IEN: 41
Section: 2.3.2.1

INTERNETWORK PROTOCOL SPECIFICATION

Version 4

Jonathan B. Postel

June 1978

Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, California 90291

(213) 822-1511

TABLE OF CONTENTS

PREFACE

This is a draft specification of the Internet Protocol. Many people have contributed the concepts and ideas embodied in this specification, credit should go to at least the following: Vint Cerf, Danny Cohen, Dave Clark, Dick Watson, and Ray Tomlinson.

Internetwork Protocol Specification

Version 4

1. INTRODUCTION

The Internet Protocol is designed for use in interconnected systems of computer communication packet-switched networks. The internet protocol provides for transmitting segments of data from sources to destinations, where sources and destinations are identified by variable length addresses. The internet protocol also provides for fragmentation and reassembly of long segments, if necessary, for transmission through "small packet" networks.

1.1. History

This protocol has been developed as one result of the ARPA sponsored internetwork experiments program. The history until January 1978 is the history of the host-to-host protocol TCP.

The first publication of the ideas on which TCP is based was a paper in the IEEE Transactions on Communications by Cerf and Kahn in 1974 [1]. Later that year a protocol specification was published by a group led by Cerf at Stanford University [2]. A second specification was prepared in 1976 by a group led by Postel at SRI for the Defense Communication Agency for the AUTODIN II network [3]. In 1977 Cerf, at ARPA, prepared a substantial revision of the TCP specification [4]. Recently Postel revised Cerf's revision to distinguish the internet aspects from the host-to-host aspects [5].

Since January ideas about the internet protocol have continued to evolve an two documents were circulated by Postel [8] and Cerf [9]. The present specification draws on both of these and the discussions of the Internetwork Working Group. A brief memo on a revision of TCP in light of these developments was circulated by Cerf [10].

1.2. Scope

The internet protocol is specifically limited in scope to provide the functions necessary to deliver a package of bits (an internet segment) from a source to a destination over an inconnected system of networks. There are no mechanism to promote relibility, flow control, sequencing, or other services commonly found in host-to-host protocols.

The protocol is intended to be utilized in gateways that interconnect sets of networks.

1.3. Documentation

No documentation beyond that cited in the History Section (1.1) above is known. Those documents do provide some background, as do a series of working notes circulated in the ARPA research community. These notes are called Internetwork Experiment Notes (or IENs) and are collected into an Internet Notebook.

1.4. Interfaces

This protocol is called on by host-to-host protocols in an internet environment. This protocol calls on local network protocols to carry the internet packet to the next gateway or destination host.

For example a TCP module would call on the internet module to take a TCP segment (including the TCP header and user data) as the data portion of an internet segment. The TCP module would provide the addresses and other parameters in the internet header to the internet module as arguments of the call. The internet module would then create an internet segment and call on the local network interface to transmit the internet segment.

In the ARPANET case, for example, the internet module would call on a local net module which would add the 1822 leader [6] to the internet segment creating an ARPANET message to transmit to the IMP.

1.5. Operation

The internet protocol implements two basic functions: addressing, and fragmentation.

The internet modules use the addresses carried in the internet header to transmit the internet packets toward their destinations. The selection of a path for transmision is called routing. Routing is not a topic discussed by the internet protocol (at least not this version of it).

The internet modules use fields in the internet header to fragment and reassemble internet packets when necessary for transmission through "small packet" networks.

The model of operation is that an internet module resides in each host engaged in internet communication, and in each gateway that interconnects networks. These modules share common rules for interpreting address fields and for fragmenting and assembling internet packets. In addition these modules (especially in gateways) may have procedures for making routing decisions, and other functions.

The internet protocol uses four key mechanisms in providing its service: Type of Service, Time to Live, Options, and Header Checksum.

The type of service is used to indicate the quality of the service desired, this may be thought of as selecting between Interactive, Bulk, or Real Time, for example. This type of service indication is to be used by gateways to select the actual transmission parameters when routing an internet packet through a particular network.

The time to live is an indication of the lifetime of an internet packet. It is set by the sender of the packet and reduced at the points along the route where it is processed. If the time to live reaches zero before the internet packet reaches its destination the internet packet is destroyed. This time to live can be thought of as a self destruct time limit.

