

XXXX Working Group
Internet Draft

H.Sandick, M.Squire, B.Cain
I. Duncan, B.Haberman

[draft-sandiick-flip-00.txt](#) Nortel Networks

February 2000
Expires XXX 2000

Fast Liveness Protocol (FLIP)

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts. Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress." The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

Abstract

Networks and network applications must be robust and reliable. For many applications and services, such as packetized voice, correcting a failure must be almost instantaneous. The first step in correcting a failure is, of course, detecting that it occurred. IP routing protocols and signaling protocols as well as many application layer protocols incorporate their own keepalive mechanisms to detect failures. Typically, these protocols detect failures on the order of seconds or tens of seconds. While there are some physical and link layer technologies that inherently supply link outage detection, not all link layers do this. In order to provide for fast failure detection over any type of lower layer, an IP layer fast keepalive protocol can be used. This draft describes such a protocol for quickly detecting when a network layer interface of a peer has failed.

Table Of Contents

1 Terminology	2
2 Introduction	3

3	Protocol Overview	4
4	Neighbor Discovery and Negotiation	5
4.1	Neighbor Discovery	5

Sandick, Squire, Cain, Duncan, Haberman

1

Internet Draft

Fast Link Liveness Protocol

December 7, 1999

4.2	Parameters	5
4.3	Negotiation Rules	6
4.4	FLIP Solicitation Protocol Description	7
4.5	FLIP Advertisement Protocol Description	7
4.6	Finite State Machine (FSM) for Neighbor Adjacency	8
5	FLIP Hello Protocol	10
5.1	Protocol Description	10
5.2	Hello Message Contents	11
5.3	Finite state machine for Hello Message Exchange	11
6	Hello Exchange over NBMA links	13
7	Default Values	13
7.1	All-FLIP-Peers	13
7.2	HelloInterval	13
7.3	PeerDeadInterval	13
7.4	FLIPAdvertisementInterval	14
7.5	FLIPNewAdvertisement	14
7.6	FLIPAdvertisementDeadInterval	14
7.7	FLIPNewAdvertisementInterval	14
8	Security Considerations	14
9	Intellectual Property Considerations	14
10	References	15
A.	Setting the HelloInterval	15
B.	Formats	16
B.1	FLIP Advertisement Message	16
B.2	FLIP Solicitation Message	18
B.3	FFLIP Hello Message	19

[1](#) Terminology

Adjacency	see Neighbor Adjacency
Adjacency failure	Communication with a neighbor interfaces is no longer possible
Device	The logical entity that owns or controls one or more logical IP interfaces. A device can be a host or a router.
GroupID	Identifies a particular set of peers, e.g. routers.

Neighbor	A directly connected peer on a shared subnet.
----------	---

Sandick, Squire, Cain, Duncan, Haberman	2
Internet Draft	Fast Link Liveness Protocol
	December 7, 1999

Interface	The physical or logical interface(S) used to communicate on a particular IP subnet. The interface is defined by the IP address for the interface on that subnet. Unnumbered interfaces have the IP address "0.0.0.0"
-----------	--

Neighbor Adjacency	Established communication with peer over on a shared IP subnet defined by the respective IP addresses of the local and peer's interfaces. In the case of unnumbered interface, the interface is defined by the interface and has the IP address "0.0.0.0".
--------------------	--

Neighbor Interface	The physical or logical interface(S) that a peer uses to communicate on a particular IP subnet.
--------------------	---

Peer	A target device whose IP connectivity is monitored by another device on the same IP link layer or subnet.
------	---

Peer failure	A Peer becomes unreachable over all of its interfaces
--------------	---

PeerID	An unique identifier for a Peer
--------	---------------------------------

[2](#) Introduction

Networks and network applications must be robust and reliable. For many applications and services, such as packetized voice, correcting a failure must be almost instantaneous. The first step in correcting a failure is, of course, detecting that it occurred. IP routing protocols and signaling protocols as well as many application layer protocols incorporate their own keepalive mechanisms to detect

failures. Typically, these protocols detect failures on the order of seconds or tens of seconds. While, there are some physical and link layer technologies that inherently supply link outage detection, not all link layers do this. In order to provide for fast failure detection over any type of lower layer, an IP layer fast keepalive protocol can be used. This draft describes such a protocol for quickly detecting when a network layer interface of a peer has failed.

