

History of IP protocol

The days of IP protocol in its current format (IPv4) are numbered. From the middle of the 1990s the Internet was largely used by universities, high tech industries and the government, but the Internet increasingly interests commercial companies and will be used by a large number of individuals and systems all expressing different needs. For example: with the imminent convergence of the computer, networks, audiovisual and leisure industry, before long every television set will become equipment for accessing the Internet, enabling billions of individuals to enjoy for example videos on demand, teleshopping or electronic commerce. Under these circumstances IPv6 (also called IPng for *new generation IP*) should offer more flexibility and efficiency, solving a whole range of new problems and never have a lack of addresses. The main objectives of this new protocol were to:

- Support billions of computers by releasing itself from the inefficiency of space for current IP addresses,
- Reduce the size of routing tables
- Simplify the protocol to enable routers to route datagrams more quickly,
- Provide better security (authentication and confidentiality) than the current IP protocol,
- Pay more attention to the type of service and particularly the services associated to real time traffic,
- Facilitate multi-destination broadcasting by making it possible to specify the range,
- Allow a computer to move without changing its address,
- Allow future development of the protocol,
- Provide the old and new protocol with a peaceful coexistence.

IPv6 protocol

IPv6 protocol responds reasonably to the prescribed objectives. It retains the best functions from IPv4, while removing or minimising the worst and adding new ones when necessary.

In general, IPv6 is not compatible with IPv4, but is compatible with all other Internet protocols including TCP, UDP, ICMP, IGMP, OSPF, BGP and DNS; sometimes slight modifications are required (in particular when working with long addresses).

The main functions of IPv6

The major innovation in IPv6 is the use of longer addresses than with IPv4.

They are coded over 16 bytes and allow the problem which made IPv6 the order of the day to be resolved: to provide an almost unlimited collection of Internet addresses.

IPv4 makes it possible to address $2^{32}=4,29.10^9$ addresses while IPv6 makes it possible to address $2^{128}=3,4.10^{38}$ addresses.

The major improvement in IPv6 is the simplification of datagram headers. The basic IPv6 datagram header only contains 7 fields (as opposed to 14 for IPv4). This change enables routers to process datagrams faster and improves their overall speed.

The third improvement consists of offering more flexibility with options. This change is essential with the new header, since compulsory fields in the old version have now become optional.

In addition, the way in which options are represented is different; it enables routers to simply ignore options which are not intended for them. This function speeds up datagram processing times.

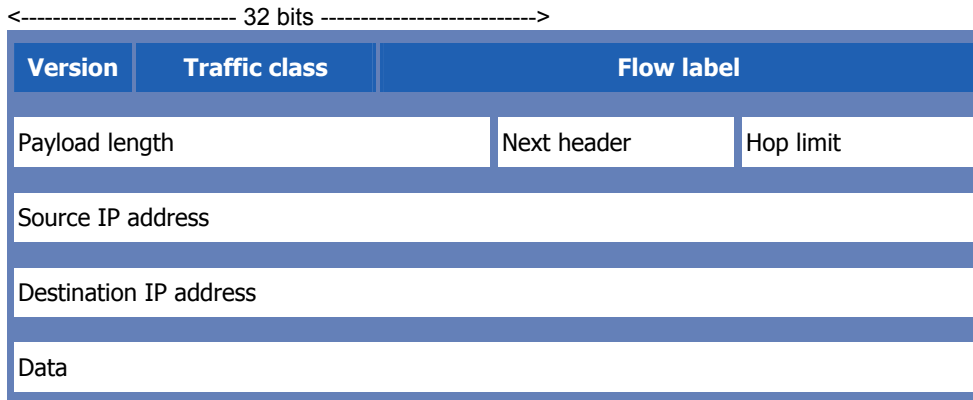
Furthermore, IPv6 provides greater security:

Authentication and confidentiality constitute the major security functions in the IPv6 protocol.

Finally, greater attention than in the past has been paid to the type of services. Although the Type of services field in the IPv4 datagram is only rarely used, the expected increase in multimedia traffic in the future requires that an interest be taken in it.

