NICC ND 1611 V2.2.3 (2009-01)

NICC Document

Multi-Service NGN Interconnect Common Transport

Network Interoperability Consultative Committee,
Ofcom,
2a Southwark Bridge Road,
London,
SE1 9HA.

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to NICC. Pursuant to the NICC IPR Policy, no investigation, including IPR searches, has been carried out by NICC. No guarantee can be given as to the existence of other IPRs which are, or may be, or may become, essential to the present document.

Foreword

This NICC Document (ND) has been produced by NICC.

Introduction

This specification forms part of the Next Generation Network, Multi-Service Interconnect Release Structure and should be read in conjunction with the associated releases of the standard Next Generation Networks, Release Definition [1].

1 Scope

The present document defines the common transport function for supporting multi-service interconnects between Next Generation Networks within the UK. This document defines the functional architecture for the common transport and specifies the protocols and interfaces that support TDM, ATM and managed IP type services on the same transmission.

2 References

For the particular version of a document applicable to this release see ND1610 [1].

2.1 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For non-specific references, the latest edition of the referenced document (including any amendments) applies.

[1] ND1610 Next Generation Networks, Release Definition

[2] SR 001 262 ETSI drafting rules Section 23:- Verbal Forms For The Expression Of Provisions

[3] ND1125 SDH Interconnect Between UK Licensed Operators, Technical Recommendations

[4] ND1122 Interconnect Between UK Licensed Operators, Based Upon Permanent ATM Connections, Technical Recommendation

[5] IEE 802.1ah Provider Backbone Bridges, Draft

[6] IEE 802.3 Local and metropolitan area networks--Specific requirements--Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, 2002

[7] IEE 802.1Q Virtual Bridged Local Area Networks, 2003

[8] IEEE 802.3ad Aggregation of Multiple Link Segments (Now part of 802.3), 2002

[9] IEEE 802.1ad Draft Standard for Local and Metropolitan Area Networks-- Virtual Bridged Local Area Networks-- Amendment 4: Provider Bridges, 2005

[10] IEEE 802.1D Virtual Bridged Local Area Networks, 2003

[11] ITU-T G.811Timing characteristics of primary reference clocks, 1997-09

[12] IEEE 802.3ae Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications, 2005

[13] IEEE 802.1w Part 3: Media Access Control (MAC) Bridges: Rapid Configuration (Now part of 802.1d - Media Access Control (MAC) Bridges 2004), 2001

[14] IEEE 802.1ag Draft Standard “Connectivity Fault Management”, 2005

[15] ITU-T G.7041 Generic Framing Procedure, 2005-08

[16] ND1614 Management of the General Connectivity of PSTN/ISDN Service Interconnect for UK NGNs

[17] ND1612 Generic IP Connectivity for PSTN / ISDN Services between UK Next Generation Networks

[18] IETF draft-ietf-bfd-v4v6-1hop-08.txt BFD for IPv4 and IPv6 (Single Hop), Draft See Annex E

[19] IEEE 802.1Qay Provider Backbone Bridge Traffic Engineering, Draft

[20] IETF RFC792 Internet Control Message Protocol, Sep 1981

[21] ITU-T G.7042 Link capacity adjustment scheme (LCAS) for virtual concatenated signals, 2001-11

[22] IETF RFC3682 “The Generalized TTL Security Mechanism”, Feb 2004.

3 Definitions, symbols and abbreviations

The key words “**shall**”, “**shall not**”, “**must**”, “**must not**”, “**should**”, “**should not**”, “**may**”, “**need not**”, “**can**” and “**cannot**” in this document are to be interpreted as defined in the ETSI Drafting Rules [2].

3.1 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ATM	Asynchronous Transfer Mode
BFD	Bidirectional Forwarding Detection
CBR	Constant Bit Rate
CC	Connectivity Check
CFM	Connectivity Fault Management
CP	Communications Provider
CSFF	Client Signal Fail Frame
CTF	Common Transport Function
CTFI	Common Transport Function Interface
ETSI	European Telecommunication Standards Institute
FDI	Forward Defect Indicator
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical & Electronic Engineers
IP	Internet Protocol
IPG	Inter-Frame Gap
ITU-T	International Telecommunication Union - Telecoms
GFP	Generic Framing Procedure – ITU-T G.7041[15]
LCAS	Link Capacity Adjustment Scheme
LoS	Loss of Signal
MAC	Medium Access Control
MTU	Maximum Transfer Unit
NGN	Next Generation Network
OAM	Operations Administration and Maintenance
PCP	Priority Code Points
PSTN	Public Switched Telephone Network
QoS	Quality of Service
SDH	Synchronous Digital Hierarchy
SFD	Start of Frame Delimiter
TDM	Time Division Multiplex
VC	Virtual Circuit
VLAN	Virtual Local Area Network

4 Common Transport Function (CTF)

The NGN interconnect that supports multiple services is built around a common, multi-purpose transport function that provides a number of transport capabilities via two transmission technologies. This common NGN Interconnect transport function is represented in Figure 1 which shows the transport function (fB1) offering the following transport capabilities:

a) Internet Protocol transport
b) Ethernet transport
c) ATM transport
d) TDM transport

The transport function offers some or all of the above capabilities via the following transmission technologies:-

i) SDH (Figure 2)
ii) Ethernet Physical (Figure 3)

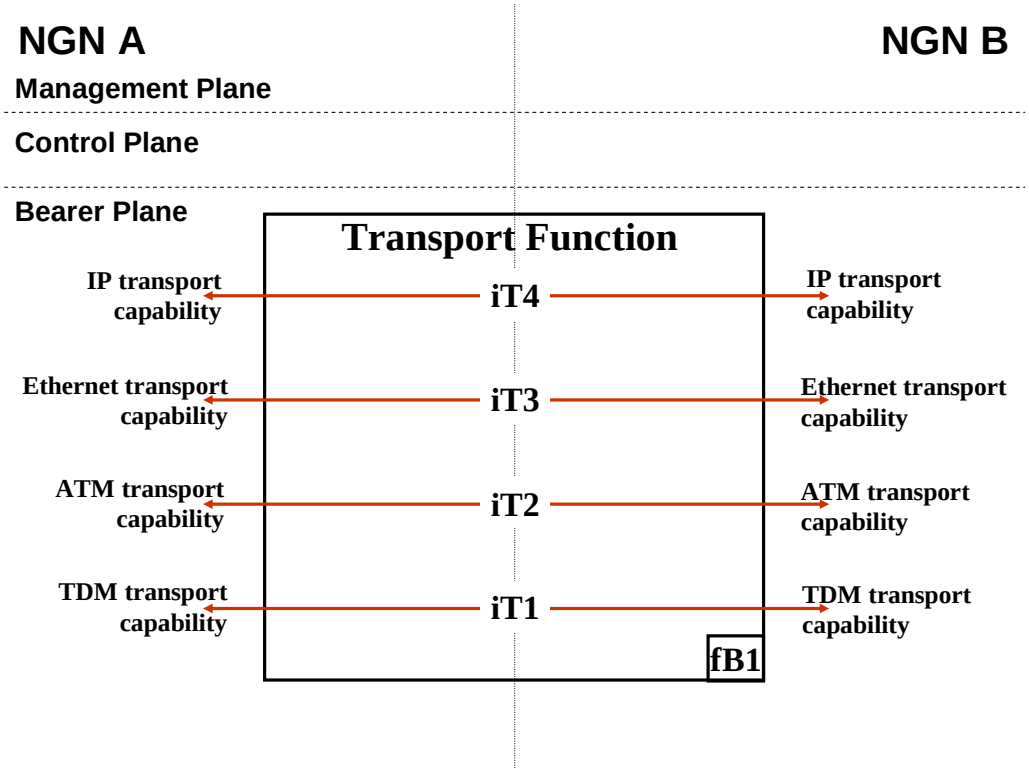


FIGURE 1: COMMON TRANSPORT FUNCTION

Table 1 gives the client network transport capability / server physical transmission technology compatibility matrix and the associated attributes.

Client Network Transport Capability	Server Physical Transmission Technology	Capability Attributes
TDM	SDH only	Bandwidth partitioned by SDH virtual container
ATM	SDH only	Bandwidth partitioned by SDH virtual container
Ethernet	SDH	Bandwidth partitioned by SDH 'n x virtual containers'. Ethernet encapsulated by GFP Ethernet VLANs each configured with fixed bandwidth, the total fitting within the underlying 'n x virtual containers' bandwidth.
Ethernet	Ethernet Physical	Ethernet VLANs each configured with fixed bandwidth, the total fitting within the underlying Ethernet physical bandwidth.
Internet Protocol	SDH or Ethernet Physical	IP service partitioned by underlying Ethernet VLAN and associated fixed bandwidth allowing IP services with overlapping IP addresses on separate VLANs.

TABLE 1: CLIENT NETWORK TRANSPORT CAPABILITY / SERVER PHYSICAL TRANSMISSION TECHNOLOGY COMPATIBILITY AND ATTRIBUTES

The transport function provides the physical termination of one or more of the transmission systems, to one or more NGNs. It also provides the framing of the transmission bit streams to provide separate virtual pipes called 'trails'.
A trail is a topological construct, which **shall** be monitored, that exists between a single (trail termination) source point and a single (trail termination) sink point, and follows a fixed network routing between these points over the lifetime of the trail (under failure-free conditions). Trails **shall** have resource (bandwidth) assigned to them. Strictly trails **shall** not reorder packets. Trails can only exist in either connection-oriented connection switched or connection-oriented packet switched mode networks. Trails do not exist in connectionless packet switched mode networks. Note that there is a 1:1 relationship between a connection and a trail in the point to point case. From the service perspective the transport layer provides trails which have the following characteristics:-

a) Separacy at the IP protocol level, i.e. overlapping IP address spaces and packet marking schemes **may** be used on separate trails. Reachability isolation is provided by separate trails.
b) Separacy at the framing level, i.e. support of non-IP services e.g. ATM. Payload format isolation is provided by separate trails.
c) Static and policed bandwidth allocation to transport trails. Resource and performance isolation is provided by separate trails.

4.1 CTF Characteristics

The following are characteristics of the Common Transport Function (CTF) and its interfaces, as shown in Figure 1:-

a) The CTF supports multiple services. The services are clients of the CTF. The CTF **should** support multiple trails per Common Transport Function Interface (CTFI) physical port.
b) The CTF **need not** offer resilient transport.
    i. The physical transmission used by the CTF **may** offer protection.

ii. The service interconnect **may** offer a resilience mechanism e.g. the Service interconnect **may** use multiple CTFIs.

c) This Common Transport Function only provides point-to-point connectivity between communications providers.

d) The CTF **should** use one of two transport types:

Ethernet with associated bandwidth policy enforcement, where a VLAN tag can be used as a form of 'service instance identifier'. (Note - in its normal sense a VLAN is a restricted broadcast domain and is not a trail under the strictest definition of the term, but is functionally equivalent for the purposes of this interconnect on a point-to-point basis only if resource (bandwidth) is assigned to the VLAN and is monitored with OAM. A VLAN can therefore serve as an instance of a trail in the context of this interconnect specification.) The term “VLAN trail” will be used in this specification where the VLAN trail is a monitored point-to-point construct with reserved bandwidth.

SDH Virtual Containers.

e) The CTFI encapsulation and its labelling scheme **shall** transparently transport the services.

f) It **shall not** be possible for a communications provider to impersonate another communications provider by using incorrect labels.

g) Trails & labels (including VLAN tags) **must** be statically provisioned (i.e. not dynamically signalled). LCAS [21] **may** be used by the CTF by bilateral agreement to adjust the capacity of the SDH VCs.

4.2 Prohibited Connectivity

The following connectivity is specifically prohibited:-

a) Trails from the same physical port of the border function that go via different physical instances of the CTF **shall** be prohibited.

b) Border Functions **shall not** behave as intermediate CTF switches. That is trails **shall** start and terminate on a border function and **shall not** transit an intermediate border function.

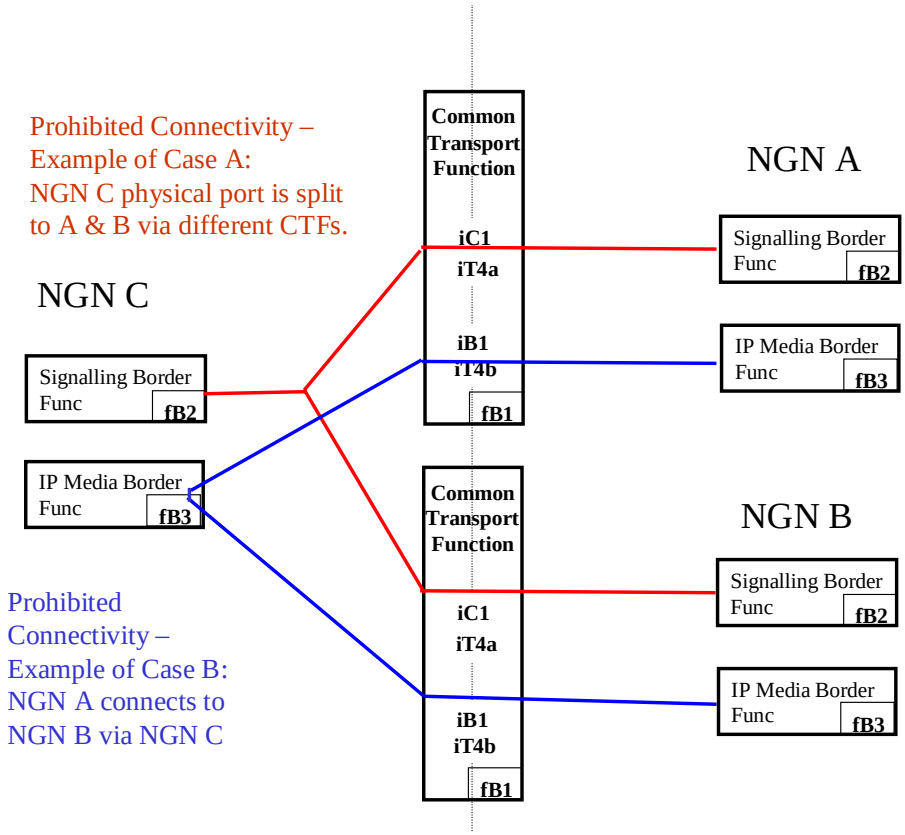


FIGURE 2: EXAMPLES OF PROHIBITED CONNECTIVITY CASES

4.3 Guaranteed Bandwidth and QoS for the Services

Each trail **shall** be constrained to a specified peak-rate to prevent contention for bandwidth (also known as bit-rate) between trails. Bandwidth sharing between trails **shall not** be permitted.
Note:

- Suitable mechanisms for bandwidth limiting are policing or shaping. Shaping is useful as a mechanism to modify the traffic profile (i.e. bandwidth & burst characteristics) of a trail without necessarily imposing discard (i.e. smoothing out bursts), while policing maintains traffic within a specified bit-rate profile, dropping packets if necessary to do so.
- Both policers and shapers are generally characterised by both a sustainable mean bit-rate and burst (bytes) parameters. A token-bucket formulation is commonly used to describe and specify behaviour. The integration period for measuring the mean bit rate (token bucket size) **shall** be agreed on a bi-lateral basis.
- The approach of using a CBR profile across the CTFI greatly simplifies the QoS implementation, reduces the risk of error, and eases fault detection.
- The maximum MTU present within any trail has a bearing on worst-case delay and jitter experienced by other trails.
- The default maximum MTU size **shall** be 2000 bytes, excluding IFG, Preamble and SFD. (This aligns with proposals within IEEE Frame Expansion Task Force IEEE 802.3as.)

4.3.1 Management of Delay and Jitter for Critical Services

Delay and jitter management on the traffic egress point from one NGN to another NGN across the CTF **should** be the responsibility of the transmitting NGN. (It is an extension of overall

responsibility for meeting performance requirements for each service, which covers the rest of the NGN.)
Note - Where stringent requirements exist for delay/jitter for particular services carried across the CTF, the approach described above of individually peak-rate limiting VLAN trails is not always sufficient. In some scenarios, transient contention between peak-rate limited VLAN trails on the CTF itself can give rise to levels of delay/jitter that may be deemed unacceptable. Aggravating combinations of factors are: (i) CTF aggregate speed below a certain level, (ii) mixture of services including jitter-sensitive and data-oriented services, (iii) large maximum MTU for data traffic, (iv) traffic carried on relatively large number of separate VLAN trails and (v) a desire to obtain high overall utilisation across the CTF.
In a mixed-services environment, prioritisation via multiple queues scheduling **may** be used to reduce jitter for a selected subset of the traffic. (Prioritisation is most effective when either the average packet-size or traffic-volume of the prioritised traffic is significantly smaller than for the other traffic.)
Each NGN CP **shall** consider applying prioritisation (or other multiple queues scheduling technique to achieve prioritisation) for traffic injected across CTF. QoS scheduling **may** be used without direct agreement or negotiation of details with the connecting NGN.
Refer to ND1613 [16] for guidance on prioritisation and scheduling.

4.3.2 Definitions and Parameters That Must Be Exchanged Between Connecting CPs Across the CTF

4.3.2.1 Bandwidth Definitions for Services Carried

Information on bandwidth definitions is contained in ND1613 [16].

4.3.2.1 Design Rules Constraining VLAN Trail Bandwidth Across CTF

No bandwidth contention exists between VLAN trails.
CPs **should** refer to the definition of bandwidth given in ND1613 [16]. Since the bandwidth definition does not include the Ethernet overheads associated with Preamble, SFD and IFG, an additional allowance **must** be made for these rather than simply ensuring the sum of VLAN trail sizes is less than the nominal CTF speed.
This amounts to the need to estimate the total bandwidth consumed by all VLAN trails on the CTF, including additional bandwidth equivalent to 20bytes per frame (For the IFG, Preamble and SFD,) and comparing this bandwidth to that available.
CPs **may** choose to limit the total bandwidth thus derived to a value somewhat less than make the upper limit lower than the available CTF bandwidth, for example, theoretical maximum as part of a delay/jitter management strategy. This is to accommodate correlation in traffic peaks with a specific service instance and between different service instances. Each CP NGN **must** provide clear information on how such constraining rules should be applied to the CTF.
If different connecting CPs implement different rules, the most stringent rule across the common CTF **should** be adopted.
Additional rules on VLAN usage may be applied on a per-service basis and are contained in service architecture documents and associated management guides, e.g. ND1612 [17] and ND1614 [16] for PSTN/ISDN services.

4.3.2.3 Parameters for Ingress Policing

Policing is commonly specified by means of a token-bucket formulation, i.e. a combination of a rate (bits/s) and a burst-size (bytes). CPs **may** apply policing on a per-VLAN trail basis at the ingress to their network, but if they do so, the police rate **must** be greater than or equal to the VLAN trail rate, properly taking into account precise bandwidth definitions. In addition, the maximum tolerable

burst-size **should** be greater than that reasonably to be expected for the service carried, and **must** be specified.
Burst behaviour is a complex area that is not well understood, especially for interconnection involving multiple concatenated networks. Taking account of this, guidance for burst tolerance is given here, based on a generous assessment of what is likely to be sufficient in practice for any service:

Burst-tolerance (bytes) = 0.03(s) x VLAN-trail-rate (bits/s)/8

This does not preclude CPs choosing to specify a different burst-tolerance. This may be on a per-service basis if preferred.

