IK2215 Advanced Internetworking

Lecture 2—Network Layer

Markus Hidell

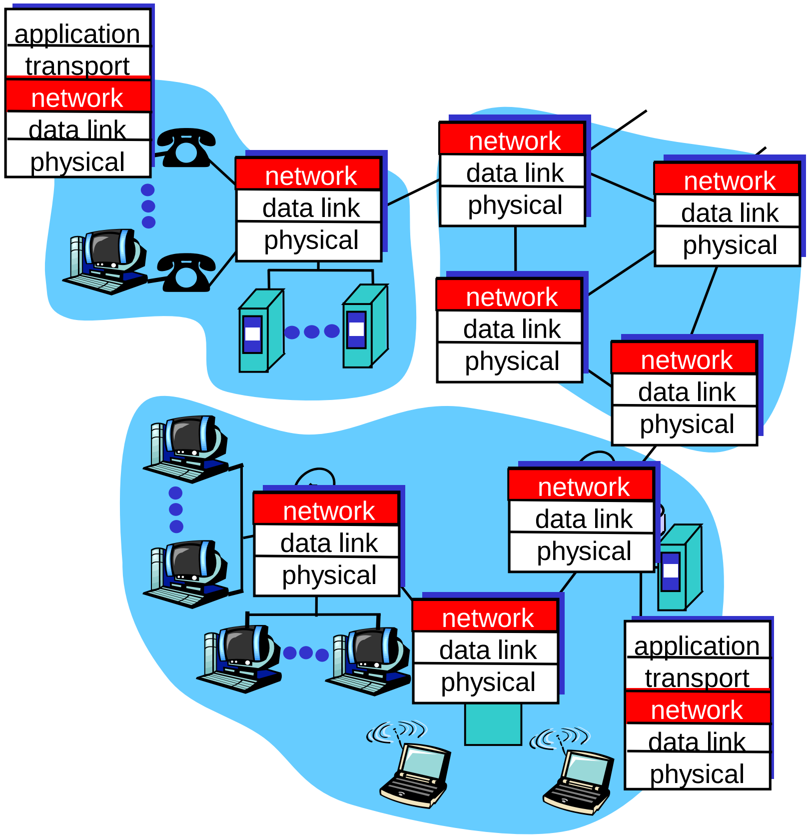
Contents

Kurose and Ross, chapter 4-5 (focus on 4.1-4.3, 5.6)

And then some.....

Network Layer

- Transport segment from sending to receiving host
- Sending side encapsulates segments into datagrams
- Receiving side delivers segments to transport layer
- Network layer protocols in every host and router
- Router examines header fields in all datagrams passing through



Network Layer Services

Connection-Oriented Services

- The network layer establishes a connection between a source and a destination
- Packets are sent along the connection.
- The decision about the route is made *once* at connection establishment
- Routers/switches in connection-oriented networks are stateful

Connectionless Services

- The network layer treats each packet independently
- Route lookup for each packet (routing table)
- IP is connectionless
- IP routers are stateless

Connection-oriented Networks

Some network architectures use network layer *connections*

- ATM, frame relay, X.25
- *Virtual circuits*

Before datagrams can flow, end-hosts and intervening routers establish a virtual circuit

Network vs transport layer connection service:

- Network: between hosts through routers
- Transport: between two processes (routers don’t care)

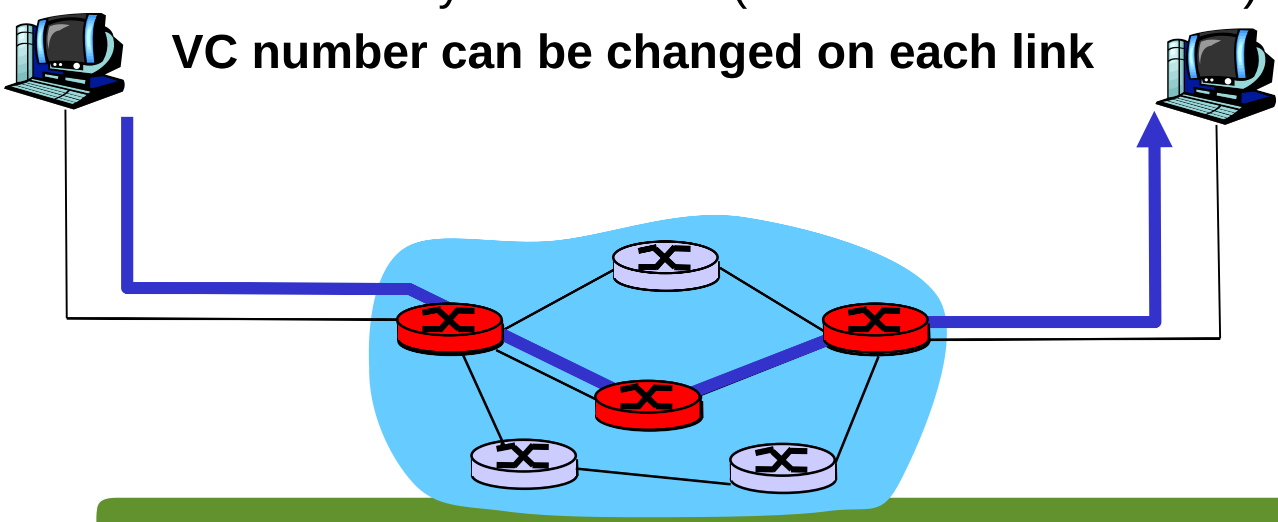
Virtual Circuits

VC consists of

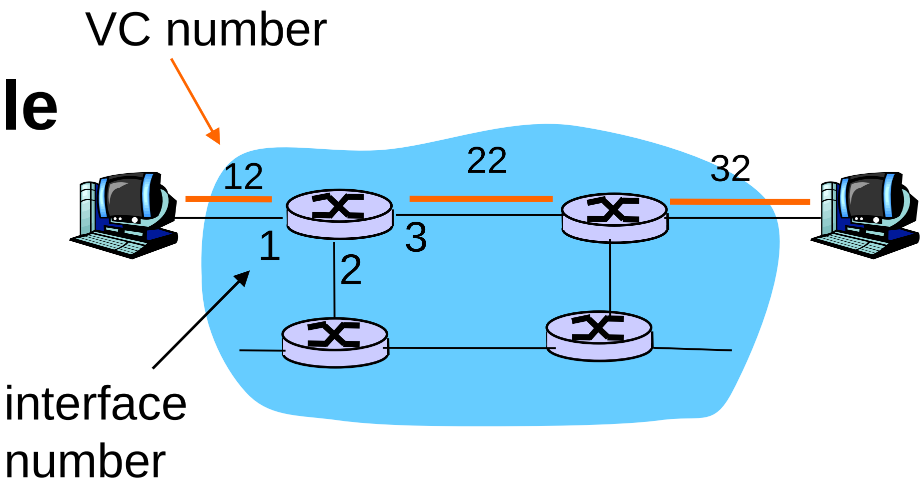
- Path from source to destination
- VC numbers (VC identifiers), one for each link along the path
- Entries in forwarding tables in router along the path

