WikiJournal of Science, 2022, 5(1):2
doi: 10.15347/wjs/2022.002
Review Article

A Survey on Internet Protocol version 4 (IPv4)

Michel Bakni,[1,i] Sandra Hanbo[ii]

Abstract

Internet Protocol version 4 (IPv4) is an internetwork protocol that is active at the internet layer according to the TCP/IP model, it was developed in 1981 within a project managed by Defense Advanced Research Projects Agency. In the following years, the use of IPv4 grew to dominate data networks around the world, becoming the backbone of the modern Internet. In this survey, we highlight the operation of the protocol, explain its header structure, and show how it provides the following functions: Quality of service control, host addressing, data packet fragmentation and reassembly, connection multiplexing, and source routing. Furthermore, we handle both address-related and fragmentation-related implementation problems, focusing on the IPv4 address space exhaustion and explaining the short and long terms proposed solutions. Finally, this survey highlights several auxiliary protocols that provide solutions to IPV, namely address resolution, error reporting, multicast management, and security.

Keywords: IPv4, Internet architecture, Internetworking, and data packet

Introduction

In the 1970s, several independent data networks were being developed in the United States and Europe, each has its local protocols and characteristics.[1] These networks need to connect, in order to exchange data, thus, the question of internetworking was arising. In this context, the Internet protocol version 4 (IPv4) was developed enabling networks to mutually exchange data blocks called the packets or datagrams.[2]

This paper is a survey on IPv4, its operations, and its relationships to other network protocols and services. The rest of this survey is divided as follows. First, we show a brief historical background on the origin of the protocol. Then, a general overview of the operation of the protocol is carried out. Next, the protocol header and its structure are fully described. After that, a detailed section is dedicated to explaining the functionalities of IPv4. This is followed by problems encountered after the protocol was implemented. Finally, the auxiliary protocols section highlights major protocols and technology used side by side with IPv4.

[1] ESTIA institute of technology
[i] m.bakni@esita.fr
ORCID: 0000-0003-2963-8799
[ii] sandra.hanbo@gmail.com
ORCID: 0000-0001-8821-0608

Received 11-12-2021; *accepted* 09-12-2022

Historical Background

After WW II was over, analog communication was dominating. To make a communication channel between two ends, a physical connection needs to be established, creating an electrical path that the data signal will follow. This operation requires switching efforts in order to allocate channels and maintain the electrical circuit close. The switching was taking place in the switching nodes, initially in manual methods and then automatically. This technology is referred to as circuit-switched telecommunication.[3]

In the early 1960s, Paul Baran, working at the RAND Corporation, was the first to observe the advantages of digital communications in terms of reliability and

WikiJournal of Science, 2022, 5(1):2
doi: 10.15347/wjs/2022.002
Review Article

throughput.[4] Later in 1964, Baran suggested a new method to create communication channels called the packet switching. Instead of using a central model where allocating, establishing, managing, and closing physical channels is managed in the core of the network, Baran suggested a distributed model with no allocated physical channels. As an alternative, Baran created a new concept to exchange data and call it the "Block message". According to Baran, network terminals create the block messages based on the need, and then send them into the network. Each message is 1024 bits long and can include only text data. Baran suggested that terminals add information to the data to be sent, including a code for the message source, and the destination address as well as indicators for the start and the end of the message (Figure 1).[5] The additional information was called: Housekeeping information. In the following years, the block message and the house keeping information were to become the data packet and the header respectively.[6]

The development of packet switching technology in the 1960s was the first major step towards Internet Protocol (IP). In a packet-switching network, digital information is carried in small chunks called packets, each of which contains a fixed number of bytes. From its source to its destination, sending one packet is totally independent of sending other packets sharing the same channel.[7]

In 1973, Louis Pouzin created CYCLADES, the first packet-switched network that successfully adopts the concept of datagram for the first time in the history of data networks. According to Pouzin, a datagram contains all the information needed to define its source and the final destination.[8] Based on that, data networks started to support connectionless channels instead of

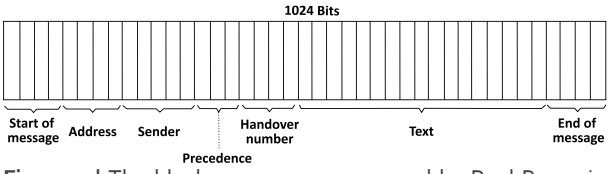


Figure 1 | The block message as proposed by Paul Baran in 1964.

the traditional connection-oriented telecommunication channels. In the connection-oriented mode, predetermined channels that connect the source of the data with its destination are needed, and establishing these channels is mandatory for data transmission to become possible. On the other hand, in the connectionless-mode, packets are routed based on source and destination addresses carried by the packets themselves. The datagrams were "eagerly embraced" by the designers of the early Internet, and this resolution had deep effects on the development of the other network protocols.[9]

In 1974, a paper titled "A Protocol for Packet Network Interconnection" was published by Bob Kahn and Vint Cerf.[10] This paper marks the starting point of the timeline shown in Figure 2, it describes a transport protocol to be activated between hosts in a packet-switching network. The suggested protocol provides several services related to data packets including flow control, process addressing, and end-to-end error checking. The protocol was called Transmission Control Program (TCP) and it was described in a specific Request for Comments (RFC) which has the codename "RFC 675".[11]

Later, the functions of the program were divided into two separated protocols: Transmission Control Protocol (TCP)[a] and IP. The two protocols were developed within a project supported by the Defense Advanced Research Projects Agency (DARPA). Regarding IP, between 1977 and 1979, several experimental versions of the protocol were released. During this period, many Internet Experiment Notes (IENs) that describe the IP versions before the official standard were released:

- IEN2, dated: August 1977, titled: "Comments on Internet Protocol and TCP". it showed the need to divide functionalities of the transmission control program into two separate protocols. In three years, the two protocols will be IPv4 and TCP. IEN2 also proposed the first version of IP

WikiJournal of Science, 2022, 5(1):2
doi: 10.15347/wjs/2022.002
Review Article

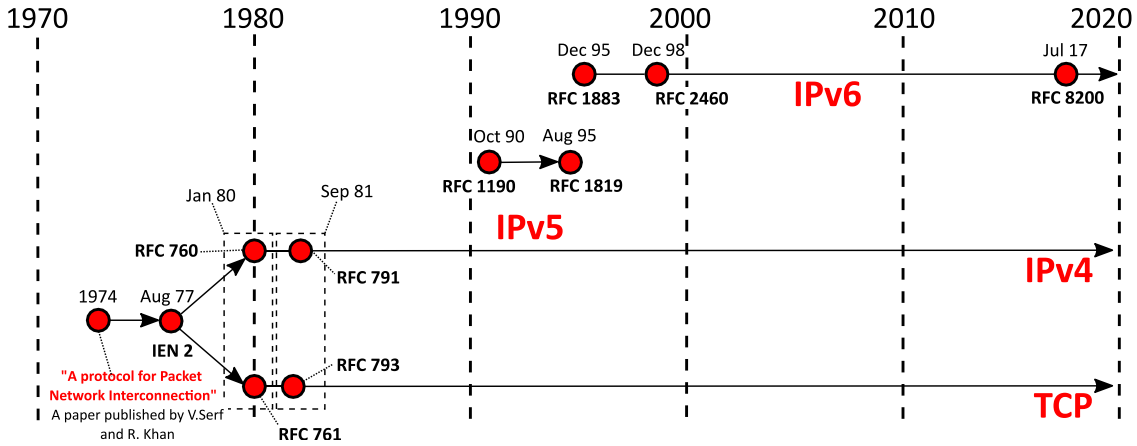


Figure 2 | Timeline for the development of the Transmission Control Protocol and Several versions of the Internet Protocol.

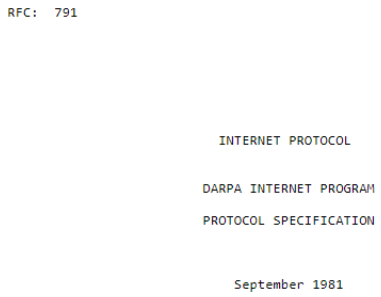
and used a version number equal to 0 to identify it in the protocol header.[b][13]

- IEN 26, February 1978, "A proposed new internet header format". It described version 1 of the protocol (IPv1).[14]
- IEN 28, February 1978, "Draft Internetwork Protocol Specification version 2". It described version 2 of the protocol (IPv2).[15]
- IEN 41, June 1978, "Internet Protocol Specification version 4"[c]. This was the first IEN to describe IPv4, however, the header structure was different from the current protocol.[16]
- IEN44, June 1978, "Latest header format". It summarized the edits on the header protocol reported in IEN41.[17]
- IEN 54, September 1978, "Internetwork Protocol Specification version 4". This note included a description identical to the structure adopted later by the official standard of the protocol.[18]

Later in 1980, IEN 128 was given the code name RFC 760 to become the first RFC dedicated to the internet protocol.[19] In September of the next year, RFC 791 was published under the title: "Internet Protocol" (Figure 3). Since then, it is the official standard of IPv4. In the next two years, the protocol had been gradually adopted in the United States, and ended as the main internetwork protocol in ARPA network starting from the 1st of January 1983.[20][21]

Internet Protocol version 5 (IPv5) was developed under the name: "Internet Stream Protocol", but it did not exceed the experimental stage.[22] Additionally, between 1988 and 1993, when IPv4 address space was being rapidly exhausted, a new version of the protocol was developed as a response, it was arbitrarily named IPv7 as its developer stated. However, the project was completely abandoned by 1993.[23]

Internet Protocol version 6 (IPv6) is the successor of IPv4. It is essentially developed to answer the IPv4 address space exhaustion problem. Whereas an IPv4 address is 32 bits long, providing a space that includes 4.3×10^9 addresses only, an IPv6 address is 128 bits long, supplying 3.4×10^{38} addresses. IPv6 was firstly described in RFC 1883.[24] Then, many modifications were included and a new standard was released in 1998 under the code name RFC 2460.[25]



RFC: 791

INTERNET PROTOCOL

DARPA INTERNET PROGRAM

PROTOCOL SPECIFICATION

September 1981

Figure 3 | The first page of IPv4 standard (RFC 791).

WikiJournal of Science, 2022, 5(1):2
doi: 10.15347/wjs/2022.002
Review Article

Lastly, in 2017, RFC 8200 was issued to cover amendments made in the past 20 years.[26] Although its address space is exhausted and it is, slowly and surely, replaced by IPv6, Internet Engineering Task Force (IETF) intends to continue fully maintaining and supporting IPv4 as well as continuing the promotion of IPv6 encouraging people to use it.[27]

On 1 April 1994, IETF published an RFC confirming the development of IPv9. However, it was an April Fool.[28]

Operation

Implementation

The main objective of the internet protocol is to allow applications, running in nodes, to exchange data packets via the network. In order to run IPv4, each node must support an IP module at the network layer of its protocol stack. Figure 4 shows the position of the network layer in the TCP/IP model, where it is located under the transport layer and above the data link layer. When an IPv4 node is sending data, the IP encapsulates the Protocol Data Unit (PDU) coming from the transport layer and passes the results to the data link layer. When the node is receiving data, the IP module

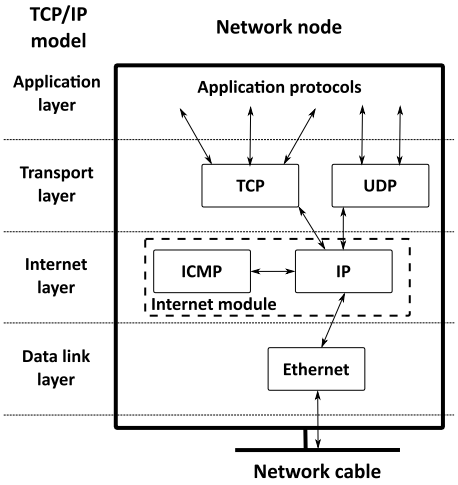


Figure 4 | Basic TCP/IP network node.

accepts PDUs of the link layer, then, the module decides which protocol will be the next to process data, either one of the protocols in the transport layer or another protocol running in the network layer. In both cases, the IP decapsulates the PDU, removing the IP header, then, delivers the result to the next protocol.[29][30]

If two nodes, referred to as data source and destination, are using the same link protocol, they can exchange data packets directly. However, if the source uses a link protocol that differs from the link protocol used in the destination, then, a device supporting the two link protocols is needed. This device is called the gateway, and it is used to forward the packet from the source to the destination.[31] To achieve that, the gateway has a single IP module that connects with the two nodes via two different link modules. In this case, the IP module will have two IP interfaces, each will be numbered according to the IP address subspace used in the network where the corresponding node is located.[32] Figure 5 shows the model of the network where IPv4 was first implemented, it is called the Advanced Research Projects Agency NETwork (ARPANET). In Figure 5, two hosts are connected via a gateway, each host supports:

1. TCP as a transport protocol to provide host-to-host communication via a virtual channel.
2. IPv4 as an internetwork protocol to provide host-to-gateway communication.
3. Two different link protocols, namely LN1 and LN2, each active in a separate segment of the network.

