Internet Draft
Expiration: May 26, 1997

M.S. Corson
University of Maryland
V. Park
Naval Research Laboratory

November 26, 1997

An Internet MANET Encapsulation Protocol (IMEP) Specification
draft-ietf-manet-imep-spec-00.txt

Status of this Memo

Abstract

This memo describes a multipurpose network-layer protocol---named the Internet MANET Encapsulation Protocol (IMEP)---designed to support the operation of many routing algorithms or other upper layer protocols intended for use in Mobile Ad hoc Networks (MANET). The protocol incorporates mechanisms for supporting link status sensing, control message aggregation and encapsulation, broadcast reliability, network-layer address resolution and provides hooks for interrouter security authentication procedures. The IMEP also puts forth a framework or architecture for MANET router and interface identification and addressing.

The present specification is high-level and incomplete, giving only a behavioral protocol description and proposed packet format. This version of this draft is intended primarily to acquaint readers with the concept of the protocol. A subsequent revision will detail the protocol's mechanisms and data structures.

1. Introduction

The primary purpose of the Internet MANET Encapsulation Protocol (IMEP) is to improve overall network performance by reducing the "number" of network control message broadcasts through encapsulation and aggregation of multiple MANET control messages (e.g. routing protocol packets, acknowledgements, link status sensing messages, network-level address resolution, etc.) into larger IMEP messages. Usage of the IMEP is desirable because per-message, multiple access delay in contention-based schemes such as CSMA/CA, IEEE 802.11, FAMA etc. is significant, and thus favors the use of fewer, larger messages. It would also be useful in reservation-based, time-slotted access schemes where smaller packets must be aggregated into appropriately-sized IP packets for transmission in a given time slot. Upper layer protocols *other than routing* may make use of this encapsulation functionality for the same purpose.

Its secondary purpose concerns the commonality of certain functionality in many network-level routing algorithms. Many algorithms intended for use in a MANET will require common functionality such as link status sensing, security authentication with adjacent routers, broadcast reliability of network control messages, etc. This common functionality can be extracted from these individual protocols and put into a unified, generic protocol useful to all. MANET routing algorithms would also benefit from a common approach to router and interface identification and addressing, and this protocol provides a framework for unifying the protocols under a common architecture.

The IMEP will run at the network layer (see Figure 1), and will be an adjunct to whichever network routing protocol is using it. Routing control packets will be encapsulated in IMEP messages, which will be further encapsulated into IP packets.

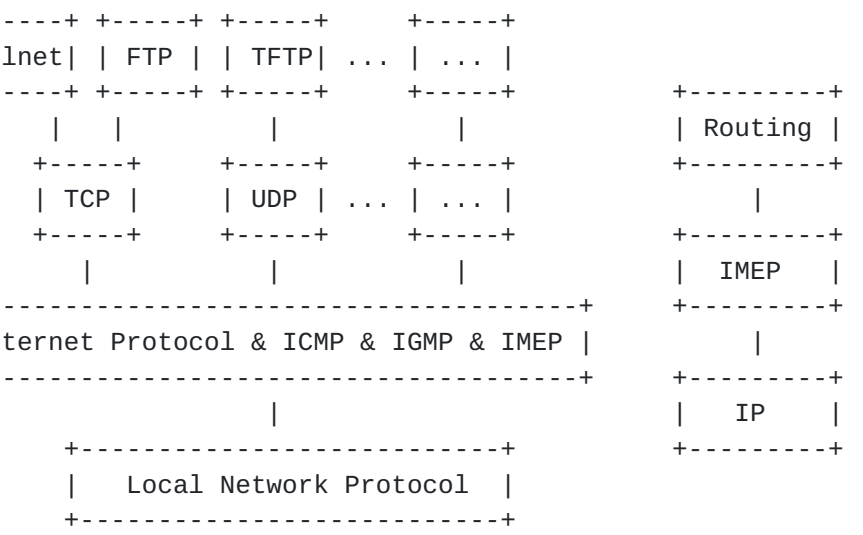


Figure 1

2.0 Terminology

This section provides definitions for the terminology used throughout this document. Many of these definitions may be replaced by or merged with those of the MANET working group's terminology draft [1] now under development.