The options provide for control functions needed or useful in certain situations, but not needed for most communications. The options include provisions for timestamps, error reports, and special routing.

The header checksum provides a verification that the information used in processing internet packets has been transmitted correctly. The data may contain errors. If the header checksum fails the internet packet is discarded at once.

The internet protocol does not provide a reliable communication facility. There are no acknowledgements either end-to-end or hop-by-hop. There is no error control for data, only a header checksum. There are no retransmissions. There is no flow control.

The internet protocol treats each internet segment as an independent entity unrelated to any other internet segment. There are no connections, or logical circuits.

2. PHILOSOPHY

2.1. Lessons Learned

It is still very early in the game to say much about lessons learned, but we will make the following observations:

Addressing:

All addressing informations should be on the outermost envelope, i.e. in the internet header.

Fragmentation:

Fragmentation must be in the domain of the gateways, yet the gateways must have the least possible knowledge of end-to-end protocols.

Features:

The outermost protocol (i.e. internet protocol) must make no assumptions about the type of service the application desires.

For example, it would have been easy to have the header checksum be instead a internet segment checksum, but it might be desired by some application to have data delivered even if it contains errors.

2.2. Related Work

The TCP development cited in the History Section (1.1) is closely related to this work. Other work on the interconnection of networks can be found in the reports of the International Network Working Group [11].

2.3. Mechanisms Explained

Addressing

A distinction is made between names, addresses, and routes [12]. A name indicates what we seek. An address indicates where it is. A route indicates how to get there. The internet protocol deals only with addresses. It is the task of higher level (i.e. host-to-host or application) protocols to make the mapping from names to addresses. It is the task of lower level (i.e. local net or gateways) procedures to make the mapping from addresses to routes.

Addresses are variable length in multiples of octets, the source and destination addresses may be of different lengths. An address begins with an one octet network number. Following the network number the address is composed of fields appropriate to the specified network.

For example, in the ARPANET the network number is to be followed by an one octet IMP number and that followed by a two octet host number. It is expected that individual hosts will specify additional address fields to distinguish different protocol services and applications.

Fragmentation

Fragmentation of an internet segment may be necessary when it originates in a local net that allows a large packet size, and must traverse a local net that limits packets to a smaller size, to reach its destination.

An internet segment can be marked "don't fragment". Any internet segment so marked is not to be internet fragmented under any circumstances (however, intranet fragmentation may be used, that is a fragmentation and reassembly across a local network which is invisible to the internet protocol module). If such an internet segment can not be delivered to its destination without fragmenting it, it is to be discarded instead.

The internet protocol fragmentation procedure utilizes information in three fields of the internet header: the identification, the more-fragments-flag, and the fragment offset.

The sender of an internet segment sets the identification field to a value that must be unique for that source-destination pair for the time the segment will be active in the internetwork system. The originator of a complete segment sets the more-fragments-flag to zero, and the fragment offset to zero.

To fragment a long internet packet, an internet protocol module (for example in a gateway), creates two new internet packets and copies the contents of the internet header fields from the long packet into both new internet headers. The data of the long packet is divided into two portions on a 8 octet boundary (the second portion might not be an even multiple of 8 octets, but the first must be). Call the number of 8 octet blocks in the first portion NFB (for Number of Fragment Blocks). The first portion of the data is placed in the first new internet packet and the total length field is set to the correct value. The more-fragments-flag is set to one. The second portion of the data is placed in the second new internet packet and the total length field is set to the correct value. The more-fragments-flag carries the same value as the long packet. The fragment offset field of the second new internet packet is set to the value of that field in the long packet plus NFB.

This procedure can be generalized for an n-way split, rather than the two-way split described.

To assemble the fragments of an internet segment an internet protocol module (for example at a destination host) combines internet packets that all have the same value for the three fields: identification, destination, and source. The combination is done by placing the data portion of each fragment in the relative position indicated by the fragment offset in that fragment's internet header. The first fragment will have the fragment offset zero, and the last fragment will have the more-fragments-flag reset to zero.

