The Migration Toward the Optical Internet

Lesson 2

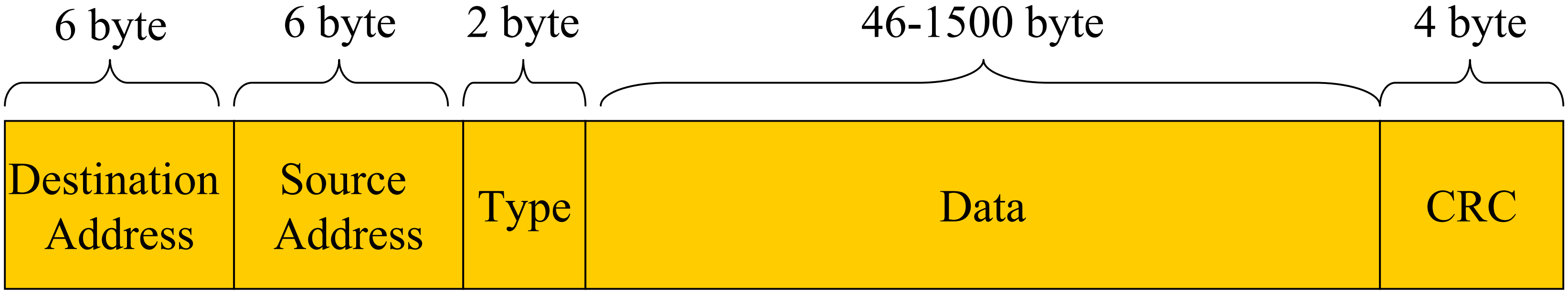
Luca Valcarenghi

TCP/IP Model: Link Layer

- Link layer: provides protocol to connect the host to the underlying network
- Many different link layers supported by TCP/IP depending on the used network hardware
  - Ethernet, token ring, Fiber Distributed Data Interface (FDDI), RS-232 serial lines, etc.

Link Layer Main Services (1)

- Encapsulation of IP datagrams into data link protocols
- Example of Ethernet Encapsulation



6 byte	6 byte	2 byte	46-1500 byte	4 byte
Destination Address	Source Address	Type	Data	CRC

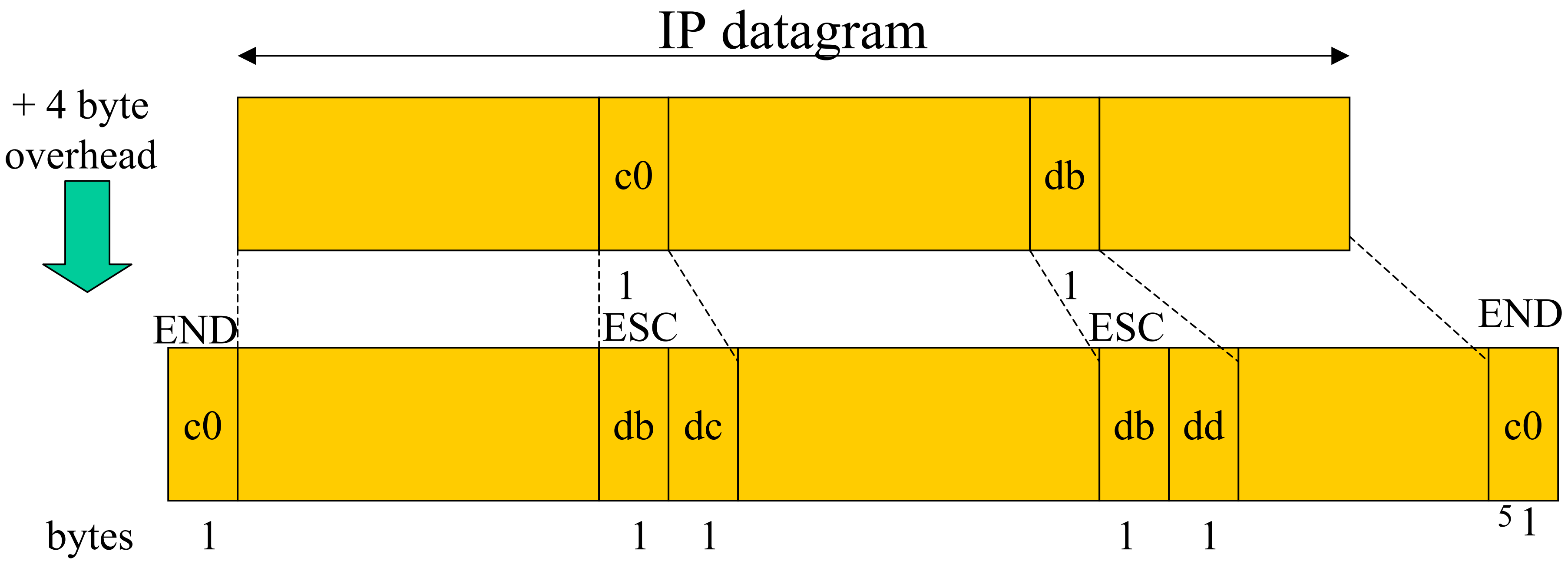
- DESTINATION and SOURCE addresses are commonly called HARDWARE ADDRESSES
- Minimum DATA SIZE: 46 bytes
- Maximum DATA SIZE: 1500 bytes
- CRC= Cyclic Redundancy Check to detect errors in the rest of the frame

