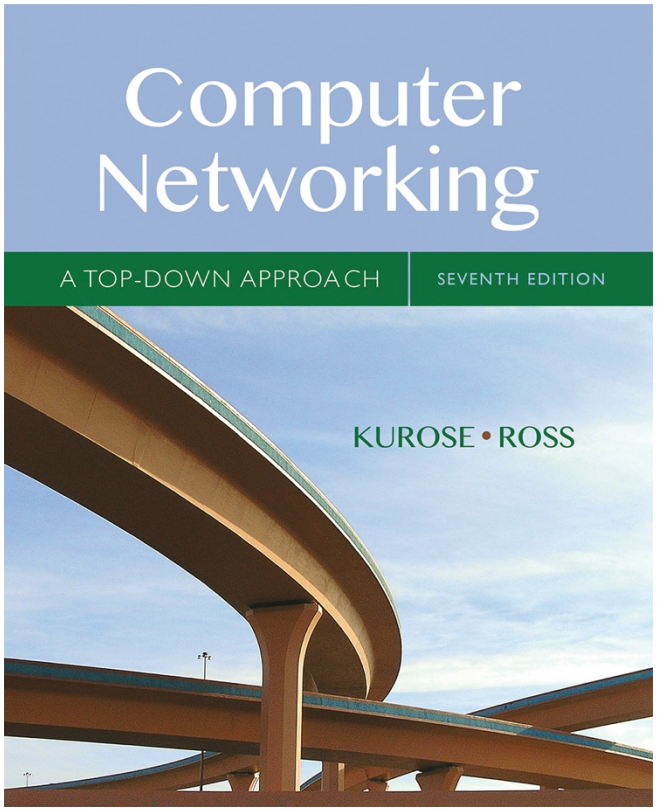
Chapter 4
Network Layer: The Data Plane



A note on the use of these Powerpoint slides:

We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you see the animations; and can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a *lot* of work on our part. In return for use, we only ask the following:

- If you use these slides (e.g., in a class) that you mention their source (after all, we'd like people to use our book!)
- If you post any slides on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