Packets carry VC number (not destination address)

VC number can be changed on each link



VC Forwarding Table



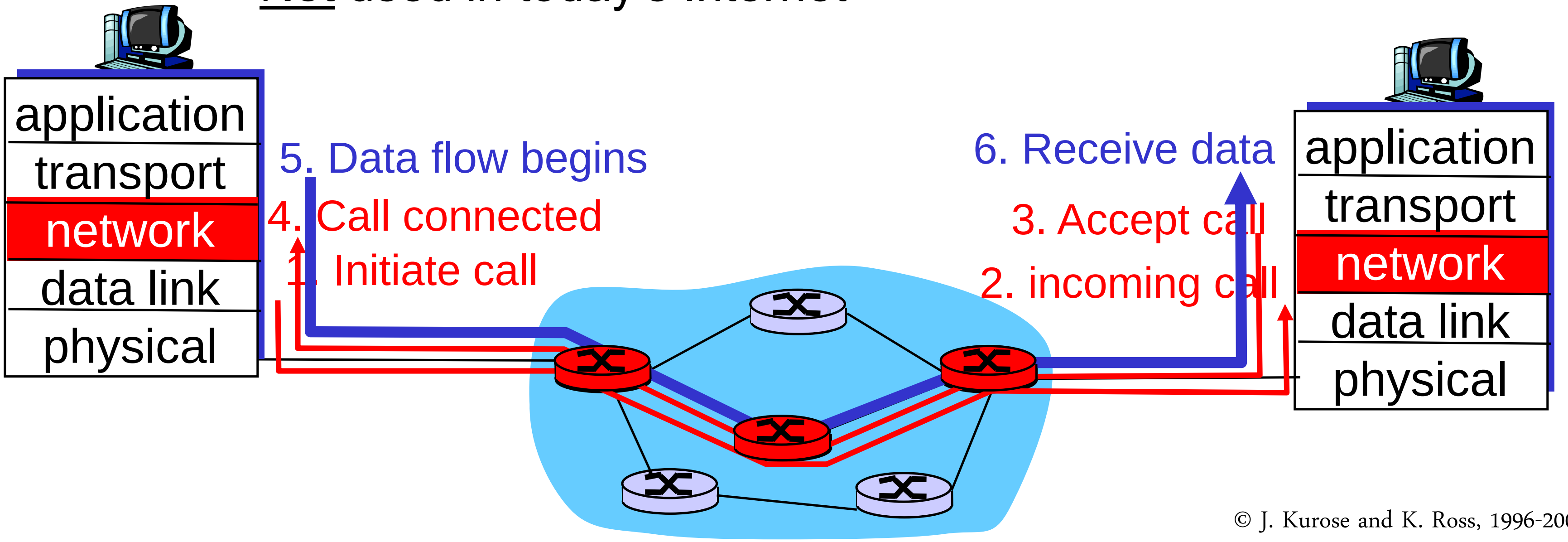
Forwarding table in northwest router:

Incoming IF	Incoming VC #	Outgoing IF	Outgoing VC #
1	12	3	22
2	63	1	18
3	7	2	17
1	97	3	87
...	...	...	...

Routers maintain connection state information!

Virtual Circuits: Signalling Protocols

- To setup, maintain, and teardown VCs
- Used in e.g., ATM, frame-relay, X.25
- Not used in today's Internet



Connectionless Networks

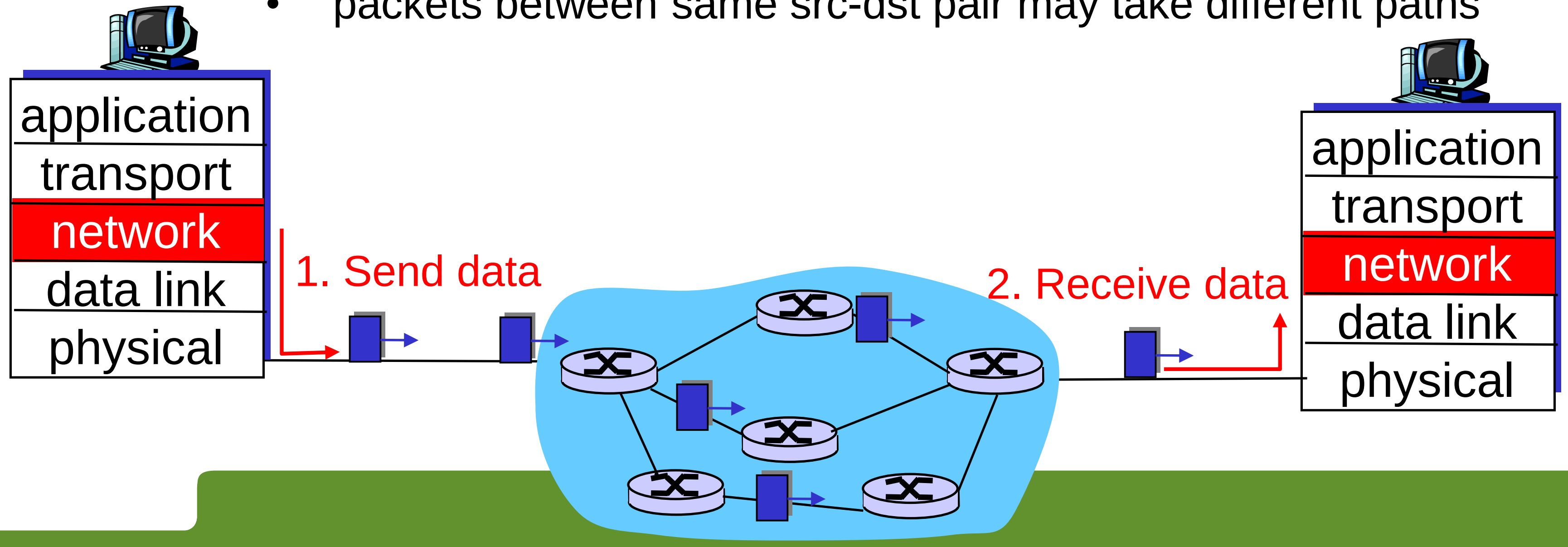
No call setup at the network layer

Routers: no state about end-to-end connections

- no network-level concept of “connection”

Packets forwarded using destination host address

- packets between same src-dst pair may take different paths



Issues in IP

Following the end2end argument, only the absolutely necessary functionality is in IP

- Best Effort Service: Unreliable and Connectionless
- Application or Transport layer handles reliability

How to deliver datagrams over multiple links (hops) in an internetwork?

