IP Precedence in Differentiated Services Using the Assured Service

draft-ietf-diffserv-precedence-00.txt

This document is an Internet Draft. 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 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 a "work in progress". Comments should be made on the list diff-serv@baynetworks.com

Abstract

This document describes the use of a set of Diff-Serv Per-Hop Behaviors (PHBs) to implement a service similar to the Precedence service described in [IP], providing also the Assured Service model described by [ASSURED].

1. Introduction

In short, this memo is intended to describe a way to implement IP Precedence in the Differentiated Services Architecture. By way of an existence proof and argument for the definition of this service, we first discuss IP Precedence, its history, intent, and present day use.

1.1. IP Precedence History

IP Precedence, and the IP Precedence Field, were first defined in [IP], as a way to convey end to end an expectation that a given IP datagram should be placed in a given link layer queue. The various values that the three bit IP Precedence Field might take were assigned to various uses, including network control traffic, routing traffic, and various levels of privilege. The least level of privilege was deemed "routine traffic".

Although early BBN IMPs implemented the service, early commercial routers and UNIX IP forwarding code generally did

not. As networks became more complex and customer requirements grew, commercial routers developed ways to implement various kinds of queuing services including Priority Queuing, which were generally based on policies encoded in filters in the routers, which looked at IP Addresses, IP Protocol numbers, TCP or UDP ports, and the like. IP Precedence was and is among the options such filters can look at, but just one.

In more recent years, however, at least five common uses of the IP Precedence Field have developed. These include:

(1) As a drop preference in a receiving router.

Routing and network control traffic is marked on transmission as being of high precedence. If a router receives the packet at a time that it deems difficult to service random traffic, such as during a bad route flap, the router may drop lower precedence traffic in order to assure the ability to receive higher precedence traffic. It does so in the belief that conserving buffer space and other resources during times of stress will help routing converge more quickly, improving overall network service.

This is, at this time, an important stability issue for certain routers located in sensitive places in the internet.

An example of such a behavior is Cisco's SPD feature.

(2) As a drop preference in a transit router.

In this case, traffic of various sorts may be marked, either by the originating host or by a router. When the packet is enqueued to subsequent congested router interfaces, the traffic is more or less subject to drop depending on its precedence setting. The predominant current use is to support routing traffic (such as BGP) across a local routing domain which may use an IGP to route between the routers, but many instances exist where traffic is marked by the first hop router and treated in this manner across a network.

This may be done using strict drop priorities, or using such techniques as Cisco's Weighted RED or Clark's RIO. The latter two implement Random Early Detection, and provide a way to select differing min-threshold and max-threshold values; in the case of WRED, the selector is IP

Precedence.

(3) As a queue selector, whether a strict priority queue, a round robin load sharing queue, or a VC in a multiplexed interface such as Frame Relay or ATM.

Such mechanisms assume that the queue that higher precedence traffic is placed in has a higher probability of delivering the traffic in a timely manner, whether due to absolute priority or due to rates assigned to queues.

This may be done using facilities such as Cisco's Priority, Custom, or Distributed Class-based Fair Queuing services, or the Newbridge 36000's classification facilities.

(4) As a selector for the weight of a packet in a Weighted Fair Queuing System.

In such a case, when a packet is enqueued in its flow's sub-queue, the weight assigned to the packet is taken from a table which is indexed by the value of the IP Precedence field. In this manner, higher precedence traffic gains a larger proportion of the link without having to configure policy for specific classes of traffic.

Examples of such include Newbridge, ACC, and Cisco Weighted Fair Queuing services.

(5) IP Precedence is used to index a min-threshold and max-threshold array on an interface configured for an extended Random Early Detection algorithm. This is similar to Clark's RIO, except that it it provides for the possibility of several levels of "in profile" and "out of profile". One could imagine using this as seven levels of "in profile" and a single "out" penalty box, as pairs of "in" and "out" precedences, or in other ways.

An example of this is Cisco's Committed Access Rate service.

In short, IP Precedence is widely deployed and widely used, if not in exactly the manner intended in [IP]. This was recognized in [HOSTREQ], which states that while the use of the IP Precedence field is valid, the specific assignment of the priorities in [IP] were merely historical.

