Orientation

- IP (Internet Protocol) is a Network Layer Protocol.
IP: The waist of the hourglass

- IP is the waist of the hourglass of the Internet protocol architecture
- Multiple higher-layer protocols
- Multiple lower-layer protocols

Network Layer Protocol

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

- Delivery service of IP is minimal

- IP provide provides an unreliable connectionless best effort service (also called: “datagram service”).
  - **Unreliable**: IP does not make an attempt to recover lost packets
  - **Connectionless**: Each packet (“datagram”) is handled independently. IP is not aware that packets between hosts may be sent in a logical sequence
  - **Best effort**: IP does not make guarantees on the service (no throughput guarantee, no delay guarantee,…)

- Consequences:
  - Higher layer protocols have to deal with losses or with duplicate packets
  - Packets may be delivered out-of-sequence

IP Service

- 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 Addresses

- IP is a network layer - it must be capable of providing communication between hosts on different kinds of networks (different data-link implementations).
- The address must include information about what network the receiving host is on. This is what makes routing feasible.

IP Addresses

- IP addresses are *logical* addresses (not physical)
- 32 bits.
- Includes a network ID and a host ID.
- Every host must have a unique IP address.
- IP addresses are assigned by ICANN (*Internet Corporation for Assigned Names and Numbers*).
IP Address as a 32-Bit Binary Number

Octet (8 bits)

<table>
<thead>
<tr>
<th>NETWORK</th>
<th>HOST</th>
</tr>
</thead>
<tbody>
<tr>
<td>172</td>
<td>.</td>
</tr>
<tr>
<td>16</td>
<td>.</td>
</tr>
<tr>
<td>122</td>
<td>.</td>
</tr>
<tr>
<td>204</td>
<td></td>
</tr>
</tbody>
</table>

8 Bits
1 Byte

Binary and Decimal Conversion

<table>
<thead>
<tr>
<th>2^7</th>
<th>2^6</th>
<th>2^5</th>
<th>2^4</th>
<th>2^3</th>
<th>2^2</th>
<th>2^1</th>
<th>2^0</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

192.57.30.224

11000000.00111001.00011110.11100000
### Classes of Network IP Addresses

<table>
<thead>
<tr>
<th>Class</th>
<th>Network Bits</th>
<th>Host Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>Class B</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>Class C</td>
<td>8</td>
<td></td>
</tr>
</tbody>
</table>

### IP Addresses as Decimal Numbers

<table>
<thead>
<tr>
<th># Bits</th>
<th>Class A</th>
<th>Class B</th>
<th>Class C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>NETWORK#</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>24</td>
<td>HOST#</td>
<td>HOST#</td>
<td>HOST#</td>
</tr>
</tbody>
</table>
An IP address such as 176.10.0.0 that has all binary 0s in the host bit positions is reserved for the network address.

<table>
<thead>
<tr>
<th>NETWORK</th>
<th>HOST</th>
</tr>
</thead>
<tbody>
<tr>
<td>176</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>10</td>
</tr>
<tr>
<td>.</td>
<td>0</td>
</tr>
<tr>
<td>.</td>
<td>0</td>
</tr>
</tbody>
</table>

An IP address such as 176.10.255.255 that has all binary 1s in the host bit positions is reserved for the broadcast address.

Private Addresses

The following ranges are available for private addressing

- 10.0.0.0 - 10.255.255.255
- 172.16.0.0 - 172.31.255.255
- 192.168.0.0 - 192.168.255.255
To create a subnet address, a network administrator borrows bits from the original host portion and designates them as the subnet field.

**SOLUTION:** Create another section in the IP address called the subnet.
IP Datagram Format

**Question:** In which order are the bytes of an IP datagram transmitted?

**Answer:**
- Transmission is row by row
- For each row:
  1. First transmit bits 0-7
  2. Then transmit bits 8-15
  3. Then transmit bits 16-23
  4. Then transmit bits 24-31

This is called network byte order or big endian byte ordering.