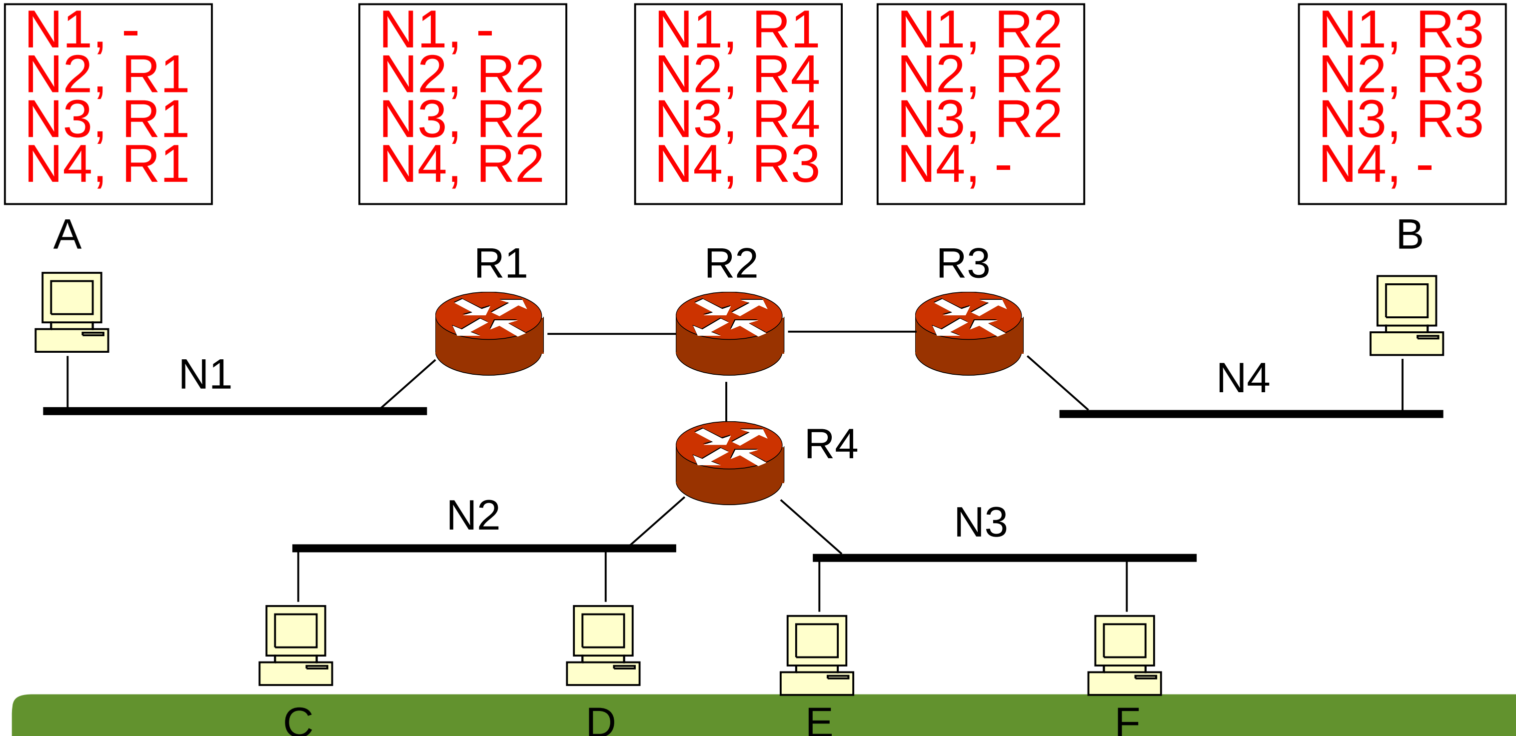
- Addressing
  - Covered earlier and should be ”prior knowledge” to you
- Best-effort delivery service
  - Forwarding of packets from one link to another
- Error handling

Next-hop Routing

How do you hold information about route from A to all other hosts?

- A → R1 → R2 → R3 → B

Table of *host/network address* and *next-hop* in every node



Internet Routing Tables

One entry per IP address → 4 billion possible entries

- Not practical for storing and searching!

The basic idea with IP addressing (and CIDR) is to *aggregate* addresses

- more specific networks (with longer prefixes) → less specific networks (with shorter prefixes)

More aggregation leads to *smaller* routing tables

The ideal situation is to have domains publishing (exporting) only a small set of prefixes

- Effective address assignment policy

Current routing tables (# of entries) are around 600000 entries long (over 50% are /24 prefixes)

Longest Prefix Matching

Prefix Match	Link Interface
11001000 00010111 00010	0
11001000 00010111 00011000	1
11001000 00010111 00011	2
otherwise	3

Examples

DA: 11001000 00010111 00010110 10100001 — **Which interface?**

DA: 11001000 00010111 00011000 10101010 — **Which interface?**

Longest Prefix Matching, cont'd

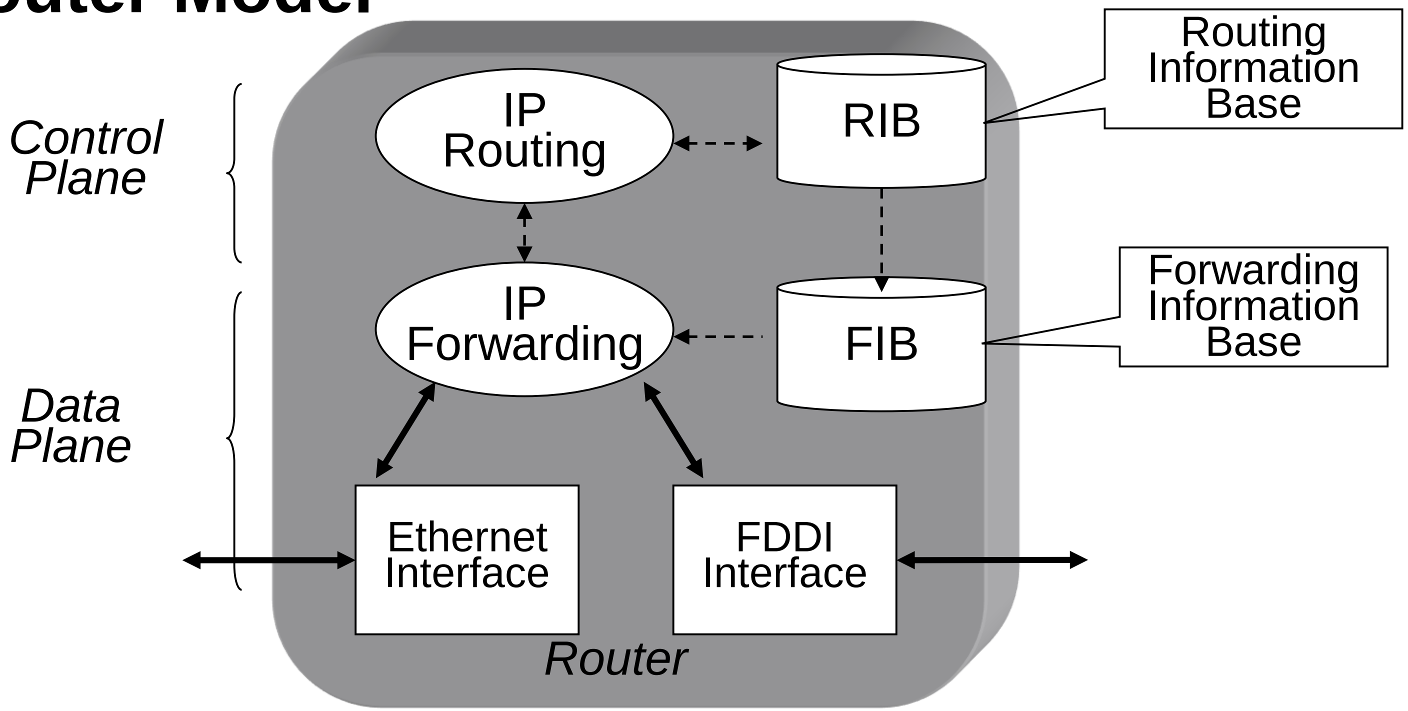
Search for the most specific entry that matches the address