1.2. The Assured Service

Clark's Assured Service [ASSURED] suggests that a contract might exist between a service provider and its peer, or between two service providers, which guarantees a certain level of service, and offers the opportunity to overload this on a best effort basis. The expectation is that traffic which is within the contracted rate, as measured by a token bucket, has a very much reduced probability of being lost, while traffic which is excess has a less sanguine prognosis. Inherent in the model is the supposition that a boundary device, probably a router, is measuring traffic and marking it either "in" or "out".

This model is being tested by many service providers today, with a view to offering a layered usage-based service level agreement. Such an agreement might include several layers of drop or delay preference, and associated rates. For example, it might offer the following four-tiered service:

(1) Routing traffic, marked with IP Precedence six or seven, may be exchanged at will and as much as necessary, but there is a charge per route flap. There is no excess traffic, so all is marked in profile.

(2) IP Telephony or similar real time services, marked with the PHB 101100, which is to say precedence five and in profile, may be exchanged up to a certain rate. Traffic which is in profile experiences very low probability of loss, apart from unplanned outages. Excess traffic is marked with the same precedence but out of profile, and is subject to random loss. The contracted bandwidth is charged at a flat rate, and there is a usage charge for excess traffic. One might imagine the use of RSVP to the edge router, or a bandwidth broker protocol as envisioned by [BROKER], to manage this contract level.

(3) Traffic to specific CIDR Prefixes (such as a VPN, marked with the PHB 100100, which is to say precedence four and in profile, may be exchanged up to a certain rate. Traffic which is in profile experiences very low probability of loss, apart from unplanned outages. Excess traffic is marked with the same precedence but out of profile, and is subject to random loss. The contracted bandwidth is charged at a flat rate, and there is a usage charge for excess traffic.

(4) Other traffic, marked with the PHB 010100, which is to say precedence two and in profile, may be exchanged up to a certain rate. Traffic which is in profile experiences low probability of loss, apart from unplanned outages, but a greater probability than traffic in the categories previously mentioned, due to the fact that this traffic is harder to engineer for. Excess traffic is marked with the same precedence but out of profile, and is subject to random loss. The contracted bandwidth is charged at a flat rate, and there is a usage charge for excess traffic.

The above is obviously but one of many possible examples. A minor variation on the theme might permit excess traffic to customers of the same service provider but drop excess traffic rather than forward it to other providers. Many other varieties of service level agreements are also possible.

One issue that we are told is important is that at least some service providers would like to be able to offer similar contracts to different customers with different cost structures. A corporate customer, for example, might obtain a contract similar to the above, while an educational customer might simply contract for precedence three service on a usage basis. The various precedence levels now map not only to in/out flags and drop preferences, but to price points in the tariff structure. This argues for additional precedence values that can be charged at different rates.

1.3. Differentiated Services Overview

Differentiated Services, as described in [FRAMEWORK], re-allocates the most significant six bits of the TOS byte as a PHB. These are, by definition, cases in a case statement rather than being comparable numbers, as [IP]'s Precedence field was. These can clearly be used, however, to implement structured services like IP Precedence if care is taken to define the matter clearly. Specifically, these PHB selectors may be modified according to a set of rules.

One expectation that clearly differs from that in [IP] is that the exact implementation of the PHB may vary from system to system. Rather than specifying a simple priority service, as [IP] does, the PHB might select one of several queues in a Class Based Queuing system, some of which have different rates than others. In such a case, the fact that the queue has a higher rate than some other queue is considered equivalent to

having higher priority, even though a strict priority model is not being followed.

1.4. The End to End Argument

[Principles] details the premises on which the Internet community has built its protocols for the past thirty years; among these premises is the end to end argument, which suggests that a network service which is useful to an application is by definition a service which the application can use at the edge to achieve a purpose all the way from itself to its peer, and in each system en route. This argument concludes that concentrating intelligence at the end or edge point is superior to embedding unnecessary intelligence in the network, because it is the end or edge that understands what needs to be achieved.

Differentiated Services modifies that model somewhat, seeing the edge as the boundary router of a service provider's network rather than or perhaps in addition to the end system itself. We agree that this clarification is necessary, in that service provider boundary routers invoke vast quantities of routing and other policy, and implementing policies such as described previously in this memo is a logical function of that boundary router.

But we also observe that the edge or end system may have specific expectations that map to the contracts that its owners write. If Voice on IP is to work well, it needs some form of "Low Delay Low Loss" service, for example, and it needs it in every service provider network that it passes. The implementation of the service may vary in each network, but the effect of the implementation must be that the relevant datagrams must experience low delay, low variation in delay, and a low loss rate. If some service provider en route fails to provide that service, the fact that the others supported it may be cold comfort; the application will not work anywhere near as well end to end as it otherwise would.

