CS 455: INTRODUCTION TO DISTRIBUTED SYSTEMS [NETWORKING]

Shrideep Pallickara
Computer Science
Colorado State University

Frequently asked questions from the previous class survey

Topics covered in today’s lecture

- OSI
- Internet Architecture
- IP routing

OSI network architecture

- Model is a product of the Open Systems Interconnection (OSI) project
- At the International Organization for Standardization (ISO)
- Partitions network functionality into 7 layers
- Physical
  - Handles transmission of raw bits
  - Standardizes electrical, mechanical, and signaling interfaces
- 0 bit should be received as 0 not 1

OSI network architecture: Data link

- Collects stream of bits into a frame
  - Puts special bit pattern at the start/end of each frame
  - Frames, not raw bits, are delivered to host
- Compute checksum for frame
  - Check for correctness and request retransmission
- Network adaptors & device drivers implement this
OSI network architecture

- **Network layer**
  - Handles routing among nodes in a **packet-switched** network
  - Unit of data exchanged is **packet** not frames

- **Layers implemented on all network nodes?**
  - Physical, data and network

---

**OSI Architecture**

Usually run only on the end host, not switches

One or more nodes within the network

---

**How messages flowing through the OSI stack will appear on the network**

- Data link layer header
- Network layer header
- Transport layer header
- Session layer header
- Presentation layer header
- Application layer header
- Data link layer trailer

---

**OSI network architecture**

- **Transport**
  - Implements process-process **channel**
  - **Messages** (not packet or frame)

- **Presentation**
  - **Format** of data exchanged between peers

- **Session**
  - **Namespace** to tie different transport-streams that are part of the application

---

**Internet architecture**

- Evolved out of experiences with ARPANET
- Funded by ARPA of the US DoD
- Around before the OSI architecture
- Unlike OSI, this is a **4-level** model
Internet protocol graph

- FTP
- HTTP
- NV
- TFTP
- TCP
- UDP
- IP
- NETn...

End-to-End protocols

Internet architecture

- DOES NOT imply strict layering
  - Bypassing immediate lower layers is possible
- Layer has an hour-glass shape
  - Wide at top and bottom
  - Narrow in the middle
  - IP is the focal point of the architecture

Protocol implementation issues

Where are the processes?

- Process-per-protocol
- Process-per-message

Process-per-message model: Associate processes per message

- Treat each protocol as a static piece of code
- Protocol graph traversed in sequence of procedure calls
- When message arrives:
  - Dispatch process to move message up the protocol graph
  - At each level procedure implementing protocol is invoked
- Sending message?
  - Application process invokes appropriate procedures
Comparison

- Process-per-protocol
  - Context switch per level
- Process-per-message
  - Procedure call per level

INTERNETWORKING

Internetwork

- Arbitrary collection of interconnected networks
  - To provide some sort of host-host packet delivery service
- Network of networks
  - Made up of lots of smaller networks

Internet Protocol (IP)

- Key tool to build scalable, heterogeneous networks
- Runs on all nodes (hosts and routers)
- Allows nodes and networks to function as a single logical network
- Possible to build an internetwork without IP
  - But IP is the only one that has faced scale issues
Example depicting how hosts (H1-H8) are logically connected

The IP service model
- Datagram model of delivery
  - Connectionless
  - Best effort
- Addressing scheme
  - Identifies all hosts in the internetwork

Datagram delivery
- Datagram is a type of packet
  - Sent in a connectionless fashion
- No need for any advance setup mechanisms
  - That tell network what to do when packet arrives
- Every datagram contains enough information
  - To forward packet to correct destination

The network makes a best effort to send datagrams across
- Things that could go wrong with the packets
  - Lost
  - Corrupted
  - Misdelivered
  - Out of order and duplicates
- When things go wrong, the network does nothing
  - No attempt to recover from the failure

Keeping routers simple was one of the original design goals of IP
- Important to run over anything
- Putting extra functionality into routers to make up for network deficiencies?
  - Not a good idea
- Higher-level protocols/apps that run above IP need to be aware of failure modes

The IP Packet format consists of a header followed by bytes of data
- Represented as a succession of 32-bit words
- Packet formats designed to align on 32-bit boundaries
  - Simplifies task of processing in software
- Transmission order
  - Top word transmitted first
  - Leftmost byte of each word transmitted first
The IPv4 packet header

<table>
<thead>
<tr>
<th>0</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>19</th>
<th>21</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version HLen TOS Length</td>
<td>Ident Flags Offset</td>
<td>TTL Protocol Checksum</td>
<td>SourceAddr DestinationAddr</td>
<td>Options (variable) Pad (variable) Data</td>
<td></td>
</tr>
</tbody>
</table>

IP Packet format

- **Version**
  - Makes it easy to redefine packet format later on
- **HLen**
  - Specifies length of header in 32-bit words
  - When there are no options (most of the time)
    - Header is 5 words or 20 bytes
- **TOS (type of service)**
  - Allow packets to be treated differently
  - Based on application needs

- **Length**
  - Length of the datagram in bytes
  - Maximum size of IP datagram is $2^{16}$ bytes
- **SECOND WORD OF IP PACKET**
  - Ident, Flags, Offset
  - Information about fragmentation

- **TTL (time to live)**
  - Hop-count not timer (as originally intended)
- **Protocol field**
  - Demultiplexing key
    - Identifies higher-level protocol
    - TCP (6), UDP (17)
- **Checksum**
  - Consider IP header as a sequence of 16-bit words

