Internet Engineering Task Force
Differentiated Services Working Group
Internet Draft
Expires February, 1999

Van Jacobson
LBNL
Kathleen Nichols
Kedarnath Poduri
Bay Networks
August, 1998

# An Expedited Forwarding PHB
<draft-ietf-diffserv-phb-ef-00.txt>

## Status of this Memo



## Abstract

The definition of PHBs (per-hop forwarding behaviors) is a critical part of the work of the Diffserv Working Group. This document describes a PHB called Expedited Forwarding. We show the generality of this PHB by noting that it can be produced by more than one mechanism and give an example of its use to produce at least one service, a Virtual Leased Line. A recommended codepoint for this PHB is given.

A pdf version of this document is available at
ftp://ftp.ee.lbl.gov/papers/ef_phb.pdf

## 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 [HEADER, ARCH]. This draft describes a particular PHB called expedited forwarding (EF). The EF PHB can be used to build a low loss, low latency, low jitter, assured bandwidth, end-to-end service through DS domains. Such a service appears to the endpoints like a point-to-point connection or a "virtual leased line". This service has also been described as Premium service [2BIT].

Loss, latency and jitter are all due to the queues traffic experiences while transiting the network. Therefore providing low loss, latency and jitter for some traffic aggregate means ensuring that the aggregate sees no (or very small) queues. Queues arise when (short term) traffic arrival rate exceeds departure rate at some node. Thus a service that ensures no queues for some aggregate is equivalent to bounding rates such that, at every transit node, the aggregate's max arrival rate is less than that aggregate's min departure rate.

Creating such a service has two parts:

1) configuring nodes so that the aggregate has a well-defined minimum departure rate. (`Well-defined' means independent of the dynamic state of the node. In particular, independent of the intensity of other traffic at the node.)

2) conditioning the aggregate (via policing and shaping) so that it's arrival rate at any node is always less than that node's configured minimum departure rate.

The EF PHB provides the first part of the service. The network boundary traffic conditioners described in [ARCH] provide the second part.

The next sections describe the EF PHB in detail and give examples of how it might be implemented. The keywords "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT", and "MAY" that appear in this document are to be interpreted as described in [Bradner97].

## 2. Description of EF per-hop behavior

### 2.1 Description

The EF PHB is defined as a forwarding treatment for a particular diffserv aggregate where the departure rate of the aggregate's packets from any diffserv node must equal or exceed a configurable rate. The EF traffic should receive this rate independent of the intensity of any other traffic attempting to transit the node. It should average at least the configured rate when measured over any time interval equal to or longer than a packet time at the configured rate. (Behavior at time scales shorter than a packet time at the configured rate is deliberately not specified.) The configured minimum rate must be settable by a network administrator (using whatever mechanism the node supports for non-volatile configuration).

The Appendix describes how this PHB can be used to construct end-to-end services.

### 2.2 Example Mechanisms to Implement the EF PHB

Several types of queue scheduling mechanisms may be employed to deliver the forwarding behavior described in section 2.1 and thus implement the EF PHB. A simple priority queue will give the appropriate behavior as long as there is no higher priority queue the could preempt the EF for more than a packet time at the configured rate. (This could be accomplished by having a rate policer such as a token bucket associated with each priority queue to bound how much the queue can starve other traffic.)

It's also possible to use a single queue in a group of queues serviced by a weighted round robin scheduler where the share of the output bandwidth assigned to the EF queue is equal to the configured rate. This could be implemented, for example, using one PHB of a Class Selector Compliant set of PHBs [HEADER].

Another possible implementation is a CBQ scheduler that gives the EF queue priority up to the configured rate.

All of these mechanisms give the basic properties required for the EF PHB though different choices result in differences in auxiliary behavior such as jitter seen by individual microflows. See Appendix **A.3 for simulations that quantify some of these differences.**

### 2.3 Recommended codepoint for this PHB

Codepoint 101100 is recommended for the EF PHB.

### 2.4 Mutability

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

### 2.5 Tunneling

When EF packets are tunneled, the tunneling packets must be marked as EF.

### 2.6 Interaction with other PHBs

Other PHBs and PHB groups may be deployed in the same DS node or domain with the EF PHB as long as the requirement of section 2.1 is met.