129.12.4.231

0.0.0.0/0
129.12.0.0/16
129.12.4.0/20
129.12.4.0/24
130.23.43.0/24
129.12.4.64/25
129.12.4.192/26
129.12.4.43/32

Best Matching Prefix

IP Router Model



A Router can be partitioned into a dataplane and a controlplane

- The dataplane is fast and special purpose – handles packet *forwarding* in real-time
- The control plane is general purpose– handles *routing* in the background

IP Forwarding

A router switches packets between network interfaces

Extracts header information from the incoming datagram

- Destination IP address

Makes a lookup in the forwarding information base by making a match against networks

- Next-Hop IP address,
- Outgoing interface,...

Modifies datagram header

Sends on outgoing interface

But a router performs much more than IPv4 lookup

- Access lists, filtering
- Traffic management
- Other protocols: Bridging, MPLS, IPv6, ...

IP Header (Revisited)

Version

HLEN – Header Length

Type of Service

Total Length

- Header + Payload

Fragmentation

- ID, Flags, Offset

TTL – Time To Live

- Limits lifetime

Protocol

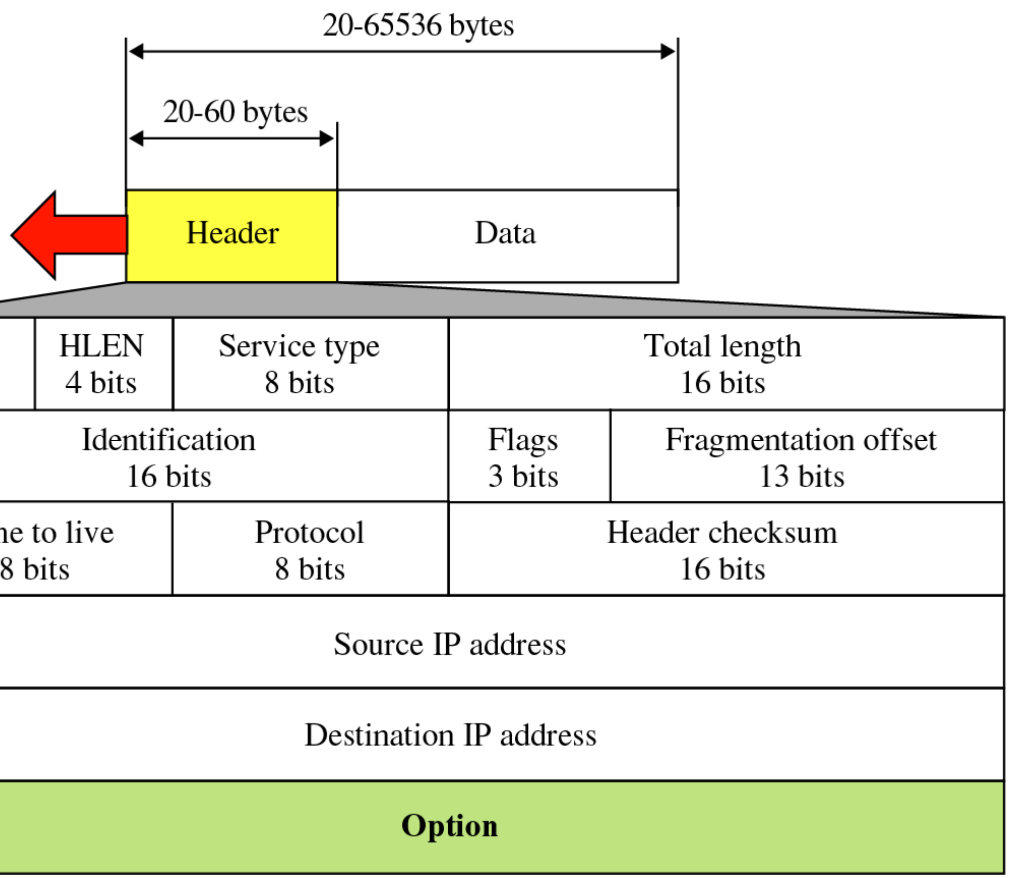
- Higher level protocol

Header checksum

IP Addresses

- Source, Destination

Options



The Length Fields

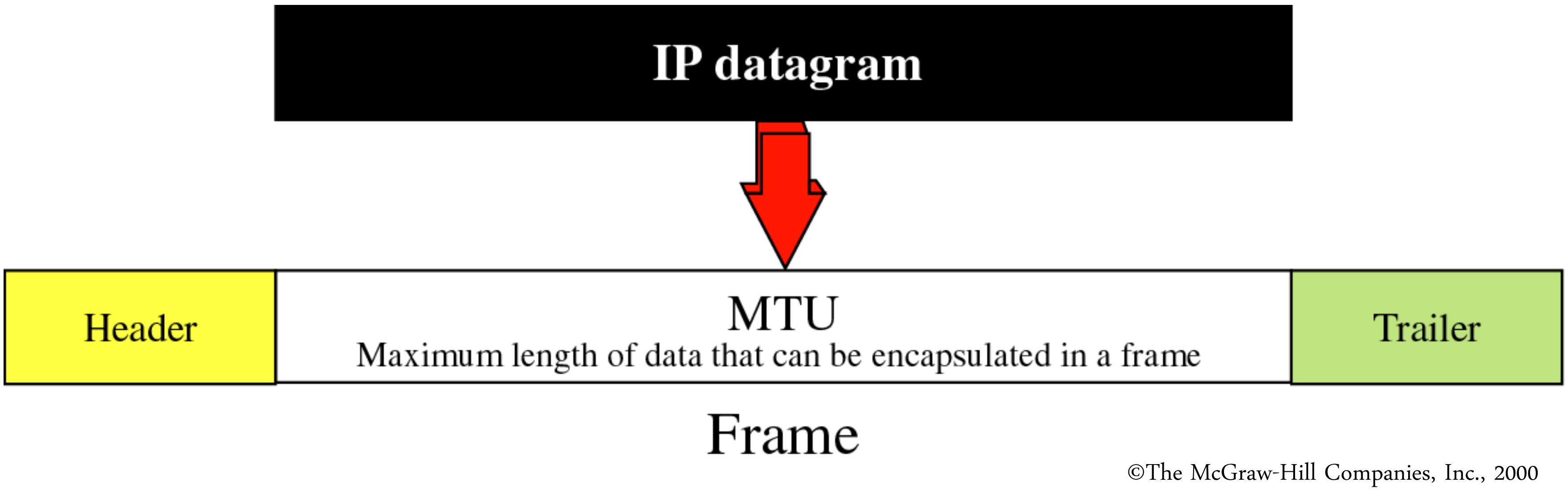
Header Length (4 bits)

- Size of IPv4 header including options.
- Expressed in number of 32-bit words (4-byte words)
- Min is 5 words (=20 bytes)
- Max is 15 words (=60 bytes) – limited size for options → limited use