We therefore argue that a set of Per-Hop Behaviors that implement an IP Precedence service are useful end-to-end, and universal definition of a set of Per-Hop Behaviors to support IP Precedence is useful to essentially all service providers.

2. IP Precedence Proposal

With that context, we now proceed to define an IP Precedence service, using Per-Hop Behaviors as the vehicle, and incorporating the Assured Service for the purpose of contract management.

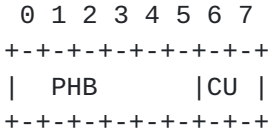
2.1. Intended Semantics

Intuitively, we wish to provide a set of queue or class selectors, each with drop preference according to Clark's Assured Service Model. We want, therefore, to provide pairs of PHBs for each queue or class; one PHB for the class marks the traffic "in profile", and one marks it "out". Traffic with different queue selector values may be relatively reordered without concern, but the "in/out" bit should not cause traffic reordering among traffic marked with the same queue selector.

The number of queues or classes that are specifiable must, in the immortal words of Mike O'Dell, be "more than three, less than nine, and probably a power of two." We believe that eight classes are required in order to support service providers' marketing of similar contracts at varying prices, or specific traffic engineering models. In addition, in this set of PHBs, one bit is used as the "in/out" bit. We also note that 802.1p is said to be an important service coming out Real Soon Now, and having three bits of IP layer queue selector to map to three bits of link layer queue selector is a good match.

2.2. Proposed Service Identifiers

The Differentiated Services proposal suggests that the DS byte is structured in this way:



We note that the existing IP Precedence field is located in bits zero through two of that octet, and that current implementations exist that perform services similar to this proposal using those bits; a simple prototype of the proposal can therefore be quickly deployed using configuration parameters using such implementations. We also note that IP systems today understand the location of the IP Precedence

field, and observe that if the bits associated with this variation on IP Precedence are in the same place, significant failures are not likely during deployment of the facility. In other words, the code need not be ubiquitous even in a single service provider's network if we are careful in our selection of bits. This argues that the bits we would like to use for this service are exactly the same set as [IP]'s Precedence bits, or a set subsuming that set with similar semantics.

We therefore propose that the following PHB numbers be selected:

111 1 00	precedence 7, in profile
111 0 00	precedence 7, out of profile
110 1 00	precedence 6, in profile
110 0 00	precedence 6, out of profile
101 1 00	precedence 5, in profile
101 0 00	precedence 5, out of profile
100 1 00	precedence 4, in profile
100 0 00	precedence 4, out of profile
011 1 00	precedence 3, in profile
011 0 00	precedence 3, out of profile
010 1 00	precedence 2, in profile
010 0 00	precedence 2, out of profile
001 1 00	precedence 1, in profile
001 0 00	precedence 1, out of profile
000 1 00	precedence 0, in profile
000 0 00	precedence 0, out of profile

In essence, a higher precedence (queue or class number) should afford a higher probability of timely delivery than a lower precedence packet, and in-profile traffic of any precedence should have a higher probability of delivery than out of profile traffic of the same precedence. If there is comparison among classes, as in a simple drop preference or simple priority queuing model, in-profile traffic of any precedence should have a greater probability of timely delivery than out of profile traffic of any precedence, without loss of sequence within a precedence. In the implementation, one could expect this to be implemented as some combination of drop preference (emphasis being on the probability of delivery), and queue characteristics (emphasis on timeliness of delivery).

Other PHBs, those whose two least significant bits are non-zero, are outside the scope of this specification and are not further discussed in this memo.

It can be argued that, since the PHBs are in fact indices in a case statement, there is no substantive reason that the exact values chosen above need be chosen. These specific values are chosen so as to be backward compatible with [IP]'s IP Precedence enumeration, and so that the in/out bit selected is contiguous with the other numbers.

The reason an in/out bit is selected, rather than letting there be some number of of "in" values and a common "out of profile" PHB, relates to cases where precedence is selecting a queue. If all traffic is in the same queue, a single PHB is clearly sufficient to mark that traffic which is out of profile. With multiple queues, however, one could imagine assigning different WFQ weights to traffic in the same queue which is in or out of profile, as well as providing different drop probabilities.

