IK2204

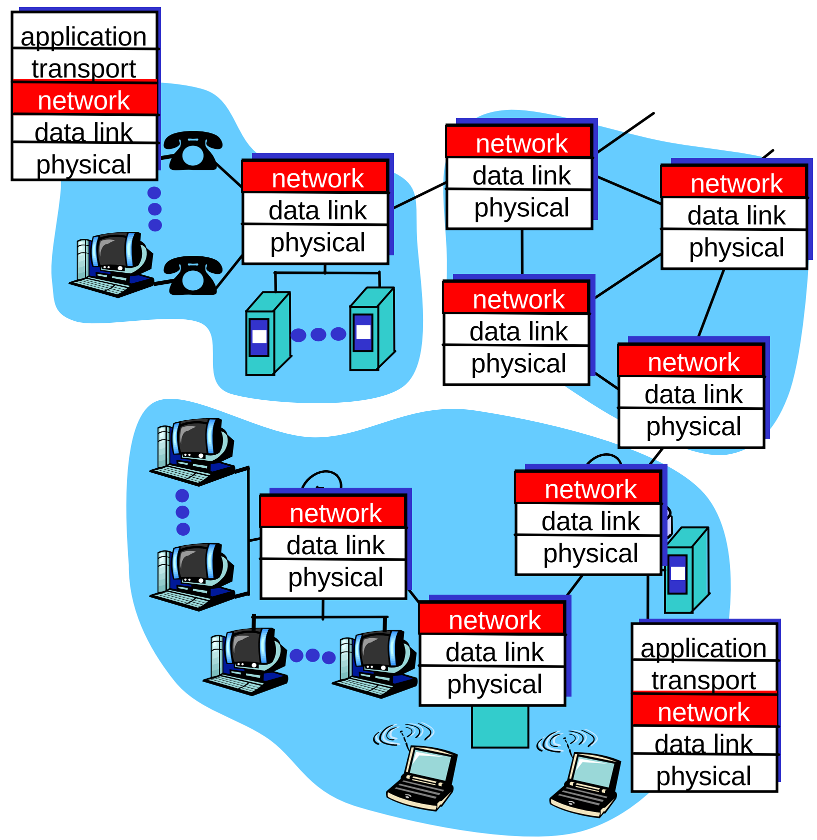
Network Layer

Contents

- Perlman, chapter 6, 8, 9, and 10
- Video lectures no. 12-20
- 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

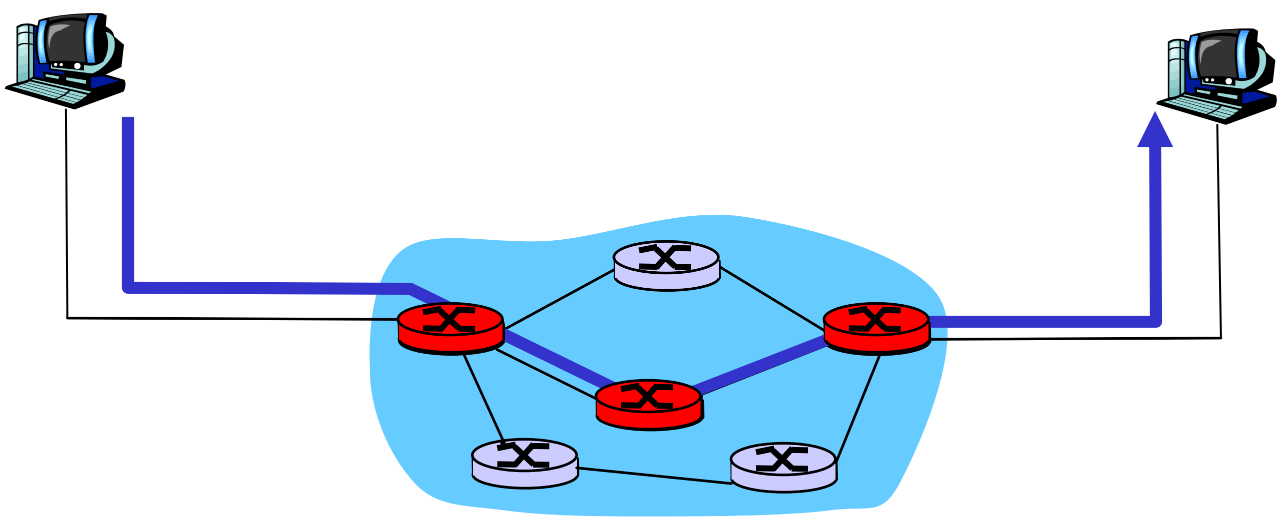
- Connection-Oriented Services
  - The network layer establishes a connection between a source and a destination
  - Packets are sent along the connection.
  - The decision about the route is made *once* at connection establishment
  - Routers/switches in connection-oriented networks are stateful
- Connectionless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

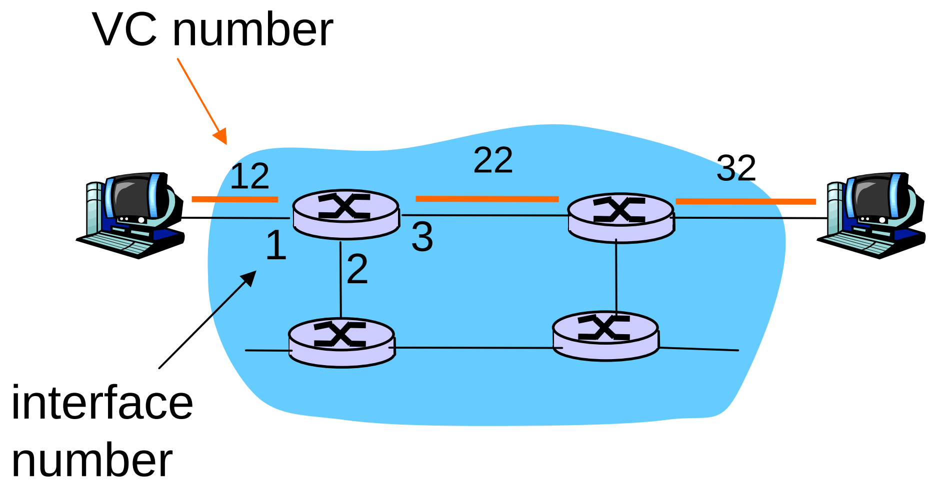
- Some network architectures use network layer *connections*
  - ATM, frame relay, X.25
  - *Virtual circuits*
- Before datagrams can flow, end-hosts and intervening routers establish a virtual circuit
- Network vs transpor layer connection service:
  - Network: between hosts through routers
  - Transport: 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



