ECE 435 – Network Engineering
Lecture 18

Vince Weaver

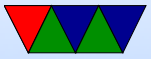
https://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

5 March 2025

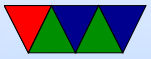
Announcements

- Don't forget HW#6, due Friday
- Midterm on Wednesday March 12th (week away)



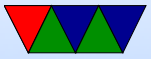
Problems with Fragments

- no way to notify other side of missing fragments
- last fragment is usually short (wasting resources)
- receiver must hold in RAM fragments to be reassembled.
- can DoS by sending lots of fragments but none complete
- fragments have no TCP/UDP header, firewall can't easily filter
- Most modern implementations set DNF on TCP connections and instead rely on path-mtu-discovery
- `https://blog.cloudflare.com/ip-fragmentation-is-broken/`



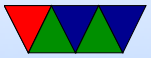
Path MTU Discovery

- Automatically determine the MTU (max transmission unit) between hosts
- Originally for routers, now also for endpoints
- Process
  - Set DNF bit on packets
  - Any router where packet size too big drops packet and sends back error via ICMP
  - Source reduces MTU and tries again until it gets through

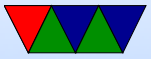


Path MTU Issues

- If MTU gets smaller, will get noticed and can adjust. No way to easily find if MTU gets bigger
- When packets encapsulated/tunneled inside of another protocol an extra header is added, which can kick things above the MTU threshold.
- Complete 3-way handshake can happen (small packets) but then drop all actual traffic. "black hole connection"
- Why would ICMP be blocked?
  - Over-zealous sysadmin

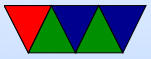


- Traffic load balancers have to keep all TCP packets on same machine, but when ICMP comes in it's not always clear who it belonds to



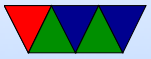
Handling if MTU discovery blocked

- Various workarounds for this. Force MTU to be Ethernet everywhere? Use TCP to probe size, treat packet drops as MTU issue not congestion?
- Interesting article `https://blog.cloudflare.com/path-mtu-discovery-`



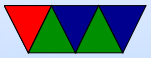
Security Issues with Fragments

- ICMP/UDP larger than MTU, cannot be reassembled
- TCP "Teardrop" attack, send fragments with overlapping offsets, confuse/crash machines
- Fragments can be constructed to obscure malicious text



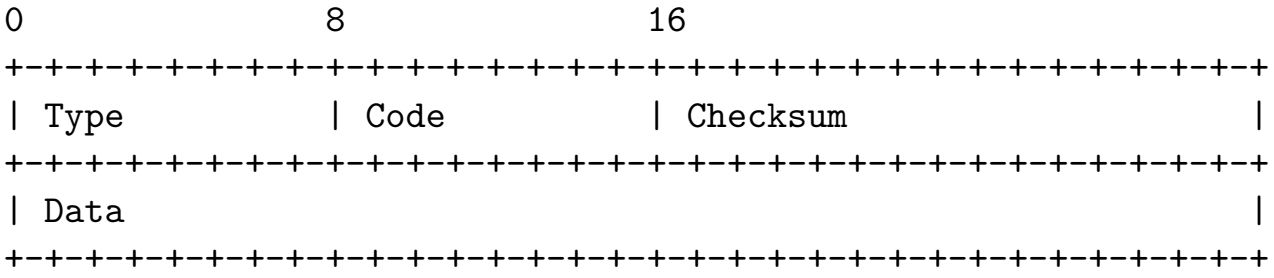
Errors

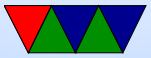
- What happens when something goes wrong with your packet?
- Does a router just drop it?
- Or does it try to let the sender know?



ICMP

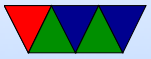
- Internet Control Message Protocol
- Carried as a payload in an IP packet
- IP header type 1
- Some sysadmins block ICMP. Why?





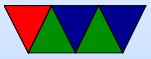
A Selection of ICMP Types/Codes

- `DESTINATION UNREACHABLE`, Also if MTU is too small but do-not-fragment set
- `SOURCE QUENCH` – should slow transmission rate (congestion), This is now usually done in transport layer
- `REDIRECT` – try the other router path
- `TIME EXCEEDED` – exceeded TTL, traceroute uses this
- `PARAMETER PROBLEM` – illegal value in header
- `ECHO`, `ECHO_REPLY` – see if machine is up
- `TIMESTAMP`, `TIMESTAMP_REPLY` – performance debug



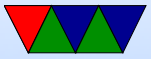
ping

- Mike Muuss in 1983
  `http://ftp.arl.army.mil/~mike/ping.html`
- Like sonar ping (Hunt for Red October), not any of the backronyms you might find.
- Ping the duck
- ICMP ECHO packet, waits for ECHO reply. Prints timing info, etc.
- Used to just say "host is alive". People would make machines called elvis.