Functionalities

From a functionality perspective, IPv4 belongs to a family of network layer protocols called the internetwork

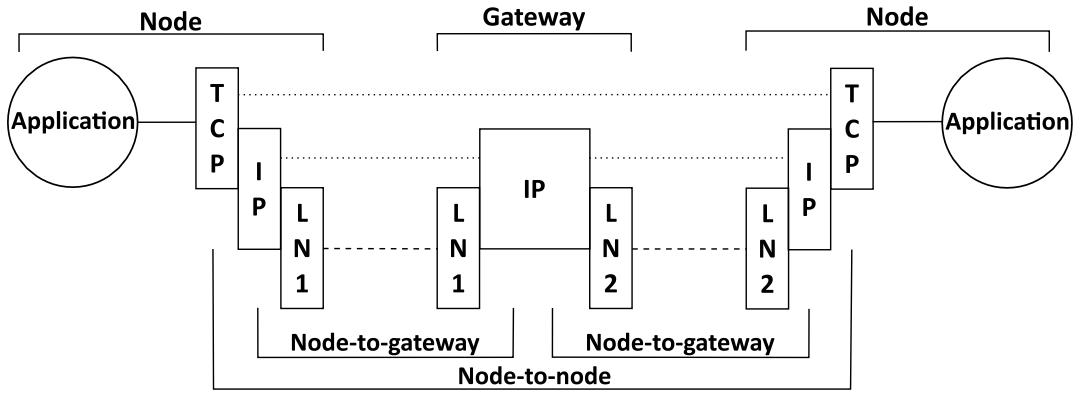


Figure 5 | ARPA model for a transmission path between two hosts and a gateway.

protocols.[2][33] Of this family's functions,[34] IPv4 provides:

- A mechanism to determine the Quality of Service (QoS) needed for each data packet separately.
- A space of digital addresses and its structure, each of which is called an internet address. Any entity in the network that supports IPv4 needs to host at least one IP address, thus, it is called a host. This functionality is called addressing, and IPv4 supports two types: classful and classless. In the classful addressing, the address space is divided into a set of groups that includes a predetermined number of addresses, each group is called a class. Addresses that belong to the same class share the same structure. On the other hand, in the classless addressing, there are no specified-length classes, and the address space is divided flexibly as needed.[35]
- Fragmentation and reassembling. If the node is sending data, IPv4 can fragment it into smaller pieces if needed. The same process can take place in any router that handles the packets on the way to their destination. Fragmentation happened when a packet's length is greater than the Maximum Transmission Unit (MTU) of the next network where the packet is to be routed. On the contrary, the reassembling can only be done at the final destination. the process aims to reconstruct the original data packet as it was before the fragmentation.[36]
- A mechanism to multiplex data from different applications together at the packet source and to demultiplex them at the destination.[37]
- An optional function to provide source routing. It is a routing mechanism that provides the source of the packets with the ability to determine an optional or mandatory route for the packets they create. If this function is used in the optional mode, the routers are recommended to use the route specified by the source. If it is used in mandatory mode, the routes must forward the packet in the route specified by the source. If this is not possible, the packet must be discarded.[38]

The previous functions will be discussed in detail in the functions section below.

IPv4 has limitations, it cannot provide the following internetwork-related functionalities:

- IPv4 does not provide host auto-configuration. The configuration of hosts' IP addresses can be achieved either manually or automatically through a dynamic configuration protocol,[39] such as Dynamic Host Configuration Protocol (DHCP).[40]
- If the packet's path includes more than a gateway, IPv4 is not able to route data packets alone. In this case, a routing protocol is needed to exchange routing information between gateways, thus, paths between IPv4 hosts through the network can be established.[41]
- IPv4 does not provide a reliable data transfer service because it uses connectionless channels. If

an error occurred while transferring data packets, the lost data cannot be restored. However, reliable transport protocols, such as TCP, can be used above IP to provide reliable data transferring service using connection-oriented channels.[42]

- IPv4 does not have any kind of flow control.[2]
- IPv4 does not have any built-in security mechanisms.[43]

Header Format

IPv4's PDU is called a packet, it consists of a header and data payload. The header length varies between 20 and 60 bytes, and the payload length can grow up to reach around 65 thousand bytes.[44]

Header fields can be classified into two types of fields: permanent and options. The length of permanent fields is 20 bytes. An IPv4 packet might include no options, however, if it has any, their maximum allowed length is 40 bytes. The structure of the IPv4 packet is shown in Figure 6.[45]

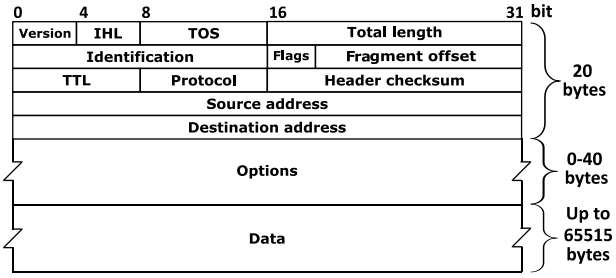


Figure 6 | Data packet structure for IPv4.

Permanent Fields

IPv4 header contains 12 permanent fields, its structure is displayed in Figure 6. In the following, there is a brief description for each field:[46]

- Version: the length is 4 bits, and the value of this field is always set to 4 in all IPv4 packets.
- Internet Header Length (IHL): 4 bits, it determines the end of the header and the start of the payload. The value represents the number of 32-bit (4 bytes) words in the header. The minimum acceptable value for this field is 5, which corresponds to the minimum header length (20 bytes).
- Type of Service (ToS): 8 bits, it contains a code used to describe the QoS needed while the packet passes via the network. The parameters used to configure QoS are the precedence, delay, throughput and reliability. The structure of this field was first defined in the RFC 791, then a new structure and mechanism to describe QoS was introduced in RFC 1349,[47] and later in RFC 2474.[48]
- Total Length: 16 bits, it defines the length of the packet in bytes. The maximum allowed value is 65535.
- Identification: 16 bits, it is used to uniquely distinguish a packet and all its fragments resulted from the fragmentation. This field helps the protocol module in the destination to detect all fragments, and reassemble them producing the original data packet again.[49]
- Flags: 3 bits, this field contains one reserved bit always set to zero and two flags: Don't fragment and more fragments (Figure 7). The first flag is used to prevent fragmentation under all circumstances (when set to 1). The second flag is used to distinguish the last fragment resulting from the fragmentation of a packet (if set to 1). These flags are only used if the packet was subjected to fragmentation.

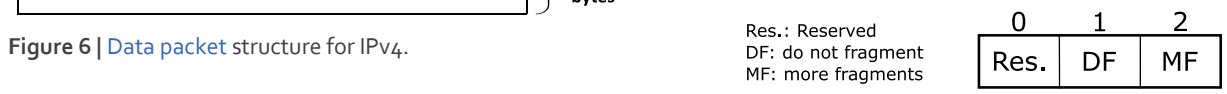


Figure 7 | The structure of Flags field in the IPv4 header.

- Fragment Offset: 13 bits, this field is set only if fragmentation is used. It contains the relative position of the fragment to the start of the original data packet before fragmentation. The fragment offset field helps to reassemble fragments correctly in the destination. The value of this field is equal to the real offset divided by 8. Thus, if this field contains 1, then, the real shift is 8 bytes. The maximum allowed value for the shift is 2^{13}=8192 which represents 65536 Bytes.[50]

- Time To Live (TTL): 8 bits, it is set by the source of the packet, and contains the maximum number of hops the packet can do. Each node processes the packet, such as routers and gateways, checks the value of this field first. If it is equal to zero, the packet is to be discarded. If not, the node subtracts one from the value of this field continues processing the packet.
- Protocol: 8 bits, it is used by the IPv4 multiplexing mechanism. The field contains codes used to define the protocol that is going to process PDU next. The codes are standardized by Internet Assigned Numbers Authority (IANA).[51]
- Header Checksum: 16 bits, it contains the output of the checksum algorithm that was applied only to the header fields. In the destination, the IP module recalculates the header checksum and compares it to the value of this field. If they matched, the header is not corrupted. RFC 791 explains the algorithm used to calculate the value of this field,[52] it must be applied to recalculate this field every time a change in the header is taking a place. For example, decreasing the value of the TTL field.
- Source address: 32 bits, it contains the IPv4 address of the sender of the packet which is called the packet source.
- Destination Address: 32 bits, it contains the IPv4 address of the packet destination.

Options

IPv4 header might include facultative fields called the options. The existence of these fields within the header is optional, however, the support of the options is mandatory in all IPv4 implementations. There is no fixed length for the options field in an IPv4 packet, it might include no options, one or more with a maximum length of 40 bytes.[53] The options field must always end at the boundary of a 32-bit word. If this is not the case, the missed bits will be completed with padding bits.[54]

IANA maintains a standardized record of IPv4 options.[55] All the options, except two, have a TLV structure, Figure 8 shows this structure which includes three fields:[53]

- Type: 8 bits, it consists of three subfields:
  - Copy flag: 1 bit, it is used with fragmentation to determine whether the option used is to be copied to all fragments (C=1) or not (C=0).
  - Class: 2 bits, it is used to indicate the functionality of the option: $(00)_2$ for "control" or $(10)_2$ for "debugging and measurement".
  - Number: 5 bits, it is a unique numerical value to distinguish each option.
- Length: 8 bits, it contains the length of the options field in bytes.
- Value: variable length, it is specified by the type of the option.

The two exceptions that do not follow the previous TLV structure are:[56]

- End of options list: 8 bits, $(00)_{16}$, it is used to mark the end of the options field.
- No operation field: 8 bits, set to $(01)_{16}$, it sits in-between two options for the purpose of separation, when the header has more than one option.

IPv4 options are rarely used because they can be a basis for launching several attacks.[57]

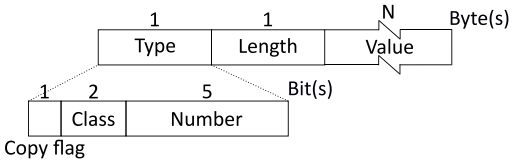


Figure 8 | IPv4 option general structure.

Functions

In this section, we discuss the following functions of IPv4:

1. Quality of service Control.
2. Addressing hosts and groups.
3. Data packet fragmentation and reassembly.
4. Multiplexing and demultiplexing of the connections of higher-layers protocols.
5. Source routing.

WikiJournal of Science, 2022, 5(1):2
doi: 10.15347/wjs/2022.002
Review Article

Quality of Service Control

When IPv4 is used, QoS is set using the ToS field in the protocol header, the original structure of this field is shown in Figure 9. RFC 791 had reserved three bits in the ToS field to determine the precedence of the packet that contains the header, and the standard defines 8 codes for these bits, each refers to a specific level of precedence. In addition to that, the standard specified the QoS of services needed for each packet in terms of three elements: delay, throughput and reliability. The mechanism is implemented by reserving a bit for each element in the ToS field creating three subfields. For each element, if the corresponding bit is set to 0, this indicates that the packet accepts regular delay, a normal throughput and regular reliability respectively. On the other hand, setting these bits to 1 indicates the need for the low delay, high throughput and higher reliability respectively. By choosing different combinations for the previously described bits, the protocol allows the making of a trade-off between the QoS elements.[58]

In 1998, Differentiated Services (DS) were introduced by defining a scalable architecture for classifying and managing data within the network.[59][60] Based on that, the ToS field was also restructured, merging together the previously mentioned subfields. As a result, the Differentiated Services CodePoint (DSCP) subfield was created (Figure 10).[61] DS depends on categorizing data packets into a specific number of classes and then tagging the packets with the class unique identifier. After that, the routers are to be configured to recognize the QoS needed for each packet based on the tag it carries. When DS is applied, the routers at the edge of the network classify the packets, while the core routers only process the packets, so that supporting QoS at the core of the network remains fast and simple.[62]

The network, where the previous set of routers are found, is called the differentiated services domain.[63] Any router in the DS domain handles each incoming packet independently from other routers, and that is why this mechanism is called Per-Hop Behavior (PHB). Four standard types of router's behavior are defined:[64]

- Default Forwarding (DF) behavior, it is the choice selected by a router processing a data packet when no other behavior fits that packet. Supporting DF behavior is mandatory on all routers that support DS. When this behavior is selected, the value of the DSCP subfield is to be set to $(00000)_2$.
- Expedited Forwarding (EF) behavior, it is the behavior corresponding to real-time applications such as sound and video. The provided QoS, when this behavior is chosen, satisfy requirements of low delay, low jitter and low data loss. If this behavior is selected, the value of the DSCP subfield is to be set to $(10110)_2$[65]

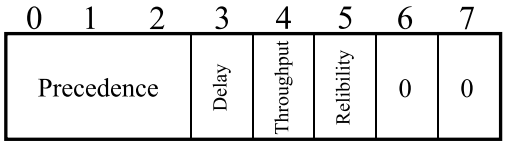


Figure 9 | The structure of the ToS field in the IPv4 header according to RFC 791.

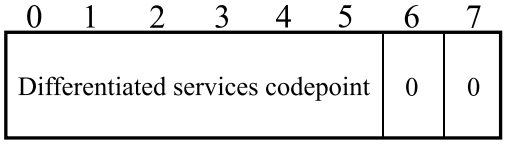


Figure 10 | The structure of the ToS field in the IPv4 header according to RFC 2474.

- Assured Forwarding (AF) behavior, it allows packets delivery as long as the data traffic does not exceed a specified threshold. If it does, the probability of discarding packets increases on a three-zone scale: low, medium and high. In order to provide users with configuration options, there are four different classes applies for each of the previously mentioned zones ending with a total of 12 possible levels to configure. Table 1 provides a recommended set of values of the DSCP subfield for all the classes if the AF behavior is selected.[66]
- Class Selector (CS) behavior, it provides a way to compatible support for the original ToS precedence subfield. If this behavior is selected, the value of the DSCP subfield is $(b_0b_1b_2000)_2$, where b_0, b_1 and b_2 are the precedence bits respectively.

