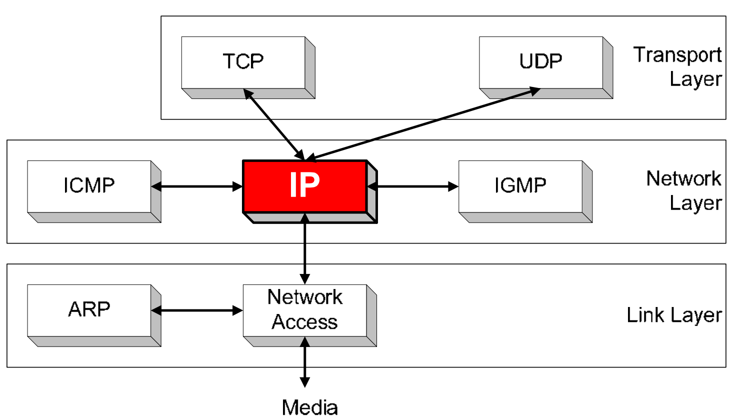
IP - The Internet Protocol

Based on the slides of Dr. Jorg Liebeherr, University of Virginia

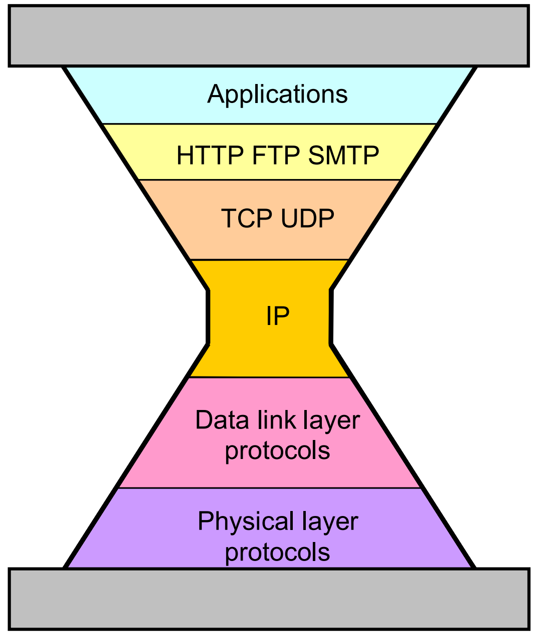
Orientation

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



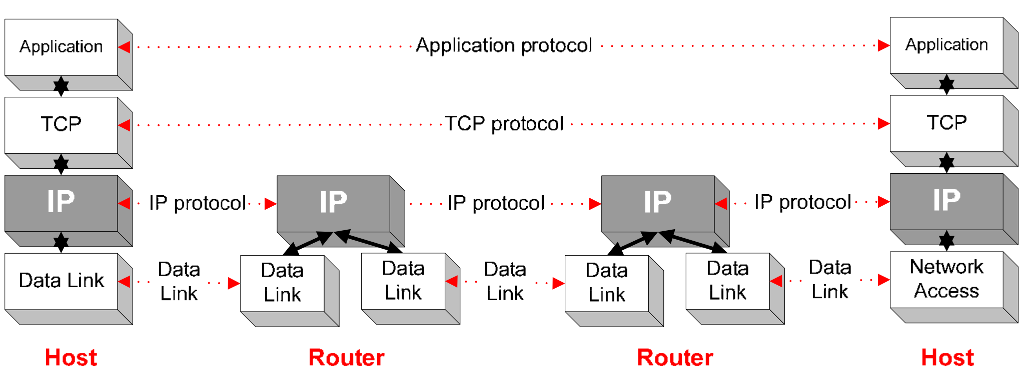
IP: The waist of the hourglass

- **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,…)

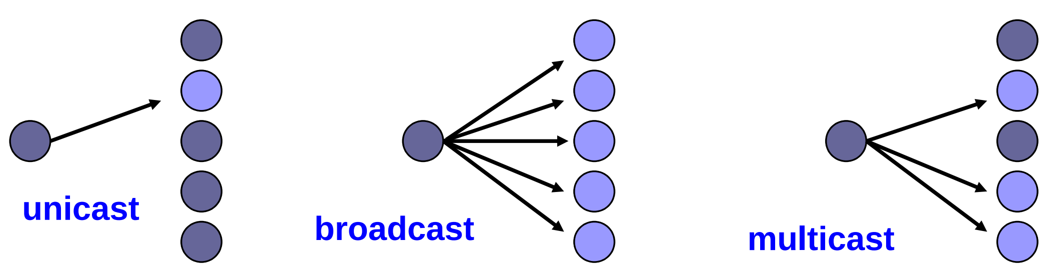
- Consequences:

  - 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

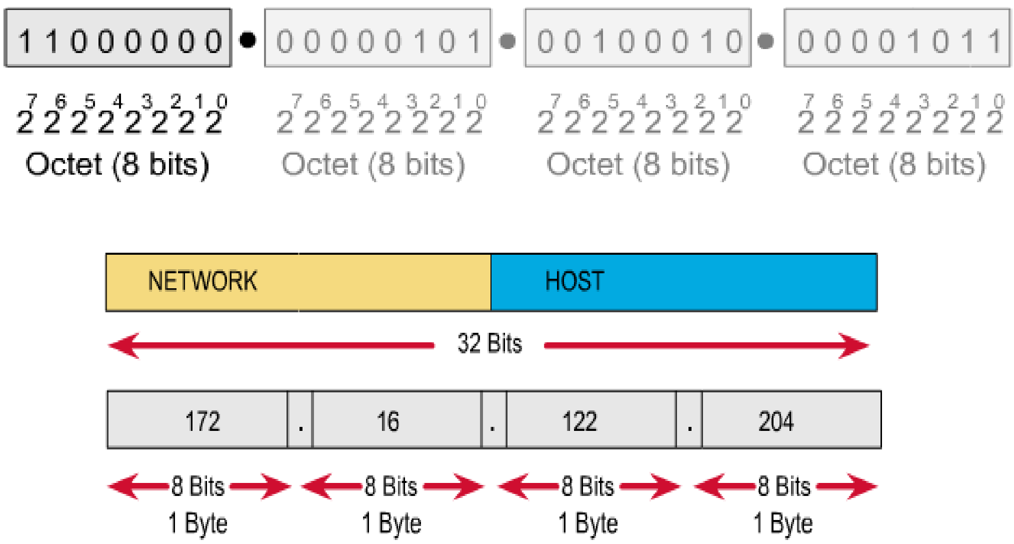
- IP is a network layer - it must be capable of providing communication between hosts on different kinds of networks (different data-link implementations).
- The address must include information about what *network* the receiving host is on. This is what makes routing feasible.

