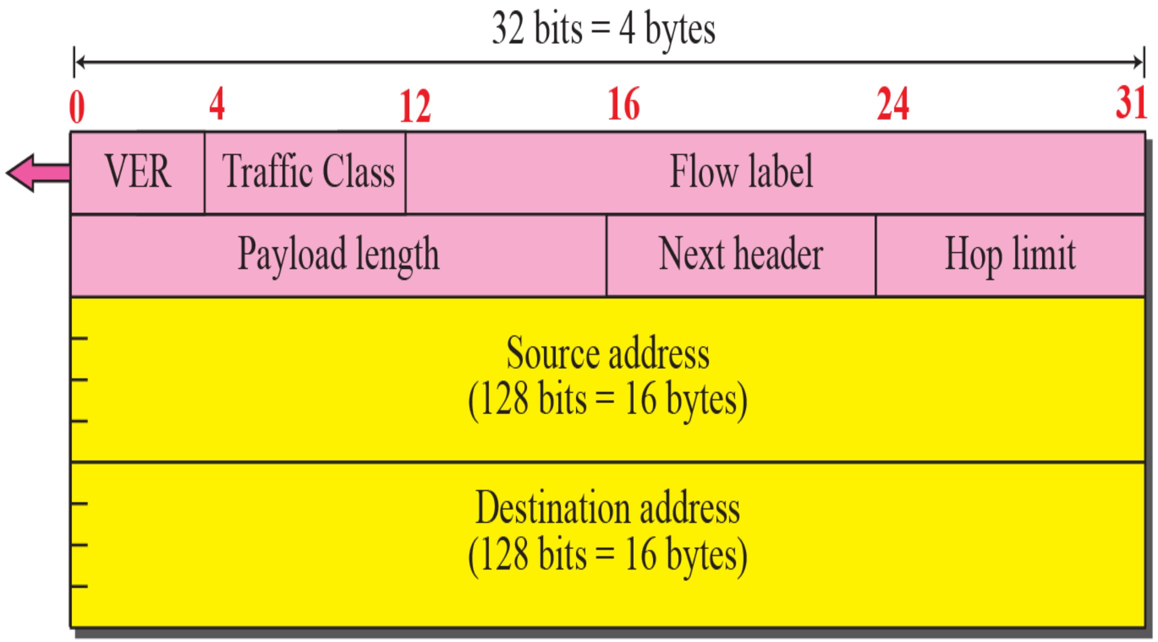
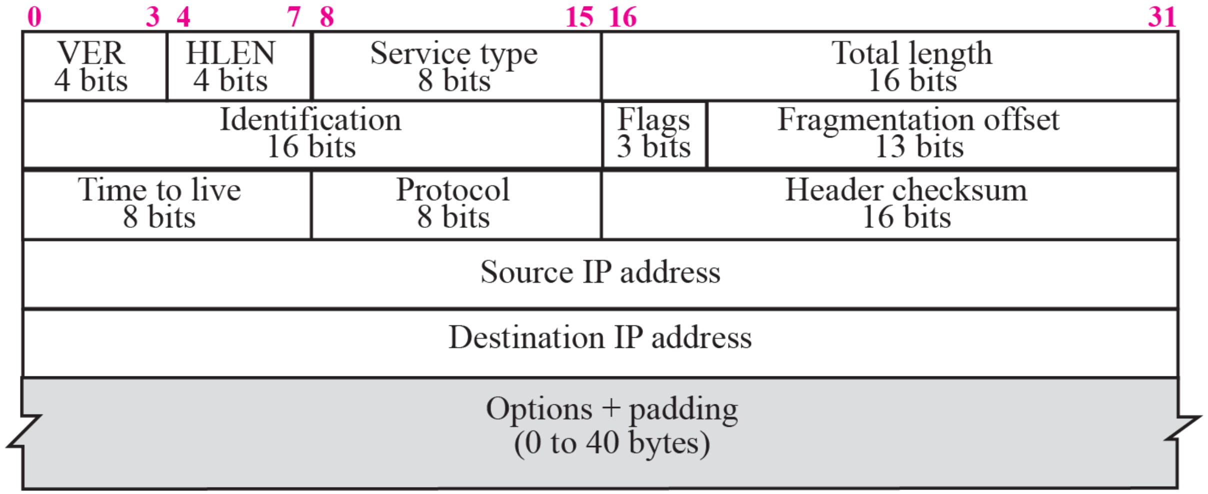
IPv6