This document describes a general-purpose peer-to-peer neighbor adjacency protocol, which is designed to detect failures at the IP protocol layer over a variety of media. This protocol uses periodic hello messages between peers on the same IP subnet to determine "link aliveness." We feel that a fast IP layer keepalive is necessary to assist in detecting failures over a variety of lower layer protocols

Sandick, Squire, Cain, Duncan, Haberman 3
Internet Draft Fast Link Liveness Protocol December 7, 1999

that may or may not provide this capability themselves. A generic fast hello protocol provides mainly two benefits. The first is a generic protocol for neighbor discovery. The second is support for fast link failures over any media type.

We present three examples to illustrate the usefulness of our protocol. Currently, there are many "IP only" Internet service providers. These providers must rely on failure notifications from a layer-2 network operated by another carrier. In some instances the IP only carrier may wish to have its own methods and parameters for determining failures. IP routing protocols (e.g. OSPF or IS-IS Hello) for example may be used, but they typically have lower frequency thresholds at which they can operate. Another scenario that illustrates the usefulness of a generic approach is IP over WDM. In this case, the IP layer may handle all of the signaling and control aspects of the network including aspects of failure detection. A third scenario might involve a synchronization protocol between web servers on the same subnet.

3 Protocol Overview

The FFLIP protocol is designed to be generic, extensible and by definition, work over any types of media including NBMA and broadcast subnets. In order to accommodate these somewhat conflicting requirements, we divide the protocol into two "sub-protocols".

The first sub-protocol, the FLIP Advertisement protocol, is used for neighbor discovery and parameter negotiation. This protocol makes use of a new type of ICMP message. These advertisements are multicast at a low frequency and are used to discover FLIP aware neighbors on a local subnet. Once neighbors have been discovered, adjacencies are formed. Advertisements contain a list of neighbors that have been heard from; a device considers an adjacency to be formed when the device finds its

address in a neighbor's advertisement. In the case of unnumbered interfaces the list can only contain one address, "0.0.0.0". Once an adjacency is established, devices use a simple negotiation method to decide on operating parameters. This negotiation is simply an agreement on the lowest common denominator of each parameter between the parameters sent in their advertisements.

For each adjacency established with the advertisement protocol, a second sub-protocol, the FLIP Hello protocol is used as a fast keepalive between two peers. This protocol is the actual protocol used for fast link failure detection. It defines a fixed format hello message that is exchanged (unicast) between every adjacency. The FLIP hello is used to determine the status of a link and therefore operates at very high frequencies; it may be desirable to implement this protocol in a hardware implementation. For example, a device may include processing for FLIP Hellos in a line card.

In order to provide fast detection for other protocols operating in a device (e.g. routing protocols), an implementation should treat FFLIP

Sandick, Squire, Cain, Duncan, Haberman

4

Internet Draft

Fast Link Liveness Protocol

December 7, 1999

detected failures as neighbor adjacency failures. FFLIP is independent of other individual hello protocols (e.g. OSPF or IS-IS Hello), and can be used to signal link failures in a device.

[4](#) Neighbor Discovery and Negotiation

[4.1](#) Neighbor Discovery

In order to discover other FFLIP capable peers on a subnet, the FLIP Advertisement is used. The FFLIP Advertisement is similar to the ICMP Router Advertisement protocol. We define a new message type: FFLIP Advertisement, which is used by peers to advertise their support of the protocol and to advertise their configured parameters.

In order to discover all neighbors on a subnet, FFLIP Advertisements are multicast to the All-FFLIP-Peers address and contain a PeerID. Devices send these messages periodically over IP interfaces that have the FLIP protocol enabled. A neighbor adjacency, i.e. bi-directional connectivity, is established by listing neighbor interfaces that have been heard in an advertisement. Once a FLIP Advertisement is heard from a neighbor, the neighbor interface is listed in a device's own FLIP Advertisement for that interface. In the case of unnumbered interfaces the address "0.0.0.0" is added to the list. If an advertisement containing the receiving device's interface has not been received by a neighbor in [FLIPAdvertisementDeadInterval] seconds, that neighbor is declared down.

All FLIP messages use ICMP Type field (X). The FLIP Advertisement is the only currently defined message with the ICMP Code field set to zero.