IP Addresses

- IP addresses are *logical* addresses (not physical)
- 32 bits. IPv4 *(version 4)*
- 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

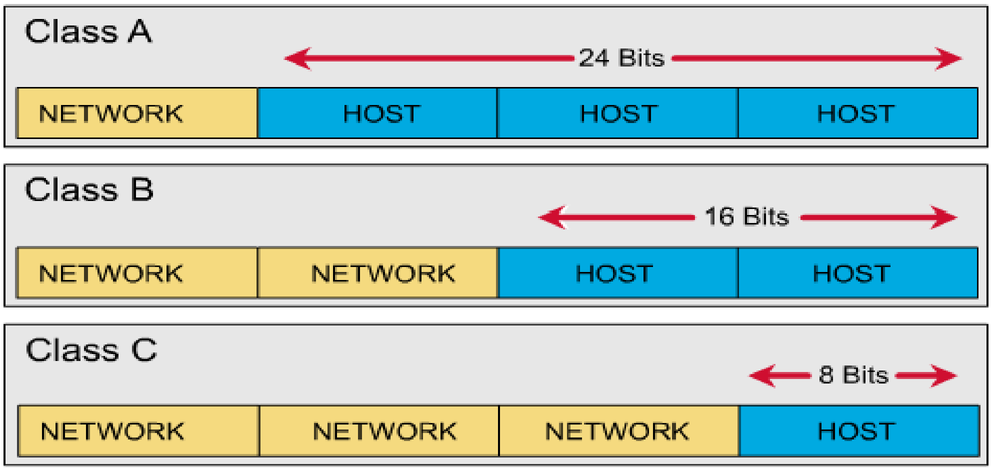


Binary and Decimal Conversion

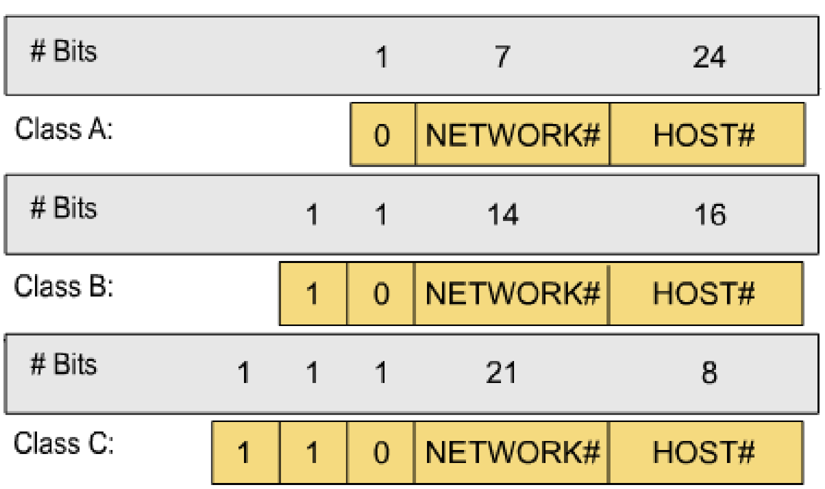
$2^{(7)}$	$2^{(6)}$	$2^{(5)}$	$2^{(4)}$	$2^{(3)}$	$2^{(2)}$	$2^{(1)}$	$2^{(0)}$
128	64	32	16	8	4	2	1

192.57.30.224
11000000.00111001.00011110.11100000

Classes of Network IP Addresses

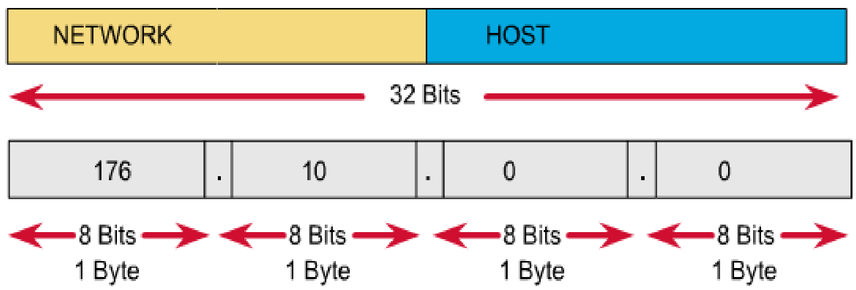


IP Addresses as Decimal Numbers



Network IDs and Broadcast Addresses

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.



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

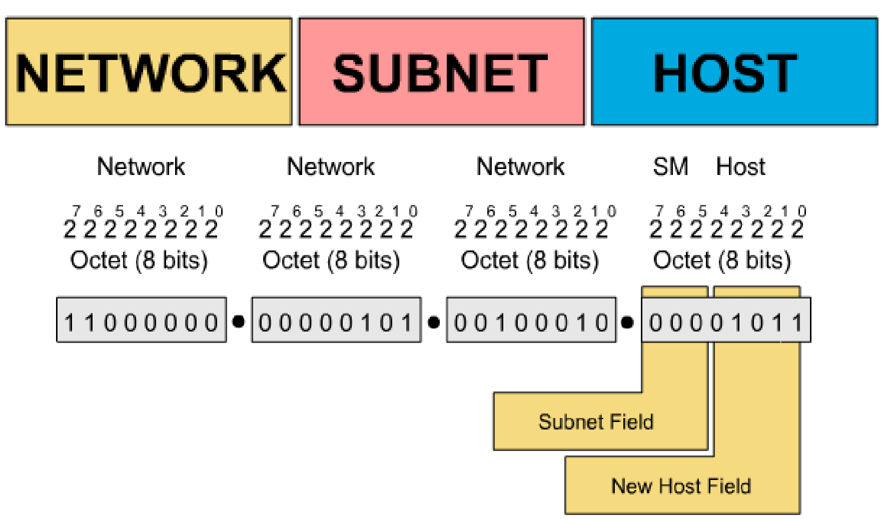
192.168.0.0 - 192.168.255.255

Subnetworks

To create a subnet address, a network administrator borrows bits from the original host portion and designates them as the subnet field.

Subnetworks

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



IP Datagram Format

<table>
<tr><td>bit # 0</td><td>7 | 8</td><td>15 | 16</td><td>23 | 24</td><td>31</td></tr>
<tr><td>version</td><td>header length</td><td>DS</td><td>ECN</td><td colspan="4">total length (in bytes)</td></tr>
<tr><td colspan="4">Identification</td><td>0</td><td>D F</td><td>M F</td><td>Fragment offset</td></tr>
<tr><td colspan="2">time-to-live (TTL)</td><td colspan="2">protocol</td><td colspan="4">header checksum</td></tr>
<tr><td colspan="8">source IP address</td></tr>
<tr><td colspan="8">destination IP address</td></tr>
<tr><td colspan="8">options (0 to 40 bytes)</td></tr>
<tr><td colspan="8">payload</td></tr>
<tr><td colspan="8">←4 bytes→</td></tr>
</table>

- 20 bytes ≤ Header Size < 2^4 x 4 bytes = 60 bytes
- 20 bytes ≤ Total Length < 2^{16} bytes = 65536 bytes

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
```

- Intel processors use this order

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
```

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)

  Will be explained later→ Fragmentation

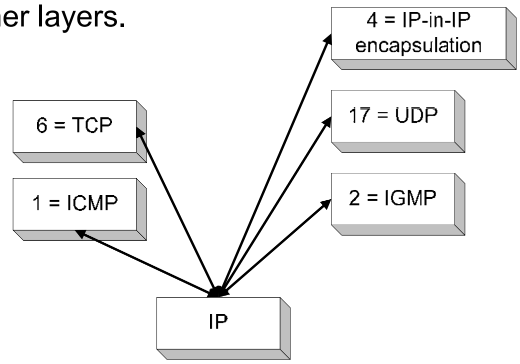
Fields of the IP Header

- **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

    → 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)**

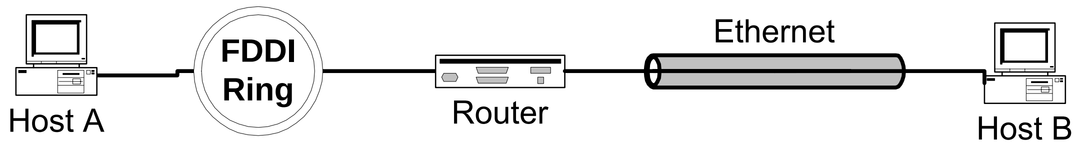
- MTUs for various data link protocols:

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?

