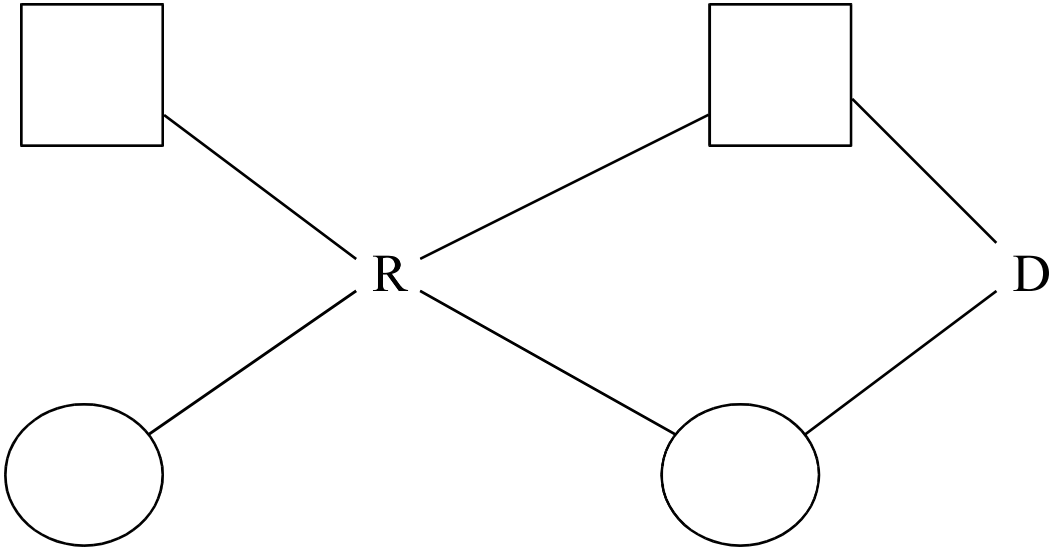
BGP

- It’s an interdomain protocol
- OK, what’s an interdomain protocol?
  - Interdomain: between domains
  - Intradomain: within a domain
- OK, what’s a domain?

BGP Configuration

- path preference rules
- which nbr to tell about which destinations
- how to “edit” the path when telling nbr N about prefix P (add fake hops to discourage N from using you to get to P)
- Possible policies that don’t converge
- Lots of theoretical problems, and in practice

Policies BGP Won't Support



Problems with BGP

- It only supports policies it happens to support
- It’s very configuration intensive
- Computation and bandwidth intensive
- Can have “incompatible policies”
  - Policies may not converge

What’s with IPv6?

- Had a perfectly good choice in 1992 (replace IP with CLNP, ISO’s connectionless layer 3 protocol)
- Result of not doing this
  - Internet might be too large and mission critical to ever migrate
  - no incentive for those that have IPv4 (in 1992, didn’t have DHCP, for instance)

Ironically, CLNP still better than IPv6

- Should at least steal good ideas (be nice if you didn’t first insult them, even better if have the grace to credit them)
- Could have had true zero-config routing in campus
- ES-IS less expensive, more robust than ND and VRRP

X.509

- It’s a format for a certificate
- What’s a certificate?
  - Name
  - Public key
  - Signature
- So what could be wrong?

Why was X.509 a poor choice

- ASN.1 encoding
  - Requires lots of code to parse
  - Certificates bigger than necessary

Why was X.509 a poor choice

- ASN.1 encoding
  - Requires lots of code to parse
  - Certificates bigger than necessary
  - I used to hate ASN.1 until I saw XML…

Why was X.509 a poor choice

- ASN.1 encoding
  - But it’s just syntax. Not really important
- Real problem is the name
  - Uses X.500 names.
  - Internet applications don’t use X.500 names
  - What good is a certificate mapping a key to a different string than the user typed?

Bad attitudes

- If we change directions now we’ll be throwing away 10 years worth of work”
- We don’t want “tourists”. If you haven’t been following the mailing list for the last 10 years and reading all our drafts, we don’t want to make it easy for you to catch up
- “If you don’t know that already you don’t belong in this group”
- Sports team mentality

Lessons

- Always seems easy to start over with new thing. Always takes longer and comes out worse.
- Start teaching this stuff like a science.
- Need calm technical discussions
- It’s never a waste of time to answer questions, rethink basic principles, prepare tutorial documents, summarize mailing list threads

Lessons

- Don’t cast something in stone before there is a plausible way of realizing it
- Don’t just dive in and start doing stuff. Think about what problem you’re solving before you try to come up with a solution.