Link Layer Main Services (2)

- Encapsulation of IP datagrams on SERIAL LINES
  - Serial Line IP (SLIP)
  - Point-to-Point Protocol (PPP)

Serial Line IP (SLIP)

- SLIP famous for connecting home systems to the Internet through RS-232 serial port and high-speed modems
- SLIP encapsulation format



SLIP Encapsulation Rules

- IP DATAGRAM is terminated with a special character called END (0xc0)
- Most implementations transmit an END character at the beginning of the DATAGRAM to prevent any line noise to be interpreted as part of this datagram
- If a byte of the IP datagram equals the END character the 2-byte sequence 0xdb, 0xdc is transmitted instead
- 0xdb called the SLIP ESC character but its value is different from ASCII ESC (0x1b)
- If a byte of the IP datagram equals the SLIP ESC the two-byte sequence 0xdb, 0xdd is transmitted instead

SLIP Encapsulation Drawbacks

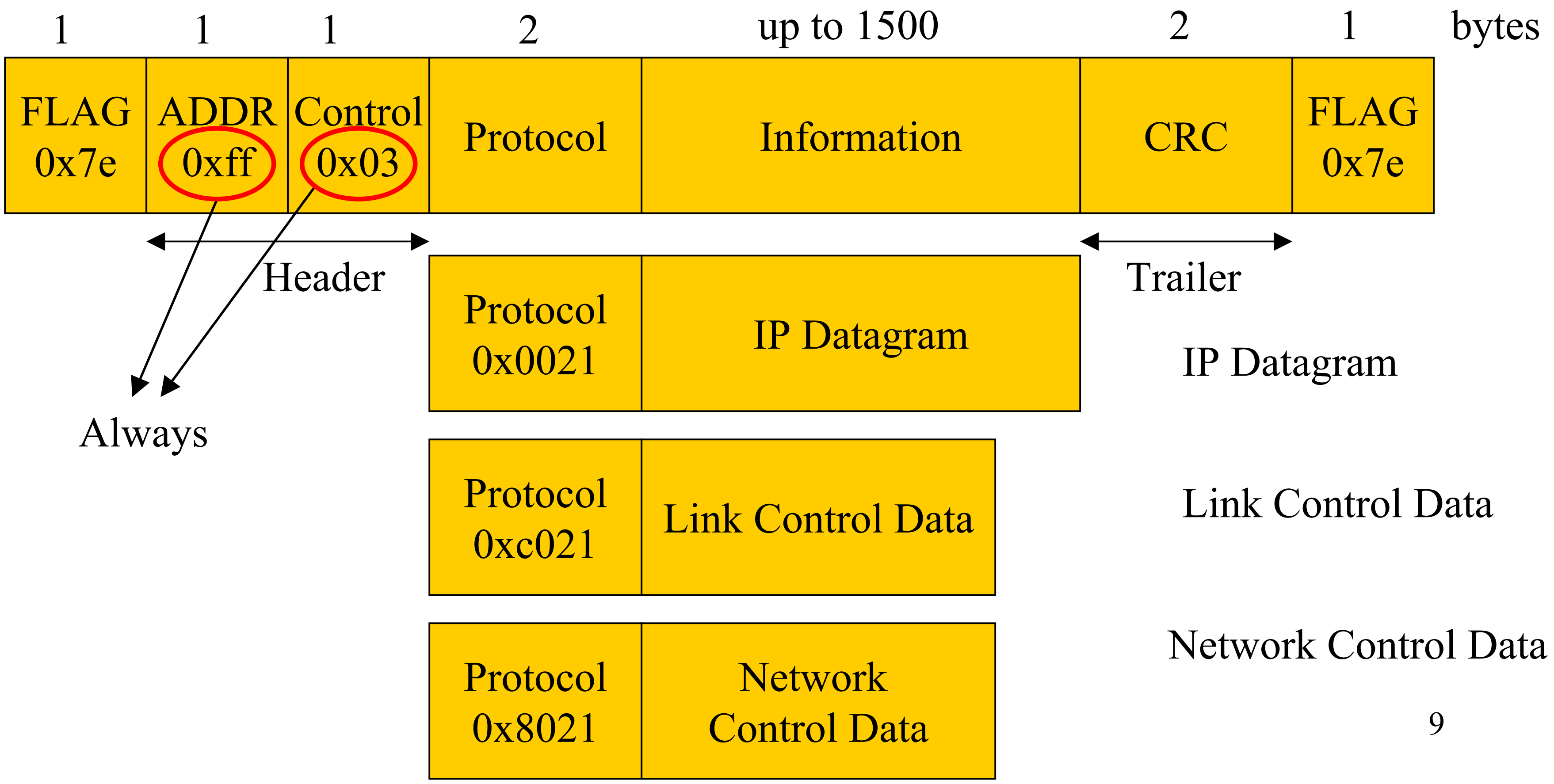
- Each end must know other end’s IP address
- No type filed for specifying higher layer protocol carried
- No CRC checksum; higher layer must provided
- Often slow: 19200 b/s