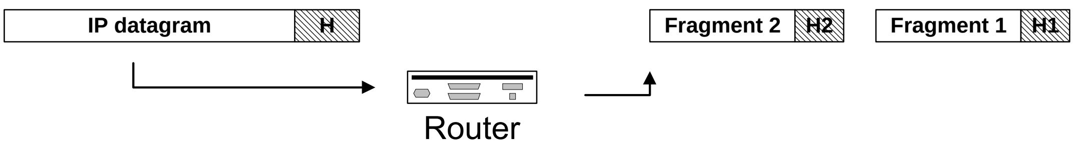


MTUs: FDDI: 4352 Ethernet: 1500

- **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 destination hosts !!



What's involved in Fragmentation?

- The following fields in the IP header are involved:

<table>
<tr><td>version</td><td>header length</td><td>DS</td><td>ECN</td><td colspan="4">total length (in bytes)</td></tr>
<tr><td colspan="4">Identification</td><td>0</td><td>DF</td><td>MF</td><td>Fragment offset</td></tr>
<tr><td colspan="2">time-to-live (TTL)</td><td colspan="2">protocol</td><td colspan="4">header checksum</td></tr>
</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:

version	header length	DS	ECN	total length (in bytes)			
Identification				0	DF	MF	Fragment offset
time-to-live (TTL)		protocol		header checksum			

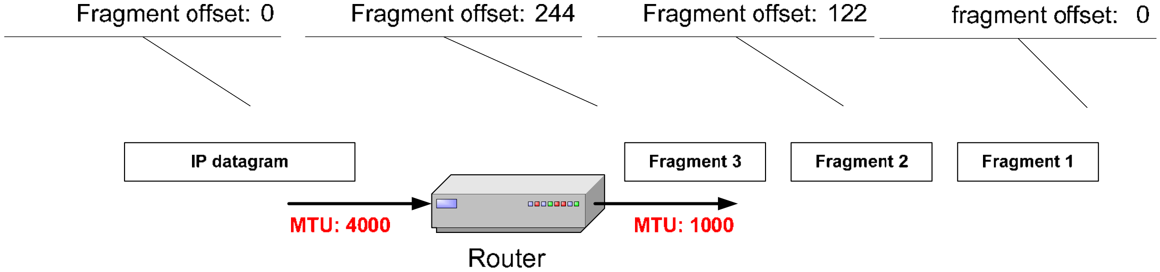
Fragment offset — Offset of the payload of the current fragment in the original datagram

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

Header length: 20	Header length: 20	Header length: 20	Header length: 20
Total length: 2400	Total length: 448	Total length: 996	Total length: 996
Identification: 0xa428	Identification: 0xa428	Identification: 0xa428	Identification: 0xa428
DF flag: 0	DF flag: 0	DF flag: 0	DF flag: 0
MF flag: 0	MF flag: 0	MF flag: 1	MF flag: 1
Fragment offset: 0	Fragment offset: 244	Fragment offset: 122	fragment offset: 0



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

WWW Resource: http://www.firewall.cx/modules.php?name=Alternative_Menu

NAT

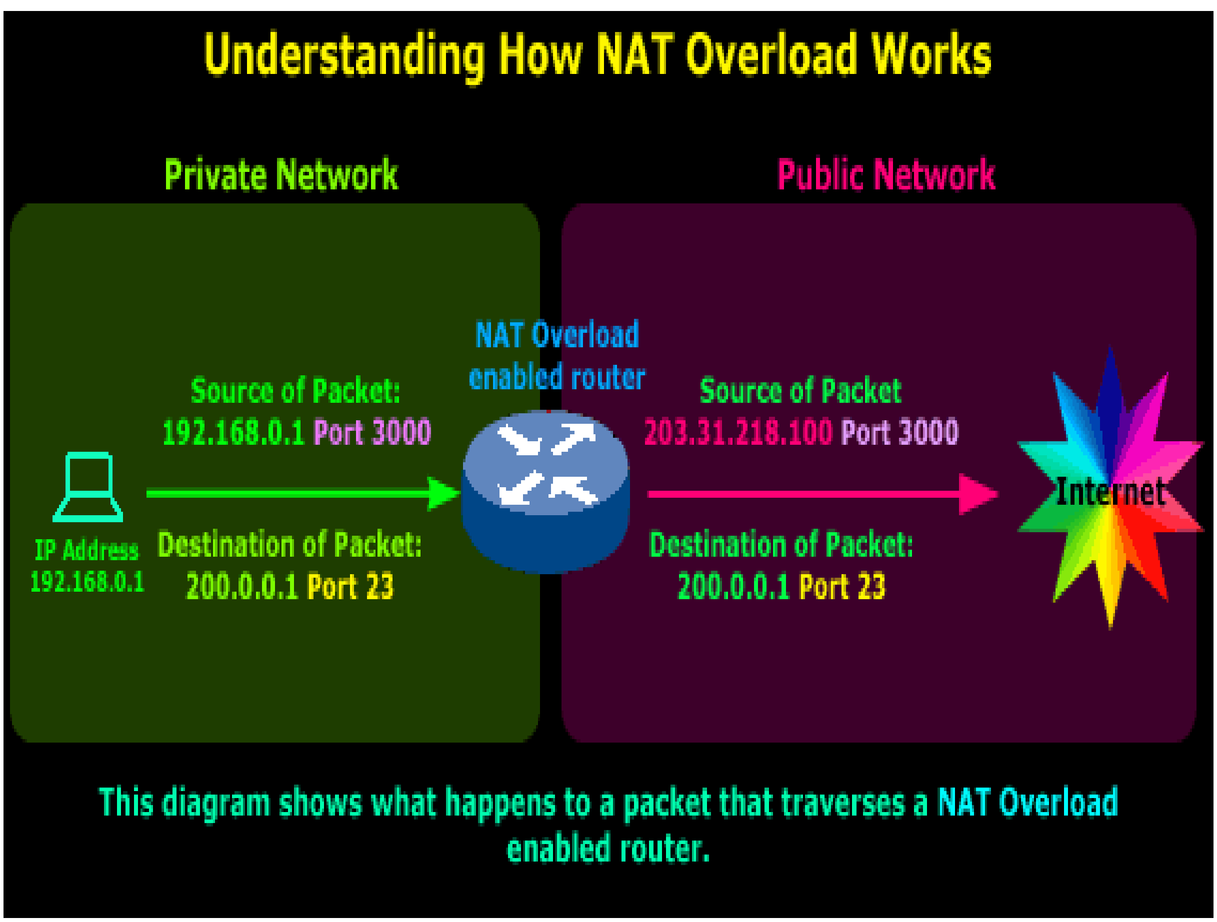
- NAT first became popular as a way to deal with the IPv4 address shortage
- Private address space
  - 10.0.0.0 - 10.255.255.255 (10.0.0.0/8)
  - 172.16.0.0 - 172.31.255.255 (172.16.0.0/12)
  - 192.168.0.0 - 192.168.255.255 (192.168.0.0/16)

