Computer Networks
ICS 651

- MPLS, IS-IS, EIGRP
- Internet Control Message Protocol, ICMP
- IP fragmentation
- Path MTU discovery
- Internet checksum

Other Routing Protocols

- MPLS, Multi-Protocol Label Switching, adds to each packet a small header which includes an integer label.
  - routers in a transit AS can be configured to recognize this label and forward the packet, unmodified, to a specific interface and next hop.
  - this Label Switching can be implemented in hardware.
  - label switching must be set up by software that computes a path through the network for each pair of source (ingress) and destination (egress) routers, assigns a different label to each path, and distributes to each router the labels and the relevant details of each path.
  - MPLS can support any protocol, not just IP
- IS-IS is a link-state protocol, typically working at the layer below IP, and so, like MPLS, more protocol-agnostic than OSPF or RIP
- EIGRP is a Distance-Vector protocol, but sends updates instead of the entire routing table. EIGRP was proprietary to Cisco until 2013, when (most of) the protocol was published as RFC 7868

Software Defined Networking

- routing is an inefficient peer-to-peer system
- can we centralize it?
- challenge: the central controller uses the network to reach the router it wants to configure
- advantages: less overhead if only sending the data that must be sent, faster response to changes, consistent configuration, easier management

Summary: Routing Protocols

- Interior Gateway Protocols, or IGPs, route packets within an autonomous system
- Exterior Gateway Protocols, or EGPs, route packets between autonomous systems
- there is only one common EGP, which is BGP. All the other routing protocols we have discussed are IGPs
- in general, each autonomous system will be running just one IGP, and may have one or more BGP routers that are used as gateways to other autonomous systems

Summary: IP Routing

- summarize, summarize, summarize
- use hierarchy whenever possible:
- route to networks, not hosts
- OSPF: route to areas, not networks
- BGP: route to autonomous systems
- routing protocols can evolve faster than the underlying IP transport
- Interior Gateway Protocols (IGP) are the most automated: RIP for small networks, OSPF and IS-IS for larger ones

Internet Control Message Protocol

- ICMP
- the Internet is complex
- how do we find out what is going wrong?
- send a packet "there and back": ICMP echo, ping
- send an ICMP error packet whenever we drop a (non ICMP error) packet
- ICMP: RFC 792

ICMP Echo

- Echo packet (type 8) or Echo Reply (type 0)
  - types 128 and 129 for ICMPv6
- checksum covers entire packet
- identifier (typically process ID on sender machine)
- sequence number (typically 1, 2, 3...)
- arbitrary data follows (could be large)
- for ping, data typically holds binary date and time (8 bytes or 12 bytes)

Other ICMP Types

[3] Destination Unreachable (network, host, protocol, prohibited...)

[11] Time Exceeded (in transit, during reassembly)

[5] Redirect: use this other router for this destination

[9] Router advertisement

[10] Router solicitation

[4] Source Quench

[12] Parameter Problem

IP fragmentation

- If a packet of size s is to be sent on an interface with MTU m,

  and if s > m

- the packet must either be dropped, or be turned into a collection of smaller packets which carry the same information
  - in IPv6 we drop the packet
  - in IPv4, we drop the packet if the Don’t Fragment (DF) bit is set, and otherwise we fragment it
- each of the smaller packets has its own IP header, which is based on the IP header of the original (too large) IP packet:
- source and destination addresses, hop limit (TTL), and most other fields are the same in the original and all the fragments
- the payload length is adjusted for each fragment
- and some fields are set so that the packet can be reassembled correctly
- each packet ID is the same as in the original datagram

IP fragmentation and reassembly information

- the fragment must contain information about where the payload belongs in the original (unfragmented) datagram
  - this information is the *fragment offset*
    - the fragment offset must be a multiple of 8, so the low-order 3 bits (which are always 0) are not sent – only the top 13 bits are sent
- the fragment should also tell us how big a datagram to expect, so we know when the reassembly is complete
  - when we get the last fragment, its *More Fragments* (*MF*) bit is 0 – every other fragment has MF = 1
- unfragmented packets have fragment offset = 0, MF = 0

IP reassembly

- when receiving a fragment (MF ≠ 0 or fragment offset ≠ 0)
  - check to see if the packet ID matches an existing reassembly context
  - if not, create a new reassembly context
  - the context includes space for the reassembled packet
  - this packet is initially *empty*
- if MF ≠ 0, we now know the final length
- copy the payload into the reassembled packet at the given fragment offset
- keep track of which parts of the packet have been filled
- once the entire packet has been filled, and we know the final length, do IP processing on the entire packet

IP reassembly – summary

- the fragment offset and MF are sufficient for reconstructing the entire packet
- they must be set correctly when fragmenting
- a fragment can be fragmented again if necessary, as long as the fragment offset and MF are set correctly
- the packet ID is an arbitrary number that helps the receiver distinguish packets from the same sender
  - no two packets with the same packet ID should ever be *in flight* at the same time

IP Path MTU Discovery

- send IPv6 packets, or IPv4 packets with DF (Don’t Fragment) set
- cannot send more than local network MTU
- if a router must drop a packet that exceeds the MTU of the outgoing interface, it can send a "destination unreachable/fragmentation needed" ICMP message, or an ICMPv6 “packet too big” message
- this ICMP message carries the MTU
- if there is no ICMP message, sender can do a binary search (on common MTU sizes) to find an MTU that works
- however, the path MTU can change!
- since ICMP message may be dropped, also needs other ways to detect dropped packets
- slow, time-consuming, error-prone...

Internet Checksum

- IP header
- ICMP, TCP, and UDP header, data, and "pseudo-header"
- pseudo-header are the IP level fields which, if corrupted, cause mis-delivery: source and destination IP addresses, protocol number, packet length
- if all bytes in packet add to n (without checksum), put -n in checksum field, so all received bytes added together give 0
- 16-bit, one's complement arithmetic checksum

16-bit 1's complement arithmetic

- add unsigned 16-bit quantities as always
- "carry-out" from 16-bit addition added back in to LSB
- "carry-out" can be accumulated in high-order part of 32-bit word, and added at end
- negation is complement, zero is 0xffff or 0x0
  - in numbers obtained by addition, zero is always 0xffff
  - if you add 1 to 0xffff, you get 0 plus a carry out bit, which when added to 0, gives 1 – this is the desired result
- example: 9ABC + 8888 = 2345