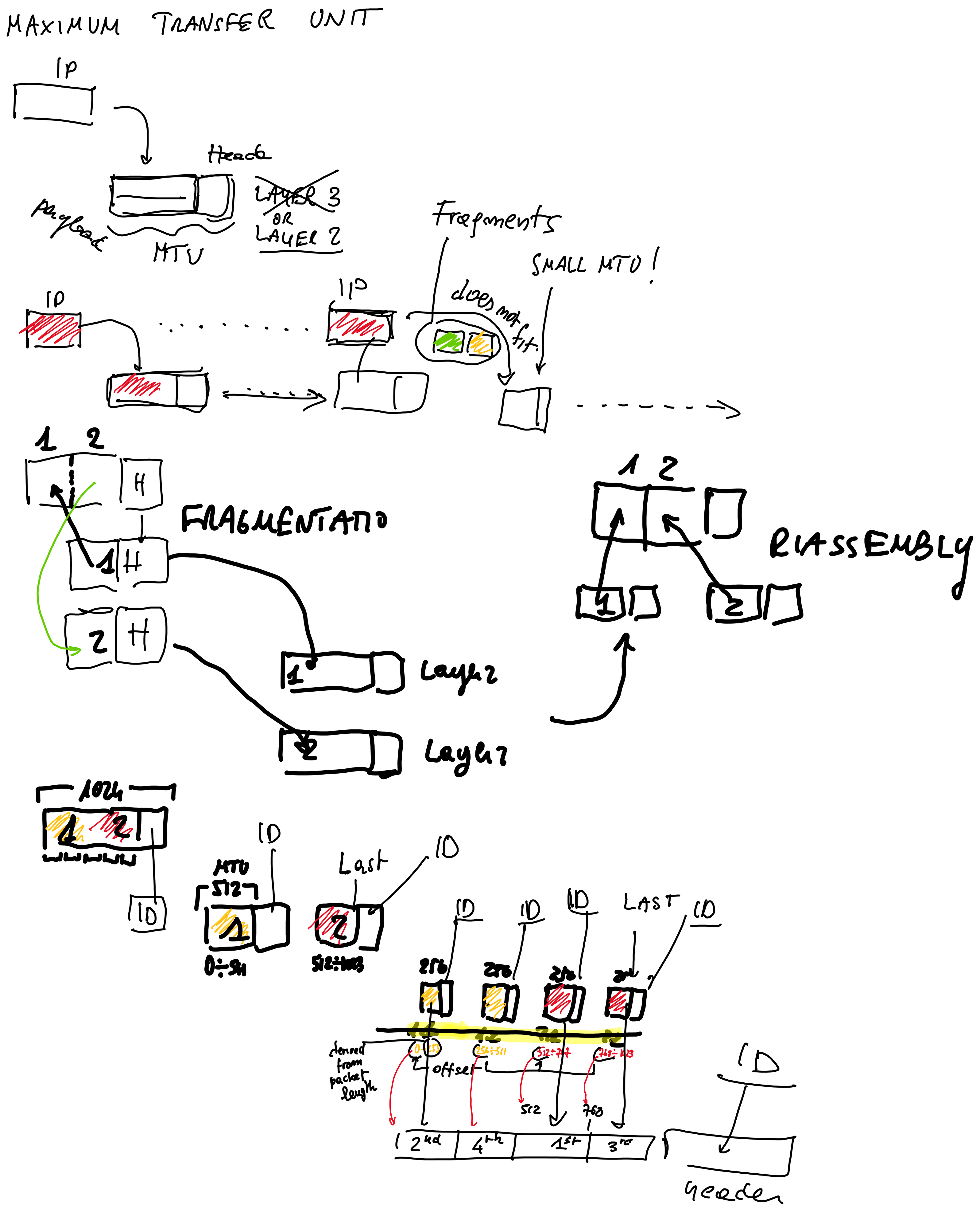
4.6 SIZE COMPATIBILITY

martedì 5 marzo 2024 15:32



FAST REASSEMBLY OF OUT OF ORDER FRAGMENTS

IP fragmentation is a process that occurs when a data packet is larger than the maximum transfer unit (MTU) of the network path through which it is being sent. When a packet is too large to be transmitted in a single piece over a network segment, it must be broken down into smaller pieces, known as fragments. These fragments can then be reassembled by the receiving host to construct the original packet.