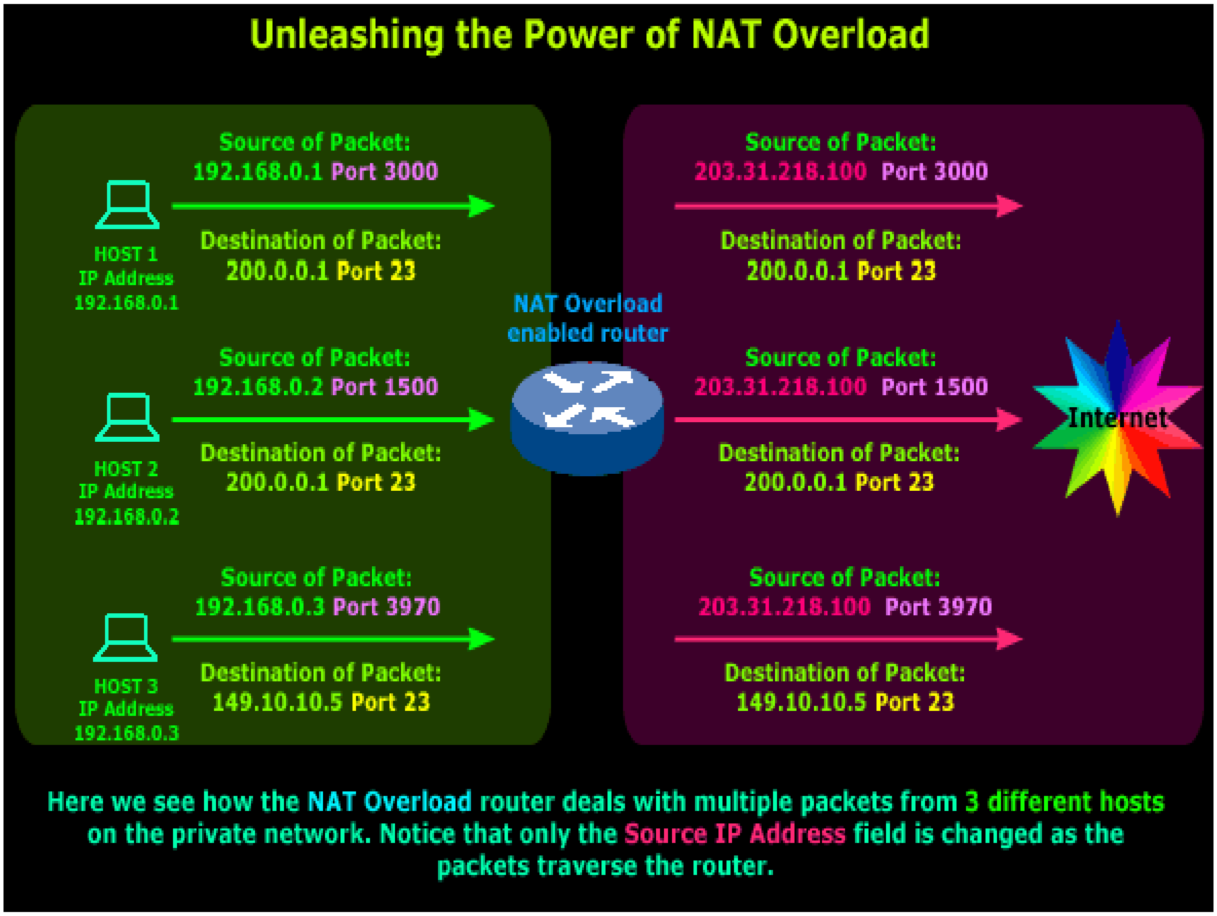
NAT Configuration

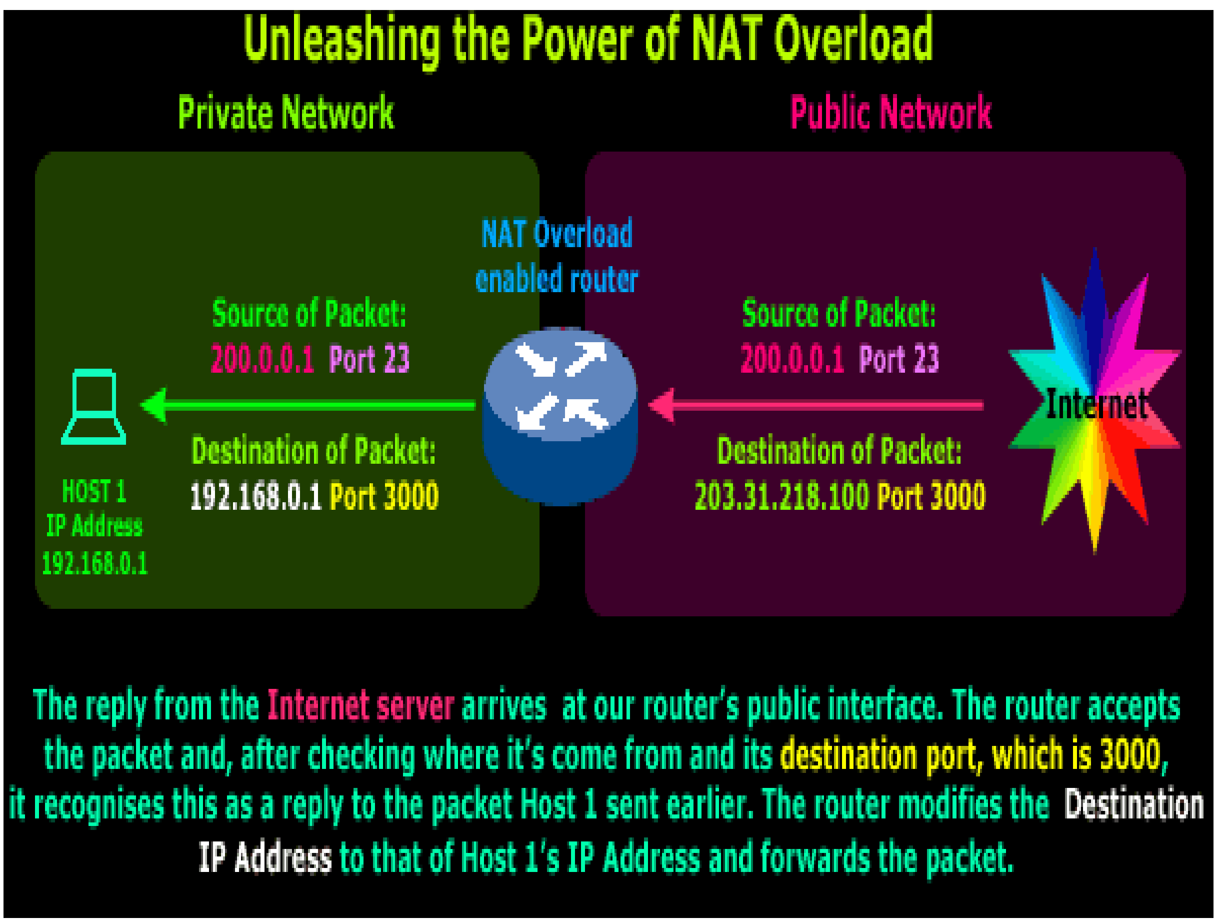
- In a typical configuration, a local network uses one of the "private" IP address subnets
- a router on that network has a private address (such as 192.168.0.1) in that address space
- The router is also connected to the Internet with a single "public" address (known as "overloaded" NAT) or multiple "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

