

# Computer Networks

## ICS 651

- Internet Protocol
- packet forwarding
- IP routing tables
- IP addressing
- local configuration

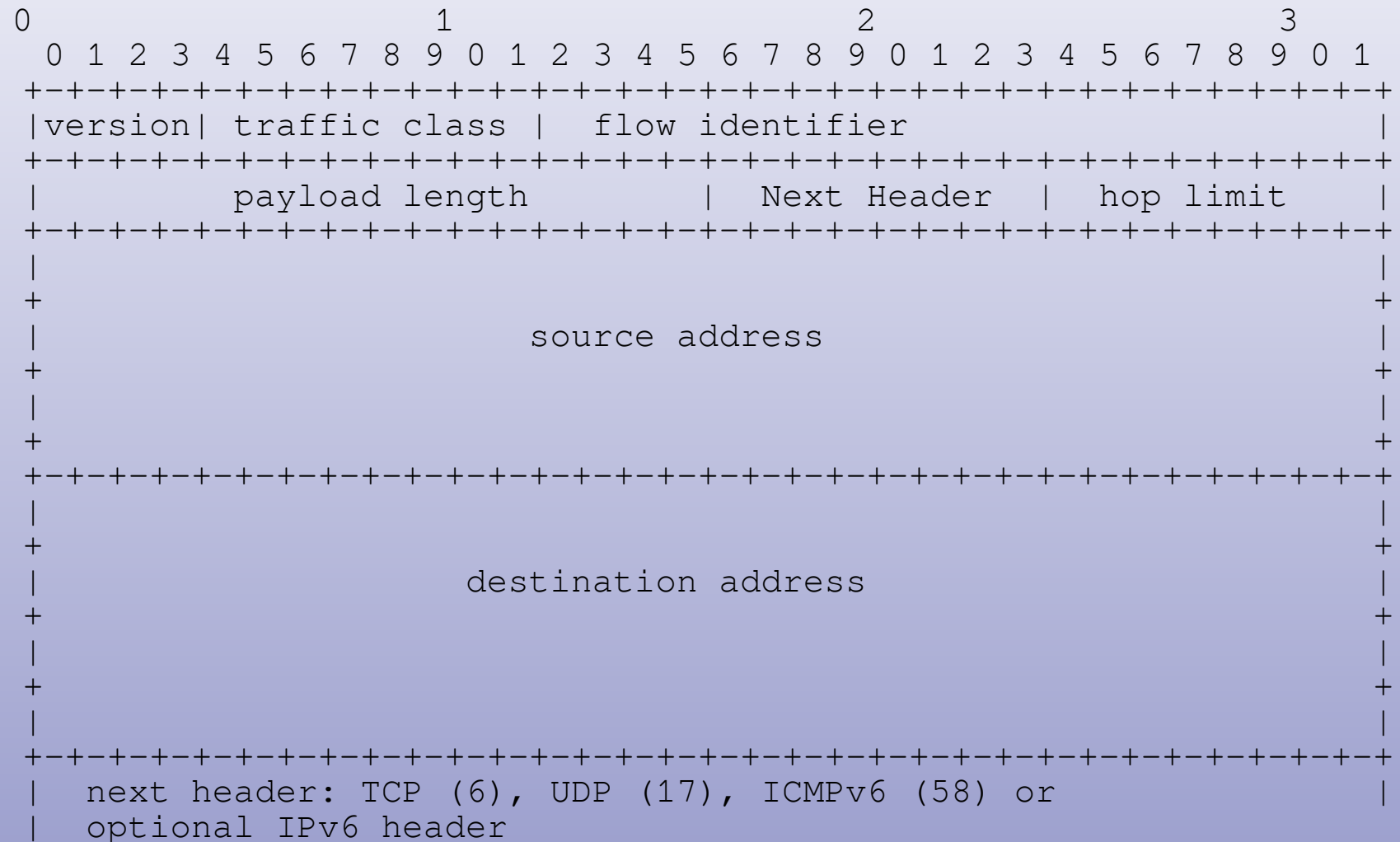
# What does DNS need from its lower layers?

