ECE 697J - Advanced Topics in Computer Networks

Packet Processing – III

9/18/03

Packet Processing Functions

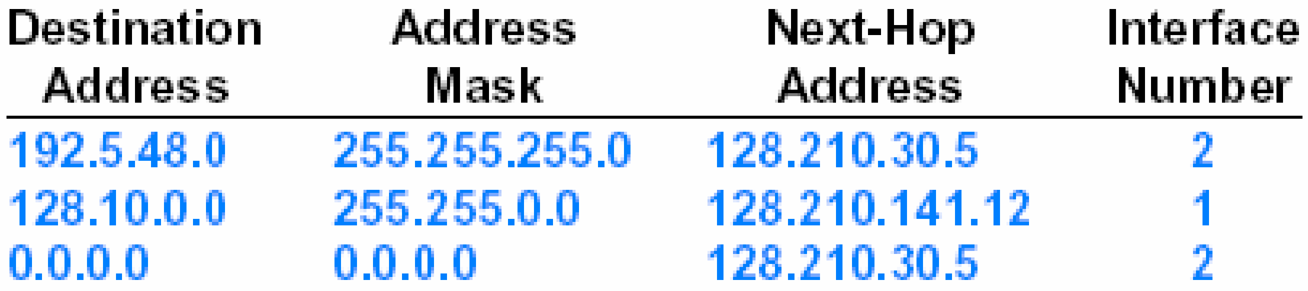
- Basic network system functionality
  - Address lookup
  - Error detection and correction
  - Fragmentation/re-assembly
  - Queuing
  - Scheduling
  - Security
  - Traffic measurement/shaping
  - Protocol demultiplexing
  - Packet classification

Address Lookup

- Related to forwarding
  - Send packet toward destination
  - Table driven
- Layer 2
  - MAC address lookup
  - Exact match
- Layer 3
  - IP address lookup
  - Longest prefix match
- Cost depends on size of table and type of lookup

IP Forwarding

- Forwarding decision is made based on routing table
  - There is an important difference between a **routing table** and a **forwarding information base** (FIB) (or **forwarding table**)
- Routing is always done on the most specific prefix
  - Most specific prefix = longest prefix
- Example routing table:

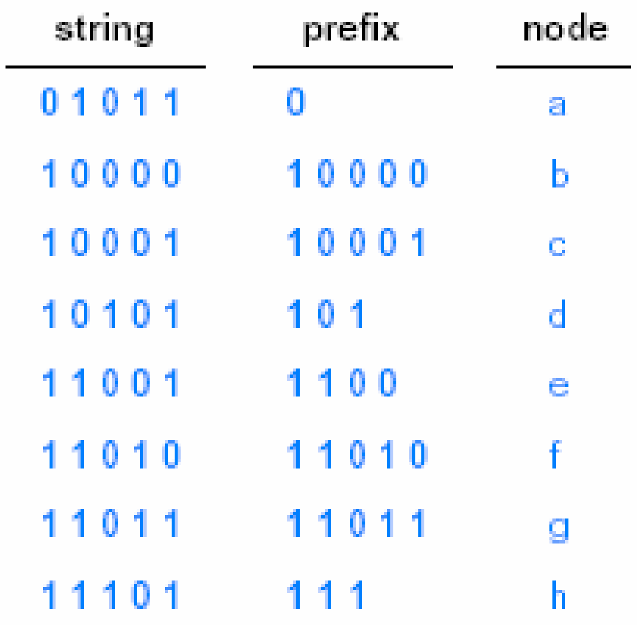


Destination Address	Address Mask	Next-Hop Address	Interface Number
192.5.48.0	255.255.255.0	128.210.30.5	2
128.10.0.0	255.255.0.0	128.210.141.12	1
0.0.0.0	0.0.0.0	128.210.30.5	2

- Routing information contains outgoing interface (and next hop)
- How to implement routing lookup?
  - Sequential search impractical (30,000 entry table)