Thanks and enjoy! JFK/KWR

Computer Networking: A Top Down Approach

7th edition
Jim Kurose, Keith Ross
Pearson/Addison Wesley
April 2016

Minor modifications made to original slides by Nathan Bowman

Chapter 4: outline

4.1 Overview of Network layer
- data plane
- control plane

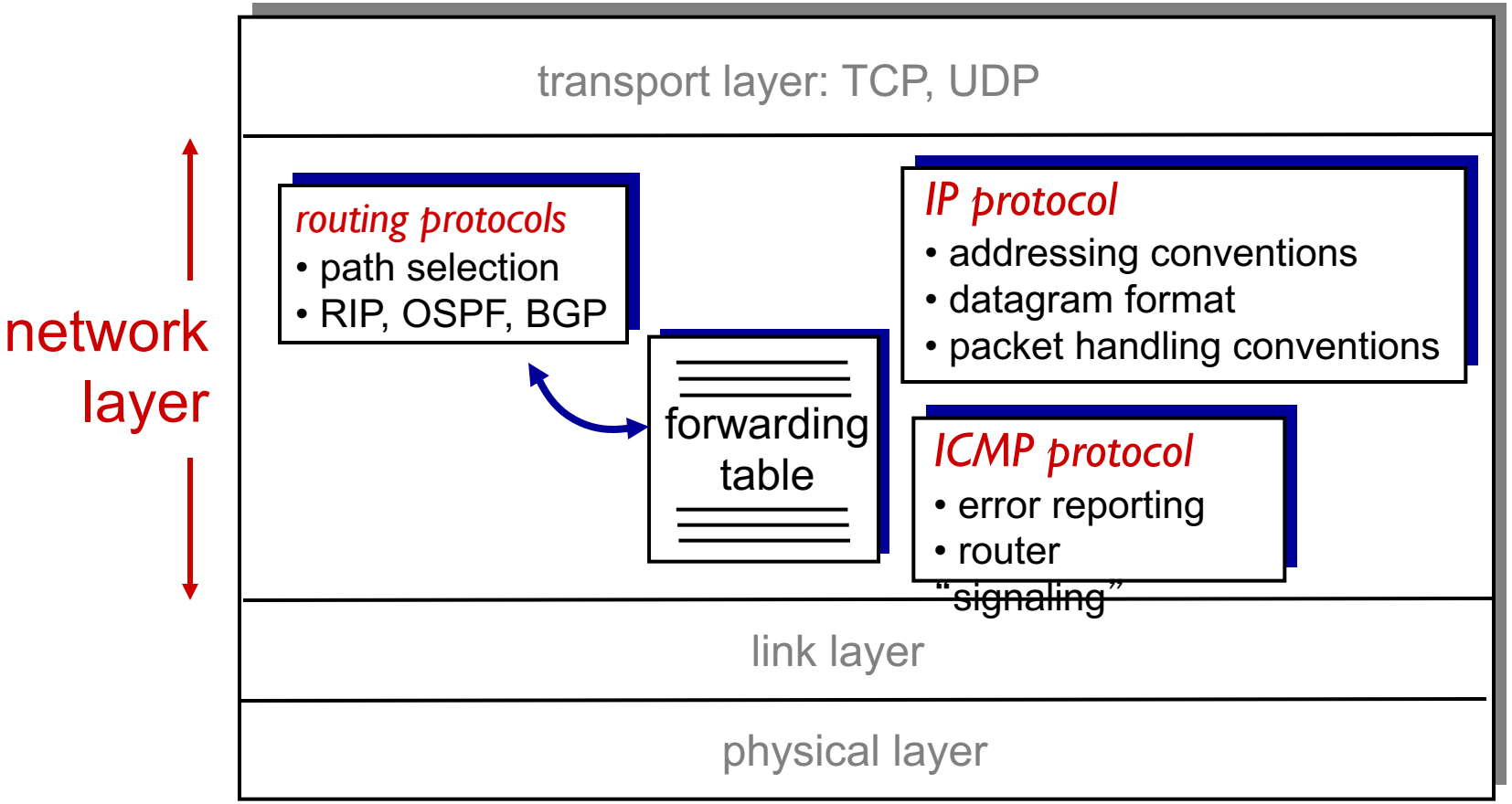
4.2 What's inside a router

4.3 IP: Internet Protocol
- datagram format
- fragmentation
- IPv4 addressing
- network address translation
- IPv6

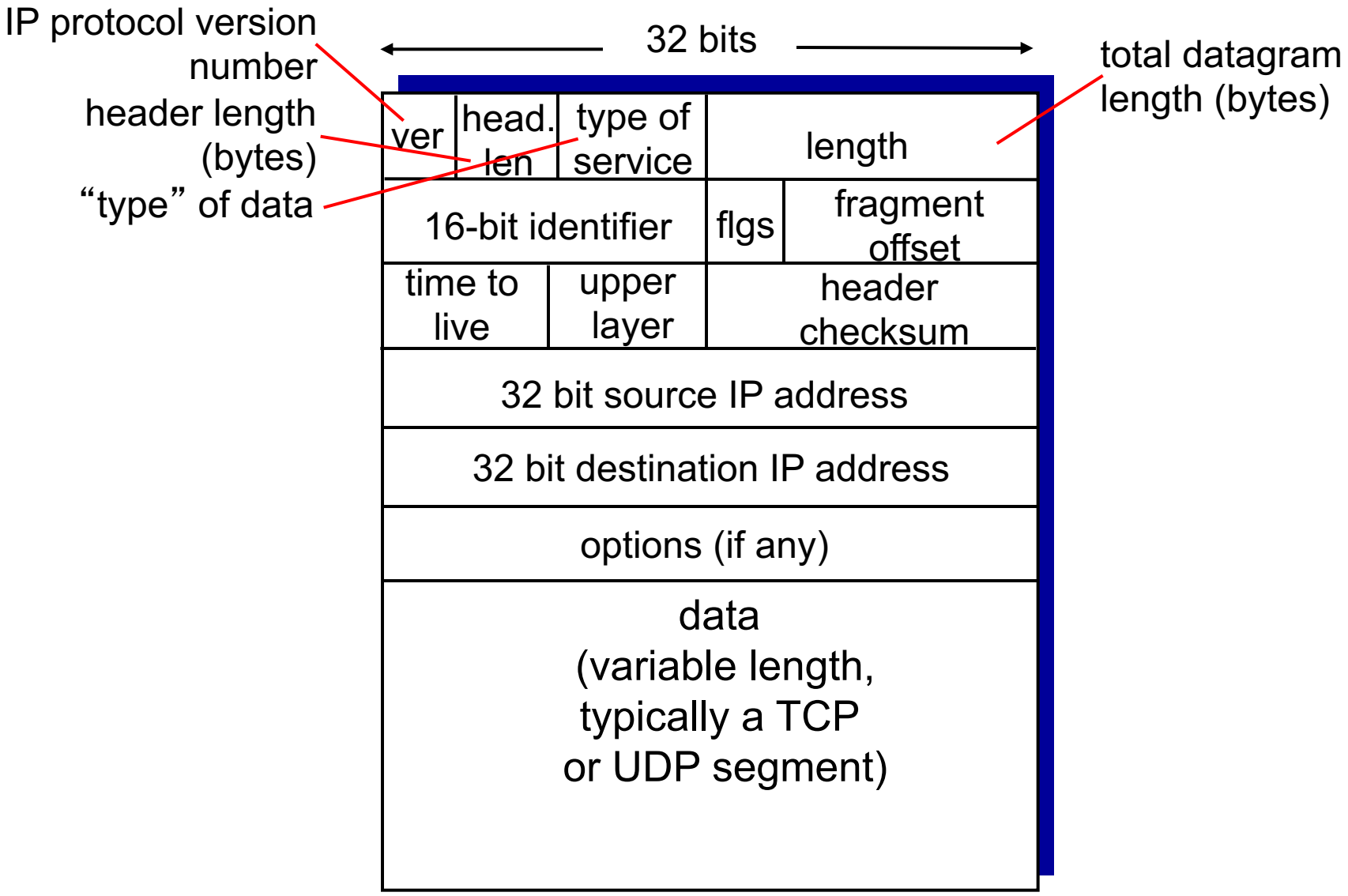
4.4 Generalized Forward and SDN
- match
- action
- OpenFlow examples of match-plus-action in action

