COMPUTER NETWORK SCE 4303

Network Layer

Dr. Yahaya Coulibaly

The Goals:

- Understand principles of network layer :
  - network layer service models
  - forwarding versus routing
  - router working fundamentals
  - routing (path selection)

- instantiation, implementation in the Internet

1. Principles of Network Layer

Network layer

- Responsible for the transport of data segments from sending to receiving host
- On the sending side it encapsulates segments into datagram
- On the receiving side, it delivers segments to transport layer

Remember

- Network layer protocols are implemented in *every* host, router
- Router examines header fields in all IP datagrams passing through it before forwarding it to next node.

Two key network-layer functions

There are mainly two functions executed at Network layer:

- *forwarding:* moving packets from router's input to appropriate router output
- *routing:* determine route taken by packets from source to destination.

Network service model

Network Architecture	Service Model	Guarantees ? Bandwidth	Loss	Order	Timing	Congestion feedback
Internet	best effort	none	no	no	no	no (inferred via loss)
ATM	CBR	constant rate	yes	yes	yes	no congestion
ATM	VBR	guaranteed rate	yes	yes	yes	no congestion
ATM	ABR	guaranteed minimum	no	yes	no	yes
ATM	UBR	none	no	yes	no	no

Connection, connection-less service

Connection, connection-less service

At Network layer:

- *Datagram* network provides network-layer *connectionless* service (e..g apply in Internet architecture)
- *Virtual-circuit (VC)* network provides network-layer *connection* service (e.g. apply in ATM architecture)

Datagram networks (connectionless)

In Datagram Networks:

- No call setup at network layer
- Routers do no keep state of end-to-end connections
- Packets are forwarded using destination host address

Inside the Router

Router architecture overview

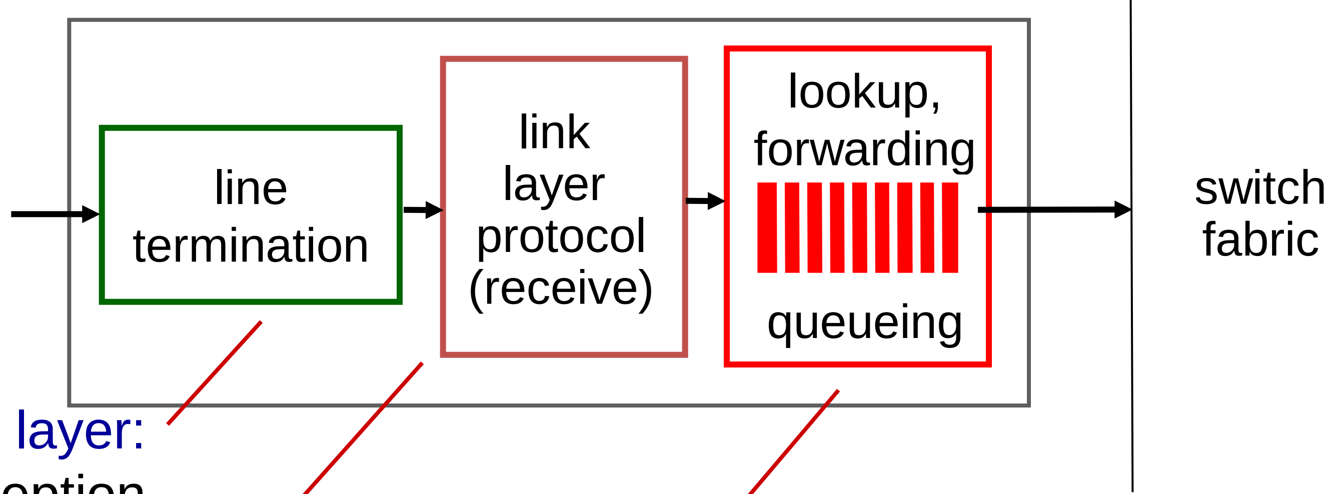
Routers perform two main functions:

- Run routing algorithms/protocol (RIP, OSPF, BGP)
- *Forward* datagrams from incoming to outgoing link

router input ports

router output ports

Input port functions



physical layer:
bit-level reception

data link layer:
e.g., Ethernet
see chapter 5

decentralized switching:

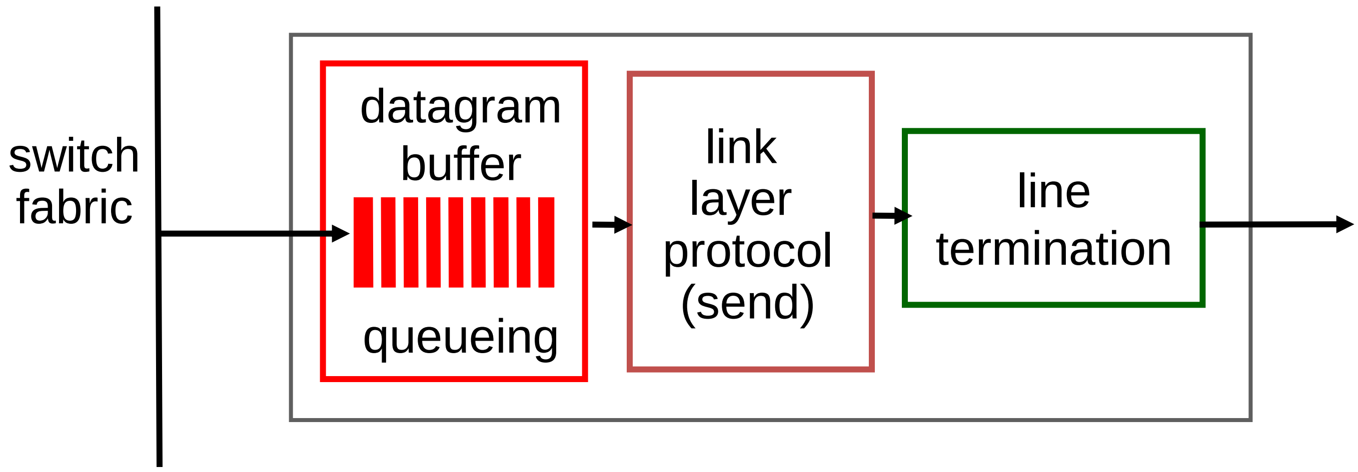
- given datagram dest., lookup output port using forwarding table in input port memory *("match plus action")*
- goal: complete input port processing at 'line speed'
- queuing: if datagrams arrive faster than forwarding rate into switch fabric

Switching fabrics

- transfer packet from input buffer to appropriate output buffer
- switching rate: rate at which packets can be transfer from inputs to outputs
  - often measured as multiple of input/output line rate
  - N inputs: switching rate N times line rate desirable
- There are three types of switching fabrics

