

Cabletron's VlanHello Protocol Specification
Version 4

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1999). All Rights Reserved.

Abstract

The VlanHello protocol is part of the InterSwitch Message Protocol (ISMP) which provides interswitch communication between switches running Cabletron's SecureFast VLAN (SFVLAN) product. Switches use the VlanHello protocol to discover their neighboring switches and establish the topology of the switch fabric.

Table of Contents

1. Introduction.....	2
1.1 Data Conventions.....	2
2. VlanHello Protocol Operational Overview.....	2
2.1 Neighbor Discovery.....	2
2.2 Port States.....	3
2.3 Topology Events.....	5
2.4 Timers.....	9
3. InterSwitch Message Protocol.....	9
3.1 Frame Header.....	10
3.2 ISMP Packet Header.....	11
3.3 ISMP Message Body.....	12
4. Interswitch Keepalive Message.....	13
5. Security Considerations.....	16
6. References.....	16
7. Authors' Addresses.....	16
8. Full Copyright Statement.....	17

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.

2. VlanHello Protocol Operational Overview

Switches use the VlanHello protocol to detect their neighboring switches and establish the topology of the switch fabric.

2.1 Neighbor Discovery

At initialization, each switch sends an Interswitch Keepalive message out all local ports except those which have been preconfigured such that they cannot be Network ports (see Section 2.2). Then, as each switch discovers its neighboring switches via incoming Interswitch Keepalive messages, it notifies its local topology services (see Section 2.3), which then build the topology tables for the switching fabric.

Each switch continues to send Interswitch Keepalive messages at regular intervals (currently 5 seconds). If a switch has not heard from one of its neighbors for some predetermined interval (see Section 2.4), notification is sent to all interested services and the neighboring switch is removed from the topology table.

Interswitch Keepalive messages are described in Section 4.

2.2 Port States

Each port on a switch can be in one of several different states. These states are listed below. Figure 1 shows how the port state changes within the VlanHello protocol.

- o Unknown. This is the default state of all ports at initialization.
- o Network. A port is deemed a Network port when the switch has received an Interswitch Keepalive message over the port from one of its neighbor switches. A transition to this state triggers a Neighbor Found event, notifying the local topology servers that the interface is functioning and a 2-way conversation has been established with the neighbor.

When the last switch is lost on a Network port, the state of the switch reverts to either Network Only (see next state) or to Unknown, and a Neighbor Lost event is triggered, notifying the local topology servers that the interface is no longer operational.

- o Network Only. Certain types of port interfaces are incapable of accessing user endstations and can only be used to access other switches. Such ports are deemed Network Only ports. If the last switch is lost from a port that has already been deemed a Network port, the VlanHello protocol checks the condition of the port interface. If it is the type of interface that can only be used to access other switches, the state of the port is set to Network Only. Otherwise, it reverts to Unknown.
- o Standby. A port is deemed a Standby port under the following conditions:

- o The neighbor switch on the port has a higher level of functionality and it has determined that the local switch is incompatible with that functionality. In this circumstance, the MAC entry for the local switch in the Interswitch Keepalive message received from the neighbor contains an assigned status of Incompatible.
- o The list of MAC entries in the Interswitch Keepalive message received from the neighbor switch does not contain an entry for the local switch. In this circumstance, the local switch assumes that communication with its neighbor will be one-way only.

The VlanHello protocol continues to listen for Interswitch Keepalive messages on a Standby port, but does not transmit any Interswitch Keepalive messages over the port. If a message is received that removes the condition under which the port state was set to Standby, the state of the port is set to Network.

- o Going to Access. When any packet other than an Interswitch Keepalive message is received over an Unknown port, the state of the port is changed to Going to Access and a timer is activated. If the timer expires without an Interswitch Keepalive message being received over the port, the port state changes to Access.
- o Access. A port is deemed an Access port when any packet other than an Interswitch Keepalive message has been received over the port and the Going to Access timer has expired. A port can also be administratively designated an Access "control" port, meaning the port is to remain an Access port, regardless of the type of messages that are received on it. Interswitch Keepalive messages are not sent over Access control ports.

