Network Layer Forwarding

CS640, 2015-02-19

Overview

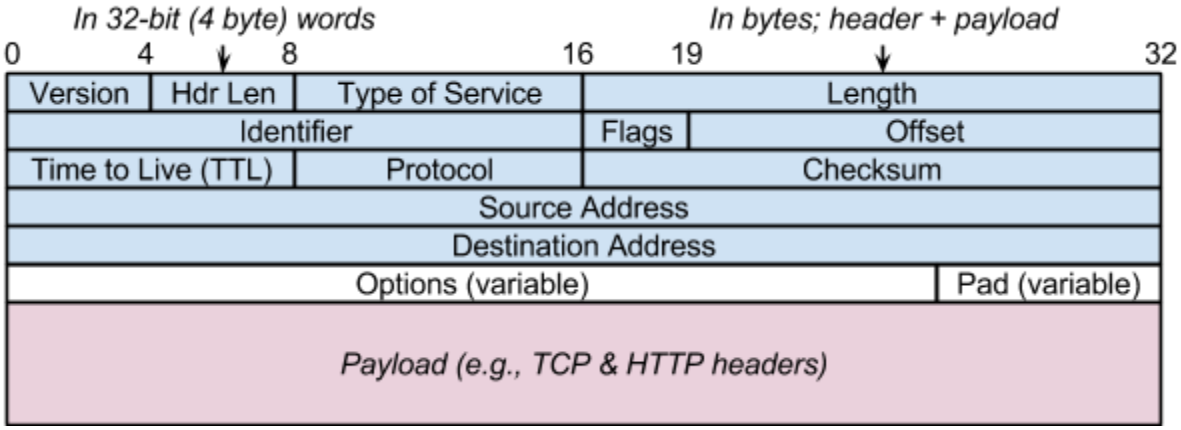
- IP Header
- Network layer forwarding
- Address resolution
- Address assignment
- Error reporting

Recap: Network Addresses

- Convert netmask in slash notation to dotted decimal
  - 255 byte for each multiple of 8
  - Next byte: 256 - $2^{(8\text{-remain})}$
  - Pad with 0 bytes
- Convert netmask in dotted decimal to slash notation
  - Add 8 for each 255 byte
  - Add 8 - $\log_2(256\text{-byte})$ for 0 < byte < 255
- Determine number of hosts given netmask in slash notation
  - $2^{(32\text{-slash})}$ - 2
- Determine address range given network address and netmask in dotted decimal
  - Minimum address: add 1 to last byte of network address
  - Maximum address:
    - For first non-zero byte in network address, round up to nearest multiple of 256 - mask byte and subtract 1
    - If first non-zero byte was not last byte, set last byte to 1
- Determine network address given host address and netmask in dotted decimal
  - If mask byte = 255, network byte = same as host byte
  - If mask byte = 0, network byte = 0
  - Network byte = host byte rounded down to nearest multiple of 256 - mask byte

Internet Protocol (IP) Header

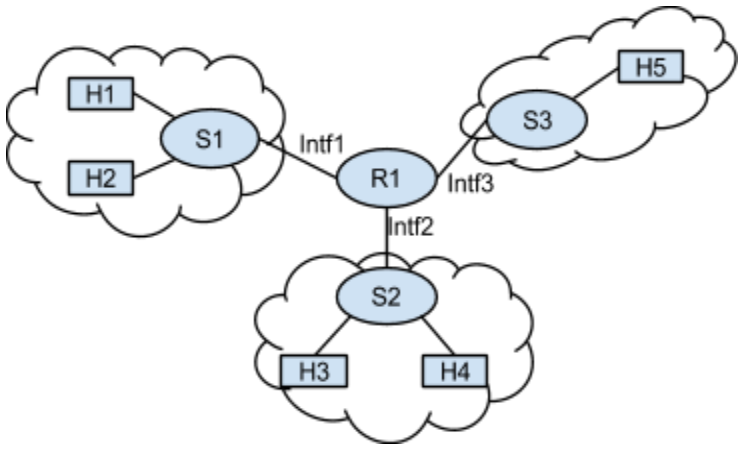
- Packet format



- Version: v4 or v6
- Header length: in 32-bit words; range is 5-15 (depending on if there are options)
- Type of service: used to define priority of traffic
- Length: in bytes; includes IP header length plus payload length
- Identifier, flags, offset: used for fragmentation
- Time to live: maximum number of hops (i.e., routers) a packet should traverse
  - Prevents packets from getting forever stuck in a routing loop
- Protocol: specifies what header comes next (TCP, UDP, ICMP, etc.)
- Checksum: computed over header and payload
- Source address and destination address
- Options: sometimes included; length varies, but always padded to 32-bit word