1. Memory
2. Bus
3. Crossbar

Output ports

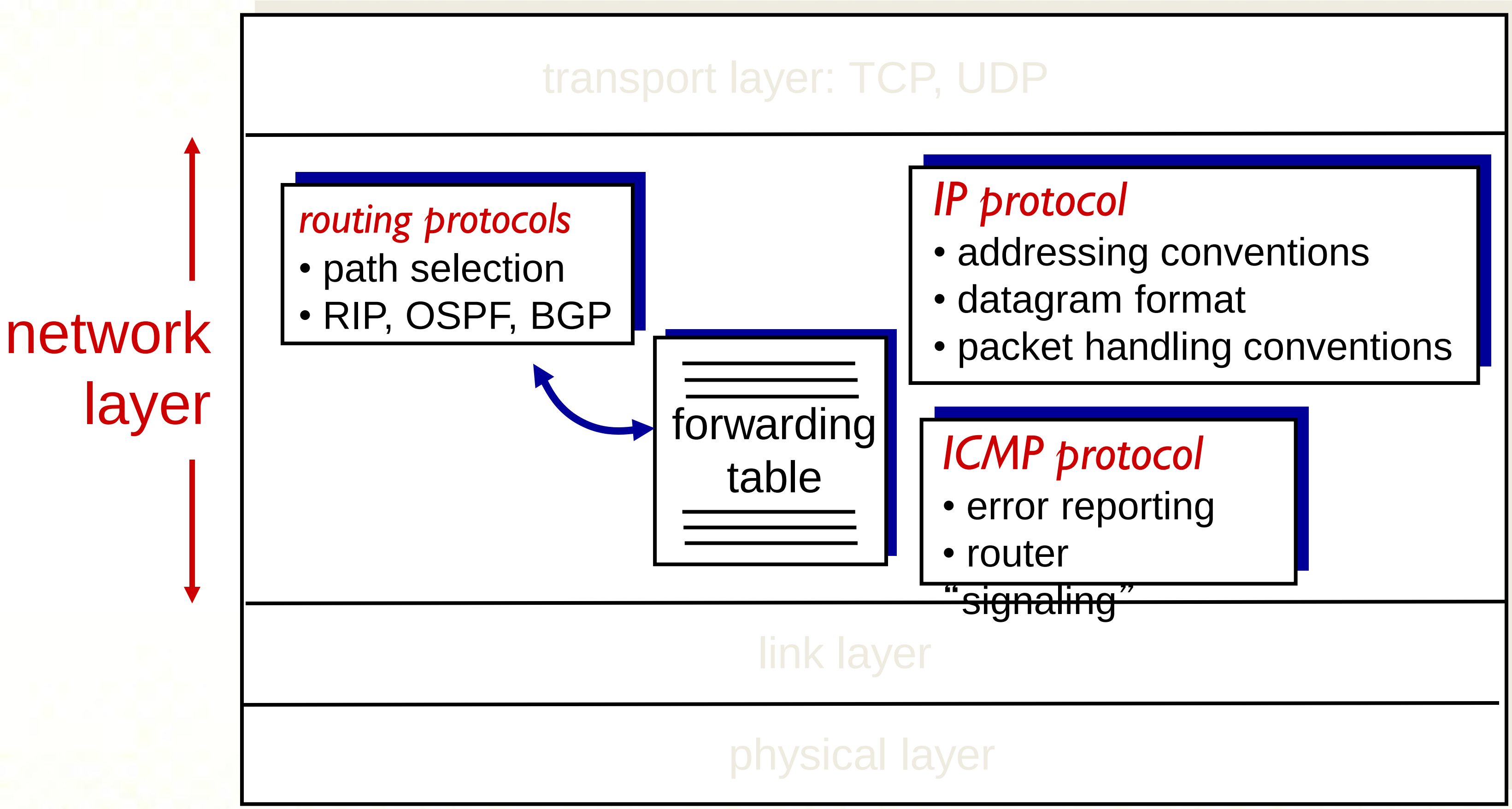


- *buffering* required when datagrams arrive from fabric faster than the transmission rate
- *scheduling discipline* chooses among queued datagrams for transmission