Table 1 | Recommended codepoints for the AF Behavior (in binary)[66]

Drop precedence	Class 1	Class 2	Class 3	Class 4
Low	001010	010010	011010	100010
Medium	001100	010100	011100	100100
High	001110	010110	011110	100110

Regarding bits 6 and 7 of the ToS, the two bits are used for the Explicit Congestion Notification (ECN), which was first described in 1999.[67] When set to $(11)_2$, the two bits provide a mechanism for routers to notify the destination that congestion currently exists in the network. Three other combinations of the two bits can be used by the data source as follows: $(00)_2$ to indicate that the mechanism is not used, and $(10)_2$ or $(01)_2$ to indicate the use of it.[68]

Addressing

IPv4 addressing is giving digital identifiers to IP hosts residing in a local area network or on the internet.[69] The digital identifier is called an IP address, it may be used to uniquely distinguish a specific host, or to identify members of a group each of which is hosting that address at the same time.[70]

Structure development

The structure of the addressing system was discussed early when IPv4 was being developed. In 1973, Pouzin argued that the structure should allow 2-level addressing: a short address that uniquely identifies the host within the local network and a long address that identifies the host within the whole network. this structure was based on the idea that a network might include thousands of hosts. Pouzin also proposed a format that includes a hexadecimal notation for addresses that can have 3 to 14 positions.[71] Although the proposed format was not adopted later in IPv4, the 2-level addressing scheme was a core concept in the IPv4 addressing structure.[72]

In the next year, Kahn and Cerf proposed the Transmission Control Program (TCP). In the addressing section, the authors proposed a two-fields address structure: network (8 bits) and identifier (16). This structure allows only 2^8=256 unique networks to be able to connect to the system, with a maximum of 2^{16}=65536 hosts in each of which.[73] Although this structure was modified later, the 8-bit length of the network part was adapted for IPv4. David Clark, in 1978, referred to this choice as a limitation for the internet growth writing the followings: "We should thus begin to prepare for the day when there are more than 256 networks participating in the internet."[74]

The initial version of IPv4, documented in IEN 123 (December 1979) and later RFC 760 (January 1980), did not consider these concerns. Although it extended the address to include 32 bits instead of 24, the proposed structure has two fixed-sized fields: network (8 bits) and hosts (24).[75] However, by September 1981, when RFC 791 was released, the address structure was modified to include 3 different modes called classes, hence the name classful addressing. The new addressing scheme allowed the following variation in lengths for network and host fields: (8,24), (16,16) and (24,8).[45] A detailed structure for the classes was illustrated in RFC 796. [76]

IPv4 Address

It is a 32-bit digital identifier, normally written Dot-decimal notation. However, it might also be written using the Binary numeral system. The IP address is divided into four parts, each of which is 8 bits long, thus, it is called an octet. The enumeration of the octets starts with one, and the first octet includes the Most Significant Bit (MSB) (Figure 11).[77]

When the dot-decimal notation is used, the IPv4 address follows the format: "#.#.#.#", where '#' represents a numerical value in the decimal numeral system. Because each octet is 8 bits long and contains only positive integers, the value in each octet varies between 0 and 255.[78][79] For example, 10.0.0.1, 172.16.254.1 and 240.0.0.9 are IP addresses written in the dot-decimal notation.

IPv4 addresses can be represented using dot-binary notation, this can be done by converting the value of each octet from the decimal numeral system into the binary. For example, 10.0.0.1 can also be written as follows: 00001010.00000000.00000000.00000001. Although an IPv4 address has two representation forms, the two

cannot be used together, thus, when written down, one notation must be used.

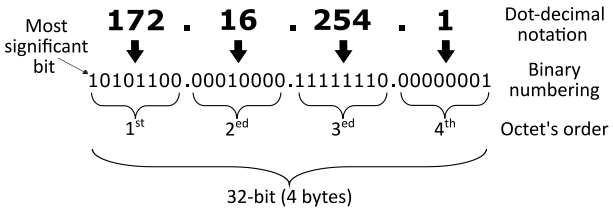


Figure 11 | A diagram of an IP address (IPv4).

IPv4 Address Space

It is the set of all IP addresses. The space includes 2^{32} addresses, approximately 4.3 billion. Based on the value of the first octet, the IPv4 address space is divided into three smaller subspaces as follows (Figure 12):

- Unicast address subspace: it covers 7/8 of the original address space and includes all addresses that have the first octet ranging between 0 and 223, regardless of the values of the other octets. These addresses are used to identify unicast hosts in data networks, i.e., they can be used to as source or destination addresses in one-to-one transmission. The unicast address subspace was further divided based on the addressing method (classful or classless). Classful addressing is the original method described in the IPv4 standard. However, it was later obsoleted and replaced by the classless method due to the rapid exhaustion of IPv4 address space, the two methods are to be covered in the next section.
- Multicast address subspace: it covers 1/8 of the original address space and includes all addresses that have the first octet ranging between 224 and 240, regardless of the values of the other octets. These addresses are used to identify multicast hosts in data networks, i.e., they can be used to as destination addresses in one-to-many transmission. The unicast address subspace is also referred to as Class D.[80]

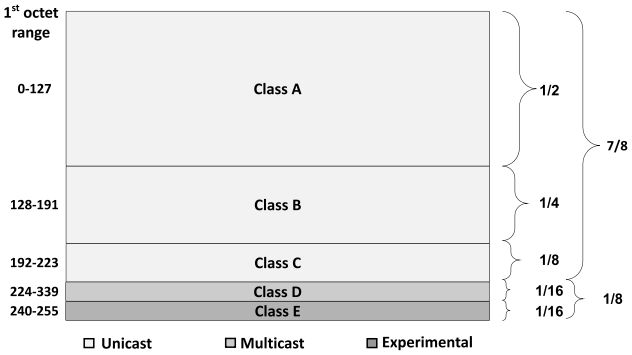


Figure 12 | IPv4 address subspaces.

- Experimental address subspace: an additional address subspace was reversed for reasons, it covers the remaining 1/16 of the original address space, it is also referred to as Class E.[81]

Addressing method

As mentioned in the previous section, IPv4 address space is divided based on two addressing methods: classful and classless. The first was the original method and the later was adapted later to overcome the IPv4 exhaustion problem. In this section, we discuss first the Classful addressing method, then, we present the classless addressing method. The IPv4 address space exhaustion problem is to be covered in the problems section.

Classful addressing

When the classful method was used, an IPv4 unicast address had the following structure (Figure 13):[82]

- Reserved bits, which cover b_{rsv} bits. This field is used to define the function and the size of the address subspace[d] to which the address belongs. The reserved bits field starts from the address's MSB covering one bit at least, and extends to the right including up to 3 bits at most.[e]
- Network identifier (NID), it is used to uniquely identify the subspace to which the address belongs. All addresses that reside in the same subspace have the same NID. Network identifier starts directly where the reserved bits end, its length, referred to as b_{NID} varies according to the number of resulted subspaces N_{spc}. The relationship between the two parameters can be mathematically written as follows: $N_{spc} = 2^{b_{NID}}$.
- Host identifier (HID), which uniquely identifies the IPv4 host. The value of this field is unique inside each subspace. Host identifier starts directly where NID ends, its length, referred to as b_{HID} varies according to the number of available addresses in the subspace N_{adr}. The relationship between the two parameters is written as follows: $N_{adr} = 2^{b_{HID}}$.[f]

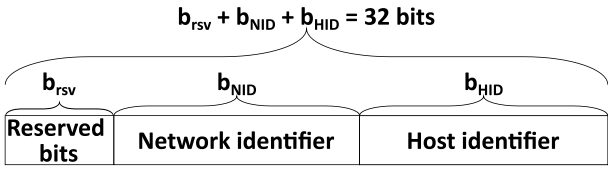


Figure 13 | IPv4 address structure used for unicast classful addressing.

When the classful addressing method was used, the unicast subspace was divided into 3 classes, Class A, Class B and Class C. Table 2 shows the length of each part of the addresses, the number of subspaces (N_{spc}) and addresses (N_{adr}) within each class. When the Internet was put to commercial use, the limited size of class C subspaces and the overwhelming size of class A subspaces led to rapid consumption of Class B subspaces and to the exhaustion of the IPv4 address space.

Classless addressing

The classless addressing was adopted in 1993 as a short-to-mid-term solution for the exhaustionIPv4 address space.[84] when used, there are no classes nor specific predetermined volumes for the subspaces, i.e., subspaces can be created based on the need.[85][86] When classless addressing is used, an IPv4 address will have the following structure (Figure 14):[87]

- A prefix, which is shared among all the addresses belongs to the same subspace. It covers b_{pfx} bits starting from the address's MSB and has no predetermined length. Instead, it can extend longer in the address creating larger subspaces as needed.
- An HID, which starts from the end of the prefix and covers all the remaining of the address.

Table 2 | Standard classes for IPv4 unicast addressing

Class	1st octet range				Field's lengths (bits)			N_{spc}	N_{adr}
	Binary		Decimal						
	From	To	From	To	b_{rsv}	b_{NID}	b_{HID}		
A	00000000	01111111	0	127	1	7	24	2^7	2^{24}
B	10000000	10111111	128	191	2	14	16	2^{14}	2^{16}
C	11000000	11011111	192	223	3	21	8	2^{21}	2^8

Classless addressing requires a hierarchical addressing system,[88] which means that the address space, independently but subsequently, is being divided into smaller subspaces. At each level of the hierarchy, the prefix length increases at the expense of HID (Figure 15).

Because IPv4 addresses are binary-based numbers, all the previously mentioned subspaces must always include a number of addresses that is multiple of 2, i.e., the number of addresses with the subspaces is a member of the set $\{2,4,8,16,\ldots,2^{32}\}$.[83]

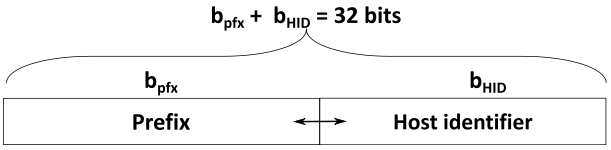


Figure 14 | IPv4 address structure used for unicast classless addressing.

No. of bits within the		No. of hosts per network	Newtwork mask		No. of equivalent classful addressing network		
Prefix	Host ID		Prefix notation	Dotted decimal notation	Class A	Class B	Class C
1	31	2147483646	/1	128.0.0.0	128	2^{-15}	2^{23}
2	30	1073741822	/2	192.0.0.0	64	2^{-14}	2^{22}
3	29	536870910	/3	224.0.0.0	32	2^{-13}	2^{21}
4	28	268435454	/4	240.0.0.0	16	4096	2^{20}
5	27	134217726	/5	248.0.0.0	8	2048	2^{19}
6	26	67108862	/6	252.0.0.0	4	1024	2^{18}
7	25	33554430	/7	254.0.0.0	2	512	2^{17}
8	24	16777214	/8	255.0.0.0	1	256	2^{16}
9	23	8388606	/9	255.128.0.0	1/2	128	2^{15}
10	22	4194302	/10	255.192.0.0	1/4	64	2^{14}
11	21	2097150	/11	255.224.0.0	1/8	32	2^{13}
12	20	1048574	/12	255.240.0.0	1/16	16	4096
13	19	524286	/13	255.248.0.0	1/32	8	2048
14	18	262142	/14	255.252.0.0	1/64	4	1024
15	17	131070	/15	255.254.0.0	1/128	2	512
16	16	65534	/16	255.255.0.0	1/256	1	256
17	15	32766	/17	255.255.128.0	1/512	1/2	128
18	14	16382	/18	255.255.192.0	1/1024	1/4	64
19	13	8190	/19	255.255.224.0	1/2048	1/8	32
20	12	4094	/20	255.255.240.0	1/4096	1/16	16
21	11	2046	/21	255.255.248.0	2^{-13}	1/32	8
22	10	1022	/22	255.255.252.0	2^{-14}	1/64	4
23	9	510	/23	255.255.254.0	2^{-15}	1/128	2
24	8	254	/24	255.255.255.0	2^{-16}	1/256	1
25	7	126	/25	255.255.255.128	2^{-17}	1/512	1/2
26	6	62	/26	255.255.255.192	2^{-18}	1/1024	1/4
27	5	30	/27	255.255.255.224	2^{-19}	1/2048	1/8
28	4	14	/28	255.255.255.240	2^{-20}	1/4096	1/16
29	3	6	/29	255.255.255.248	2^{-21}	2^{-13}	1/32
30	2	2	/30	255.255.255.252	2^{-22}	2^{-14}	1/64

Figure 15 | Table for prefixes used in classless addressing, and their equivalents in classful addressing.

Network Mask

It is a 32-bit number associated with the IPv4 unicast addresses. A network mask have the same 4-octet structure as any IPv4 address and is written using the same notation systems.[89] However, the mask has a special one-and-zero pattern: starting from its MSB, the mask includes only a sequence of ones, followed by a sequence of zeros, and the sum of the lengths of the two sequences is 32 bits. In both classful and classless addressing, the mask is used to detect the HID field of the corresponding network address. To do so, the bits in the sequence of zeros correspond to the HID field.[90]

For example, HID in a Class C network address is 8 bits occupying the 4th octet (Figure 11). The network mask that corresponds to this address is 1111 1111 . 1111 1111 . 1111 1111 . 0000 0000. However, to avoid writing long sequences of ones and zeros, the mask can be written as follows: /x, where x is the number of ones in the mask. For example, the previous network mask can shortly be written: /24, and that is the standard mask for all Class C subspaces. Table 3 shows the standard masks for unicast address subspaces when the classful addressing method is used.[g]

Address Space Management

Allocation and Assignment

The Internet needs a central authority to allocate and assign digital identifiers and to keep a reservation record. Since the beginning of the network, IANA, managed by the Information Sciences Institute in California, has provided these services.[92] However, starting from March 2000, IANA became a part of Internet Corporation for Assigned Names and Numbers (ICANN) which took control of allocation services on the Internet.[93]

IANA maintains global registers for both unicast and multicast subspaces. However, allocation differs as follows:

- Unicast subspaces allocation: following the classless addressing approach, the process is provided based on a four-level hierarchical model shown in Figure 16:[94]

1. IANA, residing at the top of the pyramid, provides the allocation service for several Regional Internet Registries (RIRs) based on geographical bases and following a three principles policy:[95]
   a. allocates RIRs subspaces distinguished with prefix /8.
   b. It is committed to allocating RIRs with subspaces that satisfy their future needs for at least 18 months.

