CS 4453 Computer Networks

Chapter 4 TCP/IP and the Internet

2015 Winter

In this chapter, we consider Internet and focus on TCP/IP protocols. The materials will mostly cover protocols in transport layer and network layer, while sometimes the related protocols at other layers will be mentioned. Since TCP/IP is next to application layer, when we develop internet applications, these protocols need to be considered.

4.1 Internet architecture

Internet Addresses: IPv4 (Internet Protocol version 4) assigns to each host a 32-bit integer address called the Internet address, or IP addresses, which is different from a host's physical addresses. An IP address encodes the identification of the network to which a host attaches as well as the the identification of a unique host on that network.

In history, a classful network architecture is used. Each address is a pair (netid, hostid), where *netid* identifies a network, and *hostid* identifies a host on that network.

class	leading (bits)	first octet (decimal)	netid	hostid	# of networks	# of address
A	0	0-127	a.	b.c.d	2^7	2^{24}
B	10	128-191	a.b.	c.d	2^{14}	2^{16}
C	110	192-223	a.b.c.	d	2^{21}	2^8

Figure 1: IPv4 addresses

Conventionally, the IP addresses are represented as 4 octets and written as 4 integers for 0 to 255. For example, 11000001 00100000 11011000 00001001 will be displayed as 193.32.216.9, which is a class B address, whose netid is 193.32. And 65.39.14.57 is a class A address. The netid is 65.

From Figure 1, we can see that use the classic method of IP address, there are at most $2^7 + 2^{14} + 2^{21}$ networks which is not sufficient for the fast development of Internet.

Later, some classless architecture, called CIDR (Classless Inter-Domain Routing) is defined. In this method, the length of netid is not fixed. For IPv4, the IP address will be a.b.c.d/n, where n is the prefix length (the length of netid in bit from most significant bit of the address), from 0 to 32. In that kind network, the number of addresses are 2^{32-n}. Using this method, we can define more networks (subnets).

Gateway addressing and subnets

The computers that are connected internally to a localized network as well as to an intermediate computer to pass the data to other networks are called Internet gateway or Internet routers.

A gateway has at least two physical interfaces, and an IP address is required for each physical interface.

An IP address specifies a connection to a network rather than to an individual machine. A machine that has n connection networks will have n IP address.

Figure 2 displays an example of networks, where two routers are used. Each of the router has three interfaces. In this example, there are five sub networks. The netid for these networks are $222.23.i$, where $i = 1, 2, 3, 4, 5$. Note that the two routers form one network. If a computer needs to communicate to a computer at other network, then the communication has to go through one or two gateways.