[4.2](#) Parameters

FFLIP Advertisements are also used to advertise the parameters necessary for the operation of the protocol. The FLIP Advertisement contains several parameters: the HelloInterval, the PeerDeadInterval, the PeerID of the transmitting device, The GroupID and the list of neighbor interfaces that the transmitting device has heard from.

The HelloInterval and the PeerDeadInterval are used to set the operational parameters of the FLIP Hello message. The HelloInterval indicates how often FFLIP Hello messages will be sent and is measured in milliseconds (ms). For example, if the value is 3 then the advertising device would send the Hello message at least every 3 ms. The PeerDeadInterval indicated how long a device should wait (from the last Hello) before declaring a neighbor failure. PeerDeadInterval is specified in milliseconds. This value MUST be larger than the HelloInterval and should be at least 3 times the value of the HelloInterval. (NOTE: Should there be a minimum multiple, say 3 time)

Sandick, Squire, Cain, Duncan, Haberman	5
Internet Draft Fast Link Liveness Protocol	December 7, 1999

The PeerID is the globally unique identifier assigned to that device. A device must use the same PeerID in all FLIP Advertisements. The PeerID allows receiving devices to correlate multiple interfaces with the peer that owns them.

The GroupID identifies the group of peers that this advertisement is targeted for.

The list of neighbor interfaces is used to establish neighbor adjacencies. The list contains the neighbor interfaces that a device has received advertisements from over a particular IP link layer. In the case of unnumbered interfaces the list can only contain one address, "0.0.0.0". If a device receives an advertisement containing its interface, the device assumes that connectivity to that neighbor has been established. In the case of an unnumbered interface the IP address "0.0.0.0" indicates the device's advertisement has been received.

Note: A device MAY use the HelloInterval and PeerDeadInterval for all of its interfaces, or it MAY use different ones over different

interfaces. These parameters MUST be configurable for each device, and the timer intervals SHOULD be configurable independently on each interface. The means by which a device configures the values for its initial operation are outside this specification.

4.3 Negotiation Rules

It is possible that two neighbors will be configured with different values for their operating parameters. Values must be agreed upon before the hello messaging can begin. FLIP uses a simple negotiation scheme that is simply lowest common denominator. FLIP negotiation is on a per adjacency basis, NOT a subnet basis. The rules for negotiation are:

- (a) If the HelloInterval values are different, then the parameters (HelloInterval and PeerDeadInterval) from the neighbor associated with the larger HelloInterval are selected to be used by both neighbors.
- (b) Otherwise, if the PeerDeadInterval values are different, then the parameters associated with the neighbor advertising the larger PeerDeadInterval are selected to be used by both neighbors.

The larger values are selected (instead of the smaller values) to accommodate environments where one of the devices cannot operate at the same parameters as the other device. In this case, the faster device must in a sense 'slow down' for the slower device.

Sandick, Squire, Cain, Duncan, Haberman 6
Internet Draft Fast Link Liveness Protocol December 7, 1999

After a negotiation is applied, a device does not change the parameters that are sent in its advertisement. The parameters sent in an advertisement are a device's configured parameters and may or may not be the actual parameters used with a given adjacency. The actual parameters used with an adjacency are those which are computed from the negotiation rules.

Each device SHOULD apply a small jitter factor to the HelloInterval, rendering the actual parameter to between 75% and 100% of the selected value. The same jitter should be applied to the PeerDeadInterval

4.4 FLIP Solicitation Protocol Description

The FLIP protocol is enabled per IP interface; FLIP protocol state is per interface, per neighbor. When a FLIP enabled interface first become enabled it sends a FLIP Solicitation message to the AllFLIPDevices address. The address is sent FLIPSolicitationReSend

times FLIPSolicitationInterval apart. When a device receives a FLIP Solicitation message it unicasts a FLIP Advertisement to the source address of the Solicitation message. The Advertisement contains the source address of the Solicitation in its list of neighbor interfaces. The FLIP Solicitation message uses the same format as the FLIP Advertisement but some parameters are set differently.