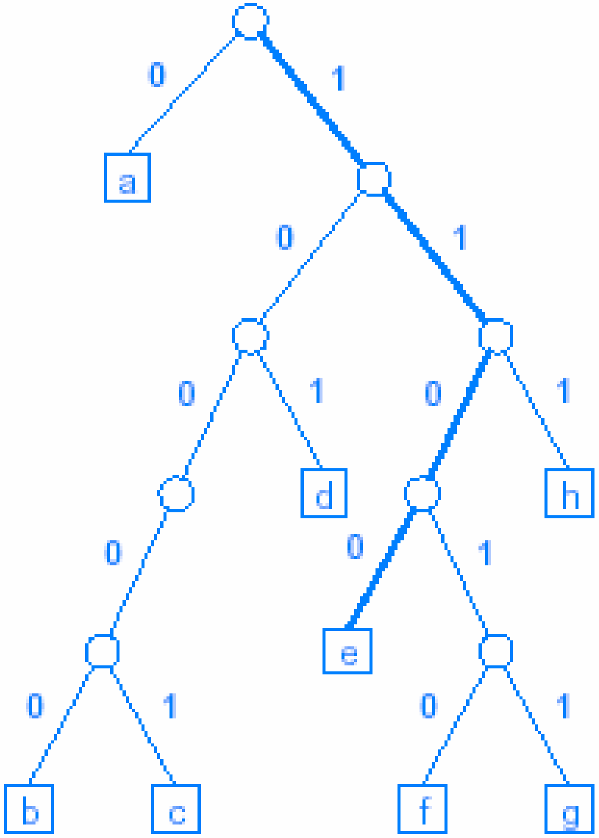
Routing Tree

- Example routing tree:



string	prefix	node
01011	0	a
10000	10000	b
10001	10001	c
10101	101	d
11001	1100	e
11010	11010	f
11011	11011	g
11101	111	h

(a)



(b)

Error Detection and Correction

- Bit errors can occur in packet
- Layer 2
  - Cyclic Redundancy Check (CRC)
- Layer 3
  - Header checksum
- Significant computation overhead
  - Layer 2 CRC done in hardware
  - Layer 3 checksum computed over packet header
- Error correction not done by network system – why?
  - More overhead
  - Error correction handled by upper layers

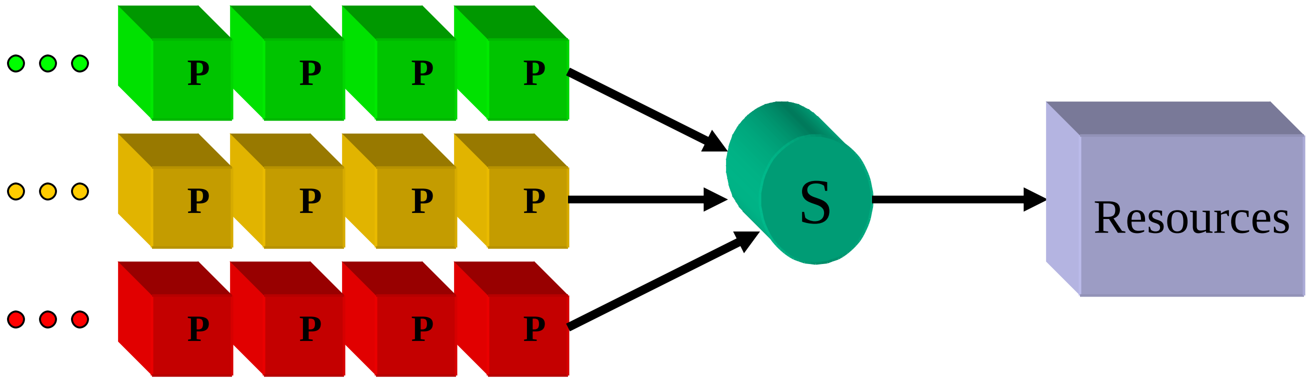
Fragmentation and Reassembly

- MTU
- IP fragments and reassembles
- ATM segments and reassembles
- Fragmentation straightforward
- Reassembly more complex – why?
  - Pieces of packet can arrive out of order
  - Pieces need to be buffered (chained buffer)
  - How much memory is needed?

