ECE 435 – Network Engineering
Lecture 22

Vince Weaver

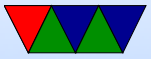
https://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

25 March 2026

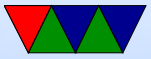
Announcements

- Don't forget HW#7, due Friday
- Also don't forget project groups/topics, also due Friday
  - Need e-mail with preliminary topic, and group members.
  - Only one e-mail per group needed



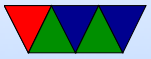
Related News

- Saw rumor Capital One granted 2630::/16, which is 1/65536th of the total IPv6 range



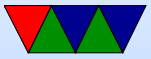
How do you get an IPv6 address?

- Manual (hard-coded)
- DHCPv6
- SLAAC



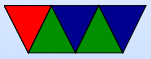
IPv6 Stateless Address Auto-Config (SLAAC)

- IPv6 StateLess Address AutoConfiguration (SLAAC) assumes on /64 subnet (so every subnet contains orders of magnitude more than the total IPv4 space for their own local network)
- Essentially large enough a system could just pick a random address and it would work



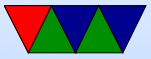
SLAAC Methods

- Three ways:
  - EUI-64 (RFC 4291) – based on MAC address
  - Stable Private (RFC 7217) – hash based, don't give away MAC
  - Privacy Extension Addresses (RFC 4941) – like above but change over time to preserve anonymity
    For security refresh daily, this does happen on MacOS/Windows, but not necessarily on Linux



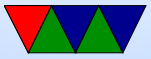
EUI-64 Link Local Example

- Linux seems to do this to set up link-local addresses
- Link-local is a non-routable IP address only used to your LAN (Local Area Network)
- IPv4 has these too but I've only ever seen Microsoft use them
- For IPv6 addresses they are on fe80::/10



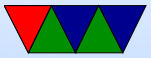
EUI-64 Link Local Example Continued

- Take MAC address (i.e. 8c:dc:d4:24:7d:45)
- Split up, put fffe in middle, flip bit 7 of top byte
  - 8c:dc:d4:24:7d:45
  - fe80::8edc:d4ff:fe24:7d45
- Can see these addresses with `ip addr`
- `ping6` can ping them on local network
- To ssh you have to specify interface, something like:
  `ssh fe80::8edc:d4ff:fe24:7d45%eth0`



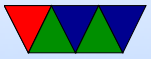
Duplicate Address Discovery (DAD)

- Once has link-local address, joins special multicast address
- `ff02::1:ffXX:XXXX` where last 6 bytes are bottom half of IP address it picks
- Sends packet to see if anyone else has address
- `ip maddr show` will show in-use multicast addresses



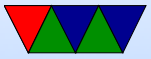
IPv6 Neighbor Discovery

- Neighbor Solicitation (NS) (RFC 4861) use with SLAAC described in RFC 4862 to get address
- Once has address, does DAD
- Once has link local address, sends out Router solicitation (RS) to multicast address ff02::2
- Router replies with (RA) router advertisement packet with info on router, maybe DNS, etc
- Now needs to get global routable address prefix: gets directly or has bits set to indicate it should use DHCPv6



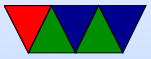
IPv6 DHCPv6

- Can provide info just like IPv4
- Not just router info, but also things like DNS servers, etc



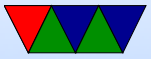
IPv6 Setup

- I've set up many many IPv4 networks, not any IPv6
- `https://lwn.net/Articles/831854/`
  Article by James Bottomley
- With IPv4, DHCP can take care of everything



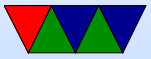
IPv6 setup issues

- It can be hard to subnet.
- It's recommended an ISP gives you a /56 but often they will just give you a /64
- That's a lot of addresses, but due to SLAAC it's assumed a network has a minimum of 2^{64} addresses so you can't split it up easily
- Annoying if you want multiple subnets at home (for wireless, DMZ, etc)