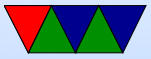
Malicious pings – Ping of Death

- Ping of death – crash any machine on network (late 90s)
  - Technically not a ping bug, but fragmentation
  - Ping typically 56 bytes, but can be 64k
  - Technically not valid, but most will try anyway
  - 64k ping broken into 8 fragments
  - Maximum can specify is 65528, add in 20 for header, 65548
  - This is bigger than 65536, buffer overflow on reassemble



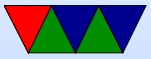
Malicious pings – Other

- Ping flood – could be used as DoS
- Broadcast ping to x.x.x.255 (no longer works)



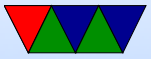
Silly use of pings

- Can you store data by constantly sending it out as pings?
  `https://github.com/yarrick/pingfs`

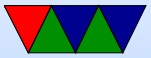


traceroute

- Van Jacobson in 1987 (also wrote tcpdump)
- Uses ICMP
- *not* tracer-t
- Send packet with TTL=1, when sends ICMP error message know where first hop is
- Send packet with TTL=2, find next
- Linux traceroute sends UDP packets as originally ICMP requests weren't supposed to generate ICMP errors
- Sends 3 packets, lists all 3 results

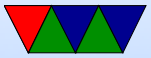


- as an aside, try `traceroute -m 50 bad.horse`



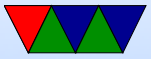
Handing out IP addresses

- If you have a machine on a network, how does it get its IP address?
- Static – given once and never changes
  - IP address
  - Netmask
  - Router / Gateway
  - DNS server
- Dynamic – each boot request IP from server



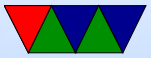
Dynamic Host Configuration Protocol (DHCP)

- RFC2131
- To get on network need IP, subnet mask, default router
- Can we automatically get this?

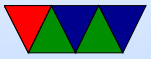


DHCP Protocol

- Device broadcasts, asking for address
- Server can respond with a fixed one (setup in config file) or handle out dynamically from range
- To avoid need for server on each subnet, can pass through
- Details
  - Broadcast DHCPDISCOVER on UDP port 67.
  - All servers send DHCPOFFER on port 68
  - Send DHCPREQUEST, respond with DHCPACK

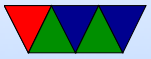


  - Timer, needs to re-request before timer is out or server might give to someone else
  - Get a "lease" from the server. Why short vs long lease?
- Can see this all in action with `dhclient -v`



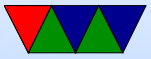
Setting up DHCP server

- Static vs Dynamic (how hand out static addresses?)
- Be careful to not hand out on network you don't own
- Recent Linux systemd DNS debate (whether to fall back to default DNS router if can't get specified one)

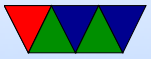


Network Booting

- Can boot computer completely from network
- DHCP server can provide a lot of the info, then server to the OS image
- PXE firmware on ethernet card
- On older machines bootp / tftp instead

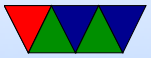


The IPv4 Catastrophe



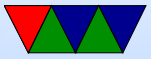
Out of IPv4 Addresses Problem

- 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



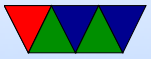
Out of IPv4 Articles

- Finding more available IP addresses proposal:
  `https://www.theregister.com/2022/06/01/ipv4_proposed_changes/`
- To read about using Class E: `https://blog.benjojo.co.uk/post/class-e-addresses-in-the-real-world`
- Interesting Article about IPv4 address Allocation link after the HAM Radio people sold off 1/4 of 44.0.0.0/8 to Amazon:
  `https://blog.daknob.net/mapping-44net/`
- In 2021 Pentagon activated some of its vast IPv4



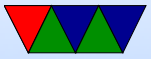
collection turns out had been unused people using them as unroutable numbers, including China military.

https://www.theregister.com/2021/04/26/defense_department_ipv6/

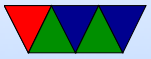


Why are we out?

