# CSE398: Network Systems Design

Instructor: Dr. Liang Cheng
Department of Computer Science and Engineering
P.C. Rossin College of Engineering & Applied Science
Lehigh University

February 14, 2005

# Outline

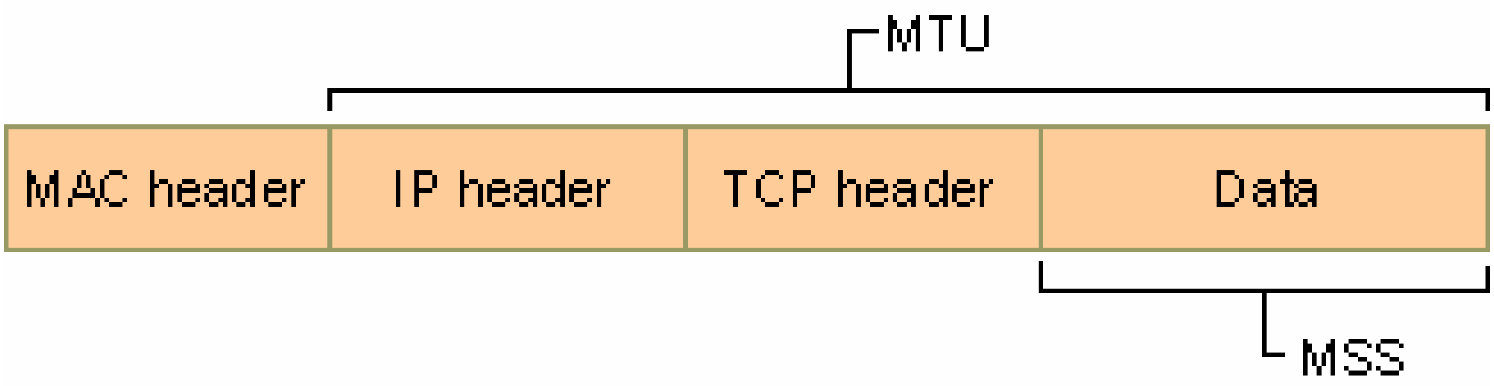
- Recap
  - Why "do not fragment"? Some concepts.
  - Packet processing algorithms (Chapter 5)
- Packet processing algorithms
- Summary and homework

# MTU

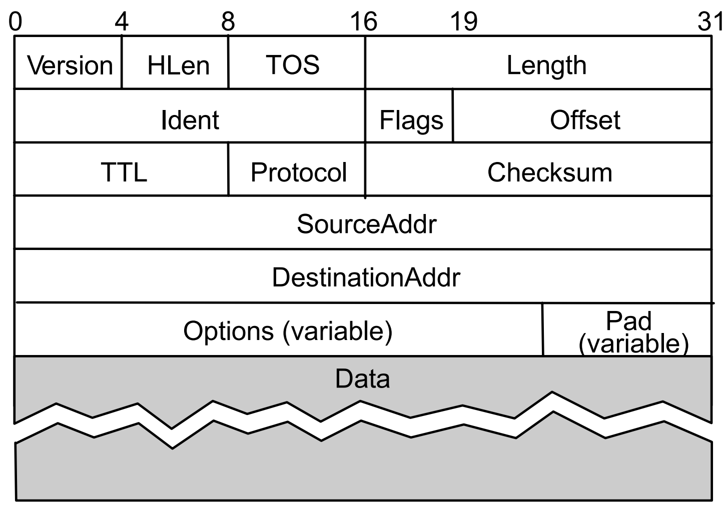
- Maximum transmission unit
- The maximum amount of data that a link-layer frame can carry. For example, Ethernet frames can carry up to 1,500 bytes of data. [From Jim Kurose's textbook, pp. 328]
- MTU is the largest IP datagram that it can carry in a frame. [Larry Peterson's textbook, pp. 242]