4.3.2.4 Controlling the Egress Traffic Profile

CPs **must** ensure VLAN trail egress traffic conforms to the VLAN trail rate and agreed burst size parameter specified for ingress to networks to which they connect. Meeting this requirement for some services **may** require CPs to implement either policing or shaping, though it should be noted that shaping is likely to be preferable for many non-RT application traffic types. For other services this traffic-conformance requirement may be met naturally as a result of fundamental features of the service. An example of this is PSTN, where correctly-applied session-control limits the rate to within the police rate, while the burst-sizes which may be estimated by statistical queuing theory, should be significantly smaller than the values specified by the above formula. Even for such services CPs **may** apply policing or shaping as a precautionary measure, for example to protect other services or VLAN trails in event of a deviation from normally expected behaviour (such as due to a failure of session-control for PSTN).
CPs should be aware that unless the shaper itself has sufficient burst-tolerance, it may lead to the imposition of jitter on individual sessions carried on the VLAN trail, which may be significant for some services. As a consequence, where strict shaping is applied, a design rule is likely to be needed for the most jitter-critical services to constrain the maximum utilisation within a trail to less than 100%, so as to limit the magnitude of any imposed delay variation, refer to ND1614 [16].

4.3.3 Use of QoS Markings

Where all traffic within a trail requires the same QoS treatment across the CTF, it **shall not** be necessary for QOS markings to be part of the CTF specification.
Where each trail consists of only one traffic type, the use of QoS markings **may not** be necessary to achieve differentiation. Inspection of trail identifiers (VLAN ID) **may** be sufficient to identify two types of traffic.
Where the use of Prioritisation results in residual 802.1p or PCP markings being left on traffic egressing from an NGN, these markings **shall** be ignored in the ingress direction arriving from the CTF.
Note - It is recognised that future services may require support of multiple traffic types with different QoS requirements within the same trail, and in this case, consideration **may** be needed to providing some differentiation at packet-level across the CTF by reference to 802.1p or PCP markings. Since a trail **shall** not re-order packets if packets within the trail are differentiated then such a construct should be called a “point to point flow”. This scenario is not considered any further in this document.

4.4 Transport Services Protocol Stacks

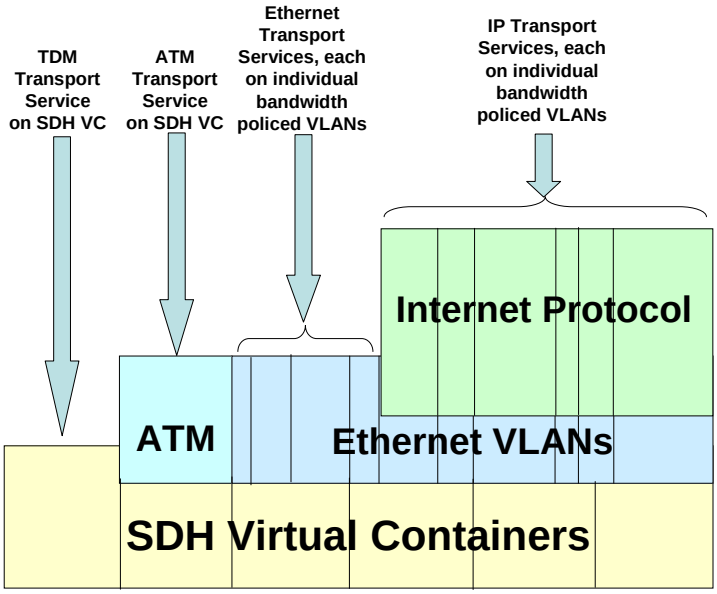


FIGURE 2: TRANSPORT SERVICES SUPPORTED BY SDH TRANSMISSION TECHNOLOGY

The dotted vertical lines in figures 2 & 3 represent the partitioning of the protocols into separate trails (including the use of VLAN trails in Ethernet in the context of this interconnect specification) by use of labels of different values.

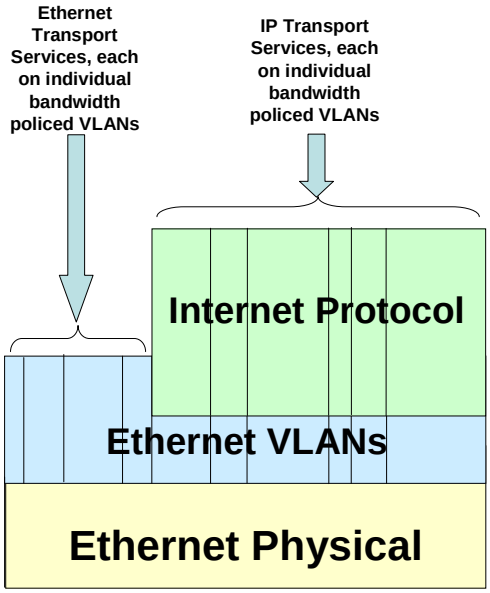


FIGURE 3: TRANSPORT SERVICES SUPPORTED BY ETHERNET TRANSMISSION TECHNOLOGY

4.5 Network Synchronisation

SDH transmission between networks **should not** provide network synchronisation**.** This is in line with the current SDH interconnect on legacy PSTNs in the UK.
For interconnects using Ethernet over fibre, there is currently no standard for conveying network synchronisation.
Unless deriving synchronisation via some other interconnect interface type, networks **should** synchronise against their own network clock that is compliant to ITU-T G.811 [11].

5 IP Transport Capability Specification – iT4

5.1 Physical Layer Options

5.1.1 Physical Interface Options

The layer 1 options for the IP transport capability are:

1. Ethernet mapped into SDH interconnect as per SDH Interconnect Between UK Licensed Operators, Technical Recommendation [3] using ITU-T G.7041 [15] framed GFP.

2. 10 Gigabit Ethernet IEEE802.3ae [12] options (not mandatory):

Transceiver Type	**Wavelength**	**IEEE Standard**	**Maximum Distance/Cable Type**
10GBASE-SR	850 nm	802.3ae	300 m over 50-micron 2000 MHz*km multimode fibre
10GBASE-LR	1310 nm	802.3ae	10 km over single-mode fibre
10GBASE-ER	1550 nm	802.3ae	40 km over single-mode fibre
10GBASE-LW	1310 nm	802.3ae	(STM-64 variant WAN Phy) Single-mode fibre
10GBASE-EW	1550 nm	802.3ae	(STM-64 variant WAN Phy) Single-mode fibre

3. IEEE Gigabit Ethernet Options IEEE802.3 [6] (not mandatory):
    a. 1000BASE-SX: 62.5 um multimode fibre: up to 275 m
    b. 1000BASE-LX: 9/10 um single-mode fibre: up to 10 km
    c. 1000BASE-T: Category 5 cable: up to 100 m

5.1.2 Protection Mechanisms

When a Border Function detects or initiates a protection switching event the Border Function **shall** initiate an ARP announcement (also known as a “Gratuitous ARP”), after the protection switching event is complete, to update the ARP cache of its peers.
Note that Border Functions may not detect all protection switching events which may lead to a failure of IP connectivity of peer Border Functions until stale ARP entries have expired
SDH Multiplex Section Protection (MSP) **may** be used to provide “across the floor” protection for the SDH layer 1 interconnect option. Native Ethernet layer 1 options do not have equivalent protection mechanisms so IEEE 802.3ad [8] Link aggregation **may** be used.

5.1.3 GFP Client Signal Fail Frame (CSFF)

Where GFP is used CSFF **should** be used to indicate failure of the far-end Ethernet connectivity where:
Upon receiving a CSFF signal the SDH/GFP function **shall** "take down" (remove carrier or light) for the SDH section only if all VLANs connectivity on that section have failed.

5.1.4 Use of SDH LCAS

The SDH Link Capacity Adjustment Scheme (LCAS), standardized by the ITU-T as G.7042 [21], is designed to manage the bandwidth allocation of a Virtually Concatenated Group. SDH LCAS is recognised as a technique for varying the capacity of an SDH VC Group for an Ethernet client carried using GFP-F, without the need for an outage of the SDH VC Group. LCAS **may** be used across an MSI by bilateral agreement.

5.2 iT4 - Layer 2 for IP Transport Capability

"Ethernet" **shall** be the layer 2 used for IP services and the following Ethernet standards **shall** be followed:

1. IEEE 802.1Q [7] VLAN tagging. Different IP services will be placed in different VLANs. The VLAN ids **shall** be agreed on a bi-lateral basis between CPs.
2. IEEE 802.3ad [8] Link Aggregation **may** be used to provide load sharing and protection (which usually takes seconds to detect failure) (Note using 802.3ad to provide protection is not a standardised Ethernet feature.)
3. Rapid Spanning Tree Protocol (IEEE 802.1w) [13] **shall not** be used to provide protection as this is not a secure protocol to operate inter-CP.
4. IEEE 802.1p priority marking [10] **should not** be used. The CTF will be dimensioned to not drop or contend traffic at the point of interconnection. Individual operators **shall** be responsible for policing traffic onto the point to point interconnect to ensure it is not overloaded. Traffic **must** be policed per VLAN trail(i.e. per service) to ensure congestion/overload of a single service does not impact the performance of other services (assuming no overbooking). Per VLAN trail queuing **may** give the best performance isolation between services but is not a requirement.

5.3 iT4 – Failure Detection

The architecture for detection of failure of the IP transport function is shown in Figure 4. There are 3 key components to consider for failure detection:

- The physical trail. This includes the IEEE 802.3 functional components and may use SDH. The physical trail may use intermediate physical transport components of differing technologies.
- The VLAN trail. This includes the IEEE 802.1 functional components. The VLAN trail may cross intermediate Ethernet switches.
- The Service Level trail (for connection-oriented services) or monitored fragment (for connectionless services) between border functions. This is IP for iT4.

Each trail or monitored fragment **should** provide its own OAM protocol(s) to detect all the failures that are relevant to the trail or monitored fragment. The physical trail termination **may** pass proprietary Forward Defect Indicators to the VLAN trail termination and raise a management alarm signal. The VLAN trail termination **may** pass proprietary Forward Defect Indicators to the service trail termination or monitored fragment termination in the Border Function and raise a management alarm signal. The service level trail or monitored fragment termination in the Border Function may

pass proprietary Forward Defect Indicators to the Application (if applicable to the application) and raise a management alarm signal. In general there are no guarantees that:

- The OAM protocol(s) will operate effectively end to end, especially in the case of the physical trail over multiple physical sections.
- That proprietary FDIs will be available.

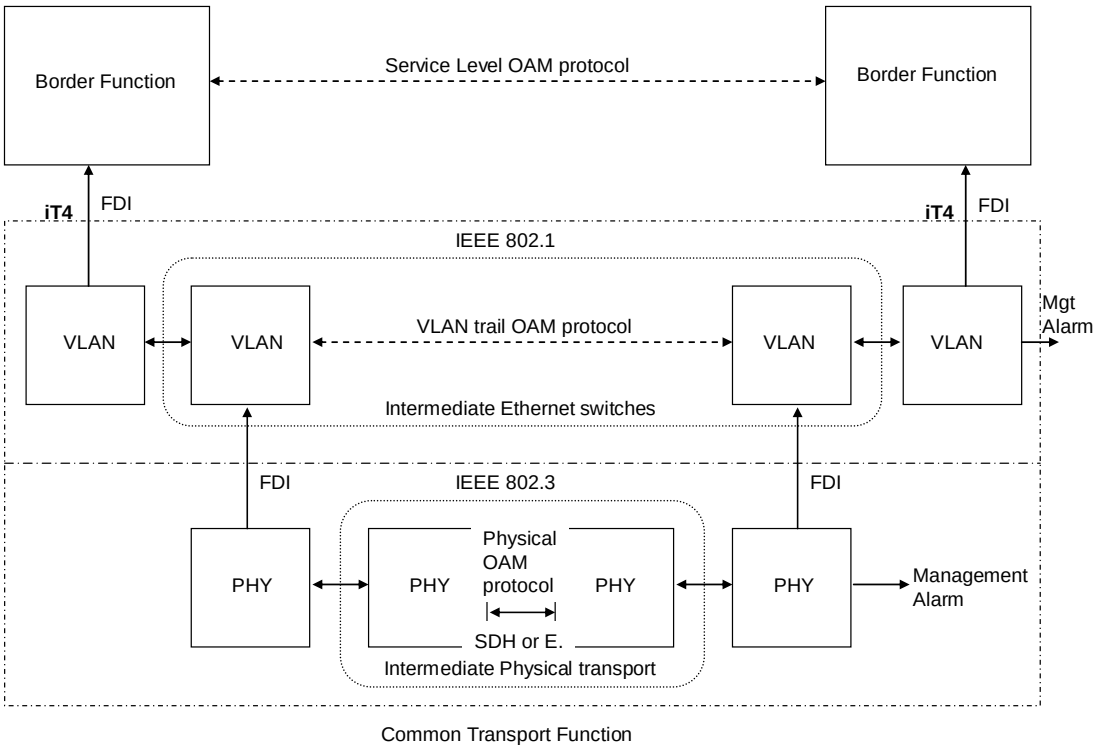


FIGURE 4: FAILURE DETECTION ARCHITECTURE

Figure 5 illustrates the recommended protocols for failure detection of iT4.

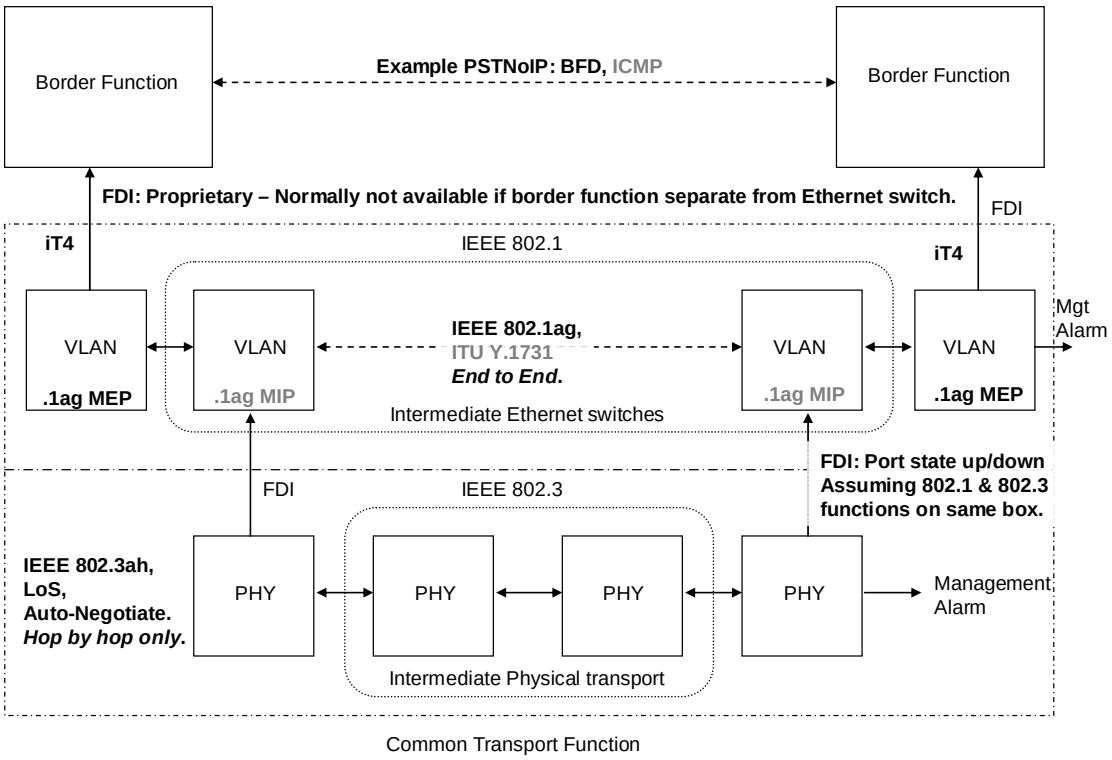


FIGURE 5: FAILURE DETECTION RECOMMENDATIONS

In order to check IP connectivity across iT4, the IP Border Functions using the IP CTF shall use BFD [18] or ICMP IP ping [20]. BFD should be used but ICMP IP ping may be used. If the Border Function is physically separate from the Ethernet switch then proprietary Forward Defect Indicators **should not** be available to the IP Border Functions from the VLAN trail functions. BFD CC rates and failure detection times **should** be determined based on the speed of any protection mechanisms implemented on the VLAN trails or physical trails i.e. client OAM **should** be slower than server layer OAM to give lower layer protection time to restore.

The VLAN trail functions **should** use IEEE 802.1ag CFM and **may** use ITU Y.1731 CC messages. The 802.1ag MEP function **should** be implemented at the VLAN trail terminations. The 802.1ag MIP functions **may** be implemented on intermediate Ethernet switches. The 802.1ag Maintenance Association Level to be used in the VLAN trail terminations **shall** be agreed on a bi-lateral basis. The VLAN trail OAM protocol **shall** provide detection of failure of the end to end VLAN trail. The CC rate per VLAN trail **shall** be negotiated on a bi-lateral basis. CC rates and failure detection times **shall** be determined based on the speed of any protection mechanisms implemented on the physical links. The VLAN trail termination function **should** raise an alarm to the management system on detection of failure. If the VLAN termination function is physically separate to the Border Function then proprietary Forward Defect Indicators **should not** be used. The VLAN trail failure detection mechanisms **may** be used to trigger VLAN trail protection when available in the future.

The physical trail functions **shall** use IEEE 802.3ah OAM and **should use** Loss of Signal and loss of IEEE 802.3 auto-negotiate signal to detect failure. Only if the intermediate physical transport is fully transparent to 802.3 signals then will the 802.3 auto-negotiate signal provide end to end detection of failure. Only if the intermediate physical transport is capable of link loss forwarding (e.g. Ethernet over SDH using GFP Client Signal Fail Frame) will Loss of Signal provide end to end detection of failure. The physical trail termination function **should** raise an alarm to the management system on detection of failure. CC rates and failure detection times **should** be determined based on the speed of any protection mechanisms implemented on the physical links.

Following the above recommendations the time taken to detect failure at a higher layer and invoke actions there will be bounded by:

- The speed of the underlying physical link OAM and protection mechanisms.
- The ability of the Border Functions to implement BFD [18].
- The ability of the management systems to trigger timely actions in the applications in response to alarms indicating failure of the transport function.

5.3.1 BFD Security Profile

BFD sessions **shall** use the The Generalized TTL Security Mechanism (GTSM) [22] to protect the integrity of the BFD sessions and their associated VLAN trails.
BFD authentication options i.e. SHA1, MD5, plain text, **should not** be used until the IETF have agreed an interoperable authentication specification for BFD. A UK BFD authentication profile will be specified in due course.

