over which the datagram must pass, and fragments the datagram if necessary. In Figure 7.10, for example, router R_1 will fragment a 1500-octet datagram before sending it over network 2.

We said that a host must insure a datagram can fit into a frame on the first network. Applications often try to choose a message size that is compatible with the underlying network. However, if an application chooses to send a large datagram, IP software on the host can perform fragmentation before sending it. In Figure 7.10, for example, if an application on host *A* creates a datagram larger than 1500 octets, IP software on the host will fragment the datagram before sending it. The point is:

> *IPv4 fragmentation occurs automatically at any point along the path when a datagram is too large for a network over which it must pass; the source only needs to insure that datagrams can travel over the first hop.*

How large should each fragment be? We said that each fragment must be small enough to fit in a single frame. In the example, a fragment must be six hundred twenty octets or smaller. A router could divide the datagram into fragments of approximately equal size. Most IPv4 software simply extracts a series of fragments that each fill the MTU, and then sends a final fragment of whatever size remains.

You may be surprised to learn that an IPv4 fragment uses the same format as a complete IPv4 datagram. The *FLAGS* field in the datagram header contains a bit that specifies whether the datagram is a complete datagram or a fragment. Another bit in the *FLAGS* field specifies whether more fragments occur (i.e., whether a particular fragment occupies the tail end of the original datagram). Finally, the *OFFSET* field in the datagram header specifies where in the original datagram the data in the fragment belongs. An interesting fragmentation detail arises because the *OFFSET* field stores a position in multiples of eight octets. That is, an octet offset is computed by multiplying the *OFFSET* field by eight. As a consequence, the size of the date in each fragment must be chosen to be a multiple of eight. Therefore, when performing fragmentation, IP chooses the amount of data in a fragment to be the largest multiple of eight less than or equal to the network MTU minus the datagram header size. Figure 7.11 illustrates IPv4 fragmentation with an MTU of 620 and a datagram header of 20 octets.

Fragmentation starts by replicating the original datagram header and then modifying the *FLAGS* and *OFFSET* fields. The headers in fragments 1 and 2 have the *more fragments* bit set in the *FLAGS* field; the header in fragment 3 has zero in the *more fragments* bit. Note: in the figure, data offsets are shown as octet offsets in decimal; they must be divided by eight to get the value stored in the fragment headers.

Each fragment contains a datagram header that duplicates most of the original datagram header (except for bits in the *FLAGS* field that specify fragmentation), followed by as much data as can be carried in the fragment while keeping the total length smaller than the MTU of the network over which it must travel and the size of the data a multiple of eight octets.