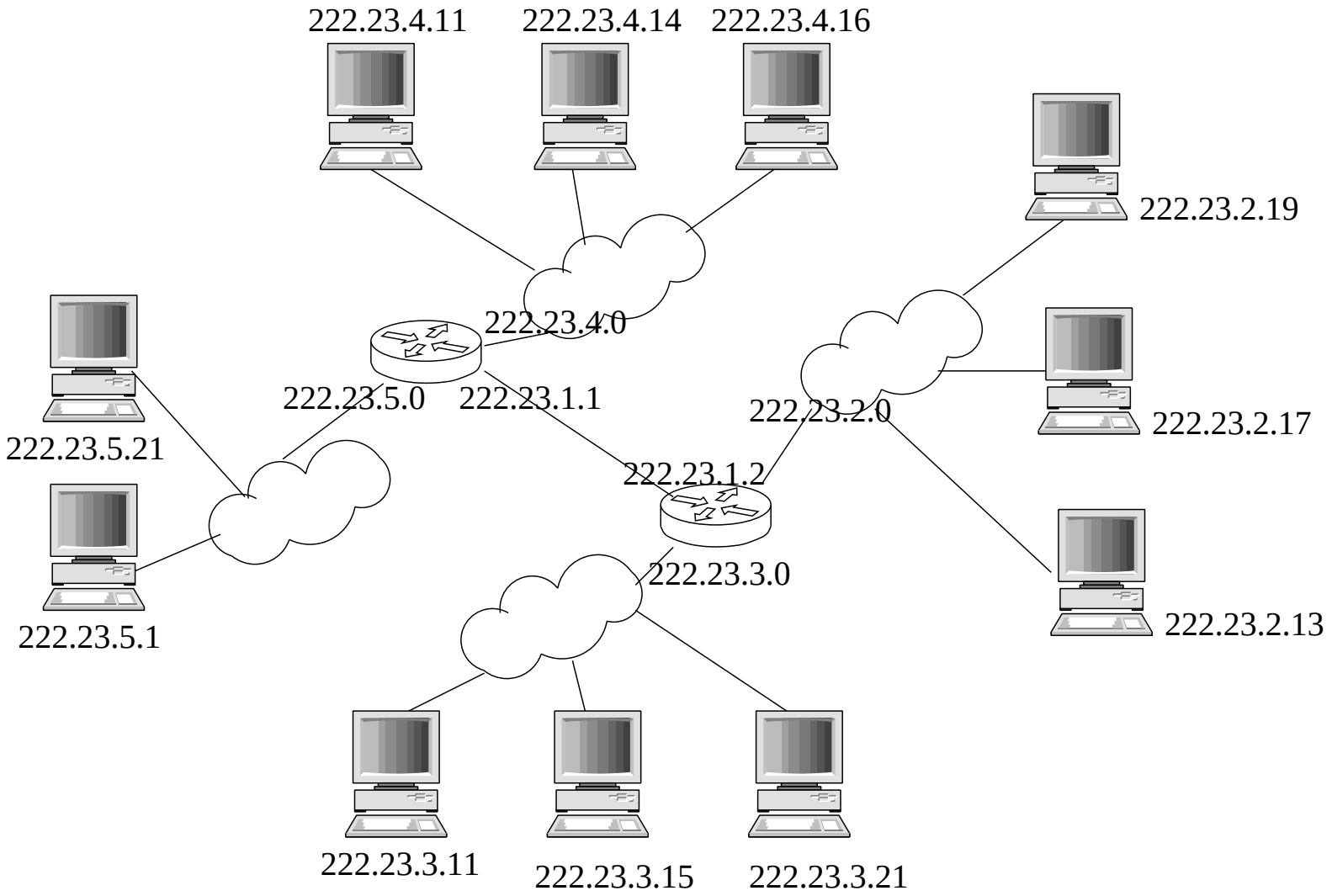


Figure 2: An example of networks

Usually, a network administrator will contact its ISP for a block of addresses from a larger block of addresses, which has been already allocated to the ISP. For example, the ISP has the address block: 200.23.16.0/20 (11001000 00010111 0001|0000 000000000). Then the ISP may assign blocks of addresses to different subnets as follows, where | is used to denote the separation of hostid and netid.

Usually, a network administrator will contact its ISP for a block of addresses from a larger block of addresses, which has been already allocated to the ISP. For example, the ISP has the address block: 200.23.16.0/20 (11001000 00010111 0001|0000 000000000). Then the ISP may assign blocks of addresses to different subnets as follows, where | is used to denote the separation of hostid and netid.

Subnet 1	200.23.16.0/23	11001000000101110001000|0000000000
Subnet 2	200.23.16.0/23	11001000000101110001001|0000000000
Subnet 3	200.23.18.0/23	11001000000101110001010|0000000000
...	...	...
Subnet 7	200.23.30.0/23	11001000000101110001111|0000000000

Loopback addressing

The address 127.0.0.0 (looks like a class A address) is reserved for loopback and is designed for testing and interprocess communication on the local machine. If a program uses the loopback address to send data, the protocol software in the computer returns the data without sending any traffic across the network.

Mapping of physical and IP addresses

In a TCP/IP network, each machine is assigned an IP address and a physical address. The goal of the Address Resolution Protocol (ARP) is to provide low-level software that hides physical addresses and allows higher level program to work with Internet addresses only.

ARP maintains a cache to store recently acquired IP-to-physical address bindings. ARP is not used for crossing networks.

APR is divided into two parts:

- When a host needs to send a packet to a destination host, it looks in the ARP cache to check if the binding between IP address and physical address is available. If the binding is available, the host extracts the physical address from the cache and uses it to send the data. Otherwise it broadcasts a request.
- Whenever an ARP packet arrives from the network, the receiving host extracts the sender's IP address and physical address. It then looks into the local cache to determine whether a binding for the sender IP address exists. If there is such a binding, the host updates the cache entry.

If the incoming ARP packet is a request, the receiving machine must verify that it is the target of the request. If so, the ARP software forms a reply by supplying its physical hardware address and sends the reply directly back to the requester. The receiver also adds the sender's address pair to its cache if the pair is not present. If the IP address in the ARP request does not match the IP address of the receiver, the request is ignored.