5.4 iT4 - SDH Transport Option Protocol Stack

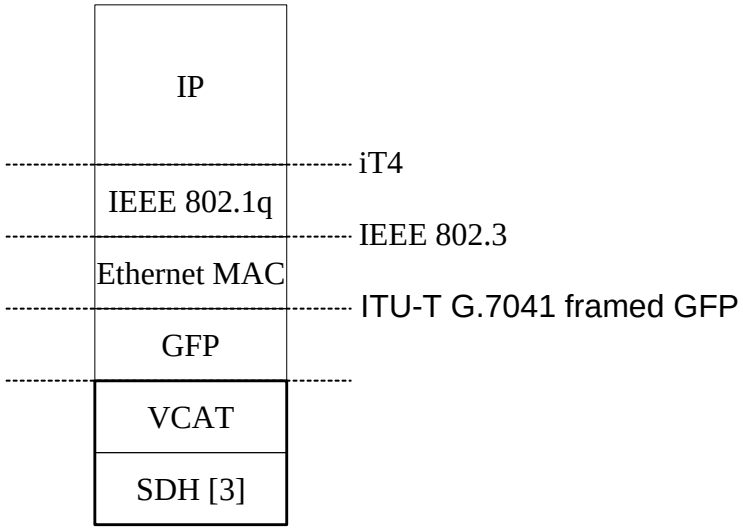


FIGURE 6: IT4 SDH TRANSPORT OPTION PROTOCOL STACK

5.5 iT4 - Ethernet Transport Option Protocol Stack

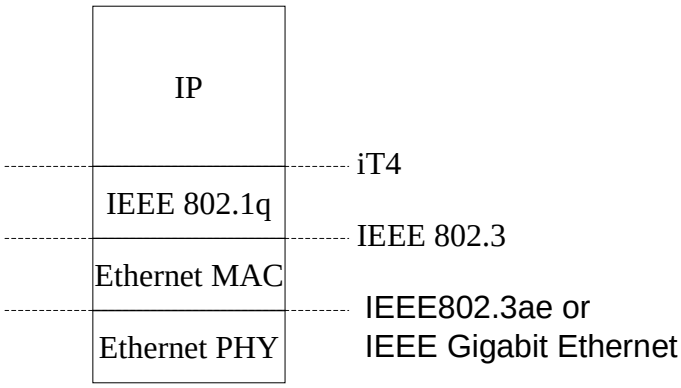


FIGURE 7: IT4 ETHERNET TRANSPORT OPTION PROTOCOL STACK

6 TDM Transport Capability– iT1

The mapping of TDM clients **shall** be as defined in SDH INTERCONNECT BETWEEN UK LICENSED OPERATORS, TECHNICAL RECOMMENDATION [3].

7 ATM Transport Capability– iT2

ATM **shall** be mapped as per INTERCONNECT BETWEEN UK LICENSED OPERATORS, BASED UPON PERMANENT ATM CONNECTIONS, TECHNICAL RECOMMENDATION [4].

8 Multi-Service Protocol Stacks

8.1 Multi-Service (iT1,2,4) over SDH Protocol Stack

Figure 8 shows the protocol stack for IP, ATM and SDH services over an SDH based transport function.

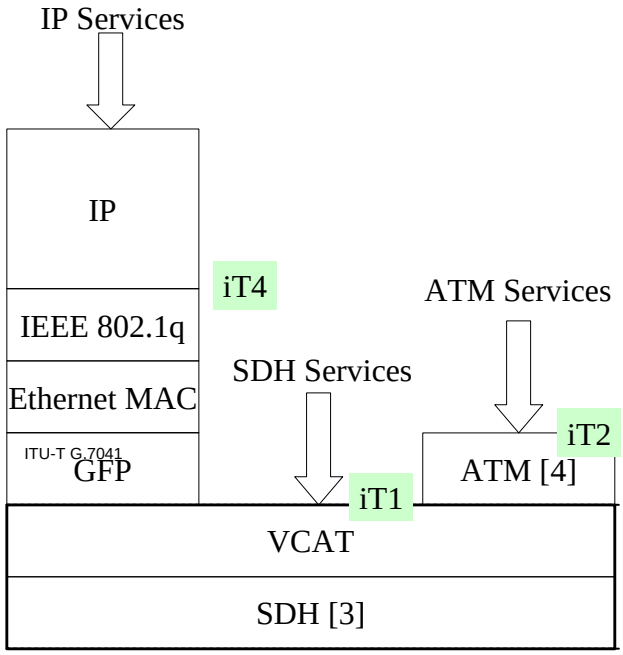


FIGURE 8: MULTI-SERVICE PROTOCOL STACK

8.2 Multi-Service (iT1,2,3,4) over Ethernet Protocol Stack

Since Ethernet standards do not yet support SDH and ATM there is no option to implement SDH or ATM over Ethernet. Only IP services **shall** be supported over the Ethernet transport function.

9 Ethernet Transport capability – iT3

This release of the NGN Interconnect common transport **shall not** support an Ethernet transport capability for Ethernet services. Later releases of this specification **may** support Ethernet services using IEEE 802.1ah [5] “Provider Backbone Bridging” or IEEE 802.1Qay [19] “Provider Backbone Bridge Traffic Engineering”.
For IP transport single tagged Ethernet frame with Ethertype 0x8100 **should** be used. For Ethernet transport double tagged IEEE 802.1ad or 802.1ah Ethernet frames **should** be used with Ethertype 0x88a8. This **may** necessitate the use of double tags and the 0x88a8 Ethertype to support IP transport on multi-service (IP & Ethernet) transport interfaces in the future.
Note that due to GFP’s limitation of an 8 bit multiplexer field it **may not** be a suitable mechanism to support future Ethernet services interconnection where each SDH VC **may** be supporting thousands of customers.

10 Security

The CTF **cannot** provide authentication or privacy for its clients (services). It is recommended that clients (services) using the CTF **should** provide their own authentication and privacy functions.

MAC filtering **shall** be implemented to prevent infinitely circulating Ethernet packets, e.g. CP A **cannot** receive CP's B Ethernet packets from CP C and CP B **cannot** route Ethernet packets to CP C via CP A.
The following protocols **shall not** be used between CPs:

1. Ethernet Spanning Tree Protocols

11 Naming, Numbering and Addressing

11.1 IP Transport Capability

11.1.1 IP Addressing

The IP addresses used by the IP client of the IP transport capability is a service specific issue which will be described in the service specific documents.

11.1.2 Ethernet VLANs Used to Provide IP Transport Capability

VLAN Tag addressing
The VLAN tag **shall** be identified by the VLAN ID (VID). Per IEEE 802.1Q, this has 12 bits, allowing the identification of 4096 VLANs within a given Ethernet network. VID values 0 and 4095 (FFF) **shall** be reserved. The maximum possible VLAN configurations **shall** be 4,094.

There **shall** be no centrally-administered VLAN-tag space for the UK, and the addressing of VLANs **shall** be done through bilateral agreement. Each interconnect point **shall** represent a separate VLAN-space, although network operators **should** give due consideration to how the VLAN separation will be maintained within their network, particularly between the Border Functions and Common Transport Function. This **may** be achieved via tag switching or physical separation.

The assignment of VLANs to interconnect relationships **shall** be service specific, with a given service requiring one or more VLANs. For example, a voice interconnection **shall** require two VLANs, one for control and one for media; if the commercial arrangements were such that each network operator owned their own capacity, this would imply that up to four VLANs could be required for the interconnect as a whole.

Annex (informative): Connectivity Examples

Figure 9 shows examples of permitted connectivity where the square boxes represent IP border functions that combine the adaptation and trail termination functions. The small circles on the square boxes represent physical ports.

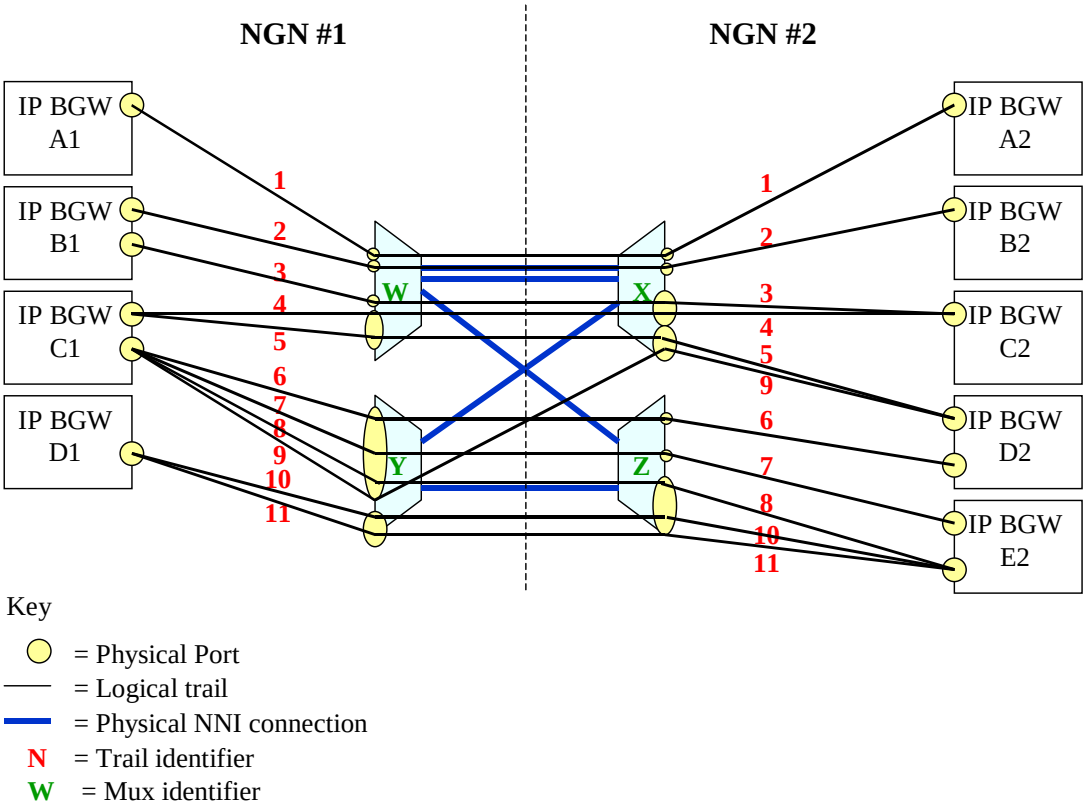


FIGURE 9: TRANSPORT FUNCTION CONNECTIVITY EXAMPLES

Annex B (informative):
iT4 - SDH Transport Option Multiplexing Hierarchy

Figure 10 shows the multiplexing hierarchy for the iB1 SDH transport Option. This shows there **may** be many IP services supported by many VLANs supported by many GFP flows etc.

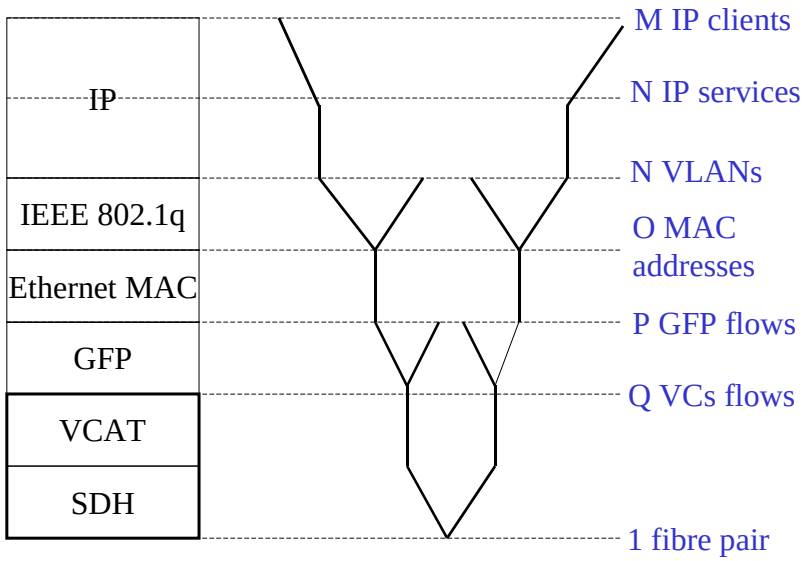


FIGURE 10: IT4 SDH TRANSPORT OPTION MULTIPLEXING HIERARCHY

Annex C (informative):
iT4 - Ethernet Transport Option Multiplexing Hierarchy

Figure 11 shows the multiplexing hierarchy for the iB1 Ethernet transport Option.

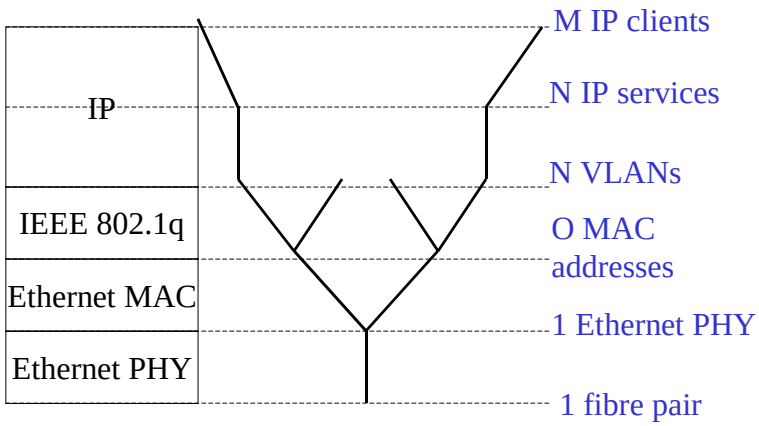


FIGURE 11: IT4 ETHERNET TRANSPORT OPTION MULTIPLEXING HIERARCHY

Annex D (innormative): Multiplexing Hierarchy For The Multi-Service Protocol Stack

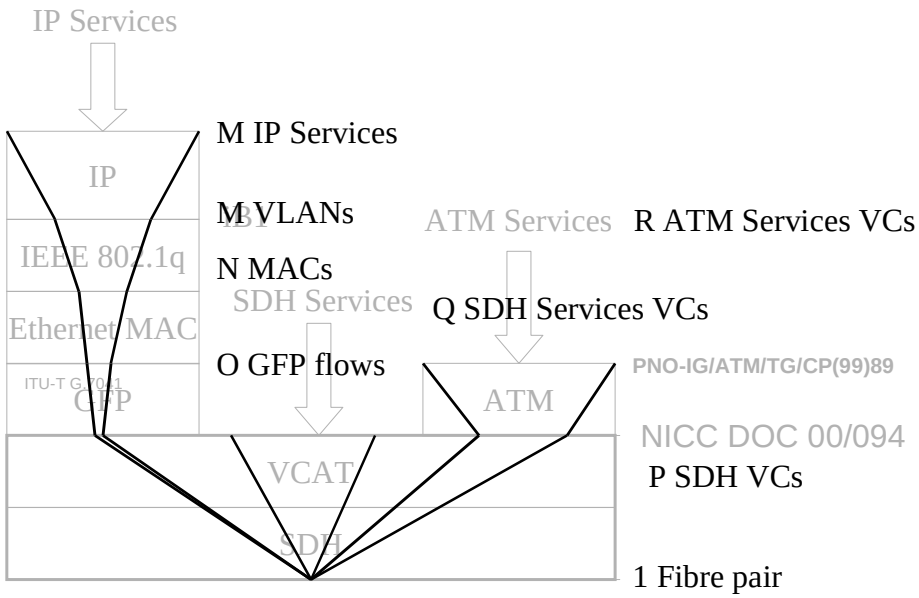


FIGURE 12: MULTIPLEXING HIERARCHY FOR THE MULTI-SERVICE SDH PROTOCOL STACK.

Annex E (informative): Relevant BFD Internet Drafts

This annex includes draft-ieft-bfd-base-08.txt, draft-ietf-bfd-generic-04.txt and draft-ietf-v4v6-1hop-08.txt.

Network Working Group
Internet Draft

D. Katz
Juniper Networks
D. Ward
Cisco Systems

Expires: September, 2008

March, 2008

Bidirectional Forwarding Detection
draft-ietf-bfd-base-08.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

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/1id-abstracts.html

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

Abstract

This document describes a protocol intended to detect faults in the bidirectional path between two forwarding engines, including interfaces, data link(s), and to the extent possible the forwarding engines themselves, with potentially very low latency. It operates independently of media, data protocols, and routing protocols. Comments on this draft should be directed to rtg-bfd@ietf.org.

Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [KEYWORDS].

Table of Contents

1. Introduction

An increasingly important feature of networking equipment is the rapid detection of communication failures between adjacent systems, in order to more quickly establish alternative paths. Detection can come fairly quickly in certain circumstances when data link hardware comes into play (such as SONET alarms.) However, there are media that do not provide this kind of signaling (such as Ethernet), and some media may not detect certain kinds of failures in the path, for example, failing interfaces or forwarding engine components.

Networks use relatively slow "Hello" mechanisms, usually in routing protocols, to detect failures when there is no hardware signaling to help out. The time to detect failures ("Detection Times") available in the existing protocols are no better than a second, which is far too long for some applications and represents a great deal of lost data at gigabit rates. Furthermore, routing protocol Hellos are of no help when those routing protocols are not in use, and the semantics of detection are subtly different--they detect a failure in the path between the two routing protocol engines.

The goal of BFD is to provide low-overhead, short-duration detection of failures in the path between adjacent forwarding engines, including the interfaces, data link(s), and to the extent possible the forwarding engines themselves.

An additional goal is to provide a single mechanism that can be used for liveness detection over any media, at any protocol layer, with a wide range of Detection Times and overhead, to avoid a proliferation of different methods.

This document specifies the details of the base protocol. The use of some mechanisms are application dependent and are specified in a separate series of application documents. These issues are so noted.

Note that many of the exact mechanisms are implementation dependent and will not affect interoperability, and are thus outside the scope of this specification. Those issues are so noted.

2. Design

BFD is designed to detect failures in communication with a forwarding plane next hop. It is intended to be implemented in some component of the forwarding engine of a system, in cases where the forwarding and control engines are separated. This not only binds the protocol more to the forwarding plane, but decouples the protocol from the fate of the routing protocol engine, making it useful in concert with various "graceful restart" mechanisms for those protocols. BFD may also be implemented in the control engine, though doing so may preclude the detection of some kinds of failures.

BFD operates on top of any data protocol being forwarded between two systems. It is always run in a unicast, point-to-point mode. BFD packets are carried as the payload of whatever encapsulating protocol is appropriate for the medium and network. BFD may be running at multiple layers in a system. The context of the operation of any particular BFD session is bound to its encapsulation.

BFD can provide failure detection on any kind of path between systems, including direct physical links, virtual circuits, tunnels, MPLS LSPs, multihop routed paths, and unidirectional links (so long as there is some return path, of course.) Multiple BFD sessions can be established between the same pair of systems when multiple paths between them are present in at least one direction, even if a lesser number of paths are available in the other direction (multiple parallel unidirectional links or MPLS LSPs, for example.)

The BFD state machine implements a three-way handshake, both when establishing a BFD session and when tearing it down for any reason, to ensure that both systems are aware of the state change.

BFD can be abstracted as a simple service. The service primitives provided by BFD are to create, destroy, and modify a session, given the destination address and other parameters. BFD in return provides a signal to its clients indicating when the BFD session goes up or down.

