MPLS Working Group
Internet Draft
Expiration Date: July 1999

L. Andersson, A. Fredette, B. Jamoussi
Nortel Networks

R. Callon
IronBridge Networks

P. Doolan
Ennovate Networks

N. Feldman
IBM Corp

E. Gray
Lucent Technologies

J. Halpern
Newbridge Networks

J. Heinanen
Telia Finland

T. E. Kilty
Northchurch Communications

A. G. Malis
Ascend Communications, Inc.

M. Girish
SBC Technology Resources, Inc.

K. Sundell
Ericsson

P. Vaananen
Nokia Telecommunications

T. Worster
General DataComm, Inc.

L. Wu, R. Dantu
Alcatel

January 1998

Constraint-Based LSP Setup using LDP

draft-ietf-mpls-cr-ldp-00.txt

Status of this Memo

Abstract

Label Distribution Protocol (LDP) is defined in [LDP] for distribution of labels inside one MPLS domain. One of the most important services that may be offered using MPLS in general and LDP in particular is support for constraint-based routing of traffic across the routed network. Constraint-based routing offers the opportunity to extend the information used to setup paths beyond what is available for the routing protocol. For instance, an LSP can be setup based on an explicit route constraint, a Service Class (SC) constraint, or both. Constraint-based routing (CR) and Traffic Engineering requirements have been proposed by [FRAME], [ARCH] and [TER]. These requirements may be met by extending LDP for support of constraint-based routed label switched paths (CRLSPs). Other uses exist for CRLSPs as well ([VPN1] and [VPN2]).

This draft specifies mechanisms and TLVs for support of CRLSPs using LDP. The Explicit Route object and procedures are extracted from [ER].

1. Introduction

The need for constraint-based routing (CR) in MPLS has been explored elsewhere [ARCH], [FRAME], and [TER]. Explicit routing is a subset of the more general constraint-based routing function. At the MPLS WG

meeting held during the Washington IETF there was consensus that LDP should support explicit routing of LSPs with provision for indication of associated (forwarding) priority. In the Chicago meeting, the decision was made that support for explicit path setup in LDP will be moved to a separate document. This document provides that support. We propose an end-to-end setup mechanism of a constraint-based routed LSP (CRLSP) initiated by the ingress LSR. We also specify mechanisms to provide means for reservation of resources for the explicitly routed LSP.

We introduce TLVs and procedures that provide support for:

- Strict and Loose Explicit Routing
- Specification of Service Class
- Specification of Traffic Parameters
- Route Pinning
- CRLSP bumping though setup/holding priority
- Handling Failures

2. CRLSP Overview

CRLSP over LDP Specification is designed with several goals in mind:

1. Meet the requirements outlined in [TER] for performing traffic engineering and provide a solid foundation for performing more general constrain-based routing.

2. Build on already specified functionality that meets the requirements whenever possible. Hence, this specifications is based on [LDP] and the Explicit Route object and procedures defined in [ER].

3. Keep the solution simple and tractable.

In this document, support for unidirectional point-to-point CRLSPs is specified. Support for point-to-multipoint, multipoint-to-point, is for further study (FFS).

Support for explicitly routed LSPs in this specification depends on the following minimal LDP behaviors as specified in [LDP]:

- Basic and/or Extended Discovery Mechanisms.

- Use the Label Request Message defined in [LDP] in downstream on demand label advertisement mode with ordered control.

- Use the Label Mapping Message defined in [LDP] in downstream on demand mode with ordered control.

- Use the Notification Message defined in [LDP].

- Use the Withdraw and Release Messages defined in [LDP].

- Loop detection (in the case of loosely routed segments of a CRLSP) mechanisms.

In addition, the following functionality is added to what's defined in [LDP]:

- The Label Request Message used to setup a CRLSP includes a CR-TLV based on the path vector defined in [ER] and specified in Section 4 of this document.

- An LSR implicitly infers ordered control from the existence of a CR-TLV in the Label Request Message. This means that the LSR can still be configured for independent control for LSPs established as a result of dynamic routing. However, when a Label Request Message includes a CR TLV, then ordered control is used to setup the CRLSP. Note that this is also true for the loosely routed parts of a CRLSP.

