

MPLS
Internet-Draft
Intended status: Standards Track
Expires: April 15, 2013

D. Frost
S. Bryant
Cisco Systems
October 12, 2012

MPLS Generic Associated Channel (G-ACh) Test Session Control
draft-frost-mpls-test-session-00

Abstract

[RFC 6374](#) defines procedures for packet loss and throughput measurement in MPLS networks. Some forms of measurement rely on the existence of a stream of test messages that flows between measurement points, from which the loss and throughput characteristics of the underlying data channel are inferred. This document presents procedures for the establishment and maintenance of such test sessions.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on April 15, 2013.

Copyright Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in [Section 4.e](#) of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
1.1.	Terminology	3
1.2.	Requirements Language	3
2.	Overview	3
3.	Simple Session Control Protocol	4
3.1.	Message Format	5
3.2.	TLV Objects	6
3.3.	Session Setup	9
3.4.	Session Maintenance and Release	9
4.	Test Session Parameters	10
4.1.	Destination Test Identifier	10
4.2.	Source Test Identifier	11
4.3.	Packet Format	11
4.4.	Path Type	12
4.5.	Payload Size Range	13
4.6.	Maximum Transmission Rate	13
5.	Test Session Control	13
6.	Security Considerations	14
7.	IANA Considerations	14
7.1.	Allocation of Associated Channel Types	15
7.2.	Creation of MPLS Simple Session Control Protocol TLV Registry	15
7.3.	Creation of MPLS Simple Session Control Protocol Session Type Registry	15
8.	Normative References	16
	Authors' Addresses	16

[1.](#) Introduction

Procedures and protocol messages for packet loss, delay, and throughput measurement in MPLS networks are documented in [[RFC6374](#)]. Packet loss measurement, in that document, is classified as either direct or inferred: direct measurement is based on comparing transmit and receive counters for all data-plane traffic flowing over the channel, while inferred measurement is based on comparing the equivalent counters for a distinct stream of test traffic. Similarly, out-of-service throughput measurement entails validating the data-plane capacity of a channel by generating a stream of test traffic at a rate that meets or exceeds the expected capacity.

The Loss Measurement (LM) protocol defined in [RFC 6374](#) relies on the existence of a test traffic stream when used to conduct inferred LM or out-of-service throughput measurement. This document defines procedures for the setup and control of such test streams via the MPLS Generic Associated Channel (G-ACh) [[RFC5586](#)].

[1.1.](#) Terminology

Term	Definition

G-ACh	Generic Associated Channel
LM	Loss Measurement
SSCP	Simple Session Control Protocol
TLV	Type-Length-Value

[1.2.](#) Requirements Language

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 [[RFC2119](#)].

[2.](#) Overview

The objective is for a device acting as an LM querier to establish a test traffic stream to the LM responder in advance of initiating the LM session; this stream can then serve as the target of the LM operation, persisting until the operation is finished. Test session setup and maintenance proceeds according to the following process:

1. The querier determines the desired parameters for the test session, encodes them in a setup message as specified later in this document, and sends it to the responder. This message is transmitted periodically until either a response is forthcoming or a timeout occurs.

2. The responder, upon receiving the test session parameters, either accepts or rejects them. In either case, it formulates a response and sends it to the querier. The response indicates whether the session is accepted or rejected and, in the latter case, parameters that the responder considers acceptable. If the session was accepted, it is now considered "alive" at the responder, which maintains state for it until it times out or is explicitly released.
3. The querier, upon receiving the responder's message, knows whether the test session is now active. If not, it can retry the attempt using parameters the responder has indicated are acceptable. If so, it now does three things: it begins sending test traffic; it periodically sends a message refreshing/verifying the test session state; and it initiates an LM session that targets this test session.
4. The querier, when finished with the measurement operation, terminates the LM session, ceases sending test traffic, and sends an advisory message to the responder that the test session has ended.

In the remainder of this document the term "querier" is replaced by "initiator" in the context of test session control.

[3.](#) Simple Session Control Protocol

This document defines a new G-ACh protocol and associated Channel

Type:

Protocol	Channel Type
Simple Session Control Protocol	0xXXXX

For this Channel Type, the ACH SHALL NOT be followed by the ACH TLV Header defined in [RFC5586].