- 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
  (Stanford gave back a class A in 2000)
- Despite being out, in 2011 reportedly only 14% of addresses being used

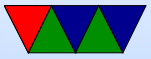


- Why not reclaim unused, such as Class E? The bane of network programmers, the out-of-date router that makes assumptions



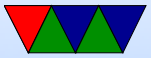
Ways to mitigate lack of addresses

- Add extra bits for addresses in ipv4 in a backward compatible way (this was generally determined to not be practical)
- Replace ipv4 with new protocol
- Have private subnetworks live behind a gateway that only requires one IPv4 address

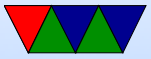


Network Address Translation (NAT)

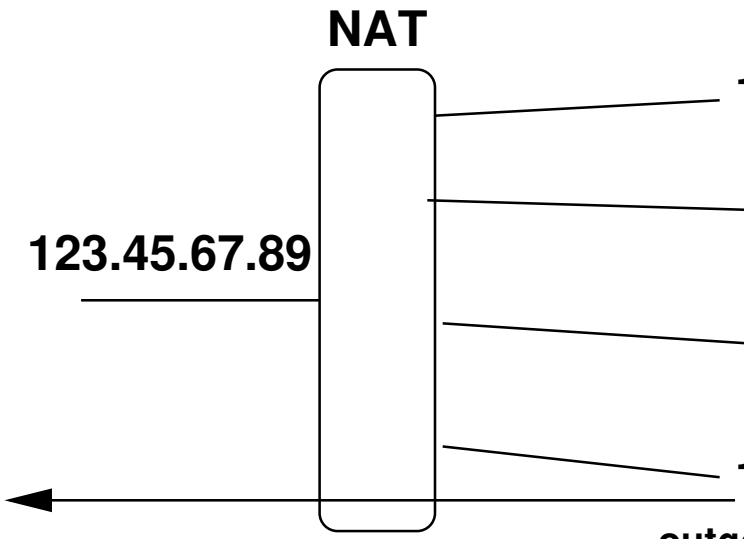
- Private IP ranges, defined in RFC 1918
  - 1 Class A: 10.0.0.0 - 10.255.255.255 (10.0.0.0/8)
  - 16 Class B: 172.16.0.0 - 172.31.255.255 (172.16.0.0/12)
  - 256 Class C: 192.168.0.0 - 192.168.255.255 (192.168.0.0/16)
- Can use for various reasons, most recently due to network depletion
- NAT: map IP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



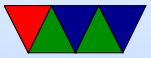
- Basic NAT has one to one mapping of external to internal IPs. Each internal host maps to unique external IP



NAT Example

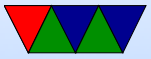


outgoing connection to 8.8.8.8 port 53
source is 192.168.8.5 port 40000
NAT remaps to 123.45.67.89 port 60000
incoming data to 123.45.67.89 60000 remapped back to 192.168.8.5 port 40000

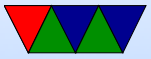


Network Address Port Translation (NAPT)

- 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 to (eaddr/port) are passed back with no additional checks
  - Restricted cone – same as above, but only external addresses that have received packets from internal are allowed through

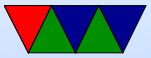


- Port restricted cone – same as above, but only allows packets from the exact address/port from original response
- Symmetric – best security – outgoing packets mapped to different eaddr/port if the destination or port differs



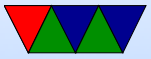
NAT Implementation

- When passing through, NAT needs to re-write dest/source/port and recompute header checksum
- Linux: IP-masquerade/iptables



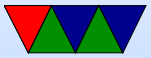
Many IP people hate NAT

- 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



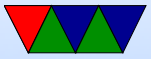
too

- Can only NAT up to 64k machines (why? how many ports are there?)



Carrier Grade NAT (CGN, CGNAT, LSN)

- Internal network uses private IP range
- Public facing server channels these through a set of external IP addresses
- NAT444 – potentially traverse 4 different IP (private in home, private in ISP, external IP)
- RFC6598 – allocate 100.64.0.0/10 for this, to avoid complications where internal/external collisions of the RFC1918 ranges



CGNAT Downsides

- Breaks end-to-end connections
- Stateful
- Doesn't fully solve IPv4 exhaustion problem cases where need a visible IP address (SSL web server?)
- Lots of devices behind a few IPs, what if get banned for spamming/security?
- Breaks port-forwarding for users, as you're in a NAT inside a NAT (port control protocol (PCP) RFC 6887 tries to work around this)