Differentiated Services WG   D. Black, EMC Corporation
INTERNET-DRAFT
Document: draft-black-diffserv-tunnels-00.txt   October 1999

Differentiated Services and Tunnels

Status of this Memo

A revised version of this draft document will be submitted to the RFC editor as a Proposed Standard for the Internet Community. Discussion and suggestions for improvement are requested. This document will expire before May, 2000.

1. Abstract

This draft discusses the interaction of Differentiated Services (diffserv) [RFC-2474, RFC-2475] with IP tunnels of various forms. The discussion of tunnels in the diffserv architecture [RFC-2475] has been found to provide insufficient guidance to tunnel designers and implementers. With the aim of providing such guidance, this document describes two conceptual models for the interaction of diffserv with IP tunnels and employs them to explore the resulting configurations and combinations of functionality. An important consideration is how and where diffserv traffic conditioning should be performed in the presence of tunnel encapsulation/decapsulation. A few simple mechanisms are also proposed that limit the complexity that tunnels would otherwise add to the diffserv traffic conditioning model; these mechanisms are also generally useful in

situations where more general traffic conditioning is inappropriate or unavailable. Security considerations for IPsec tunnels place some limits on possible functionality in some circumstances.

WARNING: The current status of this draft is highly preliminary; its major purpose is to foster discussion within the working group. Above and beyond the usual cautionary notice about not relying on Internet-Drafts, implementers are specifically warned that significant changes are expected to the contents of this draft.

2. Conventions used in this document

An IP tunnel encapsulates IP traffic in another IP header as it passes through the tunnel; the presence of these two IP headers is a defining characteristic of IP tunnels. The inner IP header is that of the original traffic; an outer IP header is attached and detached at tunnel endpoints. In general, network nodes within a tunnel operate solely on the outer IP header, and hence diffserv-capable nodes within a tunnel can only access and modify the DSCP field in the outer IP header (e.g., for an encrypted tunnel, interior nodes cannot access the inner IP header). The terms "tunnel" and "IP tunnel" are used interchangeably in this document.

This document considers tunnels to be unidirectional; bi-directional tunnels are composed of two unidirectional tunnels carrying traffic in opposite directions between the same pair of tunnel endpoints. A tunnel consists of an ingress where traffic enters the tunnel and is encapsulated by addition of the outer IP header, an egress where traffic exits the tunnel and is decapsulated by removal of the outer IP header, and interior nodes through which tunneled traffic passes between ingress and egress. This document does not make any assumptions about routing and forwarding of tunnel traffic, and in particular neither requires nor forbids route pinning of any form.

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].

Text in single square brackets labeled "Author's note:" (e.g., [Author's note: this is a note from the author.]) is editorial in nature and will be addressed in a future version of this document.

3. Diffserv and Tunnels Overview

Tunnels range in complexity from simple IP-in-IP tunnels [RFC-2003] to complex multi-protocol tunnels, such as IP in PPP in L2TP in IPsec transport mode [RFC-1661, RFC-2401, RFC-2661]. The most general tunnel configuration is one in which the tunnel is not end-to-end, i.e., the ingress and egress nodes are not the source and destination nodes for traffic carried by the tunnel. If the ingress or egress nodes do coincide with the end-to-end source or destination (respectively), the result is a simplification of this general configuration to which much of the analysis in this document remains applicable.

A primary concern for differentiated services is the use of the Differentiated Services Code Point (DSCP) in the IP header; see [RFC-2474, RFC-2475] for more extensive descriptions of the DSCP field and the diffserv architecture. Diffserv permits intermediate nodes to examine and change the value of the DSCP, which may result in the DSCP value in the outer IP header being modified between tunnel ingress and egress. When a tunnel is not end-to-end, there are circumstances in which it may be desirable to propagate the DSCP and/or some of the information that it contains to the outer IP header on ingress and/or back to inner IP header on egress. The current situation facing tunnel implementers is that [RFC-2475] offers some guidance, but is insufficient in detail. In contrast the EF PHB specification [RFC-2598] may be too specific (in 20/20 hindsight) because its requirement to use EF in the outer header of tunneled EF packets is unworkable in domains that do not support EF, and excludes other techniques for obtaining sufficient conditioning for tunneled EF traffic. In fairness to the authors of that RFC, this particular requirement responds to a guideline in the diffserv architecture RFC, specifically G.7 in Section 3 of [RFC-2575]; that guideline is also in need of revision as it is based on over-simplified assumptions about how tunnels are deployed with respect to DS domain boundaries.