MANET router or router:
: A device---identified by a "unique Router ID" (RID)---that executes a MANET routing protocol and, under the direction of which, forwards IP packets. It may have multiple interfaces, each identified by an IP address. Associated with each interface is a physical-layer communication device. These devices may employ wireless or hardwired communications, and a router may simultaneously employ devices of differing technologies. For example, a MANET router may have four interfaces with differing communications technologies: two hardwired (Ethernet and FDDI) and two wireless (spread spectrum and impulse radio).

medium:
: A communication channel such as free space, cable or fiber through which connections are established.

communications technology:
: The means employed by two devices to transfer information between them.

connection:

A physical-layer connection---which may be through a wired or wireless medium---between a device attached to an interface of one MANET router and a device utilizing the same communications technology attached to an interface on another MANET router. From the perspective of a given router, a connection is a (interface, adjacency) pair.

link:
A "logical connection" consisting of the logical *union* of one or more connections between two MANET routers. Thus a link may consist of a heterogeneous combination of connections through differing media using different communications technologies.

neighbor:
From the perspective of a given MANET router, a "neighbor" is any other router to which it is connected by a link.

adjacency:
The name given to an "interface on a neighboring router".

topology:
A network can be viewed abstractly as a "graph" whose "topology" at any point in time is defined by set of "points" connected by "edges". (This term comes from the branch of mathematics bearing the same name that is concerned with those properties of geometric configurations (such as point sets) which are unaltered by elastic deformations (such as stretching) that are homeomorphisms.)

physical-layer topology:
A topology consisting of connections (the edges) through the *same* communications medium between devices (the points) communicating using the *same* communications technology. Multiple physical-layer topologies may exist for a given medium and communications technology if adaptive or proactive power control, or other physical-layer mechanisms are employed.

network-layer topology:
A topology consisting of links (the edges) between MANET routers (the points) which is used as the basis for MANET routing. Since "links" are the logical union of physical-layer "connections", it follows that the "network-layer topology" is the logical union of the various "physical-layer topologies".

IP routing fabric:
The heterogeneous mixture of communications media and technologies through which IP packets are forwarded whose topology is defined by the network-layer topology.

3.0 Protocol Overview

The mechanisms contained in the IMEP are:

Link/Connection Status Sensing

Control Message Aggregation

Broadcast Reliability

Network-layer Address Resolution

Security Authentication

3.1 Link/Connection Status Sensing

3.1.1 Definition of Link/Connection Status

Many routing protocols require accurate knowledge of the status of links/connections between neighboring routers. "Link status" in the IP routing fabric is determined from the union of the status of physical-layer "connections" between interfaces.

The relationship of interfaces, adjacencies, connections and links is depicted in Figure 2 from the perspective of router i. Router i has two interfaces f1 and f2, each of which has a physical-layer connection with multiple interfaces attached to other routers---these interfaces are referred to as adjacencies from router i's perspective and are numbered with c's. In this figure, there are two connections (f1,c1) and (f2,c2), the logical union of which composes the logical link (i,k) between routers i and k.

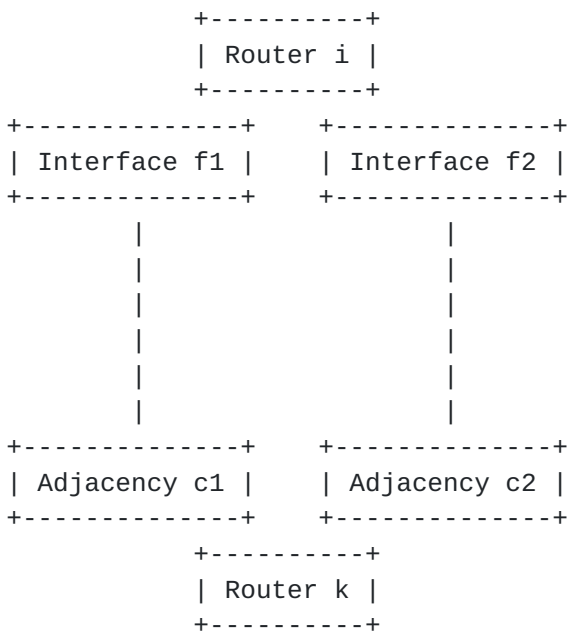


Figure 2: Shown from router i's perspective, interfaces f1 and f2 are connected to adjacencies c1 and c2 via connections (f1,c1) and (f1,c2)---the union of which forms link (i,k).