Total Length (16 bits)

- Total length of datagram including header.
- If datagram is fragmented: length of fragment.
- Expressed in bytes.
  - Max: 65535 bytes. (This is IPs length limit)
  - Many systems only accept 8K bytes

Fragmentation—MTU



If the IP datagram is larger than the MTU of the link layer, it must be divided into several pieces to fit the MTU – this is called *fragmentation*

Fragmentation, cont’d

Physical networks maximum frame size

- MTU Maximum Transfer Unit.

A host or router transmitting datagram larger than MTU of link must divide it into smaller pieces - fragments.

Both hosts and router may fragment

- But only destination host reassemble!
- Each fragment routed separately as independent datagram

In effect, only datagram service (e.g. UDP)

- TCP negotiates MTU during setup and/or path MTU discovery

3 fields of the IP header concerns fragmentation

The Fragmentation Fields

Identification: 16 bits

- ID + src IP addr ~uniquely identifies each datagram sent by a host
- The ID is copied to all fragments of a datagram upon fragmentation

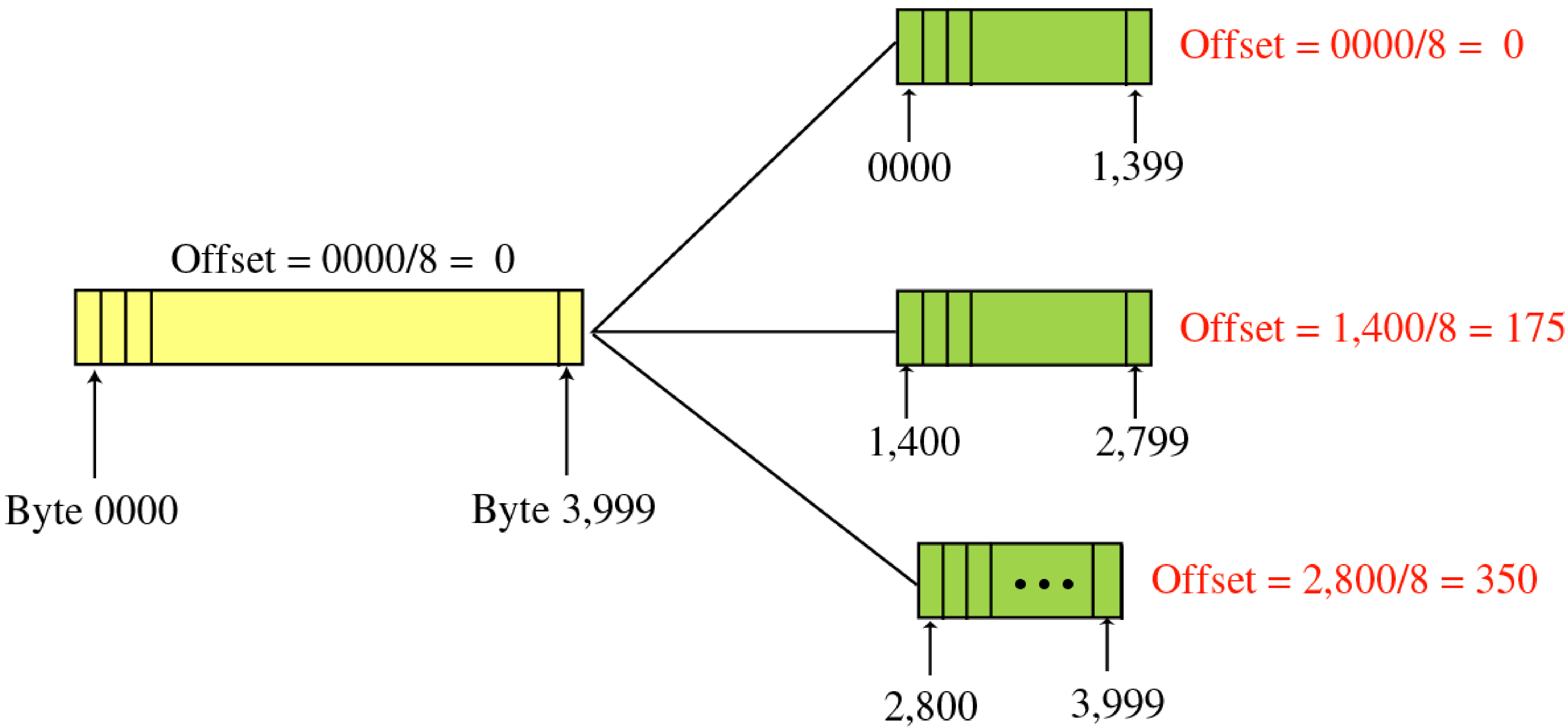
Flags: 3 bits

- RF (Reserved Fragment) – for future use (set to 0)
- DF (Don't Fragment).
  - Set to 1 if datagram should not be fragmented.
  - If set and fragmentation needed, datagram will be discarded and an error message will be returned to the sender
- MF (More Fragments)
  - Set to 1 for all fragments, except the last.

