                  Cabletron's SecureFast VLAN Operational Model
                                 Version 1.8

Status of this Memo

Copyright Notice

Abstract

   Cabletron's SecureFast VLAN (SFVLAN) product implements a distributed
   connection-oriented switching protocol that provides fast forwarding
   of data packets at the MAC layer. The product uses the concept of
   virtual LANs (VLANs) to determine the validity of call connection
   requests and to scope the broadcast of certain flooded messages.

Table of Contents

1. Introduction

   This memo is being distributed to members of the Internet community
   in order to solicit reactions to the proposals contained herein.
   While the specification discussed here may not be directly relevant
   to the research problems of the Internet, it may be of interest to
   researchers and implementers.

1.1 Data Conventions

   The methods used in this memo to describe and picture data adhere to
   the standards of Internet Protocol documentation [RFC1700]. In
   particular:

      The convention in the documentation of Internet Protocols is to
      express numbers in decimal and to picture data in "big-endian"
      order. That is, fields are described left to right, with the most
      significant octet on the left and the least significant octet on
      the right.

      The order of transmission of the header and data described in this
      document is resolved to the octet level. Whenever a diagram shows
      a group of octets, the order of transmission of those octets is
      the normal order in which they are read in English.

      Whenever an octet represents a numeric quantity the left most bit
      in the diagram is the high order or most significant bit. That
      is, the bit labeled 0 is the most significant bit.

Similarly, whenever a multi-octet field represents a numeric
quantity the left most bit of the whole field is the most
significant bit. When a multi-octet quantity is transmitted the
most significant octet is transmitted first.

1.2 Definitions of Commonly Used Terms

This section contains a collection of definitions for terms that have
a specific meaning for the SFVLAN product and that are used
throughout the text.

Switch ID

A 10-octet value that uniquely identifies an SFVLAN switch within
the switch fabric. The value consists of the 6-octet base MAC
address of the switch, followed by 4 octets of zeroes.

Network link

The physical connection between two switches. A network link is
associated with a network interface (or port) of a switch.

Network port

An interface on a switch that attaches to another switch.

Access port

An interface on a switch that attaches to a user endstation.

Port ID

A 10-octet value that uniquely identifies an interface of a
switch. The value consists of the 6-octet base MAC address of the
switch, followed by the 4-octet local port number of the
interface.

Neighboring switches

Two switches attached to a common (network) link.

Call connection

A mapping of user traffic through a switch that correlates the
source and destination address pair specified within the packet to
an inport and outport pair on the switch.

   Call connection path

      A set of 0 to 7 network links over which user traffic travels
      between the source and destination endstations. Call connection
      paths are selected from a list of alternate equal cost paths
      calculated by the VLS protocol [IDvlsp], and are chosen to load
      balance traffic across the fabric.

   Ingress switch

      The owner switch of the source endstation of a call connection.
      That is, the source endstation is attached to one of the local
      access ports of the switch.

   Egress switch

      The owner switch of the destination endstation of a call
      connection. That is, the destination endstation is attached to
      one of the local access ports of the switch.

   Intermediate switches

      Any switch along the call connection path on which user traffic
      enters and leaves over network links. Note that the following
      types of connections have no intermediate switches:

      - Call connections between source and destination endstations
         that are attached to the same switch -- that is, the ingress
         switch is the same as the egress switch. Note also that the
         path for this type of connection consists of 0 network links.

      - Call connections where the ingress and egress switches are
         physical neighbors connected by a single network link. The
         path for this type of connection consists of a single network
         link.

   InterSwitch Message protocol (ISMP)

      The protocol used for interswitch communication between SFVLAN
      switches.

   Undirected messages

      Messages that are (potentially) sent to all SFVLAN switches in the
      switch fabric -- that is, they are not directed to any particular
      switch. ISMP messages with a message type of 5, 7 or 8 are
      undirected messages.

Switch flood path

The path used to send undirected messages throughout the switch
fabric. The switch flood path is formed using a spanning tree
algorithm that provides a single path through the switch fabric
that guarantees loop-free delivery to every other SFVLAN switch in
the fabric.

Upstream Neighbor

That switch attached to the inport of the switch flood path --
that is, the switch from which undirected messages are received.
Note that each switch receiving an undirected message has, at
most, one upstream neighbor, and the originator of any undirected
ISMP message has no upstream neighbors.

Downstream Neighbors

Those switches attached to all outports of the switch flood path
except the port on which the undirected message was received.
Note that for each undirected message some number of switches have
no downstream neighbors.

Virtual LAN (VLAN) identifier

A VLAN is a logical grouping of ports and endstations such that
all ports and endstations in the VLAN appear to be on the same
physical (or extended) LAN segment even though they may be
geographically separated.

A VLAN identifier consists of a variable-length string of octets.
The first octet in the string contains the number of octets in the
remainder of the string -- the actual VLAN identifier value. A
VLAN identifier can be from 1 to 16 octets long.

VLAN policy

Each VLAN has an assigned policy value used to determine whether a
particular call connection can be established. SFVLAN recognizes
two policy values: Open and Secure.

2. SFVLAN Overview

Cabletron's SecureFast VLAN (SFVLAN) product implements a distributed
connection-oriented switching protocol that provides fast forwarding
of data packets at the MAC layer.

2.1 Features

   Within a connection-oriented switching network, user traffic is
   routed through the switch fabric based on the source and destination
   address (SA/DA) pair found in the arriving packet. For each SA/DA
   pair encountered by a switch, a "connection" is programmed into the
   switch hardware. This connection maps the SA/DA pair and the port on
   which the packet was received to a specific outport over which the
   packet is to be forwarded. Thus, once a connection has been
   established, all packets with a particular SA/DA pair arriving on a
   particular inport are automatically forwarded by the switch hardware
   out the specified outport.

   A distributed switching environment requires that each switch be
   capable of processing all aspects of the call processing and
   switching functionality. Thus, each switch must synchronize its
   various databases with all other switches in the fabric or be capable
   of querying other switches for information it does not have locally.

   SFVLAN accomplishes the above objectives by providing the following
   features:

   - A virtual directory of the entire switch fabric.

   - Call processing for IP, IPX and MAC protocols.

   - Automatic call connection, based on VLAN policy.

   - Automatic call rerouting around failed switches and links.

   In addition, SFVLAN optimizes traffic flow across the switch fabric
   by providing the following features:

   - Broadcast interception and address resolution at the ingress port.

   - Broadcast scoping, restricting the flooding of broadcast packets
      to only those ports that belong to the same VLAN as the packet
      source.

   - A single loop-free path (spanning tree) used for the flooding of
      undirected interswitch control messages. Only switches running
      the SFVLAN switching protocol are included in this spanning tree
      calculation -- that is, traditional bridges or routers configured
      for bridging are not included.

   - Interception of both service and route advertisements with
      readvertisement sourced from the MAC address of the original
      advertiser.

2.2 VLAN Principles

   Each SFVLAN switch port, along with its attached endstations, belongs
   to one or more virtual LANs (VLANs). A VLAN is a logical grouping of
   ports and endstations such that all ports and endstations in the VLAN
   appear to be on the same physical (or extended) LAN segment even
   though they may be geographically separated.

   VLAN assignments are used to determine the validity of call
   connection requests and to scope the broadcast of certain flooded
   messages.

2.2.1 Default, Base and Inherited VLANs

   Each port is explicitly assigned to a default VLAN. At start-up, the
   default VLAN to which all ports are assigned is the base VLAN -- a
   permanent, non-deletable VLAN to which all ports belong at all times.

   The network administrator can change the default VLAN of a port from
   the base VLAN to any other unique VLAN by using a management
   application known here as the VLAN Manager. A port's default VLAN is
   persistent -- that is, it is preserved across a switch reset.

   When an endstation attaches to a port for the first time, it inherits
   the default VLAN of the port. Using the VLAN Manager, the network
   administrator can reassign an endstation to another VLAN.

      Note:

         When all ports and all endstations belong to the base VLAN, the
         switch fabric behaves like an 802.1D bridging system.

2.2.2 VLAN Configuration Modes

   For both ports and endstations, there are a variety of VLAN
   configuration types, or modes.

2.2.2.1 Endstations

   For endstations, there are two VLAN configuration modes: inherited
   and static.

   - Inherited

      An inherited endstation becomes a member of its port's default
      VLAN.

- Static

    A static port becomes a member of the VLAN to which it has been
    assigned by the VLAN Manager.

  The default configuration mode for an endstation is inherited.

2.2.2.2 Ports

  For ports, there are two VLAN configuration modes: normal and
  locked.

  - Normal

    All inherited endstations on a normal port become members of the
    port's default VLAN. All static endstations are members of the
    VLAN to which they were mapped by the VLAN Manager.

    If the VLAN Manager reassigns the default VLAN of a normal port,
    the VLAN(s) for the attached endstations may or may not change,
    depending on the VLAN configuration mode of each endstation. All
    inherited endstations will become members of the new default VLAN.
    All others will retain membership in their previously mapped
    VLANs.

  - Locked

    All endstations attached to a locked port can be members only of
    the port's default VLAN.

    If the VLAN Manager reconfigures a normal port to be a locked
    port, all endstations attached to the port become members of the
    port's default VLAN, regardless of any previous VLAN membership.

  The default configuration mode for ports is normal.

2.2.2.3 Order of Precedence

  On a normal port, static VLAN membership prevails over inherited
  membership.

  On a locked port, default VLAN membership prevails over any static
  VLAN membership.

  If a statically assigned endstation moves from a locked port back to
  a normal port, the endstation's static VLAN membership must be
  preserved.

2.2.3 Ports with Multiple VLAN Membership

   A port can belong to multiple VLANs, based on the VLAN membership of
   its attached endstations.

   For example, consider a port with three endstations, a default VLAN
   of "blue" and the following endstation VLAN assignments:

   - One of the endstations is statically assigned to VLAN "red."
   - Another endstation is statically assigned to VLAN "green."
   - The third endstation inherits the default VLAN of "blue."

   In this instance, the port is explicitly a member of VLAN "blue." But
   note that it is also implicitly a member of VLAN "red" and VLAN
   "green." Any tag-based flooding (Section 4.8) directed to any one of
   the three VLANs ("red," "green," or "blue") will be forwarded out the
   port.

2.3 Tag/Length/Value Method of Addressing

   Within most computer networks, the concept of "address" is somewhat
   elusive because different protocols can (and do) use different
   addressing schemes and formats. For example, Ethernet (physical
   layer) addresses are six octets long, while IP (network layer)
   addresses are only four octets long.

   To distinguish between the various protocol-specific forms of
   addressing, many software modules within the SFVLAN product specify
   addresses in a format known as Tag/Length/Value (TLV). This format
   uses a variable-length construct as shown below:

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                             Tag                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Value length  |                                               |
   +-+-+-+-+-+-+-+-+-+                                              +
   |                         Address value                         |
   :                                                               :
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
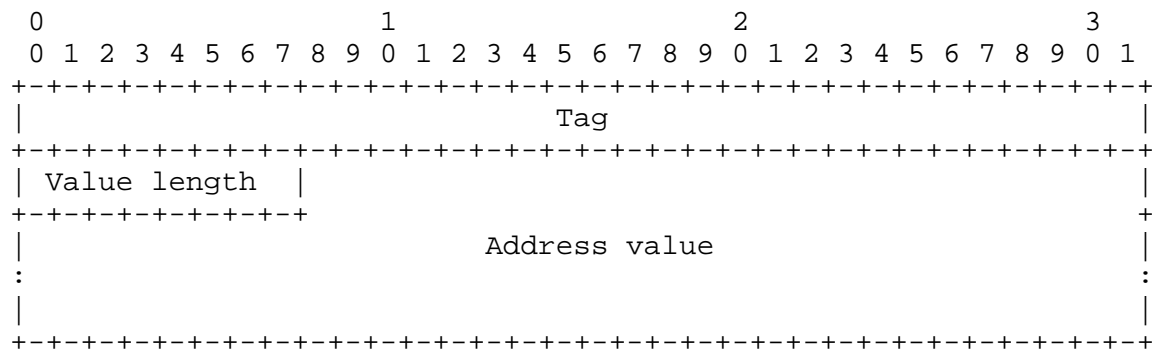
   Tag

      This 4-octet field specifies the type of address contained in the
      structure. The following address types are currently supported:

```
        Tag name          Value      Address type

        aoMacDx           1          DX ethernet dst/src/type
        aoIpxSap          2          Sap
        aoIpxRIP          3          RIP
        aoInstYP          4          YP (YP name and version)
        aoInstUDP         5          UDP (Port #)
        aoIpxIpx          6          Ipx
        aoInetIP          7          IP (Net address)
        aoInetRPC         8          RPC (Program #)
        aoInetRIP         9          INET RIP
        aoMacDXMcast      10         Multicast unknown type
        aoAtDDP           11         AppleTalk DDP
        aoEmpty           12         (no address type specified)
        aoVlan            13         VLAN identifier
        aoHostName        14         Host name
        aoNetBiosName     15         NetBIOS name
        aoNBT             16         NetBIOS on TCP name
        aoInetIPMask      17         IP Subnet Mask
        aoIpxSap8022      18         Sap 8022 type service
        aoIpxSapSnap      19         Sap Snap type service
        aoIpxSapEnet      20         Sap Enet type service
        aoDHCPXID         21         DHCP Transaction ID
        aoIpMcastRx       22         IP class D receiver
        aoIpMcastTx       23         IP class D sender
        aoIpxRip8022      24         Ipx Rip 8022 type service
        aoIpxRipSnap      25         Ipx Rip type service
        aoIpxRipEnet      26         Ipx Rip Enet service
        aoATM             27         ATM
        aoATMELAN         28         ATM LAN Emulation Name
```

   Value length

      This 1-octet field contains the length of the value of the
      address. The value here depends on the address type and actual
      value.