Anand Baswade
anand@iitbhilai.ac.in

Format of the base header



IPv6



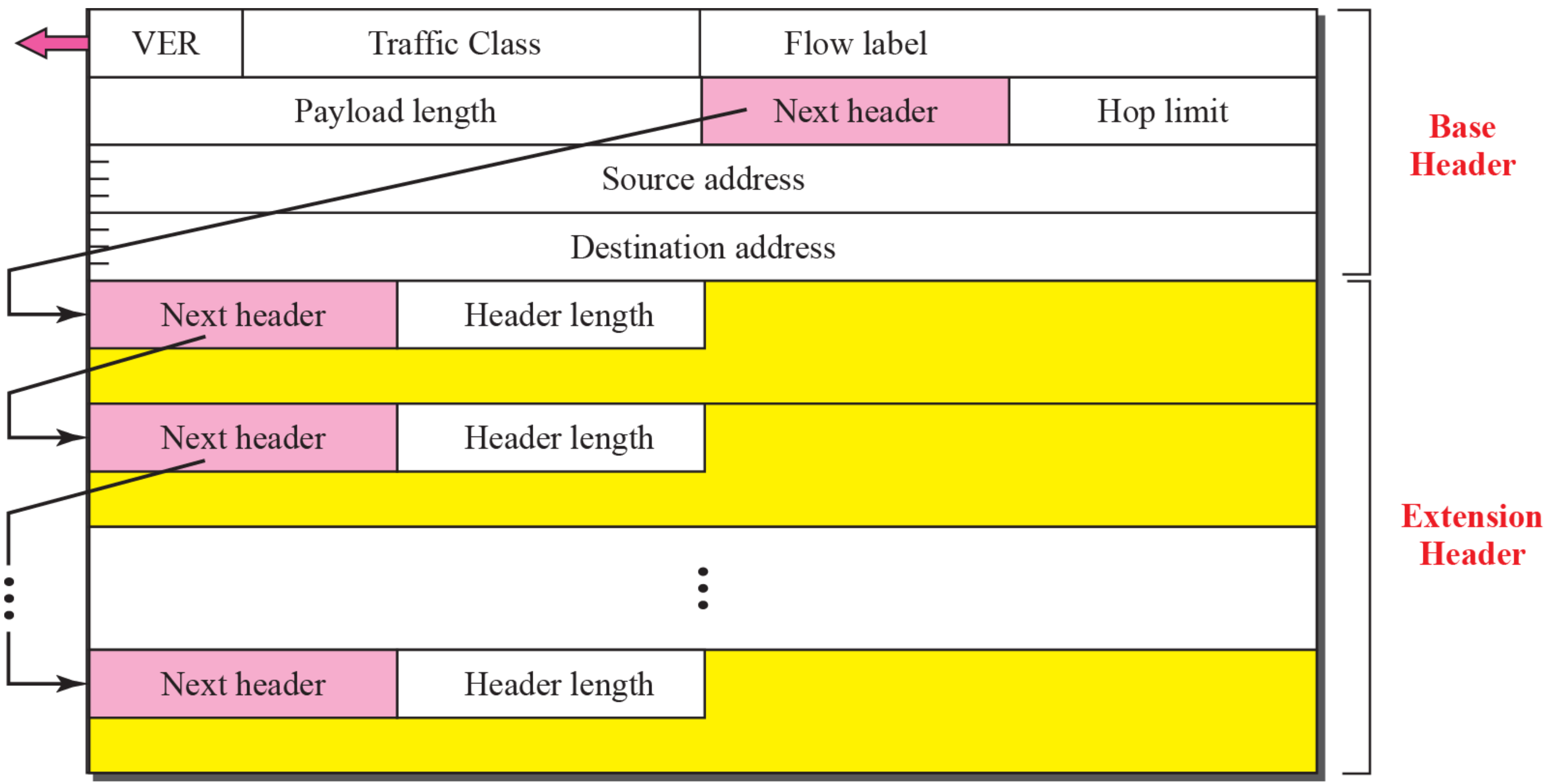
b. Header format

IPv4

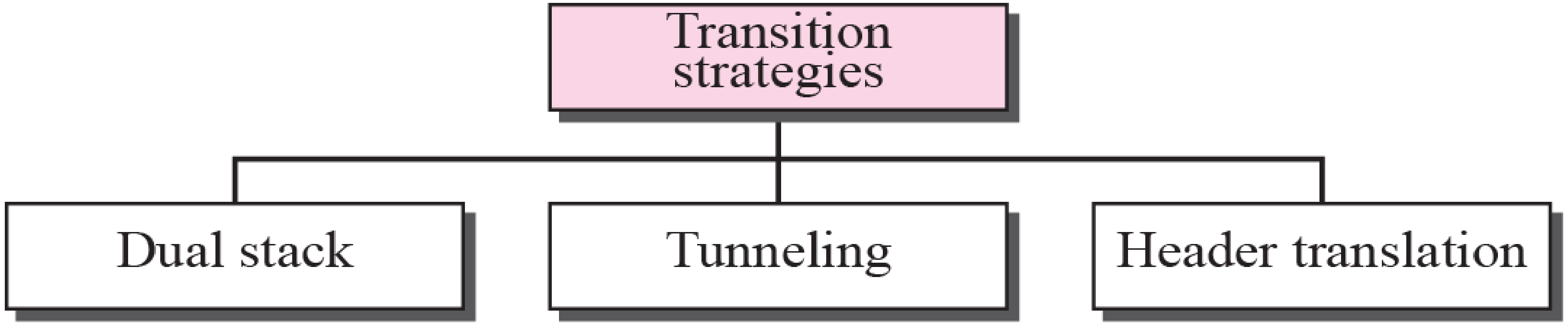
Table 27.1 *Next Header Codes*

Code	*Next Header*	*Code*	*Next Header*
0	Hop-by-hop option	44	Fragmentation
2	ICMP	50	Encrypted security payload
6	TCP	51	Authentication
17	UDP	59	Null (No next header)
43	Source routing	60	Destination option