3. Protocol Overview

BFD is a simple hello protocol that in many respects is similar to the detection components of well-known routing protocols. A pair of systems transmit BFD packets periodically over each path between the two systems, and if a system stops receiving BFD packets for long enough, some component in that particular bidirectional path to the neighboring system is assumed to have failed. Under some conditions, systems may negotiate to not send periodic BFD packets in order to reduce overhead.

A path is only declared to be operational when two-way communication has been established between systems, though this does not preclude the use of unidirectional links.

A separate BFD session is created for each communications path and data protocol in use between two systems.

Each system estimates how quickly it can send and receive BFD packets in order to come to an agreement with its neighbor about how rapidly detection of failure will take place. These estimates can be modified in real time in order to adapt to unusual situations. This design also allows for fast systems on a shared medium with a slow system to be able to more rapidly detect failures between the fast systems while allowing the slow system to participate to the best of its ability.

3.1. Addressing and Session Establishment

A BFD session is established based on the needs of the application that will be making use of it. It is up to the application to determine the need for BFD, and the addresses to use--there is no discovery mechanism in BFD. For example, an OSPF [OSPF] implementation may request a BFD session to be established to a neighbor discovered using the OSPF Hello protocol.

3.2. Operating Modes

BFD has two operating modes which may be selected, as well as an additional function that can be used in combination with the two modes.

The primary mode is known as Asynchronous mode. In this mode, the systems periodically send BFD Control packets to one another, and if a number of those packets in a row are not received by the other system, the session is declared to be down.

The second mode is known as Demand mode. In this mode, it is assumed that a system has an independent way of verifying that it has connectivity to the other system. Once a BFD session is established, such a system may ask the other system to stop sending BFD Control packets, except when the system feels the need to verify connectivity explicitly, in which case a short sequence of BFD Control packets is exchanged, and then the far system quiesces. Demand mode may operate independently in each direction, or simultaneously.

An adjunct to both modes is the Echo function. When the Echo function is active, a stream of BFD Echo packets is transmitted in such a way as to have the other system loop them back through its forwarding path. If a number of packets of the echoed data stream are not received, the session is declared to be down. The Echo function may be used with either Asynchronous or Demand modes. Since the Echo function is handling the task of detection, the rate of periodic transmission of Control packets may be reduced (in the case of Asynchronous mode) or eliminated completely (in the case of Demand mode.)

Pure asynchronous mode is advantageous in that it requires half as many packets to achieve a particular Detection Time as does the Echo function. It is also used when the Echo function cannot be supported for some reason.

The Echo function has the advantage of truly testing only the forwarding path on the remote system. This may reduce round-trip jitter and thus allow more aggressive Detection Times, as well as potentially detecting some classes of failure that might not otherwise be detected.

The Echo function may be enabled individually in each direction. It is enabled in a particular direction only when the system that loops the Echo packets back signals that it will allow it, and when the system that sends the Echo packets decides it wishes to.

Demand mode is useful in situations where the overhead of a periodic protocol might prove onerous, such as a system with a very large number of BFD sessions. It is also useful when the Echo function is being used symmetrically. Demand mode has the disadvantage that Detection Times are essentially driven by the heuristics of the system implementation and are not known to the BFD protocol. Demand mode may not be used when the path round trip time is greater than the desired Detection Time. See section 6.6 for more details.

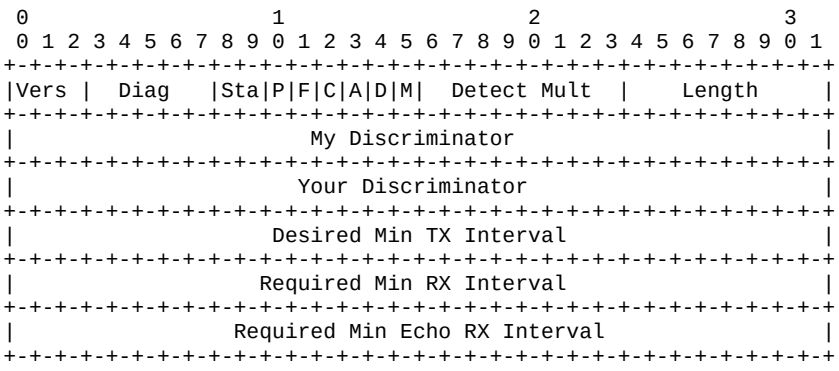
4. BFD Control Packet Format

4.1. Generic BFD Control Packet Format

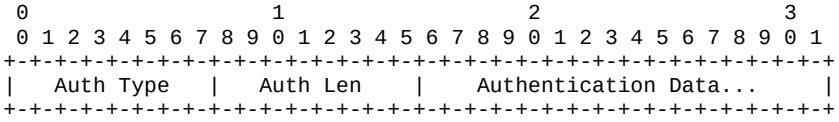
BFD Control packets are sent in an encapsulation appropriate to the environment. The specific encapsulation is outside of the scope of this specification. See the appropriate application document for encapsulation details.

The BFD Control packet has a Mandatory Section and an optional Authentication Section. The format of the Authentication Section, if present, is dependent on the type of authentication in use.

The Mandatory Section of a BFD Control packet has the following format:



An optional Authentication Section may be present:



Version (Vers)

The version number of the protocol. This document defines protocol version 1.

Diagnostic (Diag)

A diagnostic code specifying the local system's reason for the last change in session state. Values are:

0 -- No Diagnostic
1 -- Control Detection Time Expired
2 -- Echo Function Failed
3 -- Neighbor Signaled Session Down
4 -- Forwarding Plane Reset
5 -- Path Down
6 -- Concatenated Path Down
7 -- Administratively Down
8 -- Reverse Concatenated Path Down
9-31 -- Reserved for future use

This field allows remote systems to determine the reason that the previous session failed, for example.

State (Sta)

The current BFD session state as seen by the transmitting system. Values are:

0 -- AdminDown
1 -- Down
2 -- Init
3 -- Up

Poll (P)

If set, the transmitting system is requesting verification of connectivity, or of a parameter change, and is expecting a packet with the Final (F) bit in reply. If clear, the transmitting system is not requesting verification.

Final (F)

If set, the transmitting system is responding to a received BFD Control packet that had the Poll (P) bit set. If clear, the transmitting system is not responding to a Poll.

Control Plane Independent (C)

If set, the transmitting system's BFD implementation does not share fate with its control plane (in other words, BFD is implemented in the forwarding plane and can continue to function through disruptions in the control plane.) If clear, the transmitting system's BFD implementation shares fate with its control plane.

The use of this bit is application dependent and is outside the scope of this specification. See specific application specifications for details.

Authentication Present (A)

If set, the Authentication Section is present and the session is to be authenticated.

Demand (D)

If set, Demand mode is active in the transmitting system (the system wishes to operate in Demand mode, knows that the session is up in both directions, and is directing the remote system to cease the periodic transmission of BFD Control packets.) If clear, Demand mode is not active in the transmitting system.

Multipoint (M)

This bit is reserved for future point-to-multipoint extensions to BFD. It must be zero on both transmit and receipt.

Detect Mult

Detection time multiplier. The negotiated transmit interval, multiplied by this value, provides the Detection Time for the transmitting system in Asynchronous mode.

Length

Length of the BFD Control packet, in bytes.

My Discriminator

A unique, nonzero discriminator value generated by the transmitting system, used to demultiplex multiple BFD sessions between the same pair of systems.

Your Discriminator

The discriminator received from the corresponding remote system. This field reflects back the received value of My Discriminator, or is zero if that value is unknown.

Desired Min TX Interval

This is the minimum interval, in microseconds, that the local system would like to use when transmitting BFD Control packets. The value zero is reserved.

Required Min RX Interval

This is the minimum interval, in microseconds, between received BFD Control packets that this system is capable of supporting. If this value is zero, the transmitting system does not want the remote system to send any periodic BFD Control packets.

Required Min Echo RX Interval

This is the minimum interval, in microseconds, between received BFD Echo packets that this system is capable of supporting. If this value is zero, the transmitting system does not support the receipt of BFD Echo packets.

Auth Type

The authentication type in use, if the Authentication Present (A) bit is set.

0 - Reserved
1 - Simple Password
2 - Keyed MD5
3 - Meticulous Keyed MD5
4 - Keyed SHA1
5 - Meticulous Keyed SHA1

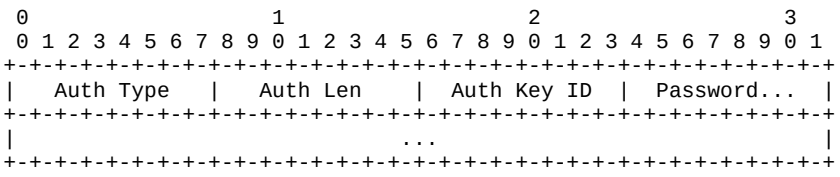
6-255 - Reserved for future use

Auth Len

The length, in bytes, of the authentication section, including the Auth Type and Auth Len fields.

4.2. Simple Password Authentication Section Format

If the Authentication Present (A) bit is set in the header, and the Authentication Type field contains 1 (Simple Password), the Authentication Section has the following format:



Auth Type

The Authentication Type, which in this case is 1 (Simple Password.)

Auth Len

The length of the Authentication Section, in bytes. For Simple Password authentication, the length is equal to the password length plus three.

Auth Key ID

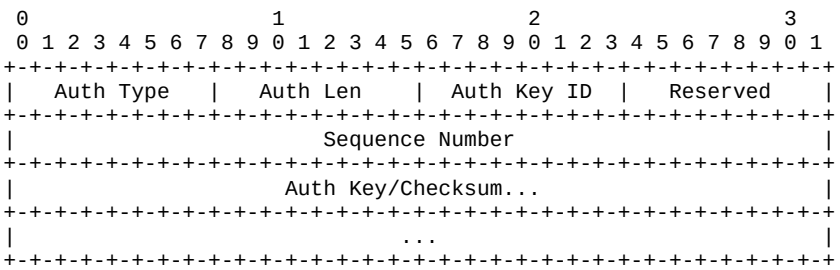
The authentication key ID in use for this packet. This allows multiple keys to be active simultaneously.

Password

The simple password in use on this session. The password MUST be from 1 to 16 bytes in length.

4.3. Keyed MD5 and Meticulous Keyed MD5 Authentication Section Format

If the Authentication Present (A) bit is set in the header, and the Authentication Type field contains 2 (Keyed MD5) or 3 (Meticulous Keyed MD5), the Authentication Section has the following format:



Auth Type

The Authentication Type, which in this case is 2 (Keyed MD5) or 3 (Meticulous Keyed MD5).

Auth Len

The length of the Authentication Section, in bytes. For Keyed MD5 and Meticulous Keyed MD5 authentication, the length is 24.

Auth Key ID

The authentication key ID in use for this packet. This allows multiple keys to be active simultaneously.

Reserved

This byte must be set to zero on transmit, and ignored on receipt.

Sequence Number

The Sequence Number for this packet. For Keyed MD5 Authentication, this value is incremented occasionally. For Meticulous Keyed MD5 Authentication, this value is incremented for each successive packet transmitted for a session. This provides

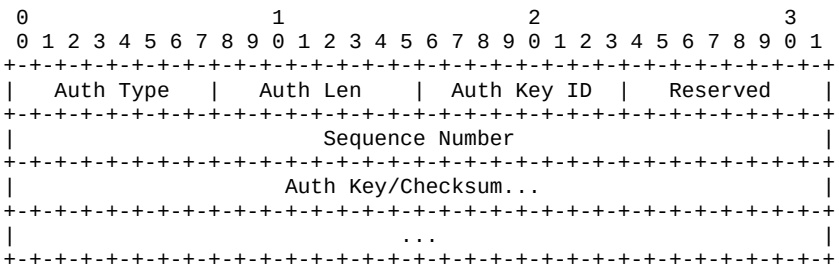
protection against replay attacks.

Auth Key/Checksum

This field carries the 16 byte MD5 checksum for the packet. When the checksum is calculated, the shared MD5 key is stored in this field. (See section 6.7.3 for details.)

4.4. Keyed SHA1 and Meticulous Keyed SHA1 Authentication Section Format

If the Authentication Present (A) bit is set in the header, and the Authentication Type field contains 4 (Keyed SHA1) or 5 (Meticulous Keyed SHA1), the Authentication Section has the following format:



Auth Type

The Authentication Type, which in this case is 4 (Keyed SHA1) or 5 (Meticulous Keyed SHA1).

Auth Len

The length of the Authentication Section, in bytes. For Keyed SHA1 and Meticulous Keyed SHA1 authentication, the length is 28.

Auth Key ID

The authentication key ID in use for this packet. This allows multiple keys to be active simultaneously.

Reserved

This byte must be set to zero on transmit, and ignored on receipt.

Sequence Number

The Sequence Number for this packet. For Keyed SHA1 Authentication, this value is incremented occasionally. For Meticulous Keyed SHA1 Authentication, this value is incremented for each successive packet transmitted for a session. This provides protection against replay attacks.

Auth Key/Checksum

This field carries the 20 byte SHA1 checksum for the packet. When the checksum is calculated, the shared SHA1 key is stored in this field. (See section 6.7.4 for details.)

5. BFD Echo Packet Format

BFD Echo packets are sent in an encapsulation appropriate to the environment. See the appropriate application documents for the specifics of particular environments.

The payload of a BFD Echo packet is a local matter, since only the sending system ever processes the content. The only requirement is that sufficient information is included to demultiplex the received packet to the correct BFD session after it is looped back to the sender. The contents are otherwise outside the scope of this specification.

6. Elements of Procedure

This section discusses the normative requirements of the protocol in order to achieve interoperability. It is important for implementors to enforce only the requirements specified in this section, as misguided pedantry has been proven by experience to adversely affect interoperability.

Remember that all references of the form "bfd.Xx" refer to internal state variables (defined in section 6.8.1), whereas all references to "the Xxx field" refer to fields in the protocol packets themselves (defined in section 4).

6.1. Overview

A system may take either an Active role or a Passive role in session initialization. A system taking the Active role MUST send BFD Control packets for a particular session, regardless of whether it has received any BFD packets for that session. A system taking the Passive role MUST NOT begin sending BFD packets for a particular session until it has received a BFD packet for that session, and thus has learned the remote system's discriminator value. At least one system MUST take the Active role (possibly both.) The role that a system takes is specific to the application of BFD, and is outside the scope of this specification.

A session begins with the periodic, slow transmission of BFD Control packets. When bidirectional communication is achieved, the BFD session comes Up.

Once the BFD session is Up, a system can choose to start the Echo function if it desires to and the other system signals that it will allow it. The rate of transmission of Control packets is typically kept low when the Echo function is active.

If the Echo function is not active, the transmission rate of Control packets may be increased to a level necessary to achieve the Detection Time requirements for the session.

Once the session is up, a system may signal that it has entered Demand mode, and the transmission of BFD Control packets by the remote system ceases. Other means of implying connectivity are used to keep the session alive. If either system wishes to verify bidirectional connectivity, it can initiate a short exchange of BFD Control packets (a "Poll Sequence"; see section 6.5) to do so.

If Demand mode is not active, and no Control packets are received in

the calculated Detection Time (see section 6.8.4), the session is declared Down. This is signaled to the remote end via the State (Sta) field in outgoing packets.

If sufficient Echo packets are lost, the session is declared down in the same manner. See section 6.8.5.

If Demand mode is active and no appropriate Control packets are received in response to a Poll Sequence, the session is declared down in the same manner. See section 6.6.

If the session goes down, the transmission of Echo packets (if any) ceases, and the transmission of Control packets goes back to the slow rate.

Once a session has been declared down, it cannot come back up until the remote end first signals that it is down (by leaving the Up state), thus implementing a three-way handshake.

A session may be kept administratively down by entering the AdminDown state and sending an explanatory diagnostic code in the Diagnostic field.

6.2. BFD State Machine

The BFD state machine is quite straightforward. There are three states through which a session normally proceeds, two for establishing a session (Init and Up) and one for tearing down a session (Down.) This allows a three-way handshake for both session establishment and session teardown (assuring that both systems are aware of all session state changes.) A fourth state (AdminDown) exists so that a session can be administratively put down indefinitely.

Each system communicates its session state in the State (Sta) field in the BFD Control packet, and that received state in combination with the local session state drives the state machine.

Down state means that the session is down (or has just been created.) A session remains in Down state until the remote system indicates that it agrees that the session is down by sending a BFD Control packet with the State field set to anything other than Up. If that packet signals Down state, the session advances to Init state; if that packet signals Init state, the session advances to Up state. Semantically, Down state indicates that the forwarding path is unavailable, and that appropriate actions should be taken by the applications monitoring the state of the BFD session. A system MAY

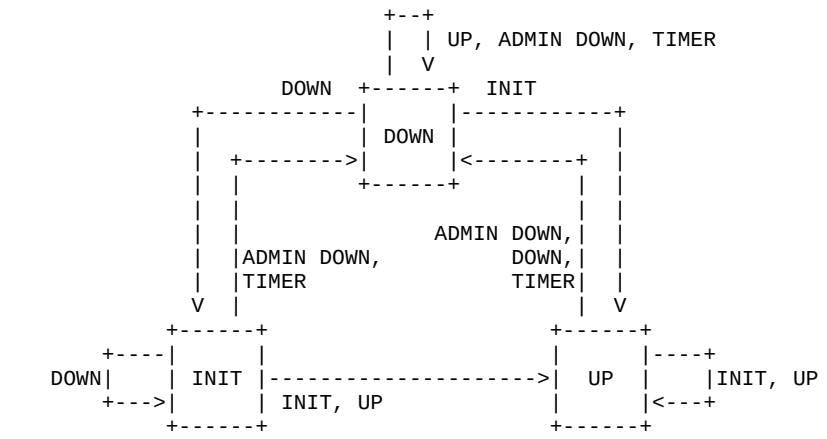
hold a session in Down state indefinitely (by simply refusing to advance the session state.) This may be done for operational or administrative reasons, among others.

Init state means that the remote system is communicating, and the local system desires to bring the session up, but the remote system does not yet realize it. A session will remain in Init state until either a BFD Control Packet is received that is signaling Init or Up state (in which case the session advances to Up state) or until the Detection Time expires, meaning that communication with the remote system has been lost (in which case the session advances to Down state.)

Up state means that the BFD session has successfully been established, and implies that connectivity between the systems is working. The session will remain in the Up state until either connectivity fails, or the session is taken down administratively. If either the remote system signals Down state, or the Detection Time expires, the session advances to Down state.

AdminDown state means that the session is being held administratively down. This causes the remote system to enter Down state, and remain there until the local system exits AdminDown state. AdminDown state has no semantic implications for the availability of the forwarding path.

The following diagram provides an overview of the state machine. Transitions involving AdminDown state are deleted for clarity (but are fully specified in sections 6.8.6 and 6.8.16.) The notation on each arc represents the state of the remote system (as received in the State field in the BFD Control packet) or indicates the expiration of the Detection Timer.



6.3. Demultiplexing and the Discriminator Fields

Since multiple BFD sessions may be running between two systems, there needs to be a mechanism for demultiplexing received BFD packets to the proper session.

