Fragmentation :

Each network imposes some maximum size on its packets. These limits have various causes, among them:

1. Hardware (e.g., the size of an Ethernet frame).
2. Operating system (e.g., all buffers are 512 bytes).
3. Protocols (e.g., the number of bits in the packet length field).
4. Compliance with some (inter)national standard.
5. Desire to reduce error-induced retransmissions to some level.
6. Desire to prevent one packet from occupying the channel too long.

INTERNETWORKING

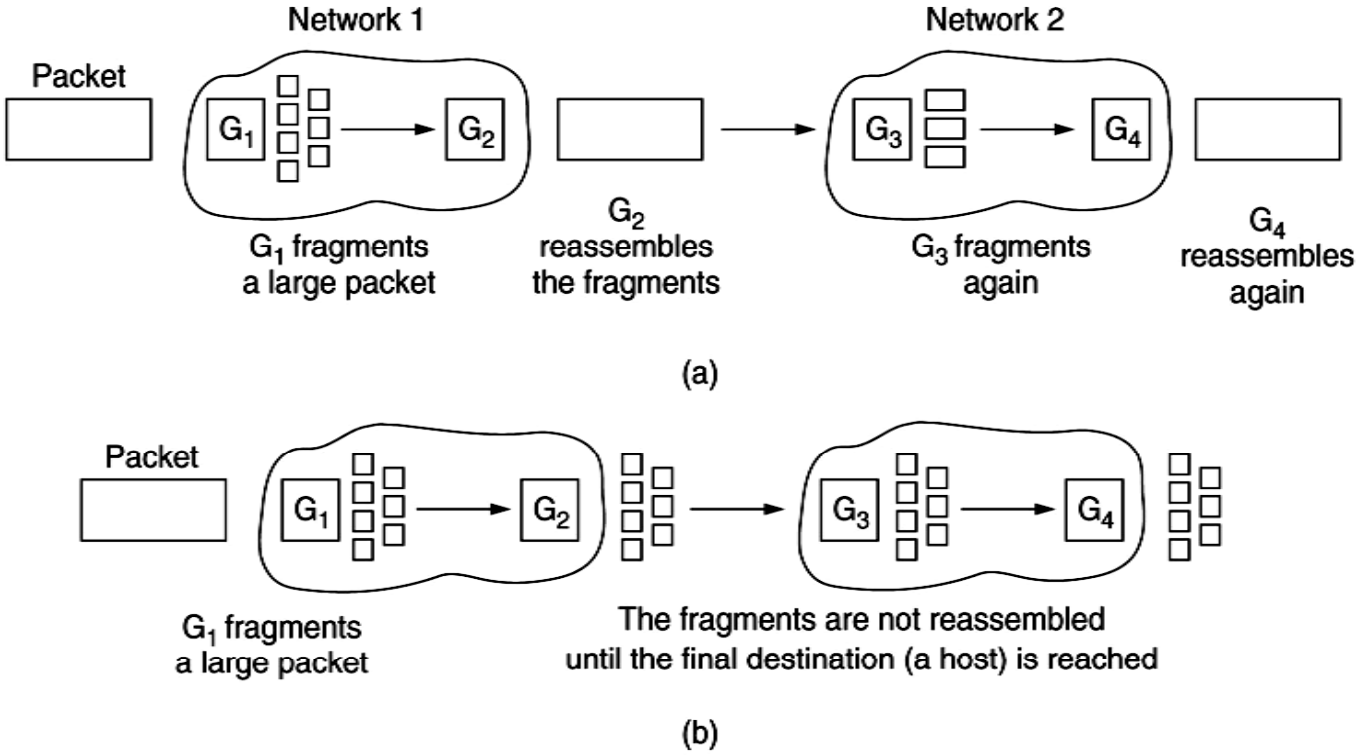
The result of all these factors is that the network designers are not free to choose any maximum packet size they wish. Maximum payloads range from 48 bytes (ATM cells) to 65,515 bytes (IP packets), although the payload size in higher layers is often larger.

An obvious problem appears when a large packet wants to travel through a network whose maximum packet size is too small. One solution is to make sure the problem does not occur in the first place. In other words, the internet should use a routing algorithm that avoids sending packets through networks that cannot handle them. However, this solution is no solution at all. What happens if the original source packet is too large to be handled by the destination network? The routing algorithm can hardly bypass the destination.