Basic datagram headers

Here is what an IPv6 datagram looks like:



Here are the meanings of the different fields:

- **The Version field** is always equal to 4 bits for IPv6. During the transition period from IPv4 to IPv6, routers must look at this field to know what type of datagram they are routing.
- **The Traffic class field** (coded over 8 bits) is used to distinguish the sources which must benefit from the flow control of others. Priorities 0 to 7 are allocated to sources capable of slowing down their speed in case of congestion. Values 8 to 15 are assigned to real time traffic (audio and video data included) where the speed is constant.
This distinction in the flows enables routers to react better in case of congestion. In each priority group, the lowest priority level relates to the least important datagrams.
- **The Flow label field** contains a unique number chosen by the source which aims to facilitate the work of the routers and allow the implementation of quality of service functions such as RSVP (*Resource reSerVation setup Protocol*). This indicator can be considered as a marker for a context in the router. The router can then conduct particular processing: choice of a route, processing of information in "real time", etc.
The flow label field can be filled with a random value which will be used to reference the context. The source will keep this value for all packets that it issues for this application and this destination. Processing is optimised since the router now only has to consult five fields to determine the origin of a packet. In addition, if a confidentiality extension is used, information relating to the port numbers is masked by intermediary routers.
- **The Payload limit field** of two bytes contains only the size of the payload, without taking into account the length of the header. For packets where the data size would be greater than 65,536 this field is worth 0 and the jumbogram option of the "hop by hop" extension is used.
- **The Next header field** has a function similar to the *protocol* field in the IPv4 packet: Quite simply it identifies the next header (in the same IPv6 datagram). It can be a protocol (from a higher layer ICMP, UDP, TCP, etc.) or an extension.
- **The Hop limit field** replaces the "TTL" (*Time-to-Live*) field in IPv4. Its value (over 8 bits) is decremented with each node crossed. If this value reaches 0 while the IPv6 packet crosses a router, it will be rejected and an ICMPv6 error message issued. It is used to prevent datagrams circulating

indefinitely. It plays the same role as the *Time to live* field in IPv4, namely that it contains a value representing the number of jumps or hops which is decremented with each passage through a router. In theory, in IPv4, there is a notion of times in second but no router uses it and so the name has changed to reflect the real use.

- Coming next, the **Source address** and **Destination address** fields.

After many discussions, it was decided that fixed length addresses equal to 16 bytes was the best compromise.

The first bits of the address - the prefix - define the type of address. The addresses beginning with 8 zeros are reserved, in particular for IPv4 addresses. Thus all addresses beginning with 8 zeros are reserved for IPv4 addresses. Two variants are supported; they are distinguished according to the following 16 bits (which is 16 bits at 0 or 1).

Geographical division using prefixes

The use of separate prefixes for addresses allocated to a provider and addresses allocated to a geographical area is a compromise between two different visions of the future of the Internet. Each of these providers has a reserved proportion of the address space (1/8 of this space). The first 5 bits which follow the prefix 010 are used to indicate on what "register" the access provider is found. Currently, there are three operational registers, for North America, Europe and Asia. Up to 29 new registers can be added later. Each register is free to divide the remaining 15 bytes as it sees fit. Another possibility is to use one byte to indicate the nationality of the provider and leave the following bytes free to define a structure for specific addresses.

The geographic model is the same as that for the current Internet network in which access providers do not play a great role. In this framework, IPv6 can manage 2 types of addresses.

Link and local site addresses only have a local specification. They can be reused by other organisations without any conflict. They cannot extend outside the limits of the organisation, which makes them well suited to those who use firewalls to protect their private network from the Internet.

Broadcast address

Multi-destination broadcast addresses have a Flag field (4 bits) and a Scope field (4 bits) following the prefix, then a group identification field (112 bits). One of the bits of the flag distinguishes permanent groups from temporary groups.

The Scope field allows a limited broadcast over a zone.

Anycast address

In addition to supporting standard point to point addresses (unicast) and multi-destination addresses (multicast) IPv6 supports a new type of first sight broadcast address (anycast).

This technique is similar to multi-destination broadcasting in the sense that the destination address is a group of addresses, but rather than try to deliver the datagram to all members in the group, it tries to deliver it to a single member of the group, that which is the closest or most capable of receiving it.

IPv6 notation

A new notation has been defined to describe IPv6 16 byte addresses. It comprises of 8 groups of 4 hexadecimal numbers separated by a colon. For example:

```
8000:0000:0000:0000:0123:4567:89AB:CDEF
```

Because several addresses have many zeros in their language, 3 optimisations have been defined. Firstly, the first zero in a group can be left out, as for example with 0123 which can be written 123. Then one or several of the groups of 4 consecutive zeros can be replaced by a double colon. So the address above becomes:

```
8000:::123:4567:89AB:CDEF
```

Finally, IPv4 addresses can be written using the representation of the address in decimal notation separated by dots and preceded by a double colon, for example:

::192.31.254.46

It is necessary to be more explicit about this address notation, but it should be known that there are a significant number of 16 byte addresses. More precisely, there are 2^{128} of them, which is approximately 3×10^{38} . If the whole world (land and sea together) was covered in computers, IPv6 could allocate 7×10^{23} addresses per m^2 .