2.4. Functional Specification of Interfaces

The following diagram illustrates the place of the internet protocol in the protocol hierarchy.

```
        +------+ +-----+ +-----+       +-----+
        !Telnet! ! FTP ! !Voice!  ...  !     !
        +------+ +-----+ +-----+       +-----+
            !       !       !             !
           +-----+       +-----+       +-----+
           ! TCP !       ! RTP !  ...  !     !
           +-----+       +-----+       +-----+
              !             !             !
           +-------------------------------+
           !       Internet Protocol       !
           +-------------------------------+
                          !
             +---------------------------+
             !   Local Network Protocol  !
             +---------------------------+
```

Protocol Relationships

Figure 1.

Internet protocol interfaces on one side to the higher level host-to-host protocols and on the other side to the local network protocol.

2.5. Problems Remaining

Major Items

A formal specification system must be selected and the formal specification created.

The protocol must be verified.

Implementation recommendations must be provided.

Examples and scenarios must be created.

Technical Points

Source Routing

It is thought that in some cases the sender may wish or need to specify the route to be traversed through the internetwork system rather than the address of the destination. Current plans call for an option to be developed to carry such information.

Longer Addresses

In some cases it may be desired to use even longer adddresses than are permitted in the regular internet header address fields.

Type of Service

The types of service classes (including priority) are yet to be defined.

Header Checksum

The header checksum algorithm must be defined.

Options

Additional options are to be defined.

Short Form Addresses

The addressing procedure of this version of the internet protocol requires that each address begin with the network field, it might be useful to allow short form of addressing when an internet message is to be delivered within a smaller environment.

Treatment of Errors

The development of error reporting conventions is needed.

2.6. Future Directions

???

2.7. Examples & Scenarios

???

3. SPECIFICATION

3.1. Formalisms Explained

???

3.2. Formal Specification

???

3.3. Internetwork Header Format

A summary of the contents of the internetwork header follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!Version!  IHL  !Header Checksum!          Total Length         !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!Type of Service! Identification!Flags!      Fragment Offset    !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Time to Live !    Pointer    !      DAL      !     SAL       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                      Destination Address                      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                         Source Address                        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                   Options                     !     Padding   !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Example Internet Packet Header

Figure 2.

Note that each tick mark represents one bit position.

Version: 4 bits

There is a Version field which indicates the "shape", or format, of the internet portion. This is version 4.

IHL: 4 bits

Internet Header Length is the length of the internet header in 32 bit words, and thus points to the beginning of the data.

Header Checksum: 8 bits

A checksum on the header only. Since some header fields may change this is recomputed and verified at each point the internet header is processed.

Total Length: 16 bits

Total Length is the length of the packet in octets including internet header and data.

Type of Service: 8 bits

Type of service.

```
  Bits 0-6: To be defined later.
  Bit 7:    Don't Fragment This Segment (DF).

     0 1 2 3 4 5 6 7
    +-+-+-+-+-+-+-+-+
    !. . . . . . . D!
    !. . . . . . . F!
    +-+-+-+-+-+-+-+-+
```

Identification: 8 bits

An identifying value assigned by the sender to aid in assembling the fragments of a segment.

Flags: 3 bits

Various Control Flags.

```
  Bit 0: Options Present (OP).
  Bit 1: To be defined later.
  Bit 3: More Fragments Flag (MF).

     0 1 2
    +-+-+-+
    !O . M!
    !P . F!
    +-+-+-+
```

Fragment Offset: 13 bits

This field indicates where in the segment this fragment belongs. The fragment offset is measured in units of 8 octets (64 bits).

Time to Live: 8 bits

This field indicates the maximum time the segment is allowed to remain the internetwork system. If this field contains the value zero then the segment should be destroyed. This field is modified in internet header processing. The time is measured in units of seconds.

Pointer: 8 bits

A pointer into the destination address which indicates the next octet to be used in address for route processing. This field may be modified in such processing.

DAL: 8 bits

Destination Address Length in octets.

SAL: 8 bits

Source Address Length in octets.

Destination Address: variable

The destination address, DAL octets in length.

Source Address: variable

The source address, SAL octets in length.