Table 3 | Network masks used with classful addressing

Class	Mask Notation	
	Dotted decimal	Prefix
A	255.0.0.0	/8
B	255.255.0.0	/16
C	255.255.255.0	/24

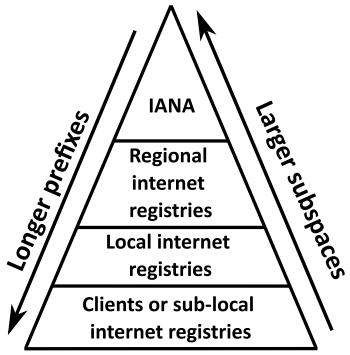


Figure 16 | The hierarchy of the unicast IP address allocation space in IPv4.

   c. It allows the RIRs to adopt their own allocation strategies and to keep their own reservation records.

2. RIRs, serve Local Internet Registers (LIRs), providing them with smaller subspaces, based on their needs.

3. LIRs assign subspaces to local internet sub-registers or directly to agents.

4. Local internet sub-registers assign subspaces to agents where subnetting and VLSM are used to locally manage the provided subspace.

Note that the allocation is distinguishable from the assignment in this context: while the first is providing Internet Service Providers (ISPs) with IPv4 subspaces, the second is giving subspaces on a customer base.[96] In addition to that, going down in the pyramid shown in Figure 16, the allocated/assigned subspaces will have longer prefixes and will, as a result, include a smaller number of addresses.[88] Figure 17 shows an example of an allocation process that started by providing the American Registry for Internet Numbers (ARIN) with a /8 prefix and ended with the client assigned a /27 subspace.

Some subspaces, usually referred to as blocks of addresses, are reserved addresses for specific protocols, or for special purposes.[97] In general, addresses from these subspaces should not be used to number hosts globally on the Internet. For example, 127.0.0.0/8 is reserved for the loopback function.[98] IANA maintains a registry for these blocks of addresses.[99]

- Multicast subspaces allocation: IANA is charged with the process of multicast subspace allocation and it maintains the IPv4 multicast space registry.[100] IANA allocates and assigns blocks of multicast addresses directly to newly developed protocols.[101]

Subnetting and VLSM

Subnetting is a mathematical operation used to logically divide an IPv4 unicast address space into two or more smaller subspaces called subnets, it can be used with both classful and classless addressing. Figure 18 shows how subnetting was being performed with the classful addressing approach: a new field, called Subnet IDentifier (SID) is created in-between NID and HID, it grows right to occupies one or more bits from HID, the number of SID bits (b_{SID}) determines how many subnets (N_{sub}) will result after performing the operation, the two are bonded together in the following formula: $N_{sub} = 2^{b_{SID}}$.[102][89] On the other hand, when the classless addressing method is used, the SID is created between the prefix and the HID following the same approach.[90]

WikiJournal of Science, 2022, 5(1):2
doi: 10.15347/wjs/2022.002
Review Article

The previous approach is called single-Level subnetting, when used, equal-volume subspaces will be created. If the subnetting algorithm is performed again on one subnet, new and smaller subspaces will be created, and this approach is called multi-level subnetting.

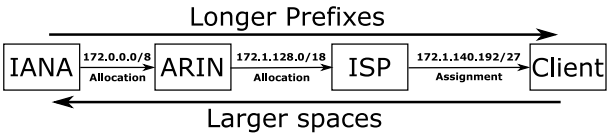


Figure 17 | Example of allocating and assignment of an IPv4 address space according to the allocation hierarchy.

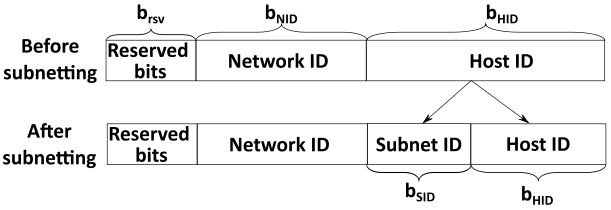


Figure 18 | Subnetting concept when used with classful addressing.

The result of the multi-level approach will be subnets that have different volumes and are distinguished with different masks.[103] Variable Length Subnet Mask (VLSM) is a special type of multi-level subnetting, it is performed on a single classful subspace. As shown in Figure 19, when compared to single-level subnetting, VLSM allows network administrators to efficiently manage their assigned subspaces based on the need.[104]

When creating new unicast addresses subspaces, two special addresses need to be considered in each subnet: the network and the broadcast, that is because they are the two extremities of the address range. The network address is the smallest address in the subspace and corresponds to all-zero HID, it is used to represent the address subspace in total. The broadcast address is the largest address in the subspace and corresponds to all-ones HID, it is used as a destination address to send packets to all hosts numbered using the addresses from the corresponding subspace. Both, the network and the broadcast addresses, should not be used to number hosts.[105]

Fragmentation and Reassembly

The idea of connecting different networks was a core concept of the internet, thus, there is no surprise that the word internet is coined from the expression: interconnected networks.[106] However, supporting different networks involves having different MTUs, leading to the question: How to transmit a data packet through a network, if it is greater than the MTU of that network? the answer is fragmentation and reassembly.[107][108]

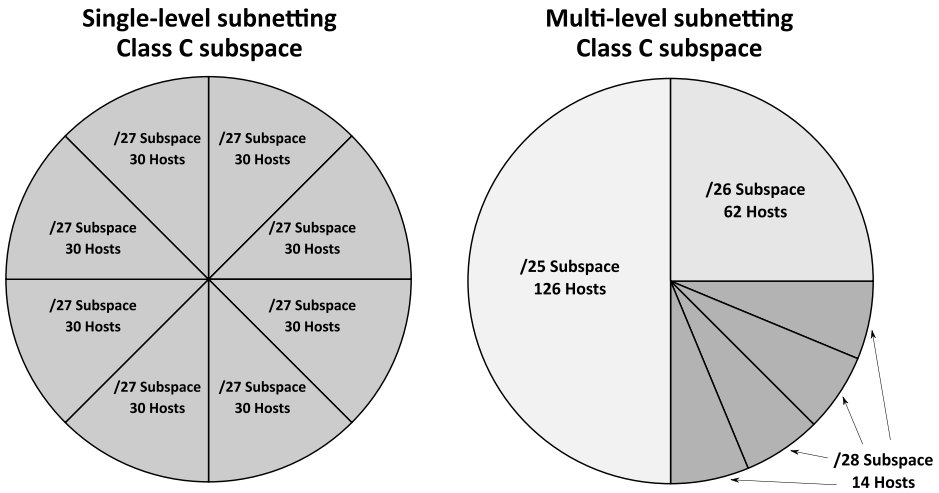


Figure 19 | Volumes comparison for subspaces resulted from subnetting a class C subspace. To the right: single-level subnetting, and to the left multi-level subnetting.

Fragmentation

When the IPv4 module receives a data packet to be sent to via network that is directly connected to the host where the module is located, it determines the value of the MTU associated with that network and compares it with the packet's length. If the packet's length is greater, then, the packet must be fragmented, each of which will become a new data packet independent from the original packet (Figure 20).[109]

IPv4 supports intranet fragmentation. This means that fragmentation occurs not only at the packet's source but also at any intermediate node that processes the packet along the path to its destination.[36]

Note that as fragments cross the networks to their destination, they can be fragmented creating smaller data packets if needed.[110]

Figure 21 shows the fragmentation algorithm as described in the IPv4 specification, it can be briefly summarized as follows:[111]

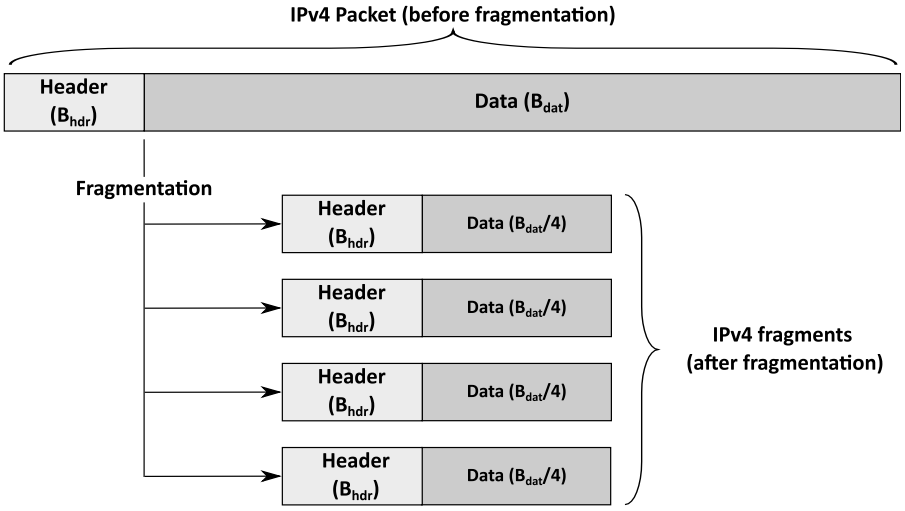


Figure 20 | Fragmentation example: an IPv4 packet is divided into smaller fragments.

1. In an IPv4 host, the IP module receives a routed data packet to be sent via a network directly connected to that host.
2. The module determines the packet's length and compares it with the MTU associated with the network via which the packet will be routed:
   a. If the length is greater, then, fragmentation is needed,
   b. Else, the packet is to be sent, as it is, to the next stage of the encapsulation, and the process is to be ended.
3. The DF flag in the packet is to be checked:
   a. If it is set, the packet is to be discarded, and the process is to be ended.
   b. Else, the fragmentation process starts as follows:
      i. The payload length of fragments is determined according to the MTU and the length of the IP header.
      ii. A part of the original payload, equal to the length determined in step (3.b.i) is sliced off to create the payload of a fragment.
      iii. A new IPv4 header is to be built for the sliced payload as follows:
         1. Calculating the length of the fragment header and adding it to the IHL field.
         2. Calculating the total length of the fragment and adding it to the Total Length field.
         3. Determining the lifetime of the fragment and adding it to the TTL field.
         4. Setting the value of the Identification field to the same value found in the Identification field in the header of the original data packet.
         5. Calculating the fragmentation offset and adding it to the offset field.
         6. Determining the value of the fragmentation flags: DF and MF and adding the values to the Flags field.
         7. Calculating the value of the checksum and adding it to the Checksum field.
         8. Copying the value of the source and destination addresses fields from the same fields in the header of the original data packet.
4. Generating the new IPv4 packet by encapsulating the payload of the fragment with the built header.
5. Sending the new packet to the next stage of encapsulating.
6. Determining whether the previous packet is the last packet by checking the value of MF:
   a. If it is set, the previous packet is the last, and the process is to be ended.
   b. Else, repeat starting from step (3.b).

Reassembly

The process of collecting fragments and using them to reconstruct the original data packet, as it was before the fragmentation, is called reassembly.[112] In IPv4, reassembly takes place only at the final destination of the packet. That is because routing for each fragment happens independently, and it is highly possible that fragments might be routed via different paths. Thus, reassembly in a given node on the path might be impossible due to the lack of all fragments.[113]

Figure 22 shows the reassembly algorithm as specified in the IPv4 specification, it can be briefly summarized as follows: [114]

1. In the final destination, the IP module receives a data packet from the data link layer.
2. The IP module checks if it is a fragment from an original packet or not:
   a. If it is not (Offset = 0 & MF=0), the packet is sent to the next stage of processing, and the process is to be ended.
   b. Else, reassembly starts as follows:

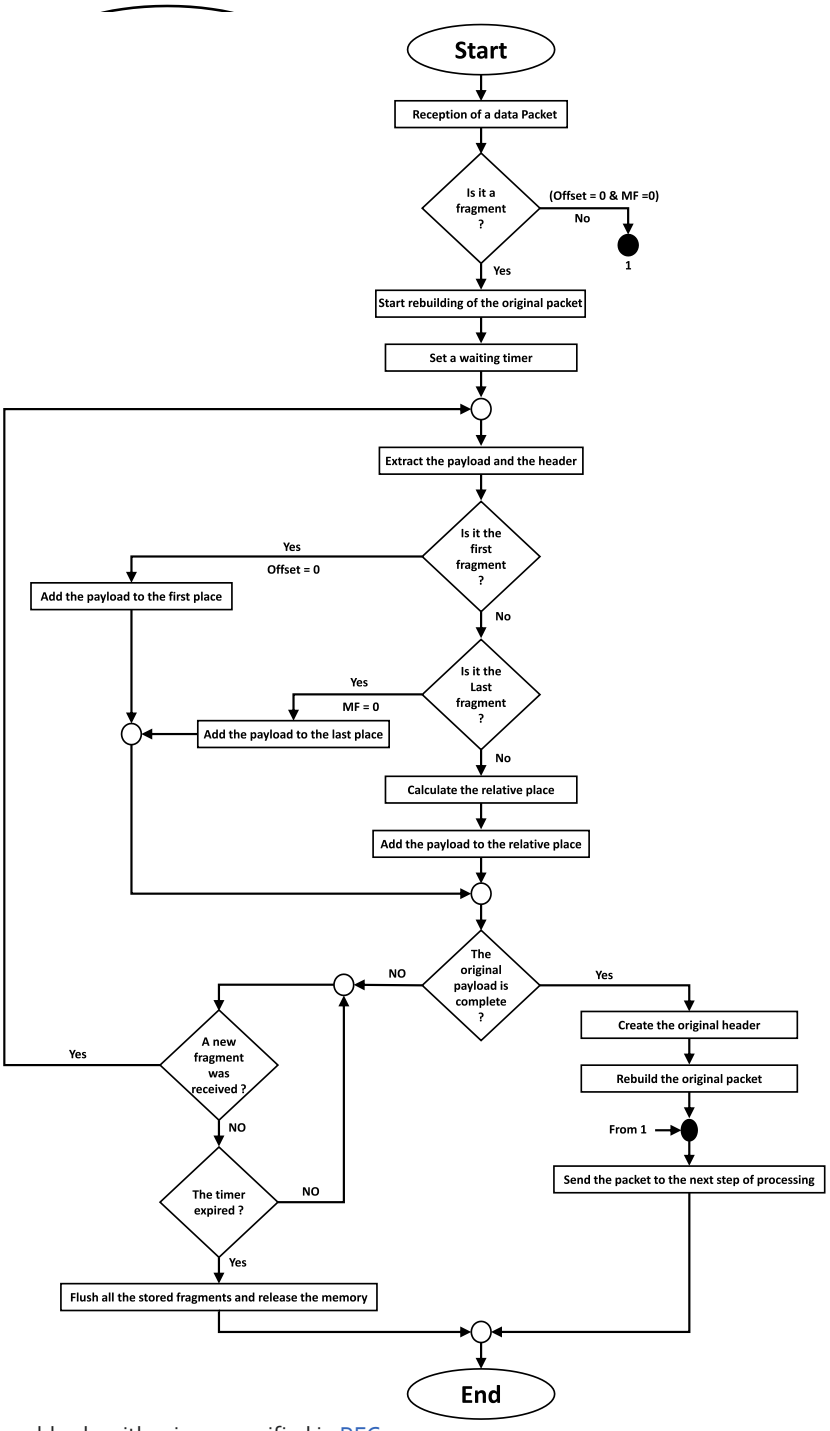


Figure 22 | IPv4 Reassembly algorithm in as specified in RFC 791.