The Internet network layer

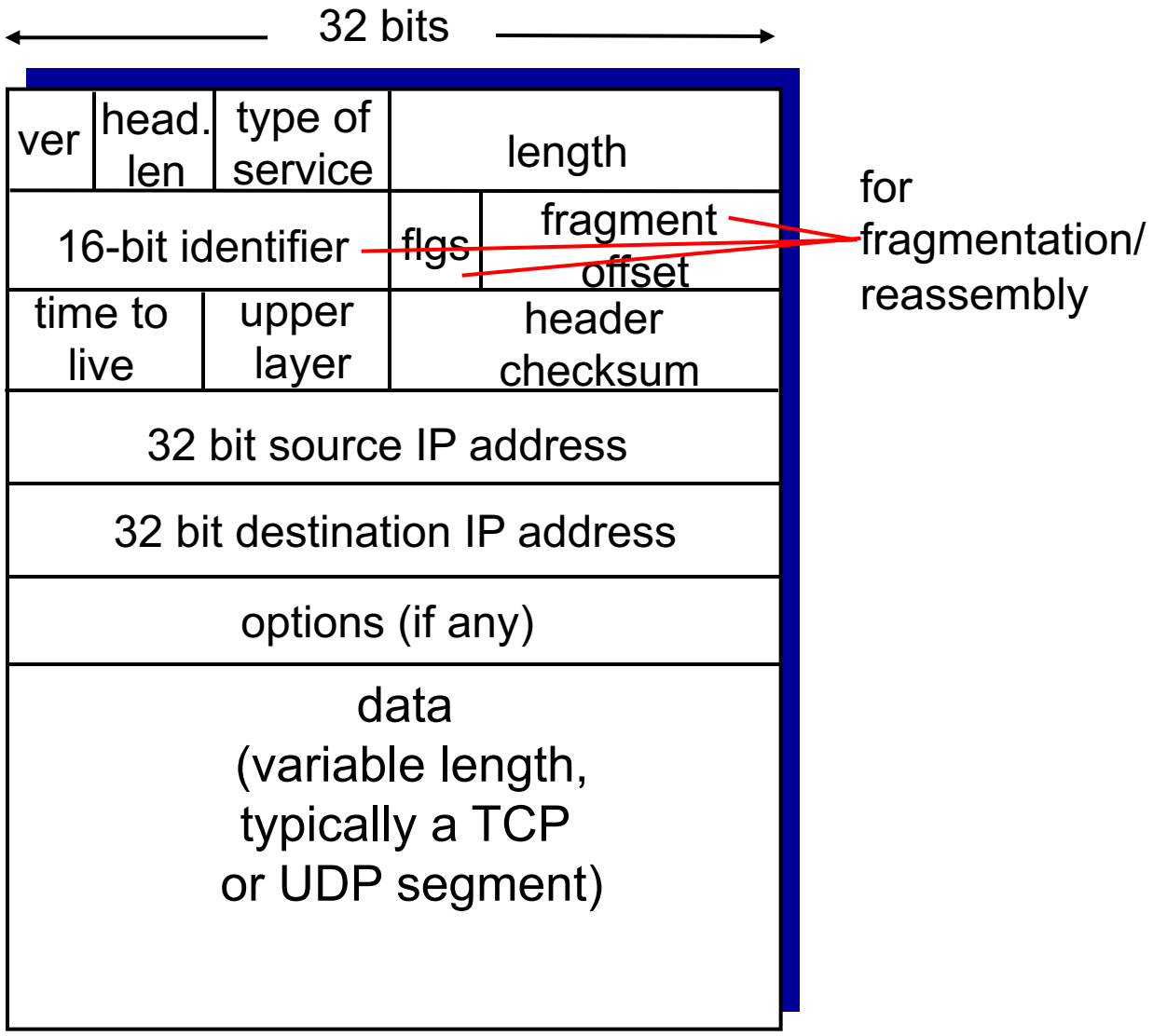
host, router network layer functions:



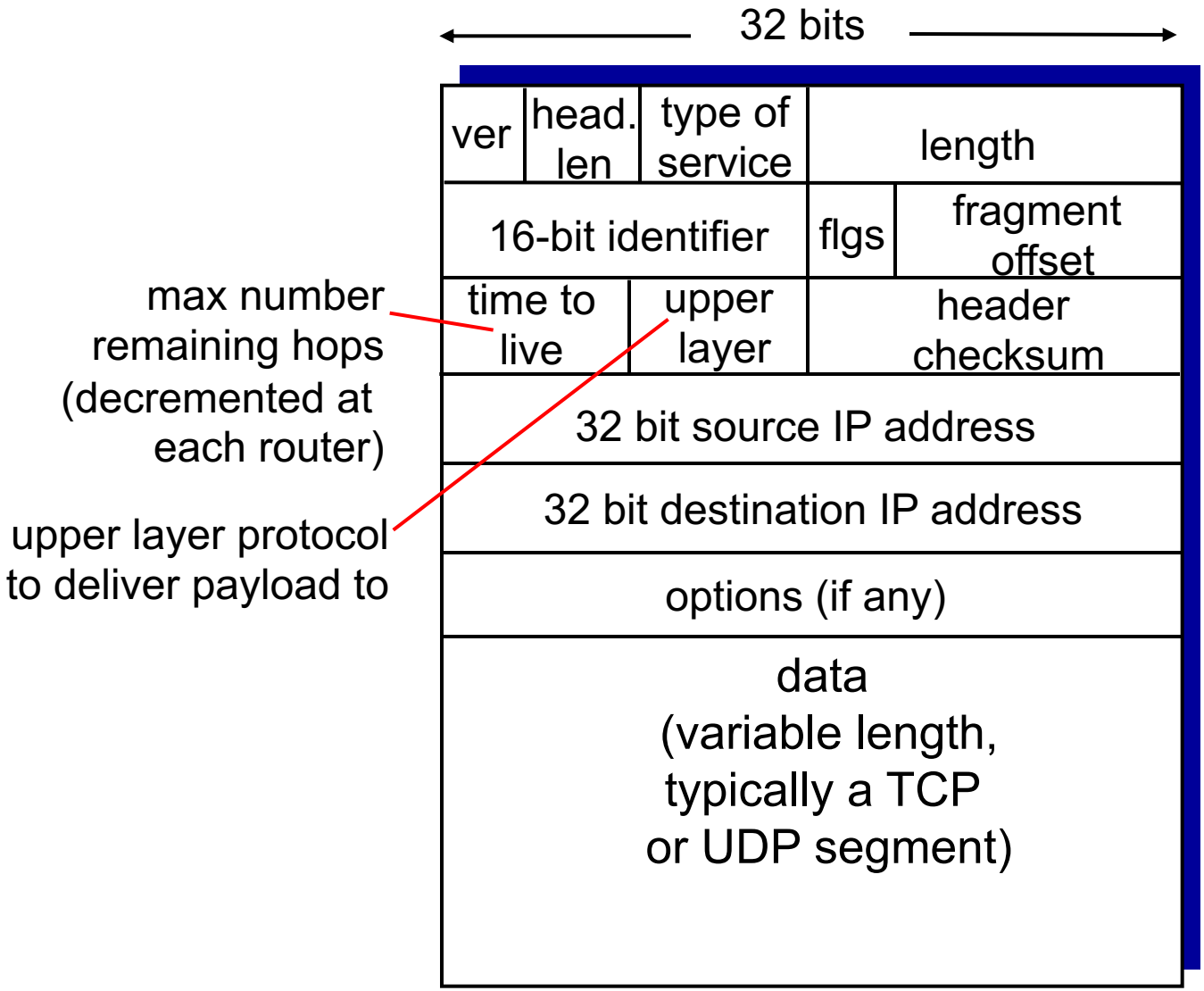
IPv4 datagram format



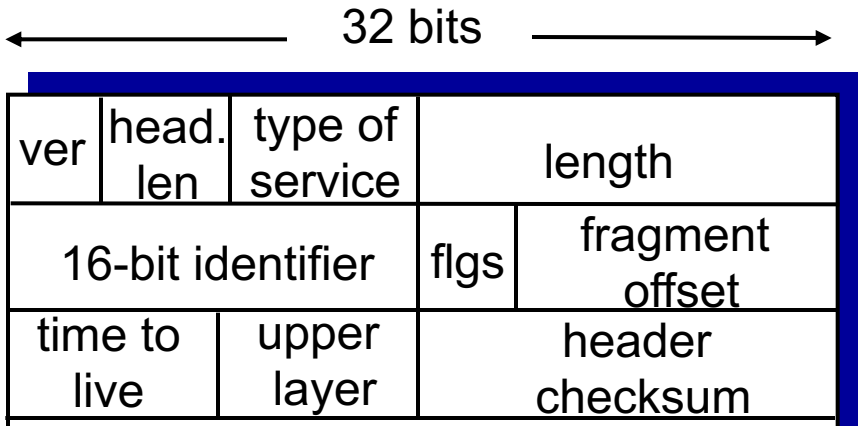
IPv4 datagram format



IPv4 datagram format



IPv4 datagram format

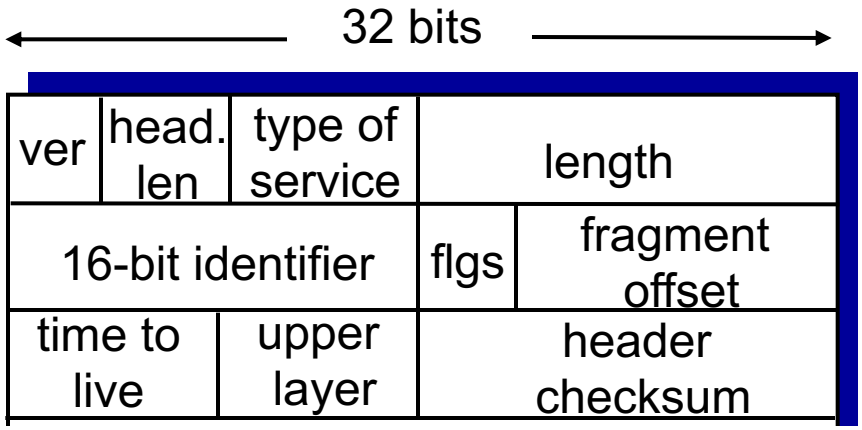


Upper-layer protocol field in IP header server purpose analogous to port number in transport-layer header

