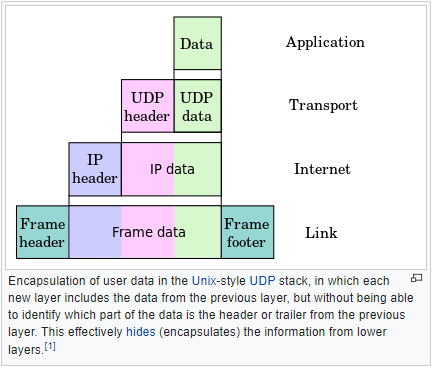
**IP Encapsulation:**



**IP Options:**

|  |  |  |
| --- | --- | --- |
| **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). |
| Option Data | Variable | Option-specific data. |

* **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*.**

**1.Loose source routing Option:**

Loose source routing uses a [source routing](https://en.wikipedia.org/wiki/Source_routing) option in [IP](https://en.wikipedia.org/wiki/Internet_Protocol) to record the set of [routers](https://en.wikipedia.org/wiki/Router_(computing)) a [packet](https://en.wikipedia.org/wiki/Packet_(information_technology)) must visit. The destination of the [packet](https://en.wikipedia.org/wiki/Packet_(information_technology)) is replaced with the next [router](https://en.wikipedia.org/wiki/Router_(computing)) the packet must visit.

The name loose source routing comes from the fact that only part of the path is set in advance.

**2. Strict source routing Option:**

**Strict source routing** is in contrast with loose source routing, in which every step of the route is decided in advance where the packet is sent.

**LSR and SSR** options are discouraged because many routers block packets containing these options.

**3.DoD Security Options:**

**CF CLASS # TYPE LENGTH DESCRIPTION**

1 0 2 130 var. DoD Basic Security

1 0 5 133 var. DoD Extended Security

**4.Timestamp Option:**

* The originating host must compose this option with a large enough timestamp data area to hold all the timestamp information expected. The size of the option does not change due to adding timestamps.
* The initial contents of the timestamp data area must be zero or internet address/zero pairs.
* If the timestamp data area is already full (the pointer exceeds the length) the datagram is forwarded without inserting the timestamp, but the overflow count is incremented by one.
* If there is some room but not enough room for a full timestamp to be inserted, or the overflow count itself overflows, the original datagram is considered to be in error and is discarded. In either case an ICMP parameter problem message may be sent to the source host.
* The timestamp option is **not copied upon fragmentation**. It is carried **in the first fragment**. Appears at

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **00** | **01** | **02** | **03** | **04** | **05** | **06** | **07** | **08** | **09** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** | **21** | **22** | **23** | **24** | **25** | **26** | **27** | **28** | **29** | **30** | **31** |
| Type | | | | | | | | Length | | | | | | | | Pointer | | | | | | | | Overflow | | | | Flag | | | |
| Timestamp [] ::: | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Type.** 8 bits. **Set to 68**.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **00** | **01** | **02** | **03** | **04** | **05** | **06** | **07** |
| C | Class | | Option number | | | | |

**C, Copy flag.** 1 bit. Cleared to 0.  
Indicates the option is not to be copied into all fragments.

**Class.** 2 bits. Set to 2.  
The option is a debugging and measurement option.

**Option.** 5 bits. Set to 4.  
The IP option number.

**Length.** 8 bits, unsigned. ? to 40.  
Size of the option in bytes.

**Pointer.** 8 bits, unsigned. 5 to ?  
The number of bytes from the beginning of this option to the end of *Timestamp[]* plus one (i.e., it points to the byte beginning the space for next timestamp). The timestamp area is full when *Pointer* is greater than *Length*.

**Overflow.** 4 bits, unsigned.  
The number of IP modules that cannot register timestamps due to lack of space.

**Flag.** 4 bits.

|  |  |
| --- | --- |
| **Flag** | **Description** |
| **0** | *Timestamp* contains only *Time* fields. |
| **1** | *Timestamp* contains *IP address* and *Time* fields. |
| **3** | The *IP address* fields are prespecified. An IP module only registers its timestamp if it matches its own address with the next specified internet address. |

**Timestamp [].** Variable length.  
One or more *Timestamp* structures.

**Timestamp.** 32 or 64 bits.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **00** | **01** | **02** | **03** | **04** | **05** | **06** | **07** | **08** | **09** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** | **21** | **22** | **23** | **24** | **25** | **26** | **27** | **28** | **29** | **30** | **31** |
| IP address | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Time | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**IP address.** 0 or 32 bits.  
Optional.

**Time.** 32 bits.  
A right-justified timestamp in milliseconds since midnight UT. If the time is not available in milliseconds or cannot be provided with respect to midnight UT then any time may be inserted as a timestamp provided the high order bit of the timestamp field is set to one to indicate the use of a non-standard value.