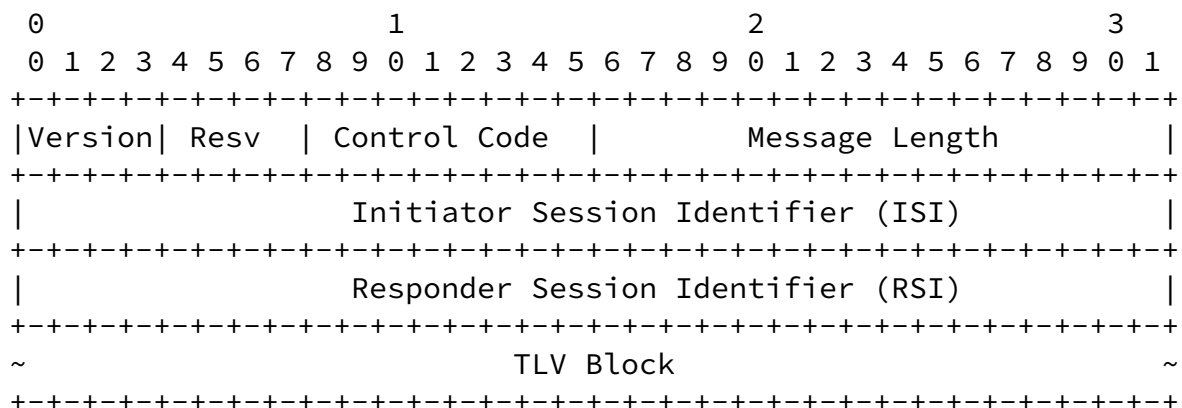
The Simple Session Control Protocol (SSCP) is a minimal "skeleton protocol" for the setup and control of point-to-point "sessions" over the G-ACh, where a session is defined abstractly as an initial agreement of application-specific parameters between the initiator and responder, followed by some form of state that is maintained between the two endpoints until either a timeout occurs or the session is explicitly released.

The only SSCP application discussed in this document is that of measurement test stream control. However, the SSCP has been defined

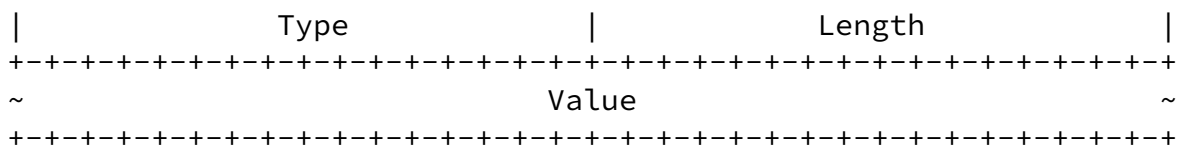
in a general form with the view that it may have other future applications.

3.1. Message Format

The following figure shows the format of an SSCP message, which follows the Associated Channel Header (ACH):



SSCP Message Format

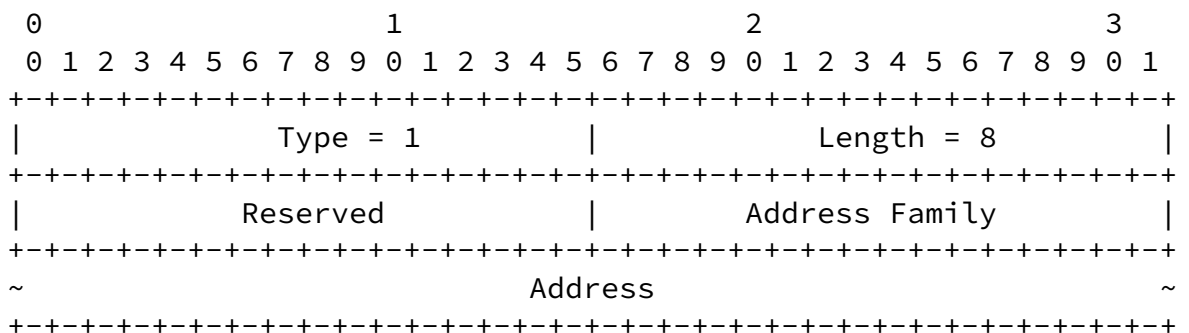


TLV Object Format

The Type field identifies the TLV Object; an IANA registry has been created to track the values of this field. Types 0-127 are reserved for use by the SSCP itself, with the rest available for application-specific allocation. The Length field specifies the length in octets of the Value field.