Queuing

- Packet processing - store and forward
  - Incoming packet placed in queue
  - Outgoing packet placed in queue
- FIFO structure
  - How big?
  - How many queues?
  - Where to place them?
- How are packets selected from queues?
  - Priority mechanisms (a.k.a. scheduling)
- Packet discard
  - Finite queue size
  - Tail drop
  - Random early discard - probabilistic

Priority Mechanisms



- Priority Queuing
  - Starvation
- Weighted Round Robin
  - Number of packets processed from a queue depending on weight
  - Weight depends on priority and average packet size
  - Why could this be unfair?
- Weighted Fair Queuing
  - Use packet size rather than number of packets

Scheduling

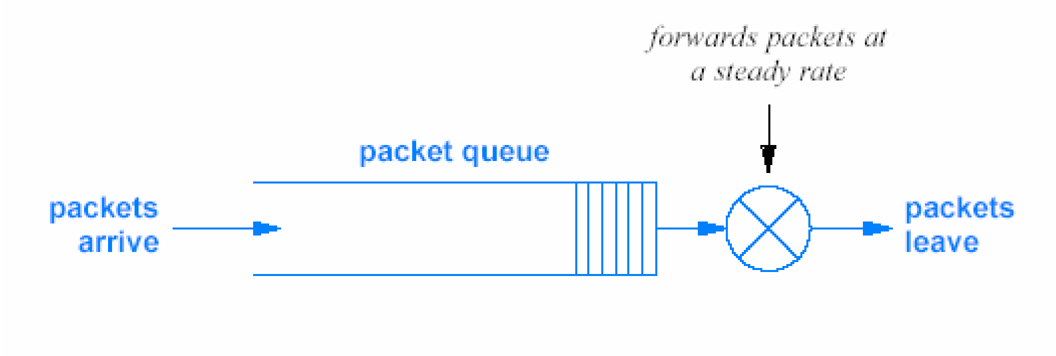
- Two types
  - Link (queue) scheduling
  - Resource scheduling
- Co-ordination of activities in network system
- Resource allocation
  - Process multiple packets
  - Process multiple protocols
  - Multiple processors
- Important when priorities are involved
- Scheduler must be fair

Security

- Authentication
- Privacy
  - VPN
- Encryption
  - Covers entire packet payload
  - Computationally intensive!
  - Performed by special hardware

Traffic Measurement, Shaping

- Traffic measurement
  - Examine header contents
  - Collect real time statistical information
- Traffic policing
  - Enforcement of QoS guarantee
  - Hard boundary - discard packet
- Traffic shaping
  - Softer form of policing
  - Does not discard packet
  - Smooth out bursty traffic
  - Leaky bucket, token bucket



Timer Management

- Fundamental function
- Timers used for
  - Protocols
    - ARP for retransmission and cache management
    - IP for re-assembly
    - TCP for retransmission
  - Scheduling
- Multiple independent timers required
  - Cost can be high
- How do we manage multiple timers with one clock?
  - Priority data structure
  - Granularity issues

Protocol Demultiplexing

- Differentiate between protocols at each layer of stack
- One protocol is used to process packet
- Example:
  - Layer 2 – Ethernet, ATM
  - Layer 3 – IP, ARP
- Use type information from header at each layer
- Layered processing