The reason IP fragmentation uses an offset value rather than simply numbering the fragments in sequence is that IP packets may travel through the network via different paths to reach their destination. These paths might not deliver the fragments in the order they were sent, meaning that fragments can arrive out of sequence, which makes sequential numbering alone insufficient for proper reassembly.

Here's an example to illustrate this concept:

Imagine you're sending a large image file from your computer to a friend's computer, and due to its size, the image is divided into several IP packets. The network path your packets are traveling through has an MTU of 1500 bytes, but your image packets are 4096 bytes each.

The first packet would be fragmented into several smaller packets because it exceeds the MTU. Here's how it would work with the fragment offset:

- First Fragment: Contains the first 1480 bytes of data (taking into account the header size). The fragment offset is set to 0, indicating that this fragment contains the start of the original IP packet data.
- Second Fragment: Contains the next 1480 bytes of data. The fragment offset is set to 1480/8 = 185 (since the fragment offset field specifies the offset in 8-byte blocks).
- Third Fragment: Contains another 1480 bytes of data. The fragment offset for this fragment is 2 * 1480/8 = 370.
- Fourth Fragment: Contains the remaining 656 bytes of data (4096 - 3*1480 = 656). The fragment offset is now 3 * 1480/8 = 555.

Each fragment also has a flag to indicate whether more fragments are following this one. The last fragment has this flag set to 0, signaling that it is the last one.

Now, if these fragments arrive out of order—for instance, if the second fragment arrives before the first—the destination host can still reassemble them correctly using the fragment offsets. It knows that the first fragment has an offset of 0, so it begins the reassembly with this fragment. As other fragments arrive, it places them in the correct sequence according to their offset values, not the order in which they were received.

In contrast, if the fragments were just numbered sequentially without offsets and arrived out of order, the receiving host would not be able to determine where in the original packet each fragment belonged. It might end up placing the fourth fragment in the position of the second, for instance, leading to a corrupted reassembly of the packet.

By using fragment offsets, the IP protocol can accommodate the unordered delivery of fragments and ensure that the original packet is reconstructed correctly and completely, maintaining the integrity of the transmitted data.

REFRAGMENTATION REQUIREMENT

Imagine you are sending a data packet with a payload size of 4096 bytes from your computer to a friend's computer over the internet. The network path through which this packet is being sent has an MTU of 1500 bytes. Because the payload exceeds this MTU, the packet will need to be fragmented.

1. Initial Fragmentation:
   - First Fragment: Contains the first 1480 bytes of data (keeping in mind some bytes are reserved for the header). The fragment offset is set to 0, as it's the beginning of the data.
   - Second Fragment: Contains the next 1480 bytes of data. The fragment offset is calculated as 1480/8 = 185, since the fragment offset field specifies the offset in terms of 8-byte blocks.
   - Third Fragment: Contains another 1480 bytes of data. The fragment offset for this fragment is 2 * 1480/8 = 370.
   - Fourth Fragment: Contains the remaining 656 bytes of data (4096 - 3*1480 = 656). The fragment offset is now 3 * 1480/8 = 555.
2. Secondary Fragmentation: Now, suppose the second fragment, while en route to its destination, encounters a network segment with an MTU of only 1000 bytes and must be fragmented again.
   - Second Fragment - Part 1: Contains the first 980 bytes of data from the second fragment. Its fragment offset doesn't change; it remains 185.
   - Second Fragment - Part 2: Contains the remaining 500 bytes of the second fragment. To calculate the new offset, you have to add the initial offset of the second fragment to the amount of data contained in the first part of the newly fragmented second fragment: 185 + 980/8 = 307.

These fragments can now be transmitted through the smaller MTU network segment. The fragment offset for each fragment indicates its position relative to the start of the original unfragmented data payload. Regardless of the path each fragment takes, as long as the offset and the "more fragments" flag are intact, the destination host can reassemble the original packet correctly by using the offset to place each fragment in the right order.

If the fragments were only numbered sequentially without offsets, any re-fragmentation would lose the positional context needed for reassembly. The offset ensures that no matter how many times a fragment is further fragmented, its position within the original packet is known and it can be correctly reassembled at the destination.