Network Working Group — Bruce Davie, Editor
Internet Draft — Anna Charny
Expiration Date: August 2001 — Fred Baker
Cisco Systems, Inc.

Jon Bennet
Riverdelta Networks

Kent Benson
Tellabs

Jean-Yves Le Boudec
EPFL

Angela Chiu
AT&T Labs

William Courtney
TRW

Shahram Davari
PMC-Sierra

Victor Firoiu
Nortel Networks

Charles Kalmanek
AT&T Research

K.K. Ramakrishnam
TeraOptic Networks

Dimitrios Stiliadis
Lucent Technologies

February 2001

An Expedited Forwarding PHB

draft-ietf-diffserv-rfc2598bis-00.txt

Status of this Memo

This document is a product of the Diffserv working group of the Internet Engineering Task Force. Please address comments to the group's mailing list at diffserv@ietf.org, with a copy to the authors.

Copyright Notice

Abstract

The PHB (per-hop behavior) is a basic building block in the Differentiated Services architecture. This document defines a PHB called Expedited Forwarding (EF). EF is intended to provide a building block for low delay and low loss services by ensuring that the EF aggregate is served at a certain configured rate.

Contents

Specification of Requirements

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 [3].

1. Introduction

Network nodes that implement the differentiated services enhancements to IP use a codepoint in the IP header to select a per-hop behavior (PHB) as the specific forwarding treatment for that packet [RFC2474, RFC2475]. This memo describes a particular PHB called expedited forwarding (EF).

The intent of the EF PHB is to provide a building block for low loss, low delay, and low jitter services. The details of exactly how to build such services are outside the scope of this specification.

The dominant causes of delay in packet networks are speed-of-light propagation delays on wide area links and queuing delays in switches and routers. Since propagation delays are a fixed property of the topology, delay and jitter are minimized when queueing delays are minimized. In this context, jitter is defined as the variation between maximum and minimum delay. The intent of the EF PHB is to provide a PHB in which suitably marked packets usually encounter short or empty queues. Furthermore, if queues remain short relative to the buffer space available, packet loss is also kept to a minimum.

To ensure that queues encountered by EF packets are usually short, it is necessary to ensure that the service rate of EF packets on a given output interface exceeds their arrival rate at that interface over long and short time intervals, independent of the load of other (non-EF) traffic. This specification defines a PHB in which EF packets are guaranteed to receive service at or above a configured rate and provides a means to quantify the accuracy with which this service rate is delivered over any time interval. It also provides a means to quantify the maximum delay and jitter that a packet may experience under bounded operating conditions.

Note that the EF PHB only defines the behavior of a single node. The specification of behavior of a collection of nodes is outside the scope of this document. A Per-Domain Behavior (PDB) specification [7] may provide such information.

When a DS-compliant node claims to implement the EF PHB, the implementation MUST conform to the specification given in this document. However, the EF PHB is not a mandatory part of the Differentiated Services architecture - a node is NOT REQUIRED to implement the EF PHB in order to be considered DS-compliant.

2. Definition of EF PHB

2.1. Intuitive Description of EF

Intuitively, the definition of EF is simple: the rate at which EF traffic is served at a given output interface should be at least the configured rate R, over a suitably defined interval, independent of the offered load of non-EF traffic to that interface. Two difficulties arise when we try to formalize this intuition:

- it is difficult to define the appropriate timescale at which to measure R. By measuring at short timescales we may introduce sampling errors; at long timescales we may allow excessive jitter.

- EF traffic clearly cannot be served at rate R if there are no EF packets waiting to be served, but it may be impossible to determine externally whether EF packets are actually waiting to be served by the output scheduler. For example, if an EF packet has entered the router and not exited, it may be awaiting service, or it may simply have encountered some processing or transmission delay within the router.

The formal definition below takes account of these issues. It assumes that EF packets should ideally be served at rate R or faster, and bounds the deviation of the actual departure time of each packet from the "ideal" departure time of that packet. We define the departure time of a packet as the time when the last bit of that packet leaves the node. The "ideal" departure time of each EF packet is computed iteratively.

