CSE398: Network Systems Design

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

March 14, 2005
Outline

- Classification & forwarding (Chapter 9)
- Switching fabrics (Chapter 10)
- Summary and homework
Recall: Packet Demultiplexing

- Used with layered protocols
- Packet proceeds through one layer at a time (inefficient)
  - On input, software in each layer chooses module at next higher layer
  - On output, type field in each header specifies encapsulation
  - Inefficient b/c sequential processing among layers
Packet Classification

- Alternative to demultiplexing for higher speed
- Considers all layers at the same time
- Linear in number of fields
- Two possible implementations
  - Software
  - Hardware
Example Classification

- Classify Ethernet frames carrying traffic to Web server
- Specify exact header contents in rule set
- Example
  - Ethernet type field specifies IP:
  - IP type field specifies TCP: 2-octet IP type is 6
  - TCP destination port specifies Web server: 2-octet TCP destination port is 80
Illustration of Encapsulated Headers

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>VERS</td>
<td>HLEN</td>
<td>SERVICE</td>
<td>IP TOTAL LENGTH</td>
<td>IP IDENT</td>
<td>FLAGS</td>
</tr>
<tr>
<td>IP TTL</td>
<td>IP TYPE</td>
<td>IP HDR. CHECKSUM</td>
<td>IP SOURCE ADDRESS</td>
<td>IP DESTINATION ADDRESS</td>
<td>IP HDR. CHECKSUM</td>
</tr>
<tr>
<td>TCP SOURCE PORT</td>
<td>TCP DESTINATION PORT</td>
<td>TCP SEQUENCE</td>
<td>TCP ACKNOWLEDGEMENT</td>
<td>TCP WINDOW</td>
<td>TCP URGENT PTR</td>
</tr>
<tr>
<td>HLEN</td>
<td>NOT USED</td>
<td>CODE BITS</td>
<td>TCP CHECKSUM</td>
<td>TCP CHECKSUM</td>
<td>Start Of TCP Data…</td>
</tr>
</tbody>
</table>

**Instructor:** Dr. Liang Cheng

**Course:** CSE398: Network Systems Design

**Date:** 03/14/05
Software Implementation of Classification

- Compare values in header fields
- Conceptually a logical and of all field comparisons

Example

```c
if ((frame type == 0x0800) && (IP type == 6) && (TCP port == 80))
    declare the packet matches the classification;
else
    declare the packet does not match the classification;
```

Optimization?
Optimizing Software Classification

- Comparisons performed sequentially ⇒ reorder comparisons to minimize effort
- Assume
  - 95.0% of all frames have frame type 0800\textsubscript{16}
  - 87.4% of all frames have IP type 6
  - 74.3% of all frames have TCP port 80
  - Values 6 and 80 do not occur in corresponding positions in non-IP packet headers
- Reordering tests can optimize processing time
  
  ```
  if ((TCP port == 80) && (IP type == 6) && (frame type == 0x0800))
    declare the packet matches the classification;
  else
    declare the packet does not match the classification;
  ```
- At each step, test the field that will eliminate the most packets
Hardware Implementation of Classification

- Can build special-purpose hardware
- Hardware can operate in parallel

- Extract needed fields
- Concatenate bits
- Place result in register
- Perform comparison
- Constant for Web classifier
  - 08.00.06.01.5016
Special Cases Of Classification

- Multiple categories
  - Classification usually involves multiple categories
  - Packets grouped together into flows
  - May have a default category
  - Each category specified with a rule set
- Variable-size headers
- Dynamic classification
Example Multi-Category Classification

- Flow 1: traffic destined for Web server
- Flow 2: traffic consisting of ICMP echo request packets
- Flow 3: all other traffic (default)
Rule Sets

- **Web server traffic**
  - 2-octet Ethernet type is 0800_{16}
  - 2-octet IP type is 6
  - 2-octet TCP destination port is 80

- **ICMP echo traffic**
  - 2-octet Ethernet type is 0800_{16}
  - 2-octet IP type is 1
  - 1-octet ICMP type is 8

```c
if (frame type != 0x0800) {
    send frame to flow 3;
} else if (IP type == 6 && TCP destination port == 80) {
    send packet to flow 1;
} else if (IP type == 1 && ICMP type == 8) {
    send packet to flow 2;
} else {
    send frame to flow 3;
}
```
Variable-Size Packet Headers

- Fields not at fixed offsets
  - Easily handled with software
- Finite cases can be specified in rules
- Each variable-size header adds one computation step
- In worst case, classification no faster than demultiplexing
Hybrid Classification

- Combines hardware and software mechanisms
  - Hardware used for standard cases
  - Software used for exceptions
Basic Types of Classification

- **Static**
  - Flows specified in rule sets
  - Header fields and values known a priori
  - Example: ?