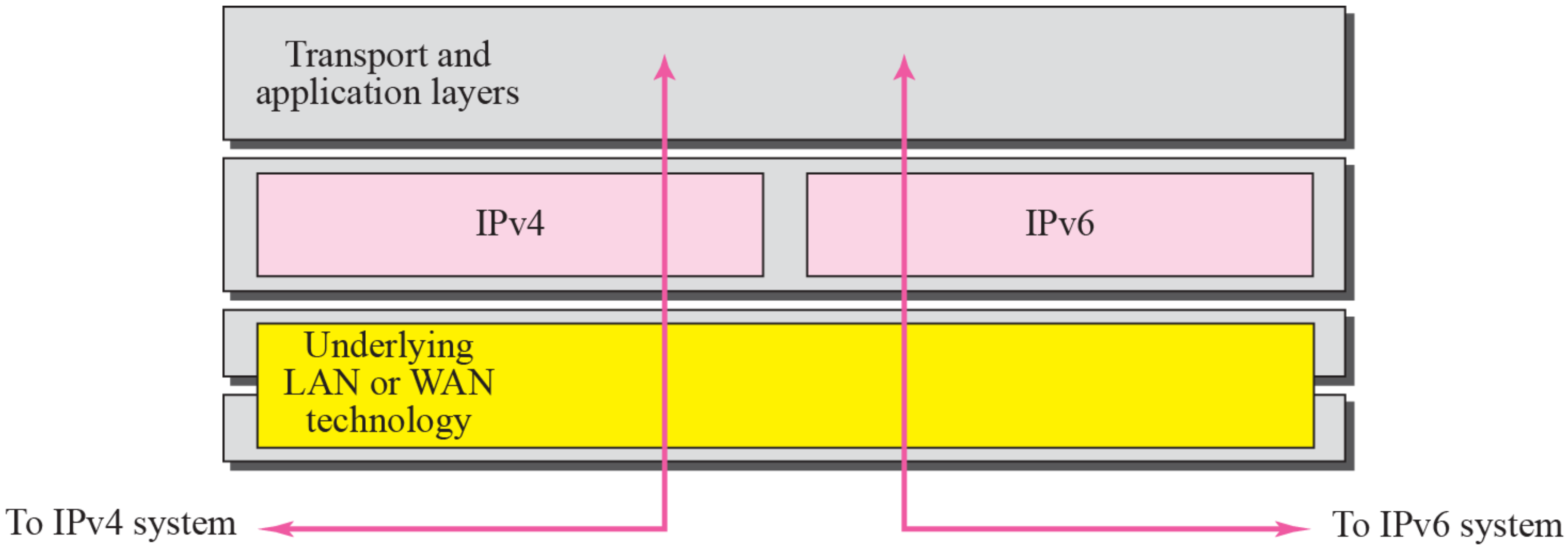
Extension header format



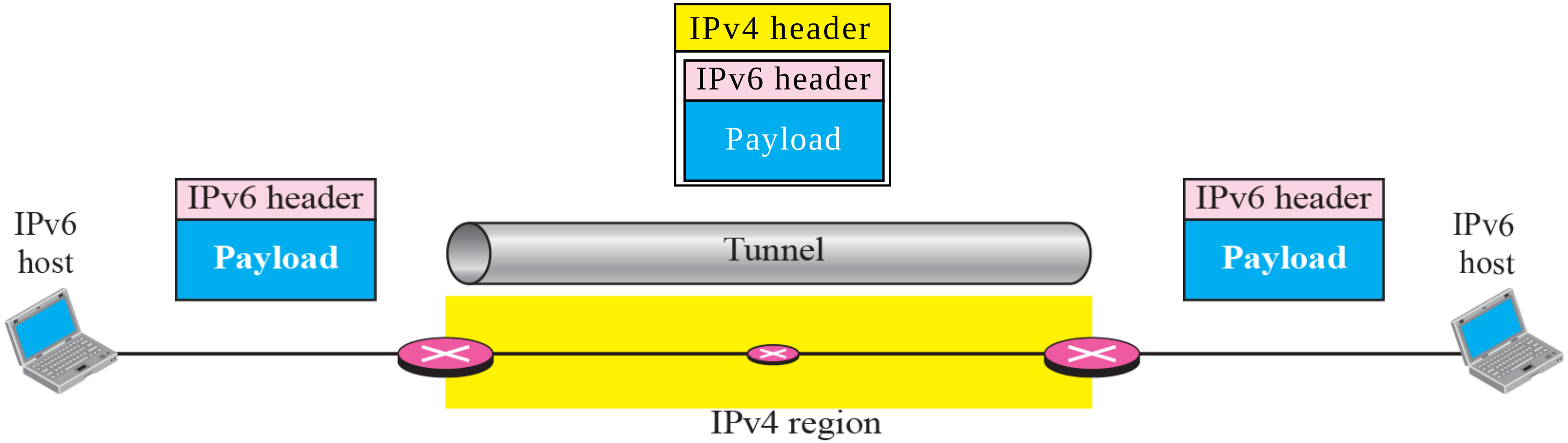
Three transition strategies



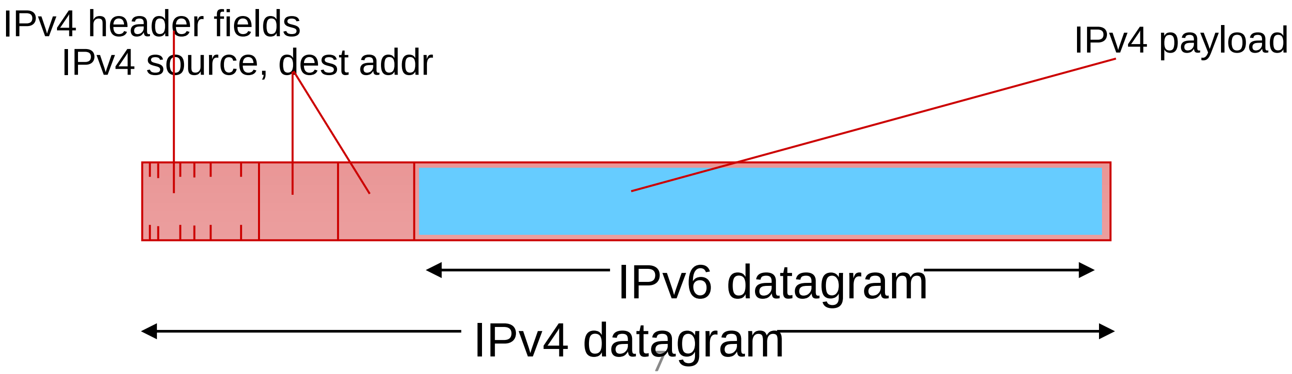
Dual stack



Device has both the stacks so can handle both ipv4 and ipv6 packets.

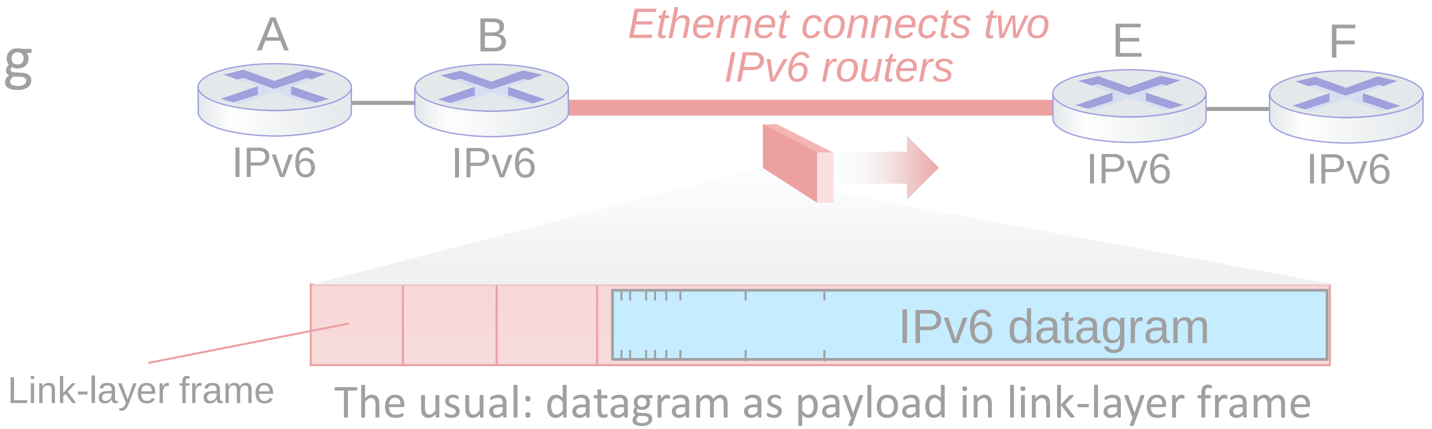


- tunneling: IPv6 datagram carried as *payload* in IPv4 datagram among IPv4 routers ("packet within a packet")
  - tunneling used extensively in other contexts (4G/5G)



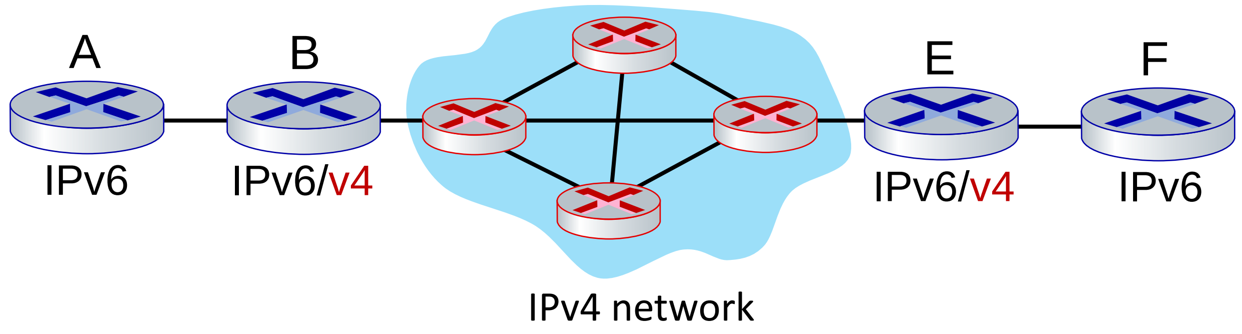
Tunneling and encapsulation

Ethernet connecting two IPv6 routers:



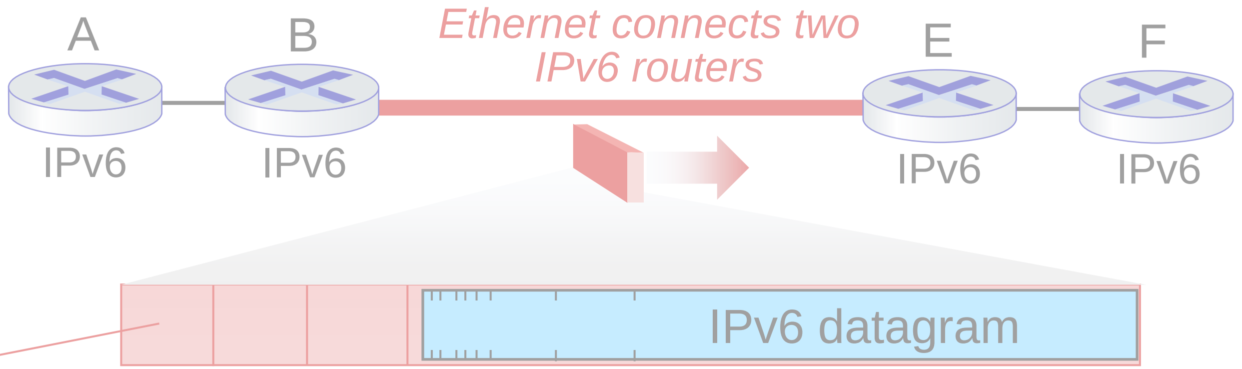
The usual: datagram as payload in link-layer frame