The status of an connection may be INcoming or OUTgoing (either of which meaning it is unidirectional) or BIdirectional. A unidirectional link is composed from one or more similarly-directed, unidirectional connections. A BIdirectional link may be composed from the union of one or more bidirectional connections, or two or more oppositely-directed, unidirectional connections, or some combination thereof. A link which is present (i.e. which has a non-null status, and is either uni or bidirectional), is referred to as "UP". A link which is not present (i.e. which has a null status) is referred to as "DOWN".

The IMEP may be configured to run in the following "connection notification" modes:

BI-directional:
: This mode requires that physical-layer connectivity between two interfaces be established in *both* IN and OUT directions before an connection is considered *present* and the upper layer routing protocol is subsequently notified.

UNI-directional:
: This mode requires that physical-layer connectivity between two interfaces need only be established in one direction (IN or OUT) before an connection is considered present and the upper layer routing protocol is subsequently notified.

As determined by the connection notification mode, the upper layer routing protocol is notified whenever there is a change (addition, modification, deletion) in the status of an interface's connections. This notification is implemented via a callback functions defined in the MANET routing policy/IMEP interface (more on this later)

3.1.2 Link/Connection Status Sensing Packet Exchange Mechanism

The IMEP uses a combination of BEACON and HELLO packets (and other packets to be described shortly) to ascertain connection (and indirectly link) status. On initialization, an interface under the control of IMEP broadcasts (the format of a BEACON packet is specified in section 4.0) to all adjacencies; i.e. interfaces that are only one hop away such as those on the same Ethernet subnet, or those within wireless transmission range of the broadcasting interface. (Note: Usage of the term "broadcast" here means to transmit a *single* copy of a packet to *all* interfaces reachable over one hop. As is the convention with other Internet routing protocols, this is done using IP multicast. An IP multicast address "ALL_IMEP_ROUTERS" will be reserved, and all MANET router interfaces will be configured to listen for this address.) a BEACON packet The purpose of a BEACON packet is to alert any adjacencies of the existence and identity of the broadcasting interface; an interface's identity is its IP address. The interface must ensure that a BEACON packet (or other "equivalent" packet, more on this later) is transmitted at least once every BEACON_PERIOD (BP) time units; i.e. no more than BP time units may pass between subsequent transmissions of a BEACON (or "BEACON-equivalent") packet.

Reception of a BEACON at an interface implies either reconfirmation or creation of "IN" (read "INcoming") status of a connection at that interface, depending on whether or not the connection already exists, respectively. Thus, BEACONs serve to tell a receiving interface that "someone else is out there." Once present, the status remains for MAX_BEACON_TIME (MBT) time units, at which time it expires (i.e. times out) if no subsequent BEACONs have been received; i.e. the link is declared DOWN and is removed from the data structures. Creation or loss of IN status may require notification of the upper level routing protocol, depending on whether or not the logical link status to which this connection is associated has been affected.

HELLO (or "HELLO-equivalent") packets are used to respond to BEACONs. The purpose of a HELLO packet is to let a "BEACONing" node know that someone hears its BEACON. A HELLO packet contains the identity (i.e. IP interface) of the interface broadcasting the HELLO and the identity of the BEACONing interface to which it is responding. A HELLO packet is generated immediately in response to a BEACON reception, and is placed in the "Awaiting Broadcast" (AB) buffer (more on the

functioning of this buffer later). Subsequently, as long as the interface is considered UP (i.e. IN or BI), a HELLO packet must be generated at least once every BP time units; i.e. no more than BP time units may pass between subsequent generations of a HELLO packet.

Reception of a HELLO at an interface implies either reconfirmation or creation of "BIdirectional" status of an connection at that interface, depending on whether or not the connection already exists, respectively. This is because reception of HELLO packet confirms that someone hears this interface (i.e. that is has OUTgoing status), and simultaneously confirms that it itself can receive them and, hence, also has INcoming status for that connection.

HELLO packets may be broadcast in one of two "Hello" modes:

Single Interface (SI):

An interface only sends HELLOs in response to BEACONs it receives. This is the standard mode which permits efficient link-layer detection of IN and BI connections. It also permits "network-layer detection" (by a routing protocol) of BIdirectional links composed of oppositely-directed, unidirectional links on the same or differing routers.