   Address value

      This variable-length field contains the value of the address. The
      length of this field is stored in the Value length field.

2.4 Architectural Overview

   The SFVLAN software executes in the switch CPU and consists of the
   following elements as shown in Figure 1:

   - The SFVLAN base services that handles traffic intercepted by the
      switch hardware. The base services are described in Section 3.

```
      +--------------------------------------------------------+
      |                                      +-----+ |
      |                        +-----------+ |  I  | |
      |                        | CALL TAP  <--(8)--> N  | |
      |                        +-----------+ |  T  | |
      |                                      |  E  | |
      |    +-----------+       +-----------+ |  R  | |
      |    |   PATH    |       | TOPOLOGY  | |  S  | |
      |    |           |       |           | |  W  | |
      |    | Lnk state <------> Lnk state <--(3)--> I  | | Flood path
      |    |           |       |           | |  T  <----(5,7,8)-->
      |    | Span tree <------> Span tree <--(4)--> C  | |
      |    +--^--------+       |           | |  H  | |
      |       |                | Discovery <--(2)--> |  | |
      |       |                +-----------+ |  M  | |
      |       |                              |  E  | |
      |    +------^--+         +--------+    |  S  | |
      |    | CONNECT >---------+--> FILTER |  |  S  | |
      |    +--^------+         |  +--------+    |  A  | | specific
      |       |               |                |  G  | | netwrk lnks
      |       |       +--------^-+    +-------+ |  E  <----(2,3,4)-->
      |    +-------<  POLICY |    | FLOOD >--(7)--> |  | |
      |    +------^---+    +-^-----+ |  P  | |
      |             |               |         |  R  | |
      |  +-----------+   +-^-----------V-+    |  O  | |
      |  | DIRECTORY <----> RESOLVE    <------(5)--> T  | |
      |  +-----^-----+   +---^----------+    |  O  | |
      |        |             |               |  C  | |
      |        | +---------^----------+    |  O  | |
      |        +----< Base Services  |    |  L  | |
      |          +------^-------------+    +-----+ |
      |                  |                          |
      +------------------|--------------------------------+
       Switch CPU        |
                         | Host control port
             +-----O----------------+
             |     ^ no cnx         |
      Layer 2 |    |                |
      ---------->O-----+-------------->O----------->
      SA/DA pr  |       known cnx    |
             +---------------------+
                 Switch hardware
```
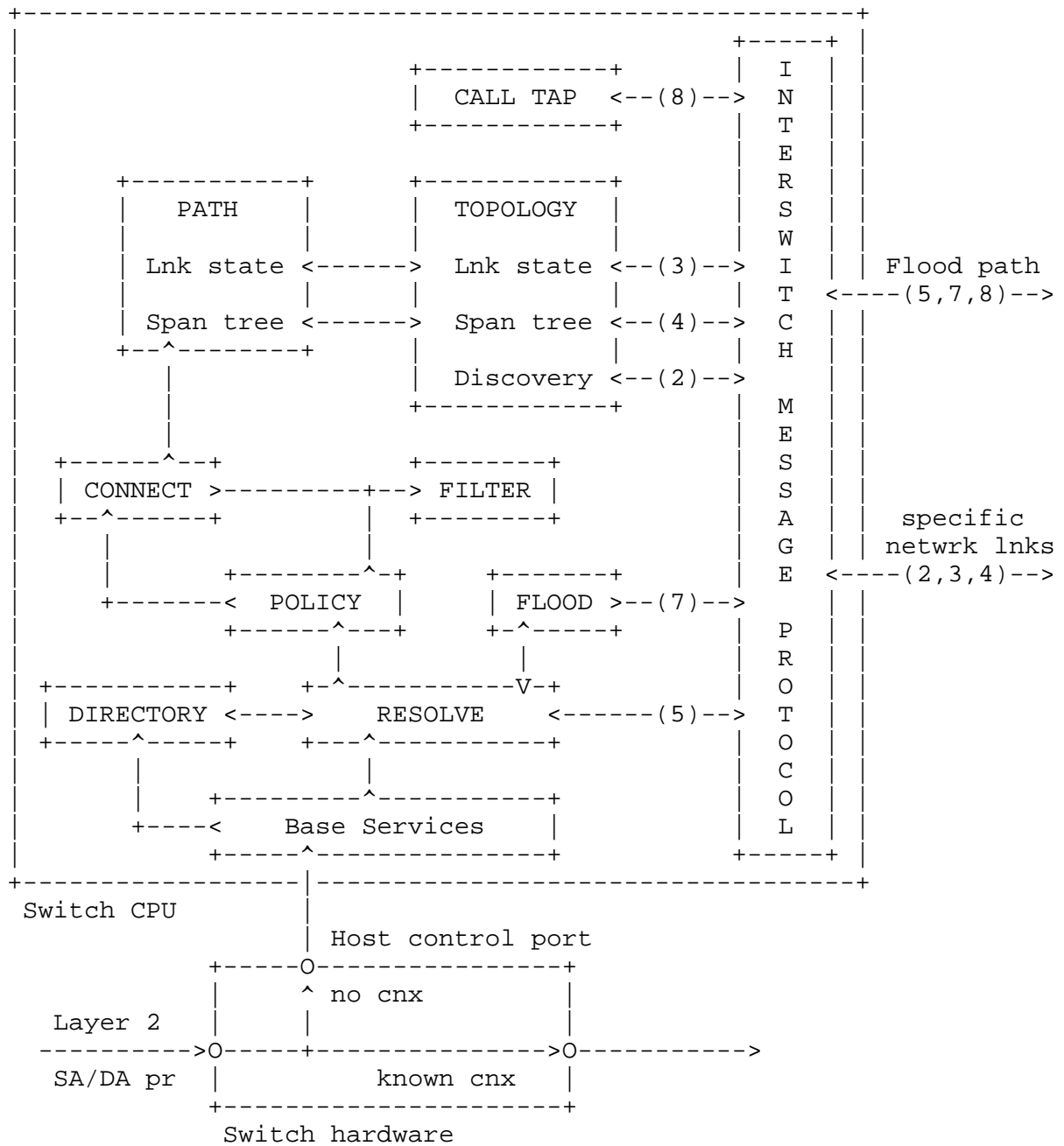
                   Figure 1: SFVLAN Architectural Overview

-   Eight call processing service centers that provide the essential
    services required to process call connections. The call
    processing service centers are described in Section 4.

-   A Call Tap module that supports the monitoring of call
    connections. The Call Tap module is described in Section 5.

-   The InterSwitch Message Protocol (ISMP) that provides a consistent
    method of encapsulating and transmitting control messages
    exchanged between SFVLAN switches. (Note that ISMP is not a
    discrete software module. Instead, its functionality is
    distributed among those service centers and software modules that
    need to communicate with other switches in the fabric.) The
    Interswitch Message Protocol and the formats of the individual
    interswitch messages are described in Section 6.

3. Base Services

   The SFVLAN base services act as the interface between the switch
   hardware and the SFVLAN service centers running on the switch CPU.
   This relationship is shown in Figure 2. This figure is a replication
   of the bottom portion of Figure 1.

```
         |      Directory        Resolve                       |
         |          ^               ^                          |
         |          |               |                          |
         |          |     +---------^-----------+              |
         |        +----<    Base Services    |              |
         |          +-----^--------------+              |
         +-------------------|-------------------------+
          Switch CPU         |
                             | Host control port
                   +-----O---------------+
                   |     ^ no cnx        |
          Layer 2  |     |               |
          ---------->O-----+-------------->O----------->
          SA/DA pr  |          known cnx   |
                   +---------------------+
                     Switch hardware
```
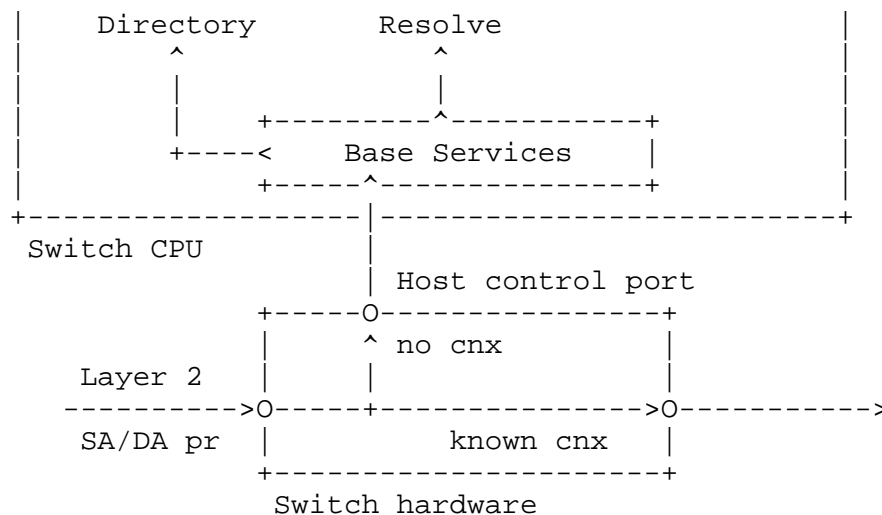
                   Figure 2: Base Services

During normal operation of the switch, data packets arriving at
any one of the local switch ports are examined in the switch
hardware. If the packet's source and destination address (SA/DA)
pair match a known connection, the hardware simply forwards the
packet out the outport specified by the connection.

If the SA/DA pair do not match any known connection, the hardware
diverts the packet to the host control port where it is picked up
by the SFVLAN base services. The base services generate a
structure known as a state box that tracks the progress of the
call connection request as the request moves through the call
processing service centers.

After creating the call's state box, the base services check to
determine if the call is a duplicate of a call already being
processed. If not, a request is issued to the Directory Service
Center (Section 4.1) to add the call's source address to the local
Node and Alias Tables. The base services then hand the call off to
the Resolve Service Center (Section 4.3) for further processing.

4. Call Processing

Call connection processing is handled by a set of eight service
centers, each with one or more servers. The servers within a
service center are called in a particular sequence. Each server
records the results of its processing in the call connection
request state box and passes the state box to the next server in
the sequence.

In the sections that follow, servers are listed in the order in
which they are called.

4.1 Directory Service Center

The Directory Service Center is responsible for cataloging the MAC
addresses and alias information for both local and remote
endstations. The information is stored in two tables -- the Node
Table and the Alias Table.

-   The Node Table contains the MAC addresses of endstations
    attached to the local switch. It also contains a cache of
    remote endstations detected by the Resolve Service Center
    (Section 4.3).   Every entry in the Node Table has one or more
    corresponding entries in the Alias Table.

- The Alias Table contains protocol alias information for each
  endstation. An endstation alias can be a network address (such
  as an IP or IPX address), a VLAN identifier, or any other
  protocol identifier. Since every endstation is a member of at
  least one VLAN (the default VLAN for the port), there is always
  at least one entry in the Alias Table for each entry in the
  Node Table.

  Note:

     The Node and Alias Tables must remain synchronized.
     That is, when an endstation's final alias is removed
     from the Alias Table, the endstation entry is removed
     from the Node Table.

Note that the total collection of all Node Tables and Alias Tables
across all switches is known as the "virtual" directory of the
switch fabric. The virtual directory contains address mappings of
all known endstations in the fabric.

4.1.1 Local Add Server

The Directory Local Add server adds entries to the local Node or
Alias Tables. It is called by the base services (Section 3) to
add a local endstation and by the Interswitch Resolve (Section
4.3.4) server to add an endstation discovered on a remote switch.

4.1.2 Inverse Resolve Server

The Directory Inverse Resolve server is invoked when a new
endstation has been discovered on the local switch (that is, when
the Local Add server was successful in adding the endstation).
The server provides two functions:

- It populates the Node and Alias Tables with local entries
  during switch initialization.

- It processes a new endstation discovered after the fabric
  topology has converged to a stable state.

In both instances, the processing is identical.

When a new endstation is detected on one of the switch's local ports, the Inverse Resolve server sends an Interswitch New User request message (Section 6.5) over the switch flood path to all other switches in the fabric. The purpose of the Interswitch New User request is two-fold:

- It informs the other switches of the new endstation address. Any entries for that endstation in the local databases of other switches should be dealt with appropriately.

- It requests information about any static VLAN(s) to which the endstation has been assigned.

When a switch receives an Interswitch New User request message from one of its upstream neighbors, it first forwards the message to all its downstream neighbors. No actual processing or VLAN resolution is attempted until the message reaches the end of the switch flood path and begins its trip back along the return path. This ensures that all switches in the fabric receive notification of the new user and have synchronized their databases.