i. Setting a hold timer and run it. If the timer is expired, reassembly is not possible, and all the collected fragments are to be discarded.
ii. Determining the payload and the header in the received fragment
iii. Determining the relative position of the fragment in the original packet,
    1. If it was the first segment (Offset = 0), then it will be added to the first position of the packet,
    2. Else, if it is the last segment (MF = 0), it will be added to the last section of the packet.
    3. Else, the segment is to be added to its relative position.
iv. Check if reassembly of the original packet payload is completed:
    1. If yes, the original packet header is created, the original packet is reconstructed and sent to the next stage of processing. The process is to be ended.
    2. Else, check for any newly received data packet that contains the same identifier value.
        a. If received, repeat starting from step (2.a).
        b. Else,
            i. wait until the hold timer expired, and periodically check for the arrival of data packets.
            ii. if a packet is received and the timer has not been expired yet, go to step (2.b.iv.2.b.i)
        c. If the timer expired, all collected fragments will be discarded, and the process is to be ended.

Multiplexing

Multiplexing is the ability to merge multiple protocol connections over a single connection, the reverse process is called splitting or demultiplexing (Figure 23).[115][30] IPv4 provides this function using the Protocol field in its header.[52] The field includes a value that specifies the following protocol in the encapsulation process, which may be another network layer protocol that performs a different function, such as Internet Control Message Protocol (ICMP), or a transport layer protocol such as TCP or a User Datagram Protocol (UDP).[37]

Source Routing

Source routing is a routing scheme in which the source of the data packet specifies the routing information for routers that are going to process the packet, the information includes a set of IPv4 addresses of network nodes. Routed to these addresses sequentially, an IPv4 packet can follow a route desired by its source.[115] IPv4 supports two types of source routing:[38]

- Loose Source Routing (LSR): it is called loose because the routers are not obliged to use the information in the option, and can route the packet based on their own routing scheme if needed.[116]
- Strict Source Routing (SSR): the word "strict" in the option's name reflects its mandatory nature as routers are obliged to route the packet based on the information carried in the option. If this is not possible, the routers are to discard the packet.[117]

The implementation of the previously described source routing types is achieved using two IPv4 options that have the same format as shown in Figure 24:[118]

- Type field, 8 bits long. It is set to 131 for LSR and to 137 for SSR.
- Length field, 8 bits. It includes the length of the option in bytes. The minimum accepted value for this field is 4.

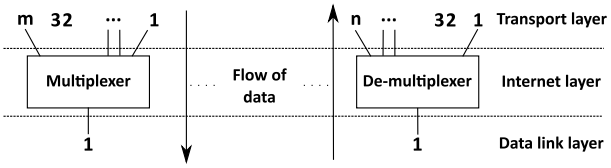


Figure 23 | m-to-1 multiplexing and 1-to-n demultiplexing for protocol connections at the network layer

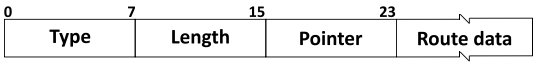


Figure 24 | Format of the two source routing options in IPv4.

- Pointer field, 8 bits. It refers to the start of an address in the next field. The specified address is to be used by the next router that is routing the packet. The minimum accepted value for this field is 4.
- Route data field, it has a variable length. This field includes a set of IPv4 addresses to be used for routing the packet, the order of the addresses is to be considered because they are to be processed sequentially as the packet is being forwarded from a router to another.

Problems

This section is dedicated to addressing problems encountered after IPv4 was implemented, especially problems related to addressing and fragmentation. Regarding the addressing, we are going to highlight the IPv4 exhaustion problem and the overlapping of address spaces. Concerning fragmentation, we will cover many related attacks and discuss the solutions.

Related to Addressing

Address Space Exhaustion

IPv4 address exhaustion is the depletion of free addresses in the space pool due to allocating to RIRs and ISPs or globally assigning to hosts on the Internet. The problem was first noticed in the early 1990s, while the Internet was rapidly expanding. Since then, solutions are being proposed and implemented.[84]

Answering the address exhaustion of IPv4 followed two strategies:[119][120]

1. A short-term strategy aimed to slow down the speed of the exhaustion and to extend the protocol lifetime as long as possible
2. A long-term strategy, aimed to replace IPv4, which has a limited number of addresses (around 4.3 billion), with another internetwork protocol that supports larger address space.

The short-term strategy was expected to prolong the lifetime of IPv4 for 3-5 years,[91] Nevertheless, in practice, the lifetime extended for more than 25 years. However, the exhaustion of the address space continued, slowly but surely. In February 2011, ICANN issued a press statement to report start using of the last free /8 block of the IPv4 address space.[121]

The two strategies were implemented in parallel, and, as shown in Figure 25, the result was:

- Two emergent solutions were developed as an answer to the short-term strategy:
  - Network Address Translation (NAT). It is a technology based on the idea that local networks can be numbered using the same address spaces, referred to as private addresses. NAT is a broad term dedicated to describing a set of mechanisms used to change network identifier values, such as IP addresses and port numbers, in data packets that pass across a

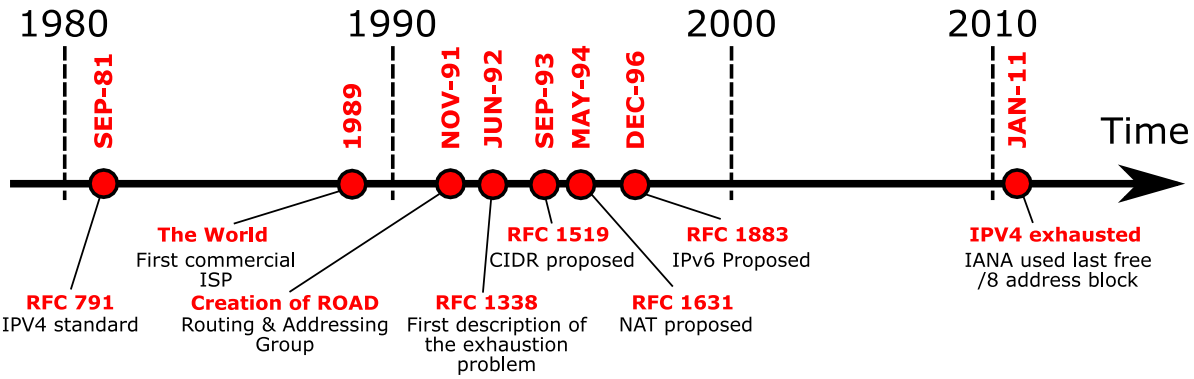


Figure 25 | IPv4 exhaustion problem timeline.

router that connect two domains: Local and global.[122]

Before NAT is applied, it requires the router to be preconfigured with pairs of identifiers: one from the global domain and the other from the local domain as well as a direction, which is used as a condition allowing only packets that satisfy it to start a NAT Session. After being preconfigured, the router monitors the movements of the packet between the domains (Figure 26). If the router found a packet that satisfies NAT requirements, a NAT session is started and a pair is allocated for this packet, and to all packets that are moving in that connection in the two directions.[123]

NAT can be classified based on how the pairs are created. If the pairs include only IP addresses and either the source or the destination address in the packet is changed, this is Basic NAT. If the pairs in the Basic NAT are fixed, it is called Static NAT, if the structure of the pair is of dynamic nature, it is Dynamic NAT. If NAT includes port numbers, as well as IP addresses, it is called Network Address Port translation (NAPT). Other types of NAT also exist including Bidirectional NAT which allows the NAT sessions to be started by packets moving in two directions between the domains, and Twice NAT Which allows the source and destination IP addresses to be changed at the same time.[124]

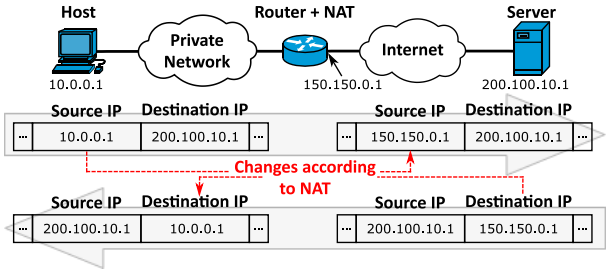


Figure 26 | Example of a NAT operation.

NAT was developed in 1994 and was first described in RFC 1631.[125] Then, a new standard was issued in 2001 including several new features such as Port address translation (PAT) that theoretically enables up to 65,000 local hosts, addressed with private addresses, to access the internet using a single public IP address and different port numbers.[126]

- Classless addressing approach as a part of a new routing method called Classless Inter-Domain Routing (CIDR).[127] As discussed above, classless addressing is a method to create address subspaces compatible with internet topology and following a hierarchical perspective.[128] On the other hand, CIDR is a routing method based on classless addressing. Because addresses are compatible with Internet topology, hosts that are located on the same site, share a part of their prefixes can be used to define a route destination. Based on this, routers can perform a mathematical operation to represent the previous set of routes, called now the aggregated routes, using an alternate single route called the aggregate route. Aggregated routes can be several, tens of thousands of routes, and can be more based on how the addressing was being done and on the position of the router that performs the aggregation in the global routing system. The previous operation is called route aggregation or route summarization, it resulted in the router keeping only the aggregate routes in its routing table and advertising them on the network (Figure 27) reducing the amount of routing information to be exchanged among routers and creating smaller and more efficient routing tables.[129]

  CIDR was first described in 1992 as an "Address Assignment and Aggregation Strategy",[130] then, in the next year, two requests for comments were issued to cover the addressing structure[131] and the routing mechanism[127] separately. Finally, in 2006, a Best Current Practice (BCP) was issued to summarize the technology and report the CIDR's effect on the global routing state.[132]

- A new internetwork protocol in accordance with the long-term strategy, it is Internet Protocol version 6 (IPv6).[24][133] The new protocol provides 128-bits addresses and defines a

space that includes 2^{128} ($3.4x10^{38}$) addresses.[134] The IPv6 design included default support for many technologies that were added to IPv4 such as route aggregation, link-local addressing, as well as supporting new technologies such as StateLess Address Auto-Configuration (SLAAC)[135] and a new type of addressing called anycast allowing a single IP address to be shared by devices, usually servers, located in multiple locations. When a data packet is destined to an anycast address, it will be routed to the closest device.[136] block of the IPv4 address space.[121]The transition toward IPv6 begun in the last years of the 20th century, but it went slower than expected, as short-term solutions proved to be midterm.[128] Thus, there is a need to support the two protocols at the same time, prompting the developers to design technologies such as the dual-stack.[137] IPv6 was documented in 1995 under RFC 1883.[24] Two years later, several minor modifications were included and a new request for comments was issued (RFC 2460).[25] This RFC was the official documentation for IPv6 for almost twenty years, until 2017, when RFC 8200 was published to adopts amendments for this protocol, based on features that were separately added in the previous two decades.[26]

IP Address Spaces Overlap

Internet subspaces overlapped when there is a set of addresses shared between two or more spaces, this occurs because of improper use of VLSM or route aggregation. As a result, two hosts might end with the same address, but each had a different mask, which will be leading to a routing problem.[138]

The solution lies with precise designing of the network and careful configuration of the routers, thus, subspaces will be completely separated.[139] Figure 28 shows an example where VLSM is used to create an addressing a 4-level hierarchy. In this example, the address subspaces 200.100.10.0/26 (green) in the third level overlaps with 200.100.10.0/27 (red) in the fourth level. Thus, the two subspaces should never be used to number hosts at the same time.

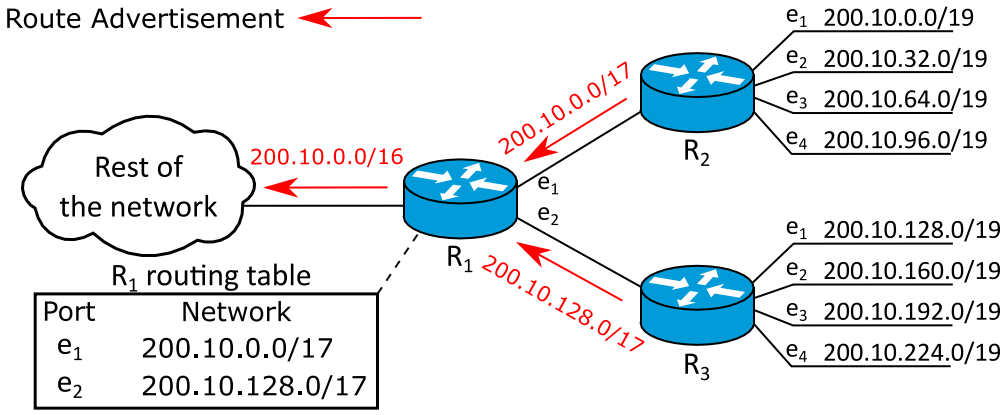


Figure 27 | An example of route aggregation as a part of CIDR.