Point-to-Point Protocol (PPP)

- PPP consists of three components
  - A way to encapsulate IP datagrams on a serial link
    - It supports both SERIAL INTERFACE present ion most computers (8 bits of data and no parity) or bit-oriented synchronous links
  - A Link Control Protocol (LCP) to establish, configure, and test the data-link connection
  - A family of Network Control Protocols (NCPs) specific to different network layer protocols

PPP Frame Format

- Similar to ISO standard High-level Data Link Control (HDLC)



1	1	1	2	up to 1500	2	1	bytes
FLAG 0x7e	ADDR 0xff	Control 0x03	Protocol	Information	CRC	FLAG 0x7e	

Protocol	Data	
Protocol 0x0021	IP Datagram	IP Datagram
Protocol 0xc021	Link Control Data	Link Control Data
Protocol 0x8021	Network Control Data	Network Control Data

PPP “Escaping”

- PPP needs to “escape” the byte value 0x7e (FLAG) when it appears in the Information field
  - Escape (ESC) byte added: 0x7d
  - Whenever ESC byte is present in the frame, next character in the frame has its 6th bit complemented
  - Example of PPP escaping
    - Byte 0x7e (if present in the data) is transmitted as the 2-byte sequence 0x7d, 0x5e
    - The byte 0x7d is transmitted as the 2-byte sequence 0x7d, 0x5d. This is the escape of the escape byte
    - By default, a byte with value less than 0x20 (i.e., and ASCII control character) is also escaped. That is 0x01 → 0x7d, 0x21
      - (The reason is to avoid bytes less than 0x20 to appear as ASCII control characters to the serial driver)

PPP Advantages

- PPP tries to reduce link layer overheads by eliminating ADDRESS and CONTROL fields (constant) and to reduce to 1 byte the PROTOCOL field
- After the reduction
  - PPP overhead = SLIP overhead + 3 bytes
  - 3 bytes
    - 1 byte PROTOCOL field
    - 2 bytes CRC

Serial Line Throughput Calculation

- Maximum Transmission Unit (MTU): upper limit to the number of bytes of data to be encapsulated into a serial line FRAME
- If IP datagram is larger ⇒ IP performs fragmentation breaking the datagram up into smaller pieces (fragments) so that each fragment is smaller than MTU
- Most existing packet networks use variable frame lengths but ATM uses fixed frame lengths (53 bytes)

Variable Frame Length

- Standard frame, such as PPP (HDLC)
- V = number of overhead bits in the frame including frame header and trailer
- K_{max} = maximum fragment length (i.e., MTU)
- Each packet is fragmented into as many MTUs as possible
- Last fragment contains what is left over
- Packets of length M fragmented into $\lceil M / K_{\max} \rceil$ fragments (MTUs)
- The first $\lceil M / K_{\max} \rceil - 1$ of these fragments contain K_{max} bits each
- The final fragment contains between 1 and K_{max} bits
- Total number of bits in the resulting frame is:

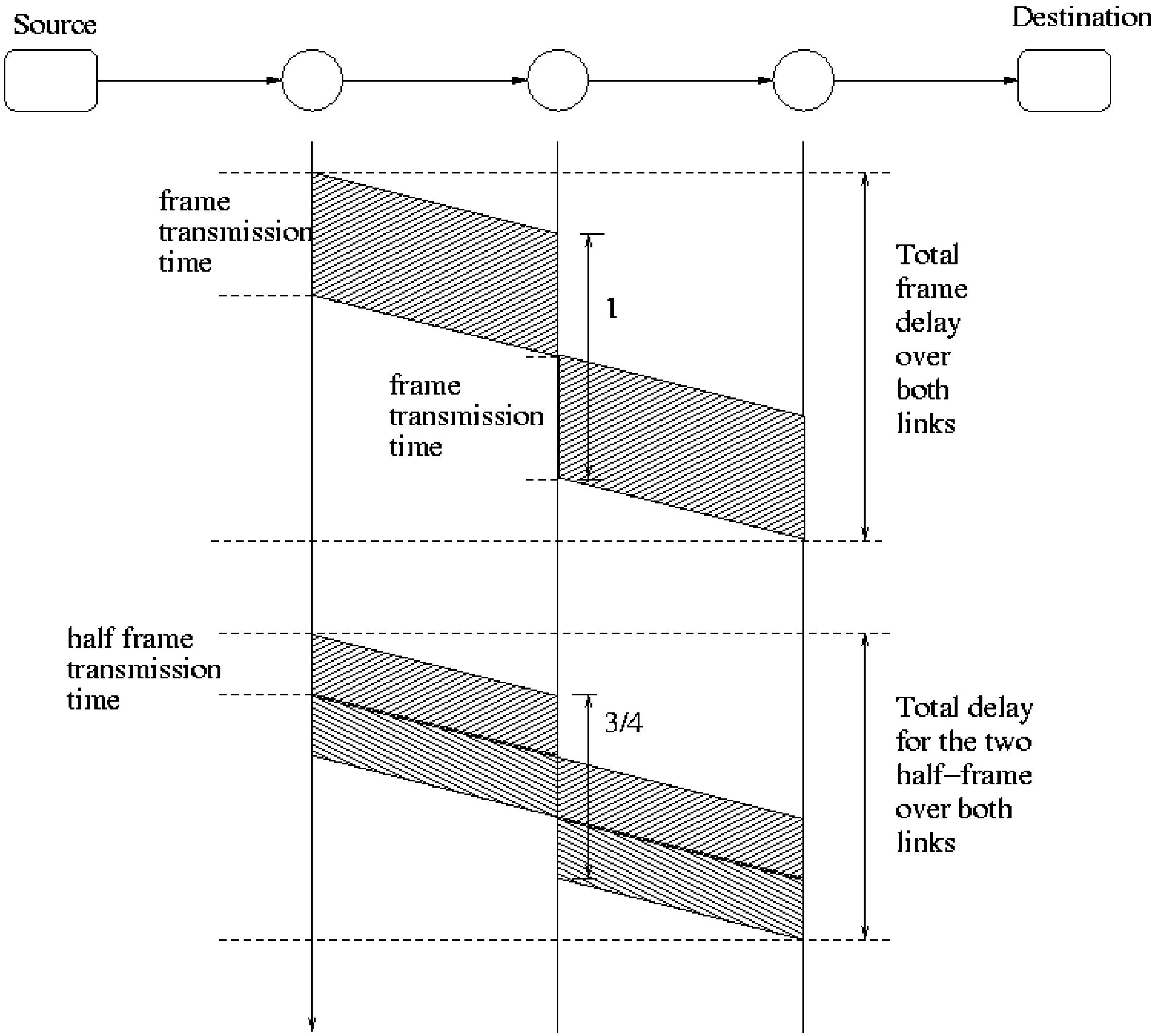
$$\text{total bits} = \underbrace{M}_{\text{original packet length}} + \underbrace{\left\lceil \frac{M}{K_{\max}} \right\rceil V}_{\text{framing overhead}}$$

MTU (K_{max}) Length Considerations

- If $K_{max} \downarrow \Rightarrow$ # of frames $\uparrow \Rightarrow$ framing overhead $\uparrow \Rightarrow \lceil M/K_{max} \rceil V \uparrow$
- $$\lim_{M \to \infty} \frac{\lceil M / K_{\max} \rceil V}{M + \lceil M / K_{\max} \rceil V} = \frac{V}{K_{\max} + V}$$
- Percentage of total bits that are overhead bits
- Each node must do processing per frame $\Rightarrow$ if $K_{max} \downarrow \Rightarrow$ number of frames $\uparrow \Rightarrow$ node processing $\uparrow$
- With optical networks high speed is required LARGE MTU
- Arguments for small MTU
  - Pipelining effect
  - Reduced message delay