Options: variable

The option field is variable in length. The format is an option-type octet, a length octet, and the actual option octets. Although it is not clear that any option can be inserted at a point that could not also recompute the header checksum the ability to have unchecksummed options is provided.

The high order bit of the option-type octet, if set, indicates that the option should NOT be included in any checksum. The length octet, which follows, includes the option-type octet and the length octet in the octet count of the option length.

The option-type octet can be viewed as having 3 fields:

1 bit	checksum exclusion flag,
2 bits	option class,
5 bits	option number.

The option classes are:

0 = control
1 = internet error
2 = experimental debugging and measurement
3 = reserved for future use

The following internet options are defined:

CKSUM	CLASS	NUMBER	LENGTH	DESCRIPTION
0	0	0	-	End of Option list. This option occupies only 1 octet; it has no length octet.
0	0	1	-	Padding. This option occupies only 1 octet; it has no length octet.
0	1	1	-	General Error Report. Used to report errors in internet packet processing.
X	2	4	var.	Internet Timestamp. Used to accumulate timestamping information during internet transit. The length field is variable and may change as the internet packet traverses the networks and gateways of the internet system.
X	2	5	var.	Satellite Timestamp. Used as above for special satellite network testing.

Specific Option Definitions

End of Option List

```
+--------+
!00000000!
+--------+
```

This option code indicates the end of the option list. This might not coincide with the end of the internet header according to the internet header length.

Padding

```
+--------+
!00000001!
+--------+
```

This option code can be used between options, for example, to align the begining of a subsequent option on a word boundary.

General Error Report

```
+--------+--------+--------+--------+--------//--------+
!00100001! length !err code!   id   !                  !
+--------+--------+--------+--------+--------//--------+
```

The general error report is used to report an error dected in processing an internet packet to the originator of that packet. The "err code" indicates the type of error detected and the "id" is copied from the identification field of the packet in error, additional octets of error information may be present depending on the err code.

ERR CODE:

No specific err codes have been defined as yet.

Internet Timestamp

```
+--------+--------+--------+--------+--------//--------+
!x1000100! length !   ?    !   ?    !        ?         !
+--------+--------+--------+--------+--------//--------+
```

No information is available on the specific format of Timestamps.

Satellite Timestamp

```
+--------+--------+--------+--------+--------//--------+
!x1000101! length !   ?    !   ?    !        ?         !
+--------+--------+--------+--------+--------//--------+
```

No information is available on the specific format of Timestamps.

Padding: variable

The Padding field is used to ensure that the data begins on 32 bit boundary. The padding is zero.

3.4. Discussion

The basic internet service is datagram oriented, and provides for the fragmentation of packets at gateways, with reassembly taking place at the destination internet protocol module in the destination host. Of course, fragmentation and reassembly of datagrams within a network or by private agreement between the gateways of a network is also allowed since this is transparent to the internet protocols and the higher-level protocols. This transparent type of fragmentation and reassembly is termed "network-dependent" (or intranet) fragmentation and is not discussed further here.

Addressing is confined to the internet header at least for the current host-to-host protocols such as TCP and two undefined but planned protocols: datagram protocol (DGP) and real-time protocol (RTP). This strategy will better support multi-protocol synchronization when this is required for multi-media teleconferencing.

Addressing

The internet packet format has variable length source and destination address fields. It is expected that the address fields will be the complete specification of the address including specifying the network and host. In addition information to select a particular protocol process within a host and other multiplexing information, such as a port number, is expected to be carried in the address fields. Carrying the port addresses in the internet header allows the same socket addresses to be used under the control of different host level protocols, which will be advantageous in developing synchronization strategies for multi-media conferencing.

An address is a variable length quantity (in multiples of octets). It is intended for the first octet of an address to be interpreted as a network identifier, and that the rest of the address identifies a host within that network. The address field is allowed to be even longer than that with the view that a host may multiplex between several functions, or further route messages based on the additional address bits.

If a host were to support two instances of TCP they could be assigned distinct addresses by using an additional octet of address beyond that needed to identify the host. Other examples of such processes are the XNET (cross-network debugger) server process, the gateway control process, or the packet echoer process. There is also the possibility of placing another whole layer of addressing hierarchy in this position.