Another type of incoming ARP packet is the reply for a past request from this receiver. In this case, first the cache is updated for the address binding. Then the receiver tries to match the reply with a previously issued request. Between the time the machine broadcasts its ARP request and receives the reply, application program or higher level protocol may generate additional requests for the same address. All requests for the same IP address are stored in a queue, and when a reply comes for the IP address, the ARP software removes items from the queue and supplies the address binding to each. If a machine does not issue a request for the IP address in reply, it is ignored.

Dynamic Host Configuration Protocol (DHCP)

DHCP is a network protocol (RFC standard) that is used to configure network devices so that they can communicate on an IP network. A DHCP client uses the DHCP protocol to acquire configuration information, such as an IP address, a default route and one or more DNS (Doman Name System) server addresses from a DHCP server. The DHCP client then uses this information to configure its host. Once the configuration process is complete, the host is able to communicate on that network.

The DHCP server maintains a database of available IP addresses and configuration information. When it receives a broadcast query from a client, the DHCP server determines the network to which the DHCP client is connected, and then allocates an IP address or prefix that is appropriate for the client, and sends to client.

DHCP servers typically grant IP addresses to clients only for a limited interval. DHCP clients are responsible for renewing their IP address before that interval has expired, and must stop using the address once the interval has expired, if they have not been able to renew it. The query is typically initiated immediately after booting, and must complete before the client can initiate IP-based communication with other hosts.

Upon disconnecting, the IP address is returned to the pool for use by another computer. This way, many other computers can use the same IP address within minutes of each other.

The DHCP server may have three methods of allocating IP-addresses:

- dynamic allocation: a network administrator assigns a range of IP addresses to DHCP, and each client computer on the LAN is configured to request an IP address from the DHCP server during network initialization. A lease concept with a controllable time period, allows the DHCP server to reclaim (and then reallocate) IP addresses that are not renewed.
- automatic allocation: The DHCP server permanently assigns a free IP address to a requesting client from the range defined by the administrator and keeps a table of past IP address assignments, so that it can preferentially assign to a client the same IP address that the client previously had.

- static allocation: The DHCP server allocates an IP address based on a table with MAC address/IP address pairs, which are manually filled in. Only clients with a MAC address listed in this table will be allocated an IP address. This feature is not supported by all DHCP servers.

The DHCP server and DHCP client must be connected to the same network link. In larger networks, this is not practical. In this case, one or more DHCP relay agents (usually routers) are used. These DHCP relay agents receive messages from DHCP clients and forward them to DHCP servers. DHCP servers send responses back to the relay agent, and the relay agent then sends these responses to the DHCP client on the local network link.

Network address translation (NAT) was proposed to slow down the speed of running out of IP address in 2001.

The basic idea behind NAT is to assign each company a single IP address or a small number of IP addresses for Internet traffic. Within the company, every computer gets a unique IP address, which is used for routing intramural traffic. However, when a packet exits the company, an address translation takes place.

Three ranges of IP addresses have been declared as private. No packets containing these address may appear on the Internet. The three ranges are:

10.0.0.0	-	10.255.255.255/8	(16,777,216 hosts)
172.16.0.0	-	172.31.255.255/12	(1,048,576 hosts)
192.168.0.0	-	192.168.255.255/16	(65,536 hosts)

When a packet leaves the company premises, it passes through a NAT box that converts the internal IP source address in to the company's true IP address. The NAT box is often combined in a single device with a firewall or integrated into the company's router.

To allow the reply packet reach the internal host, the NAT box need to remember the host's internal IP address with a TCP source port number. That means when the packet go outside, not only the IP address is replaced, the TCP port number is replace as well.

Home network is typically using a NAT-enabled router. Addressing within the home network is 10.0.0.0/24. Suppose the IP address assigned by the ISP is 138.76.29.17. The NAT translation table could be as in Figure 3.

NAT translation table	
WAN side	LAN side
138.76.28.17,5001	10.0.0.1,2334
138.76.28.17,4009	10.0.0.2,1344
138.76.28.17,2109	10.0.0.1,1344
...	...

Figure 3: NAT table

One problem is that a server behind an NAT should have a fixed IP address. But in P2P usually a peer behind an NAT will not have a fixed IP address. So another peer outside the NAT cannot initiate a TCP connection with it.

To solve that problem for P2P application, some method called connection reversal is used. Suppose peer A is outside an NAT and B is inside of the NAT. If A wants to initiates a TCP connection with B, it first find some peer C which is also outside that NAT and C has a TCP connection with B. By the help of C, A then find the way to establish a TCP connection with B.

NAT traversal is increasingly provided by Universal Plug and Play (UPnP), which is a protocol that allows a host to discover and configure a near by NAT.

