IPv4

From Wikipedia, the free encyclopedia

Internet Protocol version 4 (**IPv4**) is the fourth version in the development of the Internet Protocol (IP) Internet, and routes most traffic on the Internet.[1] However, a successor protocol, IPv6, has been defined and is in various stages of production deployment. IPv4 is described in IETF publication RFC 791 (September 1981), replacing an earlier definition (RFC 760, January 1980).

IPv4 is a connectionless protocol for use on packet-switched networks. It operates on a best effort delivery model, in that it does not guarantee delivery, nor does it assure proper sequencing or avoidance of duplicate delivery. These aspects, including data integrity, are addressed by an upper layer transport protocol, such as the Transmission Control Protocol (TCP).

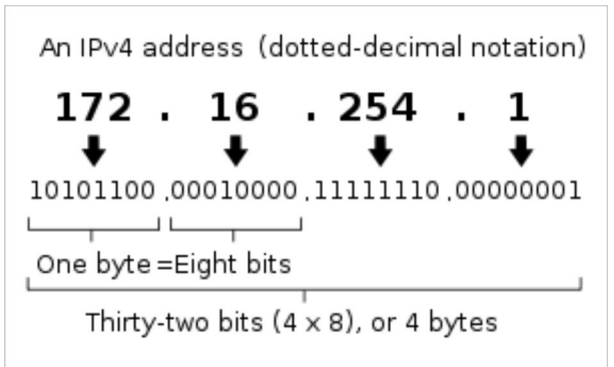
Contents

- 1 Addressing
  - 1.1 Address representations
  - 1.2 Allocation
  - 1.3 Special-use addresses
    - 1.3.1 Private networks
      - 1.3.1.1 Virtual private networks
  - 1.4 Link-local addressing
  - 1.5 Loopback
  - 1.6 Addresses ending in 0 or 255
  - 1.7 Address resolution
- 2 Address space exhaustion
- 3 Packet structure
  - 3.1 Header
  - 3.2 Data
- 4 Fragmentation and reassembly
  - 4.1 Fragmentation
  - 4.2 Reassembly
- 5 Assistive protocols
- 6 See also
- 7 Notes
- 8 References
- 9 External links
- 10 Further Reading

Addressing

IPv4 uses 32-bit (four-byte) addresses, which limits the address space to 4 294 967 296 (2^{32}) addresses. As addresses were assigned to users, the number of unassigned addresses decreased. IPv4 address exhaustion occurred on February 3, 2011, although it had been significantly delayed by address changes such as classful network design, Classless Inter-Domain Routing, and network address translation (NAT).

This limitation of IPv4 stimulated the development of IPv6 in the 1990s, which has been in commercial deployment since 2006.



Decomposition of the quad-dotted IPv4 address representation to its binary value

IPv4 reserves special address blocks for private networks (~18 million addresses) and multicast addresses (~270 million addresses).

Address representations

IPv4 addresses may be written in any notation expressing a 32-bit integer value, but for human convenience, they are most often written in the dot-decimal notation, which consists of four octets of the address expressed individually in decimal and separated by periods.

The following table shows several representation formats:

Notation	Value	Conversion from dot-decimal
Dotted decimal	192.0.2.235	N/A
Dotted hexadecimal[2]	0xC0.0x00.0x02.0xEB	Each octet, preceded by 0x, is individually converted to hexadecimal form.
Dotted octal[2]	0301.0250.0002.0353	Each octet, preceded by 0, is individually converted into octal.
Hexadecimal	0xC00002EB	The 32-bit number is expressed as the concatenation of the octets from the dotted hexadecimal.
Decimal	3221226219	The 32-bit number is expressed in decimal.
Octal	030000001353	The 32-bit number is expressed in octal.

Mixing decimal, octal and hexadecimal is allowed in dotted format per octet.

Note that in non-dotted formats, numbers bigger than 32-bit, can be given in some cases (e.g. Firefox) and will get converted mod 2^{32}.[3]

Allocation

Originally, an IP address was divided into two parts: the network identifier was the most significant (highest order) octet of the address, and the host identifier was the rest of the address. The latter was therefore also called the *rest field*. This enabled the creation of a maximum of 256 networks. This was quickly found to be inadequate.