- Traffic Parameters TLVs may optionally be carried in the Label Request Message to specify the CRLSP traffic characteristics.

- New status codes are defined to handle error notification for failure of established paths specified in the CR-TLV.

Examples of CRLSP establishment are given in Appendix A to illustrate how the mechanisms described in this draft work.

3. Required Messages and TLVs

Any Messages, TLVs, and procedures not defined explicitly in this

document are defined in the [LDP] Specification. The following subsections are meant as a cross reference to the [LDP] document and indication of additional functionality beyond what's defined in [LDP] where necessary.

3.1 Label Request Message

The Label Request Message is as defined in 3.5.8 of [LDP] with the following modifications (required only if the CR-TLV is included in the Label Request Message):

- Only a single FEC-TLV may be included in the Label Request Message.

- The Optional Parameters TLV includes the definition of the Constraint-based TLV specified in Section 4 and the Traffic Parameters TLV specified in Section 5.

- The Procedures to handle the Label Request are augmented by the procedures for processing of the CR-TLV as defined in Section 4.

- The Procedures to handle Service Classes are defined in Section 5.

3.2 Label Mapping Message

The Label Mapping Message is as defined in 3.5.7 of [LDP] with the following modifications:

- Only a single Label-TLV may be included in the Label Mapping Message.

- The FEC-Label Mapping TLV does not include any of the optional TLVs.

- The Label Mapping Message Procedures are limited to downstream on demand ordered control mode of mapping.

A Mapping message is transmitted by a downstream LSR to an upstream LSR under one of the following conditions:

1. The LSR is the egress end of the CRLSP and an upstream mapping

has been requested.

2. The LSR received a mapping from its downstream next hop LSR for an CRLSP for which an upstream request is still pending.

3.3. Notification Message

The Notification message is as defined in Section 3.5.1 of [LDP] and the Status TLV encoding is as defined in Section 3.4.7 of [LDP].

Establishment of an Explicitly Routed LSP may fail for a variety of reasons. All such failures are considered advisory conditions and they are signaled by the Notification Message.

Notification messages carry Status TLVs to specify events being signaled. New status codes are defined in Section 4.8.3 to signal error notifications associated with the establishment of a CRLSP and the processing of the CR-TLV.