Pipelining Effect

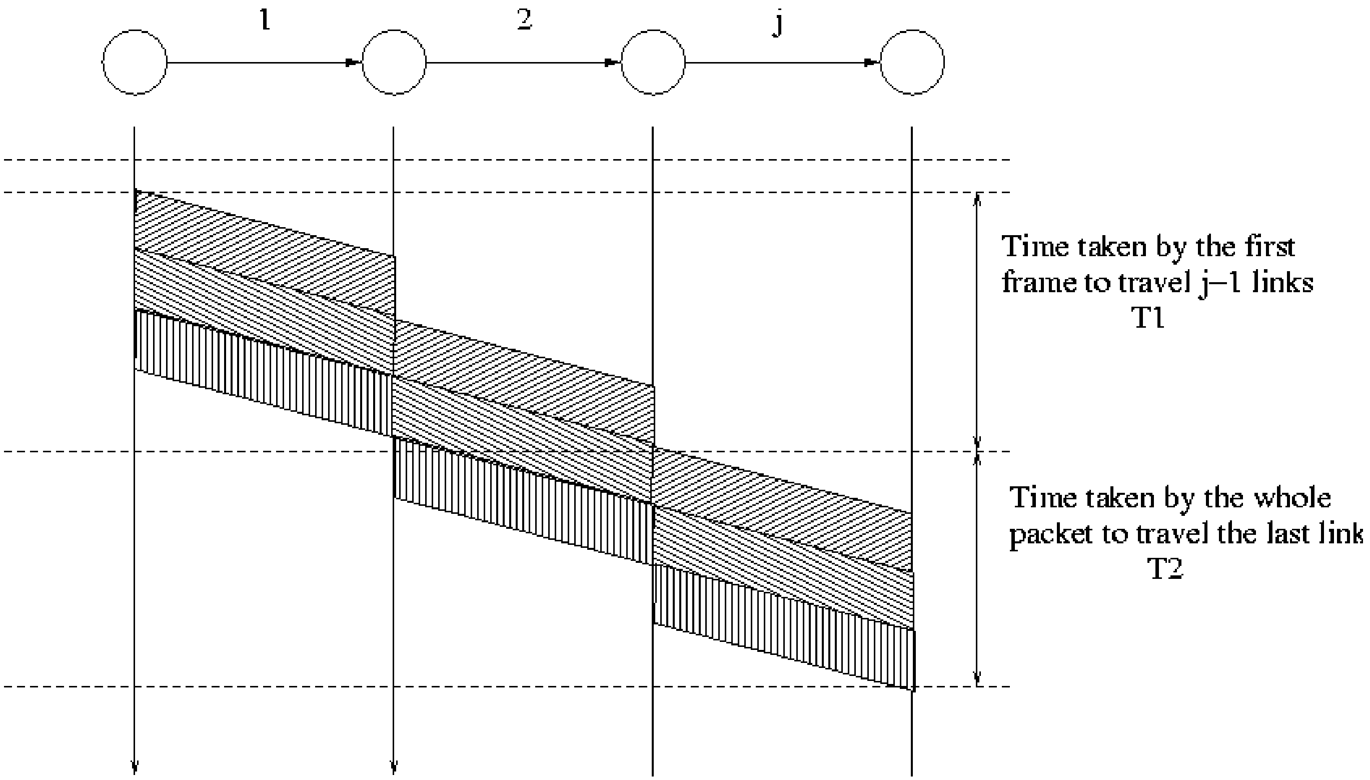
- Assumption: a packet must be completely received over one link before starting transmission over the next



Combined Effect of Overhead and Pipelining on the Message Delay (1)

- Message (file transfer) traffic
- Assumptions
  - Packet of length M fragmented into MTUs (maximum length fragments)
  - Packets must be transmitted over *j* equal-capacity link with capacity *C* b/s
  - Network lightly loaded ⇒ waiting at the nodes for other traffic can be ignored
  - TC: number of bit transmission times required for message delivery

Combined Effect of Overhead and Pipelining on the Message Delay (2)



$$T = T_1 + T_2 = \frac{1}{C}(K_{\max} + V)(j-1) + \frac{1}{C}\left(M + \left\lceil \frac{M}{K_{\max}} \right\rceil V\right)$$

Combined Effect of Overhead and Pipelining on the Message Delay (3)

- $E\{TC\}$= expected value of TC over the message lengths M
  - Assumption $E\{\lceil M/K_{max} \rceil\}= E\{M/K_{max}\}+1/2$
  - Reasonable if the distribution of M is reasonably uniform over spans of K_{max} bits

$$E\{TC\} \cong (K_{\max}+V)(j-1)+E\{M\}+\frac{E\{M\}V}{K_{\max}}+\frac{V}{2}$$

Combined Effect of Overhead and Pipelining on the Message Delay (4)