- a network with multiple hosts
- any-to-any communication of packets
- reliability is not required: DNS retransmits the query if it does not get a response (since the database is read-only, queries are idempotent)
- routing based only on IP addresses
- initial configuration:
  - a machine needs to be configured with the address of a DNS server
  - an authoritative DNS server needs the IP addresses of DNS servers of neighboring zones

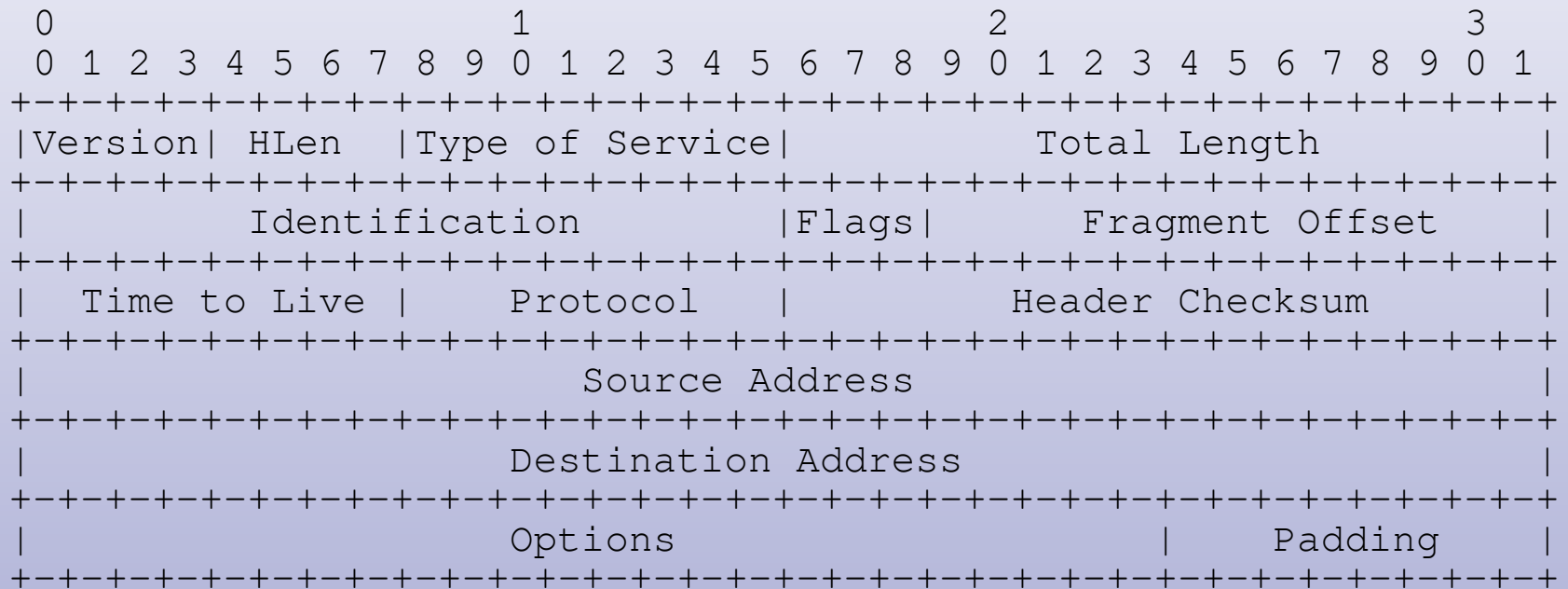
# Internet Protocol

- IP is responsible for delivering packets end-to-end in the Internet
  - reliability, correctness, and in-order delivery are not required
- IP does this by adding a header to each packet, which contains
  - source and destination address
  - protocol number
  - hop limit
  - many other fields
- every IP host that forwards a packet must send it closer to its destination
  - a *routing table* should list routes to all possible destinations