**Note:** Many computers (incl. Intel processors) store 32-bit words in little endian format. Others (incl. Motorola processors) use big endian.
Big endian vs. small endian

• Conventions to store a multibyte work
• Example: a 4 byte Long Integer
  Byte3 Byte2 Byte1 Byte0

Little Endian
  Stores the low-order byte at the lowest address and the highest order byte in the highest address.
  Base Address+0 Byte0
  Base Address+1 Byte1
  Base Address+2 Byte2
  Base Address+3 Byte3

Big Endian
  Stores the high-order byte at the lowest address, and the low-order byte at the highest address.
  Base Address+0 Byte3
  Base Address+1 Byte2
  Base Address+2 Byte1
  Base Address+3 Byte0

• Intel processors use this order
• Motorola processors use big endian.

Fields of the IP Header

• Version (4 bits): current version is 4, next version will be 6.
• Header length (4 bits): length of IP header, in multiples of 4 bytes
• DS/ECN field (1 byte)
  □ This field was previously called as Type-of-Service (TOS) field. The role of this field has been re-defined, but is “backwards compatible” to TOS interpretation
  □ Differentiated Service (DS) (6 bits):
    • Used to specify service level (currently not supported in the Internet)
  □ Explicit Congestion Notification (ECN) (2 bits):
    • New feedback mechanism used by TCP
Fields of the IP Header

- **Identification (16 bits):** Unique identification of a datagram from a host. Incremented whenever a datagram is transmitted.

- **Flags (3 bits):**
  - First bit always set to 0
  - DF bit (Do not fragment)
  - MF bit (More fragments)

Fields of the IP Header

- **Time To Live (TTL) (1 byte):**
  - Specifies longest paths before datagram is dropped.
  - Role of TTL field: Ensure that packet is eventually dropped when a routing loop occurs.
  - Used as follows:
    - Sender sets the value (e.g., 64)
    - Each router decrements the value by 1
    - When the value reaches 0, the datagram is dropped.
Fields of the IP Header

- **Protocol (1 byte):**
  - Specifies the higher-layer protocol.
  - Used for demultiplexing to higher layers.

- **Header checksum (2 bytes):** A simple 16-bit long checksum which is computed for the header of the datagram.

---

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:** Padding bytes are added to ensure that header ends on a 4-byte boundary
Maximum Transmission Unit

- Maximum size of IP datagram is 65535, but the data link layer protocol generally imposes a limit that is much smaller.

- Example:
  - Ethernet frames have a maximum payload of 1500 bytes
    $\rightarrow$ IP datagrams encapsulated in Ethernet frame cannot be longer than 1500 bytes

- The limit on the maximum IP datagram size, imposed by the data link protocol is called **maximum transmission unit (MTU)**

- MTUs for various data link protocols:
  - Ethernet: 1500
  - FDDI: 4352
  - 802.3: 1492
  - ATM AAL5: 9180
  - 802.5: 4464
  - PPP: negotiated

IP Fragmentation

- 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?

<table>
<thead>
<tr>
<th>MTUs</th>
<th>FDDI: 4352</th>
<th>Ethernet: 1500</th>
</tr>
</thead>
</table>

- **Fragmentation:**
  - IP router splits the datagram into several datagram
  - Fragments are reassembled at receiver
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 the destination hosts!!

What’s involved in Fragmentation?

- The following fields in the IP header are involved:

<table>
<thead>
<tr>
<th>Version</th>
<th>Header length</th>
<th>DS</th>
<th>ECN</th>
<th>Total length (in bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification</td>
<td>0</td>
<td>FF</td>
<td>Fragment offset</td>
<td></td>
</tr>
<tr>
<td>Time-to-live (TTL)</td>
<td>Protocol</td>
<td>Header checksum</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Identification**
When a datagram is fragmented, the identification is the same in all fragments.

**Flags**
- DF bit is set: Datagram cannot be fragmented and must be discarded if MTU is too small.
- MF bit set: This datagram is part of a fragment and an additional fragment follows this one.
What’s involved in Fragmentation?