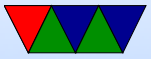
IPv6 Firewall

- IPv6 has the (now mostly obsolete) notion that every device with an IP address should be globally visible to the entire internet
- In order to stop this you need to have some sort of firewall
  - Linux has separate ipv4 and ipv6 firewalls
  - Having a NAT set up sort of gives you a firewall for free, you don't necessarily get that with IPv6



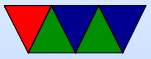
IPv6 Security Issues

- Shadow Networks – if you have a primarily ipv4 setup but various devices start up IPv6 connections without you realizing it
- Fragmentation – even though only on ends, can still have issues like IPv4 where it's hard to handle fragments as TCP port info and such only in first fragment

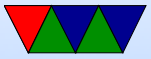


Modern IPv6 vs NAT Concerns

- Performance, NAT takes extra processing. Can small routers keep up at 1Gbps?
- Security, implicit security in NAT (internal devices not visible at all unless open outgoing connection). Can configure a firewall for ipv6 but requires extra work
  Also to get similar NAT-like behavior (blocked by default unless outgoing connection) maybe difficult
- Generally ipv6 not used by as many so dependent on your ISP not breaking things and not noticing

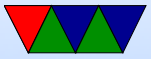


- With IPv6 it's possible your ISP can see how many devices you have behind your connection, it's a lot more difficult with NAT (at one point ISPs wanted to charge per-device)



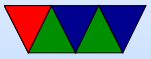
IPv4 / IPv6 Interop

- IPv6 NAT? What would that even mean?
- Can you have an internal network that's IPv4 connected to an external IPv6 network?
- Can you have an internal network that's IPv6 connected to an external IPv4 network?
  - Dual stack. Run both IPv4 and IPv6. Can fall back if one doesn't work. Need to configure two parallel network infrastructures.
  - Stateless IP/ICMP Translation



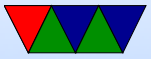
Internally fully IPv6, but each server has equivalent IPv4 on outside and the router converts them

- Tunneling, IPv6, tunnel/encapsulate inside of IPv4, then return to IPv6
- NAT64 – IPv6 internally, but has single external IPv4 gateway and does NAT/conversion to inside
- 464XLAT – nat64 at network level, SIIT internal
  Used in carrier-grade type situations, PLAT/CLAT



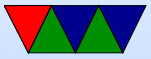
IPv6 Socket Programming

```
struct sockaddr_in6 server_addr;

sock_fd = socket(AF_INET6, SOCK_STREAM, IPPROTO_TCP);

server_addr.sin6_family = AF_INET6;

inet_pton(AF_INET6, "::1", &server_addr.sin6_addr);

server_addr.sin6_port = htons(SERVER_PORT);
```



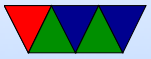
Multistack (IPv4 or IPv6) Sockets

- Can you open a socket that handles both IPv4 and IPv6?
- On most machines, yes
- Create IPv6 socket
- There's an option you can disable, IPV6_V6_ONLY
- With that disabled can accept incoming connections for either IPv4 or IPv6
- Linux (possibly) allows by default. Other OSes (including BSD and Windows) are IPv6 only by default.



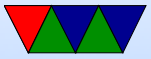
Multistack Sockers for Client

- You can do this too
- `getaddrinfo()` is smart enough when you lookup by name on DNS to in theory pick IPv6 or IPv4 for you and set up the various structs properly



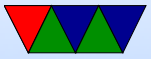
Sample code

- I actually do have some sample code
- TODO: in future migrate the homeworks to using IPv6 by default?



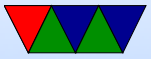
Hierarchical Routing

- Would you want to have all routers in network on flat network?
  Routing table would be a bit complex
- Split into a hierarchy
- Network made up of Autonomous Systems (AS)



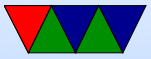
Autonomous System (AS)

- A network under control of one group, with one routing policy
- Inside an AS, interior routing, between is exterior routing.



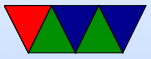
Autonomous System Numbers

- Traditionally were 16-bit numbers, but ran out. In 2007 expanded to 32-bit. X.Y (dotted decimal). Old 16-bit are 0.X
- Can look up, UMaine is AS557
  `https://bgpview.io/asn/557`



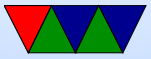
Can you get your own ASN?