If a switch receives an Interswitch New User request message but has no downstream neighbors, it does the following:

- If the endstation was previously connected to one of the switch's local ports, the switch formulates an Interswitch New User Response message by loading the VLAN identifier(s) of the static VLAN(s) to which the endstation was assigned, along with its own MAC address. (VLAN identifiers are stored in Tag/Length/Value (TLV) format. See Section 2.3.) The switch then sets the message status field to NewUserAck, and returns the message to its upstream (requesting) neighbor.

   Otherwise, the switch sets the status field to NewUserUnknown and returns the message to its upstream neighbor.

- The switch then deletes the endstation from its local database, as well as any entries associated with the endstation in its connection table.

When a switch forwards an Interswitch New User request message to its downstream neighbors, it keeps track of the number of requests it has sent out and does not respond back to its upstream neighbor until all requests have been responded to.

-   As each response is received, the switch checks the status
    field of the message. If the status is NewUserAck, the switch
    retains the information in that response. When all requests
    have been responded to, the switch returns the NewUserAck
    response to its upstream neighbor.

-   If all the Interswitch New User Request messages have been
    responded to with a status of NewUserUnknown, the switch checks
    to see if the endstation was previously connected to one of its
    local ports. If so, the switch formulates an Interswitch New
    User Response message by loading the VLAN identifier(s) of the
    static VLAN(s) to which the endstation was assigned, along with
    its own MAC address. The switch then sets the message status
    field to NewUserAck, and returns the message to its upstream
    (requesting) neighbor.

    Otherwise, the switch sets the status field to NewUserUnknown
    and returns the message to its upstream neighbor.

-   The switch then deletes the endstation from its local database,
    as well as any entries associated with the endstation in its
    connection table.

When the originating switch has received responses to all the
Interswitch New User Request messages it has sent, it does the
following:

-   If it has received a response message with a status of
    NewUserAck, it loads the new VLAN information into its local
    database.

-   If all responses have been received with a status of
    NewUserUnknown, the originating switch assumes that the
    endstation was not previously connected anywhere in the network
    and assigns it to a VLAN according to the VLAN membership rules
    and order of precedence.

If any Interswitch New User Request message has not been responded
to within a certain predetermined time (currently 5 seconds), the
originating switch recalculates the switch flood path and resends
the Interswitch New User Request message.

4.1.3 Local Delete Server

   The Directory Local Delete server removes entries (both local and
   remote) from the local Node and Alias Tables. It is invoked when
   an endstation, previously known to be attached to one switch, has
   been moved and discovered on another switch.

   Note also that remote entries are cached and are purged from the
   tables on a first-in/first-out basis as space is needed in the
   cache.

4.2 Topology Service Center

   The Topology Service Center is responsible for maintaining three
   databases relating to the topology of the switch fabric:

   - The topology table of SFVLAN switches that are physical
      neighbors to the local switch.

   - The spanning tree that defines the loop-free switch flood path
      used for transmitting undirected interswitch messages.

   - The directed graph that is used to calculate the best path(s)
      for call connections.

4.2.1 Neighbor Discovery Server

   The Topology Neighbor Discovery server uses Interswitch Keepalive
   messages to detect the switch's neighbors and establish the
   topology of the switching fabric. Interswitch Keepalive messages
   are exchanged in accordance with Cabletron's VlanHello protocol,
   described in detail in [IDhello].

4.2.2 Spanning Tree Server

   The Topology Spanning Tree server is invoked by the Topology
   Neighbor Discovery server when a neighboring SFVLAN switch is
   either discovered or lost -- that is, when the operational status
   of a network link changes.

   The Spanning Tree server exchanges interswitch messages with
   neighboring SFVLAN switches to calculate the switch flood path
   over which undirected interswitch messages are sent. There are
   two parts to this process:

   - Creating and maintaining the spanning tree
   - Remote blocking

4.2.2.1 Creating and Maintaining the Spanning Tree

   In a network with redundant network links, a packet traveling between
   switches can potentially be caught in an infinite loop -- an
   intolerable situation in a networking environment. However, it is
   possible to reduce a network topology to a single configuration
   (known as a spanning tree) such that there is, at most, one path
   between any two switches.

   Within the SFVLAN product, the spanning tree is created and
   maintained using the Spanning Tree Algorithm defined by the IEEE
   802.1d standard.

      Note:

         A detailed discussion of this algorithm is beyond the scope of
         this document. See [IEEE] for more information.

   To implement the Spanning Tree Algorithm, SFVLAN switches exchange
   Interswitch BPDU messages (Section 6.2) containing encapsulated
   IEEE-compliant 802.2 Bridge Protocol Data Units (BPDUs). There are
   two types of BPDUs:

   - Configuration (CFG) BPDUs are exchanged during the switch
      discovery process, following the receipt of an Interswitch
      Keepalive message. They are used to create the initial the
      spanning tree.

   - Topology Change Notification (TCN) BPDUs are exchanged when
      changes in the network topology are detected. They are used to
      redefine the spanning tree to reflect the current topology.

   See [IEEE] for detailed descriptions of these BPDUs.

4.2.2.2 Remote Blocking

   After the spanning tree has been computed, each network port on an
   SFVLAN switch will be in one of two states:

   - Forwarding. A port in the Forwarding state will be used to
      transmit all ISMP messages.

-   Blocking. A port in the Blocking state will not be used to
    forward undirected ISMP messages. Blocking the rebroadcast of
    these messages on selected ports prevents message duplication
    arising from multiple paths that exist in the network topology.
    Note that all other types of ISMP message will be transmitted.

    Note:

        The IEEE 802.1d standard specifies other port states used
        during the initial creation of the spanning tree. These states
        are not relevant to the discussion here.

Note that although a port in the Blocking state will not forward
undirected ISMP messages, it may still receive them. Any such
message received will ultimately be discarded, but at the cost of CPU
time necessary to process the packet.

To prevent the transmission of undirected messages to a port, the
port's owner switch can set remote blocking on the link by sending an
Interswitch Remote Blocking message (Section 6.3) out over the port.
This notifies the switch on the other end of the link that undirected
messages should not be sent over the link, regardless of the state of
the sending port.

Each SFVLAN switch sends an Interswitch Remote Blocking message out
over all its blocked network ports every 5 seconds. A flag within
the message indicates whether remote blocking should be turned on or
off over the link.

4.2.3 Link State Server

The Topology Link State server is invoked by any process that detects
a change in the state of the network links of the local switch.
These changes include (but are not limited to) changes in operational
or administrative status of the link, path "cost" or bandwidth.

The Link State server runs Cabletron's Virtual LAN Link State (VLS)
protocol which exchanges interswitch messages with neighboring SFVLAN
switches to calculate the set of best paths between the local switch
and all other switches in the fabric. (The VLS protocol is described
in detail in [IDvlsp].)

The Link State server also notifies the Connect Service Center
(Section 4.5) of any remote links that have failed, thereby
necessitating potential tear-down of current connections.

4.3 Resolve Service Center

   The Resolve Service Center is responsible for resolving the
   destination address of broadcast data packets (such as an IP ARP
   packet) to a unicast MAC address to be used in mapping the call
   connection. To do this, the Resolve Service Center attempts to
   resolve such broadcast packets directly at the access port of the
   ingress switch.

   Address resolution is accomplished as follows:

   1) First, an attempt is made to resolve the address from the switch's
      local databases by calling the following servers:

      - The Table server attempts to resolve the address from the
        Resolve Table (Section 4.3.1).

      - Next, the Local server attempts to resolve the address from the
        Node and Alias Tables (Section 4.3.2).

      - If the address is not found in these tables but is an IP
        address, the Resolve Subnet server (Section 4.3.3) is also
        called.

   2) If the address cannot be resolved locally, the Interswitch Resolve
      server (Section 4.3.4) is called to access the "virtual directory"
      by sending an Interswitch Resolve request message out over the
      switch flood path.

   3) If the address cannot be resolved either locally or via an
      Interswitch Resolve message -- that is, the destination endstation
      is unknown to any switch, perhaps because it has never transmitted
      a packet to its switch -- the following steps are taken:

      - The Unresolvable server (Section 4.3.5) is called to record the
        unresolved packet.

      - The Block server (Section 4.3.6) is called to determine whether
        the address should be added to the Block Table.

      - The Flood Service Center (Section 4.8) is called to broadcast
        the packet to other SFVLAN switches using a tag-based flooding
        mechanism.

4.3.1 Table Server

   The Resolve Table server maintains the Resolve Table which contains a
   collection of addresses that might not be resolvable in the normal
   fashion. This table typically contains such things as the addresses
   of "quiet" devices that do not send data packets or special mappings
   of IP addresses behind a router. Entries can be added to or deleted
   from the Resolve Table via an external management application.

4.3.2 Local Server

   The Resolve Local server checks the Node and Alias Tables maintained
   by the Directory Service Center (Section 4.1) to determine if it can
   resolve the address.

4.3.3 Subnet Server

   If the address to be resolved is an IP address but cannot be resolved
   via the standard processing described above, the Resolve Subnet
   server applies the subnet mask to the IP address and then does a
   lookup in the Resolve Table.

4.3.4 Interswitch Resolve Server

   If the address cannot be resolved locally, the Interswitch Resolve
   server accesses the "virtual directory" by sending an Interswitch
   Resolve request message (Section 6.4) out over the switch flood path.
   The Interswitch Resolve request message contains the destination
   address as it was received within the packet, along with a list of
   requested addressing information.

   When a switch receives an Interswitch Resolve request message from
   one of its upstream neighbors, it checks to see if the destination
   endstation is connected to one of its local access ports. If so, it
   formulates an Interswitch Resolve response message by filling in the
   requested address information, along with its own MAC address. It
   then sets the message status field to ResolveAck, and returns the
   message to its upstream (requesting) neighbor.

   If the receiving switch cannot resolve the address, it forwards the
   Interswitch Resolve request message to its downstream neighbors. If
   the switch has no downstream neighbors, it sets the message status
   field to Unknown, and returns the message to its upstream
   (requesting) neighbor.

When a switch forwards an Interswitch Resolve request message to its
downstream neighbors, it keeps track of the number of requests it has
sent out and received back. It will only respond back to its
upstream (requesting) neighbor when one of the following conditions
occurs:

- It receives any response with a status of ResolveAck

- All downstream neighbors have responded with a status of Unknown

Any Interswitch Resolve request message that is not responded to
within a certain predetermined time (currently 5 seconds) is assumed
to have a response status of Unknown.

When the Interswitch Resolve server receives a successful Interswitch
Resolve response message, it records the resolved address information
in the remote cache of its local directory for use in resolving later
packets for the same endstation. Note that this process results in
each switch building its own unique copy of the virtual directory
containing only the endstation addresses in which it is interested.

4.3.5 Unresolvable Server

The Unresolvable server is called when a packet destination address
cannot be resolved. The server records the packet in a table that
can then be examined to determine which endstations are generating
unresolvable traffic.

Also, if a particular destination is repeatedly seen to be
unresolvable, the server calls the Block server (Section 4.3.6) to
determine whether the address should be blocked.

4.3.6 Block Server

The Resolve Block server is called when a particular destination has
been repeatedly seen to be unresolvable. This typically happens
when, unknown to the packet source, the destination endstation is
either not currently available or no longer exists.

If the Block server determines that the unresolved address has
exceeded a configurable request threshold, the address is added to
the server's Block Table. Interswitch Resolve request messages for
addresses listed in the Block Table are sent less frequently, thereby
reducing the amount of Interswitch Resolve traffic throughout the
fabric.

If an address listed in the Block Table is later successfully
resolved by and Interswitch Resolve request message, the address is
removed from the table.

4.4 Policy Service Center

Once the destination address of the call packet has been resolved,
the Policy Service Center is called to determine the validity of the
requested call connection based on the VLAN policy of the source and
destination VLANs.

4.4.1 Unicast Rules Server

The Policy Unicast Rules server recognizes two VLAN policy values:
Open or Secure. The default policy for all VLANs is Open.

The policy value is used as follows when determining the validity of
a requested call connection:

- If the VLAN policy of either the source or destination cannot be
  determined, the Filter Service Center is called to establish a
  filter (i.e., blocked) for the SA/DA pair.

- If the source and destination endstations belong to the same VLAN,
  then the connection is permitted regardless of the VLAN policy.

- If the source and destination endstations belong to different
  VLANs, but both VLANs are running with an Open policy, then the
  connection is permitted, providing cut-through switching between
  different VLAN(s).

- If the source and destination endstations belong to different
  VLANs and one or both of the VLANs are running with a Secure
  policy, then the Flood Service Center (Section 4.8) is called to
  broadcast the packet to other SFVLAN switches having ports or
  endstations that belong to the same VLAN as the packet source.

  Note that if any of the VLANs to which the source or destination
  belong has a Secure policy, then the policy used in the above
  algorithm is Secure.

4.5 Connect Service Center

   Once the Policy Service Center (Section 4.4) has determined that a
   requested call connection is valid, the Connect Service Center is
   called to set up the connection. Note that connectivity between two
   endstations within the fabric is established on a switch-by-switch
   basis as the call progresses through the fabric toward its
   destination. No synchronization is needed between switches to
   establish an end-to-end connection.

   The Connect Service Center maintains a Connection Table containing
   information for all connections currently active on the switch's
   local ports.