- The following fields in the IP header are involved:

<table>
<thead>
<tr>
<th>Version</th>
<th>Header length</th>
<th>DS</th>
<th>ECN</th>
<th>Total length (in bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification</td>
<td>0</td>
<td>D</td>
<td>M</td>
<td>Fragment offset</td>
</tr>
<tr>
<td>Time-to-live (TTL)</td>
<td>Protocol</td>
<td>Header checksum</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Fragment offset**
Offset of the payload of the current fragment in the original datagram

**Total length**
Total length of the current fragment

Example of Fragmentation

- A datagram with size 2400 bytes must be fragmented according to an MTU limit of 1000 bytes

<table>
<thead>
<tr>
<th>Header length</th>
<th>Total length</th>
<th>Identification</th>
<th>DF flag</th>
<th>MF flag</th>
<th>Fragment offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>2400</td>
<td>0xa428</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>448</td>
<td>0xa428</td>
<td>0</td>
<td>0</td>
<td>244</td>
</tr>
<tr>
<td>20</td>
<td>996</td>
<td>0xa428</td>
<td>0</td>
<td>1</td>
<td>122</td>
</tr>
<tr>
<td>20</td>
<td>996</td>
<td>0xa428</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

MTU: 4000
---
MTU: 1000

IP datagram → Fragment 3 → Fragment 2 → Fragment 1 → Router
Determining the length of fragments

- To determine the size of the fragments we recall that, since there are only 13 bits available for the fragment offset, the offset is given as a multiple of eight bytes. As a result, the first and second fragment have a size of 996 bytes (and not 1000 bytes). This number is chosen since 976 is the largest number smaller than $1000-20=980$ that is divisible by eight. The payload for the first and second fragments is 976 bytes long, with bytes 0 through 975 of the original IP payload in the first fragment, and bytes 976 through 1951 in the second fragment. The payload of the third fragment has the remaining 428 bytes, from byte 1952 through 2379. With these considerations, we can determine the values of the fragment offset, which are 0, $976 / 8 = 122$, and $1952 / 8 = 244$, respectively, for the first, second and third fragment.

Network Address Translation
NAT

- NAT first became popular as a way to deal with the IPv4 address shortage
- Private address space
  - 10.0.0.0 - 10.255.255.255 (10.0.0.0/8)
  - 172.16.0.0 - 172.31.255.255 (172.16.0.0/12)
  - 192.168.0.0 - 192.168.255.255 (192.168.0.0/16)

NAT Configuration

- In a typical configuration, a local network uses one of the "private" IP address subnets
- a router on that network has a private address (such as 192.168.0.1) in that address space
- The router is also connected to the Internet with a single "public" address (known as "overloaded" NAT) or multiple "public" addresses assigned by an ISP
- As traffic passes from the local network to the Internet, the source address in each packet is translated on the fly from the private addresses to the public address(es)
NAT Configuration

- The router tracks basic data about each active connection (particularly the destination address and port).
- When a reply returns to the router, it uses the connection tracking data it stored during the outbound phase to determine where on the internal network to forward the reply.
- **TCP** or **UDP** client port numbers are used to demultiplex the packets in the case of overloaded NAT, or IP address and port number when multiple public addresses are available, on packet return.

---

**Understanding How NAT Overload Works**

This diagram shows what happens to a packet that traverses a NAT Overload enabled router.
Unleashing the Power of NAT Overload

Here we see how the NAT Overload router deals with multiple packets from 3 different hosts on the private network. Notice that only the Source IP Address field is changed as the packets traverse the router.

Unleashing the Power of NAT Overload

The reply from the Internet server arrives at our router’s public interface. The router accepts the packet and, after checking where it’s come from and its destination port, which is 3000, it recognises this as a reply to the packet Host 1 sent earlier. The router modifies the Destination IP Address to that of Host 1’s IP Address and forwards the packet.
Outgoing PPTP Client Through NAT

1. Outgoing PPTP Client Through NAT
   - a) 10.0.0.2
   - b) 10.0.0.3
   - c) 10.0.0.4

Outgoing Web Client Through NAT

1. Outgoing Web Client Through NAT
   - a) 10.0.0.2
   - b) 10.0.0.3
   - c) 10.0.0.4

<table>
<thead>
<tr>
<th>Inside IP</th>
<th>Inside Port</th>
<th>Out IP</th>
<th>Out Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.0.3</td>
<td>1034</td>
<td>204.1.1.10</td>
<td>2005</td>
</tr>
<tr>
<td>10.0.0.4</td>
<td>1025</td>
<td>204.1.1.10</td>
<td>2000</td>
</tr>
</tbody>
</table>
Outgoing Web Client Through NAT

Based on the slides of Dr. Jorg Liebeherr, University of Virginia
The IP (Internet Protocol) relies on several other protocols to perform necessary control and routing functions:

- Control functions (ICMP)
- Multicast signaling (IGMP)
- Setting up routing tables (RIP, OSPF, BGP, PIM, …)

The Internet Control Message Protocol (ICMP) is a helper protocol that supports IP with facility for:

- Error reporting
- Simple queries

ICMP messages are encapsulated as IP datagrams:
ICMP message format

4 byte header:
- Type (1 byte): type of ICMP message
- Code (1 byte): subtype of ICMP message
- Checksum (2 bytes): similar to IP header checksum. Checksum is calculated over entire ICMP message

If there is no additional data, there are 4 bytes set to zero. Each ICMP messages is at least 8 bytes long

ICMP Query message

ICMP query:
- Request sent by host to a router or host
- Reply sent back to querying host
# Example of ICMP Queries

<table>
<thead>
<tr>
<th>Type/Code</th>
<th>Description</th>
<th>Example of a Query: Echo Request and Reply</th>
</tr>
</thead>
<tbody>
<tr>
<td>8/0</td>
<td>Echo Request</td>
<td>Ping’s are handled directly by the kernel</td>
</tr>
<tr>
<td>0/0</td>
<td>Echo Reply</td>
<td>Each Ping is translated into an ICMP Echo Request</td>
</tr>
<tr>
<td>13/0</td>
<td>Timestamp Request</td>
<td>The Ping’ed host responds with an ICMP Echo Reply</td>
</tr>
<tr>
<td>14/0</td>
<td>Timestamp Reply</td>
<td></td>
</tr>
<tr>
<td>10/0</td>
<td>Router Solicitation</td>
<td></td>
</tr>
<tr>
<td>9/0</td>
<td>Router Advertisement</td>
<td></td>
</tr>
</tbody>
</table>

Each Ping is translated into an ICMP Echo Request. The Ping’ed host responds with an ICMP Echo Reply.
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>Identifier</td>
<td>sequence number</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>

ICMP Error message

- ICMP error messages report error conditions.
- Typically sent when a datagram is discarded.
- Error message is often passed from ICMP to the application program.
ICMP Error message

ICMP error messages include the complete IP header and the first 8 bytes of the payload (typically: UDP, TCP)

Frequent ICMP Error message

<table>
<thead>
<tr>
<th>Type</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0–15</td>
<td>Destination unreachable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Notification that an IP datagram could not be forwarded and was dropped. The code field contains an explanation.</td>
</tr>
<tr>
<td>5</td>
<td>0–3</td>
<td>Redirect</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Informs about an alternative route for the datagram and should result in a routing table update. The code field explains the reason for the route change.</td>
</tr>
<tr>
<td>11</td>
<td>0, 1</td>
<td>Time exceeded</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sent when the TTL field has reached zero (Code 0) or when there is a timeout for the reassembly of segments (Code 1)</td>
</tr>
<tr>
<td>12</td>
<td>0, 1</td>
<td>Parameter problem</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sent when the IP header is invalid (Code 0) or when an IP header option is missing (Code 1)</td>
</tr>
</tbody>
</table>
Some subtypes of the “Destination Unreachable”

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Reason for Sending</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Network Unreachable</td>
<td>No routing table entry is available for the destination network.</td>
</tr>
<tr>
<td>1</td>
<td>Host Unreachable</td>
<td>Destination host should be directly reachable, but does not respond to ARP Requests.</td>
</tr>
<tr>
<td>2</td>
<td>Protocol Unreachable</td>
<td>The protocol in the protocol field of the IP header is not supported at the destination.</td>
</tr>
<tr>
<td>3</td>
<td>Port Unreachable</td>
<td>The transport protocol at the destination host cannot pass the datagram to an application.</td>
</tr>
<tr>
<td>4</td>
<td>Fragmentation Needed and DF Bit Set</td>
<td>IP datagram must be fragmented, but the DF bit in the IP header is set.</td>
</tr>
</tbody>
</table>

Example: 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 unreachable message to the source host.