Each system MUST choose an opaque discriminator value that identifies each session, and which MUST be unique among all BFD sessions on the system. The local discriminator is sent in the My Discriminator field in the BFD Control packet, and is echoed back in the Your Discriminator field of packets sent from the remote end.

Once the remote end echoes back the local discriminator, all further received packets are demultiplexed based on the Your Discriminator field only (which means that, among other things, the source address field can change or the interface over which the packets are received can change, but the packets will still be associated with the proper session.)

The method of demultiplexing the initial packets (in which Your Discriminator is zero) is application-dependent, and is thus outside the scope of this specification.

Note that it is permissible for a system to change its discriminator during a session without affecting the session state, since only that system uses its discriminator for demultiplexing purposes (by having the other system reflect it back.) The implications on an implementation for changing the discriminator value is outside the scope of this specification.

6.4. The Echo Function and Asymmetry

The Echo function can be run independently in each direction between a pair of systems. For whatever reason, a system may advertise that it is willing to receive (and loop back) Echo packets, but may not wish to ever send any. The fact that a system is sending Echo packets is not directly signaled to the system looping them back.

When a system is using the Echo function, it is advantageous to choose a sedate reception rate for Control packets, since liveness detection is being handled by the Echo packets. This can be controlled by manipulating the Required Min RX Interval field (see section 6.8.3.)

If the Echo function is only being run in one direction, the system not running the Echo function will more likely wish to receive fairly rapid Control packets in order to achieve its desired Detection Time. Since BFD allows independent transmission rates in each direction, this is easily accomplished.

A system SHOULD otherwise advertise the lowest value of Required Min RX Interval and Required Min Echo RX Interval that it can under the circumstances, to give the other system more freedom in choosing its transmission rate. Note that a system is committing to be able to receive both streams of packets at the rate it advertises, so this should be taken into account when choosing the values to advertise.

6.5. The Poll Sequence

A Poll Sequence is an exchange of BFD Control packets that is used in some circumstances to ensure that the remote system is aware of parameter changes. It is also used in Demand mode (see section 6.6) to validate bidirectional connectivity.

A Poll Sequence consists of a system sending periodic BFD Control packets with the Poll (P) bit set. When the other system receives a Poll, it immediately transmits a BFD Control packet with the Final (F) bit set, independent of any periodic BFD Control packets it may be sending (see section 6.8.7). When the system sending the Poll sequence receives a packet with Final, the Poll Sequence is terminated, and any subsequent BFD Control packets are sent with the Poll bit cleared. A BFD Control packet MUST NOT have both the Poll (P) and Final (F) bits set.

If periodic BFD Control packets are already being sent (the remote system is not in Demand mode), the Poll Sequence MUST be performed by setting the Poll (P) bit on those scheduled periodic transmissions;

additional packets MUST NOT be sent.

After a Poll Sequence is terminated, the system requesting the Poll Sequence will cease the periodic transmission of BFD Control packets if the remote end is in Demand mode; otherwise it will return to the periodic transmission of BFD Control packets with the Poll (P) bit clear.

Typically, the entire sequence consists of a single packet in each direction, though packet losses or relatively long packet latencies may result in multiple Poll packets to be sent before the sequence terminates.

6.6. Demand Mode

Demand mode is requested independently in each direction by virtue of a system setting the Demand (D) bit in its BFD Control packets. The Demand bit can only be set if both systems think the session is up. The system receiving the Demand bit ceases the periodic transmission of BFD Control packets. If both systems are operating in Demand mode, no periodic BFD Control packets will flow in either direction.

Demand mode requires that some other mechanism is used to imply continuing connectivity between the two systems. The mechanism used does not have to be the same in both directions, and is outside of the scope of this specification. One possible mechanism is the receipt of traffic from the remote system; another is the use of the Echo function.

When a system in Demand mode wishes to verify bidirectional connectivity, it initiates a Poll Sequence (see section 6.5). If no response is received to a Poll, the Poll is repeated until the Detection Time expires, at which point the session is declared to be down. Note that if Demand mode is operating only on the local system, the Poll Sequence is performed by simply setting the Poll (P) bit in regular periodic BFD Control packets, as required by section 6.6.

The Detection Time in Demand mode is calculated differently than in Asynchronous mode; it is based on the transmit rate of the local system, rather than the transmit rate of the remote system. This ensures that the Poll Sequence mechanism works properly. See section 6.8.4 for more details.

Note that this mechanism requires that the Detection Time negotiated is greater than the round trip time between the two systems, or the Poll mechanism will always fail. Enforcement of this requirement is

outside the scope of this specification.

Demand mode MAY be enabled or disabled at any time, independently in each direction, by setting or clearing the Demand (D) bit in the BFD Control packet, without affecting the BFD session state. Note that the Demand bit MUST NOT be set unless both systems perceive the session to be Up (the local system thinks the session is Up, and the remote system last reported Up state in the State (Sta) field of the BFD Control packet.)

When the transmitted value of the Demand (D) bit is to be changed, the transmitting system MUST initiate a Poll Sequence in conjunction with changing the bit in order to ensure that both systems are aware of the change.

If Demand mode is active on either or both systems, a Poll Sequence MUST be initiated whenever the contents of the next BFD Control packet to be sent would be different than the contents of the previous packet, with the exception of the Poll (P) and Final (F) bits. This ensures that parameter changes are transmitted to the remote system and that the remote system acknowledges these changes.

Because the underlying detection mechanism is unspecified, and may differ between the two systems, the overall Detection Time characteristics of the path will not be fully known to either system. The total Detection Time for a particular system is the sum of the time prior to the initiation of the Poll Sequence, plus the calculated Detection Time.

Note that if Demand mode is enabled in only one direction, continuous bidirectional connectivity verification is lost (only connectivity in the direction from the system in Demand mode to the other system will be verified.) Resolving the issue of one system requesting Demand mode while the other requires continuous bidirectional connectivity verification is outside the scope of this specification.

6.7. Authentication

An optional Authentication Section may be present in the BFD Control packet. In its generic form, the purpose of the Authentication Section is to carry all necessary information, based on the authentication type in use, to allow the receiving system to determine the validity of the received packet. The exact mechanism depends on the authentication type in use, but in general the transmitting system will put information in the Authentication Section that vouches for the packet's validity, and the receiving system will examine the Authentication Section and either accept the

packet for further processing, or discard it.

Note that in the subsections below, to "accept" a packet means only that the packet has passed authentication; it may in fact be discarded for other reasons as described in the general packet reception rules described in section 6.8.6.

Implementations supporting authentication MUST support SHA1 authentication. Other forms of authentication are optional.

6.7.1. Enabling and Disabling Authentication

It may be desirable to enable or disable authentication on a session without disturbing the session state. The exact mechanism for doing so is outside the scope of this specification. However, it is useful to point out some issues in supporting this mechanism.

In a simple implementation, a BFD session will fail when authentication is either turned on or turned off, because the packet acceptance rules essentially require the local and remote machines to do so in a more or less synchronized fashion (within the Detection Time)--a packet with authentication will only be accepted if authentication is "in use" (and likewise packets without authentication.

One possible approach is to build an implementation such that authentication is configured, but not considered "in use" until the first packet containing a matching authentication section is received (providing the necessary synchronization.) Likewise, authentication could be configured off, but still considered "in use" until the receipt of the first packet without the authentication section.

In order to avoid security risks, implementations using this method should only allow the authentication state to be changed once without some form of intervention (so that authentication cannot be turned on and off repeatedly simply based on the receipt of BFD Control packets from remote systems.)

6.7.2. Simple Password Authentication

The most straightforward (and weakest) form of authentication is Simple Password Authentication. In this method of authentication, one or more Passwords (with corresponding Key IDs) are configured in each system and one of these Password/ID pairs is carried in each BFD Control packet. The receiving system accepts the packet if the Password and Key ID matches one of the Password/ID pairs configured

in that system.

Transmission Using Simple Password Authentication

The currently selected password and Key ID for the session MUST be stored in the Authentication Section of each outgoing BFD Control packet. The Auth Type field MUST be set to 1 (Simple Password.) The Auth Len field MUST be set to the proper length (4 to 19 bytes.)

Reception Using Simple Password Authentication

If the received BFD Control packet does not contain an Authentication Section, or the Auth Type is not 1 (Simple Password), then the received packet MUST be discarded.

If the Auth Key ID field does not match the ID of a configured password, the received packet MUST be discarded.

If the Auth Len field is not equal to the length of the password selected by the Key ID, plus three, the packet MUST be discarded.

If the Password field does not match the password selected by the Key ID, the packet MUST be discarded.

Otherwise, the packet MUST be accepted.

6.7.3. Keyed MD5 and Meticulous Keyed MD5 Authentication

The Keyed MD5 and Meticulous Keyed MD5 Authentication mechanisms are very similar to those used in other protocols. In these methods of authentication, one or more secret keys (with corresponding Key IDs) are configured in each system. One of the Keys is included in an MD5 [MD5] checksum calculated over the outgoing BFD Control packet, but the Key itself is not carried in the packet. To help avoid replay attacks, a sequence number is also carried in each packet. For Keyed MD5, the sequence number is occasionally incremented. For Meticulous Keyed MD5, the sequence number is incremented on every packet.

The receiving system accepts the packet if the Key ID matches one of the configured Keys, an MD5 checksum including the selected key matches that carried in the packet, and if the sequence number is greater than or equal to the last sequence number received (for Keyed MD5), or strictly greater than the last sequence number received (for Meticulous Keyed MD5.)

Transmission Using Keyed MD5 and Meticulous Keyed MD5 Authentication

The Auth Type field MUST be set to 2 (Keyed MD5) or 3 (Meticulous Keyed MD5.) The Auth Len field MUST be set to 24. The Auth Key ID field MUST be set to the ID of the current authentication key. The Sequence Number field MUST be set to bfd.XmitAuthSeq.

The current authentication key value MUST be placed into the Auth Key/Checksum field. An MD5 checksum MUST be calculated over the entire BFD control packet. The resulting checksum MUST be stored in the Auth Key/Checksum field prior to transmission (replacing the secret key, which MUST NOT be carried in the packet.)

For Keyed MD5, bfd.XmitAuthSeq MAY be incremented in a circular fashion (when treated as an unsigned 32 bit value.) bfd.XmitAuthSeq SHOULD be incremented when the session state changes, or when the transmitted BFD Control packet carries different contents than the previously transmitted packet. The decision as to when to increment bfd.XmitAuthSeq is outside the scope of this specification. See the section entitled "Security Considerations" below for a discussion.

For Meticulous Keyed MD5, bfd.XmitAuthSeq MUST be incremented in a circular fashion (when treated as an unsigned 32 bit value.)

Receipt Using Keyed MD5 and Meticulous Keyed MD5 Authentication

If the received BFD Control packet does not contain an Authentication Section, or the Auth Type is not correct (2 for Keyed MD5, or 3 for Meticulous Keyed MD5), then the received packet MUST be discarded.

If the Auth Key ID field does not match the ID of a configured authentication key, the received packet MUST be discarded.

If the Auth Len field is not equal to 24, the packet MUST be discarded.

Replace the contents of the Auth Key/Checksum field with the authentication key selected by the received Auth Key ID field. If the MD5 checksum of the entire BFD Control packet is not equal to the received value of the Auth Key/Checksum field, the received packet MUST be discarded.

If bfd.AuthSeqKnown is 1, examine the Sequence Number field. For Keyed MD5, if the Sequence Number lies outside of the range of bfd.RcvAuthSeq to bfd.RcvAuthSeq+(3*Detect Mult) inclusive (when

treated as an unsigned 32 bit circular number space), the received packet MUST be discarded. For Meticulous Keyed MD5, if the Sequence Number lies outside of the range of bfd.RcvAuthSeq+1 to bfd.RcvAuthSeq+(3*Detect Mult) inclusive (when treated as an unsigned 32 bit circular number space, the received packet MUST be discarded.

Otherwise (bfd.AuthSeqKnown is 0), bfd.AuthSeqKnown MUST be set to 1, bfd.RcvAuthSeq MUST be set to the value of the received Sequence Number field, and the received packet MUST be accepted.

6.7.4. Keyed SHA1 and Meticulous Keyed SHA1 Authentication

The Keyed SHA1 and Meticulous Keyed SHA1 Authentication mechanisms are very similar to those used in other protocols. In these methods of authentication, one or more secret keys (with corresponding Key IDs) are configured in each system. One of the Keys is included in a SHA1 [SHA1] checksum calculated over the outgoing BFD Control packet, but the Key itself is not carried in the packet. To help avoid replay attacks, a sequence number is also carried in each packet. For Keyed SHA1, the sequence number is occasionally incremented. For Meticulous Keyed SHA1, the sequence number is incremented on every packet.

The receiving system accepts the packet if the Key ID matches one of the configured Keys, a SHA1 checksum including the selected key matches that carried in the packet, and if the sequence number is greater than or equal to the last sequence number received (for Keyed SHA1), or strictly greater than the last sequence number received (for Meticulous Keyed SHA1.)

Transmission Using Keyed SHA1 and Meticulous Keyed SHA1 Authentication

The Auth Type field MUST be set to 4 (Keyed SHA1) or 5 (Meticulous Keyed SHA1.) The Auth Len field MUST be set to 28. The Auth Key ID field MUST be set to the ID of the current authentication key. The Sequence Number field MUST be set to bfd.XmitAuthSeq.

The current authentication key value MUST be placed into the Auth Key/Checksum field. A SHA1 checksum MUST be calculated over the entire BFD control packet. The resulting checksum MUST be stored in the Auth Key/Checksum field prior to transmission (replacing the secret key, which MUST NOT be carried in the packet.)

For Keyed SHA1, bfd.XmitAuthSeq MAY be incremented in a circular

fashion (when treated as an unsigned 32 bit value.)
bfd.XmitAuthSeq SHOULD be incremented when the session state changes, or when the transmitted BFD Control packet carries different contents than the previously transmitted packet. The decision as to when to increment bfd.XmitAuthSeq is outside the scope of this specification. See the section entitled "Security Considerations" below for a discussion.

For Meticulous Keyed SHA1, bfd.XmitAuthSeq MUST be incremented in a circular fashion (when treated as an unsigned 32 bit value.)

Receipt Using Keyed SHA1 and Meticulous Keyed SHA1 Authentication

If the received BFD Control packet does not contain an Authentication Section, or the Auth Type is not correct (4 for Keyed SHA1, or 5 for Meticulous Keyed SHA1), then the received packet MUST be discarded.

If the Auth Key ID field does not match the ID of a configured authentication key, the received packet MUST be discarded.

If the Auth Len field is not equal to 28, the packet MUST be discarded.

Replace the contents of the Auth Key/Checksum field with the authentication key selected by the received Auth Key ID field. If the SHA1 checksum of the entire BFD Control packet is not equal to the received value of the Auth Key/Checksum field, the received packet MUST be discarded.

If bfd.AuthSeqKnown is 1, examine the Sequence Number field. For Keyed SHA1, if the Sequence Number lies outside of the range of bfd.RcvAuthSeq to bfd.RcvAuthSeq+(3*Detect Mult) inclusive (when treated as an unsigned 32 bit circular number space), the received packet MUST be discarded. For Meticulous Keyed SHA1, if the Sequence Number lies outside of the range of bfd.RcvAuthSeq+1 to bfd.RcvAuthSeq+(3*Detect Mult) inclusive (when treated as an unsigned 32 bit circular number space, the received packet MUST be discarded.

Otherwise (bfd.AuthSeqKnown is 0), bfd.AuthSeqKnown MUST be set to 1, bfd.RcvAuthSeq MUST be set to the value of the received Sequence Number field, and the received packet MUST be accepted.

6.8. Functional Specifics

The following section of this specification is normative. The means by which this specification is achieved is outside the scope of this specification.

When a system is said to have "the Echo function active," it means that the system is sending BFD Echo packets, implying that the session is Up and the other system has signaled its willingness to loop back Echo packets.

When the local system is said to have "Demand mode active," it means that bfd.DemandMode is 1 in the local system (see section 6.8.1), the session is Up, and the remote system is signaling that the session is in state Up.

When the remote system is said to have "Demand mode active," it means that bfd.RemoteDemandMode is 1 (the remote system set the Demand (D) bit in the last received BFD Control packet), the session is Up, and the remote system is signaling that the session is in state Up.

6.8.1. State Variables

A minimum amount of information about a session needs to be tracked in order to achieve the elements of procedure described here. The following is a set of state variables that are helpful in describing the mechanisms of BFD. Any means of tracking this state may be used so long as the protocol behaves as described.

When the text refers to initializing a state variable, this takes place only at the time that the session (and the corresponding state variables) is created. The state variables are subsequently manipulated by the state machine and are never reinitialized, even if the session fails and is reestablished.

Once session state is created, and at least one BFD Control packet is received from the remote end, it MUST be preserved for at least one Detection Time (see section 6.8.4) subsequent to the receipt of the last BFD Control packet, regardless of the session state. This preserves timing parameters in case the session flaps. A system MAY preserve session state longer than this. The preservation or destruction of session state when no BFD Control packets for this session have been received from the remote system is outside the scope of this specification.

All state variables in this specification are of the form "bfd.Xx" and should not be confused with fields carried in the protocol

packets, which are always spelled out to match the names in section 4.

bfd.SessionState

The perceived state of the session (Init, Up, Down, or AdminDown.) The exact action taken when the session state changes is outside the scope of this specification, though it is expected that this state change (particularly to and from Up state) is reported to other components of the system. This variable MUST be initialized to Down.

bfd.RemoteSessionState

The session state last reported by the remote system in the State (Sta) field of the BFD Control packet. This variable MUST be initialized to Down.

bfd.LocalDiscr

The local discriminator for this BFD session, used to uniquely identify it. It MUST be unique across all BFD sessions on this system, and nonzero. It SHOULD be set to a random (but still unique) value to improve security. The value is otherwise outside the scope of this specification.

bfd.RemoteDiscr

The remote discriminator for this BFD session. This is the discriminator chosen by the remote system, and is totally opaque to the local system. This MUST be initialized to zero. If a period of a Detection Time passes without the receipt of a valid, authenticated BFD packet from the remote system, this variable MUST be set to zero.

bfd.LocalDiag

The diagnostic code specifying the reason for the most recent change in the local session state. This MUST be initialized to zero (No Diagnostic.)

bfd.DesiredMinTxInterval

The minimum interval, in microseconds, between transmitted BFD Control packets that this system would like to use at the current time. The actual interval is negotiated between the two systems. This MUST be initialized to a value of at least one second (1,000,000 microseconds) according to the rules described in section 6.8.3. The setting of this variable is otherwise outside the scope of this specification.

bfd.RequiredMinRxInterval

The minimum interval, in microseconds, between received BFD Control packets that this system requires. The setting of this variable is outside the scope of this specification. A value of zero means that this system does not want to receive any periodic BFD Control packets. See section 6.8.18 for details.

bfd.RemoteMinRxInterval

The last value of Required Min RX Interval received from the remote system in a BFD Control packet. This variable MUST be initialized to 1.

bfd.DemandMode

Set to 1 if the local system wishes to use Demand mode, or 0 if not.

bfd.RemoteDemandMode

Set to 1 if the remote system wishes to use Demand mode, or 0 if not. This is the value of the Demand (D) bit in the last received BFD Control packet. This variable MUST be initialized to zero.

bfd.DetectMult

The desired Detection Time multiplier for BFD Control packets. The negotiated Control packet transmission interval, multiplied by this variable, will be the Detection Time for this session (as seen by the remote system.) This variable MUST be a

nonzero integer, and is otherwise outside the scope of this specification. See section 6.8.4 for further information.

bfd.AuthType

The authentication type in use for this session, as defined in section 4.1, or zero if no authentication is in use.

bfd.RcvAuthSeq

A 32 bit unsigned integer containing the next sequence number for keyed MD5 or SHA1 authentication expected to be received. The initial value is unimportant.

bfd.XmitAuthSeq

A 32 bit unsigned integer containing the next sequence number for keyed MD5 or SHA1 authentication to be transmitted. This variable MUST be initialized to a random 32 bit value.

bfd.AuthSeqKnown

Set to 1 if the next sequence number for keyed MD5 or SHA1 authentication expected to be received is known, or 0 if it is not known. This variable MUST be initialized to zero.