Fragmentation Offset: 13 bits

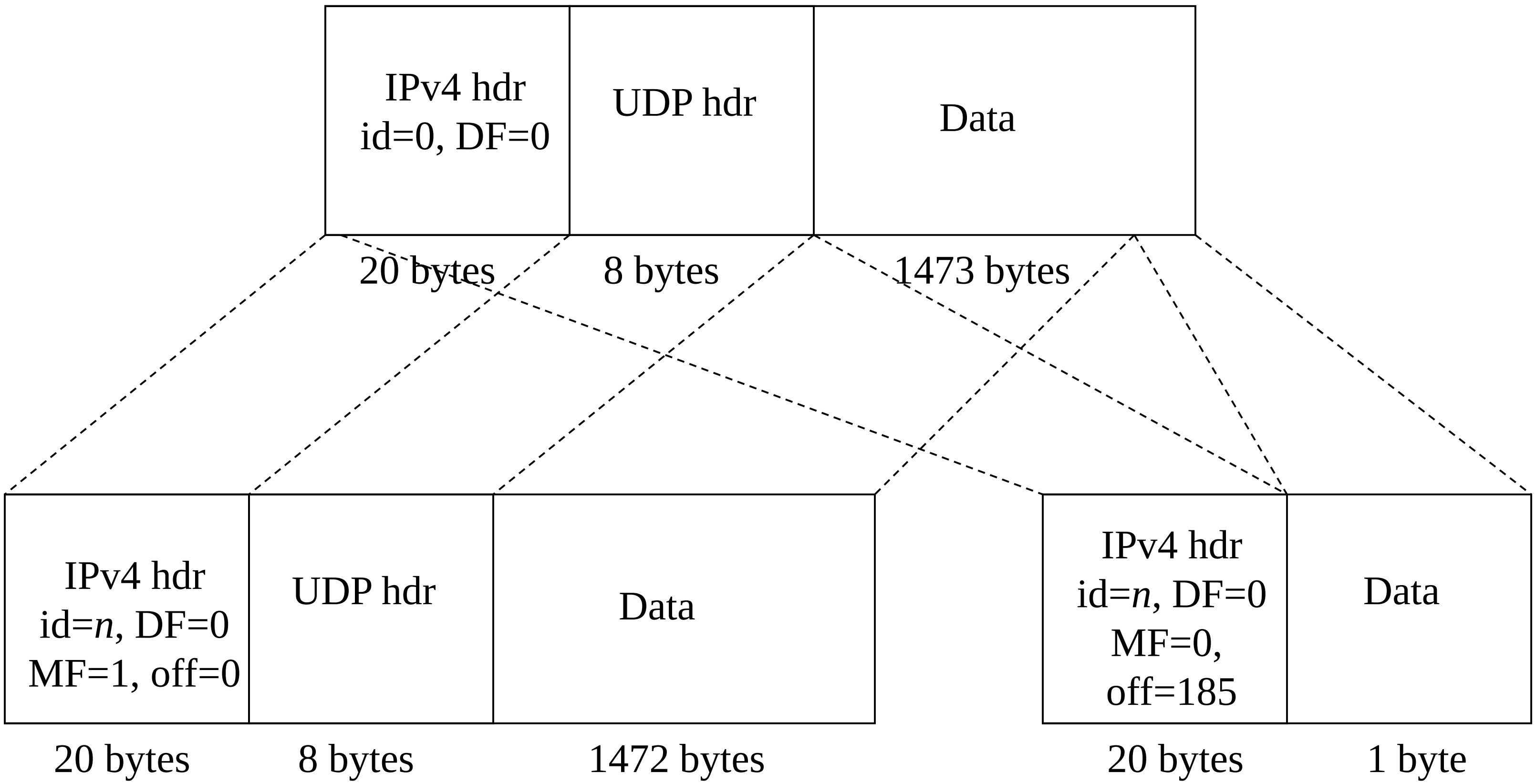
- 8-byte units: (ip→ip_frag << 3)
- Shows relative position of a fragment with respect to the whole datagram

Fragmentation Example



Fragmentation Example—Detailed

MTU = 1500 bytes



Offset = 185 → 185x8 = 1480 bytes

IP Header Checksum

Ensures integrity of header fields

- Hop-by-hop (not end-to-end)
- The header fields must be correct for proper and safe processing.
- The payload is not covered.

Other checksums

- Link-level CRC. IP assumes a strong L2 checksum/CRC. Hop-by-hop.
- L4 checksums, eg TCP/ICMP/UDP checksums cover payload. End-to-end.

Internet Checksum Algorithm, RFC 1071

- Treat header as sequence of 16-bit integers.
- Add them together
- Take the one’s complement of the result

IP Options

IPv4 options are intended for network testing or debugging

Options are variable size and comes after the fixed header

Contiguous – no separators

Fields are optional, but all IP implementations must include processing of options

- In practice many implementations do not!

Max 40 bytes - very limited use

- Max header length is 60 bytes (fixed part is 20 bytes)

IP Options Encoding

Two styles

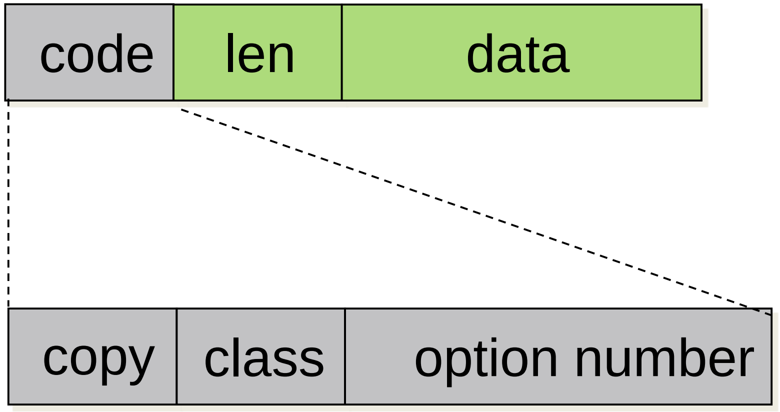
- Single byte (only code)
- Multiple byte

