IP - The Internet Protocol
ICMP - Internet Control Message Protocol

* IP Service
* Datagram Format

* Configuring a network interface
* ICMP

Orientation

* IP (Internet Protocol) is a Network Layer Protocol.

* IP’s current version is Version 4 (IPv4). It is specified in RFC 891.
Orientation

- IP is the highest layer protocol which is implemented at both routers and hosts:

IP Service

- IP provides an unreliable and connectionless service (“datagram service”).
  - **Unreliable:** IP does not guarantee that a transmitted packet will be delivered.
  - **Connectionless:** Each packet (“datagram”) is handled independently. IP is not aware that packets between hosts may be sent in a logical sequence.

- Consequences of an unreliable, connectionless service.
  - Lost packets
  - Packets are delivered out-of-sequence
  - Duplicate packets
IP Service

- IP offers a best effort service; i.e., IP does not make performance guarantees on:
  - the time until a packet is delivered
  - the packet loss rate
  - the throughput of traffic between two hosts
  - etc.

- IP provides limited support for indicating the wish to receive a certain service. This is done via the Type-of-Service (TOS) field.

- IP supports the following services:
  - one-to-one (unicast)
  - one-to-all (broadcast)
  - one-to-several (multicast)

- IP multicast also supports a many-to-many service.
- IP multicast requires support of other protocols (IGMP, multicast routing)
IP Datagram Format

- 20 bytes <= Header Size <= 2^4 * 32 bit-words = 60 bytes
- 20 bytes <= Total Length <= 2^16 bytes = 65536 bytes

<table>
<thead>
<tr>
<th>bit #</th>
<th>0</th>
<th>7</th>
<th>8</th>
<th>15</th>
<th>16</th>
<th>23</th>
<th>24</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td>version (4 bits)</td>
<td>header length</td>
<td>Type of Service/TOS (8 bits)</td>
<td>Total Length (in bytes) (16 bits)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Identification (16 bits)</td>
<td>flags (3 bits)</td>
<td>Fragment Offset (13 bits)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TTL Time-to-Live (8 bits)</td>
<td>Protocol (8 bits)</td>
<td>Header Checksum (16 bits)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source IP address (32 bits)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Destination IP address (32 bits)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Options (if any, &lt;40 bytes)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Question: If you measure an IP datagram on the network, in which order are the fields transmitted?

Answer:
- Transmission is row by row
- For each row:
  1. Transmit bits 0-7
  2. Transmit bits 8-15
  3. Transmit bits 16-23
  4. Transmit bits 24-31
- This is called network byte order or big endian byte ordering.
- Note: some computers store 32-bit words in little endian format. Which ones?
Fields of the IP Header

- **Version**: current version is 4, next version will be 6.
- **Type of Service (TOS)**: contains 4 TOS bits, where each bit indicates a desired service.
  - minimize delay
  - maximize throughput
  - maximize reliability
  - minimize cost
  - Only one bit can be set! Not supported in all applications.
- **Identification**: unique identification of a datagram from a host. Incremented whenever a datagram is transmitted.
- **Time To Live (TTL)**: specifies longest paths before datagram is dropped.

Fields of the IP Header

- **Protocol**: Specifies the higher-layer protocol. Used for demultiplexing to higher layers.

- **Header checksum**: verifies correctness of header.
Fields of the IP Header

- **Options:**
  - Security restrictions
  - Record Route: each router that processes the packet adds its IP address to the header.
  - Timestamp: each router that processes the packet adds its IP address and time to the header.
  - (loose) Source Routing: specifies a list of routers that must be traversed.
  - (strict) Source Routing: specifies a list of the only routers that can be traversed.
- **Padding:** ensures that header ends on a 4-byte boundary

Maximum Transmission Unit

- The maximum frame size of the data link protocol translates itself to a limit on the size of the IP datagram that can be encapsulated.
- This limit is called **maximum transmission unit (MTU)**
- MTUs for various data link layers:
  - Ethernet: 1500
  - FDDI: 4352
  - 802.3: 1492
  - ATM AAL5: 9180
  - 802.5: 4464
  - PPP: 296

- What if the size of an IP datagram exceeds the MTU?
  - IP datagram is fragmented into smaller units.
- What if the route contains networks with different MTUs?
IP Fragmentation

- Host A sends a large IP datagram to Host B.
- Any Problem with that?

MTUs:  
- FDDI: 4352  
- Ethernet: 1500

- IP router splits the datagram into several (=Fragmentation)

Where is Fragmentation done?

- Fragmentation can be done at the sender or at intermediate routers
- The same datagram can be fragmented several times.
- Reassembly of original datagram is only done at destination hosts!!
What’s involved in Fragmentation?