In the case when an EF packet arrives to a device when all the previous EF packets have already departed, the computation of the ideal departure time is simple. Service of the packet should (ideally) start as soon as it arrives, so the ideal departure time is simply the arrival time plus the ideal time to transmit the packet at rate R. For a packet of length L_j, that transmission time at the configured rate R is L_j/R. (Of course, a real packet will typically get transmitted at line rate once its transmission actually starts,

but we are calculating the ideal target behavior here; the ideal service takes place at rate R.)

In the case when an EF packet arrives to a device which still contains EF packets awaiting service, the computation of the ideal departure time is more complicated. There are two cases to be considered. If the previous (j-1-th) departure occurred after its own ideal departure time, then the scheduler is running "late". In this case, the ideal time to start service of the new packet is the ideal departure time of the previous (j-1-th) packet, or the arrival time of the new packet, whichever is later, because we can't expect a packet to begin service before it arrives. If the previous (j-1-th) departure occurred before its own ideal departure time, then the scheduler is running "early". In this case, service of the new packet should begin at the actual departure time of the previous packet.

Once we know the time at which service of the jth packet should (ideally) begin, then the ideal departure time of the jth packet is L_j/R seconds later. Thus we are able to express the ideal departure time of the jth packet in terms of the arrival time of the jth packet, the actual departure time of the j-1-th packet, and the ideal departure time of the j-1-th packet. Equations eq_1 and eq_2 in Section 2.2 capture this relationship.

Whereas the original EF definition did not provide any means to guarantee the delay of an individual EF packet, this property may be desired. For this reason, the equations in Section 2.2 consist of two parts: a "colorblind" set and a "packet-identity-aware" set of equations. The colorblind equations (eq_1 and eq_2) simply describe the properties of the service delivered to the EF aggregate by the device. The "packet-identity-aware" equations (eq_3 and eq_4) enable the bound on delay of an individual packet to be calculated given a knowledge of the operating conditions of the device. The significance of these two sets of equations is discussed further in Section 2.2. Note that these two sets of equations provide two ways of characterizing the behavior of a single device, not two different modes of behavior.

2.2. Formal Definition of the EF PHB

A node that supports EF on an interface I at some configured rate R MUST satisfy the following equations:

$$d_j <= f_j + E_a \quad \text{(eq\_1)}$$

where f_j is defined iteratively by

$$f_0 = 0, d_0 = 0$$

$$f_j = max(a_j, min(d_{j-1}, f_{j-1})) + l_j/R, \quad \text{for all } j > 0 \quad \text{(eq\_2)}$$

In this definition:

- d_j is the time that the last bit of the j-th EF packet to depart actually leaves the node from the interface I.

- f_j is the target departure time for the j-th EF packet to depart from I, the "ideal" time at or before which the last bit of that packet should leave the node.

- a_j is the time that the last bit of the j-th EF packet destined to the output I to arrive actually arrives at the node.

- l_j is the size (bits) of the j-th EF packet to depart from I. l_j is measured on the IP datagram (IP header plus payload) and does not include any lower layer (e.g. MAC layer) overhead.

- R is the EF configured rate at output I (in bits/second).

- E_a is the error term for the treatment of the EF aggregate. Note that E_a represents the worst case deviation between actual departure time of an EF packet and ideal departure time of the same packet, i.e. E_a provides an upper bound on (d_j - f_j) for all j.

- d_0 and f_0 do not refer to a real packet departure but are used purely for the purposes of the recursion. The time origin should be chosen such that no EF packets are in the system at time 0.

An EF-compliant node MUST be able to be characterized by the range of possible R values that it can support on each of its interfaces while conforming to these equations, and the value of E_a that can be met on each interface. R may be line rate or less. E_a MAY be specified as a worst-case value for all possible R values or MAY be expressed as a function of R.