- Similar to getting IP addresses.
- Usually you need to be a large enough group and be able to get some network connectivity
- Then you need to convince your ISP to add a route to your AS



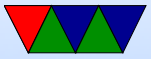
Routing

- Systems under same command (same ISP) use intra-domain routing protocol, or interior gateway protocol (IGP)
- Border routers connect to border routers of others
- Inter-domain routing, EGP (exterior gateway protocol)

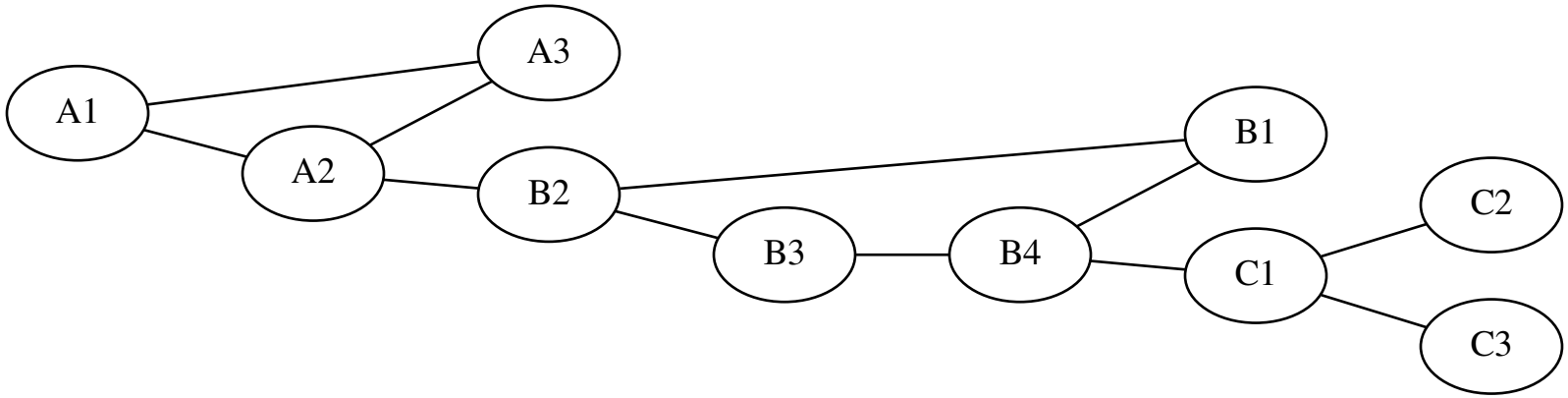


AS Types

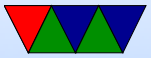
- Stub AS – like ISP with customers, one gateway to internet
- Multihomed AS – multiple gateways (why?) redundancy. traffic generally doesn't flow through
- Transit AS – traffic can flow through network
- Internet Exchange Point (IX/IXP) – where networks can meet up



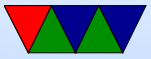
Diagram



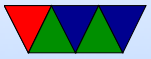
- Packet A1 - A3 internal A1 - B2 goes to border router and across, then local A1 - C2 goes to border router to B network, across local to B/C border, then finally to C
- If flat network, need to know 10 machines in routing table



- In hierarchical only need to communicate to 2-3 other routers, find way to border router

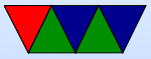


Intra-Domain Routing / Interior Gateway Protocols



Historical – RIP (Routing Information Protocol)

- Used by ARPAnet until 1979(?)
- by Xerox, included in BSD, routed RFC 2543
- distance vector routing, with hop count, max 15 hops
  - RIP advertisements over UDP port 52
  - Send advertisement every 30s, or when changes
  - Only sends to neighbors
- Routing table: dest, next hop, distance
- Algorithm



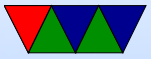
- Get table update
- Increment all hops by 1 (you're one hop away)
- Go down list.

