Tag Switching: Tag Stack Encodings

draft-rosen-tag-stack-00.txt

Status of this Memo

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 draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

To learn the current status of any Internet-Draft, please check the "1id-abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).
1. Introduction

[1] describes a need to encode a "Tag Stack" into a packet. This document specifies how the Tag Stack is encoded on Tagged Packets which are sent over PPP data links and over LAN data links.

1.1. Specification of Requirements

In this document, several words are used to signify the requirements of the specification. These words are often capitalized.

MUST

This word, or the adjective "required", means that the definition is an absolute requirement of the specification.

MUST NOT

This phrase means that the definition is an absolute prohibition of the specification.

SHOULD

This word, or the adjective "recommended", means that there may exist valid reasons in particular circumstances to ignore this item, but the full implications must be understood and carefully weighed before choosing a different course.
MAY

This word, or the adjective "optional", means that this item is one of an allowed set of alternatives. An implementation which does not include this option MUST be prepared to interoperate with another implementation which does include the option.

2. Encoding the Tag Stack

On both PPP and LAN data links, the Tag Stack is represented as a sequence of Tag Stack Entries. Each Tag Stack Entry is represented as a 4-byte quantity. This is shown in Figure 1.

| 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 | Tag Stack Entry
|---------------------------------------------|
| 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 | Stack
| ++++++++++++++++++++++++++++++++++++++++++++ | Entry
| | Tag | rsvd |CoS|S|     TTL     |
| ++++++++++++++++++++++++++++++++++++++++++++ |

Tag: tag value, 19 bits
rsvd: reserved, 3 bits
CoS: Class of Service, 2 bits
S: bottom of stack, 1 bit
TTL: 7 bits

Figure 1

The Tag Stack Entries appear after the end of all data link layer headers, but before any network layer headers. The entry representing the top of the Tag Stack appears earliest in the packet. The entry representing the bottom of the Tag Stack appears latest. The network layer packet immediately follows the Tag Stack Entry with the S bit set.

Each Tag Stack Entry is broken down into the following fields:

1. Stack End Bit (S)

   This bit is set to one for the last entry in the Tag Stack (i.e., for the bottom of the stack), and zero for all other Tag Stack Entries.

2. Time to Live (TTL)

   This seven-bit field is used to encode a time-to-live value.
When the first tag is pushed onto an untagged IP packet, the TTL value in the Tag Stack Entry MUST be set to the smaller of 127 and the value of the IP TTL field.

At every hop which tag switches a packet, the TTL value of the packet’s top Tag Stack Entry MUST be decremented by at least 1.

When the Tag Stack is popped, and the resulting Tag Stack is non-empty, the TTL value in the new top stack entry MUST be decremented by 1.

When the Tag Stack is popped, and the resulting Tag Stack is empty, and the network layer packet is an IP packet, the TTL value in the IP packet MUST be decremented by at least 1; it SHOULD be modified as follows. If the TTL value in the IP packet is less than or equal to 127, it is replaced with the TTL value from the last Tag Stack Entry. If the TTL value in the IP packet is greater than 127, it is replaced by the value which results from first subtracting the TTL value in the Tag Stack Entry from 127, and then subtracting that difference from the TTL value in the IP header. If this causes the TTL value in the IP header to become less than 1, the ordinary processing for an IP packet whose TTL has become less than 1 MUST be performed.

If at any time, the value of the TTL field in a Tag Stack Entry becomes less than 1, the packet MUST NOT be forwarded further. Depending on the tag value in the Tag Stack Entry, the packet may be silently discarded, or the packet may have its Tag Stack stripped off, and passed as an untagged packet to the ordinary processing for network layer packets which have exceeded their maximum lifetime in the network.

3. Class of Service (CoS)

This two-bit field is used to identify a class of service.

4. Reserved

These three bits are reserved. They MUST be set to zero when writing, and MUST be ignored when reading.

5. Tag Value

This 19-bit field carries the actual tag value.

A value of 0 represents the "Explicit NULL Tag". The presence of the Explicit NULL Tag means that the Tag Stack must be
popped, and the forwarding of the packet must then be based on
the resulting top tag in the Tag Stack, or, if the Tag Stack
becomes empty, on the network layer header.

3. Transporting Tagged Packets over PPP

The Point-to-Point Protocol (PPP) [PPP] provides a standard method
for transporting multi-protocol datagrams over point-to-point links.
PPP defines an extensible Link Control Protocol, and proposes a
family of Network Control Protocols for establishing and configuring
different network-layer protocols.