A valid FLIP Advertisement is one which:

1. IP header and ICMP checksums pass
2. ICMP Type = X
3. ICMP Code = 1
4. HelloInterval > 0
5. PeerDeadInterval > HelloInterval
6. Valid Address Family
7. Non-zero PeerID
8. Neighbor Interfaces = 0

[4.5](#) FLIP Advertisement Protocol Description

After the FLIP Solicitation message is sent FLIP Advertisements are sent every FFLIPAdvertisementSeconds with a small amount of jitter. When a device receives a FLIP Advertisement from a neighbor, it lists the neighbor interface in its own FLIP advertisements for that interface. If a device receives an advertisement containing its own interface in one of the neighbor fields and it has listed that neighbor in its own advertisement, a FLIP adjacency is established. If an advertisement containing the receiving device interface has not been received from a neighbor in FLIPAdvertisementDeadInterval seconds, then that neighbor is removed from subsequent advertisements (for that interface) and the adjacency is considered down.

Once a FLIP adjacency is established, a device applies the negotiation rules in [section 4.3](#) on a per neighbor basis. They immediately begin

the FLIP Hello protocol described in [section 5](#). If there is an adjacency failure, then the FLIP Hello protocol is ceased for that neighbor.

When a FLIP Advertisement is received, the source IP address of the advertisement should be stored with all neighbor state. This address is used in the FLIP Hello protocol (see [section 5](#)). In the case of unnumbered interface the source address is "0.0.0.0".

A valid FLIP Advertisement is one which:

1.
IP header and ICMP checksums pass
2.
ICMP Type = X
3.
ICMP Code = 0
4.
HelloInterval > 0
5.
PeerDeadInterval > HelloInterval
6.
Valid Address Family
7.
Non-zero PeerID

If a device chooses to modify its FLIP parameters for an interface, it MUST multicast a FLIP Advertisement with the new parameters at least FLIPNewAdvertisement times within FLIPNewAdvertisementInterval. When a FLIP Advertisement is received from a neighbor, which would cause a renegotiation, a node should immediately unicast an advertisement of its own on the interface to that neighbor. If a renegotiation occurs, the new parameters should immediately be applied to the FLIP Hello protocol.

If a device decides that a particular neighbor has not adjusted its operating parameters sufficiently, then the device MAY unicast an FLIP Advertisement to that neighbor. If the parameters are sufficiently different, the neighbor's hello protocol will fail and it will have to re-discover this device and its new operating parameters.

[4.6](#) Finite State Machine (FSM) for Neighbor Adjacency

This FSM formally details how neighbor adjacency state is created, maintained and deleted. We adopt the notation of X(y), where X is the new state transitioned to after action y has been applied. A "-" means that no state transition or action is performed. This FSM is the state machine for the FLIP adjacency protocol. Behavior global to the interface such as the periodic sending of FLIP Advertisements are not part of this FSM.

Input	State		
	0	1	2
A	$-(n)$	$0(n)$	$0(d)$
B	$1(n)$	$-(n)$	$1(d)$
C	$2(b)$	$2(b)$	$-(a)$
D	NA	NA	$-(c)$
E	NA	NA	$0(e)$

States

Init (0): This is the initial state at which the protocol begins. In this state, FLIP Advertisements are sent; No advertisements have been received for the neighbor.

Half (1): This state is transitioned to after an advertisement is received from the neighbor. But bi-directional connectivity has not yet been established, i.e. A FLIP Advertisement containing the receiving devices interface has not been received.

Full (2): When a receiving device sees its interface in its neighbor's advertisement then bi-directional connectivity has been established and this state is established. In this state, an adjacency is established and the FLIP Hello protocol should begin operation over this adjacency.

Inputs

A - FLIP Reset

- B - Advertisement from neighbor without receiving device's interface present
- C - Advertisement from neighbor with receiving device's interface present
- D - Advertisement received with parameters changed (i.e. re-negotiation)
- E - FLIPAdvertisementDeadInterval expires

Actions

Sandick, Squire, Cain, Duncan, Haberman 9
 Internet Draft Fast Link Liveness Protocol December 7, 1999

- a - Reset FLIPAdvertisementDeadInterval timer
- b - Multicast a FLIPAdvertisement on interface having added the neighbor interface address to the list of the list of Neighbor Interfaces; set FLIPAdvertisementDeadInterval timer; reset the FLIPAdvertisementInterval timer; adjacency established; begin FLIP Hello protocol
- c - Update operational parameters for FLIP Hello message exchange; unicast FLIPNewAdvertisements number of FLIP advertisements to neighbor; reset the FLIPAdvertisementInterval timer; reset FLIPAdvertisementDeadInterval;
- d - Stop FLIPAdvertisementDeadInterval timer; multicast a FLIPAdvertisement on interface; reset the FLIPAdvertisementInterval timer; stop FLIP Hello exchange;
- e - stop FLIP Hello exchange
- n - no op
- NA - Not applicable