Port number allows transport layer to determine which application to pass data to

This field helps network layer determine which transport-layer protocol to pass data to

IPv4 datagram format

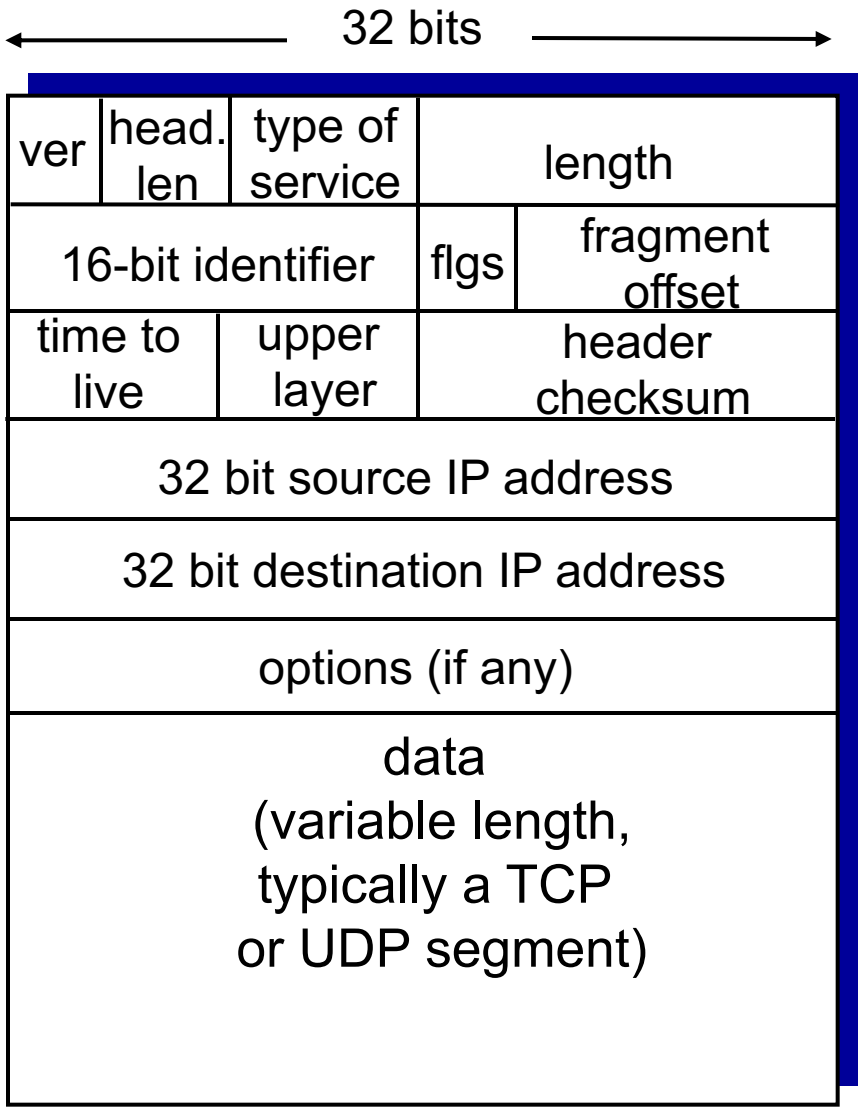


Header checksum computed similarly to UDP – header broken into 16-bit chunks and added together, then one's complement taken

