

ECE 435 – Network Engineering

Lecture 14

Vince Weaver

`http://web.eece.maine.edu/~vweaver`

`vincent.weaver@maine.edu`

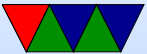
18 March 2021

Announcements

- HW#6 due Friday
- HW#7 will be posted



IPv4 Catastrophe



Out of IPv4 Addresses Problem

- IPv4 address exhaustion
- CIDR not enough
- Addresses managed by IANA globally and five regional registrars (RIR)
- Top level ran out in 2011
- All 5 RIRs finally ran out on Nov 25th, 2019
- Why are we out?
 - Always active connections – unlike dialup, many machine are on all the time



- So many devices – 4G mobile devices all have one
- Inefficiencies originally handing out. Companies like Apple, MIT, DEC, all got 16 million address Class A addresses even if didn't need them
- Despite being out, in 2011 reportedly only 14% of addresses being used
- Why not reclaim unused, such as Class E? The bane of network programmers, the out-of-date router that makes assumptions
- Stanford gave back a class A in 2000



Network Address Translation (NAT)

- Private IP ranges, defined in RFC 1918
 - 1 Class A: 10.0.0.0 - 10.255.255.255 (10.0.0.0/8)
 - 16 Class B: 172.16.0.0 - 172.31.255.255 (172.16.0.0/12)
 - 256 Class C: 192.168.0.0 - 192.168.255.255 (192.168.0.0/16)
- Can use for various reasons, most recently due to network depletion
- NAT: map IP addresses from one group to another. often public to private.
- NAT and NAPT (port translation) RFC 3022



- Basic NAT has one to one mapping of external to internal IPs. Each internal host maps to unique external IP
- Carrier Grade NAT?



Network Address Port Translation (NAPT)

- NAPT: based on port, only one external IP
 - Full cone – most common
 - once an internal address (iaddr/port) has been mapped to an external (eaddr/port) all packets from iaddr/port are sent out and any incoming are passed through with no additional checks
 - Restricted cone – same as above, but only external that have received packets from internal can send through
 - Port restricted cone – same as above, but also checks



port numbers

- Symmetric – best security – outgoing packets mapped to different eaddr/port if the destination or port differs
- When passing through, NAT needs to re-write dest/source/port and recompute header checksum
- Linux: IP-masquerade/iptables
- Many IP people hate NAT:
 - Violates the IP identifies one machine rule
 - Hard to connect two machines if both behind different NATs (NAT transversal)
 - Changes IP to be connection oriented, router has to



remember connections

- Layering violation, looks at TCP/UDP port numbers
- Only works for TCP/UDP
- Some protocols (like FTP) are even more annoying, send address in plain text in data and that has to be adjusted too
- Can only NAT up to 64k machines (why? how many ports are there?)



The Internet Protocol v6

- RFC2460 - RFC466
- Started work in 1991
- Many problems with IPv4. Most notable shortage of addresses.
- IPng. (IPv5 was an experimental stream protocol)
- Migration happening, a large amount of web traffic, especially that from phones, is already switched.
- *not* backwards compatible
- As of July 2016 12.5% of traffic is IPv6



The Internet Protocol v6 Goals

- Support billions of hosts
- Reduce size of routing tables
- Simplify the protocol (so routers can be faster)
- Better security
- Pay more attention to type of service
- Aid multicasting
- Allow roaming w/o changing address
- Co-exist with existing protocols