- Scenario:

```
Request a service at a port 80
```

```
No process is waiting at port 80
```
Delivery of an IP datagram

- View at the data link layer:
  - Internetwork is a collection of LANs or point-to-point links or switched networks that are connected by routers
Delivery of an IP datagram

- View at the IP layer:
  - An IP network is a logical entity with a network number
  - We represent an IP network as a “cloud”
  - The IP delivery service takes the view of clouds, and ignores the data link layer view

Tenets of end-to-end delivery of datagrams

The following conditions must hold so that an IP datagram can be successfully delivered

1. The network prefix of an IP destination address must correspond to a unique data link layer network (=LAN or point-to-point link or switched network). (The reverse need not be true!)
2. Routers and hosts that have a common network prefix must be able to exchange IP datagrams using a data link protocol (e.g., Ethernet, PPP)
3. Every data link layer network must be connected to at least one other data link layer network via a router.
Routing tables

- Each router and each host keeps a **routing table** which tells the router how to process an outgoing packet.
- **Main columns:**
  1. **Destination address:** where is the IP datagram going to?
  2. **Next hop:** how to send the IP datagram?
  3. **Interface:** what is the output port?
- Next hop and interface column can often be summarized as one column.
- Routing tables are set so that datagrams gets closer to the its destination.

<table>
<thead>
<tr>
<th>Destination</th>
<th>Next Hop</th>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1.0.0/24</td>
<td>direct</td>
<td>eth0</td>
</tr>
<tr>
<td>10.1.2.0/24</td>
<td>direct</td>
<td>eth0</td>
</tr>
<tr>
<td>10.2.1.0/24</td>
<td>direct</td>
<td>R4</td>
</tr>
<tr>
<td>10.3.1.0/24</td>
<td>direct</td>
<td>eth1</td>
</tr>
<tr>
<td>20.1.0.0/16</td>
<td>R4</td>
<td>eth0</td>
</tr>
<tr>
<td>20.2.1.0/28</td>
<td>R4</td>
<td>eth0</td>
</tr>
</tbody>
</table>

Routing table of a host or router:

IP datagrams can be directly delivered ("direct") or is sent to a router ("R4")

Delivery with routing tables:

1. **Destination:** 10.1.0.0/24, **Next Hop:** R3
2. **Destination:** 10.1.2.0/24, **Next Hop:** R1
3. **Destination:** 10.2.1.0/24, **Next Hop:** R4
4. **Destination:** 10.3.1.0/24, **Next Hop:** R4
5. **Destination:** 20.1.0.0/16, **Next Hop:** direct
6. **Destination:** 20.2.1.0/28, **Next Hop:** direct

Route to: **20.2.1.2**
Delivery of IP datagrams

- There are two distinct processes to delivering IP datagrams:
  1. **Forwarding**: How to pass a packet from an input interface to the output interface?
  2. **Routing**: How to find and setup the routing tables?

- Forwarding must be done as fast as possible:
  - on routers, is often done with support of hardware
  - on PCs, is done in kernel of the operating system

- Routing is less time-critical
  - On a PC, routing is done as a background process

Processing of an IP datagram in IP

![Diagram of IP datagram processing](Diagram)

IP router: IP forwarding enabled
Host: IP forwarding disabled
Processing of an IP datagram in IP

- Processing of IP datagrams is very similar on an IP router and a host
- **Main difference:**
  “IP forwarding” is enabled on router and disabled on host

- **IP forwarding enabled**
  → if a datagram is received, but it is not for the local system, the datagram will be sent to a different system

- **IP forwarding disabled**
  → if a datagram is received, but it is not for the local system, the datagram will be dropped

Processing of an IP datagram at a router

1. Receive an IP datagram
2. IP header validation
3. Process options in IP header
4. Parsing the destination IP address
5. Routing table lookup
6. Decrement TTL
7. Perform fragmentation (if necessary)
8. Calculate checksum
9. Transmit to next hop
10. Send ICMP packet (if necessary)
Routing table lookup

- When a router or host need to transmit an IP datagram, it performs a routing table lookup.

- **Routing table lookup**: Use the IP destination address as a key to search the routing table.

- Result of the lookup is the IP address of a next hop router, and/or the name of a network interface.