To overcome this limit, the high order octet of the addresses was redefined to create a set of *classes* of networks, in a system which later became known as classful networking. The system defined five classes, Class A, B, C, D, and E. The Classes A, B, and C had different bit lengths for the new network identification. The rest of an address was used as previously to identify a host within a network, which meant that each network class had a different capacity to address hosts. Class D was allocated for multicast

addressing and Class E was reserved for future applications.

Starting around 1985, methods were devised to subdivide IP networks. One method that has proved flexible is the use of the *variable-length subnet mask* (VLSM).[4][5]

Based on the IETF standard RFC 1517 published in 1993, this system of classes was officially replaced with Classless Inter-Domain Routing (CIDR), and the class-based scheme was dubbed *classful*, by contrast. CIDR was designed to permit repartitioning of any address space so that smaller or larger blocks of addresses could be allocated to users. The hierarchical structure created by CIDR is managed by the Internet Assigned Numbers Authority (IANA) and the regional Internet registries (RIRs). Each RIR maintains a publicly searchable WHOIS database that provides information about IP address assignments.

Special-use addresses

Reserved address blocks

Range	Description	Reference
0.0.0.0/8	Current network (only valid as source address)	RFC 6890
10.0.0.0/8	Private network	RFC 1918
100.64.0.0/10	Shared Address Space	RFC 6598
127.0.0.0/8	Loopback	RFC 6890
169.254.0.0/16	Link-local	RFC 3927
172.16.0.0/12	Private network	RFC 1918
192.0.0.0/24	IETF Protocol Assignments	RFC 6890
192.0.2.0/24	TEST-NET-1, documentation and examples	RFC 5737
192.88.99.0/24	IPv6 to IPv4 relay	RFC 3068
192.168.0.0/16	Private network	RFC 1918
198.18.0.0/15	Network benchmark tests	RFC 2544
198.51.100.0/24	TEST-NET-2, documentation and examples	RFC 5737
203.0.113.0/24	TEST-NET-3, documentation and examples	RFC 5737
224.0.0.0/4	IP multicast (former Class D network)	RFC 5771
240.0.0.0/4	Reserved (former Class E network)	RFC 1700
255.255.255.255	Broadcast	RFC 919

Private networks

Of the approximately four billion addresses allowed in IPv4, three ranges of address are reserved for use in private networks. These ranges are not routable outside of private networks, and private machines cannot directly communicate with public networks. They can, however, do so through network address translation.

The following are the three ranges reserved for private networks (RFC 1918):

Name	Address range	Number of addresses	*Classful* description	Largest CIDR block
24-bit block	10.0.0.0–10.255.255.255	16 777 216	Single Class A	10.0.0.0/8
20-bit block	172.16.0.0–172.31.255.255	1 048 576	Contiguous range of 16 Class B blocks	172.16.0.0/12
16-bit block	192.168.0.0–192.168.255.255	65 536	Contiguous range of 256 Class C blocks	192.168.0.0/16

Virtual private networks

Packets with a private destination address are ignored by all public routers. Two private networks (e.g., two branch offices) cannot communicate via the public internet, unless they use an IP tunnel or a virtual private network (VPN). When one private network wants to send a packet to another private network, the first private network encapsulates the packet in a protocol layer so that the packet can travel through the public network. Then the packet travels through the public network. When the packet reaches the other private network, its protocol layer is removed, and the packet travels to its destination.

Optionally, encapsulated packets may be encrypted to secure the data while it travels over the public network.

Link-local addressing

RFC 6890 defines the special address block 169.254.0.0/16 for link-local addressing. These addresses are only valid on links (such as a local network segment or point-to-point connection) connected to a host. These addresses are not routable. Like private addresses, these addresses cannot be the source or destination of packets traversing the internet. These addresses are primarily used for address autoconfiguration (Zeroconf) when a host cannot obtain an IP address from a DHCP server or other internal configuration methods.

When the address block was reserved, no standards existed for address autoconfiguration. Microsoft created an implementation called Automatic Private IP Addressing (APIPA), which was deployed on millions of machines and became a de facto standard. Many years later, in May 2005, the IETF defined a formal standard in RFC 3927, entitled *Dynamic Configuration of IPv4 Link-Local Addresses*.