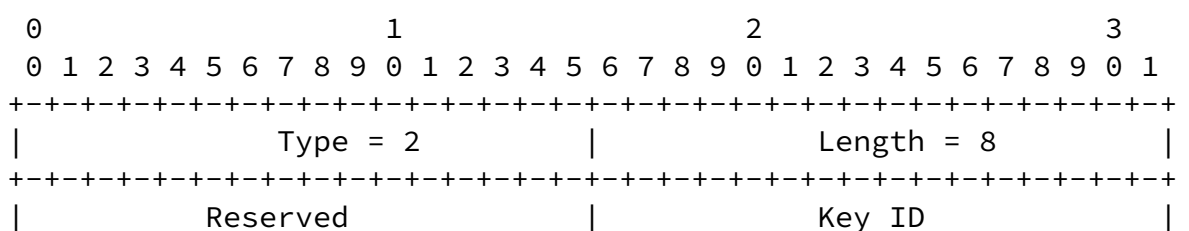
3.2. TLV Objects

3.2.1. Source Address



The Source Address allows the initiator to inform the responder of its address when sending an Initiate or Query message. The format of this object is identical to the Source Address TLV object described in [[I-D.ietf-mpls-gach-adv](#)].

3.2.2. Authentication



```

+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
~                                     Authentication Data                                     ~
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

The Authentication object allows the receiver of an SSCP message to verify the identity of the message source and the integrity of the message. The format and processing semantics of this object are specified in [[I-D.ietf-mpls-gach-adv](#)].

[3.2.3.](#) Session Type

```

      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Type = 3                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Session Type                               |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

The Session Type is included in Initiate and Query messages and indicates the type of session that the initiator seeks to establish. An IANA registry has been created to track the values for the Session Type.