   Connections are removed from the Connection Table when one of the
   endstations is moved to a new switch (Section 4.1.2) or when the
   Topology Link State server (Section 4.2.3) notifies the Connect
   Service Center that a network link has failed. Otherwise,
   connections are not automatically aged out or removed from the
   Connection Table until a certain percentage threshold (HiMark) of
   table capacity is reached and resources are needed. At that point,
   some number of connections (typically 100) are aged out and removed
   at one time.

4.5.1 Local Server

   If the destination endstation resides on the local switch, the
   Connect Local server establishes a connection between the source and
   destination ports. Note that if the source and destination both
   reside on the same physical port, a filter connection is established
   by calling the Filter Service Center (Section 4.6).

4.5.2 Link State Server

   The Connect Link State server is called if the destination endstation
   of the proposed connection does not reside on the local switch.

   The server executes a call to the Path Link State server (Section
   4.7.1) which returns up to three "best" paths of equal cost from the
   local switch to the destination switch. If more than one path is
   returned, the server chooses a path that provides the best load
   balancing of user traffic across the fabric.

4.5.3 Directory Server

   The Connect Directory server is called if the Connect Link State
   server is unable to provide a path for some reason.

   The server examines the local directory to determine on which switch
   the destination endstation resides. If the port of access to the
   destination switch is known, then a connection is established using
   that port as the outport of the connection.

4.6 Filter Service Center

   The Filter Service Center is responsible for establishing filtered
   connections. This service center is called by the Connect Local
   server (Section 4.5.1) if the source and destination endstations
   reside on the same physical port, and by the Policy Service Center
   (Section 4.4) if the VLAN of either the source or destination is
   indeterminate.

   A filter connection is programmed in the switch hardware with no
   specified outport. That is, the connection is programmed to discard
   any traffic for that SA/DA pair.

4.7 Path Service Center

   The Path Service Center is responsible for determining the path from
   a source to a destination.

4.7.1 Link State Server

   The Path Link State server is called by the Connect Link State server
   (Section 4.5.2) to return up to three best paths of equal cost
   between a source and destination pair of endstations. These best
   paths are calculated by the Topology Link State server (Section
   4.2.3).

   The Path Link State server is also called by the Connect Service
   Center to return a complete source-to-destination path consisting of
   a list of individual switch port names. A switch port name consists
   of the switch base MAC address and a port instance relative to the
   switch.

4.7.2 Spanning Tree Server

   The Path Spanning Tree server is called by any server needing to
   forward an undirected message out over the switch flood path. The
   server returns a port mask indicating which local ports are currently
   enabled as outports of the switch flood path. The switch flood path
   is calculated by the Topology Spanning Tree server (Section 4.2.2).

4.8 Flood Service Center

   If the Resolve Service Center (Section 4.3) is unable to resolve the
   destination address of a packet, it invokes the Flood Service Center
   to broadcast the unresolved packet.

4.8.1 Tag-Based Flood Server

   The Tag-Based Flood server encapsulates the unresolved packet into an
   Interswitch Tag-Based Flood message (Section 6.6), along with a list
   of Virtual LAN identifiers specifying those VLANs to which the source
   endstation belongs. The message is then sent out over the switch
   flood path to all other switches in the fabric.

   When a switch receives an Interswitch Tag-Based Flood message, it
   examines the encapsulated header to determine the VLAN(s) to which
   the packet should be sent. If any of the switch's local access ports
   belong to one or more of the specified VLANs, the switch strips off
   the tag-based header and forwards the original packet out the
   appropriate access port(s).

   The switch also forwards the entire encapsulated packet along the
   switch flood path to its downstream neighboring switches, if any.

5. Monitoring Call Connections

   The SecureFast VLAN product permits monitoring of user traffic moving
   between two endstations by establishing a call tap on the connection
   between the two stations. Traffic can be monitored in one or both
   directions along the connection path.

5.1 Definitions

   In addition to the terms defined in Section 1.2, the following terms
   are used in this description of the call tap process.

Originating Switch

   The originating switch is the switch that requests the call tap.
   Any switch along a call connection path may request a tap on that
   call connection.

Probe

   The tap probe is the device to receive a copy of the call
   connection data. The probe is attached to a port on the probe
   switch.

Probe Switch

   The probe switch (also known as the terminating switch) is the
   switch to which the probe is attached. The probe switch can be
   anywhere in the topology.

5.2 Tapping a Connection

   A request to tap a call connection between two endstations can
   originate on any switch along the call connection path -- the ingress
   switch, the egress switch, or any of the intermediate switches. The
   call connection must have already been established before a call tap
   request can be issued. The probe device can be attached to any
   switch in the topology.

5.2.1 Types of Tap Connections

   A call tap is enabled by setting up an auxiliary tap connection
   associated with the call being monitored. Since the tap must
   originate on a switch somewhere along the call connection path, the
   tap connection path will pass through one or more of the switches
   along the call path. However, since the probe switch can be anywhere
   in the switch fabric, the tap path and the call path may diverge at
   some point.

   Therefore, on each switch along the tap path, the tap connection is
   established in one of three ways:

   - The existing call connection is used with no modification.

       When both the call path and tap path pass through the switch,
       and the inport and outports of both connections are identical,
       the switch uses the existing call connection to route the tap.

   - The existing call connection is modified.

When both the call path and tap path pass through the switch,
but the call path outport is different from the tap path
outport, the switch enables an extra outport in either one or
both directions of the call connection, depending on the
direction of the tap. This happens under two conditions.

- If the switch is also the probe switch, an extra outport is
  enabled to the probe.

- If the switch is the point at which the call path and the tap path
  diverge, an extra outport is enabled to the downstream neighbor
  on that leg of the switch flood path on which the probe switch
  is located.

- A new connection is established.

  If the call path does not pass through the switch (because the
  tap path has diverged from the call path), a completely new
  connection is established for the tap.

5.2.2 Locating the Probe and Establishing the Tap Connection

To establish a call tap, the originating switch formats an
Interswitch Tap request message (Section 6.7) and sends it out over
the switch flood path to all other switches in the topology.

Note:

If the originating switch is also the probe switch, no
Interswitch Tap request message is necessary.

As the Interswitch Tap request message travels out along the switch
flood path, each switch receiving the message checks to see if it is
the probe switch and does the following:

- If the switch is the probe switch, it establishes the tap
  connection by either setting up a new connection or modifying the
  call connection, as appropriate (see Section 5.2.1). It then
  reformats the Tap request message to be a Tap response message
  with a status indicating that the probe has been found, and sends
  the message back to its upstream neighbor.

- If the switch is not the probe switch, it forwards the Tap request
  message to all its downstream neighbors (if any).

- If the switch is not the probe switch and has no downstream
  neighbors, it reformats the Tap request message to be a Tap
  response message with a status indicating that the probe is not

located on that leg of the switch flood path.   It then sends the
response message back to its upstream neighbor.

When a switch forwards an Interswitch Tap request message to its
downstream neighbors, it keeps track of the number of requests it
has sent out.

- If a response is received with a status indicating that the probe
  switch is located somewhere downstream, the switch establishes the
  appropriate type of tap connection (see Section 5.2.1).   It then
  formats a Tap response message with a status indicating that the
  probe has been found and passes the message to its upstream
  neighbor.

- If no responses are received with a status indicating that the
  probe switch is located downstream, the switch formats a Tap
  response message with a status indicating that the probe has not
  been found and passes the message to its upstream neighbor.

5.2.3 Status Field

The status field of the Interswitch Tap request/response message
contains information about the state of the tap.   Some of these
status values are transient and are merely used to track the progress
of the tap request.   Other status values are stored in the tap table
of each switch along the tap path for use when the tap is torn down.
The possible status values are as follows:

- StatusUnassigned.   This is the initial status of the Interswitch
  Tap request message.

- OutportDecisionUnknown.   The tap request is still moving
  downstream along the switch flood path.   The probe switch had not
  yet been found.

- ProbeNotFound.   The probe switch is not located on this leg of the
  switch flood path.

- DisableOutport.   The probe switch is located on this leg of the
  switch flood path, and the switch has had to either modify the
  call connection or establish a new connection to implement the tap
  (see Section 5.2.1).   When the tap is torn down, the switch will
  have to disable any additional outports that have been enabled for
  the tap.

- KeepOutport.   The probe switch is located on this leg of the
  switch flood path, and the switch was able to route the tap over
  the existing call path (see Section 5.2.1).   Any ports used for

the tap will remain enabled when the tap is torn down.

5.3 Untapping a Connection

A request to untap a call connection must be issued on the tap originating switch -- that is, the same switch that issued the tap request.

To untap a call connection, the originating switch sends an Interswitch Untap request message (Section 6.7) out over the switch flood path to all other switches in the topology. The message is sent over the switch flood path, rather than the tap connection path, to ensure that all switches that know of the tap are properly notified, even if the switch topology has changed since the tap was established.

When a switch receives an Interswitch Untap request message, it checks to see if it is handling a tap for the specified call connection. If so, the switch disables the tap connection, as follows:

- If a new connection was added for the tap, the connection is deleted from the connection table.

- If additional outports were enabled on the call connection, they are disabled.

The switch then forwards the Interswitch Untap request message to its downstream neighbor (if any). If the switch has no downstream neighbors, it formats an untap response and sends the message back to its upstream neighbor.

When a switch forwards an Interswitch Untap request message to its downstream neighbors, it keeps track of the number of requests it has sent out and does not respond back to its upstream neighbor until all untap requests have been responded to. Once all responses have been received, the switch handles any final cleanup for the tap and then sends a single Interswitch Untap response message to its upstream neighbor.

6. Interswitch Message Protocol (ISMP)

   The InterSwitch Message protocol (ISMP) provides a consistent method
   of encapsulating and transmitting messages exchanged between switches
   to create and maintain the databases and provide other control
   services and functionality required by the SFVLAN product.

6.1 General Packet Structure

   ISMP packets are of variable length and have the following general
   structure:

   - Frame header
   - ISMP packet header
   - ISMP message body

   Each of these packet segments is discussed separately in the
   following subsections.

6.1.1 Frame Header

   ISMP packets are encapsulated within an IEEE 802-compliant frame
   using a standard header as shown below:

```
        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
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    00 |                                                               |
       +       Destination address     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    04 |                               |                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+          Source address       +
    08 |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    12 |             Type              |                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
    16 |                                                               |
       +                                                               +
       :                                                               :
```
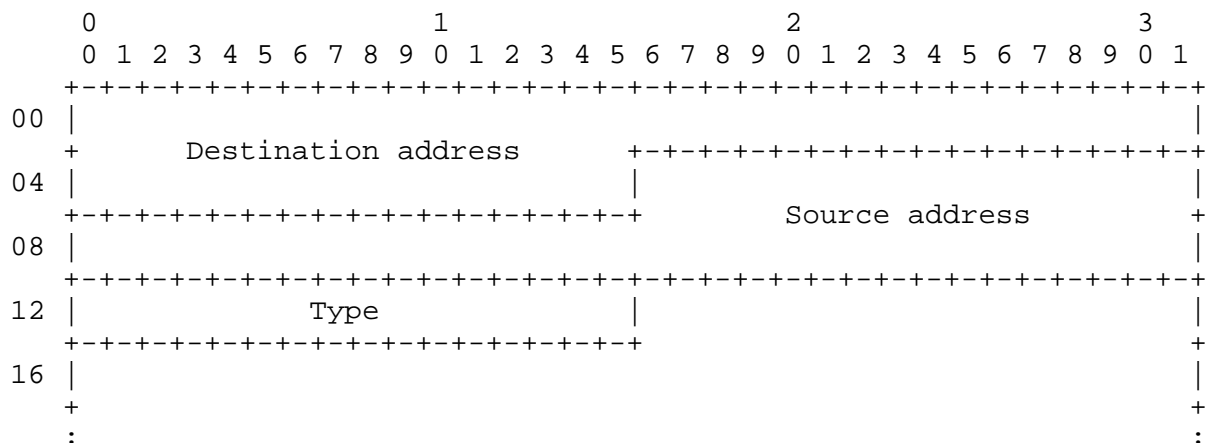
   Destination address

       This 6-octet field contains the Media Access Control (MAC) address
       of the multicast channel over which all switches in the fabric
       receive ISMP packets. Except where otherwise noted, this field

contains the multicast address of the control channel over which
all switches in the fabric receive ISMP packets -- a value of 01-
00-1D-00-00-00.

Source address

Except where otherwise noted, this 6-octet field contains the
physical (MAC) address of the switch originating the ISMP packet.

Type

This 2-octet field identifies the type of data carried within the
frame. Except where otherwise noted, the type field of ISMP
packets contains the value 0x81FD.

6.1.2 ISMP Packet Header

There are two versions of the ISMP packet header in use by the
SecureFast VLAN product.

6.1.2.1 Version 2

The version 2 ISMP packet header consists of 6 octets, as shown
below:

```
      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
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  00 |/////////////////////////////////////////////////////////////|
     ://////// Frame header ////////////////////////////////////////:
     +//////// (14 octets)  /////////+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  12 |//////////////////////////////|             Version           |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  16 |       ISMP message type       |       Sequence number        |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  20 |                                                               |
     +                                                               +
     :                                                               :
```

Frame header

This 14-octet field contains the frame header (Section 6.1.1).