This variable MUST be set to zero after no packets have been received on this session for at least twice the Detection Time. This ensures that the sequence number can be resynchronized if the remote system restarts.

6.8.2. Timer Negotiation

The time values used to determine BFD packet transmission intervals and the session Detection Time are continuously negotiated, and thus may be changed at any time. The negotiation and time values are independent in each direction for each session.

Each system reports in the BFD Control packet how rapidly it would like to transmit BFD packets, as well as how rapidly it is prepared to receive them. With the exceptions listed in the remainder of this section, a system MUST NOT transmit BFD Control packets at an interval less than the larger of bfd.DesiredMinTxInterval and

bfd.RemoteMinRxInterval. In other words, the system reporting the slower rate determines the transmission rate.

The periodic transmission of BFD Control packets SHOULD be jittered by up to 25%, that is, the interval SHOULD be reduced by a random value of 0 to 25%, in order to avoid self-synchronization. Thus, the average interval between packets may be up to 12.5% less than that negotiated.

If bfd.DetectMult is equal to 1, the interval between transmitted BFD Control packets MUST be no more than 90% of the negotiated transmission interval, and MUST be no less than 75% of the negotiated transmission interval. This is to ensure that, on the remote system, the calculated DetectTime does not pass prior to the receipt of the next BFD Control packet.

6.8.3. Timer Manipulation

The time values used to determine BFD packet transmission intervals and the session Detection Time may be modified at any time without affecting the state of the session. When the timer parameters are changed for any reason, the requirements of this section apply.

If either bfd.DesiredMinTxInterval is changed or bfd.RequiredMinRxInterval is changed, a Poll Sequence MUST be initiated (see section 6.5). If the timing is such that a system receiving a Poll Sequence wishes to change the parameters described in this paragraph, the new parameter values may be carried in packets with the Final (F) bit set, even if the Poll Sequence has not yet been sent.

If bfd.DesiredMinTxInterval is increased and bfd.SessionState is Up, the actual transmission interval used MUST NOT change until the Poll Sequence described above has terminated. This is to ensure that the remote system updates its Detection Time before the transmission interval increases.

If bfd.RequiredMinRxInterval is reduced and bfd.SessionState is Up, the previous value of bfd.RequiredMinRxInterval MUST be used when calculating the Detection Time for the remote system until the Poll Sequence described above has terminated. This is to ensure that the remote system is transmitting packets at the higher rate (and those packets are being received) prior to the Detection Time being reduced.

When bfd.SessionState is not Up, the system MUST set bfd.DesiredMinTxInterval to a value of not less than one second

(1,000,000 microseconds.) This is intended to ensure that the bandwidth consumed by BFD sessions that are not Up is negligible, particularly in the case where a neighbor may not be running BFD.

If the local system reduces its transmit interval due to bfd.RemoteMinRxInterval being reduced (the remote system has advertised a reduced value in Required Min RX Interval), and the remote system is not in Demand mode, the local system MUST honor the new interval immediately. In other words, the local system cannot wait longer than the new interval between the previous packet transmission and the next one. If this interval has already passed since the last transmission (because the new interval is significantly shorter), the local system MUST send the next periodic BFD Control packet as soon as practicable.

When the Echo function is active, a system SHOULD set bfd.RequiredMinRxInterval to a value of not less than one second (1,000,000 microseconds.) This is intended to keep received BFD Control traffic at a negligible level, since the actual detection function is being performed using BFD Echo packets.

In any case other than those explicitly called out above, timing parameter changes MUST be effected immediately (changing the transmission rate and/or the Detection Time).

Note that the Poll Sequence mechanism is ambiguous if more than one parameter change is made that would require its use, and those multiple changes are spread across multiple packets (since the semantics of the returning Final are unclear.) Therefore, if multiple changes are made that require the use of a Poll Sequence, there are three choices: 1) they MUST be communicated in a single BFD Control packet (so the semantics of the Final reply are clear), or 2) sufficient time must have transpired since the Poll Sequence was completed to disambiguate the situation (at least a round trip time since the last Poll was transmitted) prior to the initiation of another Poll Sequence, or 3) an additional BFD Control packet with the Final (F) bit *clear* MUST be received after the Poll Sequence has completed prior to the initiation of another Poll Sequence (this option is not available when Demand mode is active.)

6.8.4. Calculating the Detection Time

The Detection Time (the period of time without receiving BFD packets after which the session is determined to have failed) is not carried explicitly in the protocol. Rather, it is calculated independently in each direction by the receiving system based on the negotiated transmit interval and the detection multiplier. Note that there may be different Detection Times in each direction.

The calculation of the Detection Time is slightly different when in Demand mode versus Asynchronous mode.

In Asynchronous mode, the Detection Time calculated in the local system is equal to the value of Detect Mult received from the remote system, multiplied by the agreed transmit interval of the remote system (the greater of bfd.RequiredMinRxInterval and the last received Desired Min TX Interval.) The Detect Mult value is (roughly speaking, due to jitter) the number of packets that have to be missed in a row to declare the session to be down.

If Demand mode is not active, and a period of time equal to the Detection Time passes without receiving a BFD Control packet from the remote system, and bfd.SessionState is Init or Up, the session has gone down--the local system MUST set bfd.SessionState to Down and bfd.LocalDiag to 1 (Control Detection Time Expired.)

In Demand mode, the Detection Time calculated in the local system is equal to bfd.DetectMult, multiplied by the agreed transmit interval of the local system (the greater of bfd.DesiredMinTxInterval and bfd.RemoteMinRxInterval.) bfd.DetectMult is (roughly speaking, due to jitter) the number of packets that have to be missed in a row to declare the session to be down.

If Demand mode is active, and a period of time equal to the Detection Time passes after the initiation of a Poll Sequence (the transmission of the first BFD Control packet with the Poll bit set), the session has gone down--the local system MUST set bfd.SessionState to Down, and bfd.LocalDiag to 1 (Control Detection Time Expired.)

(Note that a packet is considered to have been received, for the purposes of Detection Time expiration, only if it has not been "discarded" according to the rules of section 6.8.6.)

6.8.5. Detecting Failures with the Echo Function

When the Echo function is active and a sufficient number of Echo packets have not arrived as they should, the session has gone down--the local system MUST set bfd.SessionState to Down, and bfd.LocalDiag to 2 (Echo Function Failed.)

The means by which the Echo function failures are detected is outside of the scope of this specification. Any means which will detect a communication failure is acceptable.

6.8.6. Reception of BFD Control Packets

When a BFD Control packet is received, the following procedure MUST be followed, in the order specified. If the packet is discarded according to these rules, processing of the packet MUST cease at that point.

If the version number is not correct (1), the packet MUST be discarded.

If the Length field is less than the minimum correct value (24 if the A bit is clear, or 26 if the A bit is set), the packet MUST be discarded.

If the Length field is greater than the payload of the encapsulating protocol, the packet MUST be discarded.

If the Detect Mult field is zero, the packet MUST be discarded.

If the Multipoint (M) bit is nonzero, the packet MUST be discarded.

If the My Discriminator field is zero, the packet MUST be discarded.

If the Your Discriminator field is nonzero, it MUST be used to select the session with which this BFD packet is associated. If no session is found, the packet MUST be discarded.

If the Your Discriminator field is zero and the State field is not Down or AdminDown, the packet MUST be discarded.

If the Your Discriminator field is zero, the session MUST be selected based on some combination of other fields, possibly including source addressing information, the My Discriminator field, and the interface over which the packet was received. The exact method of selection is application-specific and is thus outside the scope of this specification. If a matching session is not found, a new session may be created, or the packet may be discarded. This choice is outside the scope of this specification.

If the A bit is set and no authentication is in use (bfd.AuthType is zero), the packet MUST be discarded.

If the A bit is clear and authentication is in use (bfd.AuthType is nonzero), the packet MUST be discarded.

```
      If the A bit is set, the packet MUST be authenticated under the
      rules of section 6.7, based on the authentication type in use
      (bfd.AuthType.)  This may cause the packet to be discarded.

      Set bfd.RemoteDiscr to the value of My Discriminator.

      Set bfd.RemoteState to the value of the State (Sta) field.

      Set bfd.RemoteDemandMode to the value of the Demand (D) bit.

      Set bfd.RemoteMinRxInterval to the value of Required Min RX
      Interval.

      If the Required Min Echo RX Interval field is zero, the
      transmission of Echo packets, if any, MUST cease.

      If a Poll Sequence is being transmitted by the local system and
      the Final (F) bit in the received packet is set, the Poll Sequence
      MUST be terminated.

      Update the transmit interval as described in section 6.8.2.

      Update the Detection Time as described in section 6.8.4.

      If bfd.SessionState is AdminDown
          Discard the packet

      If received state is AdminDown
          If bfd.SessionState is not Down
              Set bfd.LocalDiag to 3 (Neighbor signaled session down)
              Set bfd.SessionState to Down

      Else

          If bfd.SessionState is Down
              If received State is Down
                  Set bfd.SessionState to Init
              Else if received State is Init
                  Set bfd.SessionState to Up

          Else if bfd.SessionState is Init
              If received State is Init or Up
                  Set bfd.SessionState to Up

          Else (bfd.SessionState is Up)
              If received State is Down
                  Set bfd.LocalDiag to 3 (Neighbor signaled session down)
                  Set bfd.SessionState to Down
```

Check to see if Demand mode should become active or not (see section 6.6).

If bfd.RemoteDemandMode is 1, bfd.SessionState is Up, and bfd.RemoteSessionState is Up, Demand mode is active on the remote system and the local system MUST cease the periodic transmission of BFD Control packets (see section 6.8.7.)

If bfd.RemoteDemandMode is 0, or bfd.SessionState is not Up, or bfd.RemoteSessionState is not Up, Demand mode is not active on the remote system and the local system MUST send periodic BFD Control packets (see section 6.8.7.)

If the Poll (P) bit is set, send a BFD Control packet to the remote system with the Poll (P) bit clear, and the Final (F) bit set (see section 6.8.7.)

If the packet was not discarded, it has been received for purposes of the Detection Time expiration rules in section 6.8.4.

6.8.7. Transmitting BFD Control Packets

BFD Control packets MUST be transmitted periodically at the rate determined according to section 6.8.2, except as specified in this section.

The transmit interval MUST be recalculated whenever bfd.DesiredMinTxInterval changes, or whenever bfd.RemoteMinRxInterval changes, and is equal to the greater of those two values. See sections 6.8.2 and 6.8.3 for details on transmit timers.

A system MUST NOT transmit BFD Control packets if bfd.RemoteDiscr is zero and the system is taking the Passive role.

A system MUST NOT periodically transmit BFD Control packets if bfd.RemoteMinRxInterval is zero.

A system MUST NOT periodically transmit BFD Control packets if Demand mode is active on the remote system (bfd.RemoteDemandMode is 1, bfd.SessionState is Up, and bfd.RemoteSessionState is Up) and a Poll Sequence is not being transmitted.

If a BFD Control packet is received with the Poll (P) bit set to 1, the receiving system MUST transmit a BFD Control packet with the Poll (P) bit clear and the Final (F) bit set as soon as practicable, without respect to the transmission timer or any other transmission limitations, without respect to the session state, and without

respect to whether Demand mode is active on either system. A system MAY limit the rate at which such packets are transmitted. If rate limiting is in effect, the advertised value of Desired Min TX Interval MUST be greater than or equal to the interval between transmitted packets imposed by the rate limiting function.

A system MUST NOT set the Demand (D) bit unless bfd.DemandMode is 1, bfd.SessionState is Up, and bfd.RemoteSessionState is Up.

A BFD Control packet SHOULD be transmitted during the interval between periodic Control packet transmissions when the contents of that packet would differ from that in the previously transmitted packet (other than the Poll and Final bits) in order to more rapidly communicate a change in state.

The contents of transmitted BFD Control packets MUST be set as follows:

Version

Set to the current version number (1).

Diagnostic (Diag)

Set to bfd.LocalDiag.

State (Sta)

Set to the value indicated by bfd.SessionState.

Poll (P)

Set to 1 if the local system is sending a Poll Sequence, or 0 if not.

Final (F)

Set to 1 if the local system is responding to a Control packet received with the Poll (P) bit set, or 0 if not.

Control Plane Independent (C)

Set to 1 if the local system's BFD implementation is independent of the control plane (it can continue to function through a disruption of the control plane.)

Authentication Present (A)

Set to 1 if authentication is in use on this session (bfd.AuthType is nonzero), or 0 if not.

Demand (D)

Set to bfd.DemandMode if bfd.SessionState is Up and bfd.RemoteSessionState is Up. Otherwise it is set to 0.

Multipoint (M)

Set to 0.

Detect Mult

Set to bfd.DetectMult.

Length

Set to the appropriate length, based on the fixed header length (24) plus any Authentication Section.

My Discriminator

Set to bfd.LocalDiscr.

Your Discriminator

Set to bfd.RemoteDiscr.

Desired Min TX Interval

Set to bfd.DesiredMinTxInterval.

Required Min RX Interval

Set to bfd.RequiredMinRxInterval.

Required Min Echo RX Interval

Set to the minimum required Echo packet receive interval for this session. If this field is set to zero, the local system is unwilling or unable to loop back BFD Echo packets to the remote system, and the remote system will not send Echo packets.

Authentication Section

Included and set according to the rules in section 6.7 if authentication is in use (bfd.AuthType is nonzero.) Otherwise this section is not present.

6.8.8. Reception of BFD Echo Packets

A received BFD Echo packet MUST be demultiplexed to the appropriate session for processing. A means of detecting missing Echo packets MUST be implemented, which most likely involves processing of the Echo packets that are received. The processing of received Echo packets is otherwise outside the scope of this specification.

6.8.9. Transmission of BFD Echo Packets

BFD Echo packets MUST NOT be transmitted when bfd.SessionState is not Up. BFD Echo packets MUST NOT be transmitted unless the last BFD Control packet received from the remote system contains a nonzero value in Required Min Echo RX Interval.

BFD Echo packets MAY be transmitted when bfd.SessionState is Up. The interval between transmitted BFD Echo packets MUST NOT be less than the value advertised by the remote system in Required Min Echo RX Interval, except as follows:

A 25% jitter MAY be applied to the rate of transmission, such that the actual interval MAY be between 75% and 100% of the advertised

value. A single BFD Echo packet MAY be transmitted between normally scheduled Echo transmission intervals.

The transmission of BFD Echo packets is otherwise outside the scope of this specification.

6.8.10. Min Rx Interval Change

When it is desired to change the rate at which BFD Control packets arrive from the remote system, bfd.RequiredMinRxInterval can be changed at any time to any value. The new value will be transmitted in the next outgoing Control packet, and the remote system will adjust accordingly. See section 6.8.3 for further requirements.

6.8.11. Min Tx Interval Change

When it is desired to change the rate at which BFD Control packets are transmitted to the remote system (subject to the requirements of the neighboring system), bfd.DesiredMinTxInterval can be changed at any time to any value. The rules in section 6.8.3 apply.

6.8.12. Detect Multiplier Change

When it is desired to change the detect multiplier, the value of bfd.DetectMult can be changed to any nonzero value. The new value will be transmitted with the next BFD Control packet, and the use of a Poll Sequence is not necessary. See section 6.6 for additional requirements.

6.8.13. Enabling or Disabling The Echo Function

If it is desired to start or stop the transmission of BFD Echo packets, this MAY be done at any time (subject to the transmission requirements detailed in section 6.8.9.)

If it is desired to enable or disable the looping back of received BFD Echo packets, this MAY be done at any time by changing the value of Required Min Echo RX Interval to zero or nonzero in outgoing BFD Control packets.

6.8.14. Enabling or Disabling Demand Mode

If it is desired to start or stop Demand mode, this MAY be done at any time by setting bfd.DemandMode to the proper value. Demand mode will subsequently become active under the rules described in section 6.6.

If Demand mode is no longer active on the remote system, the local system MUST begin transmitting periodic BFD Control packets as described in section 6.8.7.

6.8.15. Forwarding Plane Reset

When the forwarding plane in the local system is reset for some reason, such that the remote system can no longer rely on the local forwarding state, the local system MUST set bfd.LocalDiag to 4 (Forwarding Plane Reset), and set bfd.SessionState to Down.

6.8.16. Administrative Control

There may be circumstances where it is desirable to administratively enable or disable a BFD session. When this is desired, the following procedure MUST be followed:

```
If enabling session
   Set bfd.SessionState to Down

Else
   Set bfd.SessionState to AdminDown
   Set bfd.LocalDiag to an appropriate value
   Cease the transmission of BFD Echo packets
```

If signaling is received from outside BFD that the underlying path has failed, an implementation MAY administratively disable the session with the diagnostic Path Down.

Other scenarios MAY use the diagnostic Administratively Down.

BFD Control packets SHOULD be transmitted for at least a Detection Time after transitioning to AdminDown state in order to ensure that the remote system is aware of the state change. BFD Control packets MAY be transmitted indefinitely after transitioning to AdminDown state in order to maintain session state in each system (see section 6.8.18 below.)

6.8.17. Concatenated Paths

If the path being monitored by BFD is concatenated with other paths, it may be desirable to propagate the indication of a failure of one of those paths across the BFD session (providing an interworking function for liveness monitoring between BFD and other technologies.)

Two diagnostic codes are defined for this purpose: Concatenated Path Down and Reverse Concatenated Path Down. The first propagates forward path failures (in which the concatenated path fails in the direction toward the interworking system), and the second propagates reverse path failures (in which the concatenated path fails in the direction away from the interworking system, assuming a bidirectional link.)

A system MAY signal one of these failure states by simply setting bfd.LocalDiag to the appropriate diagnostic code. Note that the BFD session is not taken down. If Demand mode is not active on the remote system, no other action is necessary, as the diagnostic code will be carried via the periodic transmission of BFD Control packets. If Demand mode is active on the remote system (the local system is not transmitting periodic BFD Control packets), a Poll Sequence MUST be initiated to ensure that the diagnostic code is transmitted. Note that if the BFD session subsequently fails, the diagnostic code will be overwritten with a code detailing the cause of the failure. It is up to the interworking agent to perform the above procedure again, once the BFD session reaches Up state, if the propagation of the concatenated path failure is to resume.

