ECE 333: Introduction to Communication Networks
Fall 2002

Lecture 25:

- More on Internetworking and IP

Last time we introduced the topic of internetworking and discussed routing and addressing in IP. In this lecture we consider some other aspects of both internetworking in general and IP in particular.

Fragmentation and Reassembly

A basic problem when internetworking networks that are based on different protocols is that each network may have its own maximum packet size. For example in Ethernet can accept packets up to 15000 bytes, while FDDI packets may be up to 4500 bytes long.

Two possible approaches for dealing with this are

1. Use a small enough packet so that it can be carried over any network.

2. Allow packets that are too large for a network to be ***fragmented*** at routers before crossing over the network, i.e., a large packet is broken into several smaller packets; these packets must then be re-assembled into the original packet.

In IPv4, fragmentation is used.

Two different approaches to fragmentation - *transparent* and *nontransparent*

Transparent Fragmentation - the router doing the fragmentation breaks up oversized packets and addresses them all to the same exit router. The exit router re-assembles the fragments and forwards original packet. This is called transparent fragmentation because it is invisible to the other networks in the path and to the hosts.

Problems:

- All packets must exit at the same router; this is not guaranteed in a connectionless network.
- Overhead and delay involved with repeated fragmentation and reassembly.

Nontransparent Fragmentation - once a packet is fragmented, it is not reassembled until it reaches the destination.

Problems:

- Overhead is increased since the ratio of data to header is worse.
- Every host must be able to reassemble.
- Multiple layers of fragmentation are possible. (Some networks may need to fragment the fragments.)

In IPv4, non-transparent fragmentation is used.

General Fragmentation Problems

Fragments must be numbered so they can be reassembled in the same order.

What information is required? Must number all packets and indicate how many packets are in original.

One approach: use “tree numbering”:

> Packet 0 gets split into 0.0 0.1 0.2 0.3.
>
> Subsequently, packet 0.2 gets split into 0.2.0 and 0.2.1.
>
> When all packets arrive, they can be put together.

One problem with “tree numbering”:

> Suppose packet 0 is split into 4 pieces and one piece is lost (say 0.1). Also assume that the network provides reliable service through retransmissions. Upon retransmission of the packet, it takes a path that splits it in half. What happens when 0.1 of the retry arrives?

Better approach – label each fragment with fragment offset and total fragment length.

To reduce number of bits needed for fragment offset, can restrict fragments to be multiples of *b* bytes.

How large should *b* be?

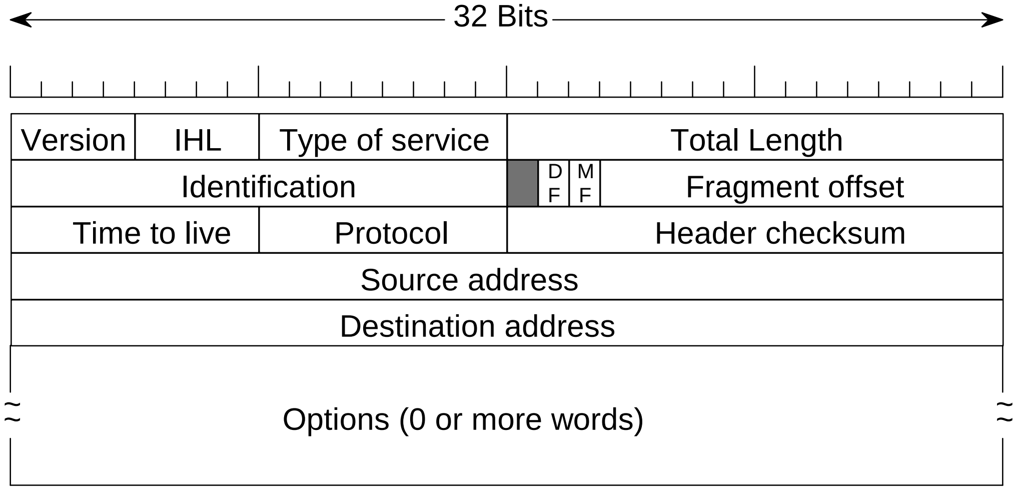
> The larger *b* is the fewer bits needed to indicate fragment offset.
>
> *b* should be chosen small enough so that a packet of size *b* can be transmitted over any network.

In IP, *b* = 8 bytes (sometimes this is referred to as the *elementary fragment size*).

The IPv4 Header

In IPv4, the maximum datagram length is 64 Kbytes. In practice this is usually set to 1500 bytes so all datagrams can be sent over Ethernet without further fragmentation.

The IPv4 header is 20-60 bytes and has the format shown below:



32 Bits

Version | IHL | Type of service | Total Length

Identification | DF | MF | Fragment offset

Time to live | Protocol | Header checksum

Source address

Destination address

Options (0 or more words)

IP Header Fields

Version - the IP version number (4 in this case). Having this field first allows subsequent fields to be properly interpreted.

IHL - IP header length- this gives the length of the header in 32-bit words, the minimum is 5 (20 bytes) and the maximum is 15 (60 bytes).

Type of service - This field allows for an indication of preference for high delay, high throughput or reliability. In practice this is often ignored. However several newer routers use this field to distinguish between different service classes of traffic. The service discipline used by the router is based on these classes.

Total length - this gives the length of the datagram in bytes including the header. Since this field is 16 bits, the maximum length of an IP datagram can be no more than 65,535 bytes.

Identification – When a datagram is fragmented, all pieces carry the same identification number.