# IPv6 header



# IPv4 header



# IP header fields

- all fields are big-endian
- version number must be first field (so IP can evolve)
- Source Address is useful for responding
- Type of Service is defined but not widely used
- Total Length is needed for protocols that may pad the packet (e.g. ethernet) and to check for errors
- header length used for IP options, e.g. timestamp
- TTL/Time To Live/Hop Limit (really a "maximum hop count") kills packet in case of routing errors
- protocols: 1 (ICMP), 6 (TCP), 17 (UDP)

# more IPv4 header fields

- header checksum (RFC 1071, <http://www.ietf.org/rfc/rfc1071.txt>) only protects header
- packet ID and fragmentation fields (discussed next lecture) allow us to send packets larger than the underlying network's MTU (Maximum Transmission Unit)
- minimum MTU for carrying IPv4 is 576 bytes, for IPv6 is 1280 bytes
- IP header options (IPv6 extension headers) allow additional, optional functionality
  - e.g. source routing

# IPv6 compared to IPv4

- huge addresses
- hop limit instead of TTL
- no fragmentation in the basic header -- fragmentation is in an optional header, and can only be done by the sender
- no header checksum
- fixed-size header with optional extension headers

# Some Properties of IP

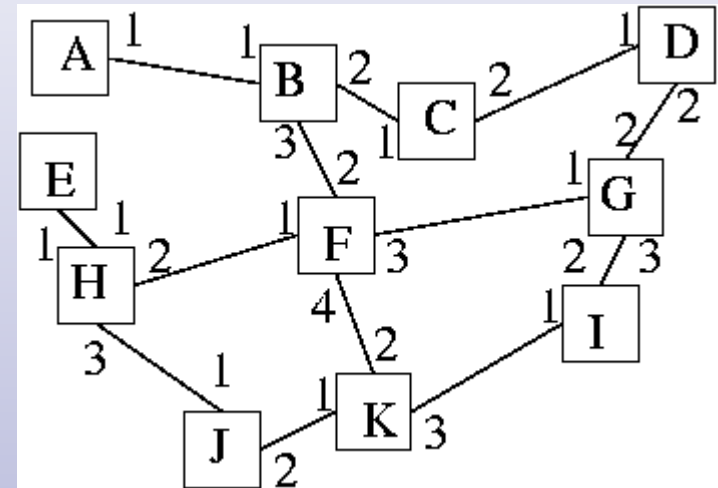
- connectionless: routing is based only on the destination address
- TTL field keeps small routing mistakes from bringing down the network
- unreliable: no packet sequence numbers, acknowledgements, nor error checking on the data
- source address is never used in "normal" IP processing
  - may be used for egress filtering
  - is used for responding to a message

# What IP adds to the simple network

- end-to-end delivery
  - using the hop-by-hop delivery to reach the next hop
- addresses
- routing
  
- IP does NOT add:
  - port numbers
  - reliable transmission
  - congestion control

# IP forwarding

- In this figure, if A sends to E, A puts E in the destination address field, and sends to B
- B chooses interface 3 and sends to F
- F chooses interface 1 and sends to H
- H chooses interface 1 and sends to A
- if H chooses the wrong interface, the packet will never be delivered



# Definitions

- a host with multiple interfaces is **multihomed**
- a host with multiple interfaces that agrees to forward packets is a **router**
  - also known as a **gateway**

# IP Forwarding Algorithm

1. build and maintain a table of pairs: (destination address, interface)
2. when a packet is received with destination A, check the checksum. If the checksum is incorrect, discard the packet
3. if A is the address of one of my interfaces, process the packet
4. otherwise, decrement the TTL (discard if new TTL=0)
5. search for A in my routing table
6. if found (A, I), forward the packet over interface I, otherwise, discard the packet