Packet Classification

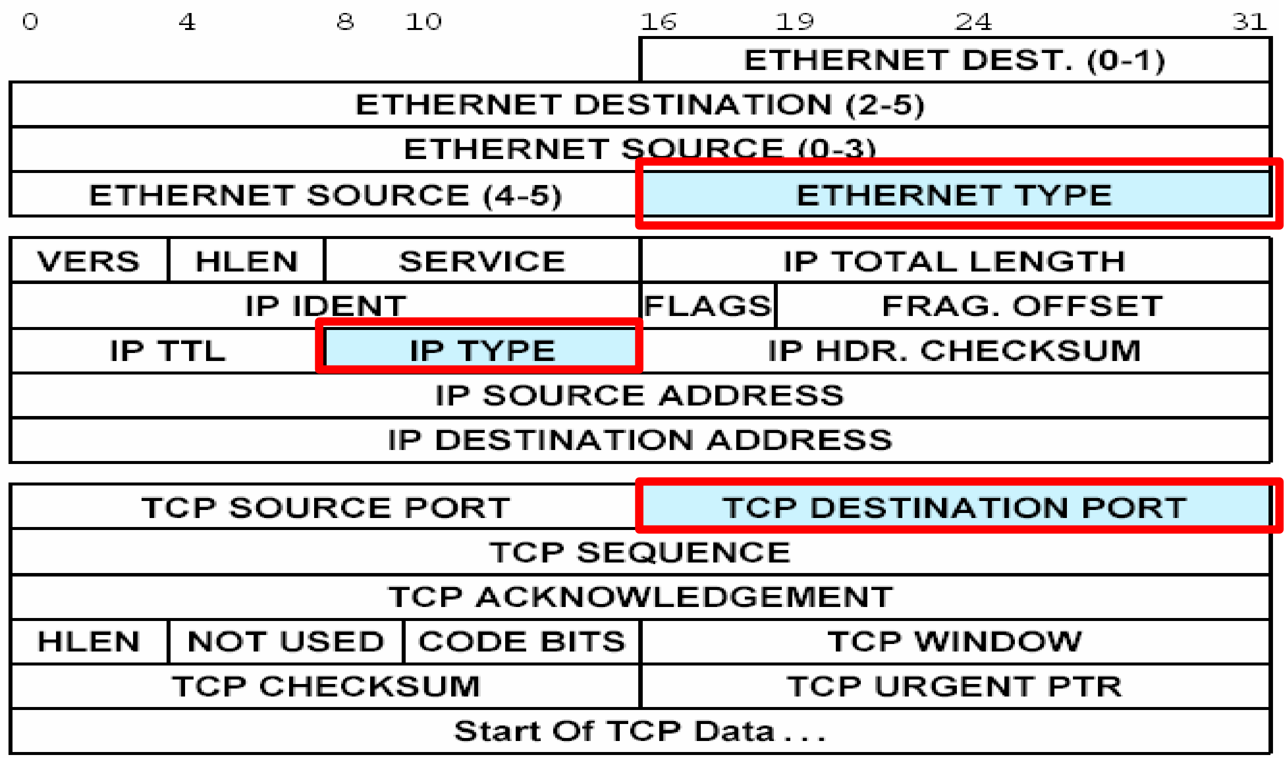
- Map packet into a “flow” or category depending on header information
- Flow – set of packets that share common characteristics
- Packet handled differently depending on flow
- Different from protocol demultiplexing
  - Maintains state information (flow table)
  - Packet classified over multiple layers
- **Rule** based

Packet Classification

- Software or hardware based methods
  - Software usually run on network processors
  - Software more flexible
  - Hardware better performance, more expensive
- Static vs. dynamic packet classification
  - Static : Header values determined *a priori*
  - Dynamic : Rules can change over time
  - Dynamic : Usually implemented in software

Example : Web Traffic

- Ethernet frame contains IP datagram
- IP datagram contains TCP segment
- TCP segment has destination port 80 (HTTP)



0 4 8 10 16 19 24 31

ETHERNET DEST. (0-1)

ETHERNET DESTINATION (2-5)

ETHERNET SOURCE (0-3)

ETHERNET SOURCE (4-5) | ETHERNET TYPE

VERS | HLEN | SERVICE | IP TOTAL LENGTH

IP IDENT | FLAGS | FRAG. OFFSET

IP TTL | IP TYPE | IP HDR. CHECKSUM

IP SOURCE ADDRESS

IP DESTINATION ADDRESS

TCP SOURCE PORT | TCP DESTINATION PORT

TCP SEQUENCE

TCP ACKNOWLEDGEMENT

HLEN | NOT USED | CODE BITS | TCP WINDOW

TCP CHECKSUM | TCP URGENT PTR

Start Of TCP Data . . .