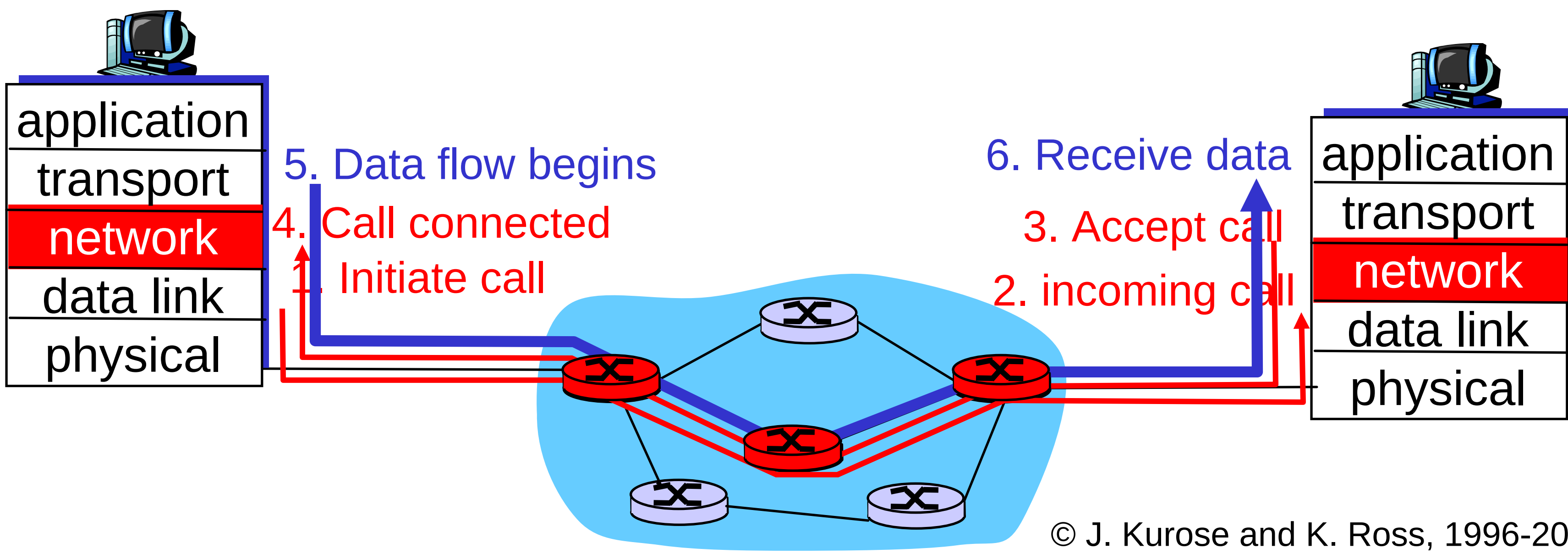
Forwarding table in
northwest router:

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!

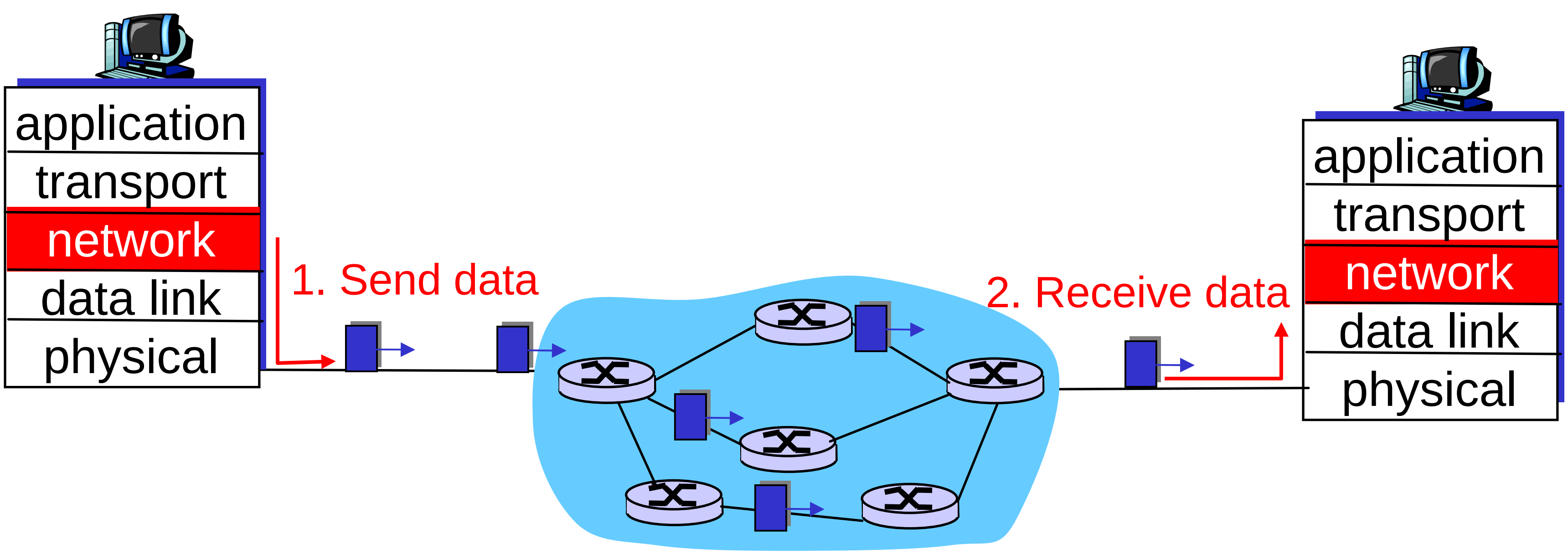
Virtual Circuits: Signalling Protocols

- Used to setup, maintain teardown VC
- Used in 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

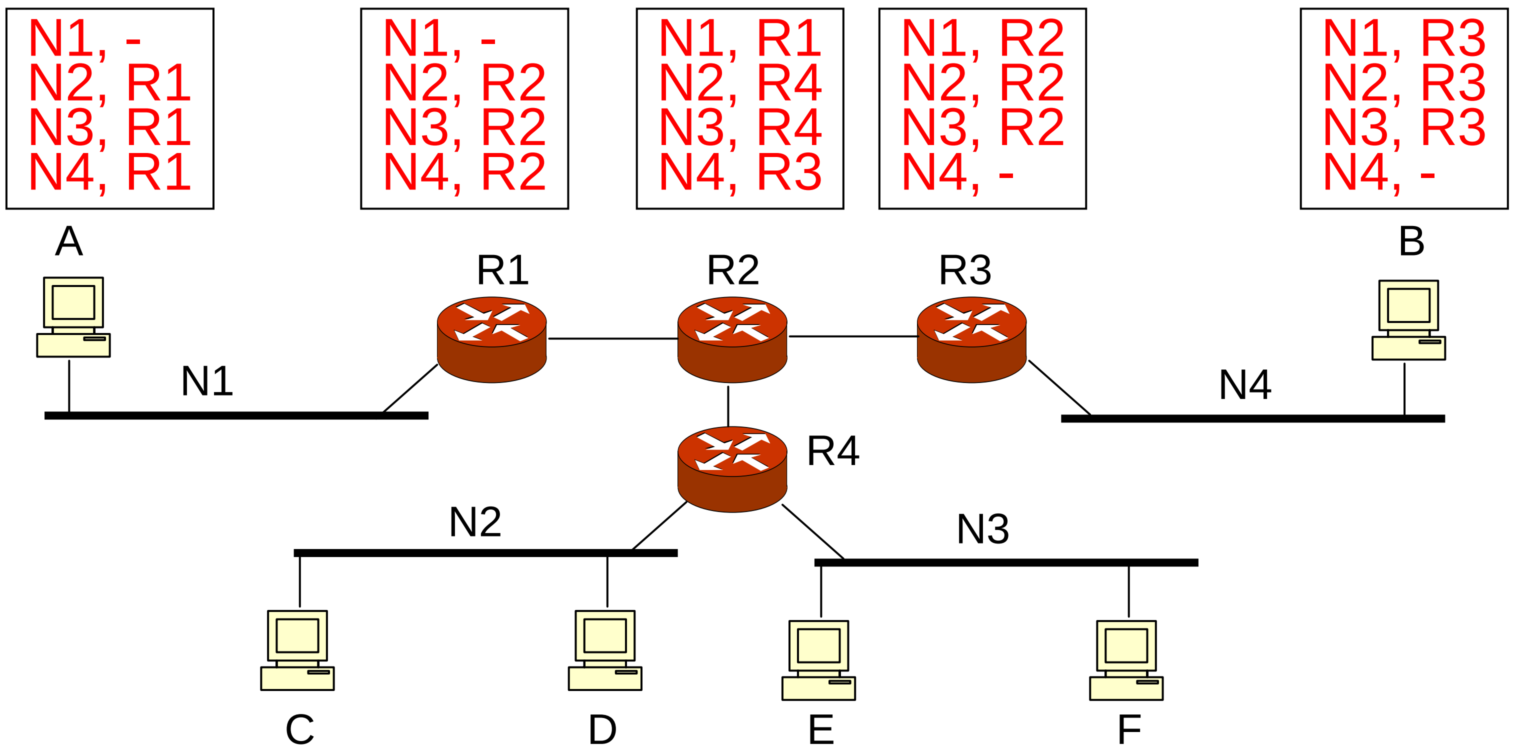


Issues in IP

- 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) is ~160000 (~60% 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