6.8.18. Holding Down Sessions

A system may choose to prevent a BFD session from being established. One possible reason might be to manage the rate at which sessions are established. This can be done by holding the session in Down or AdminDown state, as appropriate.

There are two related mechanisms that are available to help with this task. First, a system is required to maintain session state (including timing parameters), even when a session is down, until a Detection Time has passed without the receipt of any BFD Control packets. This means that a system may take down a session and transmit an arbitrarily large value in the Required Min RX Interval field to control the rate at which it receives packets.

Additionally, a system may transmit a value of zero for Required Min RX Interval to indicate that the remote system should send no packets whatsoever.

So long as the local system continues to transmit BFD Control packets, the remote system is obligated to obey the value carried in Required Min RX Interval. If the remote system does not receive any BFD Control packets for a Detection Time, it resets bfd.RemoteMinRxIvl to a small value and then can transmit at its own rate.

Backward Compatibility (Non-Normative)

Although Version 0 of this document is unlikely to have been deployed widely, some implementors may wish to have a backward compatibility mechanism. Note that any mechanism may be potentially used that does not alter the protocol definition, so interoperability should not be an issue.

The suggested mechanism described here has the property that it will converge on version 1 if both systems implement it, even if one system is upgraded from version 0 within a Detection Time. It will interoperate with a system that implements only one version (or is configured to support only one version.) A system should obviously not perform this function if it is configured to or is only capable of using a single version.

A BFD session will enter a "negotiation holddown" if it is configured for automatic versioning and either has just started up, or the session has been manually cleared. The session is set to AdminDown state and Version 1. During the holddown period, which lasts for one Detection Time, the system sends BFD Control packets as usual, but ignores received packets. After the holddown time is complete, the state transitions to Down and normal operation resumes.

When a system is not in holddown, if it doing automatic versioning and is currently using Version 1, if any Version 0 packet is received for the session, it switches immediately to Version 0. If it is currently using Version 0 and a Version 1 packet is received that indicates that the neighbor is in state AdminDown, it switches to Version 1. If using Version 0 and a Version 1 packet is received indicating a state other than AdminDown, the packet is ignored (per spec.)

If the version being used is changed, the session goes down as appropriate for the new version (Down state for Version 1 or Failing state for Version 0.)

Contributors

Kireeti Kompella and Yakov Rekhter of Juniper Networks were also significant contributors to this document.

Acknowledgments

This document was inspired by (and is intended to replace) the Protocol Liveness Protocol draft, written by Kireeti Kompella.

Demand mode was inspired by draft-ietf-ipsec-dpd-03.txt, by G. Huang et al.

The authors would also like to thank Mike Shand, John Scudder, Stewart Bryant, Pekka Savola, and Richard Spencer for their substantive input.

The authors would also like to thank Owen Wheeler for hosting teleconferences between the authors of this specification and multiple vendors in order address implementation and clarity issues.

Security Considerations

As BFD may be tied into the stability of the network infrastructure (such as routing protocols), the effects of an attack on a BFD session may be very serious. This ultimately has denial-of-service effects, as links may be declared to be down (or falsely declared to be up.)

When BFD is run over network layer protocols, a significant denial-of-service risk is created, as BFD packets may be trivial to spoof. When the session is directly connected across a single link (physical, or a secure tunnel such as IPsec), the TTL or Hop Count MUST be set to the maximum on transmit, and checked to be equal to the maximum value on reception (and the packet dropped if this is not the case.) See [GTSM] for more information on this technique. If BFD is run across multiple hops or an insecure tunnel (such as GRE), the Authentication Section SHOULD be utilized.

The level of security provided by the Authentication Section varies based on the authentication type used. Simple Password authentication is obviously only as secure as the secrecy of the passwords used, and should be considered only if the BFD session is guaranteed to be run over an infrastructure not subject to packet

interception. Its chief advantage is that it minimizes the computational effort required for authentication.

Keyed MD5 authentication is much stronger than Simple Password authentication since the keys cannot be discerned by intercepting packets. It is vulnerable to replay attacks in between increments of the sequence number. The sequence number can be incremented as seldom (or as often) as desired, trading off resistance to replay attacks with the computational effort required for authentication.

Meticulous Keyed MD5 authentication is stronger yet, as it requires the sequence number to be incremented for every packet. Replay attack vulnerability is reduced due to the requirement that the sequence number must be incremented on every packet, the window size of acceptable packets is small, and the initial sequence number is randomized. There is still a window of attack at the beginning of the session while the sequence number is being determined. This authentication scheme requires an MD5 calculation on every packet transmitted and received.

Using SHA1 rather than MD5 is believed to have stronger security properties. All comments about MD5 in this section also apply to SHA1.

If both systems randomize their Local Discriminator values at the beginning of a session, replay attacks may be further mitigated, regardless of the authentication type in use. Since the Local Discriminator may be changed at any time during a session, this mechanism may also help mitigate attacks.

IANA Considerations

This document has no actions for IANA.

Normative References

[GTSM] Gill, V., et al, "The Generalized TTL Security Mechanism (GTSM)", RFC 5082, October 2007.

[KEYWORD] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

[MD5] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.

[OSPF] Moy, J., "OSPF Version 2", RFC 2328, April 1998.

[SHA1] "Secure Hash Standard", United States of America, National Institute of Science and Technology, Federal Information Processing Standard (FIPS) 180-1, April 1993.

Authors' Addresses

Dave Katz
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, California 94089-1206 USA
Phone: +1-408-745-2000
Email: dkatz@juniper.net

Dave Ward
Cisco Systems
170 W. Tasman Dr.
San Jose, CA 95134 USA
Phone: +1-408-526-4000
Email: dward@cisco.com

Changes from the Previous Draft

All changes are purely editorial in nature.

IPR Notice

found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Full Copyright Notice

Acknowledgement

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

This document expires in September, 2008.

Network Working Group D. Katz
Cisco Systems

Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [KEYWORDS].

1. Introduction

The Bidirectional Forwarding Detection protocol [BFD] provides a liveness detection mechanism that can be utilized by other network components for which their integral liveness mechanisms are either too slow, inappropriate, or nonexistent. Other drafts have detailed the use of BFD with specific encapsulations ([BFD-1HOP], [BFD-MULTI], [BFD-MPLS]). As the utility of BFD has become understood, there have been calls to specify BFD interactions with a growing list of network functions. Rather than producing a long series of short documents on the application of BFD, it seemed worthwhile to describe the interactions between BFD and other network functions ("BFD clients") in a broad way.

This document describes the generic application of BFD. Specific protocol applications are provided for illustrative purposes.

2. Overview

The Bidirectional Forwarding Detection (BFD) specification defines a protocol with simple and specific semantics. Its sole purpose is to verify connectivity between a pair of systems, for a particular data protocol across a path (which may be of any technology, length, or OSI layer). The promptness of the detection of a path failure can be controlled by trading off protocol overhead and system load with detection times.

BFD is *not* intended to directly provide control protocol liveness information; those protocols have their own means and vagaries. Rather, control protocols can use the services provided by BFD to inform their operation. BFD can be viewed as a service provided by the layer in which it is running.

The service interface with BFD is straightforward. The application supplies session parameters (neighbor address, time parameters, protocol options), and BFD provides the session state, of which the most interesting transitions are to and from the Up state. The application is expected to bootstrap the BFD session, as BFD has no

discovery mechanism.

An implementation SHOULD establish only a single BFD session per data protocol path, regardless of the number of applications that wish to utilize it. There is no additional value in having multiple BFD sessions to the same endpoints. If multiple applications request different session parameters, it is a local issue as to how to resolve the parameter conflicts. BFD in turn will notify all applications bound to a session when a session state change occurs.

BFD should be viewed as having an advisory role to the protocol or protocols or other network functions with which it is interacting, which will then use their own mechanisms to effect any state transitions. The interaction is very much at arm's length, which keeps things simple and decoupled. In particular, BFD explicitly does not carry application-specific information, partly for architectural reasons, and partly because BFD may have curious and unpredictable latency characteristics and as such makes a poor transport mechanism.

It is important to remember that the interaction between BFD and its client applications has essentially no interoperability issues, because BFD is acting in an advisory nature (similar to hardware signaling the loss of light on a fiber optic circuit, for example) and existing mechanisms in the client applications are used in reaction to BFD events. In fact, BFD may interact with only one of a pair of systems for a particular client application without any ill effect.

3. Basic Interaction Between BFD Sessions and Clients

The interaction between a BFD session and its associated client applications is, for the most part, an implementation issue that is outside the scope of this specification. However, it is useful to describe some mechanisms that implementors may use in order to promote full-featured implementations. One way of modeling this interaction is to create an adaptation layer between the BFD state machine and the client applications. The adaptation layer is cognizant of both the internals of the BFD implementation and the requirements of the clients.

3.1. Session State Hysteresis

A BFD session can be tightly coupled to its client applications; for example, any transition out of the Up state could cause signaling to the clients to take failure action. But in some cases this may not always be the best course of action.

Implementors may choose to hide rapid Up/Down/Up transitions of the BFD session from its clients. This is useful in order to support process restarts without necessitating complex protocol mechanisms, for example.

As such, a system MAY choose not to notify clients if a BFD session transitions from Up to Down state, and returns to Up state, if it does so within a reasonable period of time (the length of which is outside the scope of this specification.) If the BFD session does not return to Up state within that timeframe, the clients SHOULD be notified that a session failure has occurred.

3.2. AdminDown State

The AdminDown mechanism in BFD is intended to signal that the BFD session is being taken down for administrative purposes, and the session state is not indicative of the liveness of the data path.

Therefore, a system SHOULD NOT indicate a connectivity failure to a client if either the local session state or the remote session state (if known) transitions to AdminDown, so long as that client has independent means of liveness detection (typically, control protocols.)

If a client does not have any independent means of liveness detection, a system SHOULD indicate a connectivity failure to a client, and assume the semantics of Down state, if either the local or remote session state transitions to AdminDown. Otherwise, the client will not be able to determine whether the path is viable, and unfortunate results may occur.

3.3. Hitless Establishment/Reestablishment of BFD State

It is useful to be able to configure a BFD session between a pair of systems without impacting the state of any clients that will be associated with that session. Similarly, it is useful for BFD state to be reestablished without perturbing associated clients when all BFD state is lost (such as in process restart situations.) This interacts with the clients' ability to establish their state

independent of BFD.

The BFD state machine transitions that occur in the process of bringing up a BFD session in such situations SHOULD NOT cause a connectivity failure notification to the clients.

A client which is capable of establishing its state prior to the configuration or restarting of a BFD session MAY do so if appropriate. The means to do so is outside of the scope of this specification.

4. Control Protocol Interactions

Very common client applications of BFD are control protocols, such as routing protocols. The object when BFD interacts with a control protocol is to advise the control protocol of the connectivity of the data protocol. In the case of routing protocols, for example, this allows the connectivity failure to trigger the rerouting of traffic around the failed path more quickly than the native detection mechanisms.

4.1. Adjacency Establishment

If the session state on either the local or remote system (if known) is AdminDown, BFD has been administratively disabled, and the establishment of a control protocol adjacency MUST be allowed.

BFD sessions are typically bootstrapped by the control protocol, using the mechanism (discovery, configuration) used by the control protocol to find neighbors. Note that it is possible in some failure scenarios for the network to be in a state such that the control protocol is capable of coming up, but the BFD session cannot be established, and, more particularly, data cannot be forwarded. To avoid this situation, it would be beneficial to not allow the control protocol to establish a neighbor adjacency. However, this would preclude the operation of the control protocol in an environment in which not all systems support BFD.

Therefore, the establishment of control protocol adjacencies SHOULD be blocked if both systems are willing to establish a BFD session but a BFD session cannot be established. A system is known to be willing to establish a BFD session if the control protocol carries signaling that indicates that both systems are willing to establish a BFD session, or it is known that the remote system is BFD-capable and BFD-enabled (the means of determining this are outside the scope of

this specification.)

If it is believed that the neighboring system does not support BFD, the establishment of a control protocol adjacency SHOULD NOT be blocked.

The setting of BFD's various timing parameters and modes are not subject to standardization. Note that all protocols sharing a session will operate using the same parameters. The mechanism for choosing the parameters among those desired by the various protocols are outside the scope of this specification. It is generally useful to choose the parameters resulting in the shortest Detection Time; a particular client application can always apply hysteresis to the notifications from BFD if it desires longer Detection Times.

Note that many control protocols assume full connectivity between all systems on multiaccess media such as LANs. If BFD is running on only a subset of systems on such a network, and adjacency establishment is blocked by the absence of a BFD session, the assumptions of the control protocol may be violated, with unpredictable results.

4.2. Reaction to BFD Session State Changes

If a BFD session transitions from state Up to AdminDown, or the session transitions from Up to Down because the remote system is indicating that the session is in state AdminDown, clients SHOULD NOT take any control protocol action.

Otherwise, the mechanism by which the control protocol reacts to a path failure signaled by BFD depends on the capabilities of the protocol.

4.2.1. Control Protocols with a Single Data Protocol

A control protocol that is tightly bound to a single failing data protocol SHOULD take action to ensure that data traffic is no longer directed to the failing path. Note that this should not be interpreted as BFD replacing the control protocol liveness mechanism, if any, as the control protocol may rely on mechanisms not verified by BFD (multicast, for instance) so BFD most likely cannot detect all failures that would impact the control protocol. However, a control protocol MAY choose to use BFD session state information to more rapidly detect an impending control protocol failure, particularly if the control protocol operates in-band (over the data protocol.)

Therefore, when a BFD session transitions from Up to Down, action

SHOULD be taken in the control protocol to signal the lack of connectivity for the path over which BFD is running. If the control protocol has an explicit mechanism for announcing path state, a system SHOULD use that mechanism rather than impacting the connectivity of the control protocol, particularly if the control protocol operates out-of-band from the failed data protocol. However, if such a mechanism is not available, a control protocol timeout SHOULD be emulated for the associated neighbor.

4.2.2. Control Protocols with Multiple Data Protocols

Slightly different mechanisms are used if the control protocol supports the routing of multiple data protocols, depending on whether the control protocol supports separate topologies for each data protocol.

4.2.2.1. Shared Topologies

With a shared topology, if one of the data protocols fails (as signaled by the associated BFD session), it is necessary to consider the path to have failed for all data protocols. Otherwise, there is no way for the control protocol to turn away traffic for the failed data protocol (and such traffic would be black-holed indefinitely.)

Therefore, when a BFD session transitions from Up to Down, action SHOULD be taken in the control protocol to signal the lack of connectivity for the path in the topology corresponding to the BFD session. If this cannot be signaled otherwise, a control protocol timeout SHOULD be emulated for the associated neighbor.

4.2.2.2. Independent Topologies

With individual routing topologies for each data protocol, only the failed data protocol needs to be rerouted around the failed path.

Therefore, when a BFD session transitions from Up to Down, action SHOULD be taken in the control protocol to signal the lack of connectivity for the path in the topology over which BFD is running. Generally this can be done without impacting the connectivity of other topologies (since otherwise it is very difficult to support separate topologies for multiple data protocols.)

4.3. Interactions with Graceful Restart Mechanisms

A number of control protocols support Graceful Restart mechanisms. These mechanisms are designed to allow a control protocol to restart without perturbing network connectivity state (lest it appear that the system and/or all of its links had failed.) They are predicated on the existence of a separate forwarding plane that does not necessarily share fate with the control plane in which the routing protocols operate. In particular, the assumption is that the forwarding plane can continue to function while the protocols restart and sort things out.

BFD implementations announce via the Control Plane Independent (C) bit whether or not BFD shares fate with the control plane. This information is used to determine the actions to be taken in conjunction with Graceful Restart. If BFD does not share its fate with the control plane on either system, it can be used to determine whether Graceful Restart in a control protocol is NOT viable (the forwarding plane is not operating.)

If the control protocol has a Graceful Restart mechanism, BFD may be used in conjunction with this mechanism. The interaction between BFD and the control protocol depends on the capabilities of the control protocol, and whether or not BFD shares fate with the control plane. In particular, it may be desirable for a BFD session failure to abort the Graceful Restart process and allow the failure to be visible to the network.

4.3.1. BFD Fate Independent of the Control Plane

If BFD is implemented in the forwarding plane and does not share fate with the control plane on either system (the "C" bit is set in the BFD Control packets in both directions), control protocol restarts should not affect the BFD Session. In this case, a BFD session failure implies that data can no longer be forwarded, so any Graceful Restart in progress at the time of the BFD session failure SHOULD be aborted in order to avoid black holes, and a topology change SHOULD be signaled in the control protocol.

4.3.2. BFD Shares Fate with the Control Plane

If BFD shares fate with the control plane on either system (the "C" bit is clear in either direction), a BFD session failure cannot be disentangled from other events taking place in the control plane. In many cases, the BFD session will fail as a side effect of the restart taking place. As such, it would be best to avoid aborting any

Graceful Restart taking place, if possible (since otherwise BFD and Graceful Restart cannot coexist.)

There is some risk in doing so, since a simultaneous failure or restart of the forwarding plane will not be detected, but this is always an issue when BFD shares fate with the control plane.

4.3.2.1. Control Protocols with Planned Restart Signaling

Some control protocols can signal a planned restart prior to the restart taking place. In this case, if a BFD session failure occurs during the restart, such a planned restart SHOULD NOT be aborted and the session failure SHOULD NOT result in a topology change being signaled in the control protocol.

4.3.2.2. Control Protocols Without Planned Restart Signaling

Control protocols that cannot signal a planned restart depend on the recently restarted system to signal the Graceful Restart prior to the control protocol adjacency timeout. In most cases, whether the restart is planned or unplanned, it is likely that the BFD session will time out prior to the onset of Graceful Restart, in which case a topology change SHOULD be signaled in the control protocol as specified in section 3.2.

However, if the restart is in fact planned, an implementation MAY adjust the BFD session timing parameters prior to restarting in such a way that the Detection Time in each direction is longer than the restart period of the control protocol, providing the restarting system the same opportunity to enter Graceful Restart as it would have without BFD. The restarting system SHOULD NOT send any BFD Control packets until there is a high likelihood that its neighbors know a Graceful Restart is taking place, as the first BFD Control packet will cause the BFD session to fail.

4.4. Interactions with Multiple Control Protocols

If multiple control protocols wish to establish BFD sessions with the same remote system for the same data protocol, all MUST share a single BFD session.

If hierarchical or dependent layers of control protocols are in use (say, OSPF and IBGP), it may not be useful for more than one of them to interact with BFD. In this example, because IBGP is dependent on OSPF for its routing information, the faster failure detection

relayed to IBGP may actually be detrimental. The cost of a peer state transition is high in BGP, and OSPF will naturally heal the path through the network if it were to receive the failure detection.

In general, it is best for the protocol at the lowest point in the hierarchy to interact with BFD, and then to use existing interactions between the control protocols to effect changes as necessary. This will provide the fastest possible failure detection and recovery in a network.