Loopback

The class A network 127.0.0.0 (classless network 127.0.0.0/8) is reserved for loopback. IP packets whose source addresses belong to this network should never appear outside a host. The modus operandi of this network expands upon that of a loopback interface:

- IP packets whose source and destination addresses belong to the network (or subnetwork) of the same loopback interface are returned to that interface;
- IP packets whose source and destination addresses belong to networks (or subnetworks) of different interfaces of the same host, one of them being a loopback interface, are forwarded regularly.

Addresses ending in 0 or 255

Networks with subnet masks of at least 24 bits, i.e. Class C networks in classful networking, and networks with CIDR suffixes /24 to /32 (255.255.255.0–255.255.255.255) may not have an address ending in 0 or

255.

Classful addressing prescribed only three possible subnet masks: Class A, 255.0.0.0 or /8; Class B, 255.255.0.0 or /16; and Class C, 255.255.255.0 or /24. For example, in the subnet 192.168.5.0/255.255.255.0 (192.168.5.0/24) the identifier 192.168.5.0 commonly is used to refer to the entire subnet. To avoid ambiguity in representation, the address ending in the octet *0* is reserved.

A broadcast address is an address that allows information to be sent to all interfaces in a given subnet, rather than a specific machine. Generally, the broadcast address is found by obtaining the bit complement of the subnet mask and performing a bitwise OR operation with the network identifier. In other words, the broadcast address is the last address in the address range of the subnet. For example, the broadcast address for the network 192.168.5.0 is 192.168.5.255. For networks of size /24 or larger, the broadcast address always ends in 255.

However, this does not mean that every address ending in 0 or 255 cannot be used as a host address. For example, in the /16 subnet 192.168.0.0/255.255.0.0, which is equivalent to the address range 192.168.0.0–192.168.255.255, the broadcast address is 192.168.255.255. One can use the following addresses for hosts, even though they end with 255: 192.168.1.255, 192.168.2.255, etc. Also, 192.168.0.0 is the network identifier and must not be assigned to an interface.[6] The addresses 192.168.1.0, 192.168.2.0, etc., may be assigned, despite ending with 0.

In the past, conflict between network addresses and broadcast addresses arose because some software used non-standard broadcast addresses with zeros instead of ones.[7]

In networks smaller than /24, broadcast addresses do not necessarily end with 255. For example, a CIDR subnet 203.0.113.16/28 has the broadcast address 203.0.113.31.

Address resolution

Hosts on the Internet are usually known by names, e.g., www.example.com, not primarily by their IP address, which is used for routing and network interface identification. The use of domain names requires translating, called *resolving*, them to addresses and vice versa. This is analogous to looking up a phone number in a phone book using the recipient's name.

The translation between addresses and domain names is performed by the Domain Name System (DNS), a hierarchical, distributed naming system which allows for subdelegation of name spaces to other DNS servers.

Address space exhaustion

Since the 1980s, it was apparent that the pool of available IPv4 addresses was being depleted at a rate that was not initially anticipated in the original design of the network address system.[8] The threat of exhaustion was the motivation for remedial technologies, such as classful networks, Classless Inter-Domain Routing (CIDR) methods, and network address translation (NAT). Eventually, IPv6 was created, which has many more addresses available.

Several market forces accelerated IPv4 address exhaustion:

- Rapidly growing number of Internet users
- Always-on devices — ADSL modems, cable modems
- Mobile devices — laptop computers, PDAs, mobile phones

Some technologies mitigated IPv4 address exhaustion:

- Network address translation (NAT) is a technology that allows a private network to use one public IP address. It permits private addresses in the private network.
- Use of private networks
- Dynamic Host Configuration Protocol (DHCP)
- Name-based virtual hosting of web sites
- Tighter control by regional Internet registries over the allocation of addresses to local Internet registries
- Network renumbering to reclaim large blocks of address space allocated in the early days of the Internet

The primary address pool of the Internet, maintained by IANA, was exhausted on 3 February 2011, when the last 5 blocks were allocated to the 5 RIRs.[9][10] APNIC was the first RIR to exhaust its regional pool on 15 April 2011, except for a small amount of address space reserved for the transition to IPv6, which will be allocated under a much more restricted policy.[11]

The accepted and standard long term solution is to use Internet Protocol Version 6. The address size was increased in IPv6 to 128 bits, providing a vastly increased address space that also allows improved route aggregation across the Internet and offers large subnetwork allocations of a minimum of 2^{64} host addresses to end-users. However IPv4 only hosts cannot directly communicate with IPv6 only hosts so IPv6 alone does not provide an immediate solution to the IPv4 exhaustion problem. Migration to IPv6 is in progress but completion is expected to take considerable time.