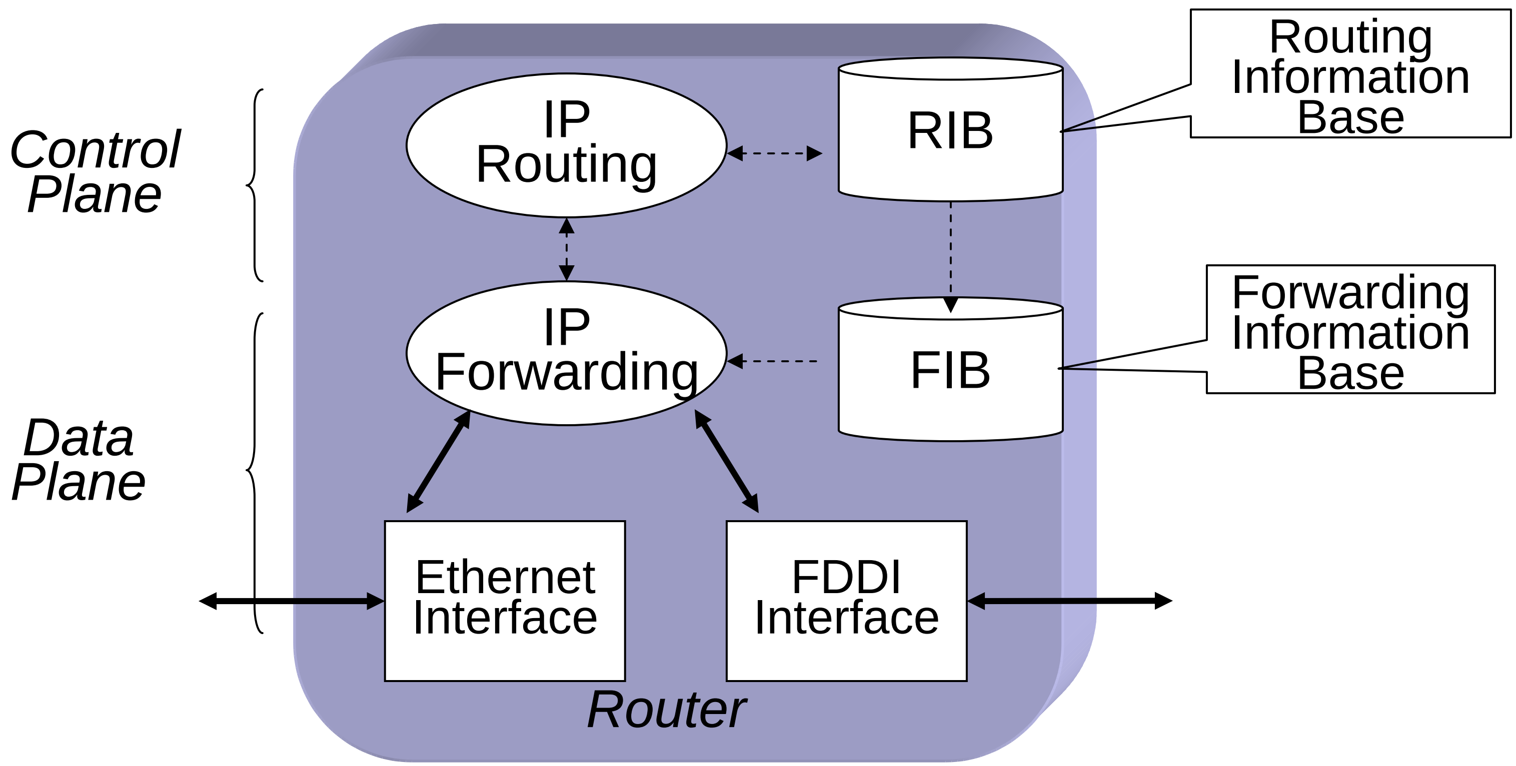
- Search for the most specific entry that matches the address

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
129.12.4.172/26
129.12.4.43/32

Best Matching Prefix

IP Router Model



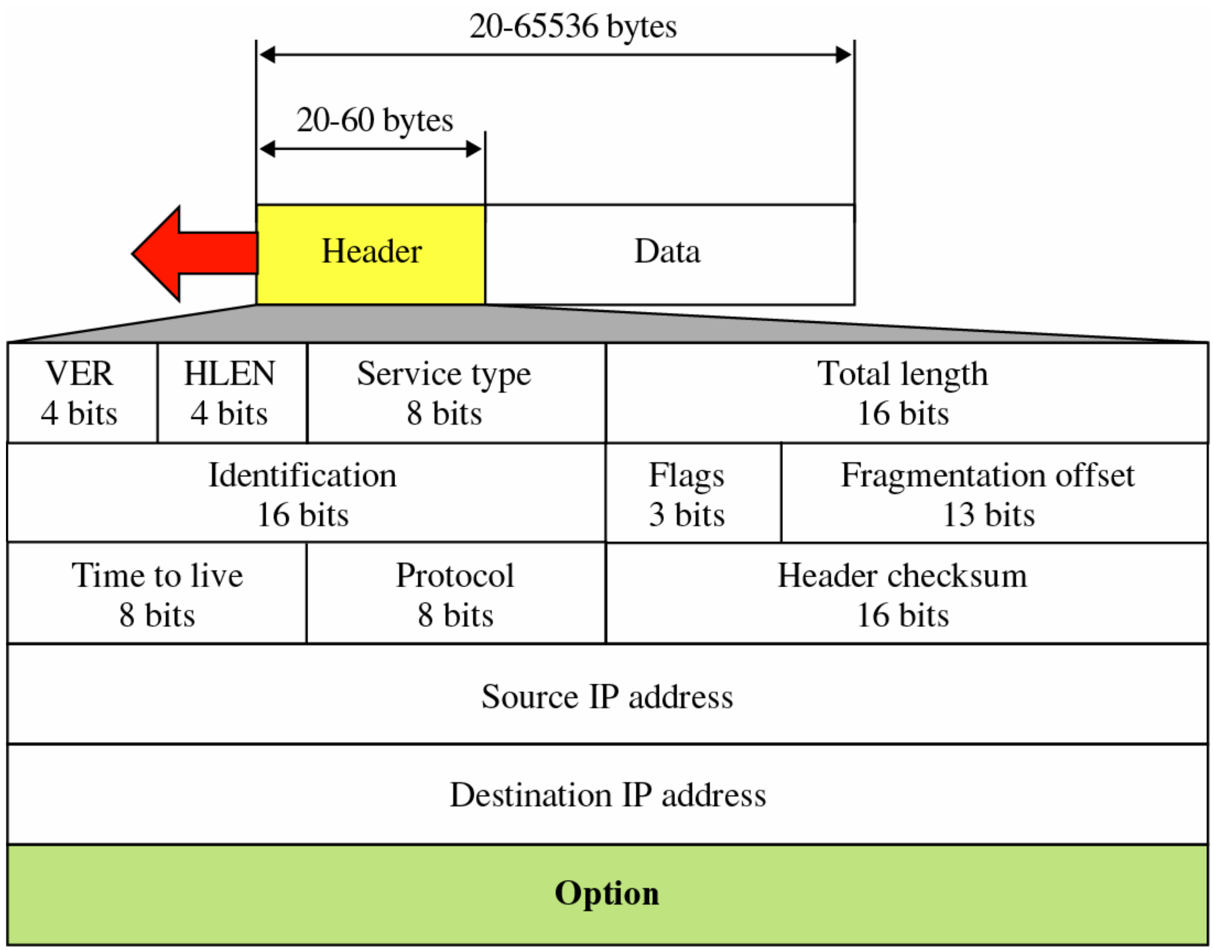
- A Router can be partitioned into a dataplane and a controlplane
  - The dataplane is fast and special purpose – handles packet *forwarding* in real-time
  - The control plane is general purpose– handles *routing* in the background