The 8 bit network number, which is the first octet of the variable length address, has a value as specified in RFC 739 [7]. In any case the latest information can be obtained from Jon Postel.

The format field of previous versions of the internet header used to distinguish the next higher level of protocol (e.g. TCP or RTP) has been eliminiated in favor of doing this demultiplexing of incoming internet segments to the proper protocol specific modules on the basis of the address field.

Fragmentation and Reassembly.

The internet identification field, (ID), is used to identify packet fragments for reassembly.

The flag bit MF is set if the packet is not the last fragment (i.e. there are More Fragments). The Fragment Offset field identifies the fragment number, relative to the beginning of the original unfragmented packet. Fragments are numbered in units of 8 octets. The fragmentation strategy is designed so than an unfragmented packet has all zero fragmentation information (MF = 0, fragment offset = 0). If an internet packet is fragmented, its text field must be broken on 8 octet boundaries.

In earlier versions of this scheme, it was proposed that only 8 bits be used for fragment offset. The fragments would be numbered in units of 64 octets. This would support packet lengths up to (2**8)*(2**6) = 2**14 = 16,384 octets. [Note that the segment total length field allows segments up to 2**16 = 65,536 octets.] This strategy could lead to serious inefficiencies in the transport of fragments, since the unit of fragmentation is so large (512 bits). For example, an ARPANET packet could hold at most one 512 bit portion of a segment. Since a typical internet header is most likely 224 bits long, fragments carried as type 3 ARPANET packets could have an efficiency of 512/(224+512) = 0.70 [and that is not counting the TCP header of 128 bits which would cause efficiency to drop to 384/(224+128+384) = 0.52 for the first fragment].

To reduce the potential inefficiency of fragmentation, this format allows 2**13 = 8192 fragments of 8 octets each for a total of 65,536 octets. [Note that this is consistent with the the segment total length field.] Fragmentation under this scheme has an efficiency of 736/(224+736) = 0.77 for internet packets carried in ARPANET type 3 packets [and 608/(224+128+608) = 0.63 for the data in the first fragment of a TCP segment]. Of course, efficiencies higher than this are possible for systems whose minimum packet size is larger than 1008 bits.

When fragmentation occurs, options are generally not copied, but remain with the first fragment. For concreteness, an example of a fragmented packet is illustrated in example 2 below.

The fields which may be affected by fragmentation include:

(1) option flag

(2) options field

(3) more fragments flag

(4) fragment offset

(5) internet header length field

(6) total length field

(7) header checksum

Type of Service

The type of service (TOS) is for internet service control. The Don't Fragment (DF) bit is set if internet fragmentation of this packet is NOT permitted. This can be used to prohibit fragmentation in cases where the receiving host does not have sufficient resources to reassembly internet fragments.

In the future this field is to carry information about priority, and Service Class. Current thinking suggests that service classes might be characterized by terms like Interactive, Bulk, and Real Time.

Time

The time to live is set by the sender to the maximum time the segment is allowed to be in the internetwork system. If the segment is in the internetwork system longer than that the segment should be destroyed. This field should be decreased at each point that the internet header processed to reflect the time spent processing the segment. Even if no local information is available on the time actually spent, the field should be decremented. The time is measured in units of seconds (i.e. the value 1 means one second). Thus the maximum time to live is 255 seconds or 4.25 minutes.

Options

The flag bit OP for Options Present is set if options are present in the internet header.

The options are just that, optional. That is, the presence or absence of an option is the choice of the sender, but each internet module must understand how to process every option.

Checksum

The internet header checksum is recomputed if the internet header is changed owing to additions or changes to internet options or due to fragmentation or a change to the address pointer field. This checksum at the raw internet level will protect the internet header fields from transmission errors.

The checksum algorithm is:

???

3.5. Examples

Example 1:

This is an example of a reasonably minimal (though not absolutely minimal) internet segment.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!Ver= 4 !IHL= 5 !Header Checksum!        Total Length = 28      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!    TOS = 1    !    Id = 111   !Flg=0!           Frag Off = 0  !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   Time = 123  !  Pointer = 2  !     DAL = 4   !     SAL = 4   !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                      destination address                      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                         source address                        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                              data                             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                              data                             !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Example Internet Packet Header

Figure 3.