Packet structure

An IP packet consists of a header section and a data section.

An IP packet has no data checksum or any other footer after the data section. Typically the link layer encapsulates IP packets in frames with a CRC footer that detects most errors, and typically the end-to-end TCP layer checksum detects most other errors.[12]

Header

The IPv4 packet header consists of 14 fields, of which 13 are required. The 14th field is optional (red background in table) and aptly named: options. The fields in the header are packed with the most significant byte first (big endian), and for the diagram and discussion, the most significant bits are considered to come first (MSB 0 bit numbering). The most significant bit is numbered 0, so the version field is actually found in the four most significant bits of the first byte, for example.

IPv4 Header Format

Offsets Octet		0								1								2								3					
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	
0	0	Version				IHL				DSCP						ECN		Total Length													
4	32	Identification																Flags			Fragment Offset									
8	64	Time To Live								Protocol								Header Checksum													
12	96	Source IP Address																													
16	128	Destination IP Address																													
20	160	Options (if IHL > 5)																													

Version

The first header field in an IP packet is the four-bit version field. For IPv4, this has a value of 4 (hence the name IPv4).

Internet Header Length (IHL)

The second field (4 bits) is the Internet Header Length (IHL), which is the number of 32-bit words in the header. Since an IPv4 header may contain a variable number of options, this field specifies the size of the header (this also coincides with the offset to the data). The minimum value for this field is 5 (RFC 791), which is a length of 5×32 = 160 bits = 20 bytes. Being a 4-bit value, the maximum length is 15 words (15×32 bits) or 480 bits = 60 bytes.

Differentiated Services Code Point (DSCP)

Originally defined as the Type of service field, this field is now defined by RFC 2474 for Differentiated services (DiffServ). New technologies are emerging that require real-time data streaming and therefore make use of the DSCP field. An example is Voice over IP (VoIP), which is used for interactive data voice exchange.

Explicit Congestion Notification (ECN)

This field is defined in RFC 3168 and allows end-to-end notification of network congestion without dropping packets. ECN is an optional feature that is only used when both endpoints support it and are willing to use it. It is only effective when supported by the underlying network.

Total Length

This 16-bit field defines the entire packet (fragment) size, including header and data, in bytes. The minimum-length packet is 20 bytes (20-byte header + 0 bytes data) and the maximum is 65,535 bytes — the maximum value of a 16-bit word. The largest datagram that any host is required to be able to reassemble is 576 bytes, but most modern hosts handle much larger packets. Sometimes subnetworks impose further restrictions on the packet size, in which case datagrams must be fragmented. Fragmentation is handled in either the host or router in IPv4.

Identification

This field is an identification field and is primarily used for uniquely identifying the group of fragments of a single IP datagram. Some experimental work has suggested using the ID field for other purposes, such as for adding packet-tracing information to help trace datagrams with spoofed source

addresses,[13] but RFC 6864 now prohibits any such use.

Flags

A three-bit field follows and is used to control or identify fragments. They are (in order, from high order to low order):

- bit 0: Reserved; must be zero.[note 1]
- bit 1: Don't Fragment (DF)
- bit 2: More Fragments (MF)

If the DF flag is set, and fragmentation is required to route the packet, then the packet is dropped. This can be used when sending packets to a host that does not have sufficient resources to handle fragmentation. It can also be used for Path MTU Discovery, either automatically by the host IP software, or manually using diagnostic tools such as ping or traceroute.

For unfragmented packets, the MF flag is cleared. For fragmented packets, all fragments except the last have the MF flag set. The last fragment has a non-zero Fragment Offset field, differentiating it from an unfragmented packet.

Fragment Offset

The fragment offset field, measured in units of eight-byte blocks (64 bits), is 13 bits long and specifies the offset of a particular fragment relative to the beginning of the original unfragmented IP datagram. The first fragment has an offset of zero. This allows a maximum offset of $(2^{13} – 1) \times 8 = 65{,}528$ bytes, which would exceed the maximum IP packet length of 65,535 bytes with the header length included (65,528 + 20 = 65,548 bytes).