## 3. Security Considerations

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

Since the end-to-end premium service constructed from the EF PHB requires that the upstream domain police and shape EF marked traffic to meet the rate negotiated with the downstream domain, the downstream domain's policer should never have to drop packets. Thus these drops should be noted (e.g., via SNMP traps) as possible security violations or serious misconfiguration. Similarly, since the aggregate EF traffic rate is constrained at every interior node, the EF queue should never overflow so if it does the drops should be noted as possible attacks or serious misconfiguration.

## 4. References

[Bradner97] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", Internet RFC 2119, March 1997.

[HEADER] K. Nichols, S. Blake, F. Baker, and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", <draft-ietf-diffserv-header-02.txt>, August 1998.

[ARCH] D. Black, S. Blake, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An Architecture for Differentiated Services", Internet Draft <draft-ietf-diffserv-arch-01.txt>, August 1998.

[2BIT] K. Nichols, V. Jacobson, and L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet", Internet Draft <draft-nichols-diff-svc-arch-00.txt>, November 1997, ftp://ftp.ee.lbl.gov/papers/dsarch.pdf.

[CBQ] S. Floyd and V. Jacobson, "Link-sharing and Resource Management Models for Packet Networks", IEEE/ACM Transactions on Networking, Vol. 3 no. 4, pp. 365-386, August 1995.

[IW] K. Poduri and K. Nichols, "Simulation Studies of Increased Initial TCP Window Size", <draft-ietf-tcpimpl-poduri-02.txt>, August, 1998.

[LCN] K. Nichols, "Improving Network Simulation with Feedback", to appear in proceedings of LCN '98, October, 1998

## 5. Authors' Addresses

Van Jacobson
Lawrence Berkeley National Laboratory
M/S 50B-2239
One Cyclotron Road
Berkeley, CA 94720
van@ee.lbl.gov

Kathleen Nichols
Bay Networks, Inc.
4401 Great America Parkway
Santa Clara, CA 95052-8185
knichols@baynetworks.com
+1 408-495-3252

Kedarnath Poduri
Bay Networks, Inc.
Bay Networks, Inc.
4401 Great America Parkway
Santa Clara, CA 95052-8185
kpoduri@baynetworks.com

Appendix. Example use of and experiences with the EF PHB

## A.1 Virtual Leased Line Service

A VLL Service, also known as Premium service [2BIT], is quantified by a peak bandwidth.

## A.2 Experiences with its use in ESNET

A prototype of the VLL service has been deployed on DOE's ESNet backbone. This uses weighted-round-robin queuing features of cisco 7xxx series routers to implement the EF PHB. The early tests have been very successful (details are available in ftp://ftp.ee.lbl.gov/talks/vj-doeqos.pdf and ftp://ftp.ee.lbl.gov/talks/vj-i2qos-may98.pdf) and work is in progress to make the service available on a routine production basis.

## A.3 Simulation Results

In section 2.2, we pointed out that a number of mechanisms may be used to implement the EF PHB. The simplest is a priority queue where the arrival rate of the queue is strictly less than its departure rate. As jitter comes from the queuing delay along the path, a feature of this implementation is that EF-marked microflows will see very little jitter at their subscribed rate if all DS nodes along the path use this implementation since packets spend little time in queues. This low-jitter behavior is not a requirement of the EF PHB, but we want to explore how other implementations, in this case WRR, compare in jitter. We've compared PQ and WRR because these seemed to be the best and worst cases, respectively, for jitter.

Our basic simulation model is implemented in ns-2 as described in [IW] and [LCN]. We've made some further modifications to ns-2, using the CBQ modules included with ns-2 as a basis to implement priority queuing and WRR.

We experimented with a six-hop topology with decreasing bandwidth in the direction of a single 1.5 Mbps bottleneck link. For our EF-marked packets, we set up sources to produce packets at a constant bit rate with a variation of +/-10% of the subscribed packet rate. The individual source rates were picked to add up to 30% of the bottleneck link or 450 Kbps. A mixture of other kinds of traffic, FTPs and HTTPs, is used to fill the link. We report jitter as the added delay normalized by the time to send a packet at the subscribed peak rate. The pdf version of this document contains graphs of percentile vs jitter and we include text tables that report the 95th percentile from each of the scenarios. We used different packet sizes for the EF-marked packets in our simulations, but always used the same packet size for all EF-marked packets in any particular simulation. We report percentile of packets seeing less than a particular normalized packet size in jitter.