Note that each tick mark represents one bit position.

Example 2:

In this example we show first an moderate size internet segment (7646 data octets), then two internet fragments that might result from the fragmentation of this segment.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!Ver= 4 !IHL= 6 !Header Checksum!      Total Length = 7670      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!    TOS = 1    !    Id = 111   !Flg=0!       Frag Off = 0      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   Time = 123  ! Pointer = 3   !    DAL = 5    !    SAL = 6    !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                      destination address                      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  dest. cont.  !               source address                  !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!          source  cont.        !          padding = 0          !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                             data                              !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                             data                              !
\                                                               \
\                                                               \
!                             data                              !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!             data              !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Example Internet Packet Header

Figure 4.

Now the first fragment that results from splitting the segment after 3824 data octets.

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   !Ver= 4 !IHL= 6 !Header Checksum!     Total Length = 3848       !
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   !    TOS = 0    !    Id = 111   !Flg=1!        Frag Off = 0     !
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   !   Time = 119  !  Pointer = 4  !    DAL = 5    !    SAL = 6    !
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   !                      destination address                      !
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   !  dest. cont.  !              source address                   !
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   !          source  cont.        !          padding = 0          !
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   !                               data                            !
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   !                               data                            !
   \                                                               \
   \                                                               \
   !                               data                            !
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   !                               data                            !
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Example Internet Packet Header

Figure 5.

And the second fragment.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!Ver= 4 !IHL= 6 !Header Checksum!       Total Length = 3846     !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!    TOS = 0     !   Id = 111   !Flg=0!           Frag Off = 49!
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   Time = 119   !  Pointer = 4  !   DAL = 5   !   SAL = 6      !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                       destination address                     !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  dest. cont.   !         source address                       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!          source  cont.           !          padding = 0       !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                             data                              !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                             data                              !
\                                                               \
\                                                               \
!                             data                              !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!             data                 !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Example Internet Packet Header

Figure 6.

Example 3:

In this example we show an example of a header containing options.

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 !Ver- 4 !IHL= 8 !Header Checksum!    Total Length = 1232        !
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 !    TOS = 1    !    Id = 111   !Flg=4!        Frag Off = 0     !
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 !  Time = 123   !  Pointer = 3  !    DAL = 6    !    SAL = 7    !
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 !                      destination address                      !
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 ! destination continued         !        source address         !
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 !             source  continued                 ! Opt. Code = x !
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 ! Opt. Len.=4   !          option value         ! Opt. Code = 1 !
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 ! Opt. Code = y !  Opt. Len.=3  !  option value ! Opt. Code = 0 !
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 !                             data                              !
 \                                                               \
 \                                                               \
 !                             data                              !
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 !                             data                              !
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Example Internet Packet Header

Figure 7.

4. VERIFICATION

Requires further research.

5. IMPLEMENTATION

5.1. What not to leave out.

???

5.2. User Interfaces

???

5.3. Mechanisms

???

5.4. Data Structures

???

5.5. Program sizes, performance data.

???

5.6. Test sequences, procedures, exerciser.

???

5.7. Parameter values: timeouts, segment sizes, buffer strategies.

???

5.8. Debugging

???

REFERENCES

[1] Vinton G. Cerf and Robert E. Kahn, "A Protocol for Packet Network Intercommunication," IEEE Transactions on Communications, volume COM-22, No. 5, May 1974, p. 637-648. (An early version of this paper appeared as INWG General Note 39, IFIP Working Group 6.1, September 1973).

[2] Vinton G. Cerf, Yogen K. Dalal, Carl Sunshine, "Specification of Internet Transmission Control Program," INWG General Note 72, IFIP Working Group 6.1, December 1974.

[3] Jonathan B. Postel, Larry L. Garlick, Raphael Rom, "Transmission Control Protocol Specification," Augmentation Research Center, Stanford Research Institute, Menlo Park, CA, 15 July 1976.

[4] Vinton G. Cerf, "Specification of Internet Transmission Control Program - TCP (Version 2)," IEN 5, March 1977.