Option Code: 1 byte

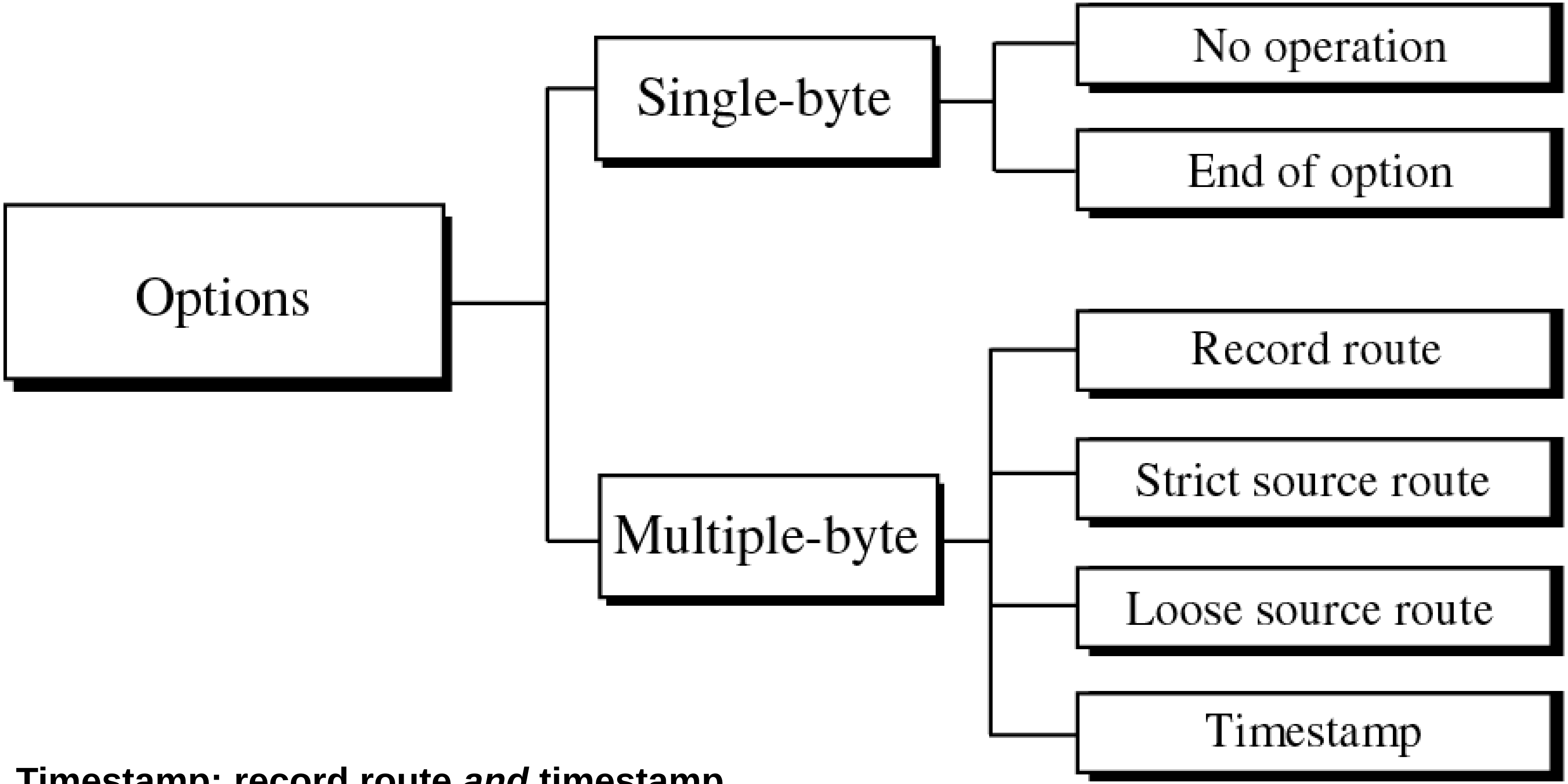
- Copy (to fragments) (1 bit)
- Class (2 bits)
  - 0 (00): Datagram or network control
  - 2 (10): Debugging and measurement
- Number (5 bits)

Option Length (len): 1 byte, defines total length of option (including code and len fields)

Data: option specific



IP Option Types



Timestamp: record route *and* timestamp

IP Option Example: Record Route

Each router records its address

The destination processes the trace

- E.g. sends the result back to the sender

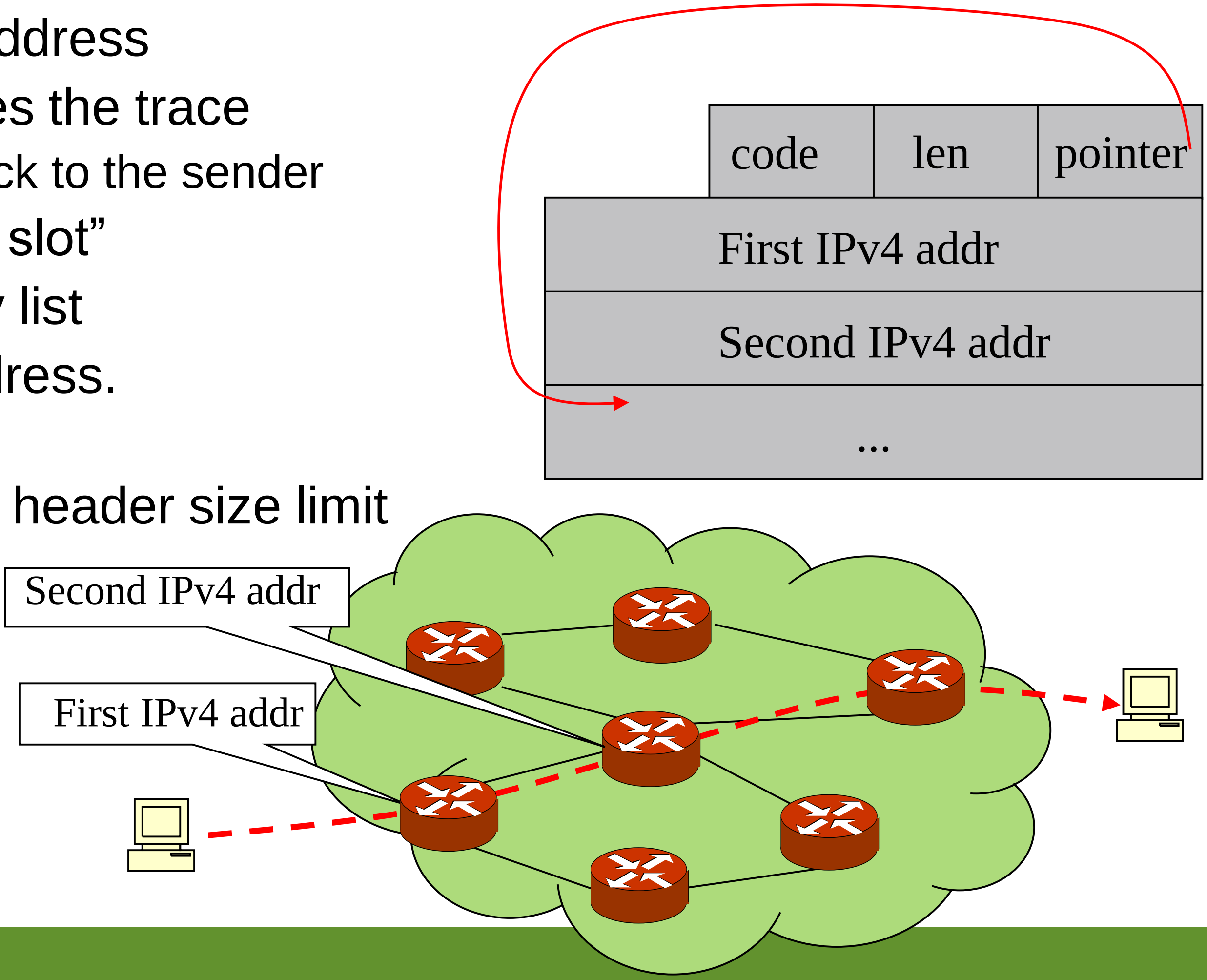
Pointer is ”next available slot”