- The following fields in the IP header are involved:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>version (4 bits)</td>
<td></td>
</tr>
<tr>
<td>header length</td>
<td></td>
</tr>
<tr>
<td>Type of Service/TOS</td>
<td></td>
</tr>
<tr>
<td>Total Length (in bytes)</td>
<td></td>
</tr>
<tr>
<td>Identification</td>
<td>is the same in all fragments</td>
</tr>
<tr>
<td>Flags</td>
<td>contains a “more fragments” bit</td>
</tr>
<tr>
<td>Fragment Offset</td>
<td>(There is also a “don’t fragment bit” that can be set).</td>
</tr>
<tr>
<td>TTL Time-to-Live (8 bits)</td>
<td></td>
</tr>
<tr>
<td>Protocol (8 bits)</td>
<td></td>
</tr>
<tr>
<td>Header Checksum (16 bits)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Identification of the datagram is changed by fragmentation.

ICMP - Internet Control Message Protocol (ICMP)
The IP (Internet Protocol) relies on several other protocols to perform necessary control and routing functions.

**Orientation**

- ICMP
- IGMP
- EGP
- RIP
- BGP
- OSPF

**ICMP**

- The Internet Control Message Protocol (ICMP) is the protocol used for error and control messages in the Internet
- ICMP provides an error reporting mechanism of routers to the sources
- All ICMP packets are encapsulated as IP datagrams
- The packet format is simple:

```plaintext
<table>
<thead>
<tr>
<th>Type (8 bits)</th>
<th>Code (8 bits)</th>
<th>Checksum (16 bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(additional information dependent on Type and Code)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

![32-bit word](32-bit-word)
Types of ICMP Packets

• Many ICMP packet types exist, each with its own format
• A Selection:

<table>
<thead>
<tr>
<th>Type Field</th>
<th>Message Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Echo Reply</td>
</tr>
<tr>
<td>3</td>
<td>Destination Unreachable</td>
</tr>
<tr>
<td>4</td>
<td>Source Quench</td>
</tr>
<tr>
<td>5</td>
<td>Redirect (Change Route)</td>
</tr>
<tr>
<td>8</td>
<td>Echo Request</td>
</tr>
<tr>
<td>11</td>
<td>Time Exceeded</td>
</tr>
<tr>
<td>12</td>
<td>Parameter Problem in Datagram</td>
</tr>
<tr>
<td>14</td>
<td>Timestamp Request</td>
</tr>
</tbody>
</table>

ICMP Message Types

• ICMP messages are either query messages or error messages

• **ICMP query messages:**
  • Echo request / Echo reply
  • Router advertisement / Router solicitation
  • Timestamp request / Timestamp reply
  • Address mask request / Address mask reply

• **ICMP error messages:**
  • Host unreachable
  • Source quench
  • Time Exceeded
  • Parameter Problem
ICMP Error Messages

- Each ICMP error message contains the header and at least the first 8 bytes of the IP datagram that triggered the error message.
- **Problem:** How to prevent that too many ICMP messages are sent?
  (e.g., an ICMP packet could trigger an ICMP packet, which triggers …).
- **ICMP error messages are not sent …**
  … for multiple fragments of the same IP datagrams
  … in response to an error message
  … in response to a broadcast packet
  … etc.

Example of a Query: ICMP Timestamp

- A system (host or router) asks another system for the current time.
- Time is measured in milliseconds after midnight UTC (Universal Coordinated Time) of the current day
- Sender sends a request, receiver responds with reply

<table>
<thead>
<tr>
<th>Type (17 or 18)</th>
<th>Code (0)</th>
<th>Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32-bit sender timestamp</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32-bit receive timestamp</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32-bit transmit timestamp</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example of an Error Message: Port Unreachable

- There are 16 different ICMP error messages (‘codes’) of type “Host Unreachable” (Code = 3)

<table>
<thead>
<tr>
<th>Code</th>
<th>Message Type</th>
<th>Code</th>
<th>Message Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Network unreachable</td>
<td>9</td>
<td>Destination network administratively prohibited</td>
</tr>
<tr>
<td>1</td>
<td>Host unreachable</td>
<td>10</td>
<td>Destination host administratively prohibited</td>
</tr>
<tr>
<td>2</td>
<td>Protocol unreachable</td>
<td>11</td>
<td>Network unreachable for TOS</td>
</tr>
<tr>
<td>3</td>
<td>Port unreachable</td>
<td>12</td>
<td>Host unreachable for TOS</td>
</tr>
<tr>
<td>4</td>
<td>Fragmentation needed but bit not set</td>
<td>13</td>
<td>Communication administratively prohibited by filtering</td>
</tr>
<tr>
<td>5</td>
<td>Source route failed</td>
<td>14</td>
<td>host precedence violation</td>
</tr>
<tr>
<td>6</td>
<td>Destination network unknown</td>
<td>15</td>
<td>precedence cutoff in effect</td>
</tr>
<tr>
<td>7</td>
<td>Destination node unknown</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Source host isolated</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ICMP Port Unreachable