The first issue raised by IP tunnels is the relationship of diffserv domain boundaries and traffic conditioning functionality to tunnel ingress and egress processing. This document proposes an approach in which traffic conditioning is performed in series with tunnel ingress or egress processing, not in parallel. This approach does not create any additional paths that transmit information across a tunnel endpoint; all diffserv information is contained in the DSCPs in the IP headers. IPsec requires that this be the case to preserve security properties at the egress of IPsec tunnels, but this model also avoids introducing out-of-band inputs to diffserv traffic conditioner blocks, which would complicate them. [Author's note: This needs to be updated to coordinate with the conceptual model draft; the conclusion won't change, but more detailed rationale will appear, along with a citation of that document.] Diffserv domain boundaries can then be positioned as appropriate for the set of traffic conditioning blocks and tunnel processing modules. One configuration of interest involves a diffserv domain boundary that passes through (i.e., divides) a network node; it is acceptable to split the boundary to create a DMZ-like region between the domains that contains the tunnel ingress or egress processing. Diffserv traffic conditioning is not appropriate for such a DMZ-like region, as that traffic conditioning is part of the operation and management of one or more diffserv domains.

4. Conceptual Models for Diffserv Tunnels

There are two important conceptual traffic conditioning models for IP tunnels. For clarity, the initial discussion of these models assumes a fully diffserv-capable network. Configurations in which this is not the case are taken up in Section 4.2.

4.1 Conceptual Models for Fully DS-capable Configurations

The first conceptual model is a uniform model that views IP tunnels as artifacts of the end to end path from a traffic conditioning standpoint; tunnels may be necessary mechanisms to get traffic to its destination(s), but have no significant impact on traffic conditioning. In this model, any packet has exactly one DS Field that is used for traffic conditioning at any point, namely the DS field in the outermost IP header; all others are ignored. Implementations of this model copy the DSCP value to the outer IP header at encapsulation and copy the outer header's DSCP value to the inner IP header at decapsulation. Support for this model allows IP tunnels to be configured without regard to diffserv domain boundaries because diffserv traffic conditioning functionality is not impacted by the presence of IP tunnels.

The second conceptual model is a pipe model that views an IP tunnel as hiding the nodes between its ingress and egress so that they do not participate fully in traffic conditioning. In this model, a tunnel egress node uses traffic conditioning information conveyed from the tunnel ingress by the DSCP value in the inner header, and ignores (i.e., discards) the DSCP value in the outer header. This model cannot completely hide traffic conditioning within the tunnel, as the effects of dropping and shaping at tunnel interior nodes may be visible to nodes beyond the tunnel egress. One class of configurations for which this model is appropriate are situations in which the ingress and egress nodes belong to the same diffserv domain, but the IP tunnel may pass through other domains. In this case, the DSCP values from the ingress node are valid at the egress node. Effective use of this pipe model in configurations other than this single domain case generally require that an inter-domain TCA (Traffic Conditioning Agreement) exist between the diffserv domains containing the tunnel ingress and egress nodes in order to specify the interpretation of the DSCP values in the inner IP headers and the resulting traffic conditioning requirements.

The pipe conceptual model is also appropriate for situations in which the DSCP carries information that is destroyed by a node or nodes within the tunnel. For example, if transit between two domains is purchased via a tunnel that uses the EF PHB [RFC-2598], the drop precedence information in the AF PHB DSCP values [RFC-2597] will be destroyed unless something is done to preserve it; an IP tunnel is one possible preservation mechanism. A tunnel that crosses one or more non-diffserv domains between its DS-capable endpoints may experience a similar information destruction phenomenon due to the limited set of DSCP codepoints that are compatible with such domains.

4.2 Considerations for Partially DS-capable Configurations

If only the tunnel egress node is DS-capable, [RFC-2475] requires that node to take responsibility for any edge traffic conditioning required by the diffserv domain for tunneled traffic from outside the domain. If the egress node would not otherwise be a DS edge node, one way to meet this requirement is to perform edge traffic conditioning at an appropriate upstream DS edge node or nodes within the tunnel, and copy the DSCP value from the outer IP header to the inner IP header as part of tunnel decapsulation processing. This preserves correct operation of the DS domain independent of how the tunnel ingress node handles the DSCP values in the inner IP headers. A second alternative discards the outer DSCP value as part of decapsulation processing, reducing the resulting traffic conditioning problem and requirements to those of an ordinary DS ingress node. One exception that the existence of the tunnel may complicate placing some traffic conditioning responsibility on the upstream node because that node would then be the tunnel ingress node, not the immediately upstream tunnel interior node.