5 FLIP Hello Protocol

5.1 Protocol Description

The FLIP Hello protocol is used as a fast keepalive protocol for determining the status of neighbor adjacencies. This protocol is designed to operate at high frequencies for quickly detecting link and device failures. An implementation may choose to implement this protocol in hardware in order to achieve the high frequency rates required to send and receive hello messages.

The FLIP Hello is a simple fixed format message exchanged by devices. After a neighbor adjacency has been established via the FLIP Advertisement protocol and parameter negotiations have resolved, a device sends unicast FLIP Hello messages to the neighbor interface. The Hellos MUST be sent every HelloInterval. Each device SHOULD apply a small jitter factor to the HelloInterval, rendering the actual parameter to between 75% and 100% of the selected value. The same jitter should be applied to the PeerDeadInterval. The jitter should be

calculated once at the beginning of the hello exchange and used until the neighbor adjacency has failed.

FLIP Hello messages contain a bit for detecting bi-directional connectivity. If the device has received a hello message from this neighbor adjacency within the last PeerDeadInterval, it sets the HelloHeard bit in the hello messages sent back to that neighbor interface. This bit will set as long as a Hello message has been received from the neighbor interface within the PeerDeadInterval. If the received message has the HelloHeard bit set, then there is successful bi-directional communication with this neighbor, and the neighbor interface is considered to be active.

Sandick, Squire, Cain, Duncan, Haberman 10
Internet Draft Fast Link Liveness Protocol December 7, 1999

A device must receive a Hello message from a neighbor interface with the HelloHeard bit set within the PeerDeadInterval or, it considers that neighbor adjacency to be down. If connectivity is lost, a FLIP protocol implementation should signal a neighbor adjacency failure. How this information is propagated within a device is an implementation issue and beyond the scope of this document.

[5.2](#) Hello Message Contents

The FLIP Hello message is a fixed format message, designed to be lightweight and therefore simple enough to be processed in hardware with minimal effort. Therefore, it carries limited information.

The FLIP Hello message is carried in an IP packet with IP protocol type [X](#). The source address in the IP header is the address that the sending device uses for the interface that it is transmitting on. The destination address is the source address from the neighbor's FLIP Advertisement. In the case of unnumbered interfaces the source and destination addresses are both set to "0.0.0.0". For Differentiated Service aware networks the DS field will be set in accordance with the Differentiated Services architecture to CS6, b'110xxx [[RFC2476](#)].

The body of the packet contains a 4-byte bit field. The first bit is the HelloHeard bit, and indicates that the sender has received a hello message from the this neighbor within the PeerDeadInterval. All other bits are reserved at this time. A valid FLIP Hello is one which:

1. IP Protocol Type = X
2. ??

[5.3](#) Finite state machine for Hello Message Exchange

This finite state machine specifies the protocol for a hello exchange with one neighbor interface.

Input	State			
	0	1	2	3
A	1(a)	-(n)	-(n)	-(n)
B	-(n)	3(c)	3(c)	-(f)
C	-(n)	2(n)	-(n)	2(d) -or- -(n)
D	NA	NA	NA	1(e)
E	-(n)	0(g)	0(g)	0(e)
F	NA	-(a)	-(b)	-(b)

-----+-----+-----+-----+-----+

States

- [0](#) - Reset
- [1](#) - Neighbor Adjacency established
- [2](#) - Hello received with HelloHeard bit Off
- [3](#) - Hello received with HelloHeard bit On:
Bi-directional connectivity established

Inputs

- A _ Neighbor Adjacency established
- B - Hello HelloHeard bit On
- C - Hello HelloHeard bit Off
- D - PeerDeadInterval Pops
- E _ Neighbor Adjacency failed
- F - HelloInterval pops