# More Realistic IP Forwarding Algorithm

1. build and maintain a table of pairs: (destination address, interface, *IP of next hop*)
2. when a packet is received with destination A, check the checksum. If the checksum is incorrect, discard the packet
3. if A is the address of one of my interfaces, process the packet
4. otherwise, decrement the TTL (discard if new TTL=0)
5. search for A in my routing table
6. if found (A, I, N), forward the packet *to next hop N* over interface I, otherwise, discard the packet

This is needed because some networks, e.g. Ethernet and WiFi, normally need the IP address of the next hop for delivery

# Address Groupings

- If the routing table needed an entry for every host, it might be very large (up to  $2^{32}$  entries for IPv4,  $2^{128}$  entries for IPv6)
- so we group entries by the initial (high-order) bits, representing the network number
- only one routing table entry is needed per network
- the remaining bits (the host number) are used, in a network-dependent way, only within a network, i.e. are of no concern in IP routing
- this summarization is most effective when the network structure is hierarchical

# Sample Routing Table: IPv4

```
% route -n
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
0.0.0.0          128.171.24.193  0.0.0.0          UG      0      0      0 eth1
10.0.0.0         172.17.70.33    255.0.0.0        UG      0      0      0 eth0
128.171.24.192   0.0.0.0         255.255.255.192  U        0      0      0 eth1
169.254.0.0      0.0.0.0         255.255.0.0      U        1002   0      0 eth0
169.254.0.0      0.0.0.0         255.255.0.0      U        1003   0      0 eth1
172.16.0.0       172.17.70.33    255.240.0.0      UG      0      0      0 eth0
172.17.70.32     0.0.0.0         255.255.255.224  U        0      0      0 eth0
192.168.0.0      172.17.70.33    255.255.0.0      UG      0      0      0 eth0
```

- “Gateway” is the IP address of the next hop router
- if there is no gateway (no G flag), the destination address should be on the directly connected network
- the network mask is a dotted-decimal representation of the number of bits in the network part of the address
  - e.g. 255.0.0.0 is an 8-bit network number
  - 255.255.255.192 is a 26-bit network number
  - how many bits does 255.240.0.0 represent?

# IPv4 Addresses

two ways of saying which part is the network and which part the host number:

1. class-based: the first few bits tell us how many bits are in the network part (class A: 8 bits, class B: 16 bits, class C: 24 bits). This is the older way of doing this (but is the standard way in IPv6).

2. class-less (newer): each routing table entry also has a **mask**, a 32-bit number of the form `111...1100...00` that has:

- a 1 bit for every bit of the address that is part of the network number, and
- a 0 bit for every bit of the address that is part of the host number
- sometimes we use a number (0..30) instead of a 32-bit mask, e.g. `128.171.10.1/255.255.255.0` can be written as `128.171.10.1/24`

this is CIDR, Classless Inter-Domain Routing

# Classless Interdomain Routing CIDR

- CIDR is a more efficient way of using IP addresses, because you can:
  - have network sizes other than  $2^8$ ,  $2^{16}$ , and  $2^{24}$  addresses
  - do multiple hierarchical subdivisions, e.g. 128.171.0.0/16 for routing to UH, and 128.171.10.0/24 for routing within UH
- CIDR was adopted around 1994, due to impending exhaustion of class B addresses
- destination 0.0.0.0 with netmask 0.0.0.0 identifies the default route – every possible address matches this route