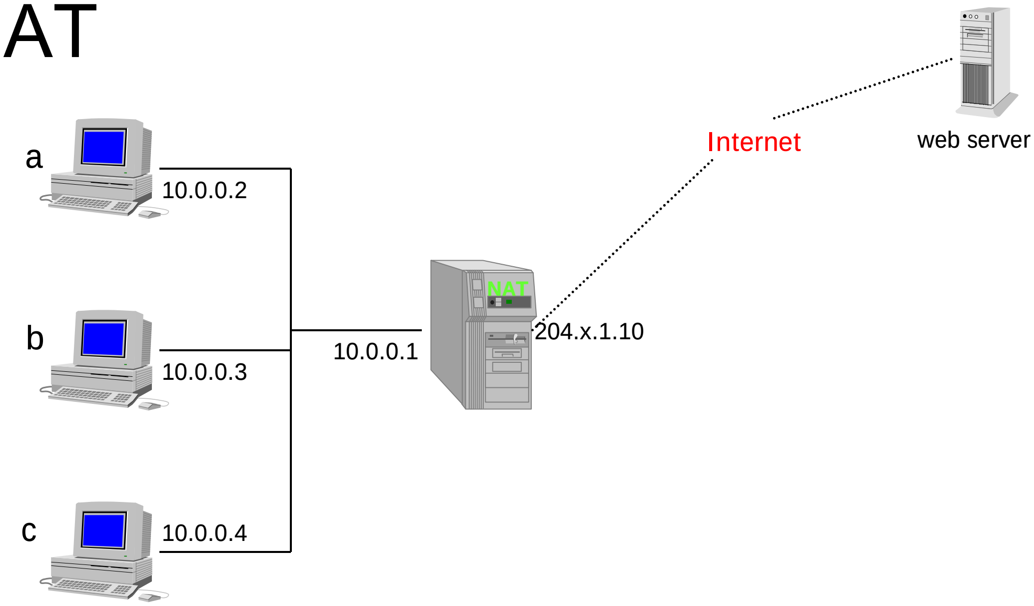
- The router tracks basic data about each active connection (particularly the destination address and port)
- When a reply returns to the router, it uses the connection tracking data it stored during the outbound phase to determine where on the internal network to forward the reply
- TCP or UDP client 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



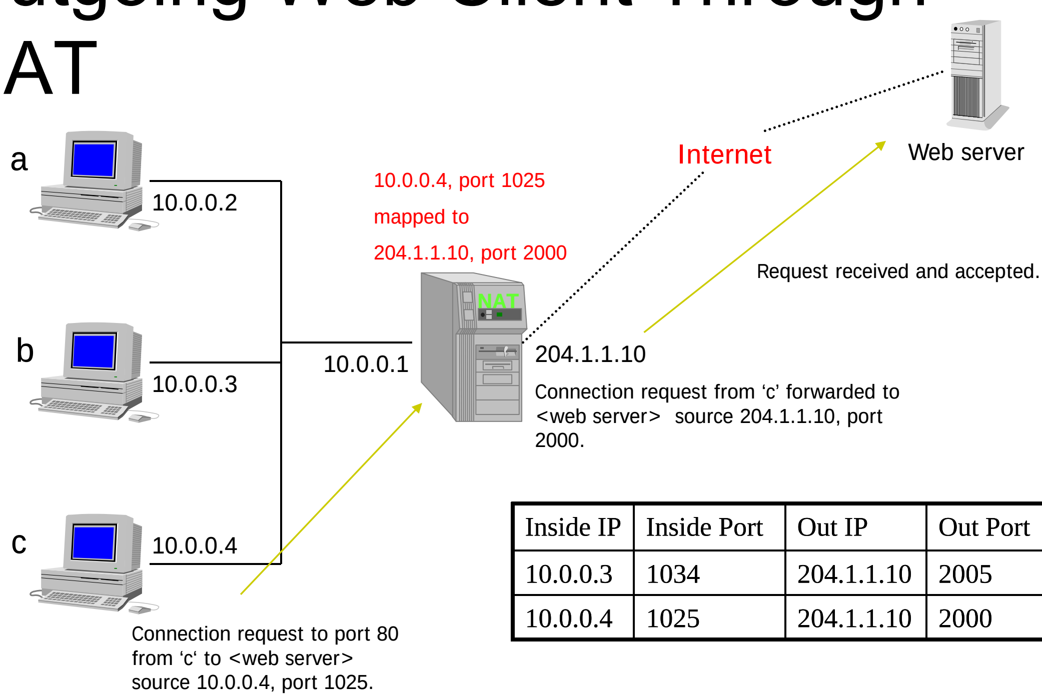

Unleashing the Power of NAT Overload
Source of Packet: 192.168.0.1 Port 3000
Destination of Packet: 200.0.0.1 Port 23
HOST 1 IP Address 192.168.0.1
Source of Packet: 192.168.0.2 Port 1500
Destination of Packet: 200.0.0.1 Port 23
HOST 2 IP Address 192.168.0.2
Source of Packet: 192.168.0.3 Port 3970
Destination of Packet: 149.10.10.5 Port 23
HOST 3 IP Address 192.168.0.3
NAT Overload enabled router
Source of Packet: 203.31.218.100 Port 3000
Destination of Packet: 200.0.0.1 Port 23
Source of Packet: 203.31.218.100 Port 1500
Destination of Packet: 200.0.0.1 Port 23
Source of Packet: 203.31.218.100 Port 3970
Destination of Packet: 149.10.10.5 Port 23
Internet
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
Private Network
Public Network
NAT Overload enabled router
Source of Packet: 200.0.0.1 Port 23
Destination of Packet: 192.168.0.1 Port 3000
HOST 1 IP Address 192.168.0.1
Source of Packet: 200.0.0.1 Port 23
Destination of Packet: 203.31.218.100 Port 3000
Internet
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