  If route not in table, add it

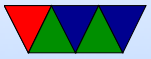
  If route there, and next hop same (but cost diff), replace it as this is new info

  If route there but cost less, replace it

- On power up, comes up with hard-coded routes and values of 1 and no next-hop. Can send packet to request immediate update from neighbors.
- Packet description



- Timers
  - Periodic timer, technically 30s, reality randomized between 25 and 35 (why?)
  - Expiration timer – 180s. If no update in this time, problem, hop count set to 16 (unreachable)
  - Garbage collection – 120s – once unreachable, advertise it as such for a while before removing so others notice
- Issues
  - Slow Convergence – a change in routing tables takes 30s per hop to propagate through



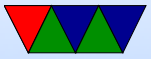
Part of why limited to 15 hops

- Instability – packets can be caught in loops. Ways to fix:

  Triggered update – send update info immediately, not wait 30s

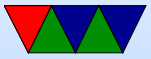
  Split Horizon – if a router sends you update info, don't send this back to it Poison Reverse – like split horizon, but when send back, mark as 16 the routes received from that interface.

- There was a RIP2



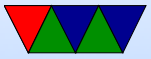
OSPF (Open Shortest Path First)

- successor to RIP. RFC2328 (5340 for IPv6)
- Idea of Areas inside of an AS. Split up into areas.
- Each area connected by backbone router
- Router on two areas is area border router
- Link-state Routing
  - State is flooded: when a change happens (and only then) it sends this state to all neighbors, which send to all neighbors, until the whole network receives it
  - each router uses Dijkstra to find least cost for self,

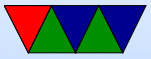


builds table

- Types of link
  - Point-to-point – routers directly connected
  - Transient Link – network with several routers can be simplified?
  - Stub Link – a network connected to only one router
  - Virtual Link – a path between two routers that traverses other routers
- load balancing – supports equal-cost multipath routing (can equally use equal cost routes)
- supports CIDR routing

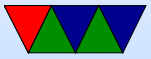


- support available for multicast
- 8-byte password for authentication
- supports hierarchical
- example? complex!



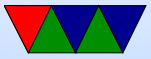
Inter-Domain Routing

- Can be complicated.
- Say company with network, and two connections to outside X,Y. Don't want to send packets out and back even if it looks like lower cost.
- Also don't want to transit packets between X and Y for outsiders. Policy.

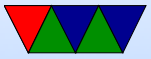


BGP (border gateway protocol)

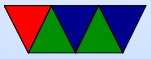
- Intro in 1989 (sketched on two napkins?), started being used 1994
- On fourth version – BGP4 RFC 4271
  support for CIDR and route aggregation
- Uses TCP (reliable) port 179.
- Works for both IPv4 and IPv6 (the latter as an extension)
- Uses path vector rather than distance vector
  - full path, not just next-hop
  - exchanges info with neighbor, but includes complete



path info to avoid looping.
  - Each AS has unique number, so if it sees itself in the path knows there is a loop.
  - Policy routing – can also reject new route based on policy
- Four types of messages – open, update, keepalive, notification
- Whole table not passed around (Due to size), only updates
- Due to size of internet, uses distance vector over link state.

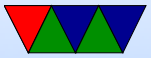


- Keeps track of all feasible paths, but only advertises the "best" one



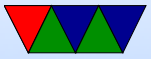
Interior / Exterior BGP

- Interior
  - Interior is a full mesh
  - iBGP makes sure that the setups for multiple gateway routers are kept synchronized
  - Can have a Route Reflector (RR) to avoid the overhead of full mesh
- Exterior
  - eBGP used to talk between other exterior routers at peers.



Routing table Size

- Example. Full BGP of internet backbone router might have more than 300,000 entries (2010) now over 965,000 (2024)
- `http://bgp.potaroo.net/`
- Some routers had limit of 512k so on August 12 2014 part of internet went down when crossed the border.
- Ipv6 currently around 205k (march 2024)



Future

- `https://isbgpsafeyet.com/`
  run by cloudflare, currently “no”
- Swiss SCION BGP replacement?

  `https://www.theregister.com/2026/03/17/switzerland_bgp_alternative/`