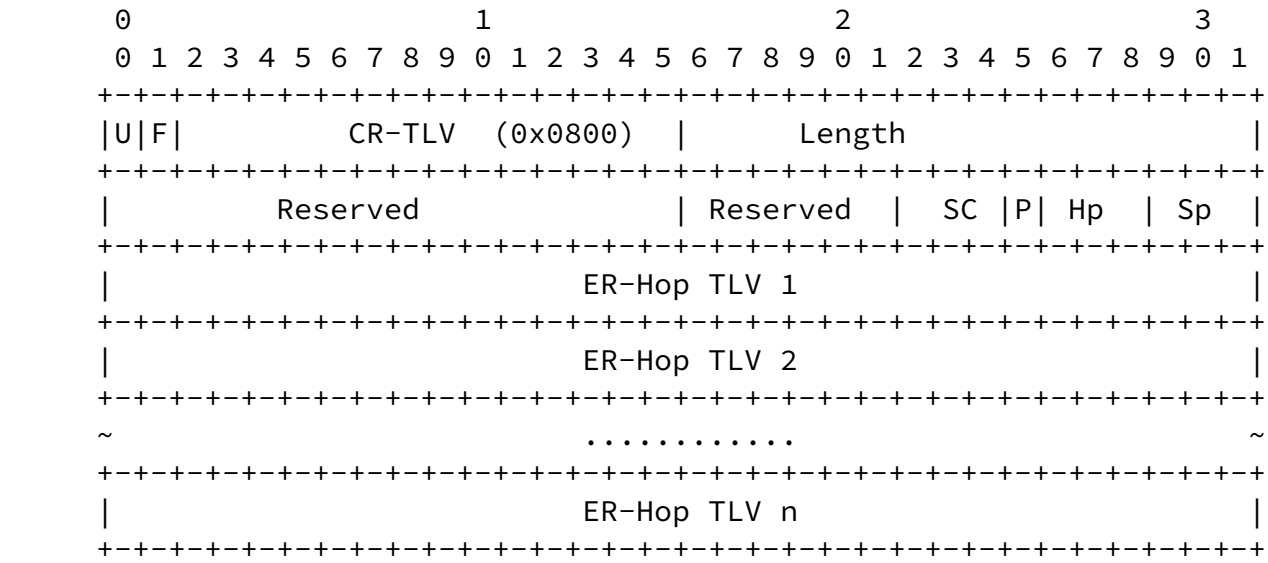
4. Constraint-based Routing TLV

Label Request Messages defined in [LDP] optionally carry the Constraint-based Routing TLV (CR-TLV) based on the path vector defined in [ER] and described in this section of the specification. The inclusion of the CR TLV in the Label Request Message indicates the path to be taken in the network even if normal routing indicates otherwise.

The format of the CR-TLV is described below.

4.1 CR-TLV

The CR-TLV is an object that specifies the path to be taken by the LSP being established. In addition, the CR-TLV may also include the the Service Class (SC) constraints associated with the LSP, a setup and a holding priority used for path bumping, and an LSP pinning request flag. Reserved bits in the CR-TLV allow for the specification of other LSP attributes in the future. If the reserved bits are exhausted, additional TLVs may be specified to allow for the indication of other LSP attributes during the CRLSP setup.



U bit

Unknown TLV bit. Upon receipt of an unknown TLV, if clear (=0), a notification must be returned to the message originator and the entire message must be ignored; if set (=1), the unknown TLV is silently ignored and the rest of the message is processed as if the unknown TLV did not exist.

F bit

Forward unknown TLV bit. This bit only applies when the U bit is set and the LDP message containing the unknown TLV is to be forwarded. If clear (=0), the unknown TLV is not forwarded with the containing message; if set (=1), the unknown TLV is forwarded with the containing message.

Type

A two byte field carrying the value of the CR-TLV type which is 0x800.

Length

Specifies the length of the value field in bytes.

Reserved

This field is reserved. It must be set to zero on transmission and must be ignored on receipt. We expect to use these fields for carrying information that support other constrain-based routing information.

P bit

When set indicates that the loosely routed segments must remain pinned-down. CRLSP must be rerouted only when adjacency is lost along the segment. When not set, it indicates that the loose segment is not pinned down and must be changed to match the underlying hop-by-hop path.

SC

The SC Field is used to specify the Service Class of the CRLSP. This field allows for the definition of up to 8 different Service Classes. Currently, Three Service Classes are defined: Best Effort (0), Throughput Sensitive (1), and Delay Sensitive (2) Service Classes. These SCs are further defined in Section 5.

Sp

A SetupPriority of value zero (0) is the priority assigned to the most important path. It is referred to as the highest priority. Four (4) is the priority for the least important path. The higher the setup priority, the more paths CR-LDP can bump to set up the path. The default value is 2. Values 5, 6, and 7 are reserved.

Hp

A HoldingPriority of value zero (0) is the priority assigned to the most important path. It is referred to as the highest priority. Four (4) is the priority for the least important path. The higher the holding priority, the less likely it is for CR-LDP to reallocate its bandwidth to a new path. The default value is 2. Values 5, 6, and 7 are reserved.

4.1.1 Setup and holding priorities

CR-LDP signals the resources required by a path on each hop of the route. If a route with sufficient resources can not be found, existing paths may be rerouted to reallocate resources to the new path. This is the process of bumping paths. Setup and holding priorities are used to rank existing paths (holding priority) and the new path (setup priority) to determine if the new path can bump an existing path.

The setupPriority of a new CRLSP and the holdingPriority attributes of the existing CRLSP are used to specify these priorities. The higher the holding priority, the less likely it is for CR-LDP to reallocate its bandwidth to a new path. Similarly, the higher the setup priority, the more paths CR-LDP can bump to set up the path.

The setup and holding priority values range from zero (0) to four (4). The value zero (0) is the priority assigned to the most important path. It is referred to as the highest priority. Four (4) is the priority for the least important path. The default values for

both setup and holding priority should be 2. By setting the default value of both setup and holding priorities at the middle of the range, all connections are initially treated the same. However, when network operators see a need for the use of path bumping, the values of setup and holding priorities can be gracefully adjusted up or down from the middle of the range.