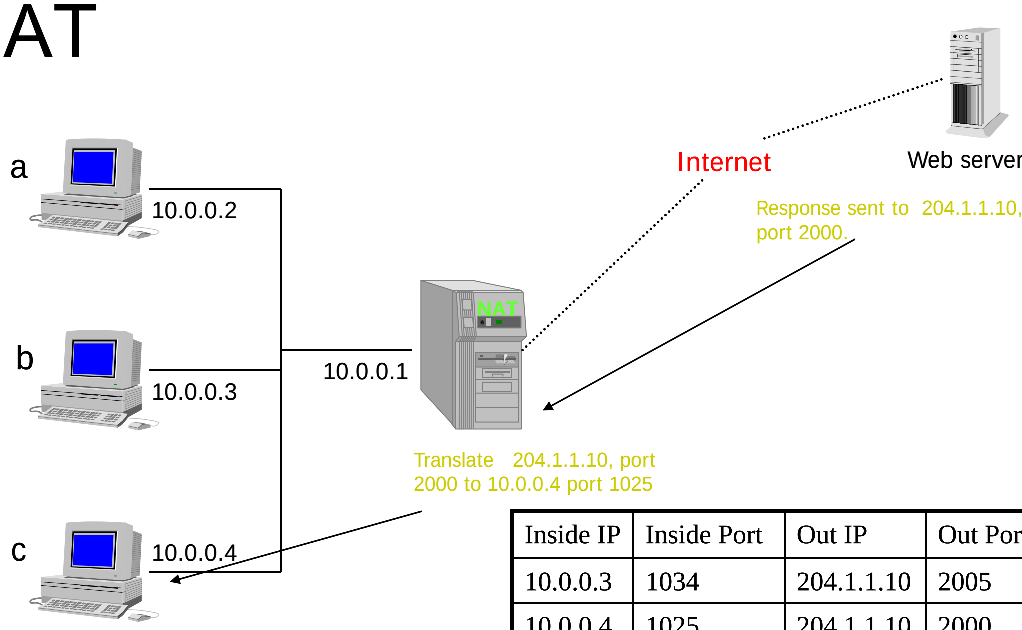


Outgoing Web Client Through NAT



Inside IP	Inside Port	Out IP	Out Port
10.0.0.3	1034	204.1.1.10	2005
10.0.0.4	1025	204.1.1.10	2000

Outgoing Web Client Through NAT



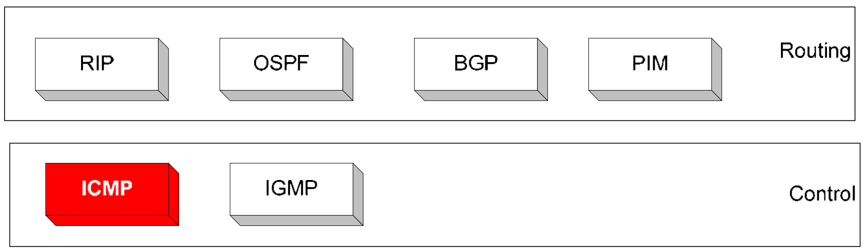
Inside IP	Inside Port	Out IP	Out Port
10.0.0.3	1034	204.1.1.10	2005
10.0.0.4	1025	204.1.1.10	2000

Internet Control Message Protocol (ICMP)

Based on the slides of Dr. Jorg Liebeherr, University of Virginia

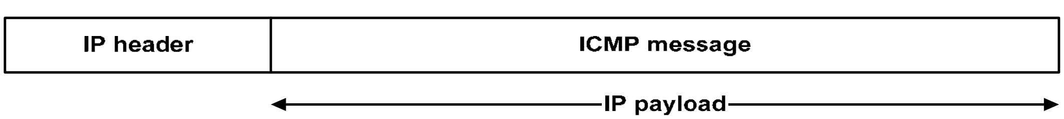
Overview

- 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, …)



Overview

- 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

bit # 0 … 7	8 … 15	16 … 23	24 … 31
type	**code**	**checksum**	
additional information or 0x00000000			

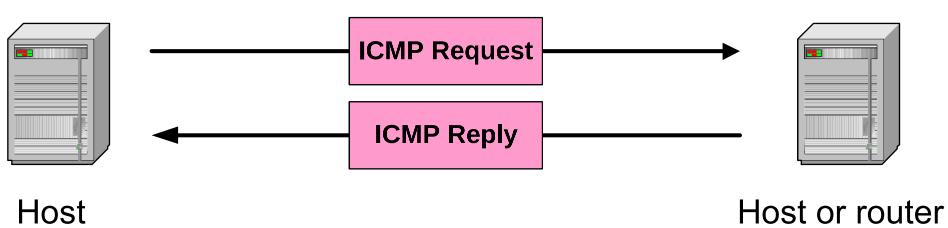
4 byte header:

- 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

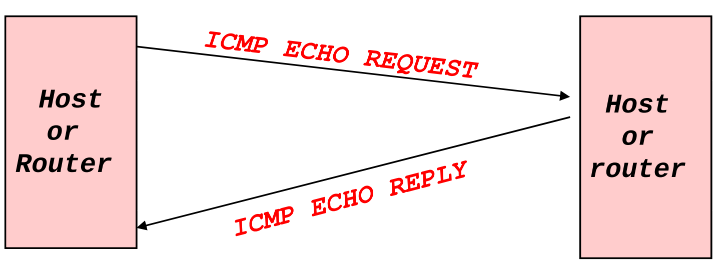
Example of ICMP Queries

Type/Code:	Description
8/0	Echo Request
0/0	Echo Reply
13/0	Timestamp Request
14/0	Timestamp Reply
10/0	Router Solicitation
9/0	Router Advertisement

The ping command uses Echo Request/ Echo Reply

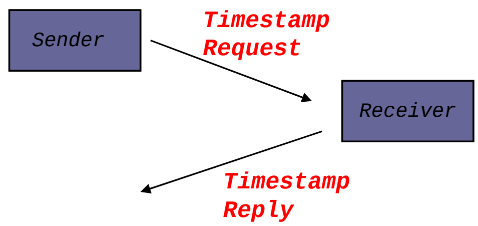
Example of a Query: Echo Request and Reply