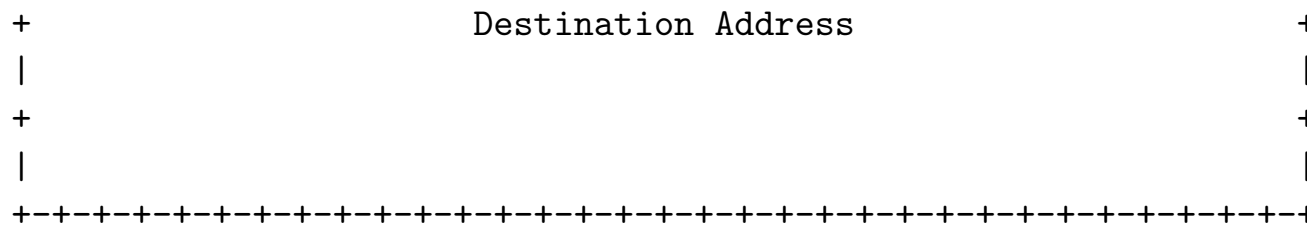
The Internet Protocol v6 features

- Address size 128 bits
 - a lot of addresses. 7×10^{23} for ever square meter
- Simpler fixed length header (speeds up processing)
 - Many fields not really used in IPv4 dropped (or made optional)
- Better support for options
- Better security support
 - IPSEC. Originally mandatory, made optional
 - Can encrypt packets at the network layer



- Quality of service (???)
- Anycast (see end of slides)
- Autoconfiguration (like DHCP)
- Minimum fragment size 1280 (up from 576)
- No checksum – was slow and recalculated often





- **Version Number** (1 nibble [4-bits]) = 6
- **Differential Services/Traffic Class** (8-bits) (QoS/congestion)
- **Flow label** (20-bits) (for streaming?)
- **Payload Length** (16-bits) 64k (header bytes not counted anymore) (if you want longer, extension for Jumbograms)
- **Next header** (8-bits) If nothing special identifies TCP or UDP If special options (fragmentation, security)



indicated

(TCP=0x6, UDP=0x11)

- **Hop Limit** (8-bits) TTL, big debate about whether 8-bits was enough
- **Source Address** (128-bits)
- **Destination Address** (128-bits)
- Why not 64-bit addresses?
- No checksum, link or transport catches issues
What does this mean for UDP?



IPv6 Options

- Happen immediately after the header.
- Should occur in numerical order (though routers might be able to handle if they don't)
- Routers should inspect in order as some later might depend on earlier.
- Plain: IPV6:Next=TCP, TCP, Data
- Example: IPV6:Next=Routing, Routing:Next=TCP, TCP, Data
- Various types



- Hop-by-hop (so far only used for jumbo frames, if that set header length is set to 0)
- Source Routing
- Fragmentation
- Authentication
- Encryption

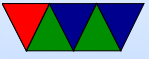


IPv6 fragmentation

- Info is in an extension header
- Routers cannot fragment, only at source
- How can this work when not know MTU?
- MTU is always greater than 1280
- Path MTU discovery protocol to discover MTU along the way (RFC 1981). (IPv4 too, set DNF and get error via ICMP) If too big, sends an error back and source needs to fragment it smaller
- Easier to have source fix things then every router along



the way being able to



IPv6 addresses

- 2^{128} is a lot. 7×10^{23} per m^2 of Earth surface
- Too long for dotted decimal, use colon hexadecimal
- X:X:X:X:X:X:X:X where X is 16 bit chunk
- F000:0123:5678:0000:0000:ABCD:0001:CAFE
- Can drop leading zeros, as well as groups of zeros
F000:123:5678::ABCD:1:CAFE

Note, cannot drop two sets of groups of zeros. Why?
Ambiguous.

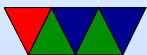
- Do not have classes RFC 4291



- encoding ipv4 addresses in ipv6?
- autoconfig – generate a unique ID, often based on MAC address and send it to router (Router solicitation). Router sends back router advertisement which has subnet prefix and can be used to generate global address Stateless address auto-config (SLAAC). Can also use DHCPv6
- ::1 ip6-localhost, fe00::0 ip6-localnet
- Example

```
inet 130.111.218.23 netmask 255.255.254.0 broadcast 130.111.219.255
inet6 2610:48:100:8da:230:18ff:feab:1c39 prefixlen 64 scopeid 0x0<global>
inet6 fe80::230:18ff:feab:1c39 prefixlen 64 scopeid 0x20<link>
```

inet6, both global and local contain MAC address



IPv6 Privacy Concerns

- Security – have MAC in your IPv6 address?