IPv4 network connecting two IPv6 routers



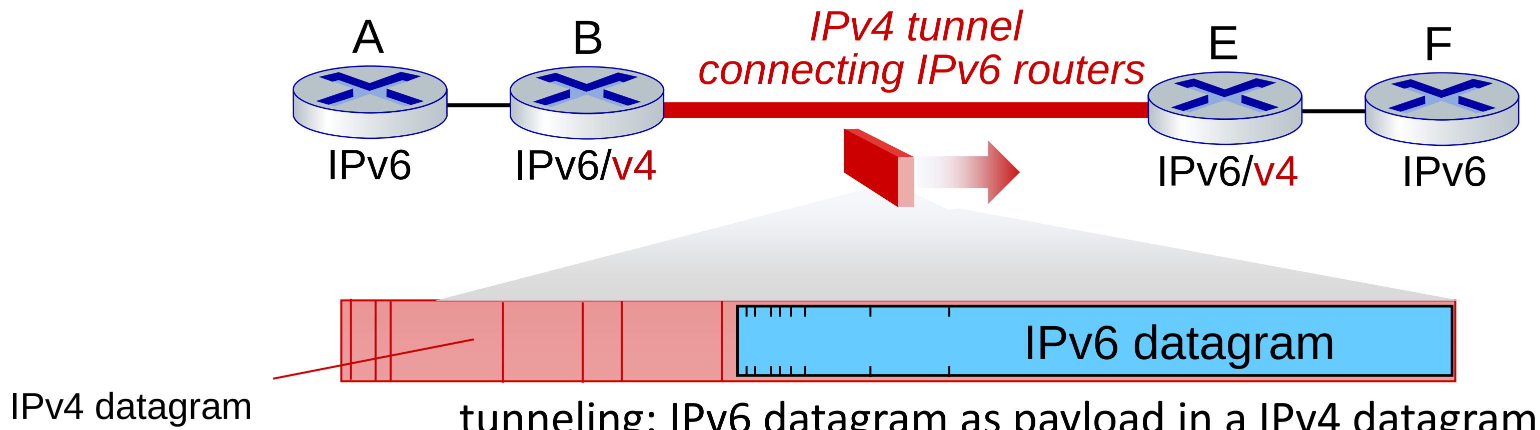
Tunneling and encapsulation

Ethernet connecting two IPv6 routers:



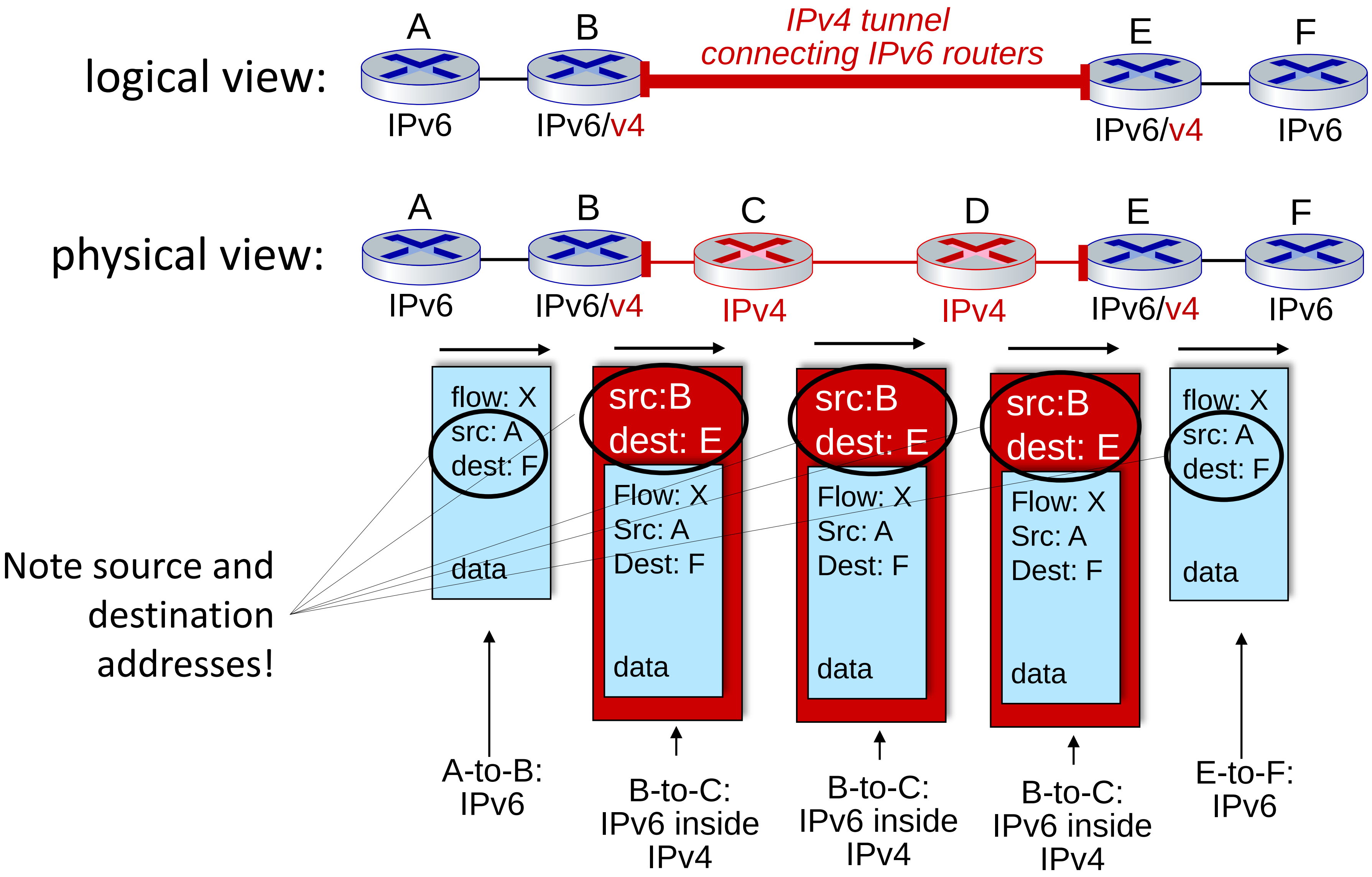
The usual: datagram as payload in link-layer frame

IPv4 tunnel connecting two IPv6 routers

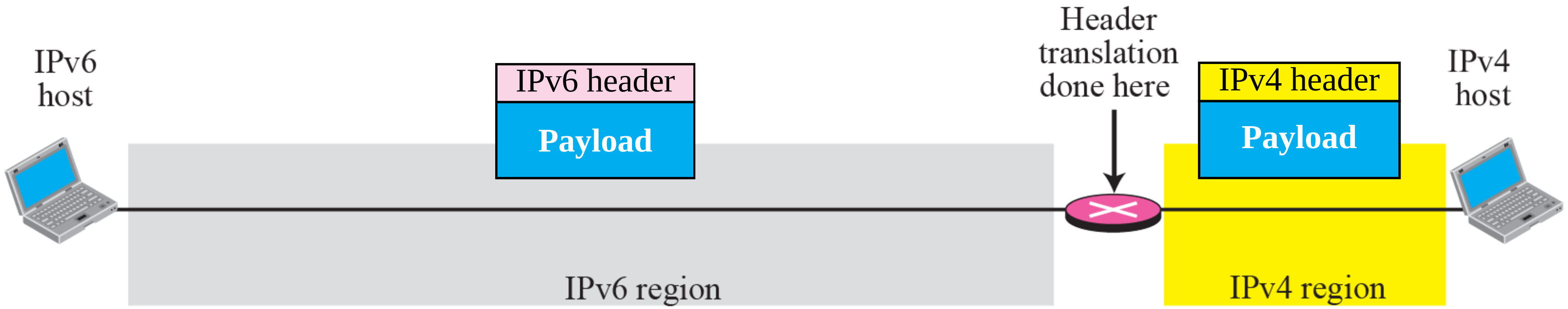


tunneling: IPv6 datagram as payload in a IPv4 datagram

Tunneling



logical view:
A
B
IPv4 tunnel
connecting IPv6 routers
E
F
IPv6
IPv6/v4
IPv6/v4
IPv6
physical view:
A
B
C
D
E
F
IPv6
IPv6/v4
IPv4
IPv4
IPv6/v4
IPv6
flow: X
src: A
dest: F
data
src:B
dest: E
Flow: X
Src: A
Dest: F
data
src:B
dest: E
Flow: X
Src: A
Dest: F
data
src:B
dest: E
Flow: X
Src: A
Dest: F
data
flow: X
src: A
dest: F
data
Note source and destination addresses!
A-to-B: IPv6
B-to-C: IPv6 inside IPv4
B-to-C: IPv6 inside IPv4
B-to-C: IPv6 inside IPv4
E-to-F: IPv6