The astute reader will note that the default PHB, whose value is zero, is relegated to "routine, out of profile" traffic status; this is consistent with current IP practice, and makes any other setting of the field a desirable improvement, encouraging deployment.

2.3. Intended PHB Modifications

This memo contemplates two algorithms for setting or changing the PHB value. One algorithm, typically executed in the originating host application or its first-hop router, sets the PHB to a given precedence, in or out of profile, according to a policy set by the network administration. The other, typically executed in the first hop router of a routing domain (next to the host, at ingress to a service provider, etc.), may change it from "in profile" to "out of profile" according to the service level agreement in force.

Clearly, there is nothing to stop a service provider from setting it to another PHB, including changing the effective precedence or using some other service. If the service provider does so, however, he gives up whatever semantic was intended by the originator, losing information and perhaps losing the benefit of the service on an end to end basis. Such policies therefore call for wisdom on the part of the network administration.

3. Potential Implementation Strategies

We now discuss a number of possible implementation strategies. These are each examples: no one approach is mandated, and these are not the only possible implementations.

3.1. Simple Drop Preference

The simplest implementation of this service is simple drop preference in a simple FIFO queue. In this case, "higher probability of timely delivery" translates directly as "higher probability of delivery", with "out of profile" traffic making way for "in profile" traffic, and lower precedence for higher.

Among these PHBs, we assume that the interface implements a Random Early Detection algorithm, and that the min-threshold and max-threshold values associated with various PHBs rise in this sequence:

```
111 1 00  precedence 7, in profile      (Highest probability
110 1 00  precedence 6, in profile       of delivery)
101 1 00  precedence 5, in profile
100 1 00  precedence 4, in profile
011 1 00  precedence 3, in profile
010 1 00  precedence 2, in profile
001 1 00  precedence 1, in profile
000 1 00  precedence 0, in profile
111 0 00  precedence 7, out of profile
110 0 00  precedence 6, out of profile
101 0 00  precedence 5, out of profile
100 0 00  precedence 4, out of profile
011 0 00  precedence 3, out of profile
010 0 00  precedence 2, out of profile
001 0 00  precedence 1, out of profile  (Lowest probability
000 0 00  precedence 0, out of profile   of delivery)
```

The strength of this approach is that it maintains order as specified, and drops the lowest precedence traffic first. The weakness of the approach is that no way is afforded to make a demonstrable difference in the variation in queuing delay experienced by the various precedences, only the difference in drop probability.

3.2. Priority Queues with Drop Preference

Another approach employs a queue per precedence, using one bit of the PHB as a drop preference within the queue. RED is used

within the queues according to its usual parameters, but with in-profile traffic having a higher min-threshold and max-threshold than out of profile traffic, and therefore experiencing a higher probability of timely delivery. Queues are ranked in priority order so that each queue, from the perspective of the next lower priority queue, implements a "low loss low delay" service. Out of profile traffic should consider the presence of lower precedence in-profile traffic in the calculation of drop probability.

The strength of this approach is that order is maintained within each precedence queue, but higher precedence traffic may be sent before lower precedence traffic. It has a weakness, however, in that apart from admission and policing, it affords lower precedence traffic no assurance of eventual transmission.

3.3. Round Robin Queuing with Drop Preference

Like the previous one, this approach employs a queue per precedence, using the one bit of the PHB as a drop preference within the queue. RED is used within the queues according to its usual parameters, but with in-profile traffic having a higher min-threshold and max-threshold than out of profile traffic. However, each queue is emptied at some rate, in round-robin order, rather than being given simple priority service.

The strength of this approach is that order is maintained within each precedence queue, but higher precedence traffic may be sent before lower precedence traffic. It also avoids the lockout issue that priority queuing systems experience. A counter-intuitive scenario can occur, however, if a high rate queue is heavily utilized while a lower rate queue is under-utilized; a packet directed to the lower rate queue can actually be better protected from loss and variation in delay when placed in an empty or very short queue.

3.4. Virtual Circuit or Virtual Channel Selection

The difference between this approach and Round Robin Queuing with Drop Preference is somewhat academic. If one has a serial line to a routing neighbor, and manages using a load sharing algorithm, the load sharing algorithm in some sense emulates the way the line would behave if it were in reality a number of different lines, or if it were one channelized line. In a virtual circuit selection model, the emulation becomes reality

- one deploys a set of rate-limited VCs to a routing neighbor, and uses them in the same way one would otherwise have used queues.