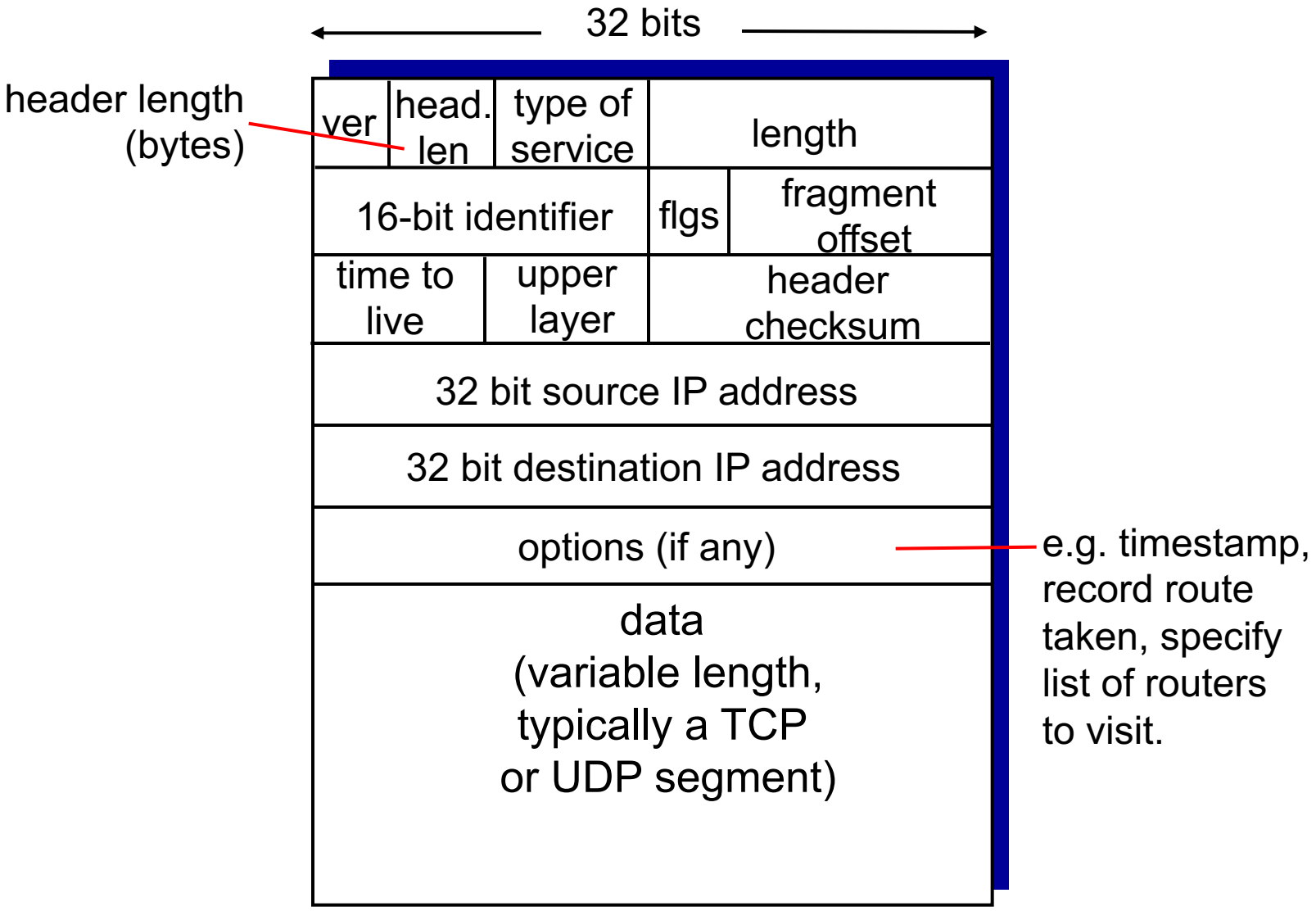
Checksum taken only over header fields, not payload

TTL field in header decreases every hop, so checksum must be checked and recomputed at every router