UPnP requires that both the host and the NAT be UPnP compatible. With UPnP, an application running in a host can request a NAT mapping between its (private IP , port) and (public IP, port) for some requested public port number. If the NAT accepts the request and creates the mapping, then nodes from the outside can initiate TCP connections to the public IP and the port.

The UPnP lets the application know the value of (public IP, port), so that the application can advertise it to the outside world. For example, a BitTorrent peer inside a NAT can ask the NAT to create a "hole" that maps (10.0.0.1, 3345) to (138.76.28.17, 5001) and advertise to its tracker that it is available at (138.76.28.17, 5001). The external peer then may send a TCP SYN packet to (138.76.28.17, 5001). When NAT receives the SYN packet, it will change the destination IP address and the port number in the packet to (10.0.0.1, 3345) and forward the packet through the NAT.

Subnet addresses

For any Internet class, subnetting is introduced to allocate a part of the host address space to network address and leave the remaining part to other physical networks. This adds flexibility and administrative benefits.

The netmask determines which bits in the IP address space represent the subnetwork address and which bits represent host addresses. The netmask also determines how many subnetworks will be created and how many nodes are included in each subnetwork.

In the netmask file, a 1 at a particular position in the netmask indicates that the particular bit in an IP address ought to be the network address. A 0 indicates that the bit belongs to the host address.

For example, for a class B network with network address 131.31.0.0, the two leftmost octets are assigned to the network address, and the right-most octets are assigned to the host number. So that network can host up to 65,534 computers.

However, class B network can be partitioned into 254 subnetworks with up to 254 host computers on each, simply by specifying a netmask 255.255.255.0. This netmask indicates that not only the first two octets, but also the third octet serve as the network address and only the fourth octet is for the host addresses. Note that the binary version of 255.255.255.0 is 11111111.11111111.11111111.00000000.

To create only two subnetworks with 32,766 hosts in each, a subnet netmask is 255.255.128.0 (i.e., 11111111.11111111.10000000.00000000).