- **Dynamic**
  - Flows created by observing packet stream
  - Values taken from headers
  - Allows fine-grain flows
  - Requires state information
  - Example: ?
Forwarding and Flow

- **Classification:** packet $\Rightarrow$ flow
  - Classification binding is usually 1-to-1

- **Forwarding**
  - Destination address $\Rightarrow$ packet disposition
  - Flow $\Rightarrow$ packet disposition
  - Forwarding binding can be 1-to-1 or many-to-1

- **Flow identification**
  - Fine-grain flow creation
Forwarding

- In a connection-oriented network
  - Flow identifiers ↔ connection identifiers
  - Efficient forwarding

- In a connectionless network
  - Route for flow determined when flow created
  - Indexing used in place of route lookup
  - Flow identifier corresponds to index of entry in forwarding cache
  - Forwarding cache must be changed when route changes
Second Generation Network Systems

- Designed for greater scale
  - Use classification instead of demultiplexing
- Decentralized architecture
  - Additional computational power on each NIC
  - NIC implements classification and forwarding
- High-speed internal interconnection mechanism
  - Interconnects NICs
  - Provides fast data path
Outline

- Classification & forwarding (Chapter 9)
- Switching fabrics (Chapter 10)
- Summary and homework
Interconnection

- Physical interconnection
  - Physical box with backplane
  - Individual blades plug into backplane slots
  - Each blade contains one or more network connections

- Logical interconnection
  - Known as *switching fabric*
  - Handles transport from one blade to another
  - Becomes bottleneck as number of interfaces scales
Illustration of Switching Fabric

- Any input port can send to any output port
Switching Fabric Properties

- Used inside a single network system
- Interconnection among I/O ports (and possibly CPU)
  - Can transfer unicast, multicast, and broadcast packets?
  - Scales to arbitrary data rate on any port?
  - Scales to arbitrary packet rate on any port?
  - Scales to arbitrary number of ports?
  - Has low overhead?
  - Has low cost?
Types of Switching Fabrics

- Space-division (separate paths)
- Time-division (shared medium)
Space-Division Fabric (separate paths)

- Can use multiple paths simultaneously
- Still have port contention
Goals and Solutions

- High speed and low cost
- Separation of physical paths
- Less parallel hardware
- Crossbar design
Space-Division (Crossbar Fabric)

- Allows simultaneous transfer on disjoint pairs of ports
- Can still have port contention
Crossbar Fabric with Queuing

- Solving contention
- Queues (FIFOs)
  - Attached to input
  - Attached to output
  - At intermediate points
Time-Division Fabric (shared bus)

- Chief advantage: low cost
- Chief disadvantage: low speed
Time-Division Fabric (shared memory)

- May be better than shared bus
- Usually more expensive
Multi-Stage Fabrics

- Compromise between pure time-division and pure space-division
- Attempt to combine advantages of each
  - Lower cost from time-division
  - Higher performance from space-division
- Technique: limited sharing
Banyan Fabric

- Example of multi-stage fabric

- Features
  - Scalable
  - Self-routing
  - Packet queues allowed, but not required

- Basic building block
  - Address added to front of each packet
  - One bit of address used to select output
4-Input and 8-Input Banyan Switches
Outline

- Classification & forwarding (Chapter 9)
- Switching fabrics (Chapter 10)
- Summary and homework
  - Classification & forwarding
  - Switching fabric
    - Two basic approaches
      - Time-division has lowest cost
      - Space-division has highest performance
    - Multistage designs compromise between two
      - Banyan fabric
Homework (due on 03/21)

8.1. (a) Problem 2 of Chapter 8 (Page 115); (b) List two protocols that uses raw sockets for their implementation.

8.2. Problem 9 of Chapter 10 (page 157; also refer to the graph in the slide#32)

Question that does not need to be handed in: How Banyan fabric compromises between the time-division and space-division fabric designs?