Time To Live (TTL)

An eight-bit time to live field helps prevent datagrams from persisting (e.g. going in circles) on an internet. This field limits a datagram's lifetime. It is specified in seconds, but time intervals less than 1 second are rounded up to 1. In practice, the field has become a hop count—when the datagram arrives at a router, the router decrements the TTL field by one. When the TTL field hits zero, the router discards the packet and typically sends an ICMP Time Exceeded message to the sender.

The program traceroute uses these ICMP Time Exceeded messages to print the routers used by packets to go from the source to the destination.

Protocol

This field defines the protocol used in the data portion of the IP datagram. The Internet Assigned Numbers Authority maintains a list of IP protocol numbers which was originally defined in RFC 790.

Header Checksum

The 16-bit checksum field is used for error-checking of the header. When a packet arrives at a router, the router calculates the checksum of the header and compares it to the checksum field. If the values do not match, the router discards the packet. Errors in the data field must be handled by the encapsulated protocol. Both UDP and TCP have checksum fields.

When a packet arrives at a router, the router decreases the TTL field. Consequently, the router must

calculate a new checksum. RFC 1071 defines the checksum calculation:

> *The checksum field is the 16-bit one's complement of the one's complement sum of all 16-bit words in the header. For purposes of computing the checksum, the value of the checksum field is zero.*

For example, consider Hex 4500003044224000800600008c7c19acae241e2b (20 bytes IP header):

Step 1) 4500 + 0030 + 4422 + 4000 + 8006 + 0000 + 8c7c + 19ac + ae24 + 1e2b = 0002`BBCF (16-bit sum)

Step 2) 0002 + BBCF = BBD1 = 1011101111010001 (1's complement 16-bit sum)

Step 3) ~BBD1 = 0100010000101110 = 442E (1's complement of 1's complement 16-bit sum)

To validate a header's checksum the same algorithm may be used – the checksum of a header which contains a correct checksum field is a word containing all zeros (value 0):

2BBCF + 442E = 2FFFD. 2 + FFFD = FFFF. the 1'S of FFFF = 0.

Source address

This field is the IPv4 address of the sender of the packet. Note that this address may be changed in transit by a network address translation device.

Destination address

This field is the IPv4 address of the receiver of the packet. As with the source address, this may be changed in transit by a network address translation device.

Options

The options field is not often used. Note that the value in the IHL field must include enough extra 32-bit words to hold all the options (plus any padding needed to ensure that the header contains an integer number of 32-bit words). The list of options may be terminated with an EOL (End of Options List, 0x00) option; this is only necessary if the end of the options would not otherwise coincide with the end of the header. The possible options that can be put in the header are as follows:

Field	Size (bits)	Description
Copied	1	Set to 1 if the options need to be copied into all fragments of a fragmented packet.
Option Class	2	A general options category. 0 is for "*control*" options, and 2 is for "*debugging and measurement*". 1, and 3 are reserved.
Option Number	5	Specifies an option.
Option Length	8	Indicates the size of the entire option (including this field). This field may not exist for simple options.
Option Data	Variable	Option-specific data. This field may not exist for simple options.

- Note: If the header length is greater than 5, i.e. it is from 6 to 15, it means that the options field is present and must be considered.
- Note: Copied, Option Class, and Option Number are sometimes referred to as a single eight-bit field – the *Option Type*.

The following two options are discouraged because they create security concerns: Loose Source and Record Route (LSRR) and Strict Source and Record Route (SSRR). Many routers block packets containing these options.[14]

Data

The data portion of the packet is not included in the packet checksum. Its contents are interpreted based on the value of the Protocol header field.

Some of the common protocols for the data portion are listed below:

Protocol Number	Protocol Name	Abbreviation
1	Internet Control Message Protocol	ICMP
2	Internet Group Management Protocol	IGMP
6	Transmission Control Protocol	TCP
17	User Datagram Protocol	UDP
41	IPv6 encapsulation	ENCAP
89	Open Shortest Path First	OSPF
132	Stream Control Transmission Protocol	SCTP

See List of IP protocol numbers for a complete list.

Fragmentation and reassembly

The Internet Protocol enables networks to communicate with one another. The design accommodates networks of diverse physical nature; it is independent of the underlying transmission technology used in the Link Layer. Networks with different hardware usually vary not only in transmission speed, but also in the maximum transmission unit (MTU). When one network wants to transmit datagrams to a network with a smaller MTU, it may fragment its datagrams. In IPv4, this function was placed at the Internet Layer, and is performed in IPv4 routers, which thus only require this layer as the highest one implemented in their design.