Software Classification

- Three classification rules required

```
if ((frame type == 0x0800) && (IP type == 6) && (TCP port == 80))
    packet matched classification
else
    packet does not match classification
```

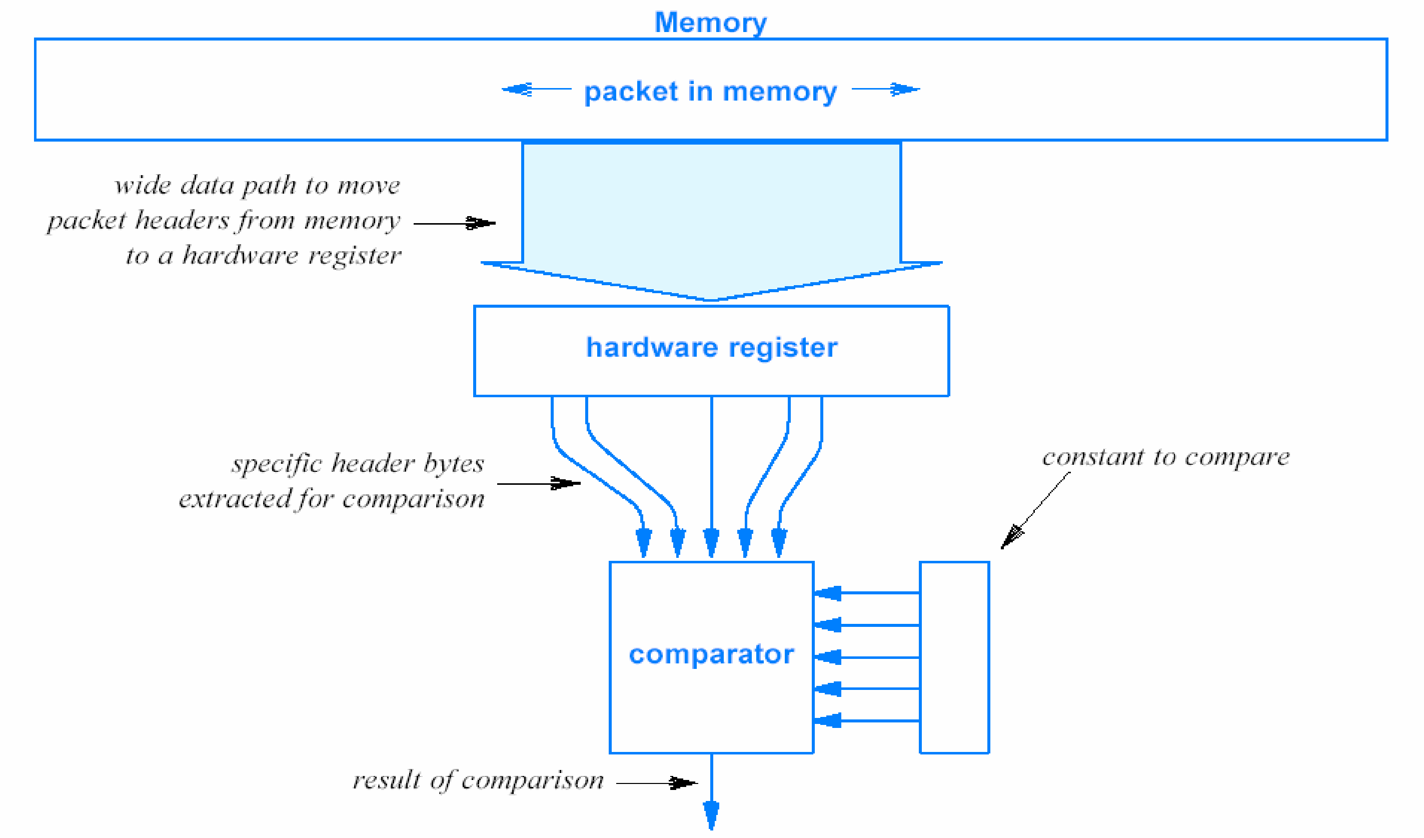
- Maximum number of comparisons is fixed
- Can be optimized by re-ordering comparisons

```
if ((TCP port == 80) && (IP type == 6) && (frame type == 0x0800))
    packet matched classification
else
    packet does not match classification
```