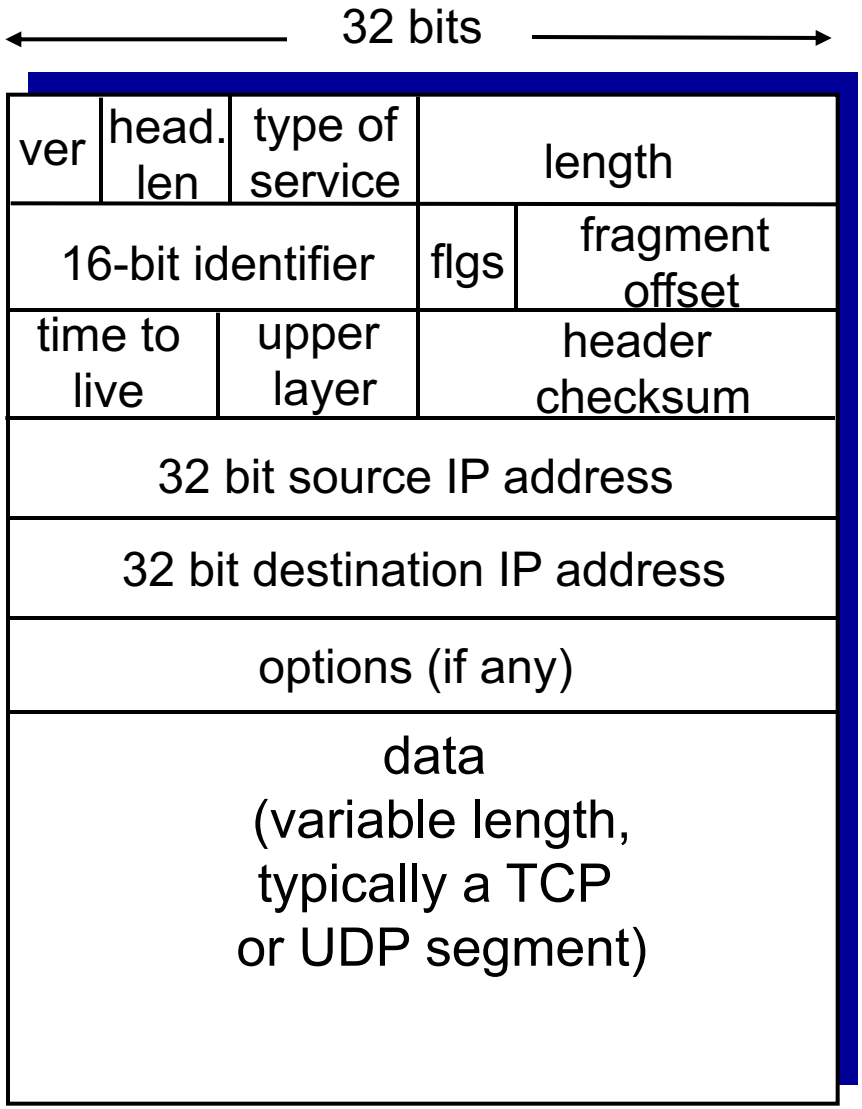
IPv4 datagram format



IPv4 datagram format



IPv4 datagram format

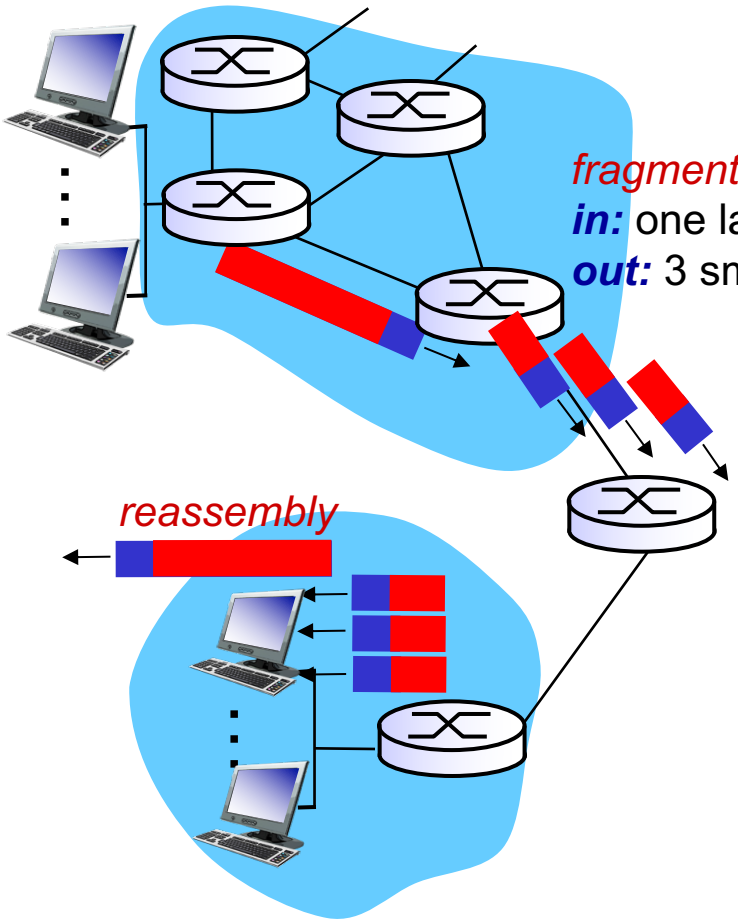


how much overhead?