INTERNETWORKING

Basically, the only solution to the problem is to allow gateways to break up packets into fragments, sending each fragment as a separate internet packet. However, as every parent of a small child knows, converting a large object into small fragments is considerably easier than the reverse process. (Physicists have even given this effect a name: the second law of thermodynamics.) Packet-switching networks, too, have trouble putting the fragments back together again.

INTERNETWORKING



(a) Transparent fragmentation.
(b) Nontransparent fragmentation.

INTERNETWORKING

Two opposing strategies exist for recombining the fragments back into the original packet.

The first strategy is to make fragmentation caused by a "small- packet" network transparent to any subsequent networks through which the packet must pass on its way to the ultimate destination. This option is shown in Fig-(a). In this approach, the small-packet network has gateways (most likely, specialized routers) that interface to other networks. When an oversized packet arrives at a gateway, the gateway breaks it up into fragments.

INTERNETWORKING

Each fragment is addressed to the same exit gateway, where the pieces are recombined. In this way passage through the small-packet network has been made transparent. Subsequent networks are not even aware that fragmentation has occurred. ATM networks, for example, have special hardware to provide transparent fragmentation of packets into cells and then reassembly of cells into packets. In the ATM world, fragmentation is called segmentation; the concept is the same, but some of the details are different.

INTERNETWORKING

Transparent fragmentation is straightforward but has some problems. For one thing, the exit gateway must know when it has received all the pieces, so either a count field or an "end of packet" bit must be provided.

For another thing, all packets must exit via the same gateway. By not allowing some fragments to follow one route to the ultimate destination and other fragments a disjoint route, some performance may be lost.

A last problem is the overhead required to repeatedly reassemble and then refragment a large packet passing through a series of small-packet networks. ATM requires transparent fragmentation.

INTERNETWORKING

The other fragmentation strategy is to refrain from recombining fragments at any intermediate gateways. Once a packet has been fragmented, each fragment is treated as though it were an original packet. All fragments are passed through the exit gateway (or gateways), as shown in Fig.-(b). Recombination occurs only at the destination host. IP works this way.

Nontransparent fragmentation also has some problems. For example, it requires *every* host to be able to do reassembly. Yet another problem is that when a large packet is fragmented, the total overhead increases because each fragment must have a header. Whereas in the first method this overhead disappears as soon as the small-packet network is exited, in this method the overhead remains for the rest of the journey.

An advantage of nontransparent fragmentation, however, is that multiple exit gateways can now be used and higher performance can be achieved. Of course, if the concatenated virtual-circuit model is being used, this advantage is of no use.

INTERNETWORKING

When a packet is fragmented, the fragments must be numbered in such a way that the original data stream can be reconstructed. One way of numbering the fragments is to use a tree.

If packet 0 must be split up, the pieces are called 0.0, 0.1, 0.2, etc. If these fragments themselves must be fragmented later on, the pieces are numbers 0.0.0, 0.0.1, 0.0.2, . . . , 0.1.0, 0.1.1, 0.1.2, etc. If enough fields have been reserved in the header for the worst case and no duplicates are generated anywhere, this scheme is sufficient to ensure that all the pieces can be correctly reassembled at the destination, no matter what order they arrive in.

INTERNETWORKING

However, if even one network loses or discards packets, end-to-end retransmissions are needed, with unfortunate effects for the numbering system. Suppose that a 1O24-bit packet is initially fragmented into four equal-sized fragments, 0.0, 0.1,0.2, and 0.3. Fragment 0.1 is lost, but the other parts arrive at the destination.

Eventually, the source times out and retransmits the original packet again. Only this time Murphy's law strikes and the route taken passes through a network with a 512-bit limit, so two fragments are generated. When the new fragment 0.1 arrives at the destination, the receiver will think that all four pieces are now accounted for and reconstruct the packet incorrectly.

INTERNETWORKING

A completely different (and better) numbering system is for the internetwork protocol to define an elementary fragment size small enough that the elementary fragment can pass through every network. When a packet is fragmented, all the pieces are equal to the elementary fragment size except the last one, which may be shorter.