We will consider the implementation of the EF PHB with a priority queue (PQ) as a kind of baseline or "ideal" case. To summarize the results we've seen for PQ jitter, jitter is most strongly dependent on packet size. For **1500 byte packets, all jitter is less than 0.5 packet times. For 160 byte** packets, 95% of packet jitter is less than 3.5 packet times with most packets having less than one packet's worth of jitter. The PQ results will be shown with the WRR results below.

Next we explored the jitter behavior for WRR implementations of the EF PHB. What we wanted to explore was how different the jitter behavior is from that of PQ implementations. Major features that can affect jitter are packet size, number of queues for the WRR scheduler, and the amount by which the guaranteed minimum service rate of the EF queue exceeds the peak arrival rate to the EF queue. We have not yet systematically explored effects of hop count, EF allocations of more or less than 30% of the link bandwidth, or more complex topologies. However, this information is simply to guide those who are interested in a low jitter implementation and is not required for implementing the EF PHB with WRR.

In our WRR simulations, we kept the link full with other traffic as described above, splitting the non-EF-marked traffic among the non-EF queues. If the WRR weight is chosen to exactly balance arrival and departure rates, our results will not be stable except for the simplest cases, so we always overallocate by a minimum of 1% of the output link bandwidth or, in this case, 3% of the peak arrival rate of EF-marked packets. We recommend at least this overallocation to implementors. In figure 1 and table 1, we show results from varying the number of individual microflows composing the EF aggregate of 450 Kbps. In this case all EF packets are 1500 bytes and the EF queue gets a weight of 31% of the output links. The leftmost curve shows the results for a PQ with 24 flows. Note that the maximum jitter of 3.2 packets occurs only for 36 flows, but the 95th percentile of all scenarios is less than 1 packet of jitter. Figure 2 and table 1 shows the results when a packet size of 160 bytes is used.

Table 1: Variation in jitter with number of EF flows

| Num of EF flows | Jitter (95th percentile 1500 Byte Packets) | Jitter (95th percentile 160 Byte Packets) |
|---|---|---|
| PQ (24) | 0.07 | 0.5 |
| 2 | **0.6** | 6.6 |
| 4 | **0.4** | 3.9 |
| 8 | **0.3** | 1.8 |
| 24 | **0.6** | 2.1 |

Next we look at the effects of overallocating the link share, that is giving a minimum service rate that exceeds the peak arrival rate by various amounts. (Of course, with WRR, that bandwidth is still available for other packets.) We fixed the number of flows at eight and the total number of queues at five (four non-EF queues). In figure 3 we report results for 1500 byte EF packets and in figure 4 we show 160 byte packets. Table 2 gives the 95th percentile values of jitter for the same. Overallocation by up to 100% still does not give the same performance as PQ, but note that most packets experience small jitter. In fact, overallocation does not appear to have much improvement associated with it.

Table 2: Variation in Jitter with Overallocation of BW to EF queues.

| % of Over-Allocation | Jitter (95th percentile 1500 Byte Packets) | Jitter (95th percentile 160 Byte Packets) |
|---|---|---|
| PQ | 0.05 | 0.5 |
| 3 | **0.3** | 2.2 |
| 30 | **0.2** | 1.4 |
| 50 | **0.15** | 1.2 |
| 70 | **0.15** | 1.2 |

| | | |
|---|---|---|
| 100 | **0.15** | 1.2 |

---

We know that increasing the number of queues at the output interfaces can lead to more variability in the service time for EF packets. We set the number of flows to eight and used a 31% weight for the 30% EF allocation and varied the number of queues at each output interface. Results are shown in figure 5 and table 3. Note that most packets experience little jitter. PQ with 8 flows is included as a baseline.

Table 3: Variation in Jitter with Number of Queues at Output Interface

| Num of Queues | Jitter (95th percentile 1500 Byte Packets) |
|---|---|
| PQ | 0.05 |
| 2 | **0.2** |
| 4 | **0.3** |
| 6 | **0.3** |
| 8 | **0.35** |

We intend to perform further studies and vary other parameters, but at present it appears that most packet jitter for WRR is low, but by overallocating the EF queue's WRR share of the output link with respect to its subscribed rate packet jitter can be reduced if desired.

As noted, WRR is probably a "worst case" while PQ is the best case. Other possibilities include WFQ or CBQ with a fixed rate limit for the EF queue, but giving it priority over other queues. We expect the latter to have performance nearly identical with PQ, though future simulations can verify this.