[3.2.4.](#) Hold Time

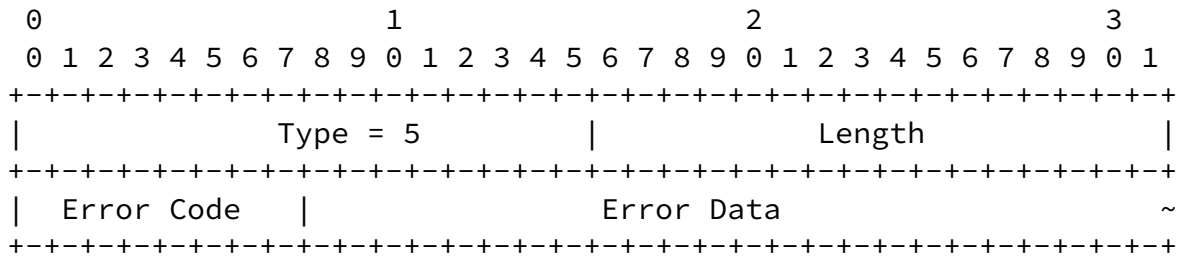
```

      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Type = 4                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Reserved                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

The Hold Time object indicates the amount of time, in seconds, the responder should keep this session alive if a refresh message is not received. When the hold timer expires, the responder discards all state associated with the session. When a refresh message is received, the responder resets its hold timer to the Hold Time.

[3.2.5.](#) Error Information



The Error Information object is included by the responder as part of a Reject message and identifies the reason for rejection in the form of an error code. Some codes may carry additional error information, in which case this information is placed in the Error Data field. The Error Data field has zero length unless otherwise noted below.

Error Code Meaning

Error Code	Meaning
0	Unspecified Error
1	Protocol Error
2	Resource Unavailable
3	Unsupported Session Type
4	Unsupported Parameter
5	Authentication Failed
6	Invalid Session

Unspecified Error: An unspecified error has prevented the requested session from being accepted. This code MUST NOT be used if a more specific code applies.

Protocol Error: A protocol error was found when parsing the incoming message.

Resource Unavailable: Node resources are not available to support the requested session.

Unsupported Session Type: Support for the Session Type indicated in the incoming message is not available.

Unsupported Parameter: Support for one or more of the requested session parameters is not available. The Error Data field consists of a sequence of TLV objects for the bad parameters, copied from the original request.

Authentication Failed: Authentication for the incoming message failed. A response message carrying this code MAY be sent as an alternative to silently dropping the offending message.

Invalid Session: The Responder Session Identifier in the incoming Refresh message is unknown or has been released.

[3.3.](#) Session Setup

The initiator begins by transmitting an Initiate message, i.e. a message with the Control Code set to Initiate. The Initiate message **MUST** also contain a single instance each of the Session Type and Hold Time objects.

Upon transmitting the first Initiate message, the initiator sets a retransmit timer. The message is retransmitted until either a response is received or a locally-determined timeout occurs. The retransmit period **SHOULD** be no shorter than three seconds.

When the responder receives an Initiate message, it determines whether it can support the requested session. If not, it sends a single Reject message to the initiator with the ISI copied from the Initiate message and with an Error Information object indicating the reason for rejection. In the case of an Unsupported Parameter error, the responder also includes a set of TLV objects that describe the parameters it supports, called the "Supported Parameters" set. This set includes the Hold Time object, which in this context indicates the longest hold time the responder supports for this session type. The other objects in the Supported Parameters set are specific to the session type.

If the responder can support the requested session, it sets the hold timer for the session to the value specified by the Hold Time object and sends a single Accept message to the initiator. The ISI of the Accept message is copied from the Initiate message, and the RSI is set at the responder's discretion.

An alternative to the above procedure is for the initiator to begin by sending a Query rather than an Initiate message. Upon receiving such a message, the responder responds with a Reject message that contains either an Error Information object or the Supported Parameters object set for this session type. The ISI of the Reject message is copied from the Initiate message.