Time To Live –This field is used to ensure that datagrams do not circulate in the network forever, for example due to a routing error. It is usually initialized to 64 and decremented at each router. When it reaches 0 the datagram is dropped.

IP Header Fields

Protocol – This field indicates the type of transport protocol, e.g. TCP, UDP, which generated the datagram.

Header Checksum – The header checksum is used to detect errors only in the header. This is a type of parity check code; it is simple to implement but only has a minimum distance of 2 (thus some double errors will not be detected). Because the TTL field is changed at each hop, the header checksum must be recomputed at each hop. If an error is detected, the datagram is discarded.

Source and destination addresses - these fields contain the 32-bit IP addresses discussed in lecture 24.

Options – The option field are not present in every datagram; these fields are used for things like source routing, route tracing, and security.

DF - "Don't fragment," this flag is set to 1 if the destination can't reassemble.

MF - "More fragments," this flag is set to 1 in all but the last fragment of a packet.

Fragment offset – This field contains the sequence number used for reordering fragments. It counts the offset of the payload data in units of 8 bytes - so 13 bits covers a maximum length IP packet.

Fragmentation in IPv4

When an IP packet is fragmented (due to frame length limits somewhere in the network), each fragment is sent as an IP packet. The header is repeated with appropriate modifications to the MF, Fragment Offset and Header Checksum fields. The options may or may not be repeated. Fragments are reassembled at the destination using the Fragment Offsets and the Identification.

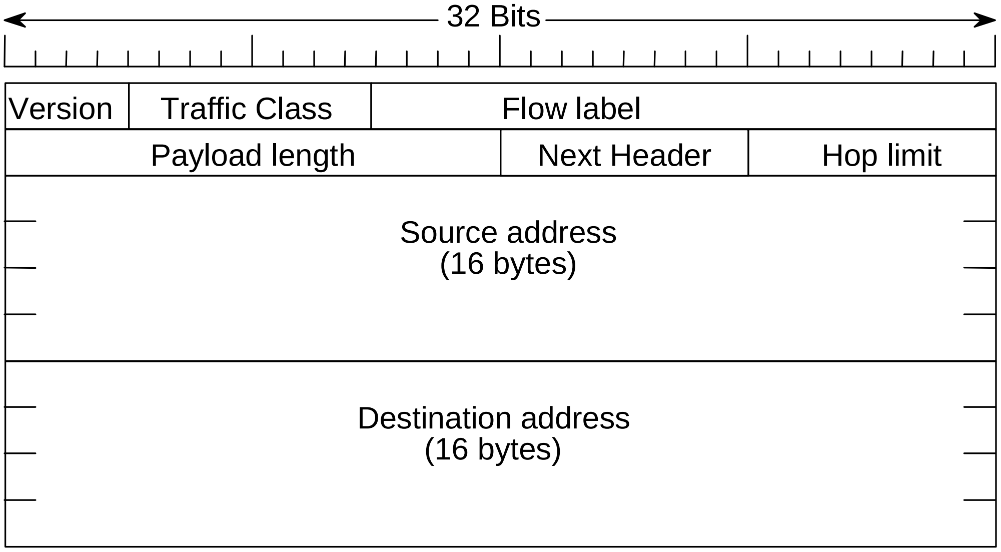
For example consider a 2500 byte datagram (including a 20 byte header) to be sent over a network that accepts a maximum payload of 1500 bytes. This datagram would be broken into 2 fragments. The first fragment would be sent as a datagram of size 1500 bytes, and the second fragment would be sent in a datagram of size 1020 bytes (1000 bytes of payload plus 20 bytes for the header). The first fragment would have a Fragment Offset of 0 and the MF flag set to 1. The second fragment would have a Fragment Offset of 185 (=1480 bytes) and the MF flag set to zero.

IPv6

In the early 1990's work began on developing a new version of IP. One of the main motivations for this was that the 32-bit address space was rapidly being used up. The result of this was a new protocol called IPv6.[1] Other motivations for a new protocol include the desire to better support newer applications, such as multimedia.

[1] Version 5 of IP was an experimental protocol.

Main IPv6 Header



The header is always 40 bytes. Other options can be specified through **extension headers**.

IPv6 header fields

Version - as in IPv4, this indicates the protocol version (6 in this case).

Traffic class - this is similar to the ToS field in IPv4.

Flow label - this field is used to indicate that certain packets belong to a "flow." Certain flows may receive special treatment. This may be used to provide a type of virtual circuit in an IP network.

Payload length - the length of the payload in bytes.

Next header- this fields specifies the type of information that follows the header -if an extension header is used it specifies the type of the extension header - if payload follows it specifies the transport layer protocol.

Hop limit - This is used in the same way as the TTL field in IPv4.

Changes in IPv6 format

- There are no header fields for fragmentation. In IPv6 only the sender can fragment a datagram. In this case an extension header is used to specify the fragment offset and other information. Since routers cannot fragment a packet, if a router receives to large of a packet it is dropped and a special control message is sent to the sender. The main reason for not allowing routers to fragment is to speed up forwarding within the network.

- There is no longer a checksum field present in the IPv6 header. This was also removed to speed up performance of routers. Also many link layer protocols already provide sufficient error checking when it is needed.

IPv6 address notation: instead of using the dotted-decimal notation, IPv6 addresses are commonly written using ***colon hexadecimal notation***. In this notation each group of 16 bits is written in hexadecimal with separated by colons, e.g. FF0C:0:0:1280:0:0:8C0A:C2.

For backward compatibility, any IPv6 address that begins with 96 zeros is assumed to contain an IPv4 address in the remaining 32 bits.