- 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



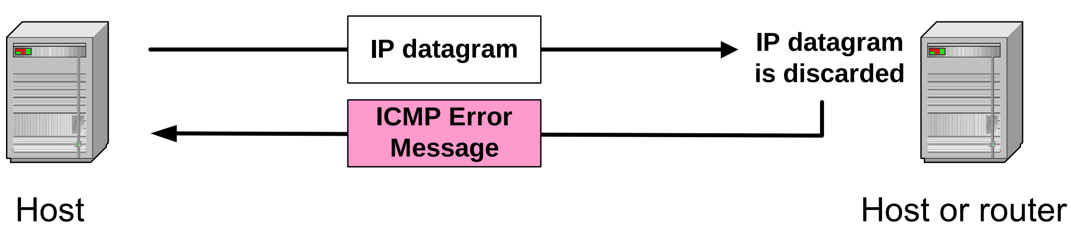
Example of a Query: ICMP Timestamp

- 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



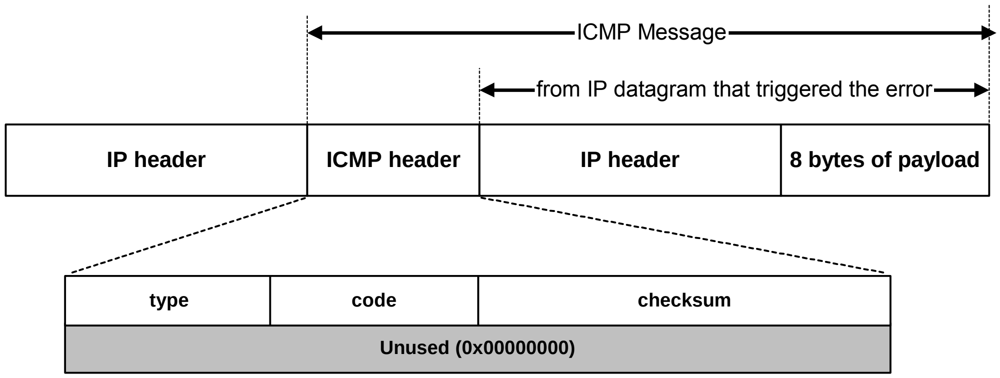
<table>
<tr><td>Type
(= 17 or 18)</td><td>Code
(=0)</td><td colspan="2">Checksum</td></tr>
<tr><td colspan="2">identifier</td><td colspan="2">sequence number</td></tr>
<tr><td colspan="4">32-bit sender timestamp</td></tr>
<tr><td colspan="4">32-bit receive timestamp</td></tr>
<tr><td colspan="4">32-bit transmit timestamp</td></tr>
</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

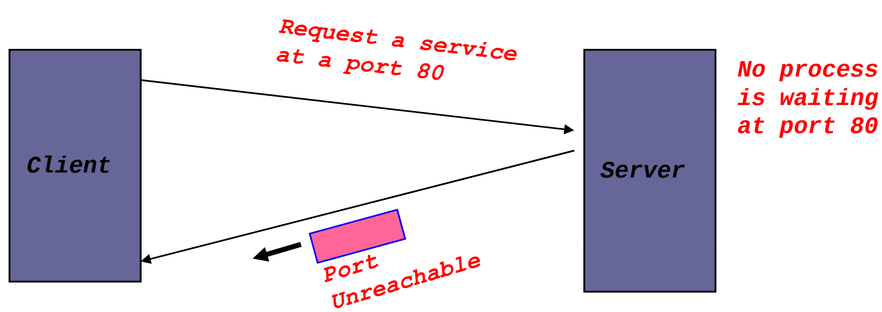
Type	Code	Description	
3	0–15	Destination unreachable	Notification that an IP datagram could not be forwarded and was dropped. The code field contains an explanation.
5	0–3	Redirect	Informs about an alternative route for the datagram and should result in a routing table update. The code field explains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

Some subtypes of the “Destination Unreachable”

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

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
  destination unreachable message to the source host.
- Scenario:

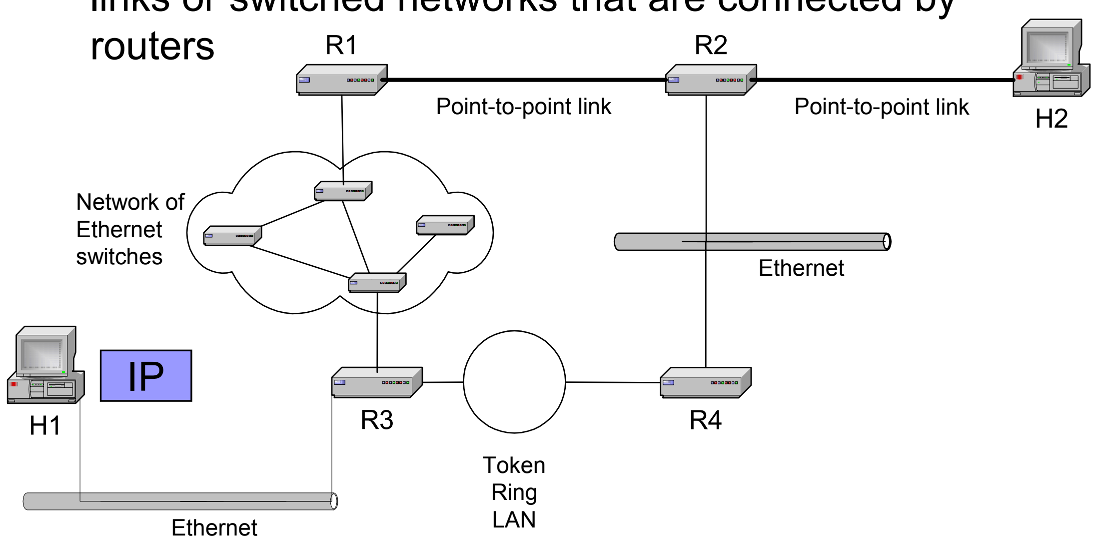


IP Forwarding

Based on the slides of Dr. Jorg Liebeherr, University of Virginia

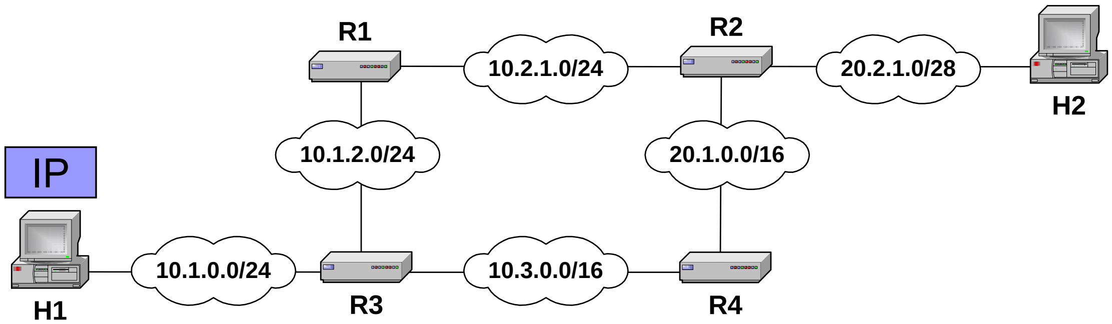
Delivery of an IP datagram

- 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 dagrams 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

Routing table of a host or router

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

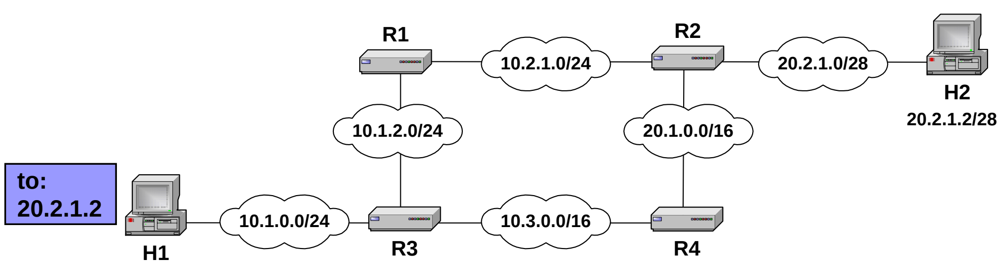
Destination	Next Hop	interface
10.1.0.0/24	direct	eth0
10.1.2.0/24	direct	eth0
10.2.1.0/24	R4	serial0
10.3.1.0/24	direct	eth1
20.1.0.0/16	R4	eth0
20.2.1.0/28	R4	eth0