Multiple Interface (MI):

An interface sends HELLOs in response to BEACONs it receives, and it also sends HELLOS in response to the BEACONs received by *all* other interfaces attached to its router. This mode is necessary to permit "link-layer detection" of BIdirectional links composed of oppositely-directed, unidirectional connections between neighboring routers. Note that only by using this Hello mode can an interface determine that it itself has "OUTgoing" status without also having "INcoming" and, hence, BIdirectional status.

To make this clear, consider Figure 3.

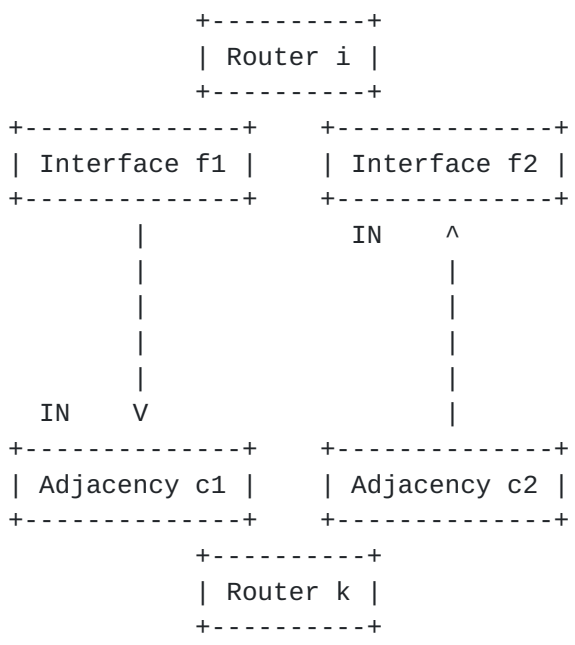


Figure 3: A bidirectional link consisting of two oppositely-directed connections.

Assume that SI Hello mode is being used, and the wireless directional connectivity is as shown. From router i's perspective, it can only receive over interface f2, and thus classifies connection (f2,c2) as IN. It is unaware that its BEACON packets being broadcast from interface f1 are being received at interface c1 on router k. However, if MI mode is used, then router k will advertise the reception of BEACON packers from f1 at c1 over connection (f2,c2) thereby informing router i that connection (f1,c1) should be classified as OUT. Of course, the reverse but same is true from router k s perspective.

The additional functionality provided by the MI mode comes at the cost of broadcasting a HELLO out *every* interface instead of only the interface over which the corresponding BEACON was received. This creates more HELLO overhead. For a given network, this cost must be balanced against the frequency of occurrence of the situation depicted in figure 3.

3.2 Control Message Aggregation

MANET routing protocols exchange control information in the form of routing control messages or "objects". To minimize the number of channel accesses generated by routing control traffic, the IMEP aggregates and encapsulates these objects into larger "IMEP object blocks". The objects are treated as "opaque" objects by the IMEP

protocol; i.e. IMEP is not aware of the contents of the objects, only of the protocol "type" of the object block (necessary for protocol demultiplexing at a receiver) and the length of each object. These object blocks are contained in yet larger "IMEP messages" which are passed to the IP layer for encapsulation and forwarding.

3.3 Broadcast Reliability

IMEP supports reliable and unreliable delivery of opaque protocol objects, where reliable delivery is also guaranteed to be delivery "in order" of transmission. IMEP uses a "point-to-multipoint selective repeat" algorithm to guarantee broadcast or multicast reliability and ordered delivery. This approach eliminates unnecessary retransmissions of the type commonly associated with "go back n" algorithms, and is in keeping with the greater IMEP goal of minimizing the number of required channel accesses.

To support reliability, each object block is given a SEQUENCE number, and is broadcast with that number and with a set of its intended receivers referred to as its "response list". When broadcast, a copy of the object block and its associated response list (i.e. the set of neighbors that are required to acknowledge this block) are stored. A retransmission timer is set to RETRANS_PERIOD (RP) time units which, upon expiration, will cause the object to be rebroadcast to any neighbors which have not acknowledged the object (this causes the retransmission timer to be set again to RP). The time the packet was initially broadcast is also stored. If the object s response list is not empty (i.e it has not been acknowledged by some adjacencies) after MAX_RETRANS_TIME (MRT) time units, the connections to those adjacencies are declared DOWN.

Acknowledgements (ACKs) are sent in response to object block receptions when (i) reliable delivery is indicated and (ii) when the receiver is contained in the response list. Once a node has ACKed a given block, it will be removed from the block's response list so that it will not be required to ACK any future retransmissions.