IPv6/IPv4 compat

- Dual stack – host runs both IPv4 and IPv6 or internal is IPv6 but router converts to IPv4 before passing on
- Tunneling – encapsulate IPv6 inside of IPv4, tunnels across IPv4, split back out to IPv6 on other side of tunnel



IPv6 Routing

- Much like IPv4
- IPv6 network address, “prefix-length” instead of netmask
- Routing table as before



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 Stateless Address AutoConfiguration (SLAAC)
assumes on /64 subnet (so every subnet contains 1000
times the IPv4 space)
- Essentially large enough a system could just pick a
random address and it would work



- 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
- No broadcast, so nodes coming up listen to multicast address
- Neighbour Solicitation (NS) (RFC 4861) use with SLAAC described in RFC 4862 to get MAC address
- One has address, sends out a packet to see if collision,



if not, it's configured

- 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 address
- Needs to figure out DHCP6 or SLAAC

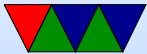


IPv6 setup issues

- It can be hard to subnet. You get 2^{64} addresses but it's hard to split those up into multiple subnets
- Setting up Firewall. Linux has separate ipv4 and ipv6 firewalls



Done with IPv6, On To Routing



Casting

- Unicast – 1:1 – one sender, one destination
- Broadcast – 1:all
- Multicast – 1:many – specify a subset of all
- Anycast – a set of equivalent hosts, which one gets the packet depends on something like closeness / latency
- Geocast – broadcast to limited geographic area



Anycast problems/benefits

- Can spread server load around (DNS servers, web servers, netflix servers)
- Can hijack connection if you can get your fake routing info into a server.



Broadcast Routing



Unicast / Multicast / Broadcast

- Unicast – send from one machine to another
- What if want to send to multiple?
 - Multi-unicast – open direct connection to each destination. Inefficient
 - Broadcast – send to **every** destination? Waste bandwidth, but also need to know all possible destinations
 - Flooding? Also too much bandwidth
 - Multi-destination routing



- Multicast Goals
 - Only send to users who want it
 - Each member only receives one copy
 - No loops
 - Path traveled should be optimal
- Spanning tree – tree with source as root and members as leaves
- Reverse-path forwarding



Why would you multicast?

- Live streams? Backups?
- Why not just multi-unicast?
 - More work on sender, many more packets sent
 - Latency between first and last packet sent



Multicast IP

- For IP, just join a class D network
- To both sender and receiver it's like sending/receiving a unicast packet
- all the hard work done by routers
- How do you join a multicast group?
- Router two tasks: group membership management, packet delivery.



- IGMP (Internet Group Management Protocol)
IGMPv3 RFC 3376
query, report, leave
querier and noquerier
router with lowest IP is querier
no real controls on who can join or send



Group Management

- IGMP (Internet Group Management Protocol)
IGMPv3 RFC 3376
query, report, leave
querier and noquerier
router with lowest IP is querier
no real controls on who can join or send



Multicast Trees

- Steiner tree – NP complete, no one uses
- Heuristics, but none generate entire tree as need centralized and global knowledge
- DVMRP (Distance-Vector Routing Protocol) original protocol, MBONE (tell story)
- Reverse path Forwarding – flood packet out all interfaces except one it came in on. Can have loops; drop dupes.



Then forward on the one that has traveled the shortest path.

Is running the routing table backwards

- Reverse path Broadcast – avoid getting multiple packets
- Protocol Independent Multicast (PIM)
DVRMP not scalable for multicast groups with sparse members
- MOSPF
- CBT



Local Network Broadcasts

- 224.0.0.0/4 was reserved from Class D for multicast
- 224.0.0.0 to 224.0.0.255 for local network broadcasts
- Things like cluster stats (ganglia, can never get to work?)
- Routing info prtocol (RPIv2) OSPF, mDNS, etc.