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)

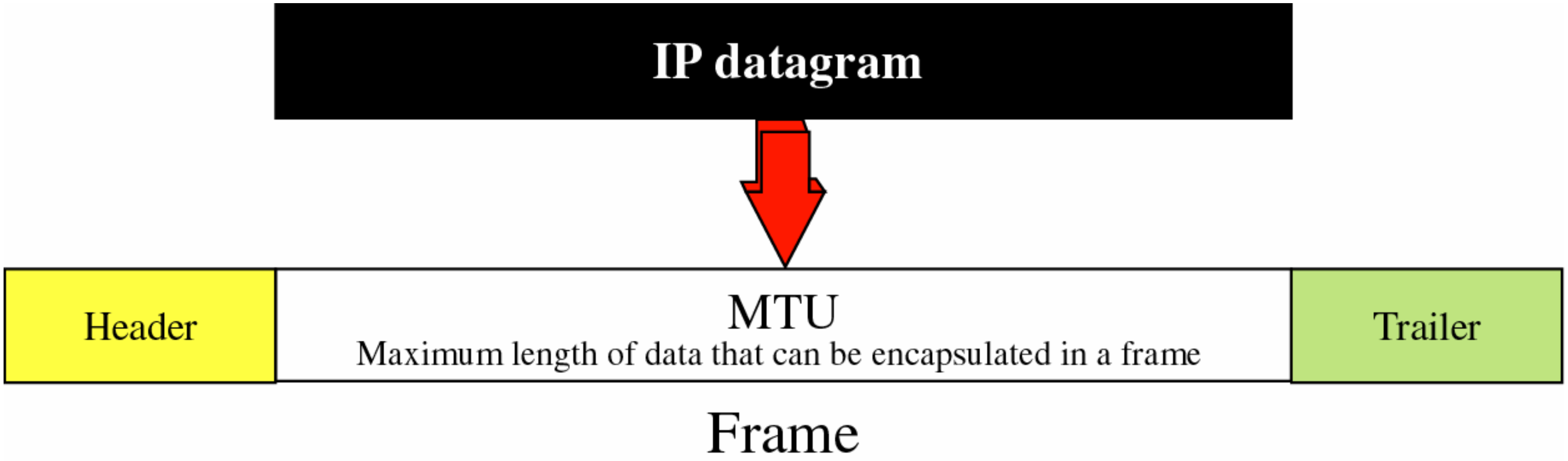
- Version
- HLEN – Header Length
- Type of Service
- Total Length
  - Header + Payload
- Fragmentation
  - ID, Flags, Offset
- TTL – Time To Live
  - Limits lifetime
- Protocol
  - Higher level protocol
- Header checksum
- IP Addresses
  - Source, Destination
- Options



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*

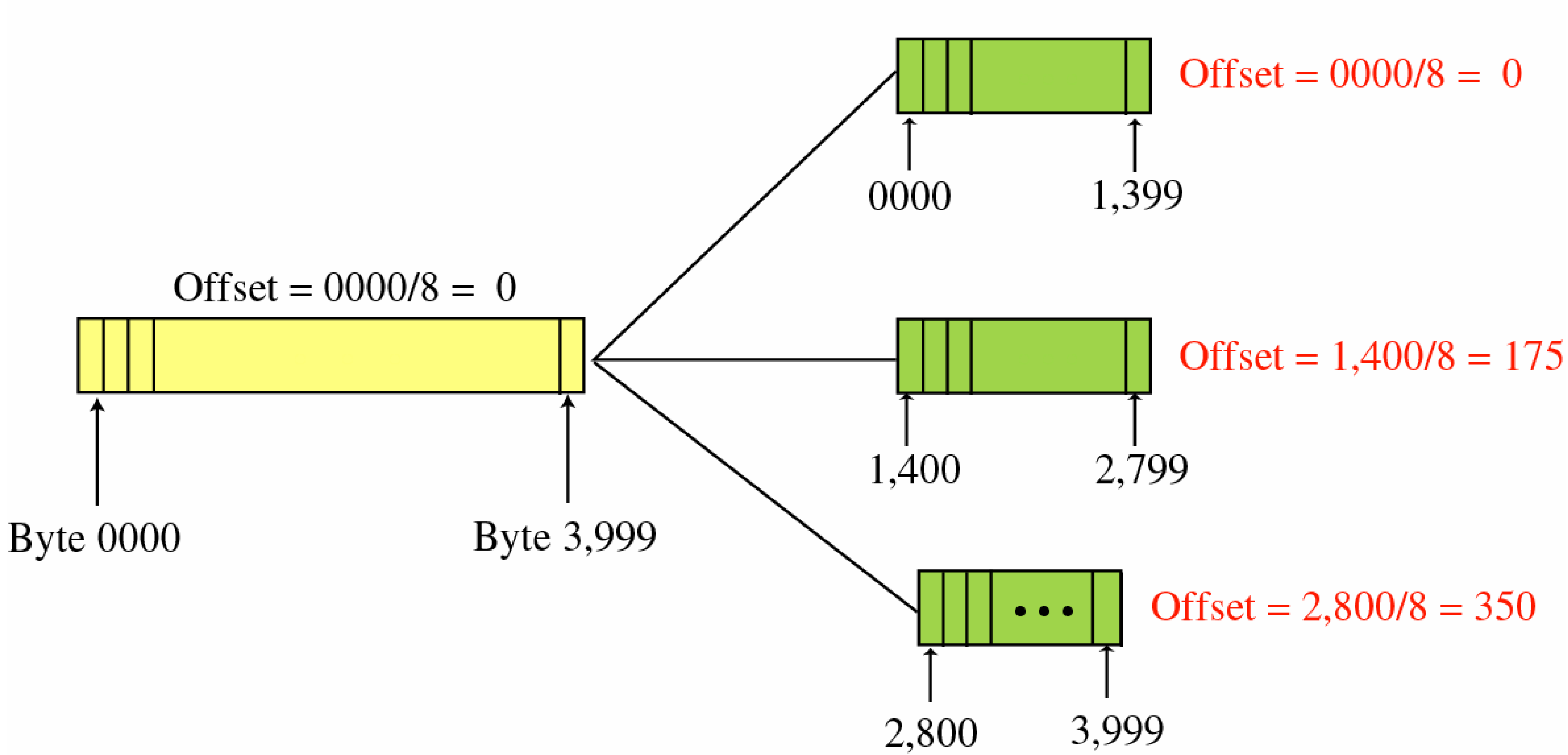
Fragmentation, cont'd

- 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 uses 576 byte MTU or path MTU discovery
- 3 fields of the IP header concerns fragmentation