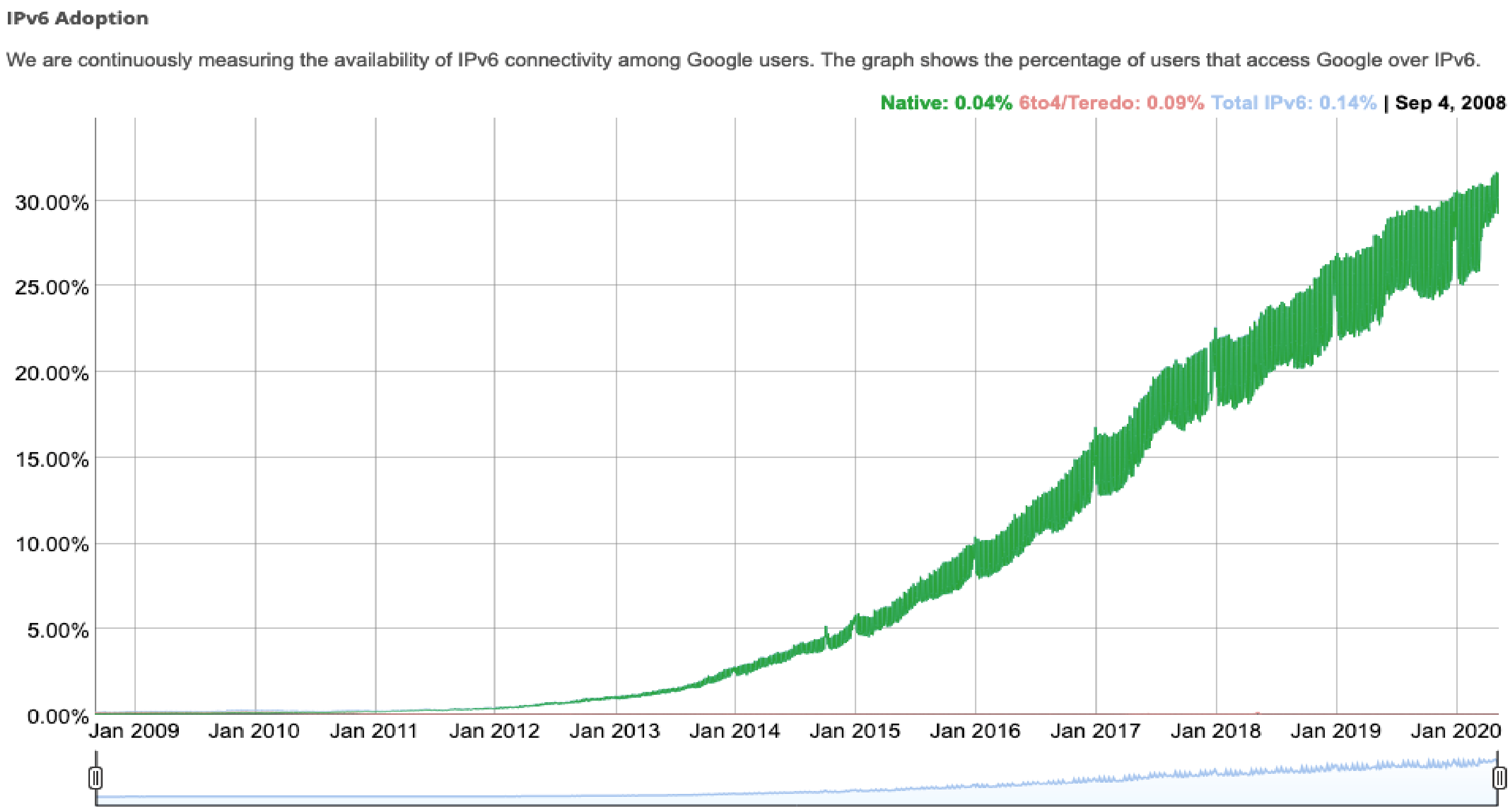


Network Address Translation - Protocol Translation (NAT-PT) [RFC 2766]

- With the help of the NAT Protocol Translation technique, the IPv4 and IPv6 networks can also communicate with each other which do not understand the address of different IP version.
- It removes sender's IP version header and adds receivers IP version header.

IPv6: adoption

- Google[1]: ~ 30% of clients access services via IPv6
- NIST: 1/3 of all US government domains are IPv6 capable
- India is leading in IPv6 expansion.



NIST: National Institute of Standards and Technology
Government agency

1 https://www.google.com/intl/en/ipv6/statistics.html

IPv6: adoption

- Google[1]: ~ 30% of clients access services via IPv6
- NIST: 1/3 of all US government domains are IPv6 capable
- Long (long!) time for deployment, use
  - 25 years and counting!
  - think of application-level changes in last 25 years: WWW, social media, streaming media, gaming, telepresence, …
  - *Why?*

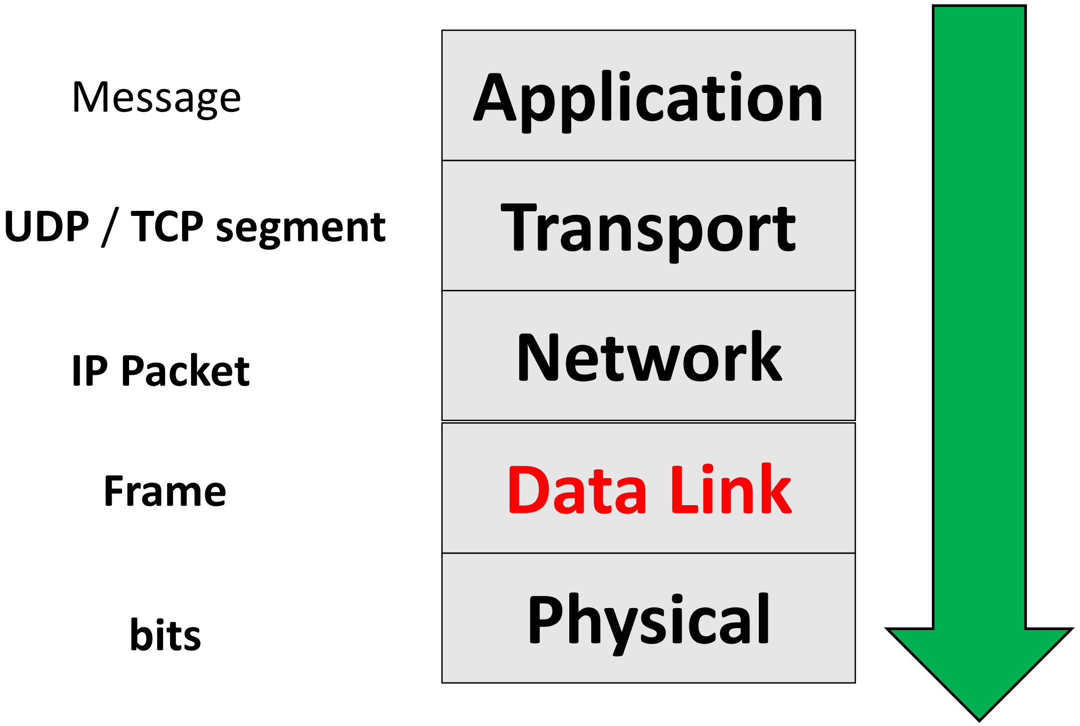
[1] https://www.google.com/intl/en/ipv6/statistics.html