- Average number of comparisons determined by order of tests

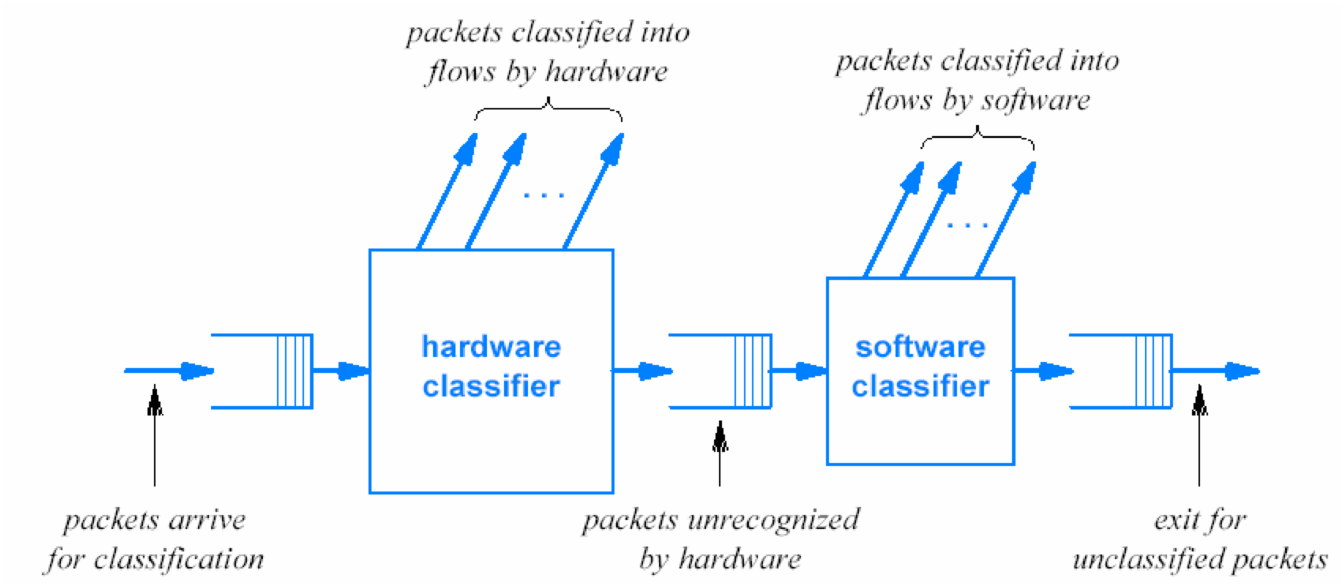
Hardware Classification

- Uses parallel hardware to extract required fields
- Example : need to compare 0x(0800060050)


Memory
packet in memory
wide data path to move packet headers from memory to a hardware register
hardware register
specific header bytes extracted for comparison
constant to compare
comparator
result of comparison

Special Packet Classification

- Can get complicated
  - Multiple rule sets
  - Variable size headers
- Hybrid classifiers