If only the tunnel ingress node is DS-capable, [RFC-2475] requires that traffic emerging from the tunnel be compatible with the network at the tunnel egress. If tunnel decapsulation processing discards the outer header's DSCP value without changing the inner header's DSCP value, then the DS-capable tunnel ingress node MUST set the inner header's DSCP to a value compatible with the network at tunnel egress. The value 0 (DSCP of 000000) is often used for this purpose in existing tunnel implementations. If the egress network is known to implement IP precedence as specified in [RFC-791], then some or all of the eight class selector DSCP codepoints defined in [RFC-2474] are usable. Use of any DSCP codepoints other than the class selectors for this purpose is NOT RECOMMENDED, as compatible operation would then require diffserv traffic conditioning at the tunnel egress node that is not DS-capable. Based on the existing use of the value 0, setting the DSCP to 0 is RECOMMENDED when a signaling convention is needed to inform the tunnel egress that a DSCP value in a packet carries no useful information. This is appropriate for the outer IP header's DSCP when a tunnel fits the pipe conceptual model, and may be useful for the inner IP header's DSCP for tunnels that do not have a TCA in place between the ingress and egress DS domains.

5. Ingress Functionality

As described in Section 3 above, this draft is based on an approach in which diffserv functionality and/or out-of-band communication paths are not placed in parallel with tunnel encapsulation processing. This model allows three possible locations for traffic conditioners with respect to tunnel encapsulation processing, as shown in the following diagram that depicts the flow of IP headers through tunnel encapsulation:

```
                                     +--------- [[2 - Outer]] -->>
                                    /
                                   /
>>---- [[1 - Before]] -------- Encapsulate ---- [[3 - Inner]] -->>
```

Of these three possible locations, [[3 - Inner]] SHOULD NOT be utilized for general traffic conditioning because it requires traffic conditioning functionality to reach inside the packet in order to operate on the inner IP header. This is difficult in general, and is impossible for IPsec tunnels and any other tunnels that employ encryption or cryptographic integrity checks. Hence traffic conditioning at [[3 - Inner]] can only be done as part of tunnel encapsulation processing, complicating both the encapsulation and traffic conditioning implementations for little apparent benefit. In many cases, the desired functionality can be achieved via a combination of traffic conditioners in the other two locations, both of which can be specified and implemented independently of tunnel encapsulation processing. Tunnel designs and specifications SHOULD allow diffserv traffic conditioning to be deployed at [[1 - Before]] and [[2 - Outer]].

An exception in which functionality may need to be deployed at [[3 - Inner]] occurs when the tunnel egress is not DS-capable, as discussed in Section 4.2 above. Setting the inner DSCP to 0 as part of encapsulation addresses a large portion of these cases, and the maximum functionality that should be provided is setting the inner DSCP to one of the class selector codepoint values. This level of functionality (set DSCP to one of the class selector codepoint values) is also appropriate for [[2 - Outer]] in configurations that do not have more general traffic conditioning in that location.

The following table summarizes the achievable relationships among the Before (B), outer (O), and inner (I) DSCP values and the corresponding locations of traffic conditioning logic.

```
Relationship       Traffic Conditioning Location(s)
------------       -------------------------------
B  = I  = O  = B   No traffic conditioning required
B != I  = O != B   [[1 - Before]]
B  = I != O != B   [[2 - Outer]]
B != I != O  = B   Limited support as part of encapsulation
                     processing, instead of [[3 - Inner]]; I can
                     be to one of class selectors.  May be
                     accomplished in some cases via a combination
                     of [[1 - Before]] and [[2 - Outer]].
B != I != O != B   Some combination of the above three cases.
```

Minimizing the number of traffic conditioning blocks is recommended as a general design principle. Implementers are cautioned that traffic conditioning may still be required even if DSCP values are not changed for purposes such as rate and burst limitation.

[Author's note: Is the above table useful?]