Delivery with routing tables

Destination	Next Hop
10.1.0.0/24	R3
10.1.2.0/24	direct
10.2.1.0/24	direct
10.3.1.0/24	R3
20.2.0.0/16	R2
30.1.1.0/28	R2

Destination	Next Hop
10.1.0.0/24	R1
10.1.2.0/24	R1
10.2.1.0/24	direct
10.3.1.0/24	R4
20.1.0.0/16	direct
20.2.1.0/28	direct

Destination	Next Hop
10.1.0.0/24	R2
10.1.2.0/24	R2
10.2.1.0/24	R2
10.3.1.0/24	R2
20.1.0.0/16	R2
20.2.1.0/28	direct



Destination	Next Hop
10.1.0.0/24	direct
10.1.2.0/24	R3
10.2.1.0/24	R3
10.3.1.0/24	R3
20.1.0.0/16	R3
20.2.1.0/28	R3

Destination	Next Hop
10.1.0.0/24	direct
10.1.2.0/24	direct
10.2.1.0/24	R4
10.3.1.0/24	direct
20.1.0.0/16	R4
20.2.1.0/28	R4

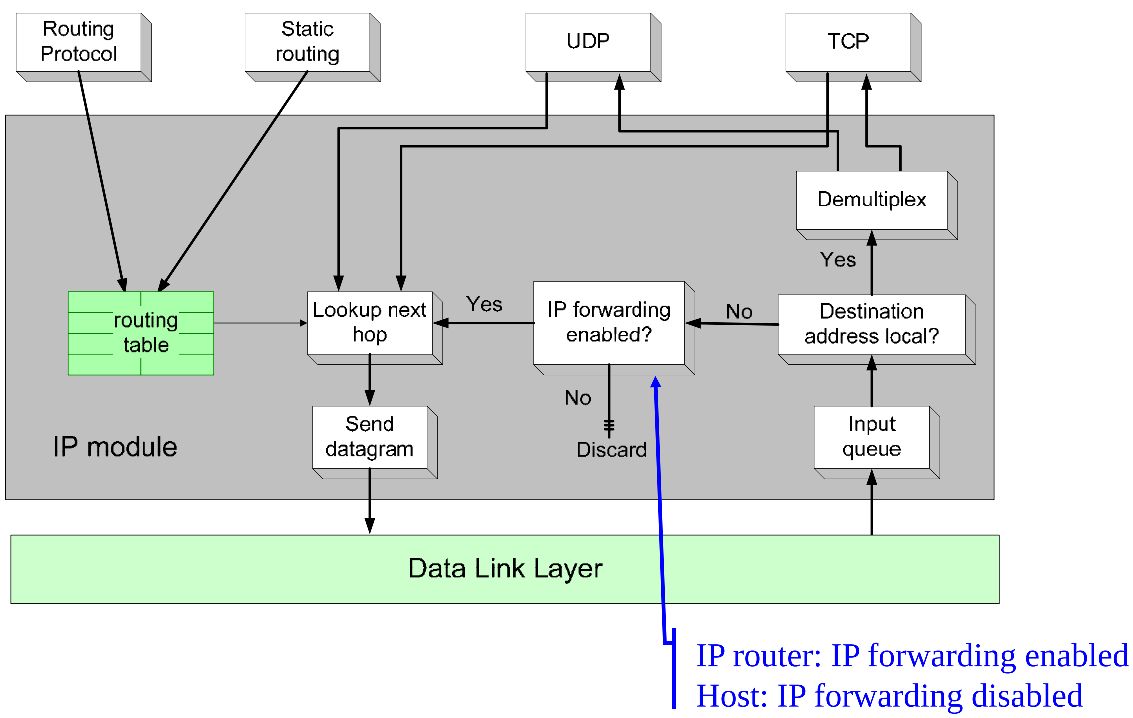
Destination	Next Hop
10.1.0.0/24	R3
10.1.2.0/24	R3
10.2.1.0/24	R2
10.3.1.0/24	direct
20.1.0.0/16	direct
20.2.1.0/28	R2

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



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

Receive an IP datagram →

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

Destination address	Next hop/ interface
network prefix *or* host IP address *or* loopback address *or* default route	IP address of next hop router *or* Name of a network interface

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 mach 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

128.143.71.21

Destination address	Next hop
10.0.0.0/8	R1
128.143.0.0/16	R2
128.143.64.0/20	R3
128.143.192.0/20	R3
128.143.71.0/24	R4
128.143.71.55/32	R3
default	R5

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

Destination	Next Hop
10.1.0.0/24	R3
10.1.2.0/24	direct
10.2.1.0/24	direct
10.3.1.0/24	R3
20.2.0.0/16	R2
20.1.1.0/28	R2

Destination	Next Hop
10.1.0.0/24	R3
10.1.2.0/24	direct
10.2.1.0/24	direct
10.3.1.0/24	R3
20.0.0.0/8	R2

How do routing tables get updated?

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

Destination	Next Hop/ interface
10.0.2.0/24	eth2

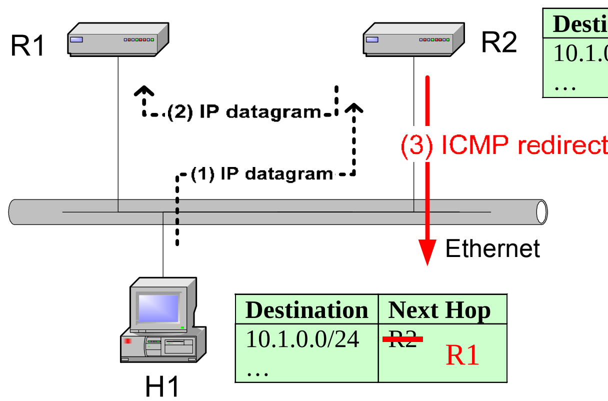
- Adding a default gateway:
  - Configuring 10.0.2.1 as the default gateway adds the entry:

Destination	Next Hop/ interface
0.0.0.0/0	10.0.2.1

- 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

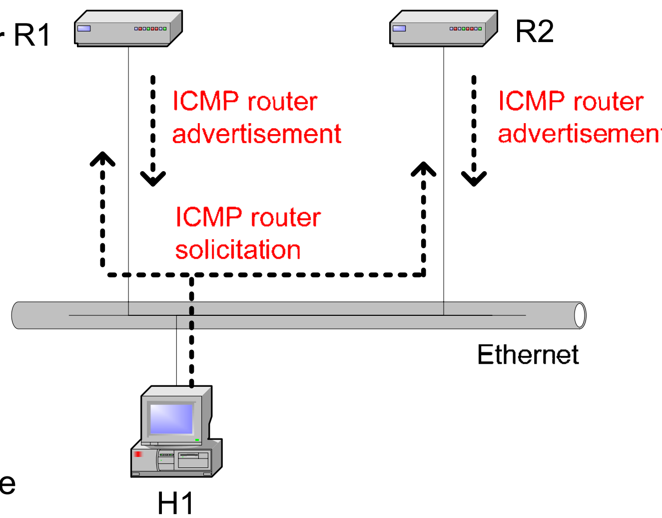


Destination	Next Hop
10.1.0.0/24	R1
…	

Destination	Next Hop
10.1.0.0/24	~~R2~~ R1
…	

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