5. Interactions With Non-Protocol Functions

BFD session status may be used to affect other system functions that are not protocol-based (for example, static routes.) If the path to a remote system fails, it may be desirable to avoid passing traffic to that remote system, so the local system may wish to take internal measures to accomplish this (such as withdrawing a static route and withdrawing that route from routing protocols.)

If it is known, or presumed, that the remote system is BFD-capable and the BFD session is not in Up state, appropriate action SHOULD be taken (such as withdrawing a static route.)

If it is known, or presumed, that the remote system does not support BFD, action such as withdrawing a static route SHOULD NOT be taken.

Bootstrapping of the BFD session in the non-protocol case is likely to be derived from configuration information.

There is no need to exchange endpoints or discriminator values via any mechanism other than configuration (via Operational Support Systems or any other means) as the endpoints must be known and configured by the same means.

6. Data Protocols and Demultiplexing

BFD is intended to protect a single "data protocol" and is encapsulated within that protocol. A pair of systems may have multiple BFD sessions over the same topology if they support (and are encapsulated by) different protocols. For example, if two systems have IPv4 and IPv6 running across the same link between them, these are considered two separate paths and require two separate BFD sessions.

This same technique is used for more fine-grained paths. For example, if multiple differentiated services [DIFFSERV] are being operated over IPv4, an independent BFD session may be run for each service level. The BFD Control packets must be marked in the same way as the data packets, partly to ensure as much fate sharing as possible between BFD and data traffic, and also to demultiplex the initial packet if the discriminator values have not been exchanged.

7. Multiple Link Subnetworks

A number of technologies exist for aggregating multiple parallel links at layer N-1 and treating them as a single link at layer N. BFD may be used in a number of ways to protect the path at layer N. The exact mechanism used is outside the scope of this specification. However, this section provides examples of some possible deployment scenarios. Other scenarios are possible and are not precluded.

7.1. Complete Decoupling

The simplest approach is to simply run BFD over the layer N path, with no interaction with the layer N-1 mechanisms. Doing so assumes that the layer N-1 mechanism will deal with connectivity issues in individual layer N-1 links. BFD will declare a failure in the layer N path only when the session times out.

This approach will work whether or not the layer N-1 neighbor is the same as the layer N neighbor.

7.2. Layer N-1 Hints

A slightly more intelligent approach than complete decoupling is to have the layer N-1 mechanism inform the layer N BFD when the aggregated link is no longer viable. In this case, the BFD session will detect the failure more rapidly, as it need not wait for the

session to time out. This is analogous to triggering a session failure based on the hardware-detected failure of a single link.

This approach will also work whether or not the layer N-1 neighbor is the same as the layer N neighbor.

7.3. Aggregating BFD Sessions

Another approach would be to use BFD on each layer N-1 link, and to aggregate the state of the multiple sessions into a single indication to the layer N clients. This approach has the advantage that it is independent of the layer N-1 technology. However, this approach only works if the layer N neighbor is the same as the layer N-1 neighbor (a single hop at layer N-1.)

7.4. Combinations of Scenarios

Combinations of more than one of the scenarios listed above (or others) may be useful in some cases. For example, if the layer N neighbor is not directly connected at layer N-1, a system might run a BFD session across each layer N-1 link to the immediate layer N-1 neighbor, and then run another BFD session to the layer N neighbor. The aggregate state of the layer N-1 BFD sessions could be used to trigger a layer N BFD session failure.

8. Other Application Issues

BFD can provide liveness detection for OAM-like functions in tunneling and pseudowire protocols. Running BFD inside the tunnel is recommended, as it exercises more aspects of the path. One way to accommodate this is to address BFD packets based on the tunnel endpoints, assuming that they are numbered.

If a planned outage is to take place on a path over which BFD is run, it is preferable to take down the BFD session by going into AdminDown state prior to the outage. The system asserting AdminDown SHOULD do so for at least one Detection Time in order to ensure that the remote system is aware of it.

Similarly, if BFD is to be deconfigured from a system, it is desirable to not trigger any client application action. Simply ceasing the transmission of BFD Control packets will cause the remote system to detect a session failure. In order to avoid this, the system on which BFD is being deconfigured SHOULD put the session into

AdminDown state and maintain this state for a Detection Time to ensure that the remote system is aware of it.

9. Interoperability Issues

The BFD protocol itself is designed so that it will always interoperate at a basic level; asynchronous mode is mandatory and is always available, and other modes and functions are negotiated at run time. Since the service provided by BFD is identical regardless of the variants used, the particular choice of BFD options has no bearing on interoperability.

The interaction between BFD and other protocols and control functions is very loosely coupled. The actions taken are based on existing mechanisms in those protocols and functions, so interoperability problems are very unlikely unless BFD is applied in contradictory ways (such as a BFD session failure causing one implementation to go down and another implementation to come up.) In fact, BFD may be advising one system for a particular control function but not the other; the only impact of this would be potentially asymmetric control protocol failure detection.

10. Specific Protocol Interactions (Non-Normative)

As noted above, there are no interoperability concerns regarding interactions between BFD and control protocols. However, there is enough concern and confusion in this area so that it is worthwhile to provide examples of interactions with specific protocols.

Since the interactions do not affect interoperability, they are non-normative.

10.1. BFD Interactions with OSPFv2, OSPFv3, and IS-IS

The two versions of OSPF ([OSPFv2] and [OSPFv3]), as well as IS-IS [ISIS], all suffer from an architectural limitation, namely that their Hello protocols are limited in the granularity of their failure detection times. In particular, OSPF has a minimum detection time of two seconds, and IS-IS has a minimum detection time of one second.

BFD may be used to achieve arbitrarily small detection times for these protocols by supplementing the Hello protocols used in each case.

10.1.1. Session Establishment

The most obvious choice for triggering BFD session establishment with these protocols would be to use the discovery mechanism inherent in the Hello protocols in OSPF and IS-IS to bootstrap the establishment of the BFD session. Any BFD sessions established to support OSPF and IS-IS across a single IP hop must operate in accordance with [BFD-1HOP].

10.1.2. Reaction to BFD State Changes

The basic mechanisms are covered in section 3 above. At this time, OSPFv2 and OSPFv3 carry routing information for a single data protocol (IPv4 and IPv6, respectively) so when it is desired to signal a topology change after a BFD session failure, this should be done by tearing down the corresponding OSPF neighbor.

ISIS may be used to support only one data protocol, or multiple data protocols. [ISIS] specifies a common topology for multiple data protocols, but work is underway to support multiple topologies. If multiple topologies are used to support multiple data protocols (or multiple classes of service of the same data protocol) the topology-specific path associated with a failing BFD session should no longer be advertised in ISIS LSPs in order to signal a lack of connectivity. Otherwise, a failing BFD session should be signaled by simulating an ISIS adjacency failure.

OSPF has a planned restart signaling mechanism, whereas ISIS does not. The appropriate mechanisms outlined in section 3.3 should be used.

10.1.3. OSPF Virtual Links

If it is desired to use BFD for failure detction of OSPF Virtual Links, the mechanism described in [BFD-MULTI] MUST be used, since OSPF Virtual Links may traverse an arbitrary number of hops. BFD Authentication SHOULD be used and is strongly encouraged.

10.2. Interactions with BGP

BFD may be useful with EBGP sessions [BGP] in order to more rapidly trigger topology changes in the face of path failure. As noted in section 3.4, it is generally unwise for IBGP sessions to interact with BFD if the underlying IGP is already doing so.

EBGP sessions being advised by BFD may establish either a one hop [BFD-1HOP] or a multihop [BFD-MULTIHOP] session, depending on whether the neighbor is immediately adjacent or not. The BFD session should be established to the BGP neighbor (as opposed to any other Next Hop advertised in BGP.)

[BGP-GRACE] describes a Graceful Restart mechanism for BGP. If Graceful Restart is not taking place on an EBGP session, and the corresponding BFD session fails, the EBGP session should be torn down in accordance with section 3.2. If Graceful Restart is taking place, the basic procedures in section 3.3 applies. BGP Graceful Restart does not signal planned restarts, so section 3.3.2.2 applies. If Graceful Restart is aborted due to the rules described in section 3.3, the "receiving speaker" should act as if the "restart timer" expired (as described in [BGP-GRACE].)

10.3. Interactions with RIP

The RIP protocol [RIP] is somewhat unique in that, at least as specified, neighbor adjacency state is not stored per se. Rather, installed routes contain a next hop address, which in most cases is the address of the advertising neighbor (but may not be.)

In the case of RIP, when the BFD session associated with a neighbor fails, an expiration of the "timeout" timer for each route installed from the neighbor (for which the neighbor is the next hop) should be simulated.

Note that if a BFD session fails, and a route is received from that neighbor with a next hop address that is not the address of the neighbor itself, the route will linger until it naturally times out (after 180 seconds.) However, if an implementation keeps track of all of the routes received from each neighbor, all of the routes from the neighbor corresponding to the failed BFD session should be timed out, regardless of the next hop specified therein, and thereby avoiding the lingering route problem.

Normative References

[BFD] Katz, D., and Ward, D., "Bidirectional Forwarding Detection", draft-ietf-bfd-base-07.txt, January, 2008.

[BFD-1HOP] Katz, D., and Ward, D., "BFD for IPv4 and IPv6 (Single Hop)", draft-ietf-bfd-v4v6-1hop-07.txt, January, 2008.

[BFD-MPLS] Aggarwal, R., and Kompella, K., "BFD for MPLS LSPs", draft-ietf-bfd-mpls-04.txt, March, 2007.

[BFD-MULTI] Katz, D., and Ward, D., "BFD for Multihop Paths", draft-ietf-bfd-multihop-06.txt, January, 2008.

[BGP] Rekhter, Y., Li, T. et al, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January, 2006.

[BGP-GRACE] Sangli, S., Chen, E., et al, "Graceful Restart Mechanism for BGP", RFC 4724, January, 2007.

[DIFFSERV] Nichols, K. et al, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December, 1998.

[ISIS] Callon, R., "Use of OSI IS-IS for routing in TCP/IP and dual environments", RFC 1195, December 1990.

[ISIS-GRACE] Shand, M., and Ginsberg, L., "Restart signaling for IS-IS", RFC 3847, July 2004.

[KEYWORD] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

[OSPFv2] Moy, J., "OSPF Version 2", RFC 2328, April 1998.

[OSPFv3] Coltun, R., et al, "OSPF for IPv6", RFC 2740, December 1999.

[OSPF-GRACE] Moy, J., et al, "Graceful OSPF Restart", RFC 3623, November 2003.

[RIP] Malkin, G., "RIP Version 2", RFC 2453, November, 1998.

Security Considerations

This specification does not raise any additional security issues beyond those of the specifications referred to in the list of normative references.

IANA Considerations

This document has no actions for IANA.

Authors' Addresses

Dave Katz
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, California 94089-1206 USA
Phone: 408-745-2000
Email: dkatz@juniper.net

Dave Ward
Cisco Systems
170 W. Tasman Dr.
San Jose, CA 95134 USA
Phone: 408-526-4000
Email: dward@cisco.com

Changes from the previous draft

A section was added to more completely describe the interaction between a BFD session and its clients.

Rules for handling session failures in non-protocol applications were clarified.

A section was added describing possible deployment strategies in conjunction with multilink technologies.

A note was added to point out potential problems when not all systems on a multiaccess network support BFD.

All other changes were purely editorial in nature.

IPR Disclaimer

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Full Copyright Notice

Acknowledgement

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

This document expires in July, 2008.

Network Working Group D. Katz
Cisco Systems

Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [KEYWORDS].

1. Introduction

One very desirable application for BFD [BFD] is to track IPv4 and IPv6 connectivity between directly-connected systems. This could be used to supplement the detection mechanisms in routing protocols, or to monitor router-host connectivity, among other applications.

This document describes the particulars necessary to use BFD in this environment. Interactions between BFD and other protocols and system functions are described in the BFD Generic Applications document [BFD-GENERIC].

2. Applications and Limitations

This application of BFD can be used by any pair of systems communicating via IPv4 and/or IPv6 across a single IP hop that is associated with an incoming interface. This includes, but is not limited to, physical media, virtual circuits, and tunnels.

Each BFD session between a pair of systems MUST traverse a separate path in both directions.

If BFD is to be used in conjunction with both IPv4 and IPv6 on a particular link, a separate BFD session MUST be established for each protocol (and thus encapsulated by that protocol) over that link.

3. Initialization and Demultiplexing

In this application, there will be only a single BFD session between two systems over a given interface (logical or physical) for a particular protocol. The BFD session must be bound to this interface. As such, both sides of a session MUST take the "Active" role (sending initial BFD Control packets with a zero value of Your Discriminator) and any BFD packet from the remote machine with a zero value of Your Discriminator MUST be associated with the session bound to the remote system, interface, and protocol.

4. Encapsulation

4.1. BFD for IPv4

In the case of IPv4, BFD Control packets MUST be transmitted in UDP packets with destination port 3784, within an IPv4 packet. The source port MUST be in the range 49152 through 65535. The same UDP source port number MUST be used for all BFD Control packets associated with a particular session. The source port number SHOULD be unique among all BFD sessions on the system. If more than 16384 BFD sessions are simultaneously active, UDP source port numbers MAY be reused on multiple sessions, but the number of distinct uses of the same UDP source port number SHOULD be minimized. An implementation MAY use the UDP port source number to aid in demultiplexing incoming BFD Control packets, but ultimately the mechanisms in [BFD] MUST be used to demultiplex incoming packets to the proper session.

BFD Echo packets MUST be transmitted in UDP packets with destination UDP port 3785 in an IPv4 packet. The setting of the UDP source port is outside the scope of this specification. The destination address MUST be chosen in such a way as to cause the remote system to forward the packet back to the local system. The source address MUST be chosen in such a way as to preclude the remote system from generating ICMP Redirect messages. In particular, the source address SHOULD NOT be part of the subnet bound to the interface over which the BFD Echo packet is being transmitted, unless it is known by other means that the remote system will not send Redirects.

4.2. BFD for IPv6

In the case of IPv6, BFD Control packets MUST be transmitted in UDP packets with destination port 3784, within an IPv6 packet. The source port MUST be in the range 49152 through 65535. The same UDP source port number MUST be used for all BFD Control packets associated with a particular session. The source port number SHOULD be unique among all BFD sessions on the system. If more than 16384 BFD sessions are simultaneously active, UDP source port numbers MAY be reused on multiple sessions, but the number of distinct uses of the same UDP source port number SHOULD be minimized. An implementation MAY use the UDP port source number to aid in demultiplexing incoming BFD Control packets, but ultimately the mechanisms in [BFD] MUST be used to demultiplex incoming packets to the proper session.

BFD Echo packets MUST be transmitted in UDP packets with destination UDP port 3785 in an IPv6 packet. The setting of the UDP source port is outside the scope of this specification. The source and destination addresses MUST both be associated with the local system. The destination address MUST be chosen in such a way as to cause the remote system to forward the packet back to the local system.

5. TTL/Hop Limit Issues

If BFD authentication is not in use on a session, all BFD Control packets for the session MUST be sent with a TTL or Hop Limit value of 255. All received BFD Control packets that are demultiplexed to the session MUST be discarded if the received TTL or Hop Limit is not equal to 255. A discussion of this mechanism can be found in [GTSM].

If BFD authentication is in use on a session, all BFD Control packets MUST be sent with a TTL or Hop Limit value of 255. All received BFD Control packets that are demultiplexed the session MAY be discarded if the received TTL or Hop Limit is not equal to 255.

In the context of this section, "authentication in use" means that the system is sending BFD control packets with the Authentication bit set and with the Authentication Section included, and that all unauthenticated packets demultiplexed to the session are discarded, per the BFD base specification.

6. Addressing Issues

Implementations MUST ensure that all BFD Control packets are transmitted over the one-hop path being protected by BFD.

On a multiaccess network, BFD Control packets MUST be transmitted with source and destination addresses that are part of the subnet (addressed from and to interfaces on the subnet.)

On a point-to-point link, the source address of a BFD Control packet MUST NOT be used to identify the session. This means that the initial BFD packet MUST be accepted with any source address, and that subsequent BFD packets MUST be demultiplexed solely by the Your Discriminator field (as is always the case.) This allows the source address to change if necessary. If the received source address changes, the local system MUST NOT use that address as the destination in outgoing BFD Control packets; rather it MUST continue to use the address configured at session creation. An implementation MAY notify the application that the neighbor's source address has changed, so that the application might choose to change the destination address or take some other action. Note that the TTL/Hop Limit check described in section 5 (or the use of authentication) precludes the BFD packets from having come from any source other than the immediate neighbor.

7. BFD for use with Tunnels

A number of mechanisms are available to tunnel IPv4 and IPv6 over arbitrary topologies. If the tunnel mechanism does not decrement the TTL or Hop Limit of the network protocol carried within, the mechanism described in this document may be used to provide liveness detection for the tunnel. The BFD Authentication mechanism SHOULD be used and is strongly encouraged.

Normative References

[BFD] Katz, D., and Ward, D., "Bidirectional Forwarding Detection", draft-ietf-bfd-base-07.txt, January, 2008.

[BFD-GENERIC] Katz, D., and Ward, D., "Generic Application of BFD", draft-ietf-bfd-generic-04.txt, January, 2008.

[GTSM] Gill, V., et al, "The Generalized TTL Security Mechanism (GTSM)", RFC 5082, October, 2007.

[KEYWORD] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

Security Considerations

In this application, the use of TTL=255 on transmit and receive, coupled with an association to an incoming interface, is viewed as supplying equivalent security characteristics to other protocols used in the infrastructure, as it is not trivially spoofable. The security implications of this mechanism are further discussed in [GTSM].

The security implications of the use of BFD Authentication are discussed in [BFD].

The use of the TTL=255 check simultaneously with BFD Authentication provides a low overhead mechanism for discarding a class of unauthorized packets and may be useful in implementations in which cryptographic checksum use is susceptable to denial of service attacks. The use or non-use of this mechanism does not impact interoperability.

IANA Considerations

This document has no actions for IANA.

Authors' Addresses

Dave Katz
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, California 94089-1206 USA
Phone: 408-745-2000
Email: dkatz@juniper.net

Dave Ward
Cisco Systems
170 W. Tasman Dr.
San Jose, CA 95134 USA
Phone: 408-526-4000
Email: dward@cisco.com

Changes from the previous draft

Section 6 was changed to lump all point-to-point link types together. Otherwise, only minor editorial changes were made.

IPR Disclaimer

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Full Copyright Notice

Acknowledgement

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

This document expires in September, 2008.

History

Document history		
Issue 1	02/05/2006	Authorised for publication on the Ofcom web site at TSG07 and NICC55.
Issue 2	11/10/2007	On TSG 28-day approval completing 9th November 2007, when the correct version numbering will be inserted.
V1.2.1	06/12/2007	Converted unedited from Issue 2 to V1.2.1 to comply with the new ND numbering rules, for publication on the NICC web site.
V2.2.1	25/06/2008	Change requests A&R 001 and A&R 002 implemented.
V2.2.2	19/08/2008	BFD “shall” to “should” editorial change and BFD reference & Annex E updated. Converted to new template.
V2.2.3	07/01/2009	BFD Security Profile sub-section added. BFD ietf-drafts changed from attachements to plain text