[3.4.](#) Session Maintenance and Release

Following the acceptance of a session, the responder maintains state for the session until the session's hold timer expires or a Release message for the session is received. It **MAY** also terminate the session if an exceptional condition occurs; in this case it **SHOULD**

send a Reject message to the initiator.

In order to maintain the session over time, the initiator sends periodic Refresh messages containing the RSI signaled by the responder in its most recent Accept message for this session. The responder responds to a Refresh with an Accept message containing its RSI and the ISI received in the Refresh. The refresh interval SHOULD be less than one-third of the Hold Time for the session.

When the initiator is finished with the session, it sends a Release message containing the RSI signaled by the responder in its most recent Accept message for this session. Upon receiving a Release, the responder discards all state associated with the session.

4. Test Session Parameters

This document defines the following Session Type for use in establishing test traffic streams for packet loss and throughput measurement:

Session Type	Value
Measurement Test Session	0x0001

Test traffic streams are negotiated via the SSCP. This negotiation determines the format and flow characteristics of the streams.

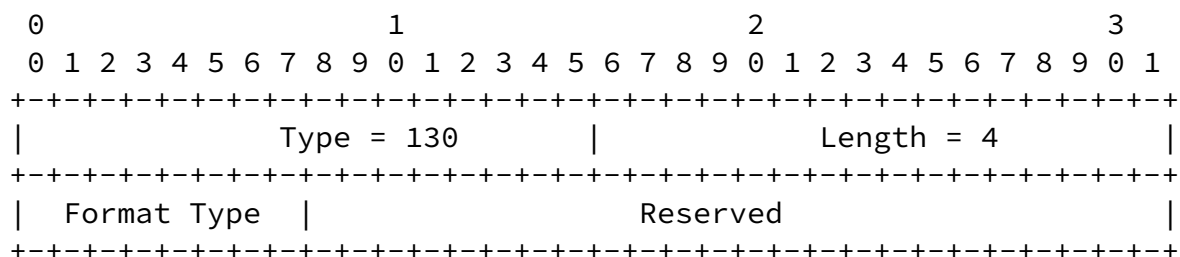
The following subsections define the SSCP parameter objects for test sessions.

4.1. Destination Test Identifier

[illegible]

high-order 12 bits are set to zero.

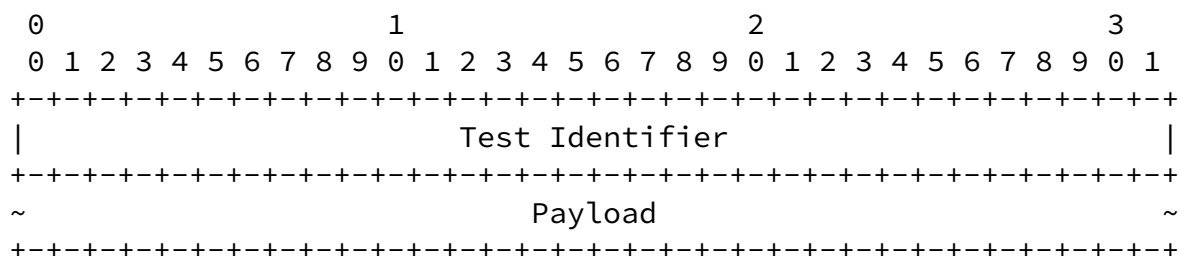
4.3. Packet Format



The Packet Format object identifies the format of test packets in this test stream. Possible values are:

Type	Meaning
0	Generic Associated Channel (G-ACh)
1	MPLS Label

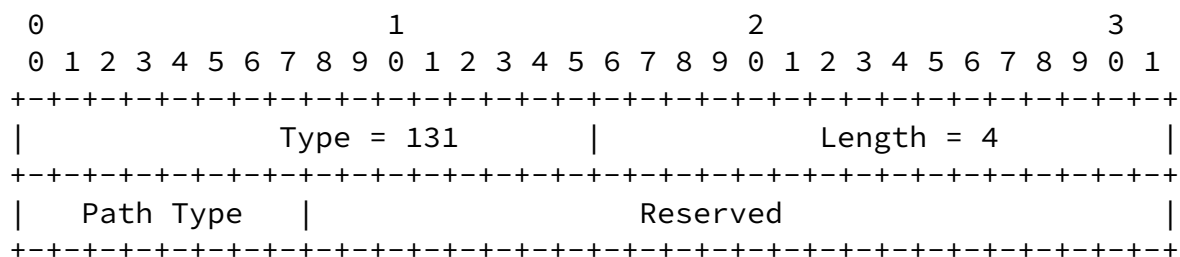
The G-ACH format indicates that test messages are sent over the G-ACH using the Channel Type allocated by IANA for test messages. In this case, test messages have the following format (after the Associated Channel Header):



In this format, the Test Identifier is set to the DTI in test messages transmitted by the test source to the test destination. In bidirectional test streams, the destination sets the Test Identifier to the STI before reflecting test messages it receives back to the source. The test message payload is set at the discretion of the test source. Support for the G-ACh format is REQUIRED.

The MPLS Label format indicates that test messages are sent as MPLS packets with a specific label at the bottom of the stack. The label values allocated by the test source and test destination are signaled via the STI and DTI objects respectively (the former only for bidirectional test streams). In this case the label serves as the test identifier; the body of the packet, i.e. the portion that follows the MPLS label stack, is considered the payload and set at the discretion of the test source.

4.4. Path Type



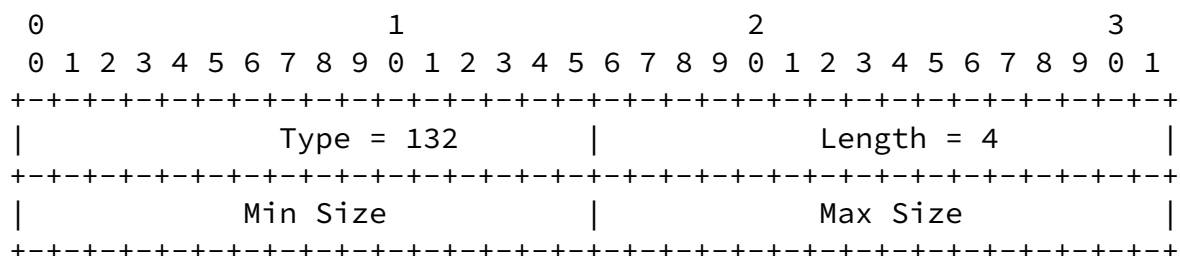
The Path Type object indicates whether the test stream is unidirectional or bidirectional. In a unidirectional stream, test packets are sent from the test source to the test destination and are then discarded. In a bidirectional stream, test packets are sent

from the source to the destination and reflected back to the source. Support for unidirectional sessions is REQUIRED.

Type Meaning

0	Unidirectional
1	Bidirectional

4.5. Payload Size Range



6. Security Considerations