3.4 Network-level Address Resolution

IMEP puts forth a framework or architecture for MANET router and interface identification and addressing. IMEP operates simultaneously on two different topological levels: the "logical network" topology level---which is concerned with interrouter connectivity--- and the "physical" topology level---which is concerned with interface connectivity. Router IDs (RID) identify routers in the logical topology, and IP addresses identify interfaces in the physical topology. There may be *multiple* IP addresses associated with a given RID.

The purpose of the Network-level Address Resolution Protocol (NARP) incorporated within IMEP is to dynamically discover the mapping between RIDs and IP addresses when necessary. This is envisioned typically only to occur when a new connection is discovered, as it is necessary to be able to associate an interface (an IP address) with a router (an RID).

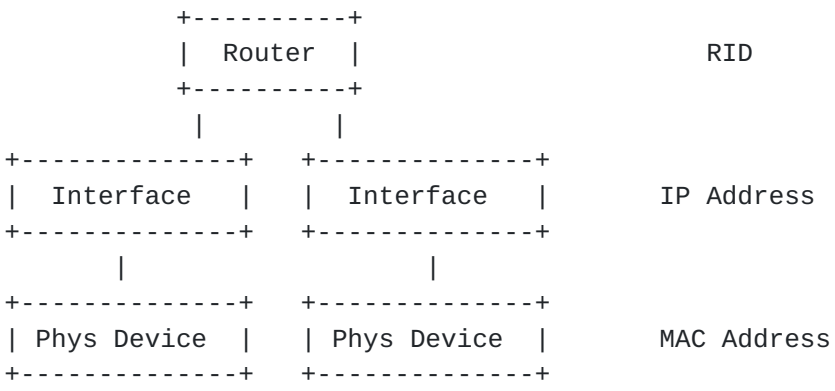


Figure 4: RIDs, IP and MAC addresses

While it is true that---as currently defined---RIDs are not "addresses" in the strict sense, they do uniquely identify a router for purposes of internal routing computations and somewhat resemble a logical "router address". Thus, the IP address-to-RID mapping is similar in spirit to IP address-to-MAC address mapping performed by the present ARP protocol. Each mapping simply associates an IP address with another identifier as shown in Figure 4. As with ARP, a "reverse" mapping is also defined as the Reverse Network-level Address Resolution Protocol (RNARP). The two mappings are shown in Figure 5.

ARP: IP --> MAC RARP: MAC --> IP

NARP: IP --> RID RNARP: RID --> IP

Figure 5: ARP/RARP and NARP/RNARP

When necessary, NARP/RNARP packets are generated, aggregated with other network control traffic and reliably broadcast within the object block of a IMEP message. However, unlike other control traffic, the NARP/RNARP objects are not opaque with respect to IMEP as they are generated and consumed by the IMEP protocol.

3.5 Security Authentication

It is expected that the IMEP protocol will include hooks for security

authentication in a fashion similar to that already performed by OSPF and other routing protocols. This will include, among others, an authentication type field in the IMEP message header.

3.6 BEACON and HELLO "Equivalency"

As stated earlier, BEACON and HELLO packets are necessary for ascertaining current connection status. From the perspective of a given router, BEACONs announce the presence of a broadcasting interface, and HELLOs simultaneously announce the presence of an adjacency and that the adjacency can receive from the broadcasting interface. However, it should be clear that the same information can be gleaned from other IMEP packets. Specifically, OBJect block transmissions (which may contain routing, NARP/RNARP and/or security objects) signal the presence of a broadcasting interface and are, in this sense, "equivalent" to BEACON packets. Similarly, ACKnowledgements both announce the presence of an adjacency and, through the process of acknowledgement, confirm that the adjacency recently received from the broadcasting interface. Thus, in this sense, ACKs are equivalent to HELLOs. The equivalency is depicted in the Figure 6.