Data Link Layer

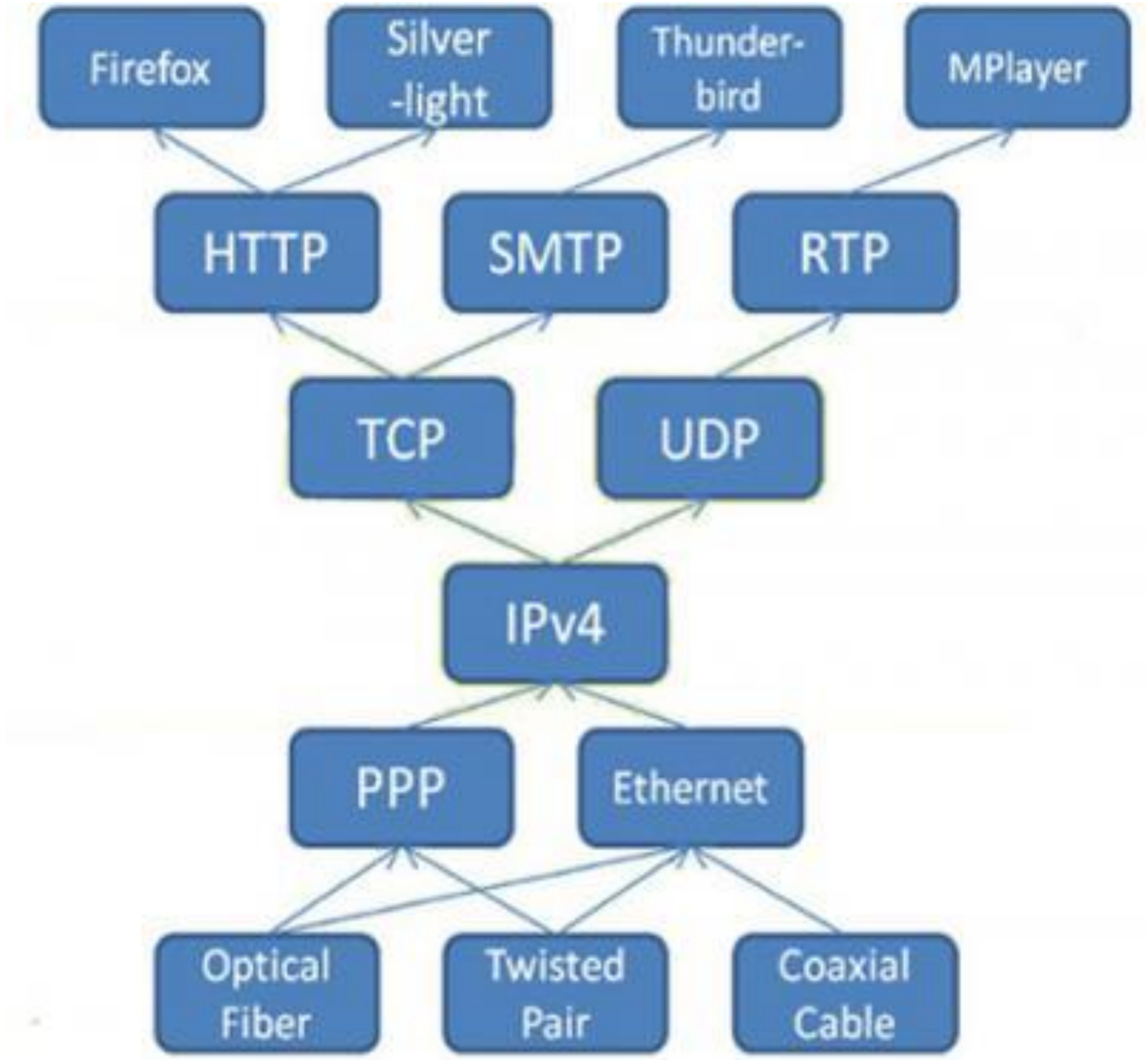
Outline

- Introduction to Data link Layer (Overview)
- Data Link Layer Services
  - Hop to Hop delivery
  - Flow control
  - Error detection and corrections
  - Multiple access protocols
- LANs
  - Addressing
  - Frame structure, Ethernet and Wi-Fi
  - Switches
  - VLANs
- Data Center networking
- Complete flow for a Web Request