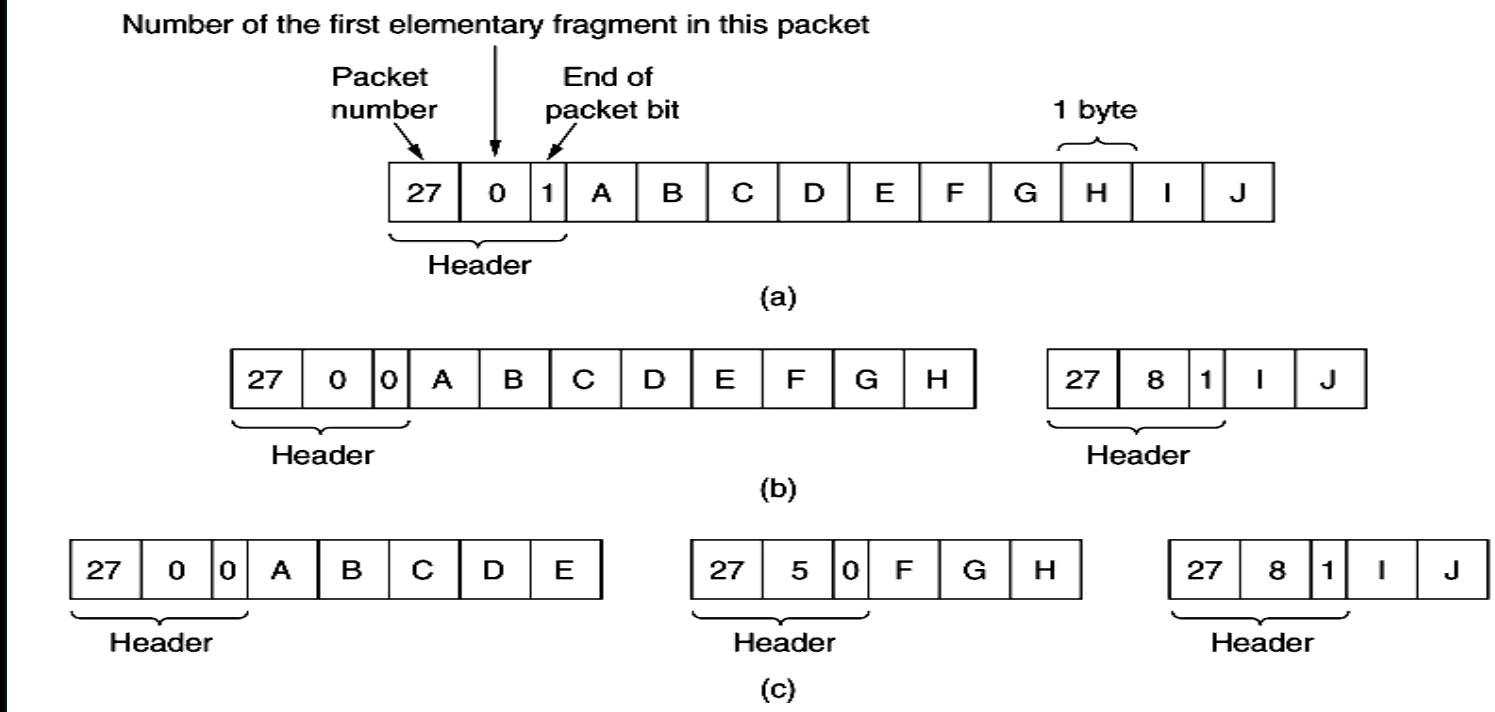
An internet packet may contain several fragments, for efficiency reasons. The internet header must provide the original packet number and the number of the (first) elementary fragment contained in the packet. As usual, there must also be a bit indicating that the last elementary fragment contained within the internet packet is the last one of the original packet.

INTERNETWORKING

This approach requires two sequence fields in the internet header: the original packet number and the fragment number. There is clearly a trade-off between the size of the elementary fragment and the number of bits in the fragment number.

Because the elementary fragment size is presumed to be acceptable to every network, subsequent fragmentation of an internet packet containing several fragments causes no problem. The ultimate limit here is to have the elementary fragment be a single bit or byte, with the fragment number then being the bit or byte offset within the original packet, as shown in Figure.

INTERNETWORKING



Fragmentation when the elementary data size is 1 byte.
(a) Original packet, containing 10 data bytes.
(b) Fragments after passing through a network with maximum packet size of 8 payload bytes plus header.
(c) Fragments after passing through a size 5 gateway.

INTERNETWORKING

Some internet protocols take this method even further and consider the entire transmission on a virtual circuit to be one giant packet, so that each fragment contains the absolute byte number of the first byte within the fragment.