The Fragmentation Fields

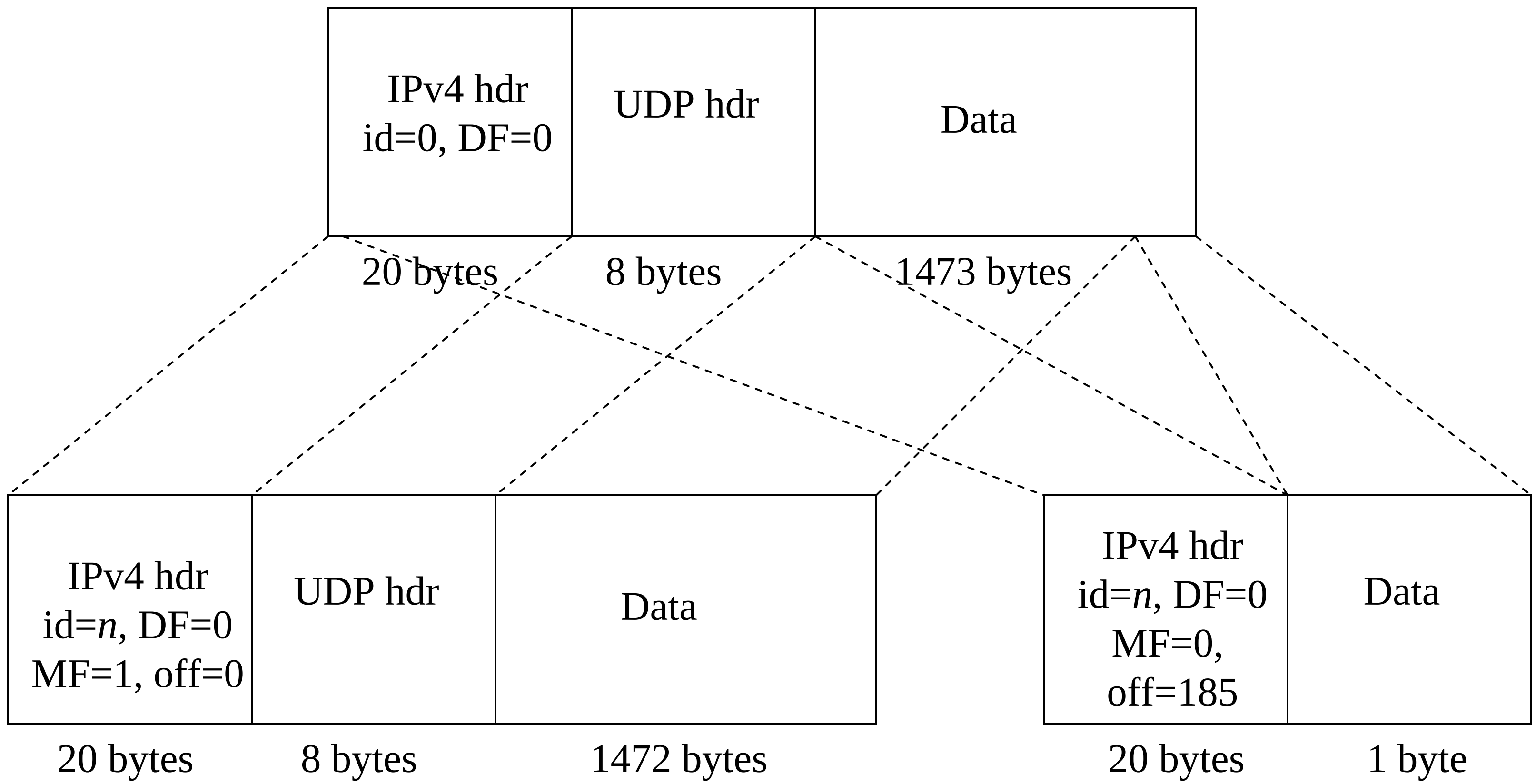
- Identification: 16 bits
  - ID + src IP addr uniquely identifies each datagram sent by a host
  - The ID is copied to all fragments of a datagram upon fragmentation
- Flags: 3 bits
  - RF (Reserved Fragment) – for future use (set to 0)
  - DF (Dont Fragment).
    - Set to 1 if datagram should not be 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

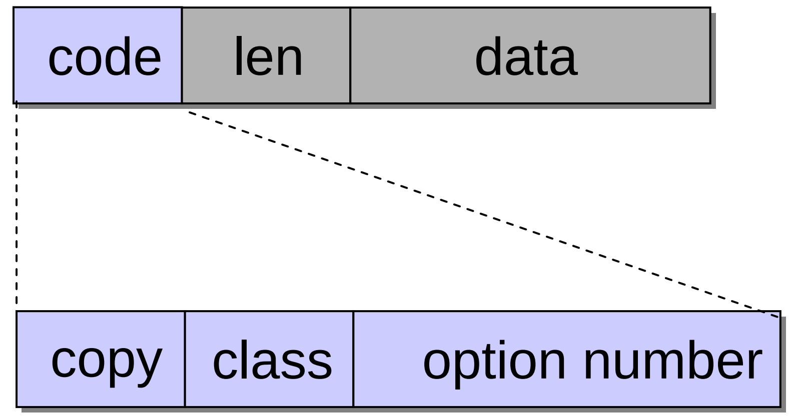
- Ensures integrity of header fields
  - Hop-by-hop (not end-to-end)
  - The header fields must be correct for proper and safe processing.
  - The payload is not covered.
- Other checksums
  - Link-level CRC. IP assumes a strong L2 checksum/CRC. Hop-by-hop.
  - L4 checksums, eg TCP/ICMP/UDP checksums 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

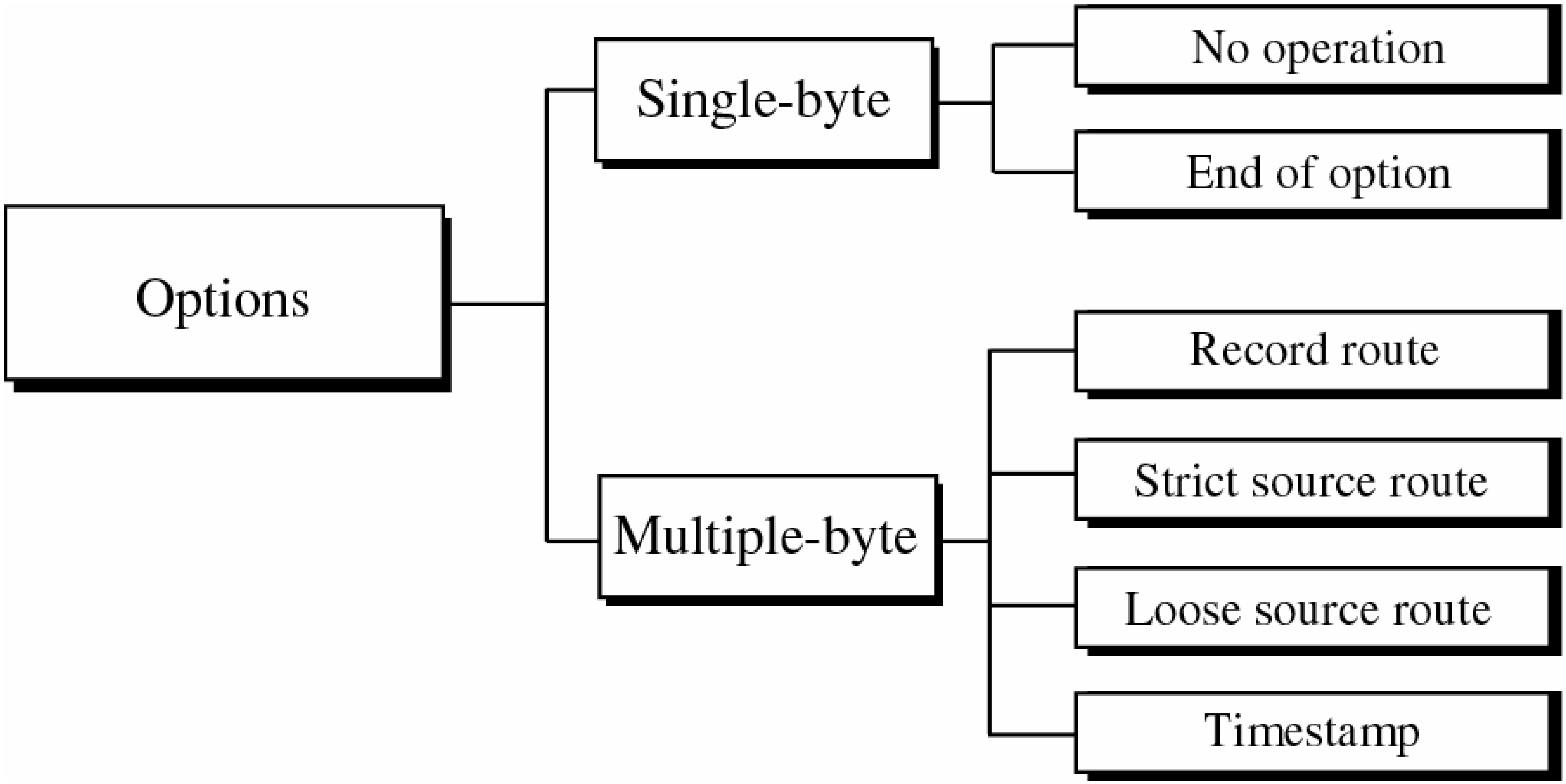
- IPv4 options are intended for network testing or debugging
- Options are variable size and comes after the fixed header.
- Contiguous – no separators
- Not required fields, but all IP implementations must include processing of options
  - In practice many implementations do not!
- Max 40 bytes - very 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