- **SourceAddr**
  - Decide if packet should be accepted
  - Also used for replies
- **DestinationAddr**
  - Full address of destination
  - Forwarding decisions are made at each router
- **Presence or absence of options**
  - Can be checked based on size of HLen

- **TOS field (Type of Service)**
  - Meant to specify how the datagram should be handled as it traversed the internet
    - Preference for low delay
    - Preference for high reliability
  - In practice TOS was not widely implemented
The 8 bits allocated to TOS can be divided into 5 parts:

- **Precedence bits**: Indicates importance of datagram.
  - Low delay
  - High throughput
  - High reliability

- **Unused**: Most Significant Bit (D T R Unused)

Providing host-to-host service model over heterogeneous collection of networks:

- Each network technology has its own idea of how large a packet can be:
  - Ethernet: 1500 bytes
  - FDDI: 4500 bytes

Every network type has a Maximum Transmission Unit (MTU):

- Largest IP datagram that it can carry in its frame
- Smaller than the largest packet-size of network
- IP datagram needs to fit in payload of link-layer frame

Fragmentation necessary when datagram path includes network with smaller MTU:

- All fragments carry same identifier in **Ident** field
  - To enable fragment reassembly
  - Chosen by the source host
- If all fragments do not arrive at receiving host?
  1. Receiver gives up reassembly
  2. **Discards** fragments that did arrive
- IP **does not attempt** to recover from missing fragments

A simple internetwork: Sending IP datagrams from H1 to H8

- Network 1 (Ethernet)
- Network 2 (Ethernet)
- Network 3 (FDDI)
- Network 4 (point-to-point)

IP datagrams traversing a sequence of physical networks:

- H1
- R1
- R2
- R3
- H8
**IPv4 Packet header**

<table>
<thead>
<tr>
<th>Field</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>4</td>
</tr>
<tr>
<td>HLen</td>
<td>4</td>
</tr>
<tr>
<td>TOS</td>
<td>1</td>
</tr>
<tr>
<td>Flags</td>
<td>1</td>
</tr>
<tr>
<td>Offset</td>
<td>1</td>
</tr>
<tr>
<td>TTL</td>
<td>8</td>
</tr>
<tr>
<td>Protocol</td>
<td>1</td>
</tr>
<tr>
<td>Checksum</td>
<td>4</td>
</tr>
<tr>
<td>Source Addr</td>
<td>16</td>
</tr>
<tr>
<td>Destination Addr</td>
<td>16</td>
</tr>
<tr>
<td>Options (variable)</td>
<td>variable</td>
</tr>
<tr>
<td>Pad (variable)</td>
<td>variable</td>
</tr>
<tr>
<td>Data</td>
<td>variable</td>
</tr>
</tbody>
</table>

**Header fields used in IP fragmentation:**

Fragmentation occurs at 8-byte boundaries

- **Start of header**
  - Ident = x
  - Offset = 64
  - Rest of header
  - 512 data bytes

- **Start of header**
  - Ident = x
  - Offset = 128
  - Rest of header
  - 376 data bytes

**DATAGRAM FORWARDING**

Datagram forwarding in IP:
- Datagrams contains IP address of destination
  - Network part uniquely identifies a single physical network
  - Hosts/routers that share the network part
    - Connected to same physical network
  - Every physical network has a router
    - Connected to at least one other physical network

A simple internetwork:
- Forwarding table for router R2

```
Network 1 (Ethernet)
Network 2 (Ethernet)
Network 3 (FDDI)
Network 4 (point-to-point)
```

- Network 1: H1, H2, H3
- Network 2: R1, H7, B3, H8
- Network 3: R2
- Network 4: H4, H5, H6

**IPv4 Packet header**

<table>
<thead>
<tr>
<th>Field</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>4</td>
</tr>
<tr>
<td>HLen</td>
<td>4</td>
</tr>
<tr>
<td>TOS</td>
<td>1</td>
</tr>
<tr>
<td>Flags</td>
<td>1</td>
</tr>
<tr>
<td>Offset</td>
<td>1</td>
</tr>
<tr>
<td>TTL</td>
<td>8</td>
</tr>
<tr>
<td>Protocol</td>
<td>1</td>
</tr>
<tr>
<td>Checksum</td>
<td>4</td>
</tr>
<tr>
<td>Source Addr</td>
<td>16</td>
</tr>
<tr>
<td>Destination Addr</td>
<td>16</td>
</tr>
<tr>
<td>Options (variable)</td>
<td>variable</td>
</tr>
<tr>
<td>Pad (variable)</td>
<td>variable</td>
</tr>
<tr>
<td>Data</td>
<td>variable</td>
</tr>
</tbody>
</table>
Example forwarding table:

For Router R2

<table>
<thead>
<tr>
<th>Network Num</th>
<th>Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R3</td>
</tr>
<tr>
<td>2</td>
<td>R1</td>
</tr>
</tbody>
</table>

Error Reporting in IP communications

- IP drops datagrams when the going gets tough
  - But does not fail silently
- IP always configured with a companion protocol
  - Internet Control Message Protocol (ICMP)

ICMP defines a collection of error messages
- When router/host is unable to process datagram successfully
  - ICMP error message sent back to source
- Examples
  - Destination host is unreachable
  - Reassembly process failed
  - TTL reached 0
  - IP header checksum failed

ICMP also defines some control messages
- Router sends control messages back to host
  - Example: ICMP-Redirect tells that there is a better route to destination
    - Network has two routers R1 and R2 and host uses R1 as default
    - When R1 receives a datagram and it knows R2 is a better choice
      - Send ICMP-Redirect to host
      - Host then uses R2 for future datagrams to that host

The contents of this slide-set are based on the following references