- Value of K_{max} minimizing TC

$$\frac{\partial E\{TC\}}{\partial K_{\max}} = (j-1) - \frac{E\{M\}V}{K_{\max}^{2}} = 0$$

$$\Downarrow$$

$$K_{\max} = \sqrt{\frac{E\{M\}V}{j-1}} \qquad (*)$$

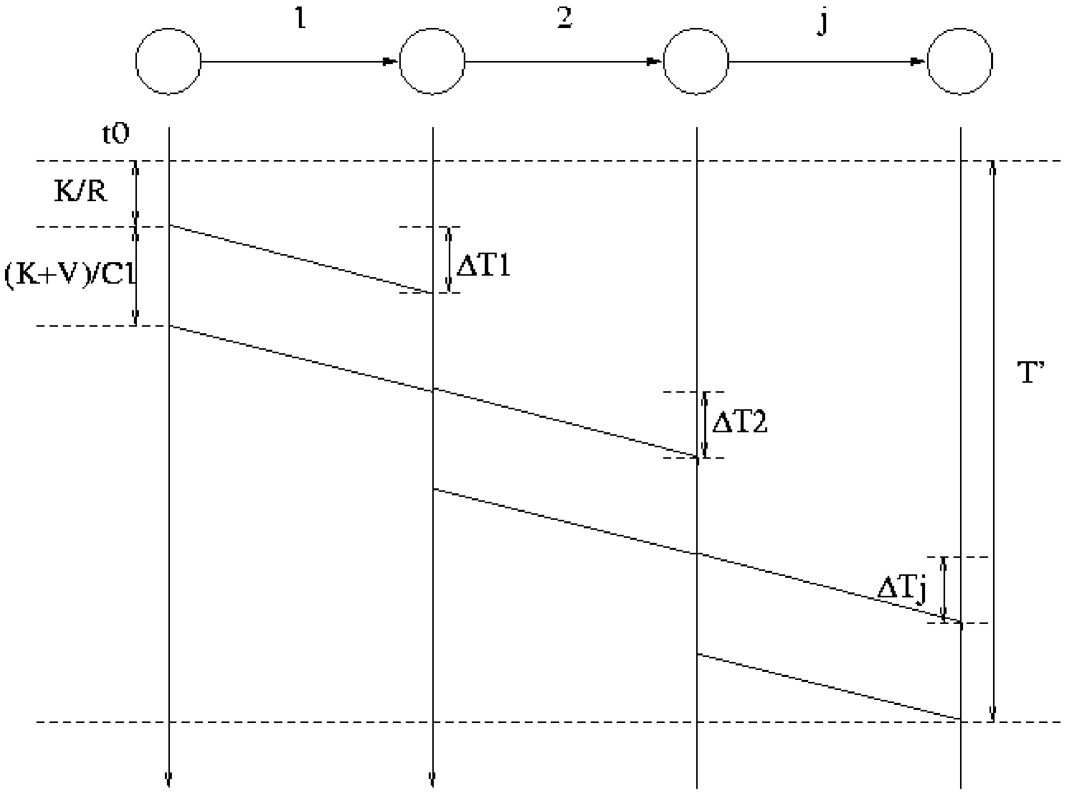
Combined Effect of Overhead and Pipelining on the Message Delay (5)

- Equation * shows the trade-off between overhead and pipelining
  - As $V \uparrow \Rightarrow K_{max}$ should be increased
  - As path length $j \uparrow \Rightarrow K_{max}$ should be reduced

Combined Effect of Overhead and Pipelining on the Message Delay (6)

- Stream type traffic
- Interested in delay from when a given bit enters the network until that bit leaves
  - For message traffic interested in the delay from arrival of the message to delivery of the complete message
- Example for stream type traffic delay
  - Light load
  - Arrival rate R
  - Fragment length K
  - K/R waiting time of the first bit during the frame assembly (CRC, etc.)
  - Path link capacity $C_1, C_2, \ldots, C_i$ ($C_i \geq R$, for each i)
  - V bits of framing overhead
  - $(K+V)/C_i$ = fragment delay on link i
  - When a given packet is completely received at the last node of the network, the first bit of the packet can be immediately released

Combined Effect of Overhead and Pipelining on the Message Delay (7)



$$T = \frac{K}{R} + (K+V)\sum_i \frac{1}{C_i}$$

$$T' = T + \sum_i \Delta T_i$$

$$\Delta T_i = \text{propagation delay on link } i$$

$$\frac{K}{R_i} \leq \frac{K+V}{C_i}$$

Combined Effect of Overhead and Pipelining on the Message Delay (8)