Note also that, since a node may have multiple inputs and complex internal scheduling, the jth packet to arrive may not be the jth

packet to depart. It is in this sense that eq_1 and eq_2 are colorblind with regard to packet identity.

In addition, a node that supports EF on an interface I at some configured rate R MUST satisfy the following equations:

$$D_j \le F_j + E_p \qquad \text{(eq\_3)}$$

where F_j is defined iteratively by

$$F_0 = 0, \; D_0 = 0$$

$$F_j = \max(A_j, \min(D_{j-1}, F_{j-1})) + L_j/R, \quad \text{for all } j > 0 \qquad \text{(eq\_4)}$$

In this definition:

- D_j is actual the departure time of the individual EF packet that arrived at time A_j, i.e., given a packet which was the j-th EF packet destined for I to arrive at the node via any input, D_j is the time at which the last bit of that individual packet actually leaves the node from the interface I.

- F_j is the target departure time for the individual EF packet which arrived at time A_j.

- A_j is the time that the last bit of the j-th EF packet destined to the output I to arrive actually arrives at the node.

- L_j is the size (bits) of the j-th EF packet to arrive at the node that is destined to output I. L_j is measured on the IP datagram (IP header plus payload) and does not include any lower layer (e.g. MAC layer) overhead.

- R is the EF configured rate at output I (in bits/second).

- E_p is the error term for the treatment of individual EF packets. Note that E_p represents the worst case deviation between actual departure time of an EF packet and ideal departure time of the same packet, i.e. E_p provides an upper bound on (D_j - F_j) for all j.

- D_0 and F_0 do not refer to a real packet departure but are used purely for the purposes of the recursion. The time origin should be chosen such that no EF packets are in the system at time 0.

It is the fact that D_j and F_j refer to departure times for the jth packet to arrive that makes eq_3 and eq_4 aware of packet identity. This is the critical distinction between the last two equations and the first two.

An EF-compliant node SHOULD be able to be characterized by the range of possible R values that it can support on each of its interfaces while conforming to these equations, and the value of E_p that can be met on each interface. E_p MAY be specified as a worst-case value for all possible R values or MAY be expressed as a function of R. An E_p value of "undefined" MAY be specified. For discussion of situations in which E_p may be undefined see the Appendix and [6].

2.3. Figures of merit

E_a and E_p may be thought of as "figures of merit" for a device. A smaller value of E_a means that the device serves the EF aggregate more smoothly at rate R over relatively short timescales, whereas a larger value of E_a implies a more bursty scheduler which serves the EF aggregate at rate R only when measured over longer intervals. A device with a larger E_a can "fall behind" the ideal service rate R by a greater amount than a device with a smaller E_a.

A lower value of E_p implies a tighter bound on the delay experienced by an individual packet. Factors that might lead to a higher E_p might include a large number of input interfaces (since an EF packet might arrive just behind a large number of EF packets that arrived on other interfaces), or might be due to internal scheduler details (e.g. per-flow scheduling within the EF aggregate).

We observe that factors that increase E_a such as those noted above will also increase E_p, and that E_p is thus typically greater than or equal to E_a. In summary, E_a is a measure of deviation from ideal service of the EF aggregate at rate R, while E_p measures both non-ideal service and non-FIFO treatment of packets within the aggregate.

For more discussion of these issues see the Appendix and [6].

2.4. Delay and jitter

Given a known value of E_p and a knowledge of the bounds on the EF traffic offered to a given output interface, summed over all input interfaces, it is possible to bound the delay and jitter that will be experienced by EF traffic leaving the node via that interface. The delay bound is

$$D = B/R + E_p \qquad \text{(eq\_5)}$$

where

- R is the configured EF service rate on the output interface

- the total offered load of EF traffic destined to the output interface, summed over all input interfaces, is bounded by a token bucket of rate r <= R and depth B

Since the minimum delay through the device is clearly at least zero, D also provides a bound on jitter. To provided a tighter bound on jitter, a device MAY advertise E_p as two separate components such that

$$E_p = E_{fixed} + E_{variable}$$

where E_fixed represents the minimum delay that can be experienced by an EF packet through the node.