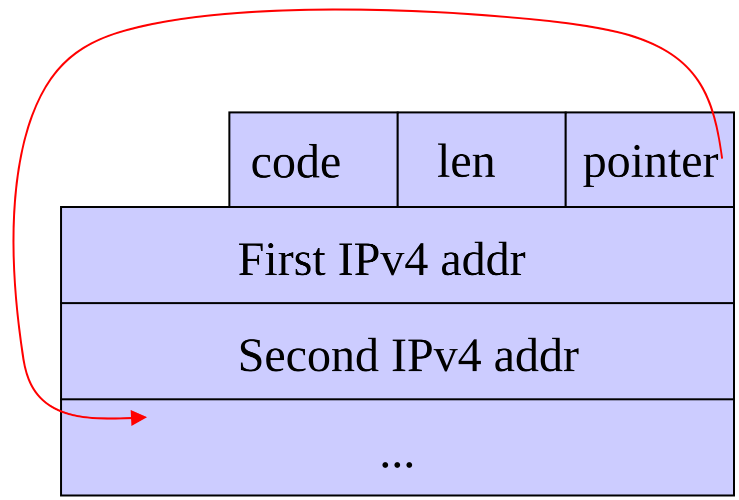
IP Option Types

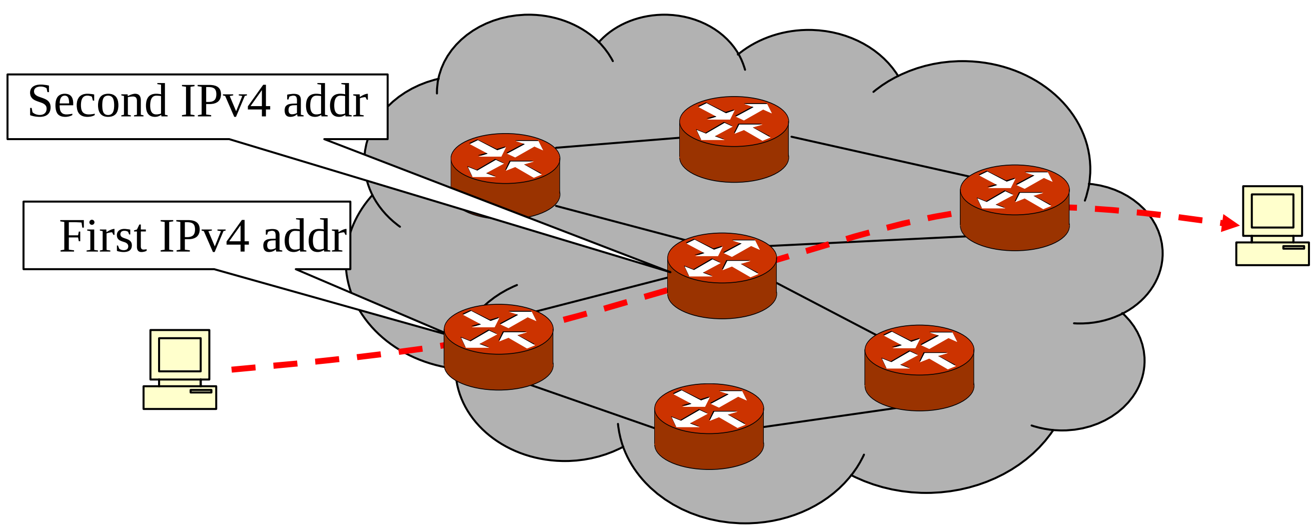


Timestamp: record route *and* timestamp

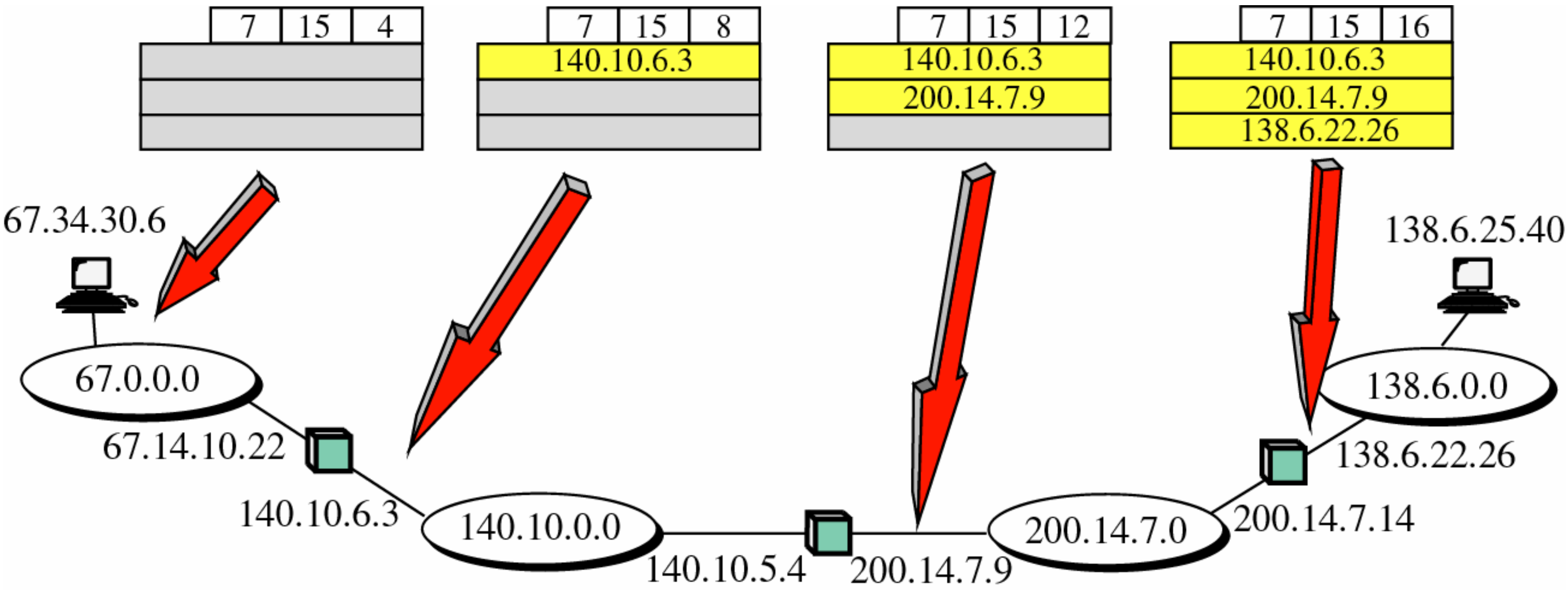
IP Option Example: Record Route

- 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



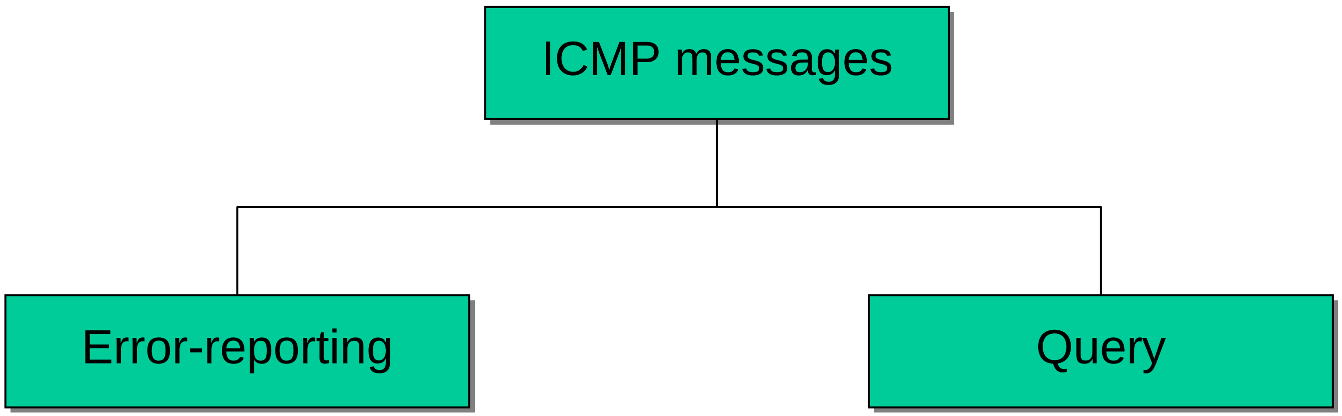


IP Options: Record Route Example



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

ICMP



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)

IPv6

- Changes since IPv4 was developed (mid 70's)
  - Provider market has changed dramatically
  - Immense increase in user and traffic on the Internet
  - Rapid technology advancement
  - Bandwidth increase from kb/s to Tb/s
- IPv4 issues
  - Too few addresses (though only 3-7% of address space used)
  - Too large routing tables
- To address these issuees IETF has standardized IPv6
  - IPv6 should keep most of the characteristics of IPv4 (good design)
  - Changing the address fields is the big thing with IPv6
  - While modifying the header, improvements have been introduced

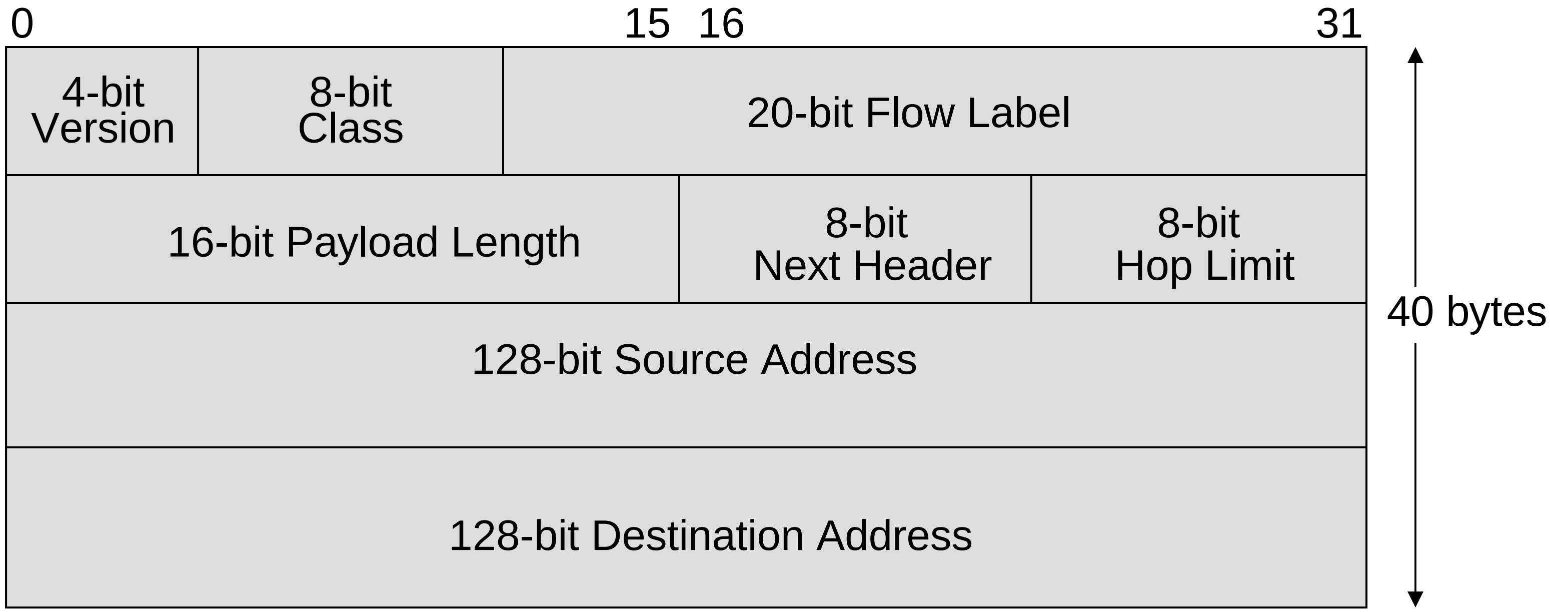
IPv6 vs IPv4

- Changes in IPv6 compared to IPv4
  - 128 bit addresses
  - extended address hierarchy
  - simplified header
  - simpler and better support for options