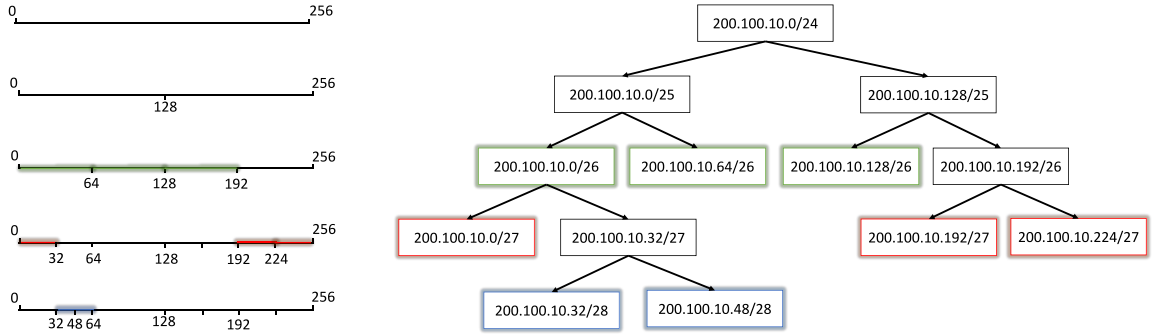


Figure 28 | Example for an IP address space overlapping.

Host mobility

When IPv4 was developed, host mobility was not considered. Thus, when wireless communication become popular in the late 1990s, a problem related to addressing was present: An IPv4 host shares a prefix with nearby local hosts and gateway routers, and the routing mechanism is based on this shared prefix. when moving to a new position where another network is used, physically and logically, the host needs to obtain a new IP address from the new network and share a new prefix with the novel nearby neighbors. When this is done, the routes to reach the host are completely changed, new routes need to be established and all previous connections will be failed.[140]

Several solutions were proposed to overcome this problem[141], including Mobile Internet Protocol (MIP) which was first proposed in 1996 in RFC 2002.[142] When MIP is implemented, the mobile host is given a long-term IP address in a network called the home network, additionally, it activates the home agent, a router on the mobile host used to route data packets to the home address when the host is away from his home networks. In the networks that support MIP, a router called the foreign agent is to be activated. When the mobile node visits an away network, it needs to register its home agent with the foreign agent of that network to obtain routing service and be integrated with the routing system of that away network.[143]

Related to Fragmentation

Supporting fragmentation is a basic service to be provided by all IPv4 implementation.[144] Some attackers took advantage of this mandatory designing and launching several fragmentation-based attacks on data networks. In the following, we discuss a set of these attacks and how to handle them:

- Tiny Fragment Attack: it is based on the fact that the smallest data packet supported by any IPv4 module has 68 bytes: a header with the maximum allowed length (60 bytes) and 8 bytes of payload. This length is not sufficient to include a transport protocol header in the packet, thus, it passes through firewalls without being verified because the firewall usually checks the port numbers at the transport protocol header. This problem is defined in RFC 1858.[145] The solution was introduced in RFC 3128 recommending to reject all data packets that have offset field value set to 0 or 1.[146]
- Overlap segments attack: It depends on a vulnerability in the reassembly algorithm: any subsequent segment can, partially or completely, overlap with another previous segment. This means that the first fragment, which contains the headers, can pass through the firewall with correct values, then, its content can be manipulated using another subsequent fragment. This attack can be countered by updating the reassembly algorithm to prevent overwriting data received previously.[147]

- Address Resolution Protocol (ARP) flooding: ARP provides the link-layer address associated with a known IP address for a remote host. When a data packet is fragmented, instead of sending an ARP request once per the original packet, it is to be sent for every fragment creating an ARP flooding which can be used as Denial-of-service (DOS) attack.[148] To prevent this, it is recommended to set a limit for the maximum number of ARP request to be sent per destination, for example, an ARP request every second per destination.[149]

Auxiliary Protocols

Address Resolution Protocols Family

Matching addresses used in the network layer with those used in the data link layer, and vice versa, is a function needed in the network protocol stack. It is provided by a family of network protocols called Address resolution protocols.[150] Examples of members of this family, used with IPv4, are the following:

- Address Resolution Protocol (ARP): it is used to discover the link address associated with a known IP address of a remote host, and this matching is essential to create the IP packet. The protocol was developed in 1982 and described in RFC 826.[151]
- Inverse Address Resolution Protocol (InARP): it does the opposite of what ARP does, i.e., it is used to discover an IP address associated with a known link address of a remote host. The protocol was created in 1992 and it is specified in RFC 1293.[152]
- Reverse Address Resolution Protocol (RARP): it is used to discover an IP address associated with a known link address in the same host that runs the protool. The protocol was developed in 1984 and described in RFC 903.[153]

Internet Control Message Protocol

Internet Control Message Protocol (ICMP) is a network-layer protocol and an integrated part of IPv4. The protocol is used to provide a packet's source and destination with a mechanism to communicate to exchange different pieces of information about the network status or packet processing. ICMP was developed in 1981 and it is described in RFC 792.[154]

The original standard defines 11 messages: redirect, destination unreachable, quench, time exceeded, parameter problem, echo and its reply which were used later to create the ping tool, timestamp and its reply, and Information request message and its reply.[155] Later, to respond to the needs of new protocols, several RFCs defined additional messages. For example, when the subnetting process was standardized, a couple of messages was created to address the new technology's needs: Address mask request and its reply.[156] On the other hand, several messages were obsoleted, due to the emergence of technologies that efficiently provide their services. For example, information request message, address mask request and their replies were obsoleted because of the development of DHCP.[157]

An IPv6-compatible version of the protocol was created in 1995, it is called Internet Control Protocol for the Internet Protocol version 6 (ICMPv6), it was primarily described in RFC 1885.[158]

Internet Group Management Protocol

Internet Group Management Protocol (IGMP) is a network-layer protocol that manages IPv4 multicast groups. It defines how hosts automatically join and leaves groups. In addition, the protocol does not add restrictions on the number of group members nor on their locations, and it also allows a host to join more than one group at the same time.[159]

The IETF have developed three versions of IGMP:

1. IGMPv1, developed in 1989 and described in RFC 1112, it covers the basic function of multicast group management such as membership[160]

2. IGMPv2, developed in 1997 and described in RFC 2236. It includes several features and enhancement to IGMPv1, such as allowing a host to request leaving a specific groupe.[161]
3. IGMPv3, which is the current version of the protocol. It was developed in 2002 and described in RFC 3376.[162] IGMPv3 is compatible back with old versions of the protocol, however, it supports many additional features such as Source-specific multicast which allows group's members to decide the whether to accept multicast traffic or not based on the source of the traffic.[163]

Internet Protocol Security

Internet Protocol Security (IPSec) is a protocol suite used to provide privacy services and authentication for other protocols active in the network layer. IPSec is used to secure connections between gateways (gateway-to-gateway), between host and gateway (host-to-gateway) or between hosts (host-to-host). It is used not only by IPv4 but also by IPv6 and other internetworking protocols.[164]

IPSec is an open standard that covers three types of protocols:[165] Authentication Headers (AH), which is used to ensure data integrity,[166] Encapsulating Security Payload (ESP), used to provide data confidentiality,[167] and Security Association (SA) which defines the rules of secure communication in the Internet environment such as how to use cryptographic keys.[168]

IP Security Working Group was established as a part of the Internet Engineering Task Force in 1993 to unify the efforts made by multiple research institutions, mainly the United stated Naval Research Laboratory (USNRL), and to set a standard for security services provided in the network layer.[169] This group had published three Requests for Comments in 1995,[170][171][172] and that was but an introduction for dozens of RFCs and standards in the following years[173] until 2005 when it was obsoleted.[169]

Application

In local area networks

IPv4 is the dominant internetworking protocol in Local area Networks (LANs). Each LAN will use IPv4 address space, commonly referred to as a network, and every node connected to the physical network needs to be configured with a valid and unique local IP address from that address space as well as a network mask that helps the node recognize the network and the host part of the given address. Additionally, it is essential for every node to be configured with the gateway IP address. It is the router address on that local network.[174] If the node needs to connect with nodes on other networks it needs to send packets to the gateway address.

There are two key protocols used in LANs along with IPv4: Dynamic host configuration protocol (DHCP)[40] and Address resolution protocol (ARP).[151] DHCP is used for dynamic host configuration, a client/server mechanism that allows a network node to dynamically obtain an IPv4 address, a network mask, gateway addresses, and other information without any manual configuration except for enabling the mechanism.[175] There are also autoconfiguration mechanisms for private IP addresses, usually refers as Automatic Private IP Addressing (APIPA)[176], these mechanisms were originally developed by Microsoft and Apple, but an open standard was released by IETF in 2005, it is described in RFC 3927.[177]

Regarding ARP, it allows the nodes to discover the link-layer addresses for other nodes in the same broadcast domain. If a node needs to send a packet to an adjacent node, it knows its IPv4 address and not the link address which is needed to fill the link layer header. This is important to L2 devices, such as switches that rely on the link-layer addresses, so they can send data frames correctly to their destination. ARP uses broadcast messages asking any nodes in the broadcast domain to reply if it has the corresponding IPv4 address. When the node reply, ARP can learn the physical address of the node.[178]

Other network services, such as Domain Name System (DNS), are based on IP or heavily rely on it. DNS is a client/server mechanism that provides name-to-address conversations and vice versa. IPv4 addresses consist of

a sequence of numbers that can easily be mistyped or forgotten.[179] When DNS is used, each IP address is associated with a name that is easy to remember by humans. When a user needs to send data or connect to an IP address, it can use the name instead of the address.[180]

In Internet

Internet is abbreviated from Internetwork meaning Interconnected Networks,[181] and all these networks are to be addressed using Internet protocol. Based on google statistics, until 2010, more than 99% of hosts on the Internet were using IPv4 natively. The percentage decreased significantly in the last decade as the IPv4 address space was exhausted (Figure 29), meanwhile the Internet continues growing thanks to IPv6. In 2022, almost 40% of hosts connected to the Internet use IPv6 natively.[182] For IPv4 hosts to connect to the internet, they need to use the public addresses assigned by IANA, RIR, or local ISPs.[88] Using NAPT, which is a form of NAT, is another common possibility in Small Office/Home Office (SOHO) networks.

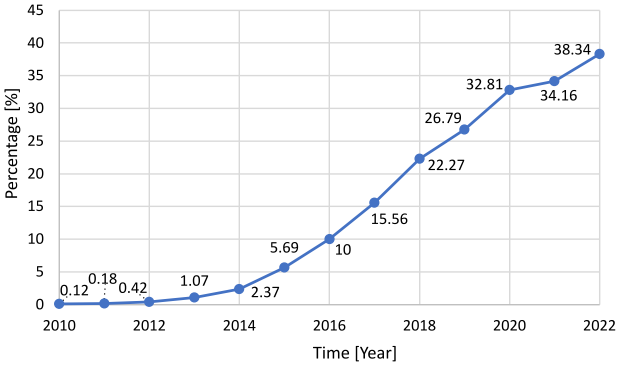


Figure 29 | IPv6 adoption on the Internet (2011-2022).

In this scenario (Figure 30), the site is assigned a public IPv4 address (200.100.10.1) and all hosts in the local network (the inside realm) are addressed using private addresses (10.0.0.0/8). NAPT takes place at the router that connects the network to the internet (outside realm) where each connection is assigned a unique port number, thus, a socket (IPv4, port number) can be now used to distinguish each connection.

Although the global routing system is entirely based on IP addressing following strictly the CIDR mechanism, IPv4 does not play any role in the routing process which is left to the routing protocols.[41] In 2022, more than 900.000 routes were advertised on the internet,[183] and more than 330.000 routes were addressed using IPv4.[184]

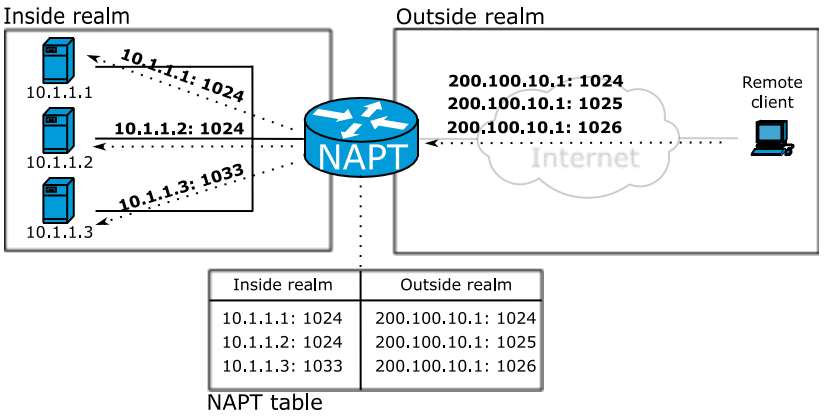


Figure 30 | PNAT in a SOHO.

The web is an information system reachable via the Internet, the access process is completely based on IP addressing. Users do not deal with IP addresses, but names instead, because the DNS service, as discussed in detail above, is supported by Web browsers. Browsing, using names instead of IP addresses, is an example of DNS use. In this case, the browser is a client that solicits a DNS server to get the IP address of the typed site.[185]

Notes

a. For TCP, refer to RFC 793 [12]
b. The counting of the versions started from zero, i.e., the first version has the version number set to 0.
c. Number 3 was not used as indexes for IP versions in IENs.
d. Usually referred to as a *network*. However, in this paper, we use the word *subspace* to avoid ambiguity.
e. In the original protocol standard reserved bits was part of network identifiers,[70] but they were always excluded from the mathematical calculations related to the identifiers and treated as a separate part of the IPv4 address.[83]
f. A distinction must be made between the number of addresses in a subspace, calculated using $2^{b_{HID}}$, and the number of addresses available for numbering hosts, which is calculated using $2^{b_{HID}} - 2$, where the two subtracted addresses are the network address and the broadcast addresses, which are served in every subspace and must not be used to numbering hosts.
g. Refer to RFC 1878 for all possible masks when the classless addressing method is used.[83]

Additional information

Competing interests

The author declares no relevant conflicts of interest.