- As $K\downarrow \Rightarrow T\downarrow$ until $(K+V)/C_i = K/R$ for some link $\Rightarrow K_{min}$
- Frame lengths usually larger than K_{min} because of other traffic in the network
- As $C_i\uparrow \Rightarrow$ main delay contribution given by K/R that is unaffected by other traffic
- For 64 kb/s voice traffic usual frame length $\Rightarrow K \le 500$ bit $\Rightarrow K/R \cong 0.008$s
  - Voice traffic delay threshold is few tens of ms
- ATM frame 53 bytes (424 bits) FIXED
- Typical maximum frame lengths K_{max} (MTU)
  - Wide Area Networks (WAN): 1-few thousand bits
  - Local Area Networks longer frame tha WAN because
    - path usually consists of a single multiaccess link
    - delay and congestion are typically less important

Summary

- Internet history
- TCP/IP reference Model
- Description of the TCP/IP reference model layers
- Comparison between ISO-OSI and TCP/IP reference model
- TCP/IP reference model: Link Layer
- Link layer protocols
  - SLIP
  - PPP
- Serial line throughput calculation

TCP/IP Reference Model: Internet/Network Layer

- Internet/Network layer task
  - ROUTING packets from source to destination
- In TCP/IP reference model this task is accomplished by Internet Protocol (IP)
- All TCP, UDP, ICMP, and IGMP data gets transmitted as IP datagrams
- IP provides an UNRELIABLE, CONNECTIONLESS DATAGRAM delivery

Internet Protocol (IP) (1)

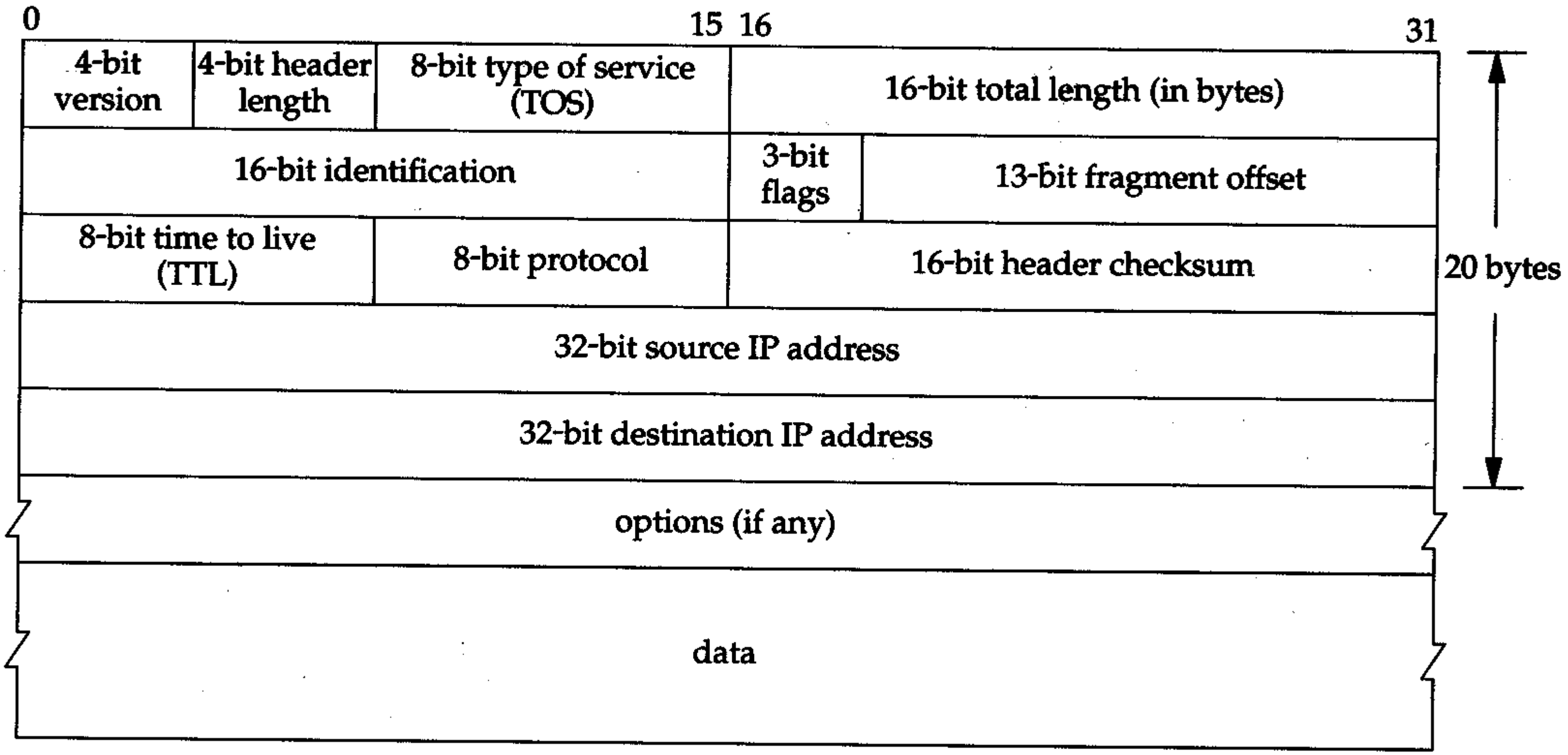
- UNRELIABLE
  - No guarantees that an IP datagram successfully gets to its destination
  - Best effort service
  - IP error (e.g., router down) handling algorithm:
    - Datagram thrown away
    - ICMP message sent back to the source
    - Any required reliability must be provided by upper layers (e.g., TCP)