In contrast, IPv6, the next generation of the Internet Protocol, does not allow routers to perform fragmentation; hosts must determine the path MTU before sending datagrams.

Fragmentation

When a router receives a packet, it examines the destination address and determines the outgoing interface to use and that interface's MTU. If the packet size is bigger than the MTU, and the Do not Fragment (DF) bit in the packet's header set to 0; the router may fragment the packet.

The router divides the packet into segments. The max size of each segment is the MTU minus the IP header size (20 bytes minimum; 60 bytes maximum). The router puts each segment into its own packet, each fragment packet having following changes:

- The *total length* field is the segment size.
- The *more fragments* (MF) flag is set for all segments except the last one, which is set to 0.
- The *fragment offset* field is set, based on the offset of the segment in the original data payload. This is

measured in units of eight-byte blocks.
- The *header checksum* field is recomputed.

For example, for an MTU of 1,500 bytes and a header size of 20 bytes, the fragment offsets would be multiples of (1500–20)/8 = 185. These multiples are 0, 185, 370, 555, 740, ...

It is possible for a packet to be fragmented at one router, and for the fragments to be fragmented at another router. For example, consider a packet with a data size of 4,500 bytes, no options, and a header size of 20 bytes. So the packet size is 4,520 bytes. Assume that the packet travels over a link with an MTU of 2,500 bytes. Then it will become two fragments:

Fragment	Total bytes	Header bytes	Data bytes	"More fragments" flag	Fragment offset (8-byte blocks)
1	2500	20	2480	1	0
2	2040	20	2020	0	310

Note that the fragments preserve the data size: 2480 + 2020 = 4500.

Note how we get the offsets from the data sizes:

- 0.
- 0 + 2480/8 = 310.

Assume that these fragments reach a link with an MTU of 1,500 bytes. Each fragment will become two fragments:

Fragment	Total bytes	Header bytes	Data bytes	"More fragments" flag	Fragment offset (8-byte blocks)
1	1500	20	1480	1	0
2	1020	20	1000	1	185
3	1500	20	1480	1	310
4	560	20	540	0	495

Note that the fragments preserve the data size: 1480 + 1000 = 2480, and 1480 + 540 = 2020.

Note how we get the offsets from the data sizes:

- 0.
- 0 + 1480/8 = 185
- 185 + 1000/8 = 310
- 310 + 1480/8 = 495

We can use the last offset and last data size to calculate the total data size: 495*8 + 540 = 3960 + 540 = 4500.

Reassembly

A receiver knows that a packet is a fragment if at least one of the following conditions is true:

- The "more fragments" flag is set. (This is true for all fragments except the last.)
- The "fragment offset" field is nonzero. (This is true for all fragments except the first.)

The receiver identifies matching fragments using the identification field. The receiver will reassemble the data from fragments with the same identification field using both the fragment offset and the more fragments flag. When the receiver receives the last fragment (which has the "more fragments" flag set to 0), it can calculate the length of the original data payload, by multiplying the last fragment's offset by eight, and adding the last fragment's data size. In the example above, this calculation was 495*8 + 540 = 4500 bytes.

When the receiver has all the fragments, it can put them in the correct order, by using their offsets. It can then pass their data up the stack for further processing.

Assistive protocols

The Internet Protocol is the protocol that defines and enables internetworking at the Internet Layer and thus forms the Internet. It uses a logical addressing system. IP addresses are not tied in any permanent manner to hardware identifications and, indeed, a network interface can have multiple IP addresses. Hosts and routers need additional mechanisms to identify the relationship between device interfaces and IP addresses, in order to properly deliver an IP packet to the destination host on a link. The Address Resolution Protocol (ARP) performs this IP-address-to-hardware-address translation for IPv4. (A hardware address is also called a MAC address.) In addition, the reverse correlation is often necessary. For example, when an IP host is booted or connected to a network it needs to determine its IP address, unless an address is preconfigured by an administrator. Protocols for such inverse correlations exist in the Internet Protocol Suite. Currently used methods are Dynamic Host Configuration Protocol (DHCP), Bootstrap Protocol (BOOTP) and, infrequently, reverse ARP.