References

1. Postel, J.; Sunshine, C.; Cohen, D. (July 1981). "The ARPA Internet Protocol.". *Computer Networks* **5** (4): 261-271. doi:10.1016/0376-5075(81)90003-9. ISSN 0376-5075.
2. RFC 791, p. 1
3. *R. M. Metcalfe (1973). Packet communication (Report). Massachusetts Institute of Technology. pp. 2–3.*
4. *Baran, P. (1960). Reliable Digital Communications Systems Using Unreliable Network Repeater Nodes (Report). RAND Corporation. p. 2. P-1995.*
5. Baran, P. (1964). "On Distributed Communications Networks". *IEEE Transactions on Communications Systems* **12** (1): 1-9. doi:10.1109/TCOM.1964.1088883.
6. Roberts, L. G. (1978). "The evolution of packet switching". *Proceedings of the IEEE* (IEEE) **66** (11): 1307-1313. doi:10.1109/PROC.1978.11141.
7. *Baran P. (August 1964). On Distributed Communications: I. Introduction to Distributed Communications Networks (PDF). rand.org (Report). The Rand corporation. RM-3420-PR.*
8. Pouzin, L. (January 1973). "Presentation and Major Design Aspects Of The CYCLADES Computer Network". *DATACOMM '73 Proceedings of the third ACM symposium on Data communications and Data networks: Analysis and design* (ACM): 80-87. doi:10.1145/800280.811034. ISBN 9781450373845.
9. Stevens, W. (2011). *TCP/IP Illustrated*. Boston, MA: Addison-Wesley. p. 5. ISBN 9780321336316.
10. Cerf, V.; Khan, R. (May 1974). "A Protocol for Packet Network Intercommunication". *IEEE Transactions on Communications* (IEEE) **22** (5): 637-648. doi:10.1109/TCOM.1974.1092259. ISSN 1558-0857.
11. *Cerf, V.; Dalal, Y.; Sunshine, C. (December 1974). "RFC 675, Specification of Internet Transmission Control Program". The Internet Society. doi:10.17487/RFC0675. Archived from the original on 31 December 2019. Retrieved 16 December 2020.*
12. *Postal, J. (September 1981). "RFC 793, Transmission Control Protocol, DARPA Internet Program, Protocol Specification". The Internet Society. doi:10.17487/RFC0793. Archived from the original on 18 September 2019. Retrieved 16 December 2020.*
13. *Postel, J. (August 1977). "IEN 2, 2.3.3.2 Comments on Internet Protocol and TCP". The Internet Society. Archived from the original (TXT) on 8 January 2019. Retrieved 18 December 2020.*
14. *Cerf, V. (February 1978). "IEN 26, 2.3.2.1 A Proposed New Internet Header Format" (PDF). The Internet Society. Archived from the original (PDF) on 16 May 2019. Retrieved 16 December 2020.*
15. *Postel, J. (February 1978). "IEN 28, Draft Internetwork Protocol Description Version 2" (PDF). The Internet Society. Archived from the original (PDF) on 16 May 2019. Retrieved 16 December 2020.*
16. *Postel, J. (June 1978). "IEN 41, Internetwork Protocol Specification Version 4" (PDF). The Internet Society. Archived from the original (PDF) on 16 May 2019. Retrieved 16 December 2020.*
17. *Postel, J. (June 1978). "IEN 44, Latest Header Format" (PDF). The Internet Society. Archived from the original (PDF) on 16 May 2019. Retrieved 16 December 2020.*
18. *Postel, J. (September 1978). "IEN 54, Internetwork Protocol Specification Version 4" (PDF). The Internet Society. Archived from the original (PDF) on 16 May 2019. Retrieved 16 December 2020.*
19. *Postel, J. (January 1980). "RFC 760, DOD Standard Internet Protocol". The Internet Society. doi:10.17487/RFC0760. Archived from the original on 16 October 2019. Retrieved 19 December 2020.*
20. *Postel, J. (November 1981). "RFC 801, NCP/TCP Transition Plan". The Internet Society. p. 2. doi:10.17487/RFC0801. Archived from the original on 11 December 2019. Retrieved 19 December 2020.*
21. *Nesser, P.; Bergstrom, A. (June 2004). "RFC 3789, Introduction to the Survey of IPv4 Addresses in Currently Deployed IETF Standards Track and Experimental Documents". The Internet Society. p. 2. doi:10.17487/RFC3789. Archived from the original on 11 December 2019. Retrieved 19 December 2020.*
22. *Delgrossi, L.; Berger, L. (August 1995). "RFC 1819, Internet Stream Protocol Version 2 (ST2), Protocol Specification - Version ST2+". The Internet*

Society. doi:10.17487/RFC1819. Archived from the original on 2019-12-19. Retrieved 19 December 2020.
23. Ullmann, R. (June 1993). "RFC 1475, TP/IX: The Next Internet". The Internet Society. p. 7. doi:10.17487/RFC1475. Archived from the original on 2020-01-09. Retrieved 19 December 2020.
24. Deering, S.; Hinden, R. (December 1995). "RFC 1883, Internet Protocol, Version 6 (IPv6) Specification". The Internet Society. doi:10.17487/RFC1883. Archived from the original on 2019-12-21. Retrieved 19 December 2020.
25. Deering, S.; Hinden, R. (December 1998). "RFC 2460, Internet Protocol, Version 6 (IPv6) Specification". The Internet Society. doi:10.17487/RFC2460. Archived from the original on 2020-01-07. Retrieved 19 December 2020.
26. Deering, S.; Hinden, R. (July 2017). "RFC 8200, Internet Protocol, Version 6 (IPv6) Specification". The Internet Society. doi:10.17487/RFC8200. Archived from the original on 2019-12-11. Retrieved 19 December 2020.
27. Schoen, S.D.; Gilmore, J.; Täht, D. (22 August 2022). "Internet-Draft, The IETF Will Continue Maintaining IPv4". The Internet Society. p. 9. Archived from the original on 2022-11-12. Retrieved 12 November 2020.
28. Onions, J. (April 1994). "RFC 1606, A Historical Perspective on The Usage Of IP Version 9". The Internet Society. doi:10.17487/RFC1606. Archived from the original on 2019-08-11. Retrieved 19 December 2020.
29. Postel, J. (September 1981). "RFC 791, Internet Protocol". The Internet Society. p. 5-6. doi:10.17487/RFC0791. Archived from the original on 1 January 2020. Retrieved 19 December 2020.
30. Socolofsky, T.; Kale, C. (January 1991). "RFC 1180, TCP/IP Tutorial". The Internet Society. p. 3. doi:10.17487/RFC1180. Archived from the original on 17 October 2020. Retrieved 19 December 2020.
31. Cerf (1974), p. 638
32. RFC 791, p. 6
33. Cerf, V. (July 1978). "IEN 48, The Catenet Model For Internetworking". The Internet Society. Archived from the original (TXT) on 13 November 2020. Retrieved 16 December 2020.
34. Callon, R. (December 1983). "Internetwork protocol". Proceedings of the IEEE (IEEE) **71** (12): 1388-1393. doi:10.1109/PROC.1983.12783. ISSN 1558-2256.
35. Lee, D. (1999). *Enhanced IP Services for Cisco Networks* (first ed.). Cisco Press. p. 26. ISBN 1578701066.
36. RFC 791, p. 8
37. RFC 1180, p. 3-4
38. RFC 791, p. 18-19
39. Blank, A. (2004). *TCP/IP Foundations*. Sybex. p. 86. ISBN 0782143709.
40. Droms, R. (March 1997). "RFC 2131, Dynamic Host Configuration Protocol". The Internet Society. doi:10.17487/RFC2131. Archived from the original on 15 November 2018. Retrieved 30 December 2020.
41. RFC 791, p. 9
42. Stevens (2011), p. 181
43. Kozierok, C. (2005). *The TCP/IP Guide: a Comprehensive, Illustrated Internet Protocols Reference*. San Francisco: No Starch Press. p. 545. ISBN 1-59327-047-X.
44. RFC 791, p. 13
45. RFC 791, p. 11
46. RFC 791, p. 11-14
47. Almquist, P. (June 1992). "RFC 1349, Type of Service in the Internet Protocol Suite". The Internet Society. doi:10.17487/RFC1349. Archived from the original on 22 October 2019. Retrieved 30 December 2020.
48. Nichols, K.; Blake, S.; Baker, F.; Black, D. (December 1998). "RFC 2474, Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers". The Internet Society. doi:10.17487/RFC2474. Archived from the original on 6 January 2020. Retrieved 30 December 2020.
49. Touch, J. (February 2013). "RFC 6864, Updated Specification of the IPv4 ID Field". The Internet Society. doi:10.17487/RFC6864. Archived from the original on 21 October 2020. Retrieved 30 December 2020.
50. Janevski, T. (2015). *Internet Technologies for Fixed and Mobile Networks* (1 ed.). Artech House. pp. 33. ISBN 9781608079216.
51. "Service Name and Transport Protocol Port Number Registry". IANA. Archived from the original (XHTML) on 22 September 2019. Retrieved 20 January 2021.
52. RFC 791, p. 14
53. RFC 791, p. 15
54. Stevens (2011), p. 192
55. "Internet Protocol Version 4 (IPv4) Parameters - IP Option Numbers". IANA. Archived from the original (XHTML) on 24 November 2019. Retrieved 26 June 2019.
56. RFC 791, p. 16-17
57. Stevens (2011), p. 192-194
58. RFC 791, p. 12-13
59. RFC 2474, p. 1-2
60. Blake, S.; Black, D.; Carlson, M.; Davies, E.; Wang, Z.; Weiss, W. (December 1998). "RFC 2475, An Architecture for Differentiated Services". The Internet Society. p. 1. doi:10.17487/RFC2475. Archived from the original on 6 September 2020. Retrieved 30 December 2020.
61. RFC 2474, p. 7
62. RFC 2475, p. 12
63. RFC 2474, p. 6
64. Babiarz, J.; Chan, K.; Baker, F. (August 2006). "RFC 4594, Configuration Guidelines for DiffServ Service Classes". The Internet Society. p. 9-11. doi:10.17487/RFC4594. Archived from the original on 30 August 2020. Retrieved 30 December 2020.
65. Davie, B.; Charny, A.; Bennett, J. C. R.; Benson, K.; Le Boudec, J.Y.; Courtney, W.; Davari, S.; Firoiu, V.; Stiliadis, D. (March 2002). "RFC 3246, An Expedited Forwarding PHB (Per-Hop Behavior)". The Internet Society. doi:10.17487/RFC3246. Archived from the original on 23 September 2020. Retrieved 30 December 2020.
66. Heinanen, J.; Baker, F.; Weiss, W.; Wroclawski, J. (March 2002). "RFC 2597, Assured Forwarding PHB Group". The Internet Society. p. 6. doi:10.17487/RFC2597. Archived from the original on 20 January 2020. Retrieved 30 December 2020.
67. Ramakrishnan, K.; Floyd, S. (January 1999). "RFC 2481, A Proposal to add Explicit Congestion Notification (ECN) to IP". The Internet Society. doi:10.17487/RFC2481. Archived from the original on 5 August 2020. Retrieved 30 December 2020.
68. Ramakrishnan, K.; Floyd, S.; Black, D. (September 2001). "RFC 3168, The Addition of Explicit Congestion Notification (ECN) to IP". The Internet Society. p. 7. doi:10.17487/RFC3168. Archived from the original on 17 May 2020. Retrieved 30 December 2020.
69. Dyson, P. (1999). *Dictionary of networking*. San Francisco: Sybex. p. 199. ISBN 0782124615.
70. RFC 791, p. 7
71. Pouzin (1973), p. 87
72. Mogul, J.; Postel, J. (August 1985). "RFC 950, Internet Standard Subnetting Procedure". The Internet Society. p. 1. doi:10.17487/RFC0950. Archived from the original on 30 October 2020. Retrieved 22 November 2020.
73. Cerf (1974), p. 642
74. Clark, C. (June 1978). "IEN 46, A Proposal for Addressing and Routing in theInternet". The Internet Society. pp. 1–2. Archived from the original (txt) on 25 May 2022. Retrieved 10 November 2022.
75. RFC 760, p. 22
76. RFC 796, p. 1
77. Blank (2004), p. 76
78. Main, A. (February 2005). "Textual Representation of IPv4 and IPv6 Addresses". The Internet Society. p. 3-4. Archived from the original on 29 September 2019. Retrieved 30 January 2021.
79. Jacobsen, O.; Lynch, D. (March 1991). "RFC 1208, A Glossary of Networking Terms". The Internet Society. p. 5. Archived from the original on 8 November 2019. Retrieved 30 January 2021.
80. Deering, S. (August 1989). "RFC 1112, Host Extensions for IP Multicasting". The Internet Society. p. 3. doi:10.17487/RFC1112. Archived from the original on 2020-02-03. Retrieved 10 January 2021.
81. Deering, S. (July 1986). "RFC 988, Host Extensions for IP Multicasting". The Internet Society. p. 3. doi:10.17487/RFC0988. Archived from the original on 2019-12-11. Retrieved 4 September 2020.
82. Reynolds, J.; Postel, J. (November 1986). "RFC 990, Assigned Numbers". The Internet Society. p. 4. doi:10.17487/RFC0990. Archived from the original on 11 December 2019. Retrieved 30 January 2021.
83. Pummill, T.; Manning, B. (December 1995). "RFC 1878, Variable Length Subnet Table For IPv4". The Internet Society. p. 2. doi:10.17487/RFC1878.