Sandick, Squire, Cain, Duncan, Haberman

12

Internet Draft Fast Link Liveness Protocol

December 7, 1999

Actions

- a - Send hello with HelloHeard bit off; reset HelloInterval timer
- b - Send hello with HelloHeard bit on; reset HelloInterval timer
- c - Indicate two-way connectivity has been established;
reset PeerDeadInterval timer
- d - Indicate two-way connectivity has been
lost; stop PeerDeadInterval timer
- e - Indicate two-way connectivity has been
lost. Stop PeerDeadInterval timer
timer; stop HelloInterval timer
- f - Reset PeerDeadInterval timer
- g - Stop HelloInterval timer
- f - Reset PeerDeadInterval timer
- n - Do nothing
- NA- Not applicable

[6](#) Hello Exchange over NBMA links

TDB

[7](#) Default Values

The following describes the values and ranges of the variables and timers used in the FLIP protocol.

[7.1](#) All-FLIP-Peers

The All-FLIP-Peers multicast address has the following possible values:

- o IPv4 _ 224.0.0.12
- o IPv6 _ FF02:0:0:0:0:0:0:C

[7.2](#) HelloInterval

The HelloInterval indicates how often FLIP Hello Messages are transmitted on an interface. The lower bound on the HelloInterval is 1 ms. Default value: 3 ms.

[7.3](#) PeerDeadInterval

Sandick, Squire, Cain, Duncan, Haberman 13
Internet Draft Fast Link Liveness Protocol December 7, 1999

The PeerDeadInterval specifies the amount of time a device should wait, since the last Hello message, before declaring a neighbor down. The PeerDeadInterval has a lower bound of 3*HelloInterval and an upper bound of 2^32 ms. Default value: 12 ms.

[7.4](#) FLIPAdvertisementInterval

The FLIPAdvertisementInterval controls how often FLIP Advertisements are sent. The lower bound on the interval is 3 seconds. The upper bound is 1800 seconds. Default value: 600 seconds.

[7.5](#) FLIPNewAdvertisement

The FLIPNewAdvertisement variable specifies how many advertisements must be sent upon the activation of an IP interface. The lower bound is 2 and the upper bound is 10. This variable should be tuned to

correspond to the loss characteristics of the media. All devices on an IP subnet should have this variable configured the same. Default value: 3.

[7.6](#) FLIPAdvertisementDeadInterval

The FLIPAdvertisementDeadInterval is the amount time that can elapse without an advertisement being received with the receiving devices IP address in the list of neighbors. The lower bound is [FLIPAdvertisementInterval]. Default value: 2*FLIPAdvertisementInterval.

[7.7](#) FLIPNewAdvertisementInterval

The FLIPNewAdvertisementInterval specifies the timeframe in which the [FLIPNewAdvertisement] advertisements must be sent. Default value: [FLIPNewAdvertisement] seconds.

[8](#) Security Considerations

TBD.

[9](#) Intellectual Property Considerations

Nortel Networks may seek patent or other intellectual property protection for some or all of the technologies disclosed in this document. If any standards arising from this document are or become protected by one or more patents assigned to Nortel Networks, Nortel intends to disclose those patents and license them on reasonable and non-discriminatory terms.

Sandick, Squire, Cain, Duncan, Haberman 14
Internet Draft Fast Link Liveness Protocol December 7, 1999

[10](#) References

- [RFC1256] Deering, S., "ICMP Router Discovery Messages", [RFC 1256](#), September 1991.
- [IANA AFN] IANA Address Family Number, <http://www.isi.edu/in-notes/iana/assignments/address-family-numbers>.
- [[RFC2474](#)] Nichols, K., Blake, S., Baker, F. and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", [RFC 2474](#), December 1998.

Appendices

A.

Setting the HelloInterval

There are three factors that affect the bandwidth utilization of FFLIP Hello messages. First, is the HelloInterval or the frequency with which the messages are sent. The second factor is the speed of the link. The third factor is how many neighbor adjacencies are maintained over one link (for multi-access links only). Let t represent the number of milliseconds between hello packets, m represent the Mbps available to L2 data over that link, p represent the size of the hello packet in bits at Layer 2, and n represent the number of devices sharing the bandwidth over that link. The general formula to determine the percentage of the Layer 2 bandwidth used by the hello messages between the n peers over that link:

$$\text{Bandwidth percentage} = \frac{pn(n-1)}{10tm}$$

For full duplex links, this percentage should be divided by two.