# Sample Routing Table: IPv6

```
% route -6n
Kernel IPv6 routing table
Destination                                Next Hop                                Flag Met Ref Use If
::/96                                       ::                                       !n   1024 0      0 lo
0.0.0.0/96                                 ::                                       !n   1024 0      0 lo
2002:a00::/24                              ::                                       !n   1024 0      0 lo
2002:7f00::/24                             ::                                       !n   1024 0      0 lo
2002:a9fe::/32                             ::                                       !n   1024 0      0 lo
2002:ac10::/28                             ::                                       !n   1024 0      0 lo
2002:c0a8::/32                             ::                                       !n   1024 0      0 lo
2002:e000::/19                             ::                                       !n   1024 0      0 lo
3ffe:ffff::/32                             ::                                       !n   1024 0      0 lo
fe80::/64                                  ::                                       U    256 0      0 eth0
fe80::/64                                  ::                                       U    256 0      0 eth1
::/0                                        ::                                       !n   -1   1 45053 lo
::1/128                                    ::                                       Un    0   3 3794 lo
fe80::250:56ff:feb0:63e/128                ::                                       Un    0   1      0 lo
fe80::250:56ff:feb0:173a/128               ::                                       Un    0   1      0 lo
ff00::/8                                   ::                                       U    256 0      0 eth0
ff00::/8                                   ::                                       U    256 0      0 eth1
::/0                                        ::                                       !n   -1   1 45053 lo
```

- each address in this table shows the number of bits in the network part of the address:
  - /24 means 24 bits are the network prefix, and  $128-24 = 104$  bits are the host part of the address

# IPv6 Addresses

- RFC 4291, IP Version 6 Addressing Architecture
- for many addresses, 64-bit network prefix and 64-bit interface identifier
- network prefix includes a routing prefix and a subnet ID that add up to 64 bits
- the number of bits in the routing prefix is distributed as part of the routing protocol, as in CIDR

# Writing IPv6 Addresses

- 8 groups of 16 bits, each group written as 4 hexadecimal digits
- groups are separated by colons: :
- only the significant digits need to be written, e.g.  
1 : 2 : 3 : 4 : 5 : 6 : 7 : 8 is a valid IPv6 address
- One sequence of 0 groups can be written as ::  
::1 is the loopback address  
fe80 :: 250 : 56ff : feb0 : 173a is a valid address
- :: / 0 is the network number of the default route

# IP Routing, details

- Frequently, more than one route in the routing table will match a given destination address
  - e.g. the default route matches every address
- if so, the route with the longest network mask is used
  - this route is called the **longest match** [sic]
- if there are multiple longest matches, the one with the lowest metric is used
- all this applies to both IPv4 and IPv6

# Routing Errors

- Routing table has more than one entry for a single destination (this is generally OK)
  - A destination might be connected, but not be in the table -- no communication is possible
  - A packet is routed in the wrong direction, but eventually gets there (not uncommon, OK)
  - A packet is routed in the wrong direction, and either starts to loop or ends up at the wrong place, so the packet is lost -- no communication, packet is discarded when TTL reaches zero
- all errors (except physical disconnection) are in the routing table

# Local Configuration

- each interface must be given its IP address
- host/router must place all the local routes, next hops, and network masks into the routing table
- host/router must know the address of at least the "default" router
- there may be further configuration for DNS, particularly the IP number(s) of DNS servers

```
% ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 172.17.70.36  netmask 255.255.255.224  broadcast 172.17.70.63
    inet6 fe80::250:56ff:feb0:173a  prefixlen 64  scopeid 0x20<link>
    ether 00:50:56:b0:17:3a  txqueuelen 1000  (Ethernet)
    RX packets 6404101  bytes 1633307132 (1.5 GiB)
    RX errors 0  dropped 331  overruns 0  frame 0
    TX packets 4336508  bytes 25639721821 (23.8 GiB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
% cat /etc/resolv.conf
# Generated by NetworkManager
nameserver 192.168.10.115
```

# Summary

- the Internet Protocol is designed to take data end-to-end under a "best effort" model
- IP does not provide:
  - reliability
  - in-order delivery
  - error-free delivery
- the major difference between IPv4 and IPv6 is in the addresses
- routing is easy once the tables are built
- summarizing (routing to networks instead of hosts) helps reduce the size of the tables