Version

   This 2-octet field contains the version number of the InterSwitch
   Message Protocol to which this ISMP packet adheres. This document
   describes ISMP Version 2.0.

ISMP message type

   This 2-octet field contains a value indicating which type of ISMP
   message is contained within the message body. The following table
   lists each ISMP message, along with its message type and the
   section within this document that describes the message in detail:

Message Name	Type	Description
Interswitch Link State message	3	See note below
Interswitch BPDU message	4	Section 6.2
Interswitch Remote Blocking message	4	Section 6.3
Interswitch Resolve message	5	Section 6.4
Interswitch New User message	5	Section 6.5
Interswitch Tag-Based Flood message	7	Section 6.6
Interswitch Tap/Untap message	8	Section 6.7

   Note:

      The Link State messages used by the VLS Protocol are not
      described in this document. For a detailed description of
      these messages, see [IDvlsp].

Sequence number

   This 2-octet field contains an internally generated sequence
   number used by the various protocol handlers for internal
   synchronization of messages.

6.1.2.2 Version 3

   The version 3 ISMP packet header is used only by the Interswitch
   Keepalive message. That message is not described in this document.
   For a detailed description of the version 3 ISMP packet header, see
   [IDhello].

6.1.3 ISMP Message Body

   The ISMP message body is a variable-length field containing the
   actual data of the ISMP message. The length and content of this
   field are determined by the value found in the message type field.

   See the following sections for the exact format of each message type.

6.2 Interswitch BPDU Message

   The Interswitch BPDU message consists of a variable number of octets,
   as shown below:
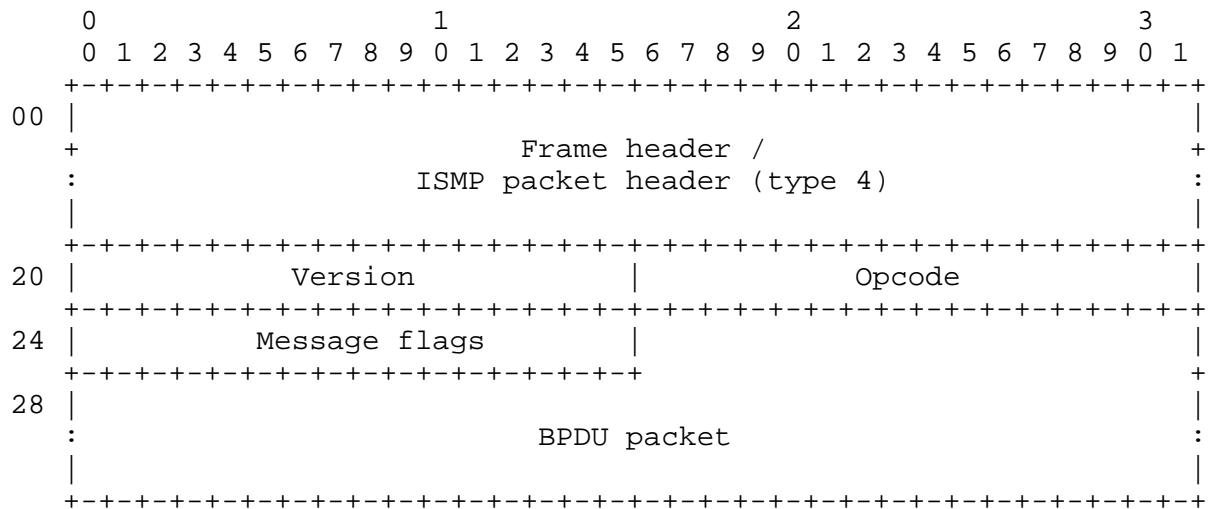
```
       0                   1                   2                   3
       0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   00 |                                                               |
      +                          Frame header /                      +
      :                   ISMP packet header (type 4)                :
      |                                                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   20 |            Version             |            Opcode            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   24 |          Message flags         |                             |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                             +
   28 |                                                               |
      :                          BPDU packet                         :
      |                                                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   Frame header/ISMP packet header

      This 20-octet field contains the frame header and the ISMP packet
      header.

   Version

      This 2-octet field contains the version number of the message
      type. This document describes ISMP message type 4, version 1.

   Opcode

      This 2-octet field contains the operation type of the message. For
      an Interswitch BPDU message, the value should be 1.

   Message flags

      This 2-octet field is currently unused. It is reserved for future
      use.

   BPDU packet

      This variable-length field contains an IEEE-compliant 802.2 Bridge
      Protocol Data Unit. See [IEEE] for a detailed description of the
      contents of this field.

6.3 Interswitch Remote Blocking Message

   The Interswitch Remote Blocking message consists of 30 octets, as
   shown below:
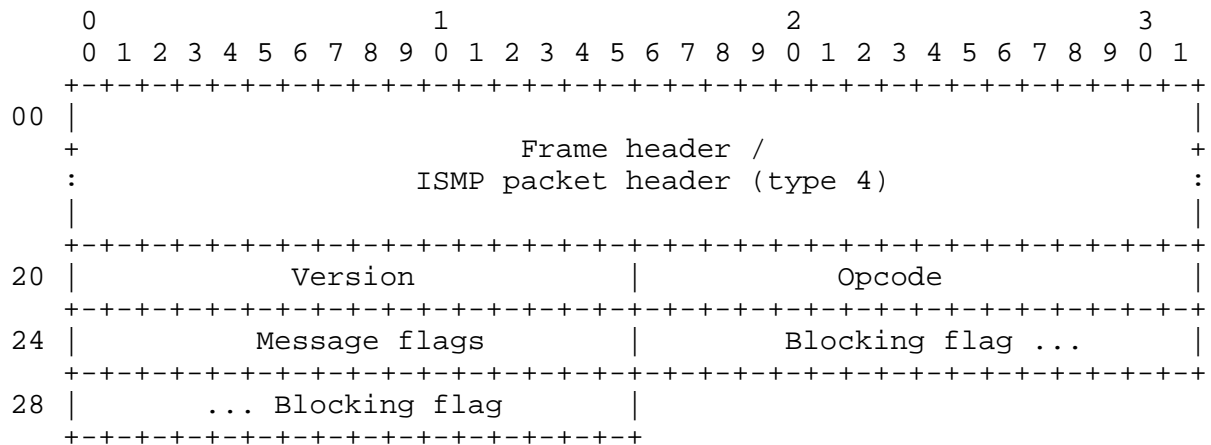
```
      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  00 |                                                               |
     +                         Frame header /                        +
     :                   ISMP packet header (type 4)                 :
     |                                                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  20 |            Version             |            Opcode             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  24 |         Message flags          |        Blocking flag ...      |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  28 |      ... Blocking flag         |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```


   Frame header/ISMP packet header

      This 20-octet field contains the frame header and the ISMP packet
      header.

   Version

      This 2-octet field contains the version number of the message
      type. This document describes ISMP message type 4, version 1.

Opcode

This 2-octet field contains the operation type of the message.
Valid values are as follows:

2   Enable/disable remote blocking
3   Acknowledge previously received Remote Blocking message

Message flags

This 2-octet field is currently unused. It is reserved for
future use.

Blocking flag

This 4-octet field contains a flag indicating the state of
remote blocking on the link over which the message was
received. A value of 1 indicates remote blocking is on and no
undirected ISMP messages should be sent over the link. A value
of 0 indicates remote blocking is off. This flag is irrelevant
if the operation type (Opcode) of the message has a value of 3.

6.4 Interswitch Resolve Message

There are two versions of the Interswitch Resolve message used by the
SecureFast VLAN product.

6.4.1 Prior to Version 1.8

The Interswitch Resolve message used by SFVLAN prior to version 1.8
consists of a variable number of octets, as shown below:

```
       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
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   00 |                                                               |
      +                       Frame header /                         +
      :               ISMP packet header (type 5)                    :
      |                                                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   20 |           Version             |            Opcode             |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   24 |           Status              |           Call Tag            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   28 |                                                               |
      +        Source MAC of packet     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   32 |                                 |                             |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+   Originating switch MAC    +
   36 |                                                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   40 |                                                               |
      +        Owner switch MAC         +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   44 |                                 |                             |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                             +
   48 |                                                               |
      :                 Known destination address                    :
      |                                                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    n |    Count      |                                               |
      +-+-+-+-+-+-+-+-+                                               +
  n+4 |                         Resolve list                         |
      :                                                              :
      |                                                              |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
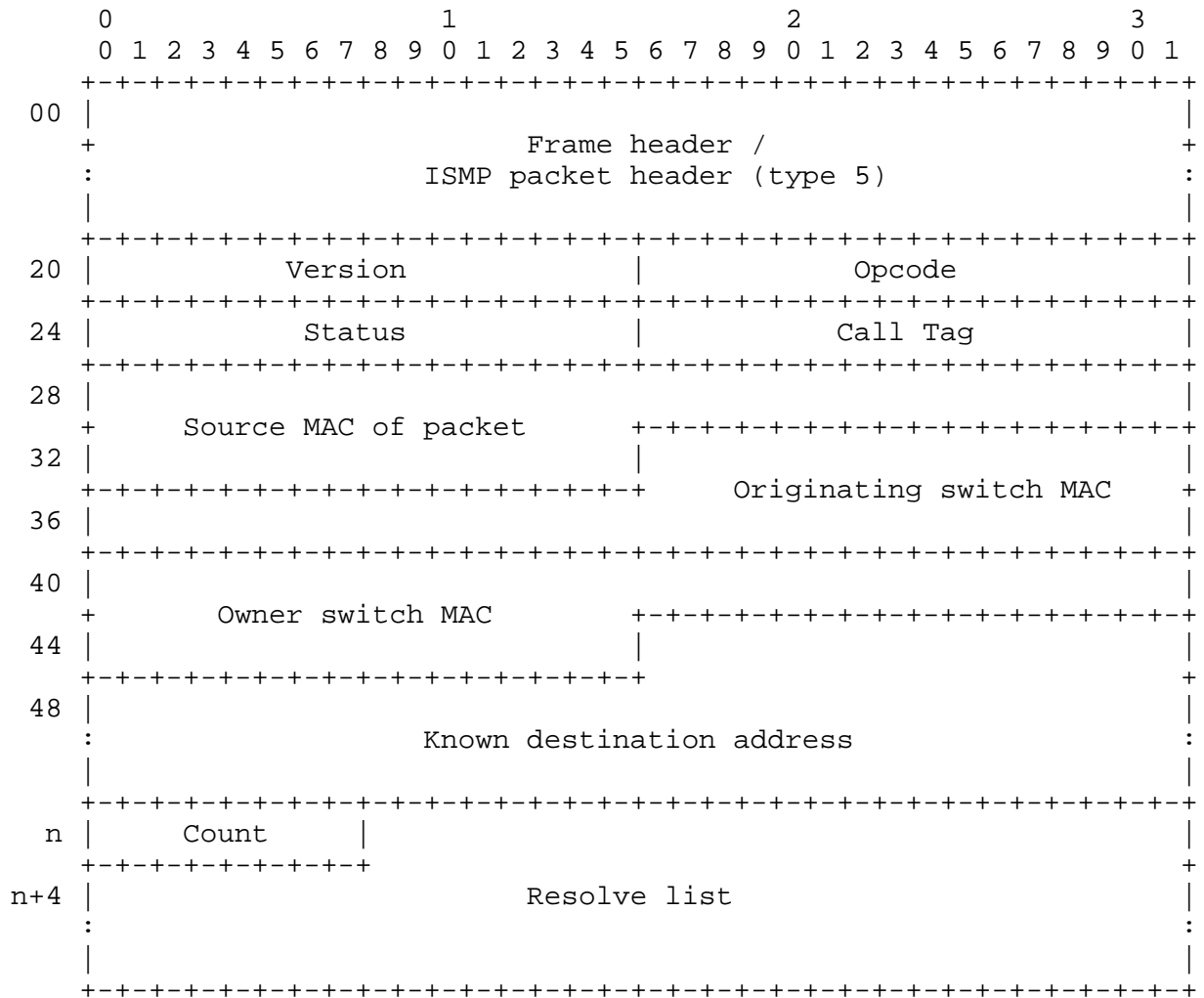
        n = 46 + length of known address TLV

   In the following description of the message fields, the term
   "originating" switch refers to the switch that issued the original
   Interswitch Resolve request. The term "owner" switch refers to that
   switch to which the destination endstation is attached. And the term
   "responding" switch refers to either the "owner" switch or to a
   switch at the end of the switch flood path that does not own the
   endstation but issues an Interswitch Resolve response because it has
   no downstream neighbors.

With the exception of the resolve list (which has a different size
and format in a Resolve response message), all fields of an
Interswitch Resolve message are allocated by the originating switch,
and unless otherwise noted below, are written by the originating
switch.

Frame header/ISMP packet header

   This 20-octet field contains the frame header and the ISMP packet
   header.

Version

   This 2-octet field contains the version number of the message
   type. This document describes ISMP message type 5, version 1.

Opcode

   This 2-octet field contains the operation code of the message.
   Valid values are as follows:

      1   The message is a Resolve request.
      2   The message is a Resolve response.
      3   (unused in Resolve messages)
      4   (unused in Resolve messages)

   The originating switch writes a value of 1 to this field, while
   the responding switch writes a value of 2.

Status

   This 2-octet field contains the status of a Resolve response
   message. Valid values are as follows:

      0   The Resolve request succeeded (ResolveAck).
      1   (unused)
      2   The Resolve request failed (Unknown).

   This field is written by the responding switch.

Call tag

   This 2-octet field contains the call tag of the endstation packet
   for which this Resolve request is issued. The call tag is a 16-
   bit value (generated by the originating switch) that uniquely
   identifies the packet.

Source MAC of packet

   This 6-octet field contains the physical (MAC) address of the
   endstation that originated the packet identified by the call tag.

Originating switch MAC

   This 6-octet field contains the physical (MAC) address of the
   switch that issued the original Resolve request.

Owner switch MAC

   This 6-octet field contains the physical (MAC) address of the
   switch to which the destination endstation is attached -- that is,
   the switch that was able to resolve the requested addressing
   information. This field is written by the owner switch.

   If the status of the response is Unknown, this field is
   irrelevant.

Known destination address

   This variable-length field contains the known attribute of the
   destination endstation address. This address is stored in
   Tag/Length/Value format. (See Section 2.3.)

Count

   This 1-octet field contains the number of address attributes
   requested or returned. This is the number of items in the resolve
   list.

Resolve list

   This variable-length field contains a list of the address
   attributes either requested by the originating switch or returned
   by the owner switch. Note that in a Resolve request message, this
   list contains only the tags of the requested address attributes
   (see Section 2.3). On the other hand, a Resolve response message
   with a status of ResolveAck contains the full TLV of each resolved
   address attribute. The number of entries in the list is specified
   in the count field.

   In an Interswitch Resolve response message, this field is
   irrelevant if the status of the response is Unknown.

6.4.2 Version 1.8

   The Interswitch Resolve message used by SFVLAN version 1.8 consists
   of a variable number of octets, as shown below:

```
        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
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    00 |                                                               |
       +                        Frame header /                        +
       :                  ISMP packet header (type 5)                 :
       |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    20 |            Version             |            Opcode             |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    24 |            Status             |           Call Tag            |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    28 |                                                               |
       +       Source MAC of packet    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    32 |                               |                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+       Originating switch MAC  +
    36 |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    40 |                                                               |
       +         Owner switch MAC      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    44 |                               |                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
    48 |                                                               |
       :                  Known destination address                   :
       |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     n |    Count      |                                               |
       +-+-+-+-+-+-+-+-+                                               +
   n+4 |                         Resolve list                         |
       :                                                              :
       |                                                              |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    n1 |                                                               |
       +     Actual dest switch MAC    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                               |                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+       Downlink chassis MAC    +
  n1+8 |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 n1+12 |                                                               |
       +        Actual chassis MAC     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                               |                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
 n1+20 |                                                               |
       +                         Domain name                          +
       :                                                              :
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
        n = 46 + length of known address TLV
        n1 = n + length of Resolve list
```

In the following description of the message fields, the term
"originating" switch refers to the switch that issued the original
Interswitch Resolve request. The term "owner" switch refers to that
switch to which the destination endstation is attached. And the term
"responding" switch refers to either the "owner" switch or to a
switch at the end of the switch flood path that does not own the
endstation but issues an Interswitch Resolve response because it has
no downstream neighbors.

With the exception of the resolve list (which has a different size
and format in a Resolve response message) and the four fields
following the resolve list, all fields of an Interswitch Resolve
message are allocated by the originating switch, and unless otherwise
noted below, are written by the originating switch.

Frame header/ISMP packet header

   This 20-octet field contains the frame header and the ISMP packet
   header.

Version

   This 2-octet field contains the version number of the message
   type. This section describes version 3 of the Interswitch Resolve
   message.

Opcode

   This 2-octet field contains the operation code of the message.
   Valid values are as follows:

      1   The message is a Resolve request.
      2   The message is a Resolve response.
      3   (unused in Resolve messages)
      4   (unused in Resolve messages)

   The originating switch writes a value of 1 to this field, while
   the responding switch writes a value of 2.

Status

   This 2-octet field contains the status of a Resolve response
   message. Valid values are as follows:

      0   The Resolve request succeeded (ResolveAck).
      1   (unused)
      2   The Resolve request failed (Unknown).

   This field is written by the responding switch.

Call tag

   This 2-octet field contains the call tag of the endstation packet
   for which this Resolve request is issued. The call tag is a 16-
   bit value (generated by the originating switch) that uniquely
   identifies the packet.

Source MAC of packet

   This 6-octet field contains the physical (MAC) address of the
   endstation that originated the packet identified by the call tag.

Originating switch MAC

   This 6-octet field contains the physical (MAC) address of the
   switch that issued the original Resolve request.

Owner switch MAC

   This 6-octet field contains the physical (MAC) address of the
   switch to which the destination endstation is attached -- that is,
   the switch that was able to resolve the requested addressing
   information. This field is written by the owner switch.

   If the status of the response is Unknown, this field is
   irrelevant.

Known destination address

   This variable-length field contains the known attribute of the
   destination endstation address. This address is stored in
   Tag/Length/Value format.

Count

   This 1-octet field contains the number of address attributes
   requested or returned. This is the number of items in the resolve
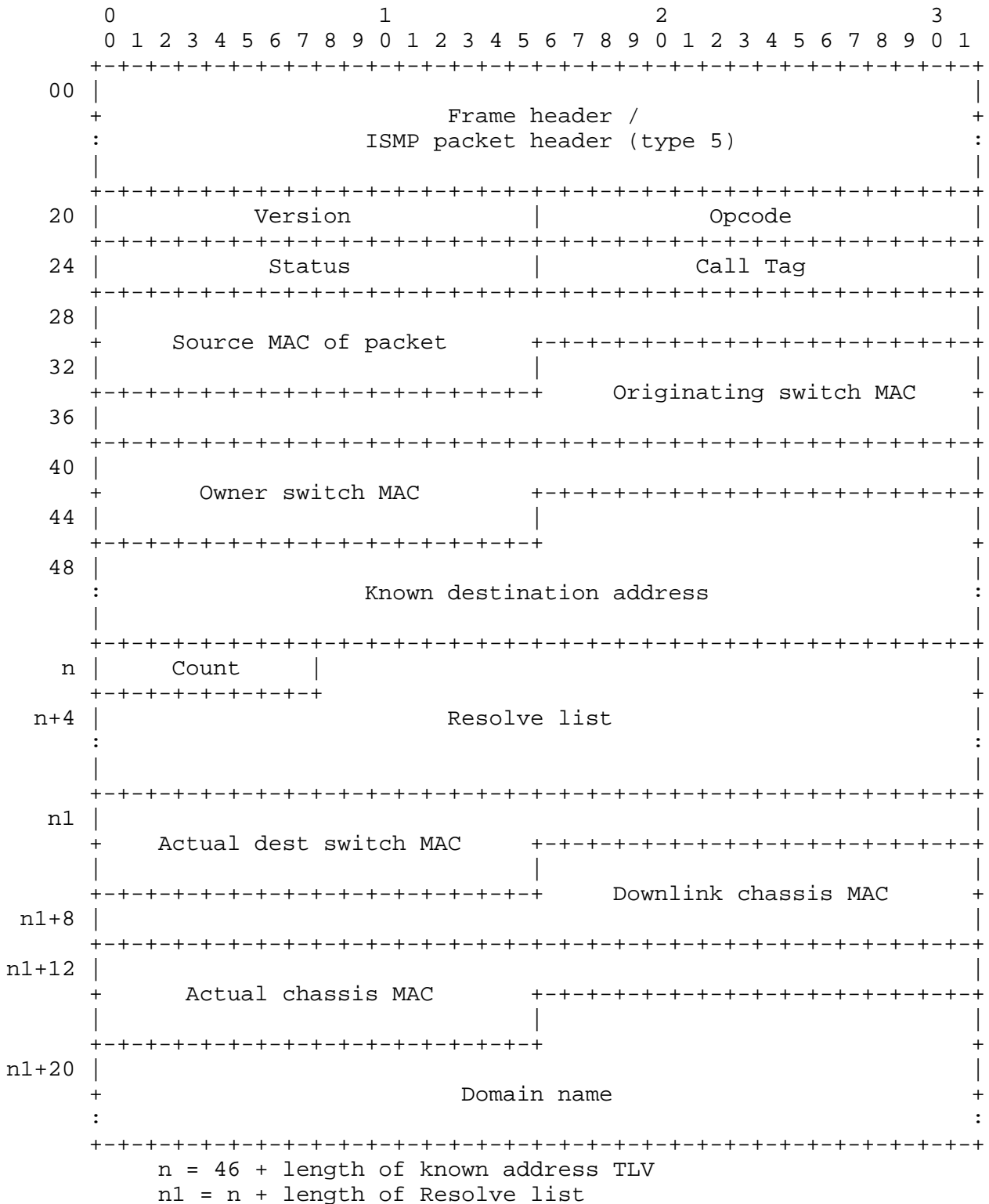   list.

Resolve list

   This variable-length field contains a list of the address
   attributes either requested by the originating switch or returned
   by the owner switch. Note that in a Resolve request message, this
   list contains only the tags of the requested address attributes.
   On the other hand, a Resolve response message with a status of
   ResolveAck contains the full TLV of each resolved address
   attribute. The number of entries in the list is specified in the
   count field.

   In an Interswitch Resolve response message, this field is
   irrelevant if the status of the response is Unknown.

Actual destination switch MAC

   This 6-octet field contains the physical (MAC) address of the
   actual switch within the chassis to which the endstation is
   attached. If the status of the response is Unknown, this field is
   irrelevant.

Downlink chassis MAC

   This 6-octet field contains the physical (MAC) address of the
   downlink chassis. If the status of the response is Unknown, this
   field is irrelevant.

Actual chassis MAC

   This 6-octet field contains the physical (MAC) address of the
   uplink chassis. If the status of the response is Unknown, this
   field is irrelevant.

Domain name

   This 16-octet field contains the ASCII name of the domain. If the
   status of the response is Unknown, this field is irrelevant.

6.5 Interswitch New User Message

   The Interswitch New User message consists of a variable number of
   octets, as shown below:
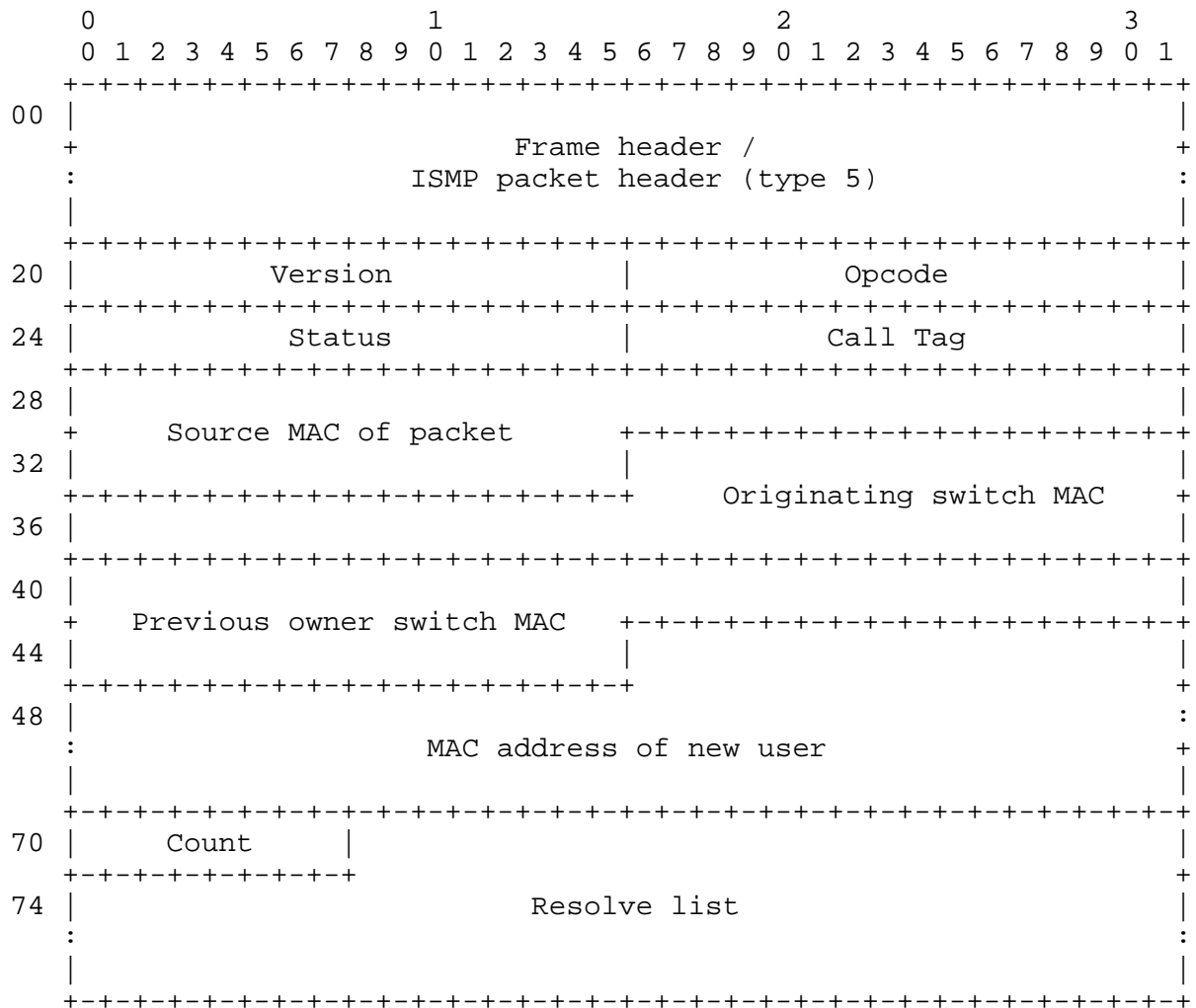
```
      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  00 |                                                               |
     +                        Frame header /                         +
     :                  ISMP packet header (type 5)                  :
     |                                                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  20 |            Version            |            Opcode             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  24 |            Status            |           Call Tag            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  28 |                                                               |
     +         Source MAC of packet     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  32 |                                  |                            |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+   Originating switch MAC     +
  36 |                                                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  40 |                                                               |
     +      Previous owner switch MAC    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  44 |                                   |                           |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+   +
  48 |                                                               :
     :                 MAC address of new user                      +
     |                                                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  70 |    Count      |                                               |
     +-+-+-+-+-+-+-+-+   +
  74 |                         Resolve list                          |
     :                                                               :
     |                                                               |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   In the following description of the message fields, the term
   "originating" switch refers to the switch that issued the original
   Interswitch New User request. The term "previous owner" switch
   refers to that switch to which the endstation was previously
   attached. And the term "responding" switch refers to either the
   "previous owner" switch or to a switch at the end of the switch flood
   path that did not own the endstation but issues an Interswitch New
   User response because it has no downstream neighbors.