6. Egress Functionality

As described in Section 3 above, this draft is based on an approach in which diffserv functionality and/or out-of-band communication paths are not placed in parallel with tunnel encapsulation processing. This model allows three possible locations for traffic conditioners with respect to tunnel decapsulation processing, as shown in the following diagram that depicts the flow of IP headers through tunnel encapsulation:

```
>>----[[5 - Outer]]-------------+
                                 \
                                  \
>>----[[4 - Inner]] --------- Decapsulate ---- [[6 - After]] -->>
```

As was the case for [[3 - Inner]] at tunnel ingress nodes, [[4 - Inner]] SHOULD NOT be employed for general traffic conditioning because it requires reaching inside the packet to operate on the inner IP header. See the discussion of [[3 - Inner]] in Section 5 for further explanation.

In contrast to the encapsulation case, the elimination of parallel functionality and data paths from decapsulation causes a potential loss of information. As shown in the above diagram, decapsulation reduces two DSCP values to one DSCP value, and hence necessarily loses information in the most general case, even if arbitrary functionality is allowed. Beyond this, allowing arbitrary functionality poses a structural problem, namely that the DSCP value from the outer IP header should to be presented as an out-of-band input to the traffic conditioning block at [[6 - After]], significantly complicating the traffic conditioning model and implementations at that location. To avoid such complications, this document proposes a simpler approach of defining a few primitive DSCP combination operations that can be performed as part of decapsulation, leaving the full generality of traffic conditioning functionality to be implemented at [[5 - Outer]] and [[6 - After]]. These operations should be straightforward to add to tunnel implementations and are expected to yield most of the benefits of a more fully general approach without imposing the complexity of such an approach on tunnel implementations.

The following four primitive DSCP operations are proposed for incorporation into tunnel decapsulation. Each takes an Inner and an Outer DSCP value as arguments and produces a Result DSCP value for the IP header of the decapsulated packet. The operations are described in "Name: Pseudo-code specification" format.

```
(1) Discard: Result = Inner;
(2) Overwrite: Result = Outer;
(3) Conditional Overwrite: If (Outer != 0), Result = Outer;
                           Else Result = Inner;
(4) Conditional Discard: If (Inner != 0), Result = Inner;
                         Else Result = Outer;
```

The rationale for the choice of these functions is that of the two DSCP values, one of them usually contains useful information, and the other is of little value. In terms of the conceptual models discussed in Section 3, Discard corresponds to the pipe model, Overwrite corresponds to the uniform model, and the two Conditional operations are motivated by the use of 0 as an "escape value" indicating that the useful information is in the other header's DSCP (see Section 4.2). IPsec tunnels and other tunnels with similar security properties MUST default to Discard, and SHOULD not choose a different function in the absence of an adequate security analysis.

[Author's note: The above section is particularly tentative, and needs WG discussion, starting from whether the "simpler approach" is simple enough or too simple. Recommendations about what the list should be and what MUST/SHOULD/MAY be implemented in tunnels will emerge from that discussion. The author's current inclination is that at least one of the first two functions is a MUST (but choosing which one to implement, or implementing both is a MAY), the third function is a SHOULD, and the fourth function is a MAY. The IPsec discussion probably needs to be expanded.]

6.1 Limited Decapsulation Functionality Rationale

As a sanity check on the simpler approach proposed in the above section (6), this subsection considers a situation in which a more complex approach might be required. The four DSCP combination functions proposed above are actually selection functions; one of the two DSCPs is selected to pass onward as the DSCP for the decapsulated packet. This is a poor match to situations in which both DSCPs are carrying information that is needed to perform outgoing traffic conditioning (i.e., at [[6 - After]]) correctly.

As an example, consider a situation in which two different AF groups [RFC-2597] are being used by the two domains at the tunnel endpoints, there is an intermediate domain along the tunnel that uses RFC 791 IP precedences, this domain is transited by setting the DSCP to zero, and the tunnel egress is at a node that would not otherwise be an edge node for that diffserv domain. This situation is shown in the following IP header flow diagram where I is the tunnel ingress node, E is the tunnel egress node and the vertical lines are domain boundaries. The node at the left-hand vertical line sets the DSCP in the outer header to 0 in order to obtain compatibility with the middle domain:

```
                     |                   |
               +-----|-------------------|------+
              /      |                   |       \
>>-----------I-------|-------------------|--------E---------->>
                     |                   |
   Ingress DS Domain        RFC 791         Egress DS domain
                         IP Precedence
                             Domain
```

In this situation, the DS edge node for the egress domain (i.e., the node at the right-hand vertical line) can select the appropriate AF group (e.g., via an MF classifier), but cannot reconstruct the drop precedence information that was removed from the outer header when it transited the RFC-791 compliant domain (although it can construct new information via metering and marking). The original drop precedence information is preserved in the inner IP header's DSCP, and could be combined with the AF class selection communicated via the outer IP header's DSCP. On the other hand, the same result can be obtained in most cases by placing traffic conditioning functionality at location [[6 - After]] in the tunnel egress node, and discarding the outer header. This works as long as the appropriate AF class for the egress domain is implicit in the fact that traffic is emerging from the tunnel, or can be determined via BA or MF classification of the emerging traffic.