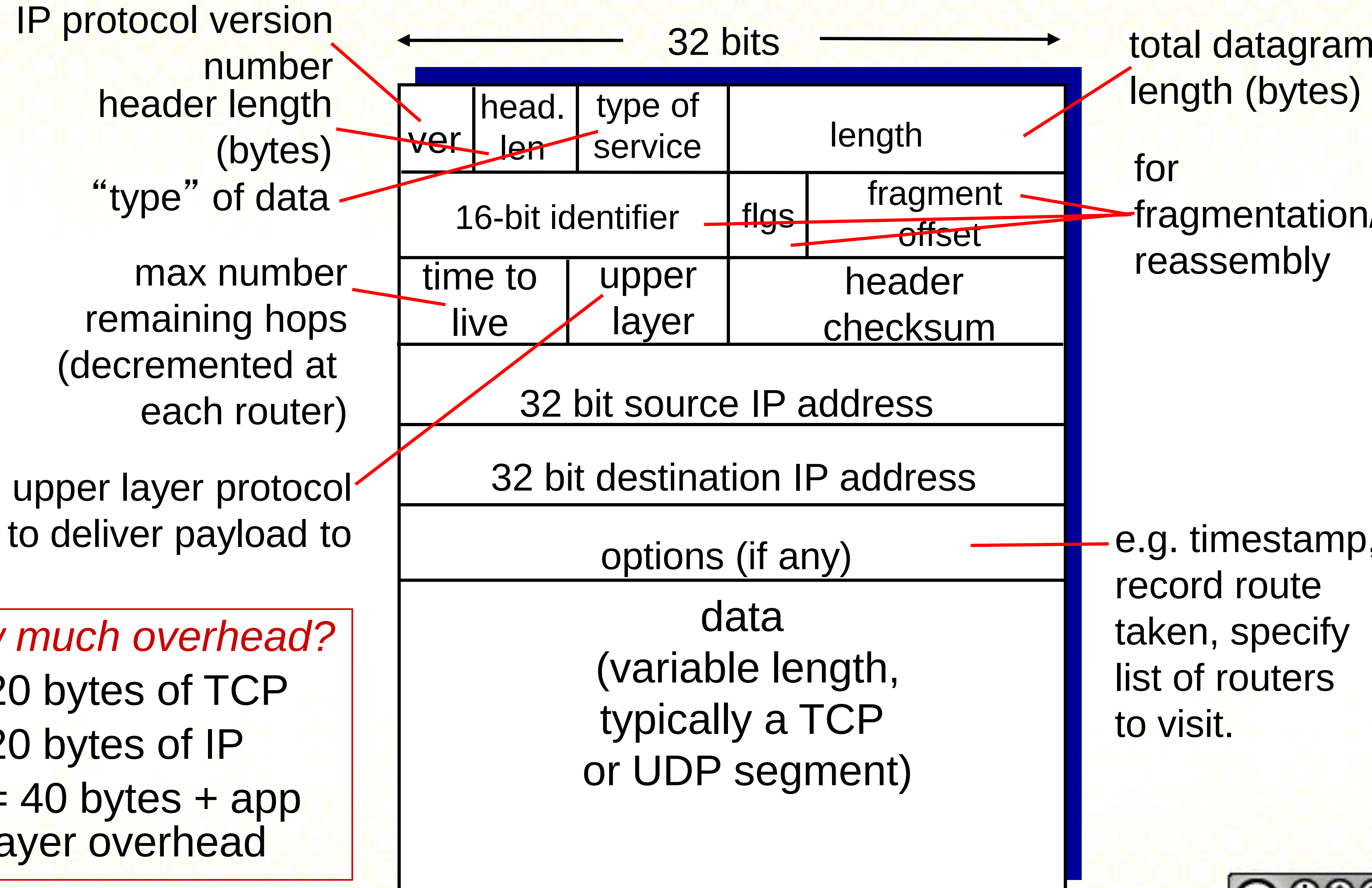
IP: Internet Protocol

The Internet network layer

host, router network layer functions:



IP 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 that determines the size of packet to be transmitted.
  - *different link types, different MTUs*
- Thus, large IP datagram divided ("fragmented") within network
  - one datagram becomes several datagrams
  - "reassembled" only at final destination
  - IP header bits are used to identify, order related fragments