This document describes a simple control protocol that allows two devices to negotiate a session via the MPLS Generic Associated Channel. The most important security considerations are those that apply to securing MPLS connectivity in general; these are documented in [[RFC5920](#)]. The control protocol described in this document exchanges session data in cleartext, as this information is no more sensitive than that contained in other protocol messages that are commonly sent in cleartext. The main security considerations specific to this protocol are those concerning the verification of message authenticity and integrity, and possible denial of service.

An authentication mechanism based on cryptographic message hashing is included in the protocol, enabling receivers to verify that protocol messages were generated by a trusted source and were not corrupted or otherwise modified in transit. This mechanism also affords protection against denial-of-service attempts made by unauthorized devices. Receivers, in addition, SHOULD employ sensible rate-limiting policies to guard against the possibility of intentional or accidental denial-of-service by authorized devices. For example, implementations SHOULD anticipate the effects of receiving a large number of Initiate or Query messages within a short period of time, and take appropriate precautions to avoid resource exhaustion in such scenarios.

7. IANA Considerations

This document makes the following requests of IANA:

- o Allocation of Associated Channel Types
- o Creation of MPLS Simple Session Control Protocol TLV Registry
- o Creation of MPLS Simple Session Control Protocol Session Type Registry

7.1. Allocation of Associated Channel Types

IANA is requested to allocate an entry in the Pseudowire Associated Channel Types registry [[RFC5586](#)] for the MPLS Simple Session Control

Protocol, as follows:

Value	Description	TLV Follows	Reference
(TBD)	MPLS Simple Session Control Protocol	No	(this draft)

IANA is also requested to allocate an entry in the same registry for MPLS test messages, as follows:

Value	Description	TLV Follows	Reference
(TBD)	MPLS Test Message	No	(this draft)

[7.2.](#) Creation of MPLS Simple Session Control Protocol TLV Registry

IANA is requested to create a new registry, "MPLS Simple Session Control Protocol TLVs", with fields and initial allocations as follows:

Type	Application Name	Description	Reference
1	Simple Session Control Protocol	Source Address	(this draft)
2	Simple Session Control Protocol	Authentication	(this draft)
3	Simple Session Control Protocol	Session Type	(this draft)
4	Simple Session Control Protocol	Hold Time	(this draft)
5	Simple Session Control Protocol	Error Information	(this draft)
128	Test Session Control	Destination Test Identifier	(this draft)
129	Test Session Control	Source Test Identifier	(this draft)
130	Test Session Control	Packet Format	(this draft)
131	Test Session Control	Path Type	(this draft)
132	Test Session Control	Payload Size Range	(this draft)
133	Test Session Control	Maximum Transmission Rate	(this draft)

[7.3.](#) Creation of MPLS Simple Session Control Protocol Session Type Registry

IANA is requested to create a new registry, "MPLS Simple Session Control Protocol Session Types", with fields and initial allocations as follows:

Session Type	Description	Reference
1	Test Session Control (this draft)	

[8.](#) Normative References

- [I-D.ietf-mpls-gach-adv]
Frost, D., Bryant, S., and M. Bocci, "MPLS Generic Associated Channel (G-ACh) Advertisement Protocol", [draft-ietf-mpls-gach-adv-02](#) (work in progress), May 2012.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", [RFC 3032](#), January 2001.
- [RFC5586] Bocci, M., Vigoureux, M., and S. Bryant, "MPLS Generic Associated Channel", [RFC 5586](#), June 2009.
- [RFC5920] Fang, L., "Security Framework for MPLS and GMPLS Networks", [RFC 5920](#), July 2010.
- [RFC6374] Frost, D. and S. Bryant, "Packet Loss and Delay Measurement for MPLS Networks", [RFC 6374](#), September 2011.

Authors' Addresses

Dan Frost
Cisco Systems

Email: danfrost@cisco.com

Stewart Bryant
Cisco Systems

Email: stbryant@cisco.com