[5] Vinton G. Cerf and Jonathan B. Postel, "Specification of Internetwork Transmission Control Program - TCP Version 3," Information Sciences Institute, IEN 21, January 1978.

[6] Bolt Beranek and Newman, "Specification for the Interconnection of a Host and an IMP," BBN technical Report 1822, January 1976 (Revised).

[7] J. Postel, "Assigned Numbers," RFC 739, NIC 42341, 11 November 1977.

[8] Jonathan B. Postel, "Draft Internetwork Protocol Specification - Version 2," Information Sciences Institute, IEN 28, February 1978.

[9] Vinton G. Cerf, "A Proposed New Internet Header Format," Advanced Research Projects Agency, IEN 26, February 1978.

[10] Vinton G. Cerf, "A Proposal for TCP Version 3.1 Header Format," Advanced Research Projects Agency, IEN 27, February 1978.

[11] INWG, the International Network Working Group, Chairman: Mr. Derek L. A. Barber, Project EIN, National Physical Laboratory, Teddington, Middlesex, England.

[12] John Shoch, "A Note On Inter-Network Naming, Addressing, and Routing," Xerox Palo Alto Research Center, IEN 19, January 1978.

GLOSSARY

1822

BBN Report 1822, "The Specification of the Interconnection of a Host and an IMP". The specification of interface between a host and the ARPANET.

Address

An address is a variable length quantity (in multiples of octets).

ARPANET message

The unit of transmission between a host and an IMP in the ARPANET. The maximum size is about 1012 octets (8096 bits).

ARPANET packet

A unit of transmission used internally in the ARPANET between IMPs. The maximum size is about 126 octets (1008 bits).

DAL

Destination Address Length, an internet header field, which specifies the destination address length in octets.

Destination

The (variable' length) destination address, an internet header field.

DF

The Don't Fragment bit carried in the type of service field.

DGP

DataGram Protocol: A host-to-host protocol for communication of raw data.

Flags

An internet header field carrying various control flags.

fragment

A portion of a logical unit of data, in particular an internet fragment is a portion of an internet segment.

Fragment Offset

This internet header field indicates where in the internet segment this fragment belongs.

header

Control information at the beginning of a message, segment, packet or block of data.

Identification

An internet header field identifying value assigned by the sender to aid in assembling the fragments of a segment.

IHL

The internet header field Internet Header Length is the length of the internet header measured in 32 bit words.

IMP

The Interface Message Processor, the packet switch of the ARPANET.

internet fragment

A portion of the data of an internet segment with an internet header.

internet packet

Either an internet segment or an internet fragment.

internet segment

The unit of data exchanged between a pair of internet modules (includes the internet header).

leader

Control information at the beginning of a message or block of data. In particular, in the ARPANET, the control information on an ARPANET message at the host-IMP interface.

MF

The More-Fragments Flag carried in the internet header Flags field.

module

An implementation, usually in software, of a protocol or other procedure.

more-fragments-flag

A flag indicating whether or not this internet packet contains the end of an internet segment, carried in the internet header Flags field.

NFB

The Number of Fragment Blocks in a portion of an internet packet. That is, the length of a portion of data measured in 8 octet units.

octet

An eight bit byte.

Options

The internet header Options field may contain several options, and each option may be several octets in length. The options are used primarily in testing situations, for example to carry timestamps.

packet

A package of data with a header which may or may not be logically complete. More often a physical packaging than a logical packaging of data.

Padding

The internet header Padding field is used to ensure that the data begins on 32 bit word boundary. The padding is zero.

RTP

Real Time Protocol: A host-to-host protocol for communication of time critical information.

SAL

Source Address Length, an internet header field, which specifies the source address length in octets.

segment

A logical unit of data, in particular an internet segment is the unit of data transfered between the internet module and a higher level module.

Source

The (variable length) source address, an internet header field.

TCP

Transmission Control Protocol: A host-to-host protocol for reliable communication in internetwork environments.

Total Length

The internet header field Total Length is the length of the packet in octets including internet header and data.

Type of Service

An internet header field which indicates the type of service for this internet fragment.

Version

The Version field indicates the format of the internet header.

XNET

A cross-net debugging protocol.