Three other types of ports are recognized: the host management port, host data port, and host control port. These ports are designated at initialization and are used to access the host CPU. Interswitch Keepalive messages are not sent over these ports.

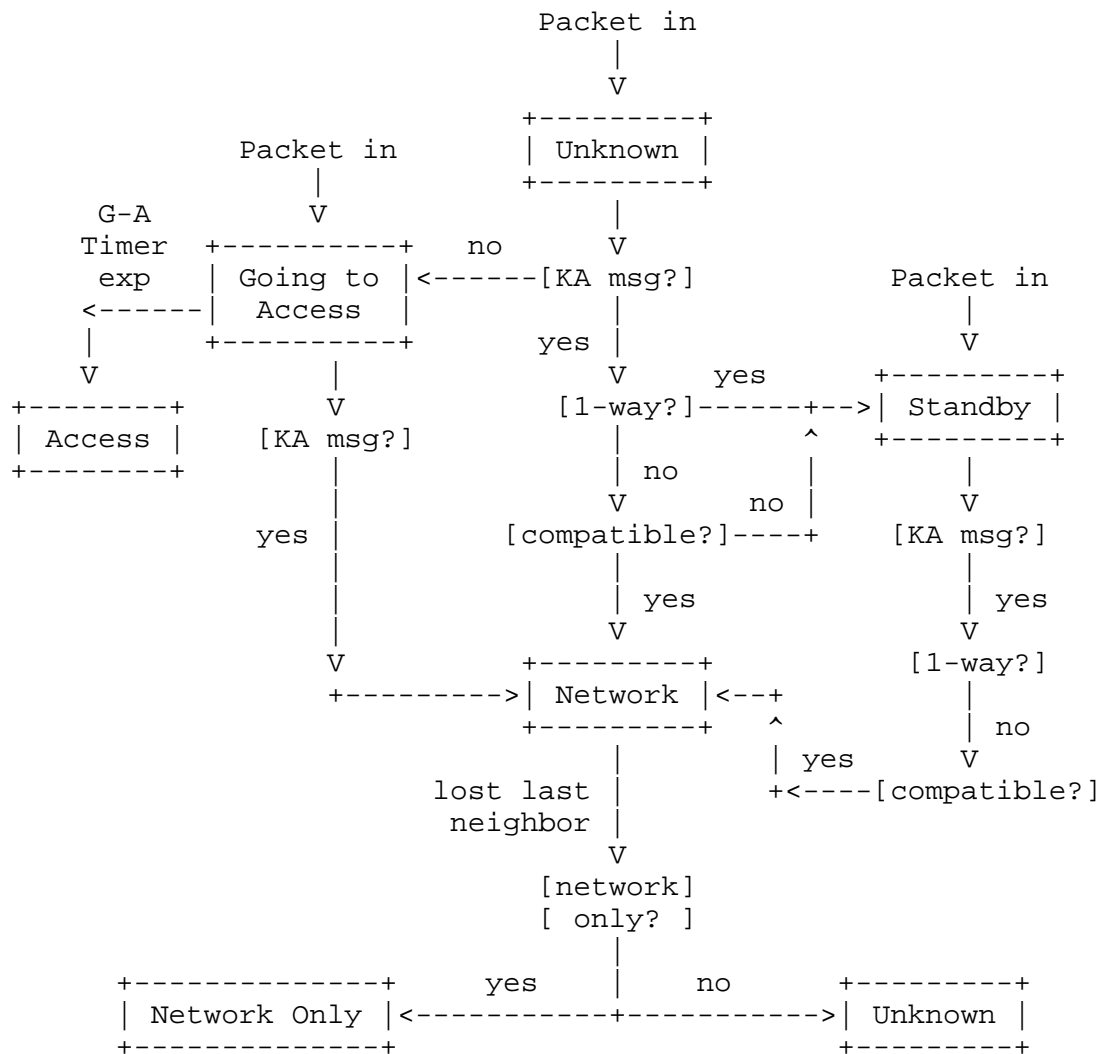


Figure 1: Port State Machine

2.3 Topology Events

When the VlanHello protocol discovers new information about the status of one of its network ports, it notifies its local topology service center so that the service center can build or modify the topology tables for the switch fabric. This notification takes the form of a system event, described in a structure known as a topology relay structure. These structures are linked in a first-in/first-out (FIFO) queue and processed by the topology servers in the order in which they were received.