The following table shows how much bandwidth a 64-byte Hello message uses over various media speeds and hello intervals. Full duplex communication is assumed for a single pair of neighbors. Note: 64-byte size packet is selected because it is the minimum size on Ethernet subnets.

millisecond = ms

Percentage of link for 64-byte packet
transmitted every time period

 10Mb 100Mb 1Gigb
+-----+-----+-----+-----+

3ms	64 B	1.7%	.17%	.017%	
	+-----+	+-----+	+-----+	+-----+	+
6ms	64 B	.85%	.085%	.0085%	
	+-----+	+-----+	+-----+	+-----+	+
9ms	64 B	.57%	.057%	.0057%	
	+-----+	+-----+	+-----+	+-----+	+
12ms	64 B	.43%	.043%	.0043%	
	+-----+	+-----+	+-----+	+-----+	+
		.			
		.			
		.			

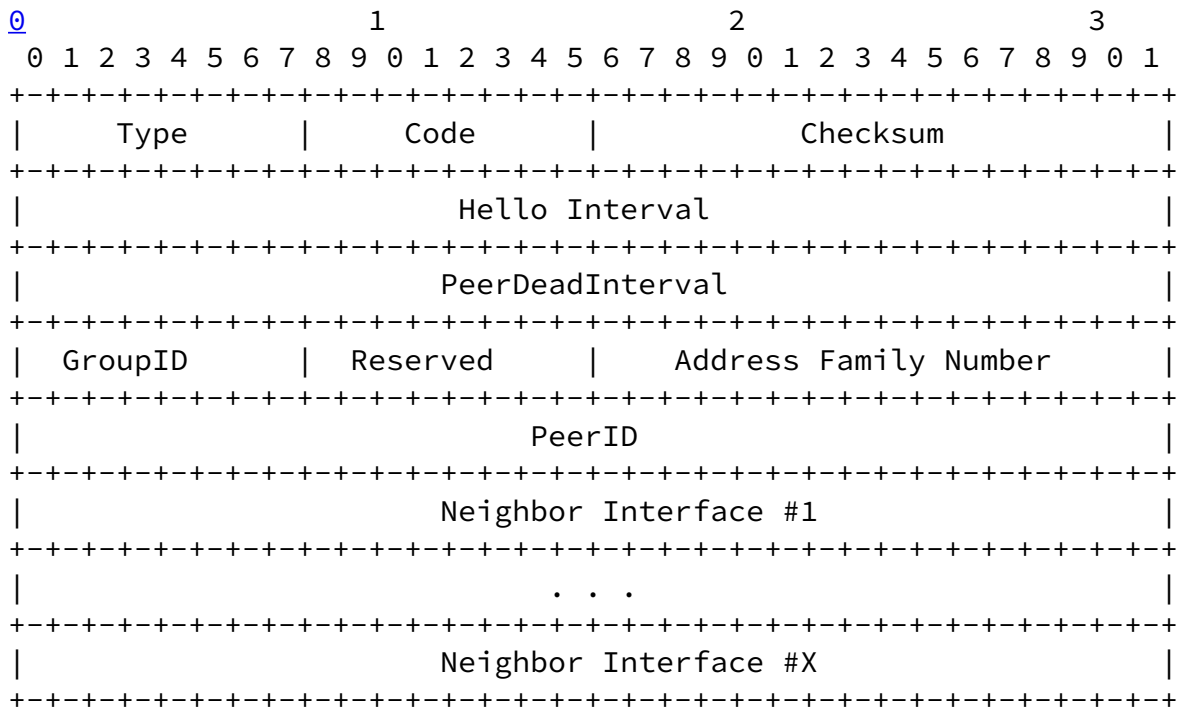
These numbers must be multiplied for each neighbor adjacency maintained over an interface. So for example if there are 10 peers over a 100 Mb interface sending hellos every 9ms would use .57% over switched full duplex and 1.1 % over switched half duplex and 12.54% over a shared media.

There are several factors to consider when setting the HelloInterval and the PeerDeadInterval: speed of the link, quality of the link, timer precision, number of neighbors, etc. Additionally, if there is a self healing subnet technology in the middle of the subnet, the timers may set with a large enough interval to allow the subnet to fix the link fault before FFLIP events determine a neighbor adjacency failure to be down.

B.