4.2 Internet protocol and datagrams

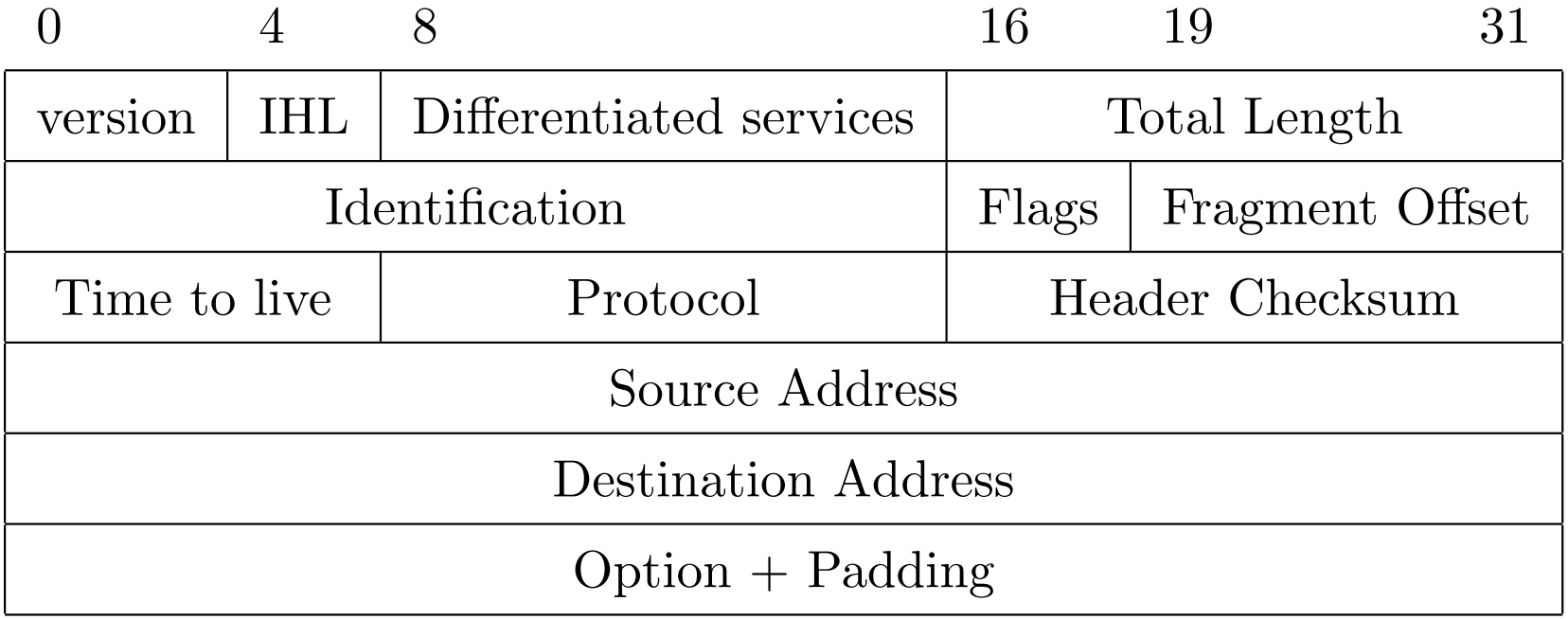


Figure 4: IPv4 header

The items in IPv4 header are as follows.

- Version (4 bits): The version of IP that was used to create the datagram. Machines reject the datagrams with protocol versions that differ from theirs.
- Internet Header Length (IHL) (4 bits): Length of header in 32-bit words. The minimum value is 5.

- Differentiated services (8 bits): Previous called Type of Services (TOS). It provides guidance to end IP modules and to routers along the packet's path about the packet's relative priority. The Type of service field is broken into five subfields as follows:

Precedence	D	T	R	Unused

The first three bits (0-7) indicate datagram precedence, allowing the sender to indicate the importance of each datagram,. Bit sets D, T and R specify that the type of transport: D for low delay, T for high throughput and R for high reliability.

Since the name changed to Differentiated service (RFC 2474), now the top 6 bits are used to mark the packet with its service class (DSCP). The last 2 bits are used to carry explicit congestion notification information (ECN), such as whether the packet has experienced congestion. The main reason for changing this field is that many different types of services, including voice, video, streaming music, etc, have some different requirements for forwarding.

- Total length (16 bits): Total IP packet length, in octets. This includes both header and data. Since the field is only 16 bits long, the maximum size of an IP datagram is 64KB.
- Identification (16 bits): A unique integer identifies the datagram, created from the source address, destination address and user protocol. Retransmissions of IP datagram contains the same identification number. All the fragments of a packet contains the same identification number.
- Flags (3 bits): Indicates whether it is the last fragment of the original datagram. First bit is reserved (unused). Second bit signals DF (Don't fragment) and third bit signals MF (More fragments).

- Fragment Offset (13 bits): Indicate where in the original datagram this fragment belongs, measured in 64-bit units (sequence number of fragments).
- Time to live (8 bits): Specifies how long (in seconds) a packet is allowed to remain in the Internet. Gateways and hosts that process datagram must decrement the TTL field as time passes and remove from the network when time has expired. In practice, the field has become a hop count, when the datagram arrives at a router, the router decrements the TTL field by one. When the TTL field hits zero, the router discards the packet and typically sends an ICMP Time Exceeded message to the sender.
- Protocol (8 bits): Indicates the next higher level protocol.

- Header Checksum (16 bits): An error-detecting code (for the header only. The data will be checked at transport layer). Since some header fields may change during transit, this is reverified and recomputed at each router. If the checksum is wrong, the router will discard the fragment.
- Source Address (32 bits): Coded to allow a variable allocation of bits to specify the network and the end system attached to the specified network.
- Destination Address (32 bits): Same characteristics as source address.

- Options (variable): Encodes the options requested by the sending user, such as security label, source routing, record routing, and timestamp. The list of options may be terminated with an EOL (End of Options List, 0x00) option.
- Padding (variable): Used to ensure that the packet header is a multiple of 32 bits in length.

The underlying physical network transports datagram. Each datagram travels in a distinct physical frame, for efficient Internet transportation.

However, different physical networks allow a different size date. For example, proNET-10 allows 2044 bytes per frame, and 2044 bytes is called this network's maximum transfer unit (MTU). Some MTU size can be quite small ($\leq$ 128 bytes).

Since a datagram may travel across many types of physical network, IP should select a maximum datagram size to ensure that each datagram will always fit into one frame.

Limiting datagram to fit the smallest possible MTU in the Internet makes transfers inefficient. On the other hand, allowing a datagram to be larger than a network MTU means that some datagram will not fit into a single network frame.

TCP/IP software chooses a convenient initial datagram size and arranges a way to divide large datagrams into smaller piece when they need to travel over a network with smaller MTU. The smaller pieces are called fragments, the process of dividing datagrams into smaller pieces is called fragmentation. Fragmentation occurs at a gateway somewhere along the path between the datagram source and its ultimate destination.