- 6 The neighbor switch has been previously seen on a different port. The specified port is the previous port.
- 7 The specified port is being reassigned to another topology agent. Event is generated by the current (old) agent.
- 8 The port is looped -- that is, the Keepalive message was generated by the receiving switch.
- 9 The port is crossed -- that is, a Keepalive message was received on a port not owned by this topology agent.
- 10 The neighbor switch's functional level has changed.
- 11 The neighbor switch is running an incompatible version of the protocol.
- 12 Two-way communication with the neighbor switch has been lost.
- 13 The neighbor switch's Keepalive message sequence number has been reset, indicating the switch itself has been reset.

Delta options mask

This 4-octet field contains a bit map specifying the feature(s) gained or lost by the neighbor switch (events 2 and 3 only). Valid values are as specified for the next field, Current options mask.

Current options mask

This 4-octet field contains a bit map specifying the features of the neighbor switch. Bit assignments are as follows:

- | | |
|-------|--|
| 1 | (unused) |
| 2 | The switch is a VLAN switch. |
| 4 | The switch has link state capability. |
| 8 | The switch has loop-free flood path capability. |
| 16 | The switch has resolve capability. |
| 32 | (unused) |
| 64 | The switch has tag-based flood capability. |
| 128 | The switch has tap capability. |
| 256 | The switch has message connection capability. |
| 512 | The switch has redundant access capability. |
| 1024 | The switch is an isolated switch. |
| 4096 | The switch is an uplink. (SFVLAN V1.8 only) |
| 8192 | The switch is an uplink to core. (SFVLAN V1.8 only) |
| 16384 | The port is an uplink port. (SFVLAN V1.8 only) |
| 32768 | The port is an uplink flood port. (SFVLAN V1.8 only) |

Port number

This 4-octet field contains the logical number of the local port for which the event was generated.

Port neighbor switch identifier

This 10-octet field contains the internal identifier of the neighbor switch discovered on the port. The identifier consists of the 6-octet physical (MAC) address of the neighbor switch, followed by the 4-octet logical port number (local to the neighbor switch) on which the neighbor was discovered.

Port neighbor IP address

This 4-octet field contains the Internet Protocol (IP) address of the neighbor switch.

Neighbor chassis MAC address

This 6-octet field contains the physical (MAC) address of the chassis of the neighbor switch.

Neighbor chassis IP address

This 4-octet field contains the Internet Protocol (IP) address of the chassis of the neighbor switch.

Neighbor functional level

This 4-octet field contains the functional level of the neighbor switch, as determined by the version level of the SecureFast VLAN software under which this switch is operating. Valid values are as follows:

- 1 The switch is running a version of SFVLAN prior to Version 1.8.
- 2 The switch is running SFVLAN Version 1.8 or greater.

Topology agent

This 4-octet field contains a pointer to the topology agent that generated the event. The pointer here can reference any of the topology agents that send Interswitch Keepalive messages -- that is, any agent running the VlanHello protocol.

Next event

This 4-octet field contains a pointer to the next event relay structure in the list.

2.4 Timers

The VlanHello protocol uses three timers.

