CS 372 Lecture #30

IP Datagram Fragmentation

- Maximum Transmission Unit (MTU)
- Datagram re-assembly

Note: Many of the lecture slides are based on presentations that accompany *Computer Networking: A Top Down Approach,* 6th edition, by Jim Kurose & Keith Ross, Addison-Wesley, 2013.

Encapsulation across multiple hops

- Whenever a datagram transits a physical network, it is encapsulated in a frame appropriate to that network.
  - Hardware frame encapsulation/un-encapsulation at the link layer
  - Hardware frame formats may differ

- Each router in the path from the source to the destination:
  - *Un-encapsulates* incoming datagram from frame (link layer)
  - *Processes* datagram - determines next hop (network layer)
  - *Re-encapsulates* datagram in outgoing frame (link layer)

- Datagram itself is (almost!) unchanged
  - may be fragmented

MTU

- Every hardware technology specification includes the definition of the maximum size of the frame data area
  - *maximum transmission unit* (MTU)
- Any datagram encapsulated in a hardware frame must be smaller than the MTU for that hardware
- Internet has variety of network technologies with different MTUs
  - Source doesn't know *path MTU*
    - smallest MTU for complete route to destination
  - Easy for source to limit IP datagram size to *local MTU*
    - link layer must pass local MTU up to transport layer for TCP segments
  - A downstream network might have smaller MTU than local network

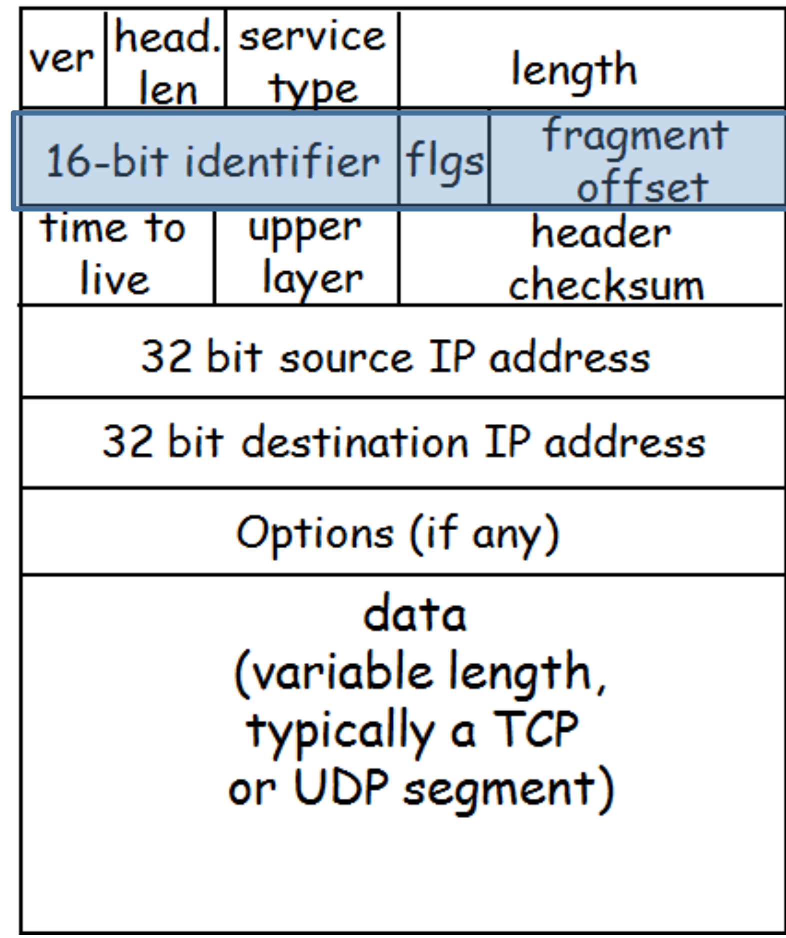
Choosing datagram size

- IP datagrams can be much larger than hardware MTUs
  - IP TOTAL LENGTH: maximum is 65,535 bytes (including header)
- Typical hardware frames:
  - Ethernet: 1500 bytes
  - Token ring: 2048 or 4096 bytes
- One technique - limit datagram size to smallest MTU of *any* known network
  - Problem: impedes use of newer technologies
- IPv4 uses *fragmentation*

Fragmentation

- Router detects <u>inbound</u> datagram that is larger than the <u>outbound</u> *network MTU*
  - Splits datagram into fragments
  - Each fragment is smaller than the MTU of the outbound network
- Each fragment is an <u>independent datagram</u>
  - Includes all header fields
  - All fragments have the IDENTIFIER of the original datagram
  - 3-bit FLAGS field indicates if datagram is a fragment
  - 13-bit FRAGMENT OFFSET gives order of fragment in original datagram