  - possible to extend the protocol
  - support for autoconfiguration (plug-and-play)
  - support for QoS treatment
  - host mobility
  - security
  - provider selection
  - no fragmentation in routers

IPv6 Simplifications

- Fixed format headers
  - Use extension headers instead of options
- Remove header checksum
  - Rely on link layer and higher layers to check integrity of data
- Remove hop-by-hop segmentation
  - Fragmentation only by sender due to path MTU discovery

IPv6 Header



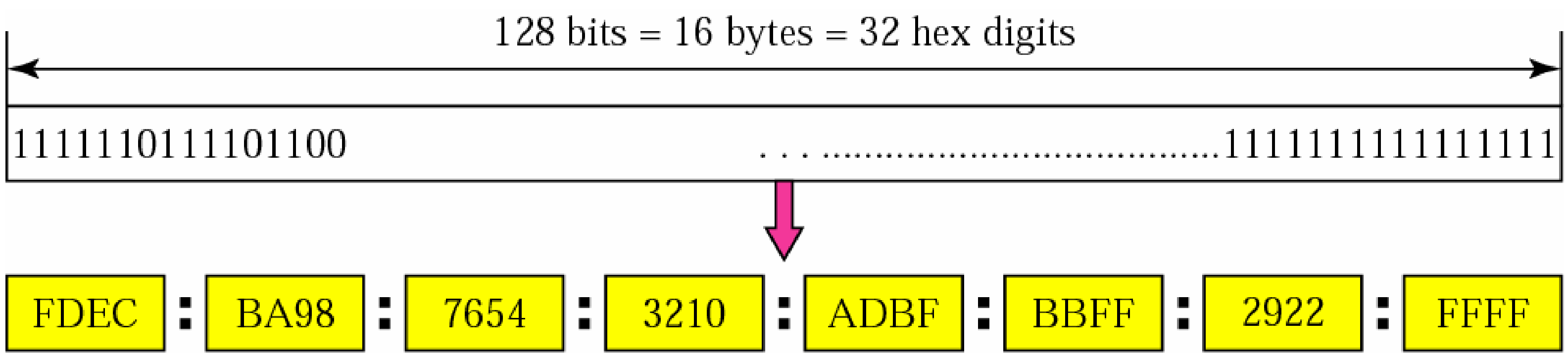
- **Version** Only field identical to IPv4. Code is 6 in IPv6
- **Class** New field. Revised concept of priority bits. Facilitates handling of real-time traffic.
- **Flow Label** New field. To distinguish packets requiring the same treatment.
- **Payload Length** Replaces *length* field in IPv4. Gives length of data following IPv6 header
- **Next Header** Replaces *protocol* field in IPv4. Extension headers can be used.
- **Hop Limit** Replaces *TTL* field in IPv4. Hop limit more accurately reflects the use of TTL.
- **Src Address** Revised *source address* field. 128 bits in IPv6 vs 32 bits in IPv4.
- **Dst Address** Revised *destination address* field. 128 bits in IPv6 vs 32 bits in IPv4.