The *Protocol* field is excluded because the *Next header* field from a datagram's last IP header specifies the type of protocol (for example, UDP or TCP).

All fields relating to fragmentation have been withdrawn, because IPv6 has a different approach to fragmentation.

To start with, all computers and routers conforming to IPv6 must support datagrams of 576 bytes. This rule places fragmentation into a secondary role. In addition, when a computer sends an IPv6 datagram which is too large, contrary to what happens with fragmentation, the router can not transmit it and returns an error message to the source. This message tells the source computer to break off sending of new datagrams to this destination. To have a computer which immediately transmits datagrams of the right size is much more efficient than to see routers fragmenting them on the fly.

Finally the *Checksum* field no longer exists because its calculation reduces performance too much. Indeed, the reliability of current networks, combined with the fact that the data link and transport layers conduct their own monitoring, the gain in quality of an additional checksum is not worth the price to be paid to calculate it.

Extension headers

This header supplies additional information in an efficient way. Each one is optional. If more than one header is present, they must appear immediately after the fixed header, preferably in the order of the list. Some headers have a fixed format; others contain a variable number of variable fields. For this, each item is coded as a triplet (Type, Length, Value). Type is a field of a byte which specifies the nature of the option. The different types have been chosen so that the first 2 bits tell routers that do not know how to execute the options what to do.

The choices are:

- skip the option
- destroy the datagram
- return an ICMP message to the source
- destroy the datagram without returning an ICMP message, which is a multi-destination datagram (in order to avoid to large a number of ICMP reports by return)

Length is a field of a byte. It specifies the size of the *Value* field (from 0 to 255) which contains unspecified information addressed to the recipient.

Hop by hop headers

Hop by hop headers contain information intended for all routers on the path.

Routing headers

Routing headers give the list of one or several routers that must be visited on the journey to the destination. Two forms of routing are implemented together: strict routing (the integral route is defined) and loose routing (only compulsory routers are defined).

The first four *routing* extension header fields contain 4 integers of a byte:

- the type of next header
- the type of routing (currently 0)
- the number of addresses present in the header (1 to 24)

- an address giving the next address to be visited.

This last field begins with a 0 value and is incremented with each address visited.

Fragmentation headers

Fragmentation headers process fragmentation in a similar way to [IPv4](#). The header contains the datagram ID, the fragment number and a bit specifying if there are other fragments to follow. In IPv6, contrary to [IPv4](#), only the source computer can fragment the datagram. The routers on the path cannot do so. This enables the source computer to fragment the datagram into pieces and use the Fragmentation header to transmit the pieces.

Authentication

The *Authentication* header provides a mechanism allowing the recipient of a datagram to ensure the source's identity. In [IPv4](#), no similar guarantee is given.

The use of data encryption for the datagram (its payload) reinforces its security; only the true recipient can read it.

When an originator and receiver want to communicate securely, they must firstly agree on one or several secret keys known only to them. A 32 bit key number is assigned to each of the 2 keys.

The key numbers are global so that, for example if A uses key 4 to communicate with B, A cannot use this key to communicate with C. Other parameters are associated with each key, such as its time to live, etc.

To send an authenticated message, the source computer firstly builds a datagram containing all the IP headers and the payload, then it replaces the fields which can be changed by 0s (for example: the *Hop limit* field). The datagram is completed with 0s to become a 16 byte multiple. In a similar way, the secret key used is also completed with 0s to become a 16 byte multiple. Then, a cipher checksum is calculated after concatenation of the completed security key, from the complete datagram and again, from the completed security key.

The *Authentication* header contains 3 parts. The first has 4 bytes specifying the number of the following header, the length of the authentication header and 16 zero bits. The second defines the 32 bit key number. The third contains the cipher checksum (with MD5 or other algorithm).

The recipient uses the key number to find the secret key. The completed value of the secret key is added before and after the payload itself is completed, the variable header fields are emptied of their zeros, then the cipher checksum is calculated. If the result of the calculation is equal to the cipher checksum contained in the Authentication header, the recipient is sure that the datagram indeed comes from the source with which it shares the secret key. It is also sure that the datagram hasn't been falsified without its knowledge in the background.

For datagrams which must be sent secretly, the extension header *Ciphered payload* must be used. This header begins with a 32 bit key number, followed by the ciphered payload.

Destination options

Destination option headers are used for fields that do not need to be interpreted and understood by the recipient computer. In the original version of IPv6, the only destination option that was defined was the null option. It makes it possible to complete this header with 0s to become an 8 byte multiple. This header will not be used initially. It has been defined to ensure that new routing software can take it into account, in the event that one day someone contemplates a destination option.