Archived from the original on 29 September 2019. Retrieved 30 January 2021.
84. Fuller, D.; Li, T.; Yu, J.; Varadhan, K. (June 1992). "RFC 1338, Supernetting: an Address Assignment and Aggregation Strategy". The Internet Society. p. 2. doi:10.17487/RFC1338. Archived from the original on 28 July 2019. Retrieved 30 January 2021.
85. Rekhter, Y.; Li, T. (September 1993). "RFC 1518, An Architecture for IP Address Allocation with CIDR". The Internet Society. p. 1-5. doi:10.17487/RFC1518. Archived from the original on 28 November 2020. Retrieved 30 January 2021.
86. Fuller, V.; Li, T.; Yu, J.; Varadhan, K. (September 1993). "RFC 1519, Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy". The Internet Society. p. 2-10. doi:10.17487/RFC1519. Archived from the original on 12 August 2020. Retrieved 30 January 2021.
87. Kozierok (2005), p. 359
88. Housley, R.; Curran, J.; Huston, G.; Conrad, D. (August 2013). "RFC 7020, the Internet Numbers Registry System". The Internet Society. p. 3. doi:10.17487/RFC7020. Archived from the original on 30 September 2019. Retrieved 30 January 2021.
89. Dyson (1999), p. 365
90. RFC 1878, p. 3-6
91. Fuller, V.; Li, T. (August 2006). "RFC 4632, Classless Inter-Domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan". The Internet Society. p. 5. doi:10.17487/RFC4632. Archived from the original on 23 August 2020. Retrieved 4 September 2020.
92. Cerf, V. (August 1990). "RFC 1174, IAB Recommended Policy on Distributing Internet Identifier Assignment and IAB Recommended Policy Change to Internet "Connected" Status". The Internet Society. p. 2. doi:10.17487/RFC1174. Archived from the original on 8 November 2019. Retrieved 22 January 2020.
93. Carpenter, B.; Baker, F.; Roberts, M. (June 2000). "RFC 2860, Memorandum of Understanding Concerning the Technical Work of the Internet Assigned Numbers Authority". The Internet Society. p. 2. doi:10.17487/RFC2860. Archived from the original on 11 August 2012. Retrieved 2 January 2019.
94. RFC 7020, p. 3-5
95. "Internet Assigned Numbers Authority (IANA) Policy For Allocation of IPv4 Blocks to Regional Internet Registries". ICANN.org. Archived from the original on 6 November 2020. Retrieved 1 January 2020.
96. Hubbard, K.; Kosters, M.; Conrad, D.; Karrenberg, D.; Postel, J. (November 1996). "RFC 2050, Internet Registry IP Allocation Guidelines". The Internet Society. p. 4. doi:10.17487/RFC2050. Archived from the original on 11 December 2019. Retrieved 22 January 2020.
97. Cotton, M.; Vegoda, L.; Haberman, B. (April 2013). "RFC 6890, Special-Purpose IP Address Registries". The Internet Society. p. 1. doi:10.17487/RFC6890. Archived from the original on 30 August 2020. Retrieved 22 January 2020.
98. RFC 6890, p. 7
99. "IANA IPv4 Special-Purpose Address Registry". IANA. Archived from the original (XHTML) on 8 June 2021. Retrieved 10 June 2021.
100. "IPv4 Multicast Address Space Registry". IANA. Archived from the original (XHTML) on 6 February 2019. Retrieved 14 September 2020.
101. Cotton, M.; Vegoda, L.; Meyer, D. (March 2010). "RFC 5771, IANA Guidelines for IPv4 Multicast Address Assignments". The Internet Society. p. 2-3. doi:10.17487/RFC5771. Archived from the original on 23 October 2020. Retrieved 22 November 2020.
102. RFC 950, p. 5
103. Kozierok (2005), p. 294-296
104. Stevens (2011), p. 41-42
105. Mogul, J. (October 1984). "RFC 919, Broadcasting Internet Datagrams". The Internet Society. p. 5-6. doi:10.17487/RFC0919. Archived from the original on 20 October 2020. Retrieved 22 November 2020.
106. RFC 1208, p. 8
107. Huston, G. (January 2016). "Evaluating IPv4 and IPv6 Packet Fragmentation". RIPE NCC. Archived from the original on 25 April 2018.
108. RFC 791, p. 7-8
109. RFC 1208, p. 7
110. Stevens (2011), p. 488
111. RFC 791, p. 26
112. Kozierok (2005), p. 347-348
113. Stevens (2011), p.388
114. RFC 791, p. 28
115. Callon (1983), p. 1392
116. RFC 791, p. 16
117. RFC 791, p. 19-20
118. RFC 791, p. 18-20
119. RFC 4632, p. 18
120. Egevang, K.; Francis, P. (May 1994). "RFC 1631, The IP Network Address Translator (NAT)". The Internet Society. p. 2. doi:10.17487/RFC1631. Archived from the original on 6 August 2020. Retrieved 10 November 2020.
121. "Available Pool of Unallocated IPv4 Internet Addresses Now Completely Emptied" (PDF). ICANN. 3 February 2011. Archived from the original (PDF) on 20 September 2019. Retrieved 20 September 2019.
122. Dyson (1999), P.264-265
123. RFC 3022, p. 7-8
124. RFC 2663, p. 9-14
125. RFC 1631, p. 1
126. Srisuresh, P.; Egevang, K. (January 2001). "RFC 3022, Traditional IP Network Address Translator (Traditional NAT)". doi:10.17487/RFC3022. Archived from the original on 25 December 2019. Retrieved 22 September 2019.
127. RFC 1519, p. 1
128. RFC 4632, p. 4
129. RFC 4632, p. 8-9
130. RFC 1338, p. 1
131. RFC 1518, p.1
132. RFC 4632, p. 19
133. Kozierok (2005), p. 366
134. Kozierok (2005), p. 376
135. Thomson, S.; Narten, T.; Jinmei, T. (September 2007). "RFC 4862, IPv6 Stateless Address Autoconfiguration". doi:10.17487/RFC4682. Archived from the original on 19 December 2019. Retrieved 24 September 2019.
136. Hinden, R.; Deering, S. (February 2006). "RFC 4291, IP Version 6 Addressing Architecture". p. 12. doi:10.17487/RFC4291. Archived from the original on 16 February 2020. Retrieved 24 September 2019.
137. Nordmark, E.; Gilligan, R. (October 2005). "RFC 4213, Basic Transition Mechanisms for IPv6 Hosts and Routers". p. 4. doi:10.17487/RFC4213. Archived from the original on 26 December 2019. Retrieved 24 September 2019.
138. Wegner, J.; Rockell, R. (2000). IP Addressing and Subnetting INC IPV6: Including IPv6. Syngress. p. 219. ISBN 1928994016.
139. Kozierok (2005), p. 405
140. Stevens (2011), p. 215-216
141. Zhu, Z.; Wakikawa, R.; Zhang, L. (July 2011). "RFC 6301, A Survey of Mobility Support in the Internet". The Internet Society. p. 7. doi:10.17487/RFC6301. Archived from the original on 2022-04-23. Retrieved 31 October 2022.
142. Perkins, C. (October 1996). "RFC 2002, IP Mobility Support". The Internet Society. p. 1. doi:10.17487/RFC2002. Archived from the original on 2022-04-01. Retrieved 31 October 2022.
143. Perkins, C. (November 2010). "RFC 5944, IP Mobility Support". The Internet Society. p. 7. doi:10.17487/RFC5944. Archived from the original on 2022-03-18. Retrieved 1 November 2022.
144. RFC 791, p. 23
145. Ziemba, G.; Reed, D.; Traina, P. (October 1995). "RFC 1858, Security Considerations for IP Fragment Filtering". The Internet Society. p. 3. doi:10.17487/RFC1858. Archived from the original on 10 August 2020. Retrieved 15 November 2020.
146. Miller, I. (January 2001). "RFC 3128, Protection Against a Variant of the Tiny Fragment Attack". RFC 3128, The Internet Society. p. 3. doi:10.17487/RFC3128. Archived from the original on 11 August 2012. Retrieved 22 July 2018.
147. RFC 1858, p. 5-8
148. Stevens (2011), p. 497
149. Braden, R. (October 1989). "RFC 1122, Requirements for Internet Hosts -- Communication Layers". The Internet Society. p. 22-23. doi:10.17487/RFC1122. Archived from the original on 30 September 2020. Retrieved 22 July 2018.
150. Kozierok (2005), p. 204-206

151. Plummer, D. (November 1982). "RFC 826, An Ethernet Address Resolution Protocol or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware". The internet Society. p. 1. doi:10.17487/RFC0826. Archived from the original on 19 February 2020.
152. Bradley, T.; Brown, C.; Malis, A. (January 1992). "RFC 1293, Inverse Address Resolution Protocol". The internet Society. p. 1. doi:10.17487/RFC1293. Archived from the original on 2 January 2020.
153. Finlayson, R.; Timothy, M.; Mogul, J.; Theimer, M. (June 1984). "RFC 903, a Reverse Address Resolution Protocol". The internet Society. p. 1. doi:10.17487/RFC0903. Archived from the original on 2 January 2020.
154. Postel, J. (December 1995). "RFC 792, Internet Control Message Protocol". The internet Society. p. 1. doi:10.17487/RFC0792. Archived from the original on 17 October 2020. Retrieved 15 September 2020.
155. RFC 792, p. 20
156. RFC 950, p. 10
157. Kozierok (2005), p. 614
158. Conta, A. (December 1995). "RFC 1885, Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification". The internet Society. p. 1. doi:10.17487/RFC1885. Archived from the original on 16 December 2019. Retrieved 15 September 2020.
159. Stevens (2011), p. 435-436
160. RFC 1112, p. 11
161. Fenner, W. (November 1997). "RFC 2236, Internet Group Management Protocol, Version 2". The Internet Society. doi:10.17487/RFC2236. Archived from the original on 21 October 2020. Retrieved 18 February 2017.
162. Cain, B.; Deering, S.; Kouvelas, I.; Fenner, B.; Thyagarajan, A. (October 2002). "RFC 3376, Internet Group Management Protocol, Version 3". The Internet Society. doi:10.17487/RFC3376. Archived from the original on 28 March 2019. Retrieved 1 June 2021.
163. H. Holbrook, B. Cain (August 2006). "RFC 4607, Source-Specific Multicast for IP". The Internet Society. doi:10.17487/RFC4607. Archived from the original on 11 August 2012. Retrieved 1 December 2020.
164. Frankel, S.; Krishnan, S. (February 2011). "RFC 6071, IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap". The Internet Society. p. 4. doi:10.17487/RFC6071. ISSN 2070-1721. Archived from the original on 25 December 2019. Retrieved 20 September 2020.
165. Thayer, R.; Doraswamy, N. (November 1998). "RFC 2411, IP Security Document Roadmap". The Internet Society. p. 1. doi:10.17487/RFC2411. Archived from the original on 11 December 2019. Retrieved 28 September 2020.
166. Kent, S.; Atkinson, R. (November 1998). "RFC 2402, IP Authentication Header". The Internet Society. p. 1. doi:10.17487/RFC2402. Archived from the original on 11 December 2019. Retrieved 28 September 2020.
167. Kent, S.; Atkinson, R. (November 1998). "RFC 2406, IP Encapsulating Security Payload (ESP)". The Internet Society. p. 1. doi:10.17487/RFC2406. Archived from the original on 18 December 2019. Retrieved 28 September 2019.
168. Maughan, D.; Schertler, M.; Schneider, M.; Turner, J. (November 1998). "RFC 2408, Internet Security Association and Key Management Protocol (ISAKMP)". The Internet Society. p. 1. doi:10.17487/RFC2408. Archived from the original on 11 December 2019. Retrieved 28 September 2020.
169. "IP Security Protocol (ipsec) - Group History". IETF. Archived from the original on 13 September 2019. Retrieved 28 September 2019.
170. Atkinson, R. (August 1995). "RFC 1825, Security Architecture for the Internet Protocol". The Internet Society. p. 1. doi:10.17487/RFC1825. Archived from the original on 18 December 2019. Retrieved 28 September 2019.
171. Atkinson, R. (August 1995). "RFC 1826, IP Authentication Header". The Internet Society. p. 1. doi:10.17487/RFC1826. Archived from the original on 19 December 2019. Retrieved 28 September 2020.
172. Atkinson, R. (August 1995). "RFC 1827, IP Encapsulating Security Payload (ESP)". The Internet Society. p. 1. doi:10.17487/RFC1827. Archived from the original on 11 December 2019. Retrieved 28 September 2019.
173. "IP Security Protocol (ipsec)". Archived from the original on 13 September 2019. Retrieved 28 September 2019.
174. Kozierok (2005), P. 329
175. RFC 2131, P. 2-3
176. Kozierok (2005), P. 1195-1197
177. Cheshire, S.; Aboba, B.; Guttman, E. (May 2005). "RFC 3927, Dynamic Configuration of IPv4 Link-Local Addresses". The Internet Society. p. 1, 3. doi:10.17487/RFC3927. Archived from the original on 2022-03-02. Retrieved 12 November 2022.
178. RFC 826, P. 4
179. Mockapetris, P. (November 1987). "RFC 1034, Dynamic Configuration of IPv4 Link-Local Addresses". The Internet Society. p. 2-3. doi:10.17487/RFC1034. Archived from the original on 2022-10-26. Retrieved 12 November 2022.
180. Stevens, (2011), P. 511
181. Dyson (1999), P. 194
182. "Google IPv6, Statistics". google.com. Archived from the original on 9 November 2022. Retrieved 12 November 2022.
183. "Growth of the BGP Table - 1994 to Present". potaroo.net. Archived from the original on 3 November 2022. Retrieved 12 November 2022.
184. "IPv4 stats, prefixes advertised pool". potaroo.net. Archived from the original on 22 April 2022. Retrieved 12 November 2022.
185. Stevens (2011), p. 19