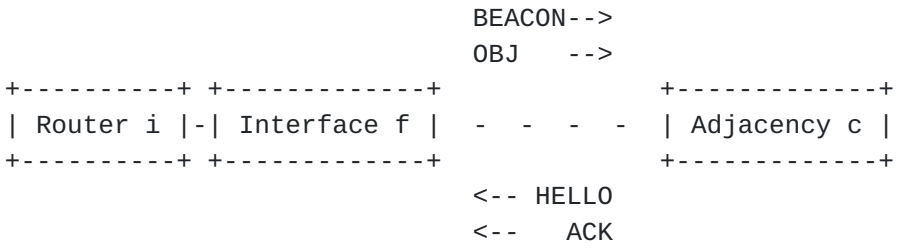


Figure 6: BEACON and HELLO Equivalency

Transmission or reception of a BEACON or HELLO equivalent packet affects the link status sensing timers as would transmission or reception of a BEACON or HELLO, respectively. Thus, during periods of heavy data, it is expected that BEACONs and HELLOs will rarely be transmitted as their respective "equivalent" packets will serve their role in link status sensing. During periods of light or no traffic, BEACONs or HELLOs will be transmitted as necessary to satisfy the aforementioned timing requirements.

3.7 Connection Failure Detection

It should be noted here that there are two events that can signal connection failure: expiration of the MRT timer or expiration of the BPT timer. Thus the CONN_DEAD_TIME (CDT) value---the time at which a connection, once UP (i.e. IN, OUT/BI), will be declared DOWN if its UP status is not confirmed---is CDT = min(MBT,MRT). Note that separate timers are used to monitor IN and OUT connection status. Thus, a connection may lose its OUT status while still retaining IN

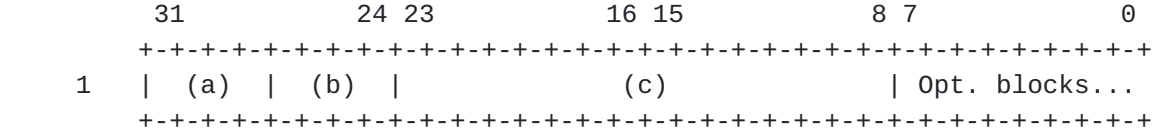
status and vice versa. Obviously, a connection satisfying both IN and OUT timing requirements is marked at BI.

4.0 Protocol Message Format

The following describes the message format of the proposed protocol. An IMEP message format consists of several fixed, mandatory fields followed by a self-formatting byte stream. The stream is aligned along "byte" boundaries---not 32-bit word boundaries---to save transmission overhead at the cost of extra processing at a router. An IMEP message typically contains at least one of several optional blocks. A message containing no blocks is a BEACON message.

```
<IMEP message> ::= <IMEP_VERSION> <BLOCK_FLAGS> <IMEP_LENGTH>
                   [Ack Block]
                   [Hello Block]
                   [Object Block]
```

The fixed field formats are:



(a) IMEP_VERSION (4-bit unsigned integer):
This field specifies the protocol version number, which may range from 0-15. The initial protocol version number is zero.

(b) BLOCK_FLAGS (4-bit bitmask):
This field contains single-bit flags, each of which specifies the presence (flag = 1) or absence (flag = 0) of a block. All bits set equal to zero indicates that this is a BEACON packet--a packet intended only to announce the existence of this interface (whose IP address is contained in the IP header) to any potential adjacencies.

bit 27: Ack block flag

bit 26: Hello block flag

bit 25: Object block flag

bit 24: unused

(c) IMEP_LENGTH (16-bit unsigned integer):
This field specifies the total length of an IMEP message

(measured in bytes) which must lie in the following range:

3 < IMEP_LENGTH <= MAX_IMEP_LENGTH <= 65535

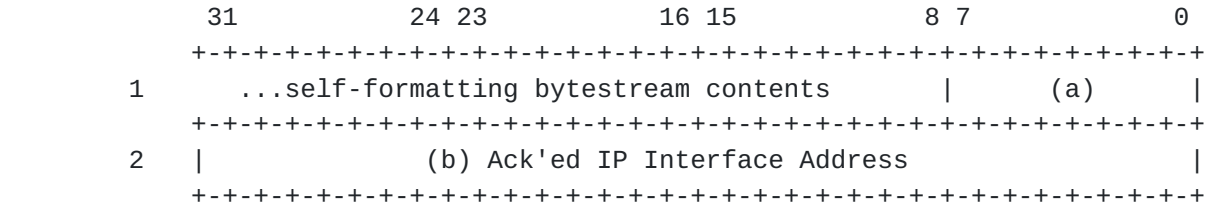
The ACK Block format is:

<Ack Block> ::= <NUM_ACKS> <Ack List>

<Ack List> ::= <ACK> |
<Ack List> <ACK>

NUM_ACKS (8-bit unsigned integer):
This field indicates the length of the Ack List which may range from 0-255.