IPv6 Addresses

- **An IPv6 unicast address identifies an interface connected to an IP subnet (as is the case in IPv4)**
- **One big difference between IPv6 and IPv4 is that IPv6 routinely allows each interface to be identified by several addresses**
  - facilitates management
- **IPv6 has three address categories:**
  - unicast - identifies exactly one interface
  - multicast - identifies a group; packets get delivered to all members of the group
  - anycast - identifies a group; packets normally get delivered to nearest member of the group
- **128 bits results in 2^{128} addresses**
  - Distributed over the Earth: 665,570,793,348,866,943,898,599/m^2
  - Pessimistic estimate with hierarchies: ~1,564 addresses/m^2

IPv6 Address Format

- Colon hexadecimal notation (eight 16 bit hexadecimal integers)

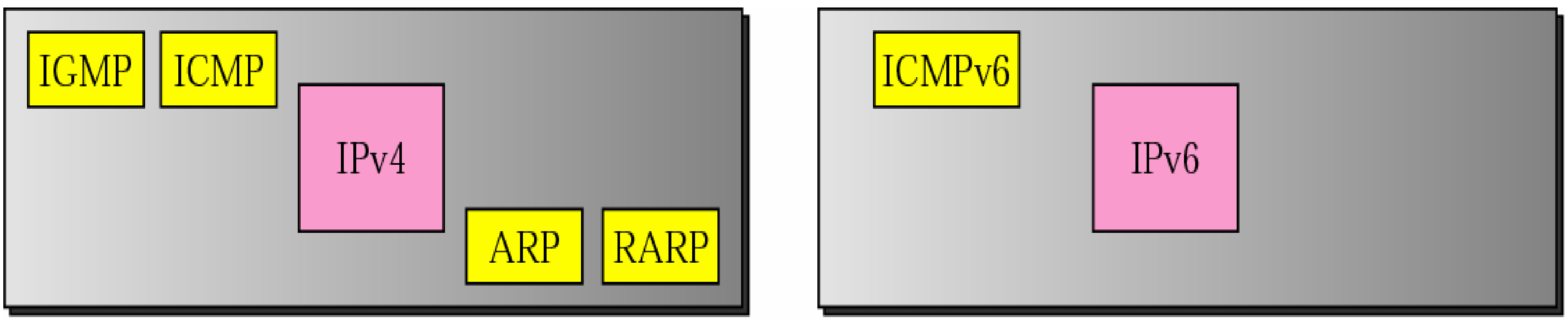


IPv6 Address Abbreviations and CIDR

- Leading zeros may be oppressed
  - FDEC:BA98:0074:3210:000F:BBFF:0000:FFFF
  - FDEC:BA98:74:3210:F:BBFF:0:FFFF
- Zero compression: one of a series of zeros may be replaced by ::
  - But only once
  - FDEC:0:0:0:0:BBFF:0:FFFF
  - FDEC::BBFF:0:FFFF
- CIDR notation to specify the first N bits of an address
  - FDEC:0:0:0:0:BBFF:0:FFFF/60

Network Layer Comparison—IPv4 vs IPv6

- ICMPv4 has been modified to be more suitable for IPv6, and thus updated to ICMPv6
- ARP and IGMP in version 4 are now part of ICMPv6
- RARP has been dropped due to limited use (DHCP does the job of RARP)
- As in ICMPv4, ICMPv6 messages are divided into 2 categories:
- Error-reporting (somewhat different messages in v6 vs v4)
- Query (rather different messages in v6 vs v4)



Network layer in version 4 — Network layer in version 6

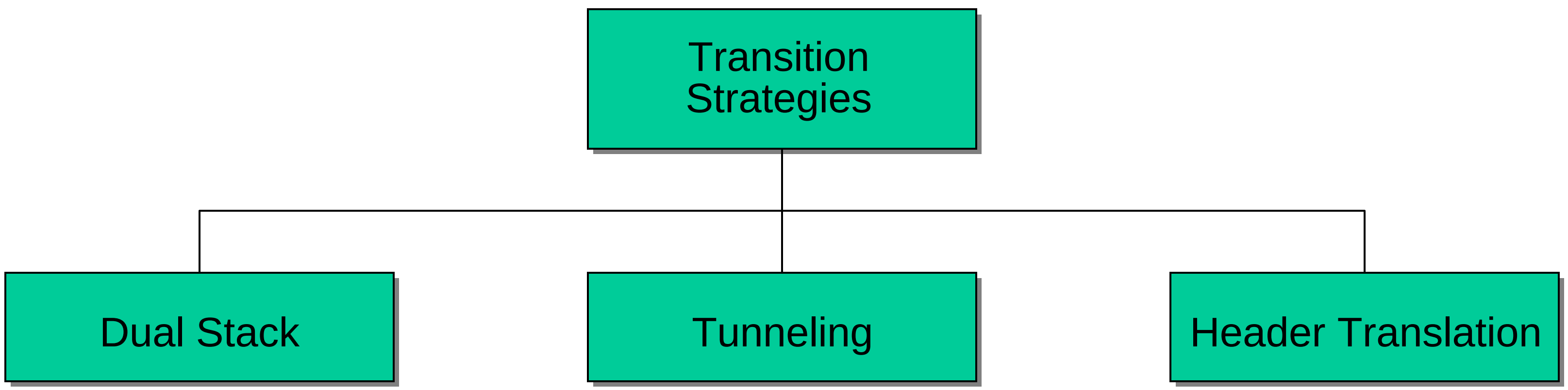
ICMPv4 vs ICMPv6

Error Report Message – Type	Ver 4	Ver 6
Destination unreachable	Yes	Yes
Source quench	Yes	No
Packet too big	No	Yes
Time exceeded	Yes	Yes
Parameter problem	Yes	Yes
Redirection	Yes	Yes

Query Message – Type	Ver 4	Ver 6
Echo request and reply	Yes	Yes
Timestamp request and reply	Yes	No
Address mask request and reply	Yes	No
Router solicitation and advertisement	Yes	Yes
Neighbour solicitation and advertisement	ARP	Yes
Group membership	IGMP	Yes

Transition from IPv4 to IPv6

- Because of the large number of systems on the Internet, the transition from IPv4 to IPv6 cannot happen suddenly
- Transition should be smooth to prevent problems
- Transition strategies have been devised by IETF



IPv6 Summary

- **IPv6 has:**
  - 128-bit address space
  - revised header format
  - new options
  - allowance for extension
  - support for special handling of packet flows
  - increased security measures
- **IPv6 uses hexadecimal colon notation with abbreviation methods**
- **IPv6 has three address types: unicast, anycast, and multicast**
- **IPv4, ICMPv4, ARP, RARP, and IGMP replaced with IPv6 and ICMPv6**
- **IPv4 to IPv6 transition strategies are dual-stack, tunneling, and header translation**

What’s Next

- Lab 2
  - Moved to rooms A44 and A45 in the Electrum 3 building
  - Based on Cisco Networking Academy material
- Video lectures
  - Make sure you go through them!
- Next lecture will be a sum-up
  - Chapter 11, 12, 13, 14