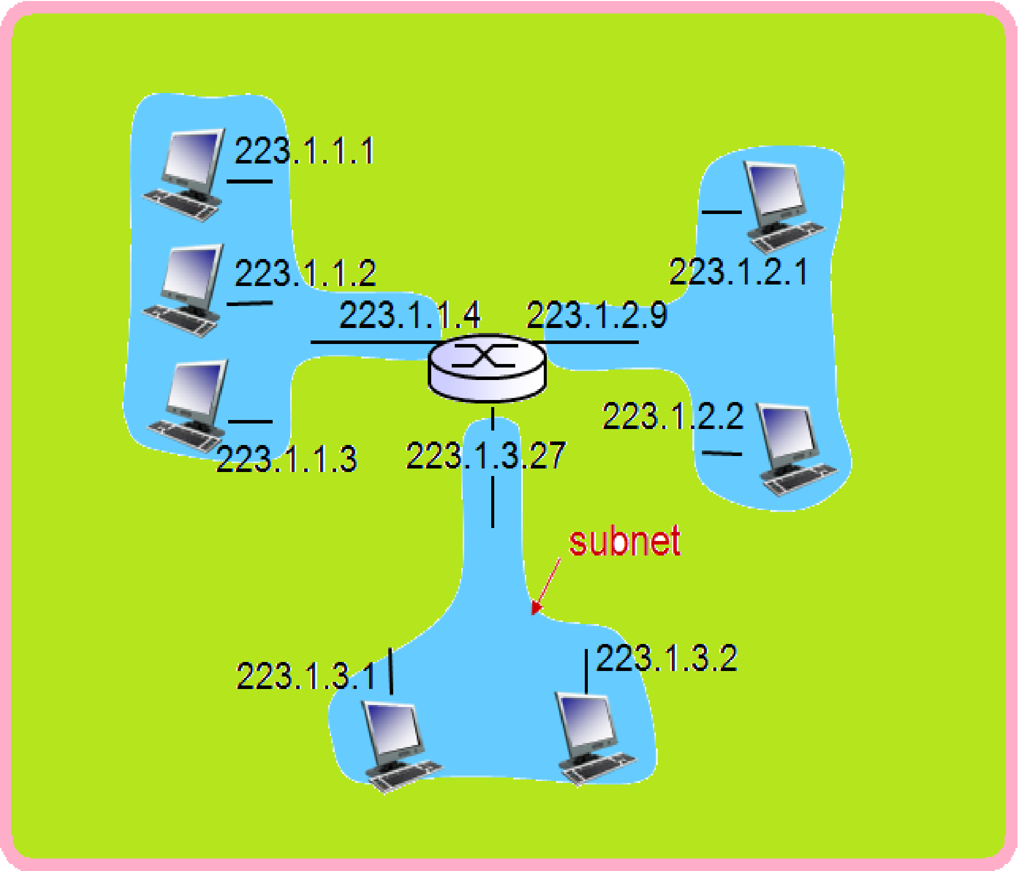
IPv4 addressing

IP addressing: introduction

- *IP address:* 32-bit identifier for host, router *interface*
- *interface:* connection between host/router and physical link
  - router's typically have multiple interfaces
  - host typically has one or two interfaces (e.g., wired Ethernet, wireless 802.11)
- *One IP addresses associated with each interface*

Subnets

- IP address:
  - –subnet part: consists of high order bits
  - –host part – contains low order bits
- *what's a subnet ?*
  - –device interfaces with same subnet part of IP address
  - –can physically reach each other *without intervening (or overriding) router*



network consisting of 3 subnets

IP addressing: CIDR (Classless)

CIDR: Classless InterDomain Routing

- more flexibly than original system of Internet Protocol (IP) address scheme i.e. classful addressing : A (subnet -8bit), B (subnet -16bit), C (subnet -24 bit))
- can avoid situations where large numbers of IP addresses are unused
- subnet portion of address of arbitrary length
- address format: a.b.c.d/x, where x is # bits in subnet portion of address.

subnet part ← → | host part

11001000 00010101 00010010 00000000

e.g. 200.21.18.0/23

DHCP: Dynamic Host Configuration Protocol

goal: allow host to *dynamically* obtain its IP address from network server when it joins network