Internet Protocol (IP) (2)

- CONNECTIONLESS
  - IP does not maintain any state information about successive datagrams
  - Each datagram handled independently from all other datagrams
  - IP datagrams can get delivered out of order
- DATAGRAM
  - Unit of information of the Internet/Network layer
  - Usually represented in 32-bit words and identified from 0 to 31
  - The 4 bytes in the 32-bit value are transmitted in order: bit 0-7 first, then 8-15, etc. ⇒BIG ENDIAN byte ordering
  - Current version is no. 4 ⇒ IPv4
  - Recently proposed version no. 6 ⇒ IPv6
    - IPv6 includes QoS features

IP Datagram Format (1)

IP Header



<table>
<tr><th>0</th><th></th><th>15</th><th>16</th><th>31</th></tr>
<tr><td>4-bit version</td><td>4-bit header length</td><td>8-bit type of service (TOS)</td><td colspan="2">16-bit total length (in bytes)</td></tr>
<tr><td colspan="3">16-bit identification</td><td>3-bit flags</td><td>13-bit fragment offset</td></tr>
<tr><td colspan="2">8-bit time to live (TTL)</td><td>8-bit protocol</td><td colspan="2">16-bit header checksum</td></tr>
<tr><td colspan="5">32-bit source IP address</td></tr>
<tr><td colspan="5">32-bit destination IP address</td></tr>
<tr><td colspan="5">options (if any)</td></tr>
<tr><td colspan="5">data</td></tr>
</table>

IP Datagram Format (2)

- VERSION: IP protocol version
- HEADER LENGTH: number of 32-bit words in the header
  - header max length: 60 bytes (15x4=60)
  - problem for explicit routing and record route
- TYPE OF SERVICE (TOS)
  - 3-bit precedence filed (ignored)
  - 4 TOS bits (not supported by old TCP/IP implementations)
  - 1 unused bit (must be 0)

IP Datagram Format (3)

- TOTAL LENGTH: total length of the IP datagram in bytes
  - maximum size of an IP datagram 65535 bytes
- IDENTIFICATION: uniquely identifies each datagram sent by a host
  - normally increments by 1 each time a datagram is sent
- FLAG: one bit as the "more fragments" bit. This bit turned on for each fragment comprising a datagram except for the final fragment
- FRAGMENT OFFSET: offset in 8-byte units of this fragment from the beginning of the original datagram

IP Datagram Format (4)

- TIME TO LIVE (TTL): set an upper linit on the number of routers through which a datagram can pass
  - usually initialized by the sender to some value (32 or 64)
  - decremented by one by every router handling the datagram
- PROTOCOL: identifies which protocol gave the data for IP to send

IP Datagram Format (5)

- HEADER CHECKSUM: calculated over the IP header only
  - checksum calculated as the 16-bit ones's complement sum of the header (the entire header considered as a sequence of 16-bit words)
  - sender sets the value of the checksum
  - receiver calculates checksum as the sender and compares it with the 16-bit header checksum in the datagram
  - if checksum error ⇒IP discard the received datagram
  - no error message is generated
- SOURCE IP ADDRESS: IP address of the sender
- DESTINATION IP ADDRESS: IP address of the receiver
  - never changed except when EXPLICIT ROUTING is used
- OPTIONS: variable-length lists of optional information for the datagram
  - Examples: security and handling restrictions, record route, strict source routing, etc.