- RFC 792: If, in the destination host, the IP module cannot deliver the datagram because the indicated protocol module or process port is not active, the destination host may send a destination unreachable message to the source host.

- Scenario:
ICMP Port Unreachable

- Format of the Port Unreachable Message

<table>
<thead>
<tr>
<th>Ethernet Header</th>
<th>IP header</th>
<th>ICMP header</th>
<th>IP header of datagram from client</th>
<th>at least 8 bytes from IP datagram of client</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type (= 3)</td>
<td>Code (=0-15)</td>
<td>Checksum</td>
<td>Unused (Set to 00...0)</td>
<td></td>
</tr>
</tbody>
</table>

Note: Both UDP and TCP store the Port Number in the first 8 byte!

The PING program

- PING (=Packet IntetNet Groper) is a program that utilizes the ICMP echo request and echo reply messages.
- PING is used to verify if a certain host is up and running. It is used extensively for fault isolation in IP networks.
- PING can be used with a wide variety of options, e.g., :

- f Flood ping. Outputs packets as fast as they come back or one hundred times per second, whichever is more.
- R Record route. Includes the RECORD_ROUTE option in the ECHO_REQUEST packet and displays the route buffer on returned packets.
- s Packetsize Specifies the number of data bytes to be sent (Default is 56)
Running Ping

aida:~ ; ping mng.poly.edu
PING mng.poly.edu (128.238.42.105): 56 data bytes
64 bytes from 128.238.42.105: icmp_seq=0 ttl=128 time=0.718 ms
64 bytes from 128.238.42.105: icmp_seq=1 ttl=128 time=3.408 ms
64 bytes from 128.238.42.105: icmp_seq=2 ttl=128 time=3.171 ms
64 bytes from 128.238.42.105: icmp_seq=3 ttl=128 time=0.701 ms
64 bytes from 128.238.42.105: icmp_seq=4 ttl=128 time=0.693 ms
64 bytes from 128.238.42.105: icmp_seq=5 ttl=128 time=1.528 ms
64 bytes from 128.238.42.105: icmp_seq=6 ttl=128 time=0.689 ms
64 bytes from 128.238.42.105: icmp_seq=7 ttl=128 time=3.077 ms
^C
--- mng.poly.edu ping statistics ---
8 packets transmitted, 8 packets received, 0% packet loss
round-trip min/avg/max = 0.689/1.748/3.408 ms

Running Ping to a different machine

aida:~ ; ping www.cologne.de
PING fileserv1.cologne.de (194.94.233.1): 56 data bytes
64 bytes from 194.94.233.1: icmp_seq=0 ttl=240 time=447.080 ms
64 bytes from 194.94.233.1: icmp_seq=1 ttl=240 time=368.383 ms
64 bytes from 194.94.233.1: icmp_seq=2 ttl=240 time=353.992 ms
64 bytes from 194.94.233.1: icmp_seq=3 ttl=240 time=323.380 ms
64 bytes from 194.94.233.1: icmp_seq=4 ttl=240 time=353.782 ms
64 bytes from 194.94.233.1: icmp_seq=5 ttl=240 time=326.356 ms
^C
--- fileserv1.cologne.de ping statistics ---
7 packets transmitted, 6 packets received, 14% packet loss
round-trip min/avg/max = 323.380/362.162/447.080
Echo Request and Reply

- PING’s are handled directly by the kernel
- Each Ping is translated into an ICMP Echo Request
- The Ping’ed host responds with an ICMP Echo Reply

Format of Echo Request and Reply

<table>
<thead>
<tr>
<th></th>
<th>Type (0 or 8)</th>
<th>Code (0)</th>
<th>Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identifier</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sequence number</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>optional</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Identifier is set to process Id of querying process
- Sequence number is incremented for each returning packet
Unix Tools for Configuring Networks

**netstat** - show the network status

**ifconfig** - configure network interface parameters

Display Network Status

- The **netstat** command plots (a lot of) network related data structures in the operating system.

  - **netstat -i** plots information on the network interface

<table>
<thead>
<tr>
<th>Name</th>
<th>Mtu</th>
<th>Network</th>
<th>Address</th>
<th>Ipkts</th>
<th>Ierrs</th>
<th>Opkts</th>
<th>Oerrs</th>
<th>Coll</th>
</tr>
</thead>
<tbody>
<tr>
<td>ed1</td>
<td>1500</td>
<td>128.238.42/24</td>
<td>aida</td>
<td>2394482</td>
<td>35664</td>
<td>346880</td>
<td>223</td>
<td>119591</td>
</tr>
<tr>
<td>lo0</td>
<td>16384</td>
<td>your-net</td>
<td>localhost</td>
<td>153</td>
<td>0</td>
<td>153</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- **netstat -rn** displays the IP routing table
- **netstat -a** displays information on TCP and UDP ports
- **netstat -s** displays operational statistics of various networking protocols
Configuring a Network Interface

- The `ifconfig` command is used for setting and testing network interfaces (in Unix systems) that run TCP/IP.
- `ifconfig -a` plots a report on all interfaces.

```
: aida:/ ; ifconfig -a
ed1: flags=8863<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST> mtu 1500
     inet 128.238.42.114 netmask 0xffffff00 broadcast 128.238.42.255
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST>    mtu 16384
     inet 127.0.0.1 netmask 0xff000000
```

- `ifconfig ed1 down` Disables interface ed1
- `ifconfig ed1 192.0.1.8 up` Assigns to interface ed1 the IP address 192.0.1.8