The ACK format is:



(a) SEQUENCE (8-bit unsigned integer):
Indicates the sequence number of the protocol object block being ack'ed, which may range from 0-255.

(b) ACKIPADDR (32-bit unsigned integer):
IP interface address of the interface which originally sent the protocol object block that is being acknowledged.

The Hello Block format is:

<Hello Block> ::= <NUM_HELLOS> <Hello List>

<Hello List> ::= <HELLO> |
<Hello List> <HELLO>

NUM_HELLOS (8-bit unsigned integer):
This field indicates the length of the Hello List which may range from 0-255.

HELLO (32-bit unsigned integer):

This field contains the IPv4 address of the interface being "hello'ed".

The Object Block format consists of a set of fixed fields followed by an Object List and an optional Response List:

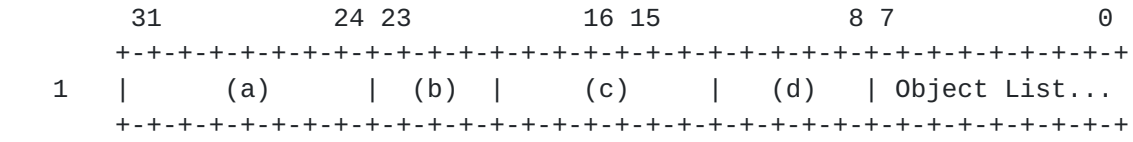
```
<Object Block> ::= <SEQUENCE> <PROTOCOL_TYPE>
                   <NUM_OBJECTS> <NUM_RESPONSES>
                   <Object List> [ <Response List> ]

<Object List> ::= <OBJECT> |
                  <Object List> <OBJECT>

<OBJECT> ::= <LENGTH_FLAG> <OBJECT_LENGTH> <OBJECT_DATA>

<Response List> ::= <RESPONSE> |
                    <Response List> <RESPONSE>
```

The fixed fields format is:



(a) SEQUENCE (8-bit unsigned integer):
A sequence counter uniquely indicating the Object Block ID within a sender's transmission queue, ranging from 0-255. This value may rollover, but for the envisioned applications, the 8-bit value should be large enough so that no two Object Blocks in the retransmission queue ever have the same SEQUENCE number.

(b) PROTOCOL_TYPE (4-bit unsigned integer):
This field indicates the protocol responsible for the opaque objects in the object block--necessary for demultiplexing the Object Block at a receiver. The field may range from 0-15.

value 0: reserved

value 1: NARP/RNARP

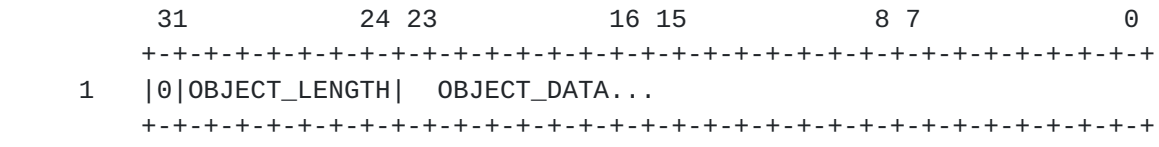
value 2: TORA

values 3-15: unassigned

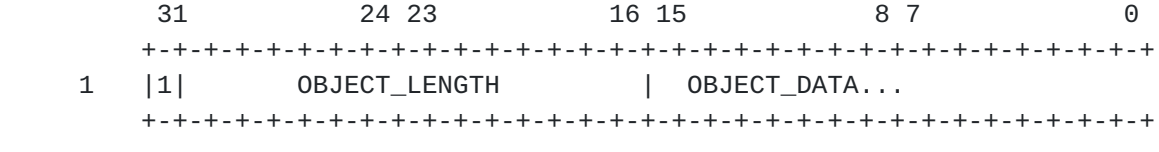
(c) NUM_OBJECTS (7-bit unsigned integer):
This field indicates the length (i.e. number of objects) of the Object List contained in the Object Block, which may range from 0-127.

(d) NUM_RESPONSES (5-bit unsigned integer):
This field indicates the length (i.e. number of responses) of the Response List contained in the Object Block, which may range from 0-31. The value 0 indicates unreliable delivery is desired, and that no interfaces need respond. The value 31 indicates BROADCAST, and that ALL receiving interfaces should respond. In both cases, the Response List is omitted.