Each fragment contains a datagram header that duplicates most of the original datagram header, except for a bit in the flag field, followed by as much data as can be carried in the fragment of a limited MTU. Fragments must be reassembled to produce a complete copy of the datagram before it can be processed at the destination.

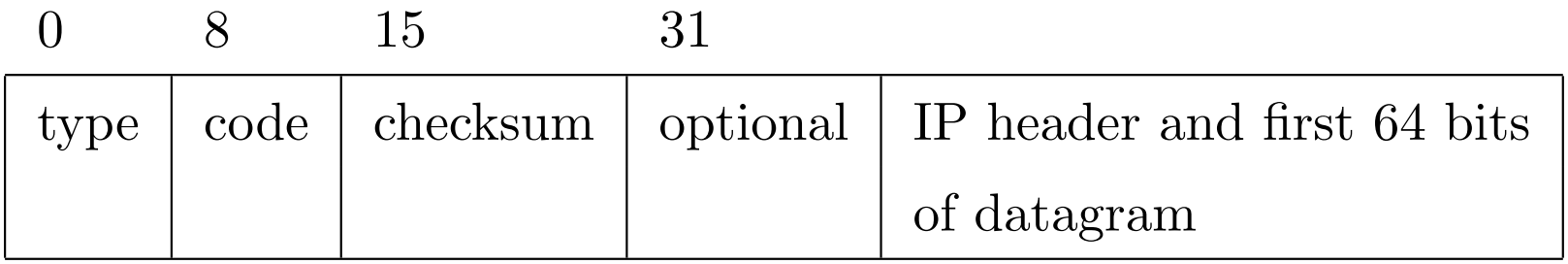
ICMP Internet control message protocol (RFC 792) is used to report error or provide information about unexpected circumstances for routers.

ICMP is often considered part of IP but architecturally it lies above IP, as ICMP messages are carried inside IP datagram as the IP payload.

Some important ICMP types

Type	Code	Description
0	0	Echo reply (used to ping)
3	0	Destination network unreachable
3	1	Destination host unreachable
3	2	Destination protocol unreachable
3	3	Destination port unreachable
3	6	Destination network unknown
5	0	Redirect Datagram for the Network
8	0	Echo request (used to ping)
10	0	Router discovery/selection/solicitation
11	0	TTL expired in transit
13	0	Timestamp
14	0	Timestamp reply

The format of ICMP message is as follows.



ICMP is not restricted to gateways. Any machine can send an ICMP message to another machine. Thus a way exists to report errors to the original source.

Most errors are from the original source, but some do not. However, the datagram only contains field that specify the original source and the ultimate destination and gateways can establish and change their own routing table. So it cannot know the set of intermediate machines that processed the datagram. Basically, ICMP reports to the original source.

Example of applications of ICMP: Traceroute program.

Traceroute in the source sends a series of ordinary IP datagrams to the destination. Each of these datagrams carries a UDP segment with an unlikely UDP port number.

The first of these datagrams has a TTL 1, the second of 2, the third of 3, and so on.

The source also starts timers for each of the datagrams. When the nth datagram arrives at the nth router, the router observes that the TTL of the datagram has just expired. So the router will discard the datagram and sends an ICMP warning message (type 11, code 0) to the sender. This warning message includes the router's IP address.

When the final daragram arrived the traced destination, the destination host will send back a unreachable ICMP message (type 3, code 3) because the port number is unlikely UDP port.

IPv6

IPv6 was developed by the Internet Engineering Task Force (IETF) to deal with the long-anticipated problem of IPv4 address exhaustion.

IPv6 uses a 128-bit address, allowing for 2^{128}, or approximately 3.4×10^{38} addresses, or more than 7.9×10^{28} times as many as IPv4, which uses 32-bit addresses.

IPv6 addresses consist of eight groups of four hexadecimal digits separated by colons, for example 2001:0db8:85a3:0042:1000:8a2e:0370:7334.

The hexadecimal digits are not case-sensitive; e.g., the groups 0DB8 and 0db8 are equivalent.

An IPv6 address may be abbreviated by using one or more of the following rules:

1. Remove one or more leading zeros from one or more groups of hexadecimal digits(For example, convert the group 0042 to 42.)
2. Omit one or more consecutive sections of zeros, using a double colon (::) to denote the omitted sections. The double colon may only be used once in any given address, as the address would be indeterminate if the double colon was used multiple times. (For example, 2001:db8::1:2 is valid, but 2001:db8::1::2 is not permitted.)

Hybrid dual-stack IPv6/IPv4 implementations recognize a special class of addresses, the IPv4-mapped IPv6 addresses.