With the exception of the resolve list, all fields of an Interswitch
New User message are allocated by the originating switch, and unless
otherwise noted below, are written by the originating switch.

Frame header/ISMP packet header

   This 20-octet field contains the frame header and the ISMP packet
   header.

Version

   This 2-octet field contains the version number of the message
   type. This document describes ISMP message type 5, version 1.

Opcode

   This 2-octet field contains the operation code of the message.
   Valid values are as follows:

      1   (unused in a New User message)
      2   (unused in a New User message)
      3   The message is a New User request.
      4   The message is a New User response.

   The originating switch writes a value of 3 to this field, while
   the responding switch writes a value of 4.

Status

   This 2-octet field contains the status of a New User response
   message. Valid values are as follows:

      0   VLAN resolution successful (NewUserAck)
      1   (unused)
      2   VLAN resolution unsuccessful (NewUserUnknown)

   This field is written by the responding switch.

Call tag

   This 2-octet field contains the call tag of the endstation packet
   for which this New User request is issued. The call tag is a 16-
   bit value (generated by the originating switch) that uniquely
   identifies the packet that caused the switch to identify the
   endstation as a new user.

Source MAC of packet

   This 6-octet field contains the physical (MAC) address of the
   endstation that originated the packet identified by the call tag.

Originating switch MAC

   This 6-octet field contains the physical (MAC) address of the
   switch that issued the original New User request.

Previous owner switch MAC

   This 6-octet field contains the physical (MAC) address of the
   switch to which the endstation was previously attached -- that is,
   the switch that was able to resolve the VLAN information. This
   field is written by the previous owner switch.

   If the status of the response is Unknown, this field is
   irrelevant.

MAC address of new user

   This 24-octet field contains the physical (MAC) address of the new
   user endstation, stored in Tag/Length/Value format.

Count

   This 1-octet field contains the number of VLAN identifiers
   returned. This is the number of items in the resolve list. This
   field is written by the previous owner switch.

   If the status of the response is Unknown, this field and the
   resolve list are irrelevant.

Resolve list

   This variable-length field contains a list of the VLAN identifiers
   of all static VLANs to which the endstation belongs, stored in
   Tag/Length/Value format (see Section 2.3). The number of entries
   in the list is specified in the count field. This list is written
   by the previous owner switch.

   If the status of the response is Unknown, this field is
   irrelevant.

6.6 Interswitch Tag-Based Flood Message

   There are two versions of the Interswitch Tag-Based Flood message
   used by the SecureFast VLAN product.

6.6.1 Prior to Version 1.8

   The Interswitch Tag-Based Flood message used by SFVLAN prior to
   version 1.8 consists of a variable number of octets, as shown below:

```
       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
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   00 |                                                               |
      +                         Frame header /                        +
      :                  ISMP packet header (type 7)                  :
      |                                                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   20 |             Version           |            Opcode             |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   24 |             Status            |           Call Tag            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   28 |                                                               |
      +        Source MAC of packet      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   32 |                                  |                            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+    Originating switch MAC    +
   36 |                                                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   40 |    Count      |                                               |
      +-+-+-+-+-+-+-+-+                                               +
   44 |                            VLAN list                          |
      :                                                               :
      |                                                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    n |                                                               |
      +                                                               +
      :                         Original packet                       :
      +                                                               +
      |                                                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
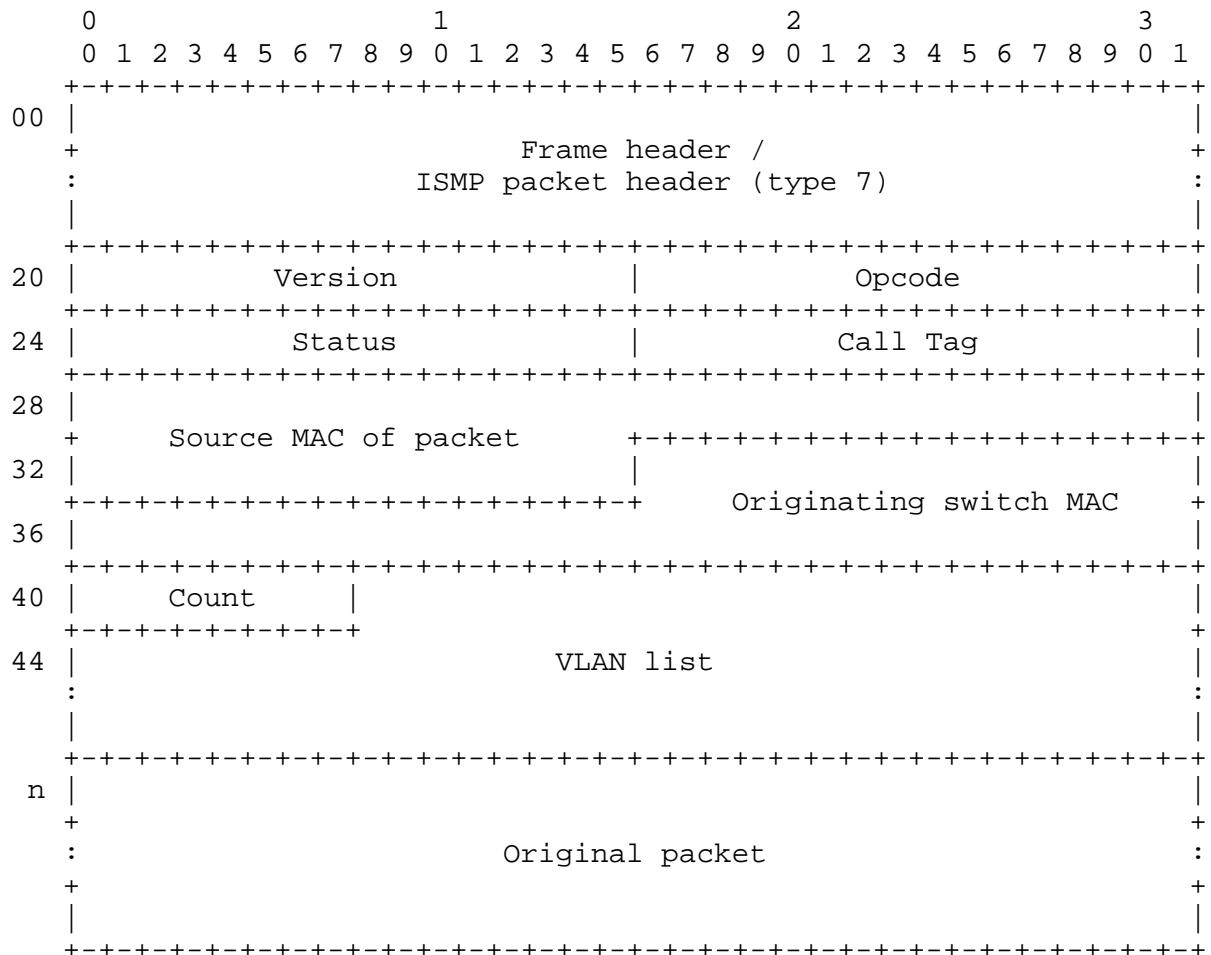
        n = 41 + length of VLAN list

Frame header/ISMP packet header

   This 20-octet field contains the frame header and the ISMP packet
   header.

Version

   This 2-octet field contains the version number of the message
   type. This document describes ISMP message type 7, version 1.

Opcode

   This 2-octet field contains the operation code of the message. The
   value here should be 1, indicating the message is a flood request.

Status

   This 2-octet field is currently unused. It is reserved for future
   use.

Call tag

   This 2-octet field contains the call tag of the endstation packet
   encapsulated within this tag-based flood message. The call tag is
   a 16-bit value (generated by the originating switch) that uniquely
   identifies the packet.

Source MAC of packet

   This 6-octet field contains the physical (MAC) address of the
   endstation that originated the packet identified by the call tag.

Originating switch MAC

   This 6-octet field contains the physical (MAC) address of the
   switch that issued the original tag-based flooded message.

Count

   This 1-octet field contains the number of VLAN identifiers
   included in the VLAN list.

VLAN list

   This variable-length field contains a list of the VLAN identifiers
   of all VLANs to which the source endstation belongs. Each entry
   in this list has the following format:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Value length  |                                               |
+-+-+-+-+-+-+-+-+                                               +
|                      VLAN identifier value                    |
:                                                               :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   The 1-octet value length field contains the length of the VLAN
   identifier. VLAN identifiers can be from 1 to 16 characters long.

Original packet

   This variable-length field contains the original packet as sent by
   the source endstation.

6.6.2 Version 1.8

   The Interswitch Tag-Based Flood message used by SFVLAN version 1.8
   consists of a variable number of octets, as shown below:

      Note:

         SFVLAN version 1.8 also recognizes the Interswitch Tag-Based
         Flood message as described in Section 6.6.1.

```
        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
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    00 |                                                               |
       +                         Frame header /                        +
       :                  ISMP packet header (type 7)                  :
       |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    20 |          VLAN identifier       |            Version            |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    24 |            Opcode              |            Status             |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    28 |           Call tag            |                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+     Source MAC of packet       +
    32 |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    36 |                                                               |
       +      Originating switch MAC    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    40 |                               |       Count       |           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+           +
    44 |                                                               |
       :                           VLAN list                          :
       |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     n |                                                               |
       +                                                               +
       :                        Original packet                       :
       +                                                               +
       |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