The OBJECT format is:



or



LENGTH_FLAG (1-bit field, bit 31 in format figures):

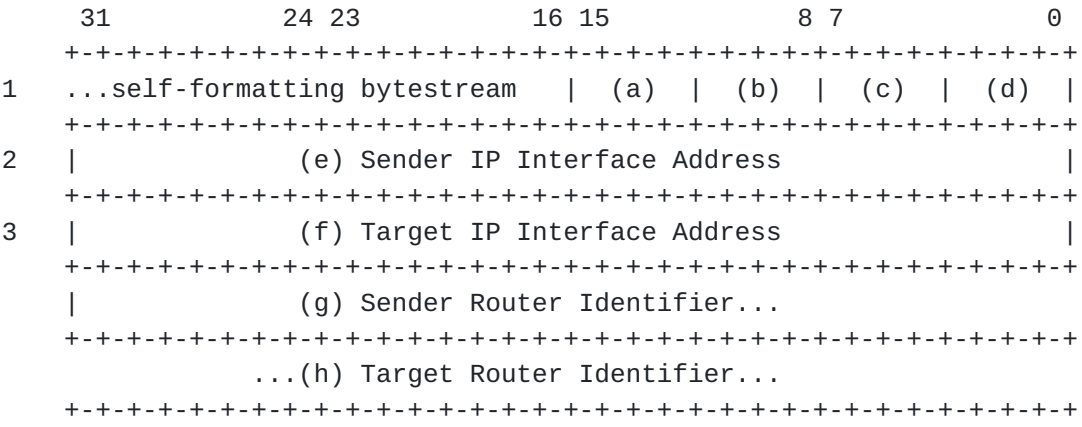
0: indicates 7-bit OBJECT_LENGTH

1: indicates 15-bit OBJECT_LENGTH

OBJECT_LENGTH (7 or 15-bit unsigned integer):
May range from 0-127 or 0-32767 as required indicating the length of the OBJECT_DATA in bytes.

OBJECT_DATA:
Opaque bytestream of data of length OBJECT_LENGTH.

NARP/RNARP Packet Format (> 10 bytes):



Both NARP and RNARP packets share the same format which is similar but not identical to traditional link-layer ARP/RARP packets:

(b) VERSION (4-bit unsigned integer):
This field specifies the version number of the NARP/RNARP protocol. The current version maps 32-bit IPv4 addresses. A future version would be required to map 128-bit IPv6 addresses.

0 (IPv4)

1 (IPv6)

2-15 (unused)

(b) PROTOCOL_TYPE (8-bit unsigned integer):
This field specifies the network-layer routing protocol type being mapped.

(c) PROTOCOL_LENGTH (4-bit unsigned integer):
This field specifies length of the protocol Router IDentifier (RID) in bytes.

(d) FRAME_TYPE (4-bit unsigned integer):
Indicates whether packet is a NARP or RNARP, and whether it is a request or reply.

0x00 (NARP Request)

0x01 (NARP Reply)

0x02 (RNARP Request)

0x03 (RNARP Reply)

(e) SENDER_INTERFACE_ADDR (32-bit unsigned integer):
IP interface address of the sender of the NARP/RNARP message.

(f) TARGET_INTERFACE_ADDR (32-bit unsigned integer):
IP interface address of the target of the NARP/RNARP message.

(g) SENDER_ROUTER_ADDR (length specfied in (c):
Router identifier of the sender of the NARP/RNARP message.

(h) TARGET_ROUTER_ADDR (length specfied in (c):
Router identifier of the target of the NARP/RNARP message.

5. Summary

The preceding gives only a high-level protocol description, specifying what is to be exchanged and, generally, why. Details on how the protocol is to be implemented will be given in a subsequent version of this draft.

6. References

[1] C. Perkins, "Mobile Ad Hoc Networking Terminology," draft-ietf-manet-term-00.txt, October 1997.

Authors' Addresses:

M. Scott Corson
Institute for Systems Research
A.V. Williams Building (115)
University of Maryland
College Park, MD 20742
(301) 405-6630
corson@isr.umd.edu

Vincent Park
Information Technology Division
Code 5540
Naval Research Laboratory
Washington, DC 20375
(202) 767-5098
vpark@itd.nrl.navy.mil