[Author's note: This is a flimsy start at "proof by example". The intention is to start a requirements discussion in the WG and capture the results in the next version of this draft.]

7. Summary of Advice to Tunnel Implementers

[Author's note: To be written after WG discussion; WG consensus is a prerequisite to this sort of text that will be full of MUST/SHOULD/MAY items. Will probably also add some discussion and recommendations about existing tunnel specifications in light of this advice.]

8. Diffserv and Protocol Translators

A related issue involves protocol translators, of which a specific example is the Stateless IP/ICMP translator [SIIT]. These translators are not tunnels because they do not add or remove a second IP header to/from packets (e.g., in contrast to IPv6 over IPv4 tunnels [RFC-1933]) and hence do not raise concerns of information propagation between inner and outer IP headers. The primary interaction between translators and diffserv is that the translation boundary is likely to be a diffserv domain boundary (e.g., the IPv4 and IPv6 domains may have different policies for traffic conditioning and DSCP usage), and hence such translators

SHOULD permit the insertion of diffserv edge node processing, including traffic conditioning and/or the simplified ingress functional addition discussed in Section 5.

9. Security Considerations

The security considerations for the diffserv architecture discussed in [RFC-2474, RFC-2475] apply when tunnels are present; readers are referred to those documents for further background. One of the requirements noted there is that a tunnel egress node in the interior of a diffserv domain is the DS ingress node for traffic exiting the tunnel, and is responsible for performing appropriate traffic conditioning. The primary security implication is that the traffic conditioning is responsible for dealing with theft- and denial-of-service threats posed to the diffserv domain by traffic exiting from the tunnel. The IPsec architecture [RFC-2401] places a further restriction on tunnel egress processing; the outer header MUST be discarded unless the properties of the traffic conditioning that results are known and have been adequately analyzed for security vulnerabilities. This includes both the [[5 - Outer]] and [[6 - After]] traffic conditioning blocks on the tunnel egress node, if present, and may involve traffic conditioning performed by an upstream DS-edge node that is the DS domain ingress node for the encapsulated tunneled traffic.

10. References

[RFC-791] J. Postel, "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC-1661] W. Simpson, "The Point-to-Point Protocol (PPP)", STD 51, RFC 1661, July 1994.

[RFC-1933] R. Gilligan and E. Nordmark, "Transition Mechanisms for IPv6 Hosts and Routers", RFC 1933, April 1996.

[RFC-2003] C. Perkins, "IP Encapsulation within IP,", RFC 2003, October 1996.

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

[RFC-2401] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

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

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

December 1998.

[RFC-2597] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski, "Assured Forwarding PHB Group", RFC 2597. June 1999.

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

[RFC-2661] W. Townsley, A. Valencia, A. Rubens, G. Pall, G. Zorn, and B. Palter. "Layer Two Tunneling Protocol "L2TP"", RFC 2661, August 1999.

[SIIT] E. Nordmark, "Stateless IP/ICMP Translator (SIIT)", draft-ietf-ngtrans-siit-06.txt, Work in Progress, IETF ngtrans WG, July 1999.

[Author's note: This needs to be extended by additional tunnel RFC references as part of writing Section 7, the references section of the Tunnel MIB RFC (RFC 2667) provides a good starting point.]

11. Acknowledgments

Some of this material is based on discussions with Brian Carpenter, and is derived in part from his presentation on this topic to the diffserv WG at its summer 1999 meeting in Oslo. Credit is also due to a significant number of people working on tunnel specifications [names will appear here in a future version] who have discovered limitations of the diffserv architecture RFC (2475) in the area of tunnels. Their kind patience with the time it has taken to address this set of issues has been greatly appreciated.

12. Author's Address

David L. Black
EMC Corporation
42 South St.
Hopkinton, MA   01748
Phone: +1 (508) 435-1000 x75140
Email: black_david@emc.com