- o Send Hello timer. The Send Hello timer is used to control the interval at which Interswitch Keepalive messages are sent.
- o Aging timer. The Aging Timer is used to detect when communication with a neighboring switch has been lost.
- o Going to Access timer. The Going to Access timer is used to synchronize the transition of a port state to Access and prevent a port from being prematurely designation as an Access port during network initialization. If an Unknown port receives any packet other than an Interswitch Keepalive message, the port state is set to Going To Access. If the switch receives an Interswitch Keepalive message over that port before the timer expires, the port state is changed to Network. Otherwise, when the timer expires, the port state is changed to Access.

3. InterSwitch Message Protocol

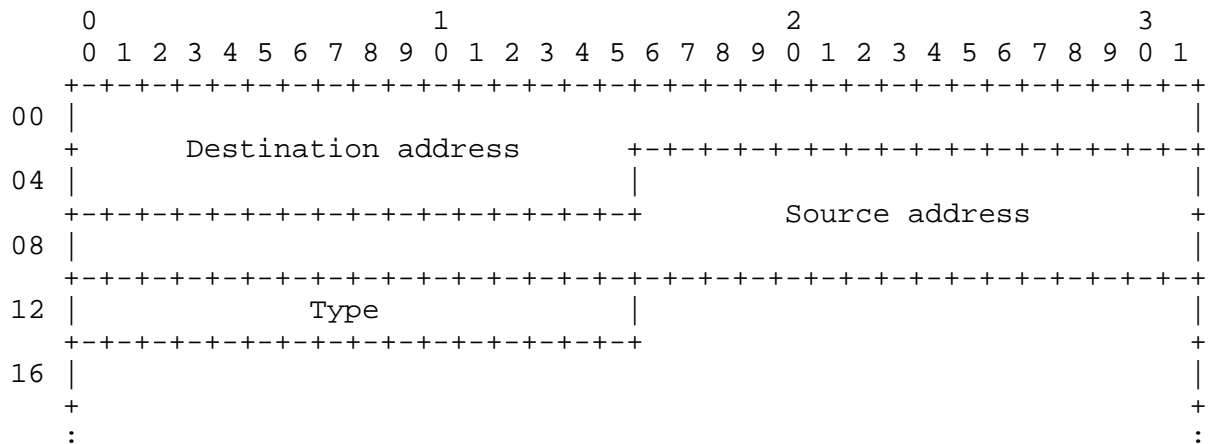
The VlanHello protocol operates as part of the InterSwitch Message Protocol (ISMP) -- part of Cabletron's SecureFast VLAN (SFVLAN) product, as described in [IDSfvlan]. ISMP provides a consistent method of encapsulating and transmitting network control messages exchanged between SFVLAN switches.

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

- o Frame header
- o ISMP packet header
- o ISMP message body

3.1 Frame Header

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



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. The destination address fields of all ISMP packets contain a value of 01-00-1D-00-00-00.

Source address

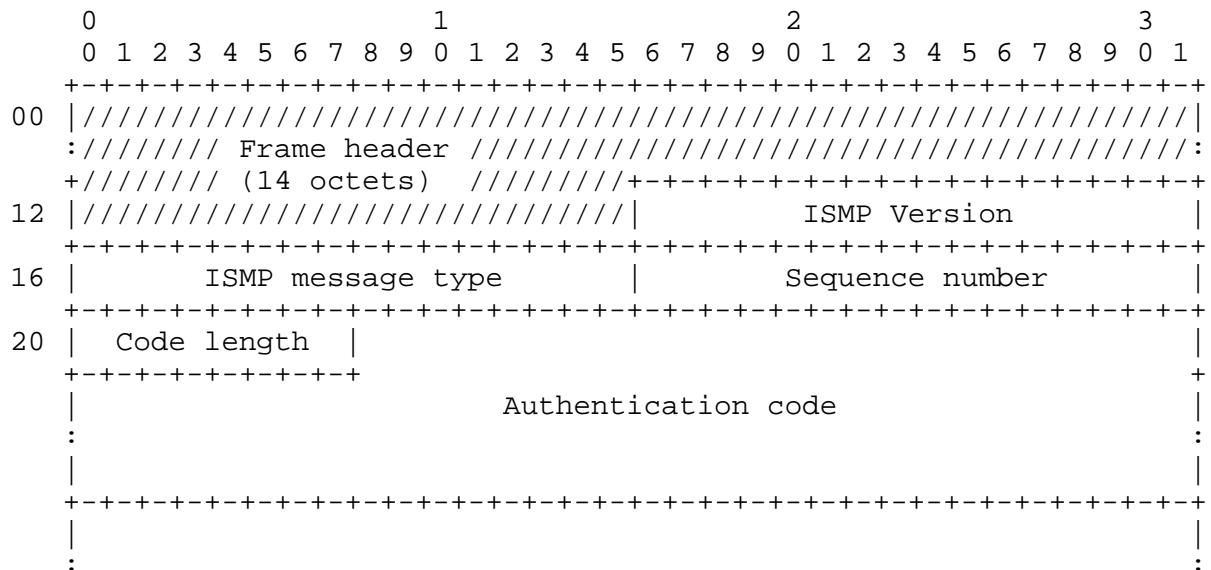
This 6-octet field contains the physical (MAC) address of the switch originating the ISMP packet.