See also

- Classful network
- Classless Inter-Domain Routing
- Internet Assigned Numbers Authority
- Legacy internet
- IPv6
- List of assigned /8 IPv4 address blocks
- List of IP protocol numbers
- Regional Internet Registry

Notes

1. ^ As an April Fools' joke, proposed for use in RFC 3514 as the "Evil bit".

References

1. ^ "BGP Analysis Reports" (http://bgp.potaroo.net/index-bgp.html). Retrieved 2013-01-09.
2. ^ ***a b*** "INET(3) man page" (http://www.unix.com/man-page/Linux/3/inet_addr/). Retrieved 2010-11-28.
3. ^ http://superuser.com/questions/736583/strange-dotless-decimal-notation-of-ip-address-how-does-it-work

4. ^ "Planning Classless Routing: TCP/IP" (http://technet.microsoft.com/en-us/library/cc779089%28WS.10%29.aspx). Technet.microsoft.com. 2003-03-28. Retrieved 2012-01-20.
5. ^ "HP Networking: switches, routers, wired, wireless, HP TippingPoint Security" (http://www.3com.com/other/pdfs/infra/corpinfo/en_US/501302.pdf). 3com.com. Retrieved 2012-01-20.
6. ^ Robert Braden (October 1989). "Requirements for Internet Hosts – Communication Layers" (http://tools.ietf.org/html/rfc1122#page-31). IETF. p. 31. RFC 1122 (https://tools.ietf.org/html/rfc1122).
7. ^ Robert Braden (October 1989). "Requirements for Internet Hosts – Communication Layers" (http://tools.ietf.org/html/rfc1122#page-66). IETF. p. 66. RFC 1122 (https://tools.ietf.org/html/rfc1122).
8. ^ "World 'running out of Internet addresses' " (http://technology.inquirer.net/infotech/infotech/view/20110121-315808/World-running-out-of-Internet-addresses). Retrieved 2011-01-23.
9. ^ Smith, Lucie; Lipner, Ian (3 February 2011). "Free Pool of IPv4 Address Space Depleted" (http://www.nro.net/news/ipv4-free-pool-depleted). Number Resource Organization. Retrieved 3 February 2011.
10. ^ ICANN,nanog mailing list. "Five /8s allocated to RIRs – no unallocated IPv4 unicast /8s remain" (http://mailman.nanog.org/pipermail/nanog/2011-February/032107.html).
11. ^ Asia-Pacific Network Information Centre (15 April 2011). "APNIC IPv4 Address Pool Reaches Final /8" (http://www.apnic.net/publications/news/2011/final-8). Retrieved 15 April 2011.
12. ^ RFC 1726 section 6.2
13. ^ Savage, Stefan. "Practical network support for IP traceback" (http://portal.acm.org/citation.cfm?id=347057.347560). Retrieved 2010-09-06.
14. ^ "Cisco unofficial FAQ" (http://www.faqs.org/faqs/cisco-networking-faq/section-23.html). Retrieved 2012-05-10.

External links

- RFC 791—Internet Protocol
- http://www.iana.org — Internet Assigned Numbers Authority (IANA)
- http://www.networksorcery.com/enp/protocol/ip.htm — IP Header Breakdown, including specific options
- RFC 3344 — IPv4 Mobility
- IPv6 vs. carrier-grade NAT/squeezing more out of IPv4 (http://www.networkworld.com/news/2010/060710-tech-argument-ipv6-nat.html)

Address exhaustion:

- RIPE report on address consumption as of October 2003 (http://www.ripe.net/rs/news/ipv4-ncc-20031030.html)
- Official current state of IPv4 /8 allocations, as maintained by IANA (http://www.iana.org/assignments/ipv4-address-space)
- Dynamically generated graphs of IPv4 address consumption with predictions of exhaustion dates—Geoff Huston (http://www.potaroo.net/tools/ipv4/index.html)
- IP addressing in China and the myth of address shortage (http://www.apnic.net/community/about-the-internet-community/internet-governance/articles/ip-addressing-in-china-2004)
- Countdown of remaining IPv4 available addresses (http://www.inetcore.com/project/ipv4ec

/index_en.html) (estimated)

Further Reading

Internet-Protocol-
Header explained

Retrieved from "http://en.wikipedia.org/w/index.php?title=IPv4&oldid=615291397"
Categories: Internet Protocol | Internet Standards | Internet layer protocols | Network layer protocols | IPv4