Source creates an empty list

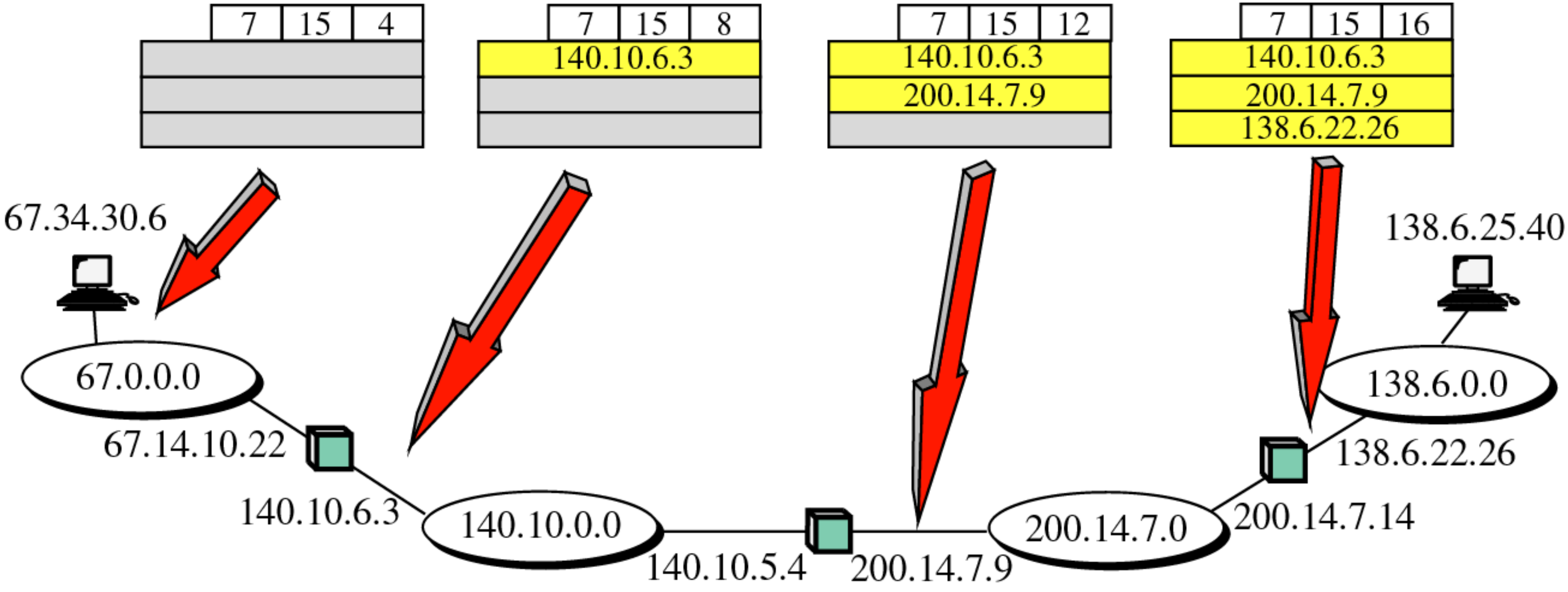
Every router adds its address.

- Increments pointer

Limited to nine hops – IP header size limit

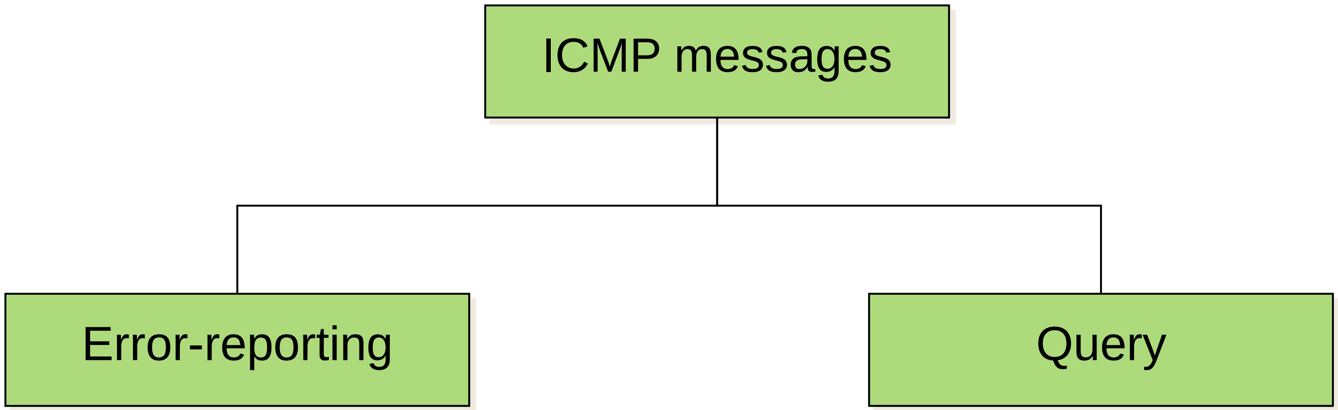


IP Options: Record Route Example



Note that pointer is an index, starting with *code* at index 1

ICMP—Internet Control Message Protocol



Type	Message
3	Destination unreachable
4	Source quench
11	Time exceeded
12	Parameter problem
5	Redirection

Type	Message
8/0	Echo request/reply
13/14	Timestamp request/reply
17/18	Address mask request/reply
10/9	Router solicitation/advertisement

ICMP Error Reporting

One of the main responsibilities of ICMP

- Recall that IP is an unreliable protocol, and errors may occur

ICMP does not correct errors

- Left to higher level protocols

Error messages are always sent back to the *original source*

- Because the only information available in the datagram about the route is the source and destination IP addresses

ICMP uses the source address of the IP packet to send the error message back to the source (originator)

ICMP Error Restrictions

An ICMP Error is not returned in response to:

- A datagram carrying another ICMP Error
- A datagram destined to IP broadcast or multicast address
- A datagram sent as link-layer broadcast (e.g., Ethernet)
- An IP fragment other than the first
- A datagram whose source address does not define a single host (e.g., 0.0.0.0)

Reason is the risk of creating:

- Loops
- Packet explosions (broadcast storms)