Type

This 2-octet field identifies the type of data carried within the frame. The type field of ISMP packets contains the value 0x81FD.

3.2 ISMP Packet Header

The ISMP packet header consists of a variable number of octets, as shown below:



Frame header

This 14-octet field contains the frame header.

ISMP Version

This 2-octet field contains the version number of the InterSwitch Message Protocol to which this ISMP packet adheres. The VlanHello protocol uses ISMP Version 3.0.

ISMP message type

This 2-octet field contains a value indicating which type of ISMP message is contained within the message body. VlanHello Interswitch Keepalive messages have a message type of 2.

Sequence number

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

Code length

This 1-octet field contains the number of octets in the Authentication code field of the message.

Authentication code

This variable-length field contains an encoded value used for authentication of the ISMP message.

3.3 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.

The format of the VlanHello Interswitch Keepalive message is described in the next section.

Version

This 2-octet field contains the version number of the VlanHello protocol to which this message adheres. This document describes VlanHello Version 4.

Switch IP address

This 4-octet field contains the Internet Protocol (IP) address of the sending switch.

Switch ID

This 10-octet field contains the internal ISMP identifier of the sending switch. The identifier is generated by the sending switch and consists of the 6-octet physical (MAC) address of the switch, followed by a 4-octet value containing the logical port number over which the switch sent the packet.

Chassis MAC

This 6-octet field contains the physical (MAC) address of the chassis of the sending switch.

Chassis IP address

This 4-octet field contains the Internet Protocol (IP) address of the switch chassis.

Switch type

This 2-octet field contains the type of the switch. Currently, the only value recognized here is as follows:

- 2 The switch is an SFVLAN switch.

Functional level

This 4-octet field contains the functional level of the sending switch, as determined by the version level of the SecureFast VLAN software under which this switch is operating. Valid values are as follows:

- 1 The switch is running a version of SFVLAN prior to Version 1.8.
- 2 The switch is running SFVLAN Version 1.8 or greater.

Switch MAC address

This 6-octet field contains the base MAC address of the neighboring switch.

Assigned neighbor state

This 4-octet field contains the assigned state of the neighboring switch as perceived by the sending switch. Currently, the only value valid here is 3, indicating a state of Network

5. Security Considerations

Security concerns are not addressed in this document.

6. References

- [RFC1700] Reynolds, J. and J. Postel, "Assigned Numbers", STD 2, RFC 1700, October 1994.
- [IDSfvlan] Ruffen, D., Len, T. and J. Yanacek, "Cabletron's SecureFast VLAN Operational Model", RFC 2643, August 1999.
- [IDvlsp] Kane, L., "Cabletron's VLS Protocol Specification", RFC 2642, August 1999.

7. Authors' Addresses

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

Phone:(603) 332-9400
EMail: daveh@ctrn.com

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

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

17. Full Copyright Statement

Copyright (C) The Internet Society (1999). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