            n = 41 + length of VLAN list
```
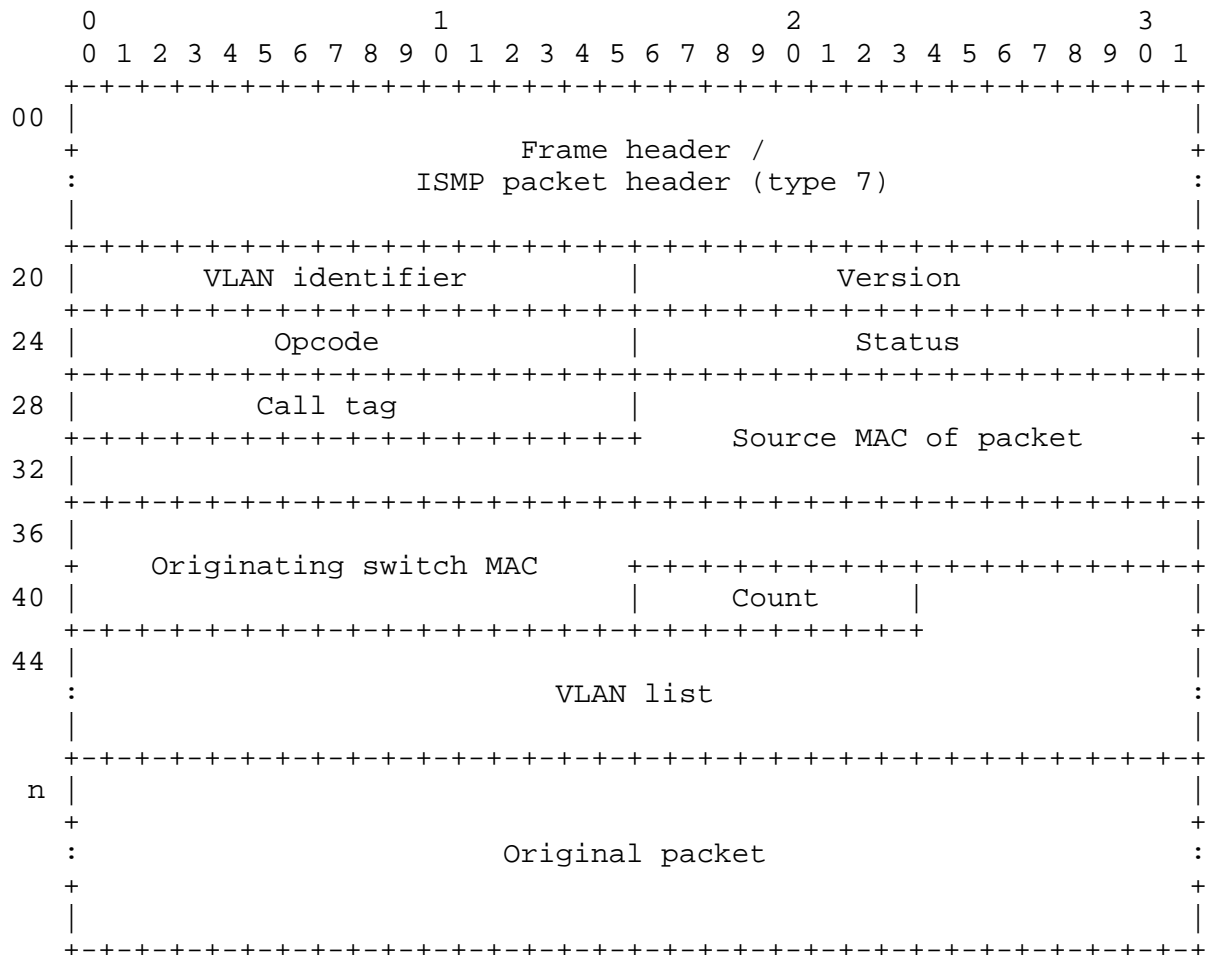
   Frame header/ISMP packet header

      This 20-octet field contains the frame header and the ISMP packet
      header.

- The frame header source address contains a value of 02-00-1D-
  00-xx-yy, where xx-yy is a value set by the VLAN Manager
  application to tag the frame header with the VLAN identifier.
  This value ranges from 2 to 4095. For example, a value of 100
  would be set as 00-64.

- The frame header type field contains a value of 0x81FF. Note
  that this differs from all other ISMP messages.

VLAN identifier

   This 2-octet field contains the VLAN identifier of the packet
   source.

Version

   This 2-octet field contains the version number of the message
   type. This section describes version 2 of the Interswitch Tag-
   Based Flood message.

Opcode

   This 2-octet field contains the operation code of the message.
   Valid values here are as follows:

   1 The message is a flood request. The original packet is
      complete within this message.

   2 The message is a fragmented flood request. The first portion
      of the original packet is contained in this message.

   3 The message is a fragmented flood request. The second portion
      of the original packet is contained in this message.

Status

   This 2-octet field is currently unused. It is reserved for future
   use.

Call tag

   This 2-octet field contains the call tag of the endstation packet
   encapsulated within this tag-based flood message. The call tag is
   a 16-bit value (generated by the originating switch) that uniquely
   identifies the packet.

Source MAC of packet

   This 6-octet field contains the physical (MAC) address of the
   endstation that originated the packet identified by the call tag.

Originating switch MAC

   This 6-octet field contains the physical (MAC) address of the
   switch that issued the original tag-based flooded message.

Count

   This 1-octet field contains the number of VLAN identifiers
   included in the VLAN list.

VLAN list

   This variable-length field contains a list of the VLAN identifiers
   of all VLANs to which the source endstation belongs. Each entry
   in this list has the following format:

```
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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Value length  |                                               |
+-+-+-+-+-+-+-+-+-+                                              +
|                         VLAN identifier value                 |
:                                                               :
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
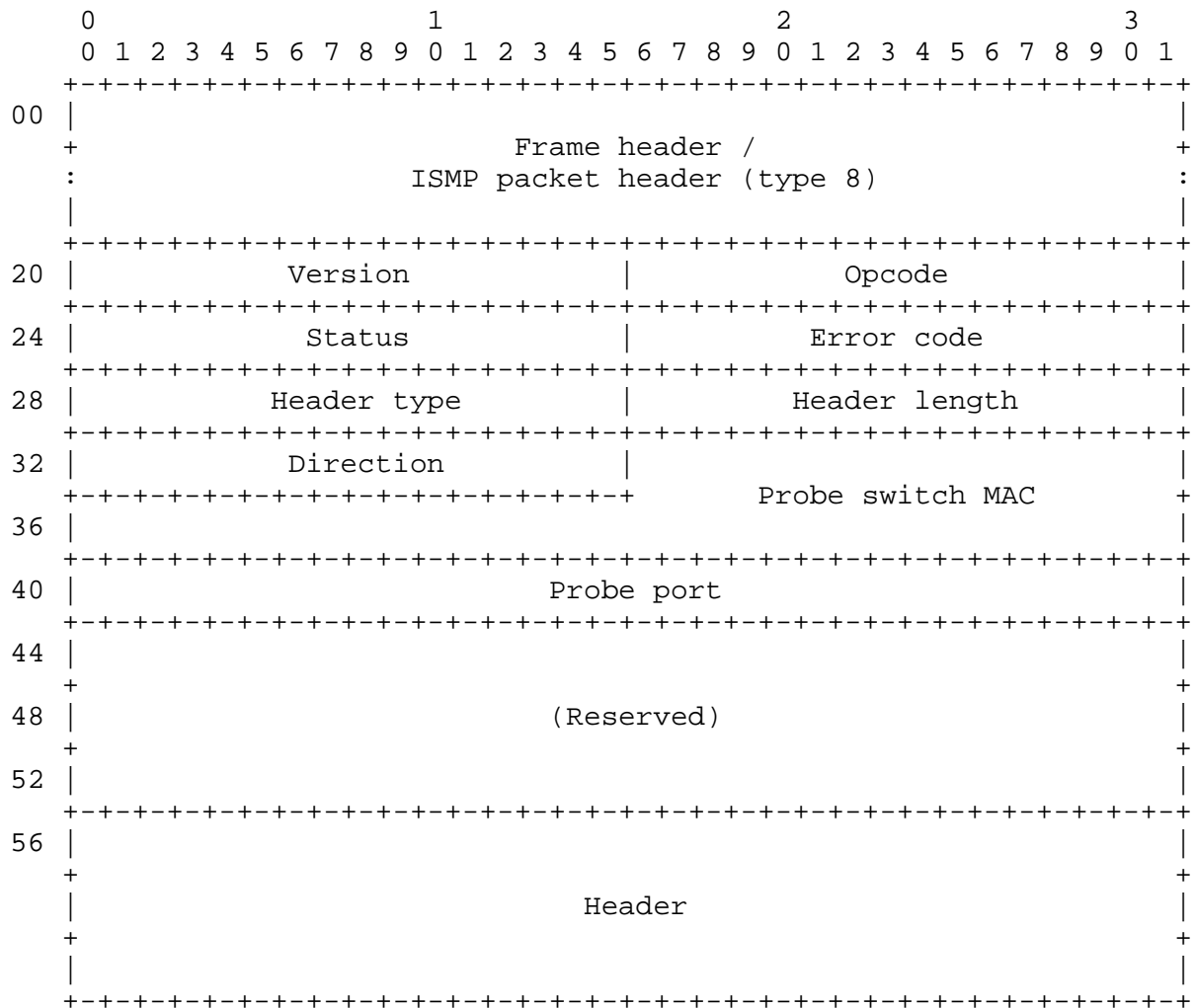
   The 1-octet value length field contains the length of the VLAN
   identifier. VLAN identifiers can be from 1 to 16 characters long.

Original packet

   This variable-length field contains the original packet as sent by
   the source endstation.

6.7 Interswitch Tap/Untap Message

   The Interswitch Tap/Untap message consists of a variable number of
   octets, as shown below:

```
        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
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    00 |                                                               |
       +                        Frame header /                        +
       :                  ISMP packet header (type 8)                 :
       |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    20 |            Version             |            Opcode             |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    24 |            Status             |          Error code           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    28 |         Header type           |         Header length         |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    32 |         Direction             |                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+         Probe switch MAC       +
    36 |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    40 |                          Probe port                           |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    44 |                                                               |
       +                                                               +
    48 |                          (Reserved)                           |
       +                                                               +
    52 |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    56 |                                                               |
       +                                                               +
       |                            Header                             |
       +                                                               +
       |                                                               |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   Frame header/ISMP packet header

      This 20-octet field contains the frame header and the ISMP packet
      header.

   Version

      This 2-octet field contains the version number of the message
      type. This document describes ISMP message type 8, version 1.

Opcode

   tet field contains the operation type of the message. ues are as
   follows:

      1 The message is a Tap request.
      2 The message is a Tap response.
      3 The message is an Untap request.
      4 The message is an Untap response.

Status

   This 2-octet field contains the current status of the tap request.
   Valid values are as follows:

      1 Switch must disable outport on untap. (DisableOutport)
      2 Switch must keep outports on untap. (KeepOutport)
      3 Probe not found this leg of spanning tree. (ProbeNotFound)
      4 Still searching for probe switch. (OutportDecisionUnknown)
      5 Unassigned. (StatusUnassigned)
      6 (reserved)
      7 (reserved)
      8 (reserved)
      9 (reserved)

   See Section 5.2.3 for details on the use of this field.

Error code

   This 2-octet field contains the response message error code of the
   requested operation. Valid values are as follows:

      1 Operation successful. (NoError)
      2 No response heard from downstream neighbor. (Timeout)
      3 Port does not exist on probe switch. (BadPort)
      4 Message invalid. (InvalidMessage)
      5 Version number invalid. (IncompatibleVersions)

Header type

   This 2-octet field contains the type of information contained in
   the header field. Currently, valid values are as follows:

   1 (reserved) 2 Header contains destination and source endstation
      MAC addresses.

Header length

   This 2-octet field contains the length of the header field.
   Currently, this field always contains a value of 12.

Direction

   This 2-octet field contains a value indicating the type of tap.
   Valid values are as follows:

   1 (reserved)
   2 Tap is bi-directional and data should be captured flowing in
      either direction over the connection.
   3 Tap is uni-directional and data should be captured only when it
      flows from the source to the destination.

Probe switch MAC

   This 6-octet field contains the physical (MAC) address of the
   switch to which the probe is attached.

Probe port

   This 4-octet field contains the logical port number (on the probe
   switch) to which the probe is attached.

Reserved

   These 12 octets are reserved.

Header

   This variable-length field contains the header that identifies the
   connection being tapped. The length of the header is stored in
   the length field.

   Currently, this field is 12 octets long and contains the 6-octet
   physical address of the connection's destination endstation,
   followed by the 6-octet physical address of the connection's
   source endstation, as shown below:

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   +   Destination MAC address    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                              |                                |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+     Source MAC address      +
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
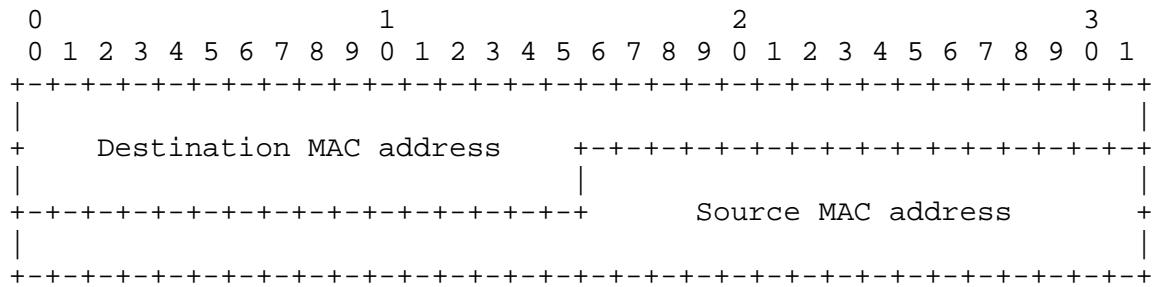
7. Security Considerations

   Requested call connections are established or denied based on the
   VLAN policy of the source and destination addresses specified within
   the packet. Section 4.4.1 discusses this process in detail.

8. References

   [RFC1700]   Reynolds, J. and J. Postel, "Assigned Numbers", STD 2,
                RFC 1700, October 1994.

   [IEEE]   "IEEE Standard 802.1d -- 1990"

   [IDvlsp]   Kane, L., "Cabletron's VLS Protocol Specification", RFC
                2642, August 1999.

   [IDhello]   Hamilton, D. and D. Ruffen, "Cabletron's VlanHello
                Protocol Specification", RFC 2641, August 1999.

9. Authors' Addresses

   Dave Ruffen
   Cabletron Systems, Inc.
   Post Office Box 5005
   Rochester, NH 03866-5005

   Phone: (603) 332-9400
   EMail: ruffen@ctron.com

   Ted Len
   Cabletron Systems, Inc.
   Post Office Box 5005
   Rochester, NH 03866-5005

   Phone: (603) 332-9400
   EMail: len@ctron.com

   Judy Yanacek
   Cabletron Systems, Inc.
   Post Office Box 5005
   Rochester, NH 03866-5005

   Phone: (603) 332-9400
   EMail: jyanacek@ctron.com

10. Full Copyright Statement

Acknowledgement