Network Layer Forwarding

- Each packet contains destination IP
- Router constructs a table of route entries
  - Subnet, gateway, mask, interface
  - For now, assume route table is statically specified
- For each entry in forwarding table
  - Bitwise AND pkt's dst IP and entry mask
  - If result matches entry subnet
    - If gateway is empty
      - Destination is reachable via link layer forwarding
      - Deliver datagram directly to destination (via link layer forwarding)
    - Else
      - Destination is in a different (sub)network
      - Deliver datagram to gateway (via link layer forwarding)
- Example
  - Topology



- R1's routing table

Subnet	Gateway	Mask	Interface
10.0.0.0		255.255.255.0	Intf1
10.0.2.0		255.255.254.0	Intf2
10.0.4.0		255.255.252.0	Intf3
0.0.0.0	10.0.4.1	0.0.0.0	Intf3

- ****Out which interface will each packet be sent?***
  - Destination is 10.0.2.10 => send out Intf2 to 10.0.2.10
  - Destination is 10.0.6.10 => send out Intf3 to 10.0.6.10
  - Destination is 10.0.10.10 => send out Intf3 to 10.0.4.1

- Example
  - R1's routing table

Subnet	Gateway	Mask	Slash	Interface
10.0.2.0		255.255.254.0	23	Intf2
10.0.2.128		255.255.255.128	25	Intf1
10.0.3.0		255.255.255.0	24	Intf3
0.0.0.0	10.0.3.1	0.0.0.0		Intf3

  - ****Out which interface will each packet be sent?***
    - Destination is 10.0.2.10 => send out Intf2 to 10.0.2.10
    - Destination is 10.0.2.130 => send out Intf1 to 10.0.2.130
    - Destination is 10.0.3.10 => send out Intf3 to 10.0.3.10
    - Destination is 10.0.4.10 => send out Intf3 to 10.0.3.1
- Longest prefix match
  - Routing entries could overlap
  - Use "most specific" route possible
  - "Most specific" is determined by the length of the prefix
    - i.e., the # of 1's bits in the mask
    - i.e., the largest number if mask is written using slash notation

Address Resolution

- How do we get packet to specific host or gateway?
  - Use link layer forwarding
  - We need to know host or gateway's link layer address (i.e., MAC address)
- Address Resolution Protocol (ARP)
  - Used to translate IP addresses to MAC addresses
  - ARP table contains <IP address, MAC address> tuples
  - ****How do we populate the ARP table?***
    - Statically
      - Lots of work; prone to errors when hosts leave the network, join the network, or move within the network
    - Exchange request/reply messages between hosts

    - Link layer broadcast of request sent on interface where original packet should be sent
        - Broadcast MAC address of all 1s (i.e., FF:FF:FF:FF:FF:FF)
        - Request contains IP address whose MAC address we want to know
    - Host/gateway that has the IP constructs an ARP reply with its MAC address
        - Use link layer forwarding to send reply back to requestor -- we should be able to know the reverse path if we used learning switches
    - Entries in ARP table timeout after some period of time

Address Assignment

- Dynamic Host Configuration Protocol (DHCP)
    - Assigns IP addresses to hosts
    - Host (DHCP client) broadcasts a request for an IP address
    - DHCP server receives the request, picks an address, and sends reply with assigned IP
        - Configure DHCP server with range of IPs to assign

Error Reporting

- Error situations
    - Router does not have any routing entry that matches packet
    - Router does not have entry in ARP table and ARP resolution fails
    - TTL of packet is zero
- Internet Control Message Protocol (ICMP)
    - Used to send error message back to sender of IP packet
    - Also used for ping: echo request and echo reply