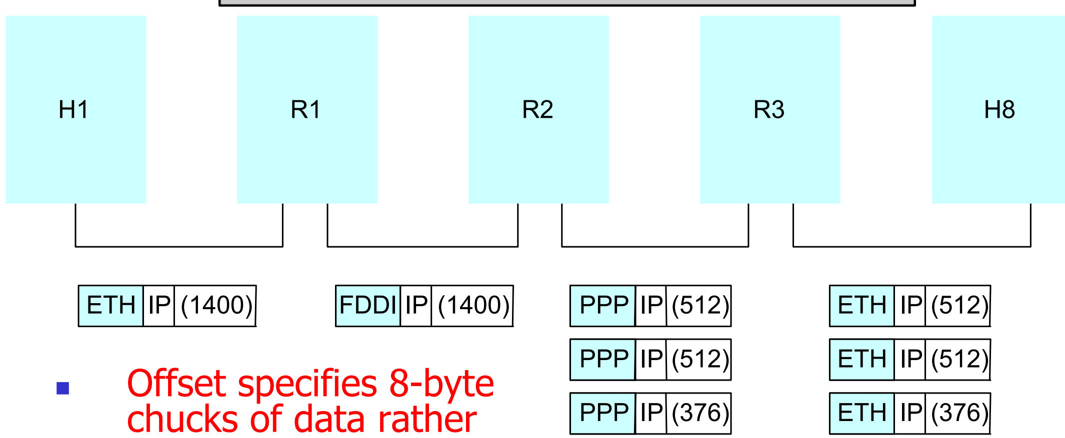
# MSS

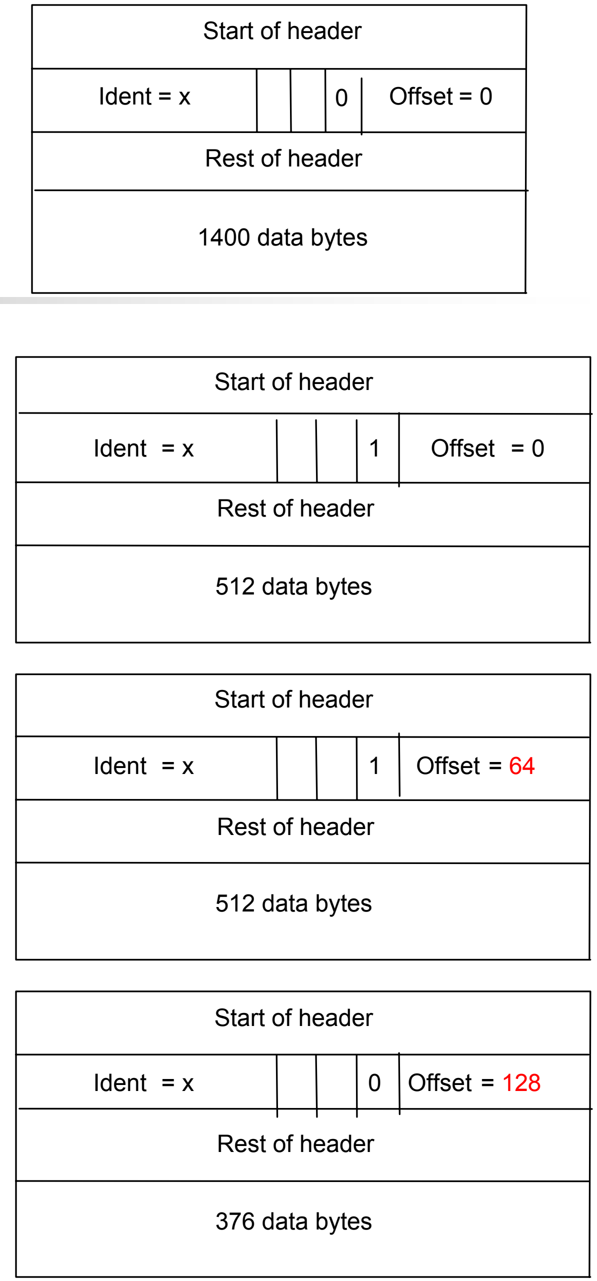
- Maximum segment size
- The maximum amount of data that can be placed in a TCP segment. [From Jim Kurose's textbook, pp. 230]



# Fragmentation




- Offset specifies 8-byte chucks of data rather than individual bytes

# Path MTU

- Path maximum transmission unit (PMTU) discovery is described in RFC 1191
  - Normally, a host sets DF in all datagrams so that if the route changes and the new PMTU is lower, it will be discovered.
  - ICMP Destination Unreachable

# Example

- ping -f -n <number of pings> -l <size> <destination ip address>
- ping -f -n 1 -l 1472 www.lehigh.edu
- ping -f -n 1 -l 1473 www.lehigh.edu

# TCP Connection Recognition Algorithm (partially shown)

Given: a copy of traffic passing across a network.

Produce: a record of TCP connections present in the traffic.