Formats

[B.1](#) FLIP Advertisement Message



IP Fields:

Source Address	An IP address belonging to the interface from which this message is sent.
Destination Address	The configured multicast Advertisement Address or the IP address of a neighboring host.
Time-to-Live	1

ICMP Fields:

Type	X
Code	0 {FLIP Advertisement}
Checksum	The 16-bit one's complement of the one's complement sum of the ICMP message, starting with the ICMP Type. For computing the checksum, the checksum field is set to 0.
HI	HelloInterval _ This interval is the time period between successive FLIP Hello messages. Measured in milliseconds

PDI	PeerDeadInterval	_	If this device does not receive a FFLIP Hello message from the neighbor interface within this time period, it will consider the link to be down. Measured in milliseconds.
GroupID	Identifies a particular set of peers, e.g. routers.		
Reserved	must be all zeros		
Address Family Number	0	Reserved	
	1	IP (IP version 4)	
	2	IP6 (IP version 6)	
PeerID	The sending peer's globally unique IP identifier. Length is per Address Family.		
Neighbor Interface X	A list of all source IP addresses of all FLIP Advertisements that have been heard on this interface. Length of each address is per Address Family.		

B.2 FLIP Solicitation Message

The message is the same format as the FLIP Advertisement. The fields are set the same with the following exceptions:

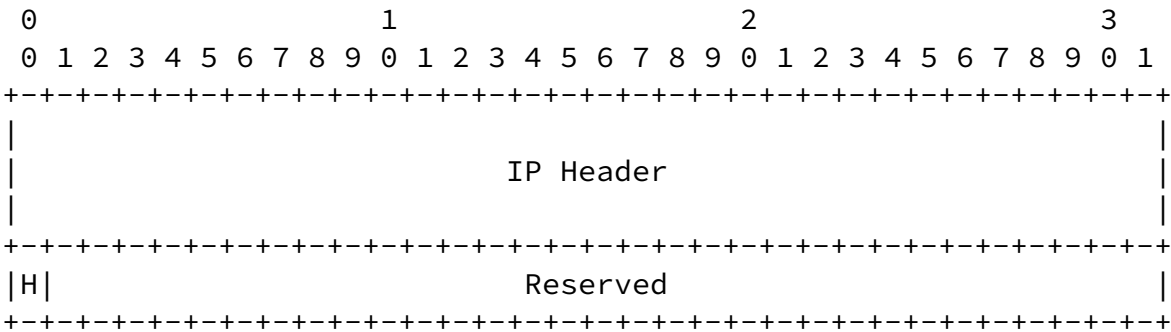
IP Fields:

Destination Address	The configured multicast Advertisement
---------------------	--

ICMP Fields:

Type	X
Code	1 {FLIP Solicitation}
Neighbor Interface X	A list of all source IP addresses of all FLIP Advertisements that have been heard on this interface. The field should be empty and MUST be ignored by the

B.3 FFLIP Hello Message



IP Fields:

DSCP	This is the Differential Services Code Point. If supported set to CS6, b'110xxx		
Source Address	An IP address belonging to the interface from which this message is sent.		
Destination Address	The IP address of a neighboring host.		
Time-to-Live	1		
IP Protocol Type	FFLIP Protocol (type X)		

Hello Fields:

H Bit	HelloHeard bit. The transmitting device has received a hello message from the target of this message.
Reserved	set to all zeros

Authors' Addresses

Hal Sandick
Nortel Networks
[4309](#) Emperor Blvd

Suite 200
Durham, NC 27703
email: hsandick@nortelnetworks.com

Matt Squire
Nortel Networks
[4309](#) Emperor Blvd
Suite 200
Durham, NC 27703

Sandick, Squire, Cain, Duncan, Haberman
Internet Draft Fast Link Liveness Protocol

19
December 7, 1999

email: msquire@nortelnetworks.com

Brad Cain
Nortel Networks
[600](#) Technology Park
Billerica, MA 01821
Email: bcain@nortelnetworks.com

Brian Haberman
Nortel Networks
[4309](#) Emperor Blvd
Suite 200
Durham, NC 27703
email: haberman@nortelnetworks.com

Ian Duncan
Nortel Networks
[130](#) COLONNADE ROAD
NEPEAN, ONTARIO K2E 1B6
email: iduncan@nortelnetworks.com