This section defines the Network Control Protocol for establishing
and configuring Tag Switching over PPP.

3.1. Introduction

PPP has three main components:


2. A Link Control Protocol (LCP) for establishing, configuring,
and testing the data-link connection.

3. A family of Network Control Protocols for establishing and
configuring different network-layer protocols.

In order to establish communications over a point-to-point link, each
end of the PPP link must first send LCP packets to configure and test
the data link. After the link has been established and optional
facilities have been negotiated as needed by the LCP, PPP must send
Tag Switching Control packets to enable the transmission of tagged
packets. Once the Tag Switching Control Protocol has reached the
Opened state, tagged packets can be sent over the link.

The link will remain configured for communications until explicit LCP
or Tag Switching Control Protocol packets close the link down, or
until some external event occurs (an inactivity timer expires or
network administrator intervention).
3.2. A PPP Network Control Protocol for Tag Switching

The Tag Switching Control Protocol is responsible for enabling and disabling the use of tag switching on a PPP link. It uses the same packet exchange mechanism as the Link Control Protocol (LCP). Tag Switching Control Protocol packets may not be exchanged until PPP has reached the Network-Layer Protocol phase. Tag Switching Control Protocol packets received before this phase is reached should be silently discarded.

The Tag Switching Control Protocol is exactly the same as the Link Control Protocol [1] with the following exceptions:

1. Frame Modifications

   The packet may utilize any modifications to the basic frame format which have been negotiated during the Link Establishment phase.

2. Data Link Layer Protocol Field

   Exactly one Tag Switching Control Protocol packet is encapsulated in the PPP Information field, where the PPP Protocol field indicates type hex 80?? (Tag Switching).

3. Code field

   Only Codes 1 through 7 (Configure-Request, Configure-Ack, Configure-Nak, Configure-Reject, Terminate-Request, Terminate-Ack and Code-Reject) are used. Other Codes should be treated as unrecognized and should result in Code-Rejects.

4. Timeouts

   Tag Switching Control Protocol packets may not be exchanged until PPP has reached the Network-Layer Protocol phase. An implementation should be prepared to wait for Authentication and Link Quality Determination to finish before timing out waiting for a Configure-Ack or other response. It is suggested that an implementation give up only after user intervention or a configurable amount of time.

5. Configuration Option Types

   None.
3.3. Sending Tagged Packets

Before any tagged packets may be communicated, PPP must reach the Network-Layer Protocol phase, and the Tag Switching Control Protocol must reach the Opened state.

Exactly one tagged packet is encapsulated in the PPP Information field, where the PPP Protocol field indicates either type hex 00?? (Tag Switching -- Unicast) or type hex 00?? (Tag Switching -- Multicast). The maximum length of a tagged packet transmitted over a PPP link is the same as the maximum length of the Information field of a PPP encapsulated packet.

The format of the Information field itself is as defined in section 2.

Note that two codepoints are defined for tagged packets; one for multicast and one for unicast. Once the TSCP has reached the Opened state, both tag switched multicasts and tag switched unicasts can be sent over the PPP link.

3.4. Tag Switching Control Protocol Configuration Options

There are no configuration options.

4. Transporting Tagged Packets over LAN Media

A pair of two byte ethertype values will be obtained, one representing "Tag Switching -- Unicast" and one representing "Tag Switching -- Multicast".

These can be used with either the Ethernet encapsulation or the 802.3 SNAP/SAP encapsulation to carry tagged packets.

Exactly one tagged packet is carried in each frame.

The Tag Stack Entries immediately precede the network layer header, and follow any data link layer headers, including any VLAN headers that may exist.
5. Security Considerations

Security considerations are not discussed in this document.

6. Authors’ Addresses

Eric C. Rosen
Cisco Systems, Inc.
250 Apollo Drive
Chelmsford, MA, 01824

E-mail: erosen@cisco.com

Dan Tappan
Cisco Systems, Inc.
250 Apollo Drive
Chelmsford, MA, 01824

E-mail: tappan@cisco.com

Dino Farinacci
Cisco Systems, Inc.
170 Tasman Drive
San Jose, CA, 95134

E-mail: dino@cisco.com

Yakov Rekhter
Cisco Systems, Inc.
170 Tasman Drive
San Jose, CA, 95134

E-mail: yakov@cisco.com

Guy Fedorkow
Cisco Systems, Inc.
250 Apollo Drive
Chelmsford, MA, 01824

E-mail: fedorkow@cisco.com
7. References