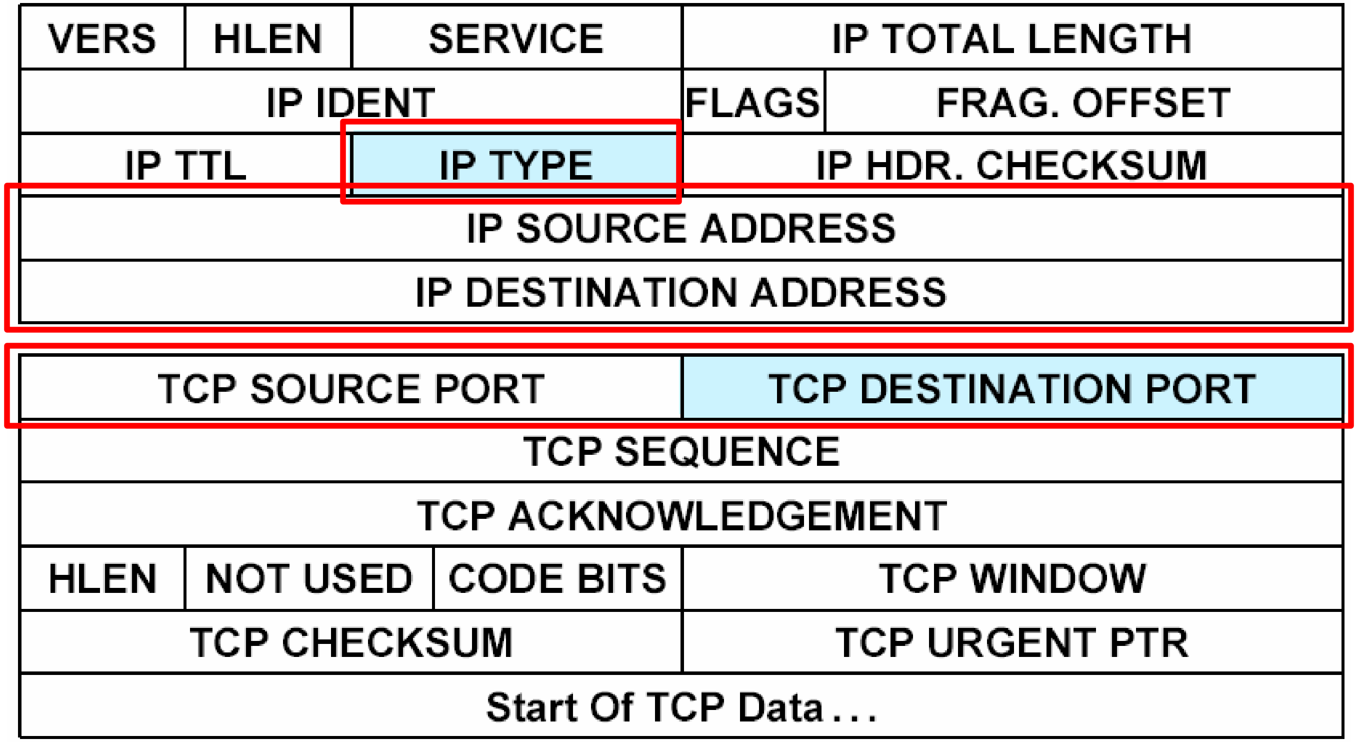
packets classified into flows by hardware
packets classified into flows by software
. . .
. . .
hardware classifier
software classifier
packets arrive for classification
packets unrecognized by hardware
exit for unclassified packets

Dynamic Classification

- Performed by software
  - Flexible
  - More processing overhead
- Flow creation
  - “n-tuple” → n fields from packet headers
  - TCP flags used to determine status of flow
- Flow table
  - Store flow record
  - Expensive operation to update flow record