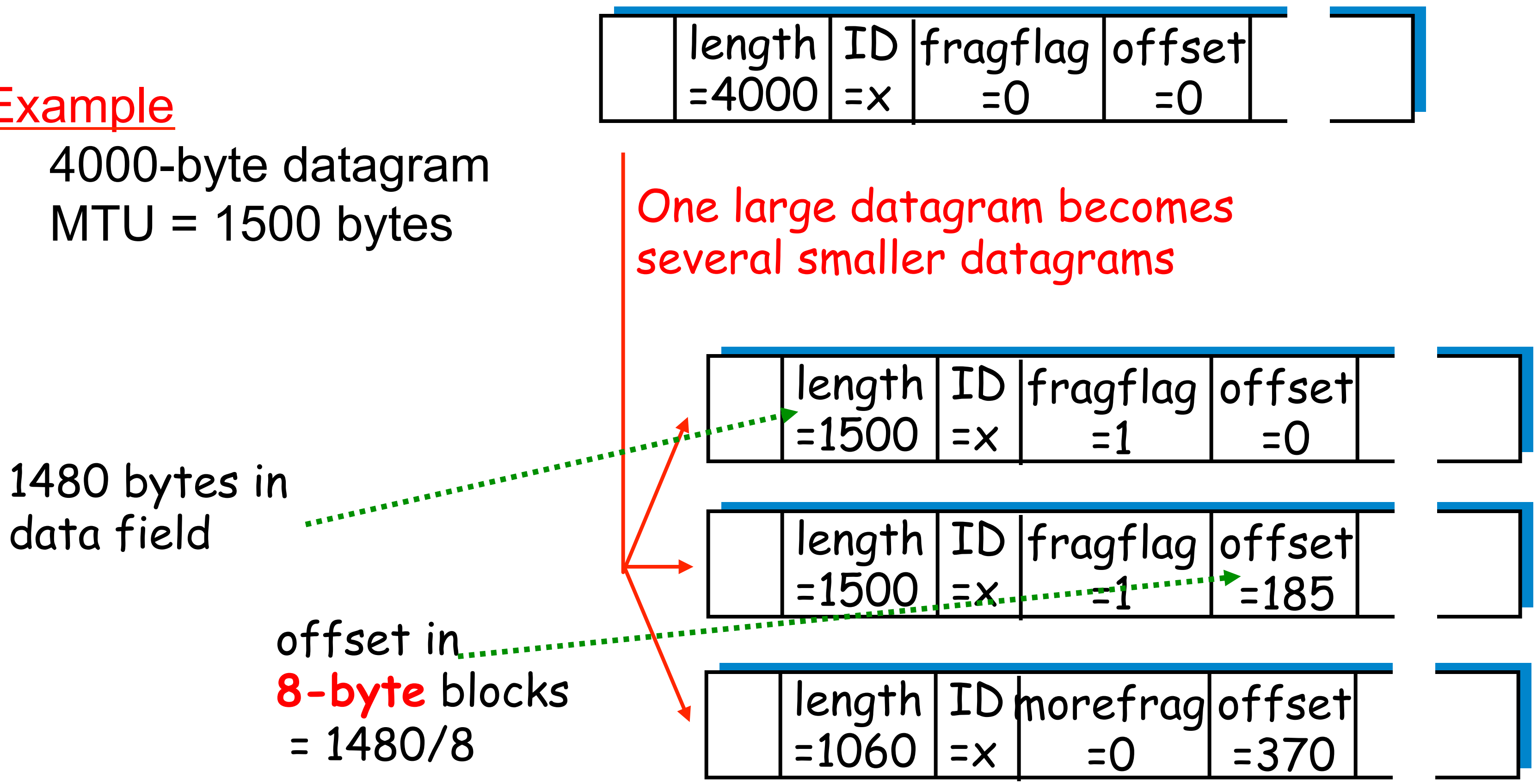
IP datagram format



IP Fragmentation and Reassembly

Example

- 4000-byte datagram
- MTU = 1500 bytes



fragflag: 0 = not a fragment, 1 = is a fragment
dontfrag: 0 = OK to fragment, 1 = do not fragment
morefrag: 0 = last fragment, 1 = more fragments

Re-assembly

- Reconstruction of original datagram is called *re-assembly*
  - performed by protocol stack at final destination
- IDENTIFICATION field in each fragment matches original datagram
  - fragments from different datagrams can arrive out of order and still be sorted out
- FLAGS field bit identifies fragment containing end of data from original datagram
- FRAGMENT OFFSET specifies re-assembly order
  - counted in 8-byte blocks

Re-assembly

- Once a datagram is fragmented, it stays fragmented until it reaches destination.
- The final destination performs the reassembly.
  - Reassembly is not done along the way, even if an outbound network has a larger MTU.
- Two advantages to having the final destination do the reassembly:
  - It reduces the amount of information needed by routers.
  - It allows routers to change dynamically.
  - It avoids re-fragmenting

Fragment loss

- IP does not guarantee datagram delivery, so fragments may be lost.
- Destination
  - Sets timer with first fragment
  - If timer expires before all fragments arrive
    - one or more fragments assumed lost
    - Destination drops *entire* original datagram
    - Source is assumed to retransmit entire original datagram
- Reasons for the all-or-nothing behavior:
  - There's no mechanism for a receiver to tell a sender which fragments arrived.
  - The sender does not know about any fragmentation that happened along the way.
  - Even if the sender could retransmit a single fragment, it might take a different route and be fragmented differently.

Fragmenting a fragment

- Fragment may encounter subsequent network with even smaller MTU
- Router fragments the fragment to fit local MTU
- Resulting (sub)fragments look just like original datagrams (except for size and offsets)
- No need to reassemble hierarchically
- IP does not distinguish among "original" fragmentation and subsequent fragmentations
- Re-assembly is entirely the responsibility of the protocol on the destination host

Summary Lecture #30

- Definitions:
  - MTU
    - local MTU, network MTU, path MTU
- fragmentation, fragments, sub-fragments
- re-assembly