- 20 bytes of TCP
- 20 bytes of IP
- = 40 bytes + app layer overhead

IP fragmentation, reassembly

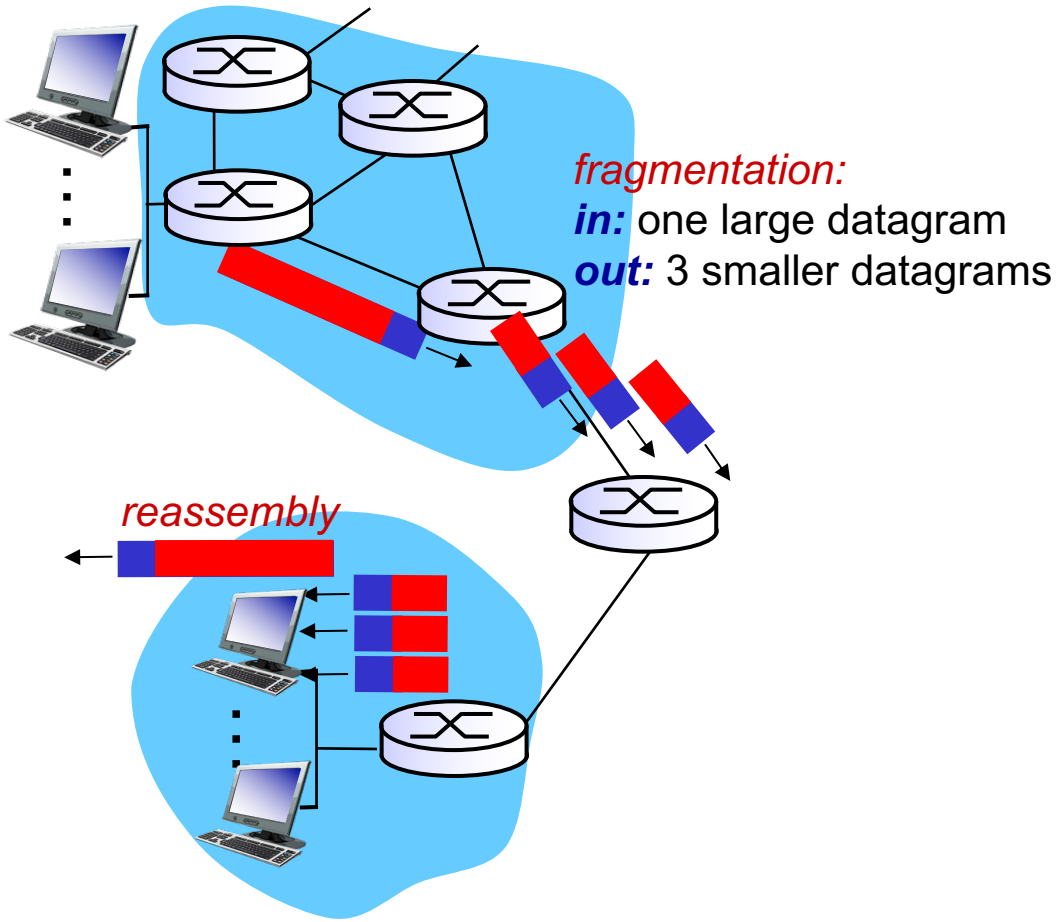
- network links have MTU (max.transfer size) - largest possible link-level frame
  - different link types, different MTUs
- large IP datagram divided ("fragmented") within net
  - one datagram becomes several datagrams
  - "reassembled" only at final destination
  - IP header bits used to identify, order related fragments



IP fragmentation, reassembly

- Need to know several pieces of information about fragments to successfully reassemble

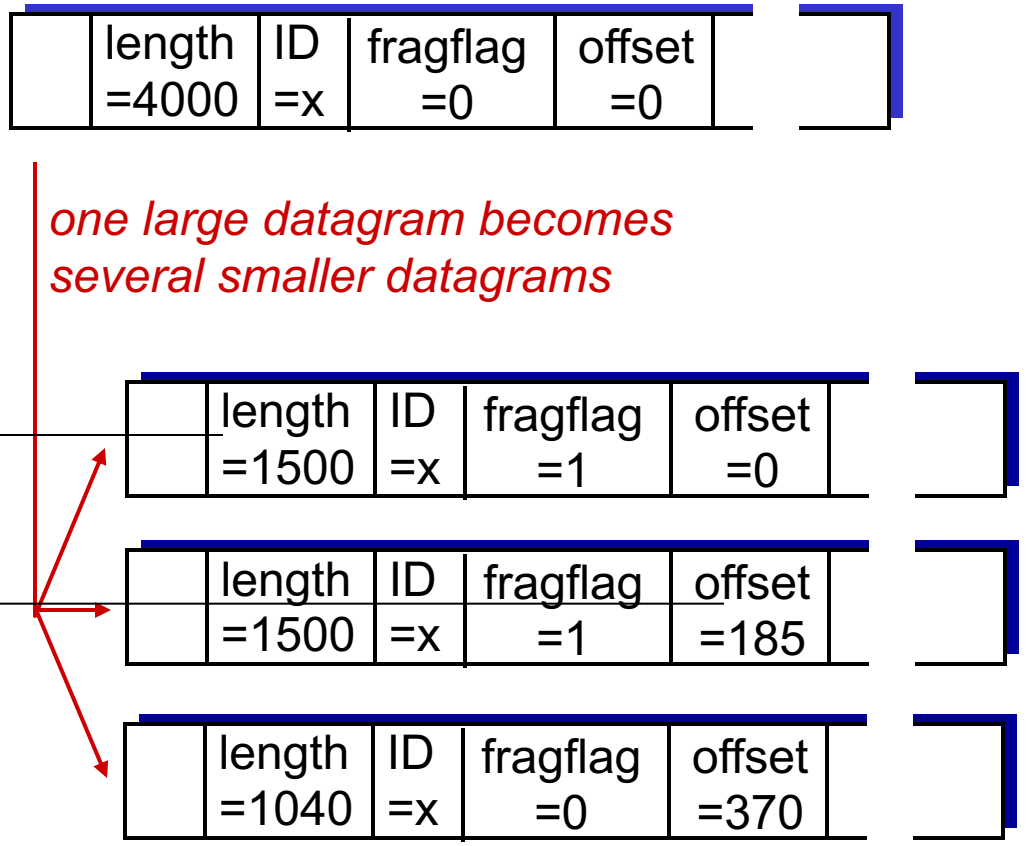
- Is datagram fragmented?
- Have we received all fragments?
- What is the order of fragments?



IP fragmentation, reassembly

example:

- 4000 byte datagram
- MTU = 1500 bytes



IP fragmentation, reassembly

- Note that IP not a reliable protocol, so no need for ID # in general
- ID number always added, but not generally useful except for reassembling after fragmentation