2.5. Loss

The EF PHB is intended to be a building block for low loss services. However, under sufficiently high load of EF traffic (including unexpectedly large bursts from many inputs at once), any device with finite buffers may need to discard packets. Thus, it must be possible to establish whether a device conforms to the EF definition even when some packets are lost. This is done by performing an "off-line" test of conformance to equations 1 through 4. After observing a sequence of packets entering and leaving the node, the packets which did not leave are assumed lost and are notionally removed from the input stream. The remaining packets now constitute the arrival stream (the a_j's) and the packets which left the node constitute the departure stream (the d_j's). Conformance to the equations can thus be verified by considering only those packets that successfully passed through the node.

In addition, to assist in meeting the low loss objective of EF, a node MAY be characterized by the operating region in which loss of EF

due to congestion will not occur. This MAY be specified, using a token bucket of rate r <= R and burstsize B, as the sum of traffic across all inputs to a given output interface that can be tolerated without loss.

In the event that loss does occur, the specification of which packets are lost is beyond the scope of this document. However it is a requirement that those packets not lost MUST conform to the equations of Section 2.2.

2.6. Microflow misordering

Packets belonging to a single microflow within the EF aggregate passing through a device SHOULD NOT experience re-ordering in normal operation of the device.

2.7. Recommended codepoint for this PHB

Codepoint 101110 is RECOMMENDED for the EF PHB.

2.8. Mutability

Packets marked for EF PHB MAY be remarked at a DS domain boundary only to other codepoints that satisfy the EF PHB. Packets marked for EF PHBs SHOULD NOT be demoted or promoted to another PHB by a DS domain.

2.9. Tunneling

When EF packets are tunneled, the tunneling packets SHOULD be marked as EF. A full discussion of tunneling issues is presented in [5].

2.10. Interaction with other PHBs

Other PHBs and PHB groups may be deployed in the same DS node or domain with the EF PHB. The equations of Section 2.2 MUST hold for a node independent of the amount of non-EF traffic offered to it.

If the EF PHB is implemented by a mechanism that allows unlimited preemption of other traffic (e.g., a priority queue), the implementation SHOULD include some means to limit the damage EF traffic could inflict on other traffic. This will be reflected in the range of supported R values as described in section 2.2.

3. Security Considerations

To protect itself against denial of service attacks, the edge of a DS domain SHOULD strictly police all EF marked packets to a rate negotiated with the adjacent upstream domain. Packets in excess of the negotiated rate SHOULD be dropped. If two adjacent domains have not negotiated an EF rate, the downstream domain SHOULD use 0 as the rate (i.e., drop all EF marked packets).

4. IANA Considerations

This document allocates one codepoint, 101110, in Pool 1 of the code space defined by [RFC2474].

5. Acknowledgments

This document draws heavily on the original EF PHB definition of Jacobson, Nichols and Poduri. It was also greatly influenced by the work of the EFRESOLVE team of Armitage, Casati, Crowcroft, Halpern, Kumar, and Schnizlein.

6. References

[1] V. Jacobson, K. Nichols, K. Poduri, "An Expedited Forwarding PHB", RFC 2598, June 1999

[2] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, BCP 14, March 1997

[3] K. Nichols, S. Blake, F. Baker, D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, December 1998.

[4] D. Black, S. Blake, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998.

[5] D. Black, "Differentiated Services and Tunnels", RFC 2983, October 2000.

[6] A. Charny et al., "Supplemental Information for the New Definition of the EF PHB", Work in Progress, February 2001.

[7] K. Nichols and B. Carpenter, "Definition of Differentiated Services Per Domain Behaviors and Rules for their Specification",

Work in Progress, January 2001.