- can renew its lease on address in use
- allows reuse of addresses (only hold address while connected/"on")
- support for mobile users who want to join network (more shortly)

DHCP overview:

- host broadcasts "DHCP discover" msg [optional]
- DHCP server responds with "DHCP offer" msg [optional]
- host requests IP address: "DHCP request" msg
- DHCP server sends address: "DHCP ack" msg

NAT: network address translation

motivation: local network uses just one IP address as far as outside world is concerned:

- range of addresses not needed from ISP: just one IP address for all devices
- can change addresses of devices in local network without notifying outside world
- can change ISP without changing addresses of devices in local network
- devices inside local net not explicitly (precisely) addressable, visible by outside world (a security plus)

NAT: network address translation

implementation: NAT router must:

- *outgoing datagrams: replace* (source IP address, port #) of every outgoing datagram to (NAT IP address, new port #)
  . . . remote clients/servers will respond using (NAT IP address, new port #) as destination addr

- *remember (in NAT translation table)* every (source IP address, port #) to (NAT IP address, new port #) translation pair

- *incoming datagrams: replace* (NAT IP address, new port #) in dest fields of every incoming datagram with corresponding (source IP address, port #) stored in NAT table

IP V6

IPv6: motivation

- *initial motivation:* 32-bit address space soon to be completely allocated.
- additional motivation:
  - header format helps speed processing/forwarding
  - header changes to facilitate QoS

IPv6 datagram format:

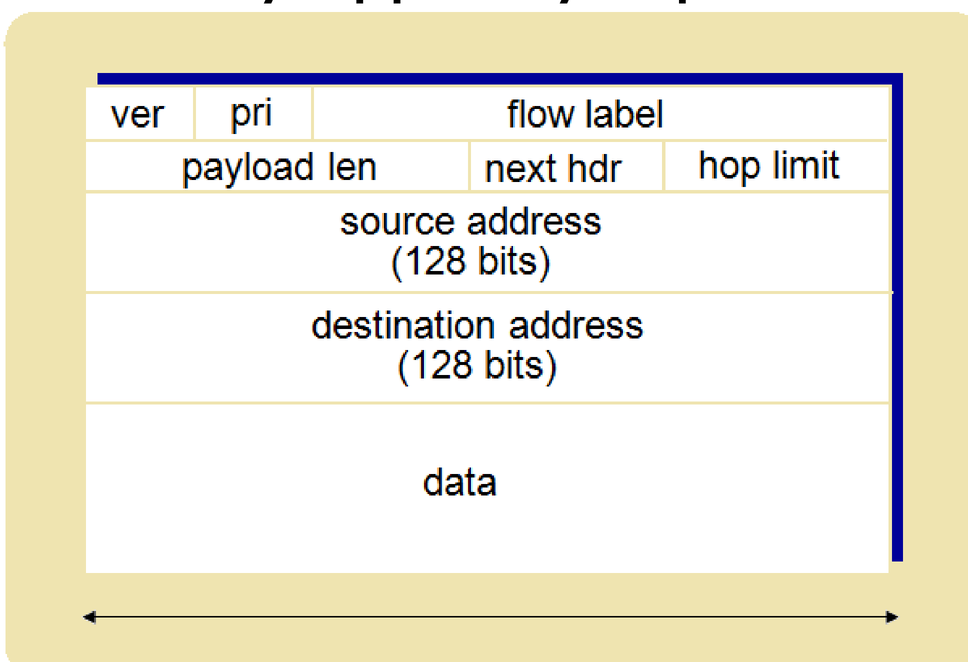
- fixed-length 40 byte header
- no fragmentation allowed

IPv6 datagram format

priority: identify priority among datagrams in flow

flow Label: identify datagrams in same "flow."
(concept of "flow" not well defined).

next header: identify upper layer protocol for data



32 bits