Flow Creation

- 5-tuple
  - Most commonly used version



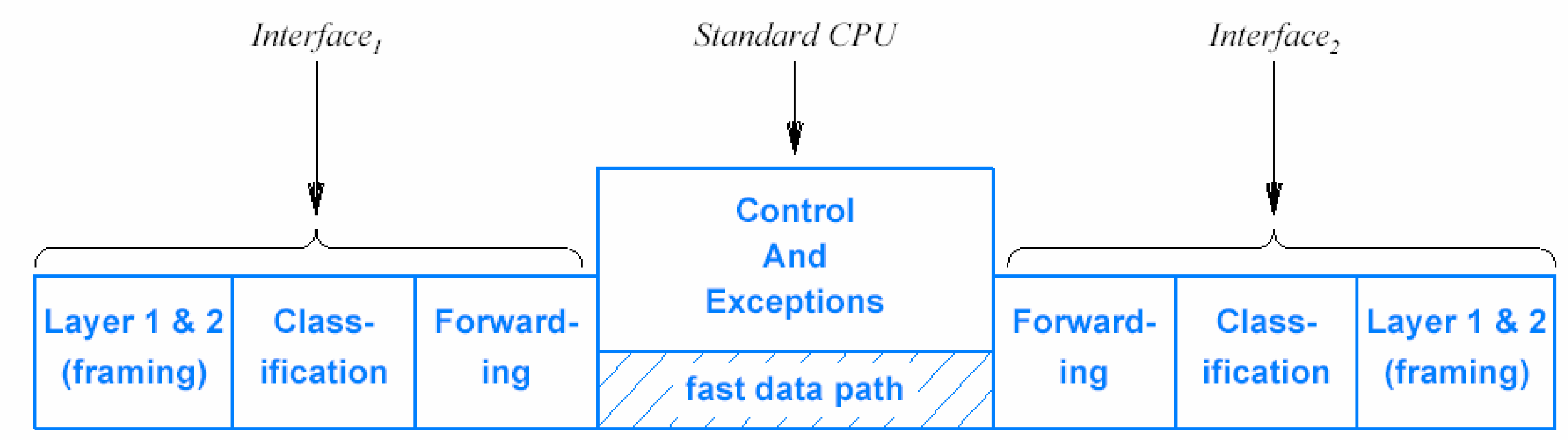
<table>
<tr><td>VERS</td><td>HLEN</td><td colspan="2">SERVICE</td><td colspan="2">IP TOTAL LENGTH</td></tr>
<tr><td colspan="4">IP IDENT</td><td>FLAGS</td><td>FRAG. OFFSET</td></tr>
<tr><td colspan="2">IP TTL</td><td colspan="2">IP TYPE</td><td colspan="2">IP HDR. CHECKSUM</td></tr>
<tr><td colspan="6">IP SOURCE ADDRESS</td></tr>
<tr><td colspan="6">IP DESTINATION ADDRESS</td></tr>
<tr><td colspan="4">TCP SOURCE PORT</td><td colspan="2">TCP DESTINATION PORT</td></tr>
<tr><td colspan="6">TCP SEQUENCE</td></tr>
<tr><td colspan="6">TCP ACKNOWLEDGEMENT</td></tr>
<tr><td>HLEN</td><td colspan="2">NOT USED</td><td>CODE BITS</td><td colspan="2">TCP WINDOW</td></tr>
<tr><td colspan="4">TCP CHECKSUM</td><td colspan="2">TCP URGENT PTR</td></tr>
<tr><td colspan="6">Start Of TCP Data . . .</td></tr>
</table>

Flow Forwarding

- Flow determines how to dispose packet
  - Classification : packet → flow
  - Forwarding : flow → next hop
- Create “route cache”
  - Stores next hop information for a flow
  - Provides next hop information
  - Avoid routing table lookup, more efficient
  - Drawback :
    - Route cache needs to be updated when routing table changes

Current Network Systems

- Features
  - Use of classification instead of demultiplexing
  - De-centralized architecture, interfaces forward packets
  - Fast data path, slow data path
- Conventional CPU to handle exceptions
- Scalability



Summary

- Overview of packet processing functionality
  - Table lookup
  - Classification
    - Dynamic (flow based) classification
  - Queuing/Scheduling
- Task level granularity
  - Building blocks
- Next class
  - Read IP lookup paper
  - Chapter 7