Top Down Approach

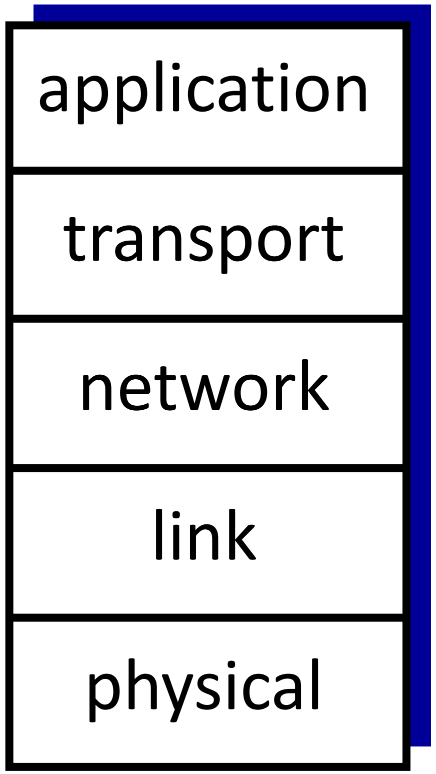


The Internet Hourglass

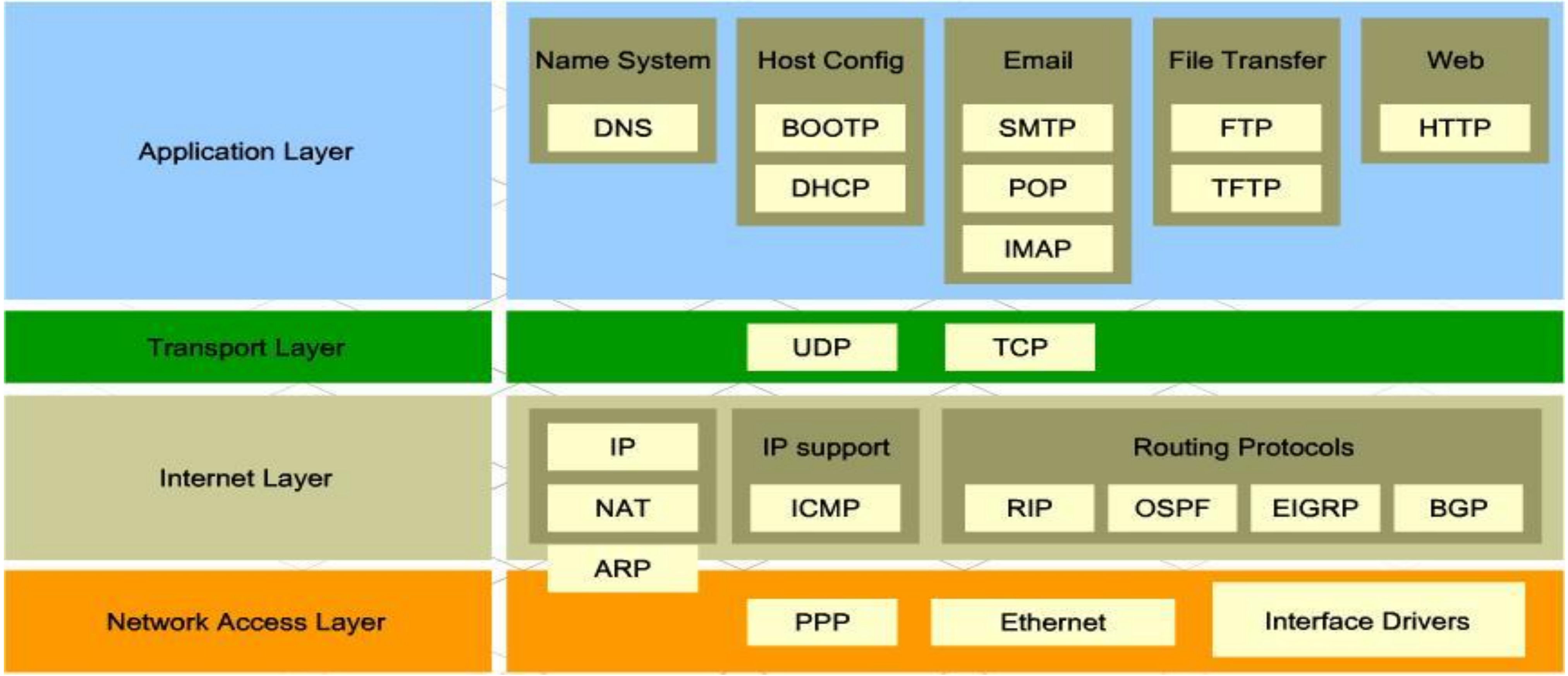


Internet protocol stack

- *application:* supporting network applications
  - HTTP, SMTP, IMAP, FTP
- *transport:* process-process data transfer
  - TCP, UDP
- *network:* routing of datagrams from source to destination
  - IP, routing protocols
- Data Link: The data link layer is responsible for moving frames from one hop (node) to the next. (Organize bits into frames; hop to hop delivery)



Protocols @ Different layers

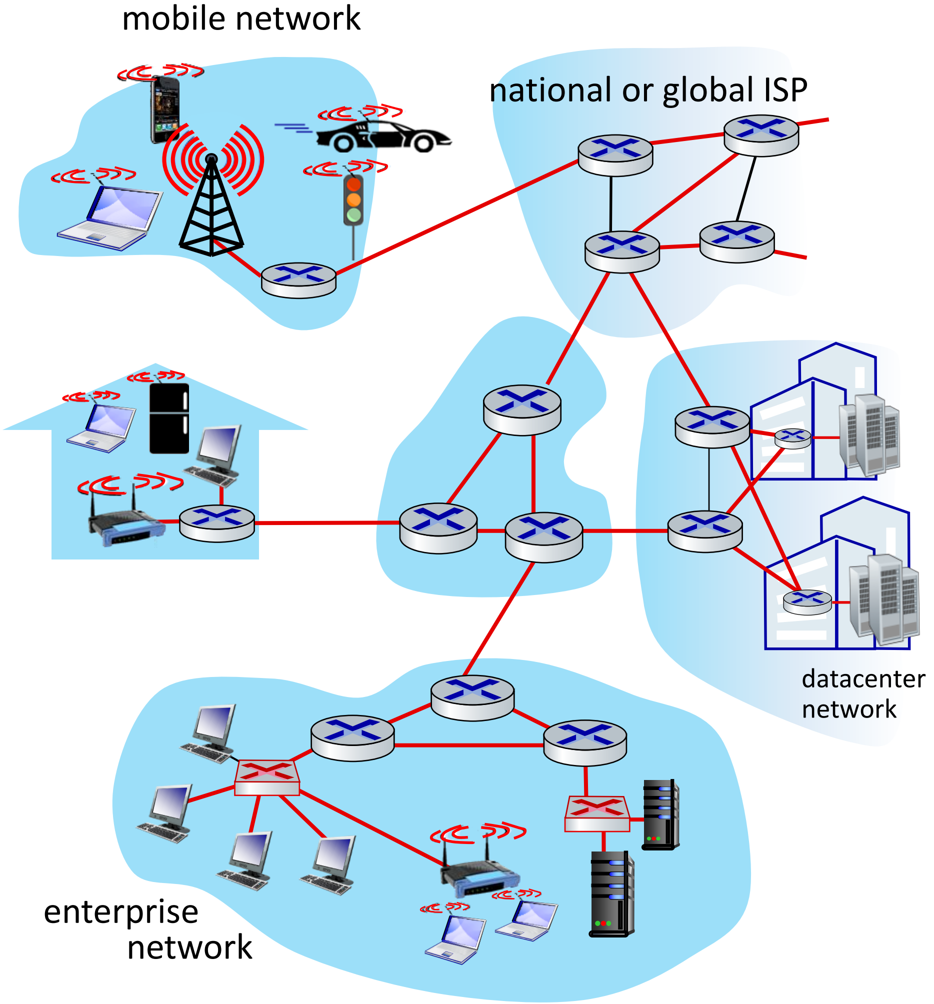


[Source: http://walkwidnetwork.blogspot.com/2013/04/application-layer-internet-protocol.html](http://walkwidnetwork.blogspot.com/2013/04/application-layer-internet-protocol.html)

Link layer: introduction

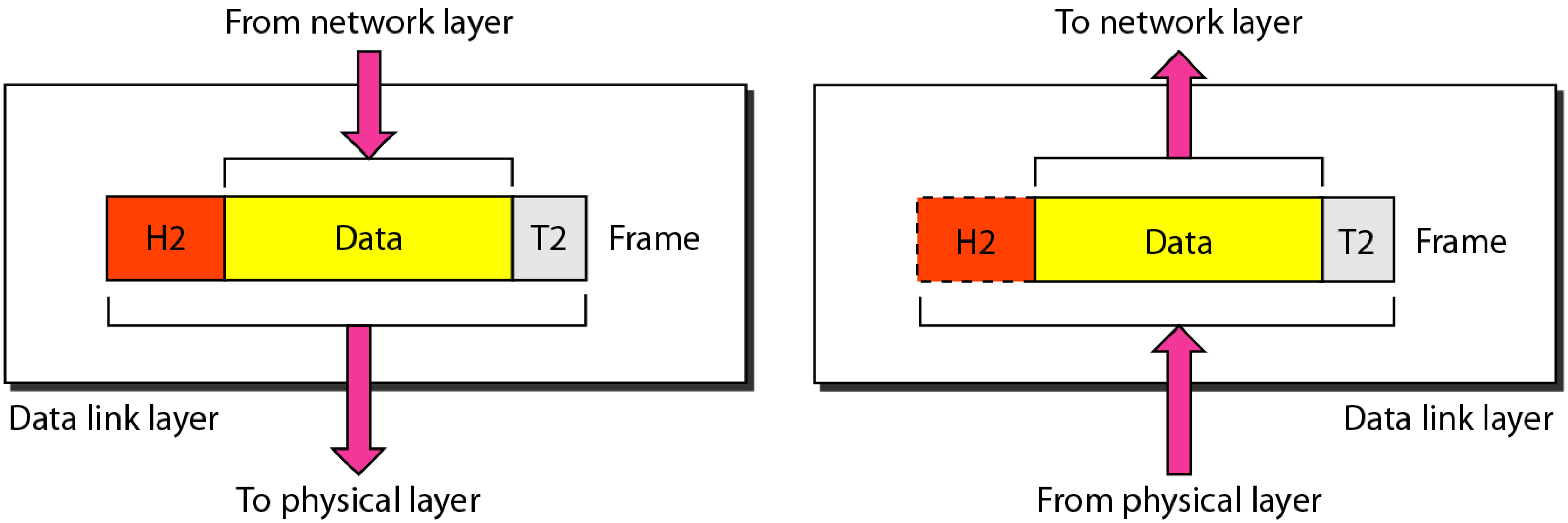
terminology:

- hosts and routers: nodes
- communication channels that connect adjacent nodes along communication path: links
  - wired
  - wireless
  - LANs
- layer-2 packet: *frame*, encapsulates datagram



link layer has responsibility of transferring datagram from one node to physically adjacent node over a link

Data Link Layer



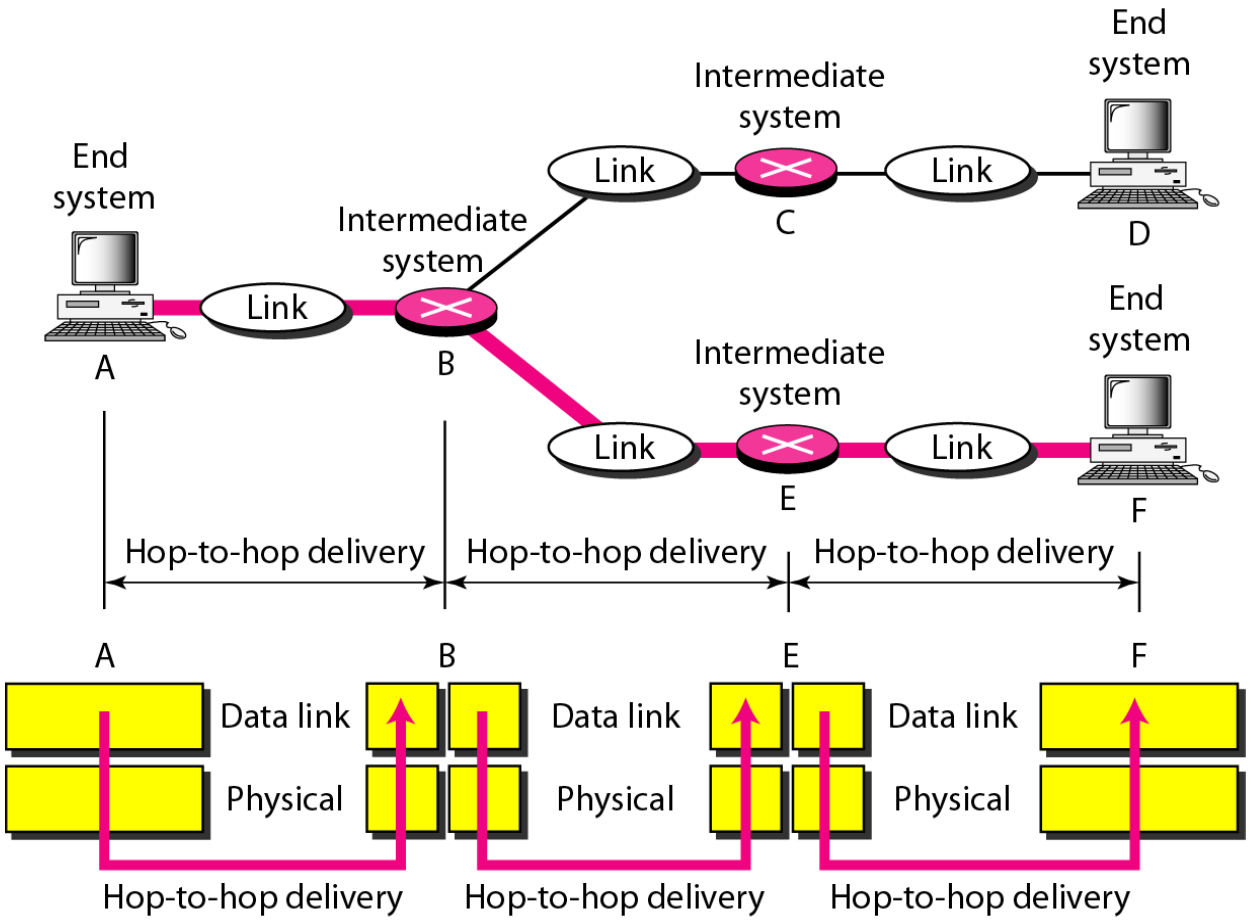
Link layer: context

- datagram transferred by different link protocols over different links:
  - e.g., WiFi on first link, Ethernet on next link
- each link protocol provides different services
  - e.g., may or may not provide reliable data transfer over link

transportation analogy:

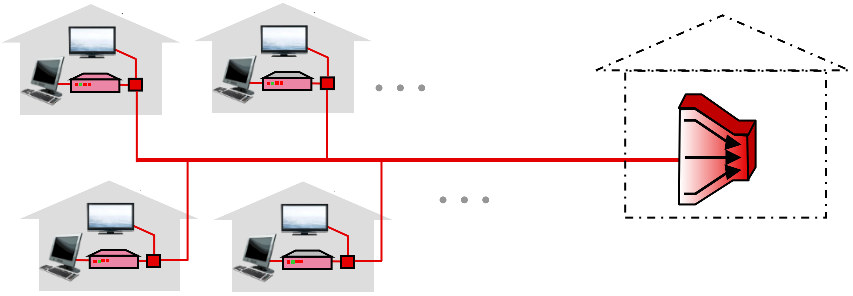
- trip from Bhilai to Agra
  - Taxi: Bhilai to Raipur
  - plane: Raipur to Delhi
  - train: Delhi to Agra
- tourist = datagram
- transport segment = communication link
- transportation mode = link-layer protocol
- travel agent = routing algorithm

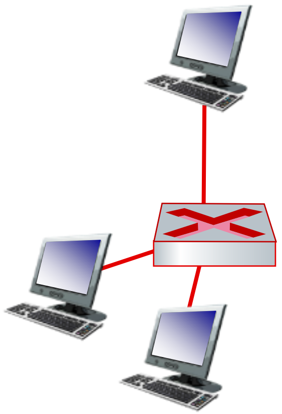
Hop to Hop Delivery (Node to Node)



Link layer: services

- **framing, link access:**
  - encapsulate datagram into frame, adding header, trailer
  - channel access if shared medium
  - “MAC” addresses in frame headers identify source, destination (different from IP address!)
- **reliable delivery between adjacent nodes**
  - we already know how to do this!
  - wireless links: high error rates
    - *Q:* why both link-level and end-end reliability?






Link layer: services (more)

- flow control:
  - pacing between adjacent sending and receiving nodes
- error detection:
  - errors caused by signal attenuation, noise.
  - receiver detects errors, signals retransmission, or drops frame
- error correction:
  - receiver identifies *and corrects* bit error(s) without retransmission
- half-duplex and full-duplex:
  - with half duplex, nodes at both ends of link can transmit, but not at same time