Appendix: Implementation Examples

This appendix is not part of the normative specification of EF. However, it is included here as a possible source of useful information for implementors.

A variety of factors in the implementation of a node supporting EF will influence the values of E_a and E_p. These factors are discussed in more detail in [6], and include both output schedulers and the internal design of a device.

A priority queue is widely considered as the canonical example of an implementation of EF. A "perfect" output buffered device (i.e. one which delivers packets immediately to the appropriate output queue) with a priority queue for EF traffic will provide both a low E_a and a low E_p. We note that the main factor influencing E_a will be the inability to pre-empt an MTU-sized non-EF packet that has just begun transmission at the time when an EF packet arrives at the output interface, plus any additional delay that might be caused by non-pre-emptable queues between the priority queue and the physical interface. E_p will be influenced primarily by the number of interfaces.

Another example of an implementation of EF is a weighted round robin scheduler. Such an implementation will typically not be able to support values of R as high as the link speeds, because the maximum rate at which EF traffic can be served in the presence of competing traffic will be affected by the number of other queues and the weights given to them. Furthermore, such an implementation is likely to have a value of E_a that is higher than a priority queue implementation, all else being equal, as a result of the time spent serving non-EF queues by the round robin scheduler.

Finally, it is possible to implement hierarchical scheduling algorithms, such that some non-FIFO scheduling algorithm is run on sub-flows within the EF aggregate, while the EF aggregate as a whole could be served at high priority or with a large weight by the top-level scheduler. Such an algorithm might perform per-input scheduling or per-microflow scheduling within the EF aggregate, for example. Because such algorithms lead to non-FIFO service within the EF aggregate, the value of E_p for such algorithms may be higher than for other implementations. For some schedulers of this type it may be difficult to provide a meaningful bound on E_p that would hold for any pattern of traffic arrival, and thus a value of "undefined" may

be most appropriate.

Authors' Addresses

Bruce Davie
Cisco Systems, Inc.
300 Apollo Drive
Chelmsford, MA, 01824

E-mail: bsd@cisco.com

Anna Charny
Cisco Systems
300 Apollo Drive
Chelmsford, MA 01824

E-mail: acharny@cisco.com

Fred Baker
Cisco Systems
170 West Tasman Dr.
San Jose, CA 95134

E-mail: fred@cisco.com

Jon Bennett
RiverDelta Networks
3 Highwood Drive East
Tewksbury, MA 01876

E-mail: jcrb@riverdelta.com

Kent Benson
Tellabs Research Center
3740 Edison Lake Parkway #101
Mishawaka, IN 46545

E-mail: Kent.Benson@tellabs.com

Jean-Yves Le Boudec
ICA-EPFL, INN
Ecublens, CH-1015
Lausanne-EPFL, Switzerland

E-mail: leboudec@epfl.ch

Angela Chiu
AT&T Labs
100 Schulz Dr. Rm 4-204
Red Bank, NJ 07701

E-mail: alchiu@att.com

Bill Courtney
TRW
Bldg. 201/3702
One Space Park
Redondo Beach, CA 90278

E-mail: bill.courtney@trw.com

Shahram Davari
PMC-Sierra Inc
411 Legget Drive
Ottawa, ON K2K 3C9, Canada

E-mail: shahram_davari@pmc-sierra.com

Victor Firoiu
Nortel Networks
600 Tech Park
Billerica, MA 01821

E-mail: vfirou@nortelnetworks.com

Charles Kalmanek
AT&T Labs-Research
180 Park Avenue, Room A113,
Florham Park NJ

E-mail: crk@research.att.com.

K.K. Ramakrishnan
TeraOptic Networks, Inc.
686 W. Maude Ave
Sunnyvale, CA 94086

E-mail: kk@teraoptic.com

Dimitrios Stiliadis
Lucent Technologies
1380 Rodick Road
Markham, Ontario, L3R-4G5, Canada

E-mail: stiliadi@bell-labs.com

7. Full Copyright