In these addresses, the first 80 bits are zero, the next 16 bits are one, and the remaining 32 bits are the IPv4 address. The first 96 bits written in the standard IPv6 format, and the remaining 32 bits written in the customary dot-decimal notation of IPv4.

For example, ::ffff:192.0.2.128 represents the IPv4 address 192.0.2.128. A deprecated format for IPv4-compatible IPv6 addresses is ::192.0.2.128.

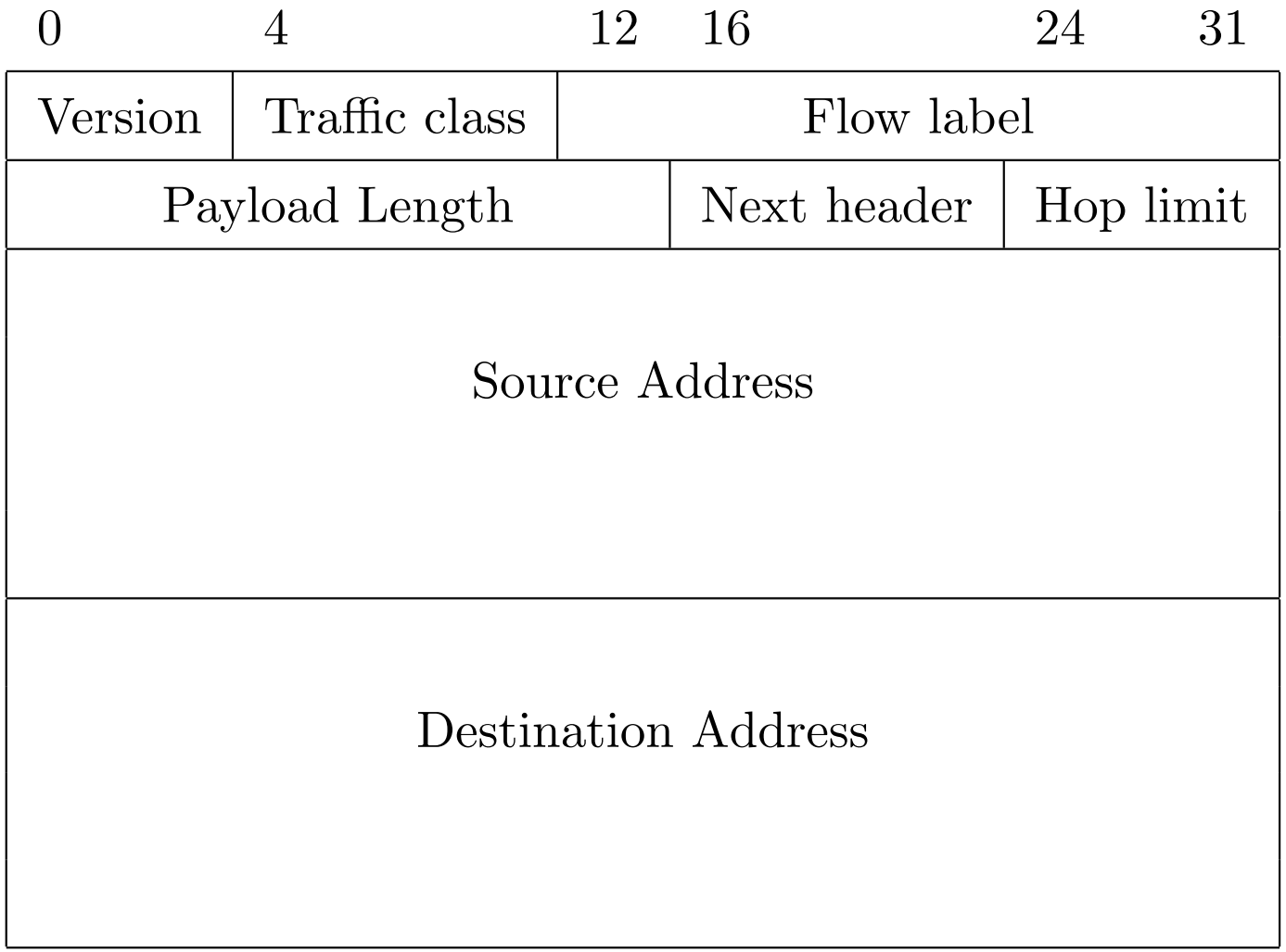
An IPv6 packet has two parts: a header and payload.

The header consists of a fixed portion with minimal functionality required for all packets and may be followed by optional extensions to implement special features.