<table>
<thead>
<tr>
<th>Destination address</th>
<th>Next hop/interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>network prefix</td>
<td>IP address of next hop router</td>
</tr>
<tr>
<td>host IP address</td>
<td>or Name of a network interface</td>
</tr>
<tr>
<td>loopback address</td>
<td>or Name of a network interface</td>
</tr>
<tr>
<td>default route</td>
<td>or Name of a network interface</td>
</tr>
</tbody>
</table>

Type of routing table entries

- **Network route**
  - Destination addresses is a network address (e.g., 10.0.2.0/24)
  - Most entries are network routes

- **Host route**
  - Destination address is an interface address (e.g., 10.0.1.2/32)
  - Used to specify a separate route for certain hosts

- **Default route**
  - Used when no network or host route matches
  - The router that is listed as the next hop of the default route is the default gateway (for Cisco: “gateway of last resort”)

- **Loopback address**
  - Routing table for the loopback address (127.0.0.1)
  - The next hop lists the loopback (lo0) interface as outgoing interface
Routing table lookup: Longest Prefix Match

- **Longest Prefix Match:** Search for the routing table entry that has the longest match with the prefix of the destination IP address

1. Search for a match on all 32 bits
2. Search for a match for 31 bits
...
32. Search for a match on 0 bits

Host route, loopback entry
→ 32-bit prefix match
Default route is represented as 0.0.0.0/0
→ 0-bit prefix match

<table>
<thead>
<tr>
<th>Destination address</th>
<th>Next hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.0.0/8</td>
<td>R1</td>
</tr>
<tr>
<td>128.143.0.0/16</td>
<td>R2</td>
</tr>
<tr>
<td>128.143.64.0/20</td>
<td>R3</td>
</tr>
<tr>
<td>128.143.192.0/20</td>
<td>R3</td>
</tr>
<tr>
<td>128.143.71.0/24</td>
<td>R4</td>
</tr>
<tr>
<td>128.143.71.55/32</td>
<td>R3</td>
</tr>
<tr>
<td>default</td>
<td>R5</td>
</tr>
</tbody>
</table>

The longest prefix match for 128.143.71.21 is for 24 bits with entry 128.143.71.0/24

Datagram will be sent to R4

---

Route Aggregation

- Longest prefix match algorithm permits to aggregate prefixes with identical next hop address to a single entry
- This contributes significantly to reducing the size of routing tables of Internet routers

<table>
<thead>
<tr>
<th>Destination</th>
<th>Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1.0.0/24</td>
<td>R3</td>
</tr>
<tr>
<td>10.1.2.0/24</td>
<td>direct</td>
</tr>
<tr>
<td>10.2.1.0/24</td>
<td>direct</td>
</tr>
<tr>
<td>10.3.1.0/24</td>
<td>R3</td>
</tr>
<tr>
<td>20.2.0.0/16</td>
<td>R2</td>
</tr>
<tr>
<td>20.1.1.0/28</td>
<td>R2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Destination</th>
<th>Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1.0.0/24</td>
<td>R3</td>
</tr>
<tr>
<td>10.1.2.0/24</td>
<td>direct</td>
</tr>
<tr>
<td>10.2.1.0/24</td>
<td>direct</td>
</tr>
<tr>
<td>10.3.1.0/24</td>
<td>R3</td>
</tr>
<tr>
<td>20.0.0.0/8</td>
<td>R2</td>
</tr>
</tbody>
</table>
How do routing tables get updated?

- Adding an interface:
  - Configuring an interface eth2 with 10.0.2.3/24 adds a routing table entry:

<table>
<thead>
<tr>
<th>Destination</th>
<th>Next Hop/ interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.2.0/24</td>
<td>eth2</td>
</tr>
</tbody>
</table>

- Adding a default gateway:
  - Configuring 10.0.2.1 as the default gateway adds the entry:

<table>
<thead>
<tr>
<th>Destination</th>
<th>Next Hop/ interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0.0.0/0</td>
<td>10.0.2.1</td>
</tr>
</tbody>
</table>

- Static configuration of network routes or host routes
- Update of routing tables through routing protocols
- ICMP messages

Routing table manipulations with ICMP

- When a router detects that an IP datagram should have gone to a different router, the router (here R2)
  - forwards the IP datagram to the correct router
  - sends an ICMP redirect message to the host
- Host uses ICMP message to update its routing table
ICMP Router Solicitation

ICMP Router Advertisement

- After bootstrapping a host broadcasts an **ICMP router solicitation**.
- In response, routers send an **ICMP router advertisement** message
- Also, routers periodically broadcast **ICMP router advertisement**

This is sometimes called the Router Discovery Protocol