Initialize a connection table, C, to empty;

For each IP datagram that carries a TCP segment {

Extract the IP source, S, and destination, D, addresses;

Extract the source, $P_1$, and destination, $P_2$, port numbers;

Use $(S, D, P_1, P_2)$ as a lookup key for table C and

If the segment has the RESET bit set, delete the entry;

Else if the segment has the FIN bit set, mark the connection

closed in one direction, removing the entry from C if the connection was previously closed in the other;

Else if the segment has the SYN bit set, mark the connection as

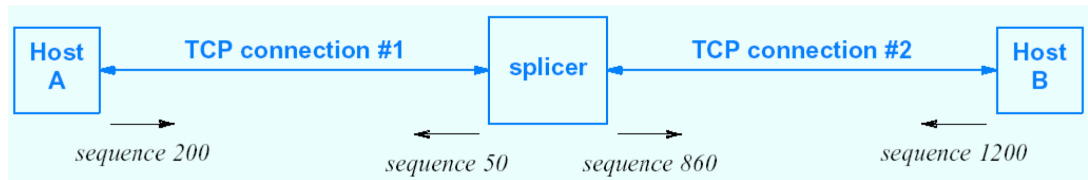
being established in one direction, making it completely established if it was previously marked as being established in the other;

}

- Problem 11 of Chapter 5 (Page 64).

# TCP Splicing

- Join two TCP connections
- Allow data to pass between them
- To avoid termination overhead translate segment header fields
  - Acknowledgement number
  - Sequence number




| Connection & Direction | Sequence Number | Connection & Direction | Sequence Number |
|---|---|---|---|
| Incoming #1 | 200 | Incoming #2 | 1200 |
| Outgoing #2 | 860 | Outgoing #1 | 50 |
| Change | 660 | Change | -1150 |

# TCP Splicing Algorithm

```
Given: two TCP connections.
Produce: sequence translations for splicing the connection.
Compute D1, the difference between the starting sequences
    on incoming connection 1 and outgoing connection 2;
Compute D2, the difference between the starting sequences
    on incoming connection 2 and outgoing connection 1;
For each segment {
    If segment arrived on connection 1 {
        Add D1 to sequence number;
        Subtract D2 from acknowledgement number;
    } else if segment arrived on connection 2 {
        Add D2 to sequence number;
        Subtract D1 from acknowledgement number;
    }
}
```

# Outline

- Recap: packet processing algorithms
- Packet processing functions
- Summary and homework

# Goals

- Identify functions that occur in packet processing
- Devise set of operations sufficient for all packet processing
- Find an efficient implementation for the operations

# What We Will Consider

- Address lookup and packet forwarding
- Error detection and correction
- Fragmentation, segmentation, & reassembly
- Frame and protocol demultiplexing
- Packet classification
- Queueing and packet discard
- Scheduling and timing
- Security: authentication and privacy
- Traffic measurement and policing
- Traffic shaping

# Address Lookup & Packet Forward

- Forwarding requires address lookup
- Lookup is table driven
- Cost depends on
  - size of table and type of lookup
- Two types
  - Exact match (typically layer 2)
  - Longest-prefix match (typically layer 3)

# Error Detection and Correction

- Data sent with packet used as verification
  - Checksum
  - CRC
- Cost proportional to
  - size of packet
- Often implemented with special-purpose hardware
  - CRC chip
  - Incremental update (e.g., hop count) in the Internet checksum
- Error correction for multimedia applications
  - Why for multimedia?
  - Why not for all?

# Fragmentation, Segmentation, and Reassembly

- IP fragments and reassembles datagrams
- ATM segments and reassembles AAL5 packets
  - Same idea; details differ
- Cost is high because
  - State must be kept and managed
  - Unreassembled fragments occupy memory

# Frame & Protocol Demultiplexing

- Traditional technique used in layered protocols
- Type appears in each header
  - Assigned on output
    - IP -> Ethernet: IP module sets the frame type to $(0800)_{16}$
  - Used on input to select "next" protocol
- Cost of demultiplexing proportional to
  - number of layers

# Packet Classification

- An static example
  - A frame containing an IP datagram that carries a TCP segment
  - A frame containing an IP datagram that carries a UDP datagram
  - A frame containing an IP datagram that carries a ICMP message
  - A frame that contains something other than the above (?)
- Alternative to demultiplexing
- Crosses multiple layers in one step
  - Achieves lower cost
- More on classification later in the course
  - Agere's Functional Programming Language (FPL)

# Outline

- Recap
  - Packet processing algorithms
- Packet processing functions
- Summary and homework

# Homework (due on 02/21)

- The Lab#2 group report and an individual report that includes what you have learned and your role in the lab project.
- Please also study Problem 11 of Chapter 5 (Page 64).