The fixed header occupies the first 40 octets (320 bits) of the IPv6 packet.

Extension headers carry options that are used for special treatment of a packet in the network, e.g., for routing, fragmentation, and for security using the IPsec framework.

IPv6 header format:



The fields of IPv6 are as follows:

- Version (4 bits) The constant 6 (bit sequence 0110).
- Traffic Class (8 bits) The bits of this field hold two values. The 6 most-significant bits are used for DSCP, which is used to classify packets. The remaining two bits are used for ECN; priority values subdivide into ranges: traffic where the source provides congestion control and non-congestion control traffic.
- Flow Label (20 bits) Originally created for giving real-time applications special service. The flow label when set to a non-zero value now serves as a hint to routers and switches with multiple outbound paths that these packets should stay on the same path so that they will not be reordered. It has further been suggested that the flow label be used to help detect spoofed packets.

- Payload Length (16 bits) The size of the payload in octets, including any extension headers. The length is set to zero when a Hop-by-Hop extension header carries a Jumbo Payload option.
- Next Header (8 bits) Specifies the type of the next header. This field usually specifies the transport layer protocol used by a packet's payload. When extension headers are present in the packet this field indicates which extension header follows. The values are shared with those used for the IPv4 protocol field, as both fields have the same function.

- Hop Limit (8 bits) Replaces the time to live field of IPv4. This value is decremented by one at each intermediate node visited by the packet. When the counter reaches 0 the packet is discarded.
- Source Address (128 bits) The IPv6 address of the sending node.
- Destination Address (128 bits) The IPv6 address of the destination node(s).

Some properties of IPv6:

- The packet header in IPv6 is simpler than that used in IPv4, with many rarely used fields moved to separate optional header extensions.
- IPv6 routers do not perform fragmentation. IPv6 hosts are required to either perform path MTU discovery, perform end-to-end fragmentation, or to send packets no larger than the IPv6 default minimum MTU size of 1280 octets.

- The IPv6 header is not protected by a checksum; integrity protection is assumed to be assured by both link-layer and higher-layer (TCP, UDP, etc.) error detection. UDP/IPv4 may actually have a checksum of 0, indicating no checksum; IPv6 requires UDP to have its own checksum. Therefore, IPv6 routers do not need to recompute a checksum when header fields (such as TTL or hop count) change.
- The TTL field of IPv4 has been renamed to Hop Limit, reflecting the fact that routers are no longer expected to compute the time a packet has spent in a queue.
- The IPv6 defined subnet *anycast addresses* (RFC 2526) , that is assigned to one or more network interfaces (typically belonging to different nodes), with the property that a packet sent to an anycast address is routed to the "nearest" interface having that address, according to the routing protocols' measure of distance.

The IPv6 specification defines Extension Headers:

- Routing Header - Similar to the source routing options in IPv4. Used to mandate a specific routing.
- Authentication Header (AH) - A security header which provides authentication and integrity.
- Encapsulating Security Payload (ESP) Header - A security header which provides authentication and encryption.
- Fragmentation Header - The Fragmentation Header is similar to the fragmentation options in IPv4.
- Destination Options Header - This header contains a set of options to be processed only by the final destination node. Mobile IPv6 is an example of a Destination Options Header.
- Hop-by-Hop Options Header - A set of options needed by routers to perform certain management or debugging functions.

A new version of ICMP has been defined for IPv6 (RFC 4443). ICMPv6 added new types and codes required by the new IPv6 functionality.

These include the "Packet TooBig" type and an "unrecognized IPv6 options" error codes. It also includes some group management protocol.

After the IPv6 has been used, the IPv4 are still using. Some methods are used to handle that situation.

Dual-stack (or native dual-stack) refers to side-by-side implementation of IPv4 and IPv6. That is, both protocols run on the same network infrastructure, and there's no need to encapsulate IPv6 inside IPv4 or vice-versa (using tunneling). Dual-stack is defined in RFC 4213.

Although this is the most desirable IPv6 implementation, it is not always possible, since outdated network equipment may not support IPv6. Some network equipment (such as a CMTS) or customer equipment (like cable modems) may require software updates or hardware upgrades to support IPv6. This means cable network operators must resort to "tunneling" until the backbone equipment supports native dual-stack.

The basic idea of tunneling is that suppose an IPv6 datagram has to go through a router that only support IPv4, then the whole datagram is put into the data field of an IPv4 datagram. The source address of the IPv4 datagram will be the last IPv6 supported router and the destination address is the next IPv6 supported router. Between these two routers, a IPv4 tunnel is formed.