The strengths and weaknesses are very similar to those of Round Robin Queuing, except that this allows one to capitalize on the capabilities of a link layer such as ATM or Frame Relay.

3.5. IEEE 802.1d (previously 802.1p) Service Marks

The difference between this approach and Round Robin Queuing with Drop Preference is also somewhat academic; an 802.1d switch employs round robin queuing within itself, so the queue management is again deployed through the link layer network.

It is worth noting, however, that the bits must be mapped: 802.1d traffic classes are a three bit number, which has an interesting set of rules. If the switch implements eight classes, the number selects the class. If it implements four classes, the two most significant bits of the number select the class and the least significant bit has no defined utility. If it implements two classes, the most significant bit selects that class. We therefore suggest this mapping algorithm:

(1) If an 802.1d switch implements eight classes, the mapping from IP Precedence to 802.1d traffic class is to place the precedence number (bits zero through two of the PHB) into the traffic class field.

(2) If an 802.1d switch implements one, two, or four classes, the mapping from IP Precedence to 802.1d traffic class is to place the two most significant bits of the precedence number (bits zero and one of the PHB) into the traffic class field's most significant bits, and copy the in/out bit (bit three) into the least significant bit of the traffic class. In this manner, it is available should the switch decide to consider it a drop preference bit. A corollary suggestion is being submitted to IEEE 802.1.

4. Acknowledgments

The authors note that there were a number of reviewers even of the first drafts of this note, whose inputs are very much appreciated.

5. References

[IP] RFC 791, "Internet Protocol". J. Postel. Sep-01-1981.

[HOSTREQ]
RFC 1122, "Requirements for Internet hosts - communication layers". R.T. Braden. Oct-01-1989.

[FRAMEWORK]
Nichols, "Differentiated Services Operational Model and Definitions", 02/11/1998, draft-nichols-dsopdef-00.txt

[PRINCIPLES]
RFC 1958, "Architectural Principles of the Internet". B. Carpenter. June 1996.

[ASSURED]
Clark and Wroclawski, "An Approach to Service Allocation in the Internet", 08/04/1997, draft-clark-diff-svc-alloc-00.txt

[BROKER]
Nichols and Zhang, "A Two-bit Differentiated Services Architecture for the Internet", 12/23/1997, draft-nichols-diff-svc-arch-00.txt

6. Security Considerations

The Differentiated Services Architecture explicitly requires each network to guard its own doors; if a system behaves in a manner inappropriate to its contracts, the intended behavior is that the system's communications will experience greater unreliability and may be shut down entirely, by way of a punishment. This proposal changes this in no way - it makes the situation no better and no worse.

This said, there is a backwards compatibility consideration which is one of the primary motivations for the submission of this idea, which can behave like a security issue. This is that RFC 791 reserves IP Precedence values 6 and 7 for router-to-router traffic, and many routers in the internet use this fact to isolate network control traffic during outage recovery and route changes.

To insure continued stability, it is vital that a domain with legacy routers carefully allocate their PHB's to avoid overloading the drop preference controls on the legacy equipment. Thus, we recommend that domains use PHBs with the pattern 11XXXX, when legacy routers are in the path, only for critical routing traffic such as inter-router keep-alive and route update messages.

7. Author's Addresses

Fred Baker
Cisco Systems
519 Lado Drive
Santa Barbara, California 93111
Phone: (408) 526-4257
Email: fred@cisco.com

Scott Brim
Newbridge Networks Inc.
146 Honness Lane
Ithaca, New York 14850
Phone: (607) 273-5472
Email: swb@newbridge.com

Tony Li
Juniper Networks, Inc.
385 Ravendale Drive
Mountain View, CA 94043
Phone: (650) 526-8000

Email: tli@juniper.net

Frank Kastenholz
Argon Networks
25 porter rd
Littleton ma 01460
Phone: (978) 386-0665
Email: kasten@argon.com

Shantigram Jagannath
Bay Networks
3 Federal Street,
Billerica, MA -01821
Phone: (978) 916-8598
Email: jagan@baynetworks.com

John K. Renwick
Ascend Communications
High-Performance Networking Division
10250 Valley View Rd
Eden Prairie, MN 55344
Phone: (612) 996-6847
Email: jkr@min.ascend.com