An existing path can be bumped if and only if the setupPriority of the new path is numerically less than the holdingPriority of the existing path.

To illustrate the use of the setup and holding priority, consider a network which supports two service types (e.g., video and data services). The video traffic is given a low setup priority because new video paths can use an alternate public network if the primary network cannot accommodate the new path. However, the video traffic is given a high holding priority since it is undesirable for the path to be rerouted during an active LSP. For data traffic, high setup and holding priorities are desirable since data paths cannot be established on an alternate network.

The setup and holding priorities can be different to allow setup at one priority and holding at an independent priority. This would allow some calls not to invoke bumping and not to be bumped at the same time.

The setupPriority of a CRLSP should not be higher (numerically less) than its holdingPriority since it might bump an LSP and be bumped by next "equivalent" request.

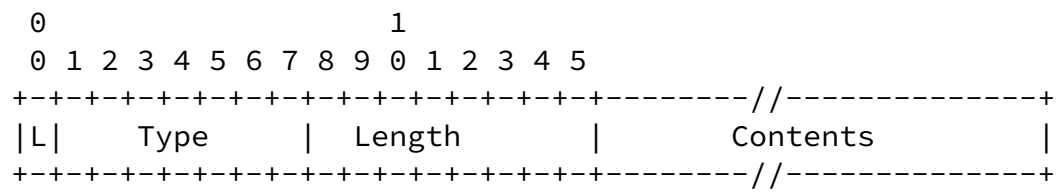
Bumping by default only happens as a last resort when there are no

routes available for a given path.

During the instantiation of a path that must bump other paths, lower holding priority paths are bumped before higher priority paths. The decision as to which of the available paths are bumped at each intermediate node by the new path is arbitrary.

4.2 ER-Hop TLV

The contents of a constraint-based route TLV are a series of variable length ER-Hop TLVs. Each ER-Hop TLV has the form:



L

The L bit is an attribute of the ER-Hop. The L bit is set if the ER-Hop represents a loose hop in the explicit route. If the bit is not set, the ER-Hop represents a strict hop in the explicit route.

Type

A seven-bit field indicating the type of contents of the ER-Hop. Currently defined values are:

Value	Type
0	Reserved
1	IPv4 prefix
2	IPv6 prefix
32	Autonomous system number

Length

The Length field contains the total length of the ER-Hop in bytes. It

includes the L bit, Type and Length fields. The length must always be a multiple of 4, and at least 4.

Contents

A variable length field containing the node or abstract node that is the consecutive nodes that make up the explicit routed LSP.

4.3 Applicability

The CR-TLV in this version of the specification is intended for unicast only. CRLSPs for multicast are FFS.

4.4 Semantics of the CR-TLV

Like any other LSP an CRLSP is a path through a network. The difference is that while other paths are setup solely based on information in routing tables or from a management system, the constraint-based route is calculated at one point at the edge of network based on criteria, including but not limited to routing information. The intention is that this functionality shall give desired special characteristics to the LSP in order to better support the traffic sent over the LSP. The reason for setting up CRLSPs, might be that one wants to assign certain bandwidth or other Service Class characteristics to the LSP, or that one wants to make sure that alternative routes use physically separate paths through the network.

A CRLSP is represented in a Label Request Message as a list of nodes or groups of nodes along the constraint-based route. When the CRLSP is established, all or a subset of the nodes in a group may be

traversed by the LSP. Certain operations to be performed along the path can also be encoded in the constraint-based route.

The capability to specify, in addition to specified nodes, groups of nodes, of which a subset will be traversed by the CRLSP, allows the system a significant amount of local flexibility in fulfilling a request for a constraint-based route. This allows the generator of the constraint-based route to have some degree of imperfect information about the details of the path.

The constraint-based route is encoded as a series of ER-Hops

contained in a constraint-based route TLV. Each ER-Hop may identify a group of nodes in the constraint-based route. A constraint-based route is then a path including all of the identified groups of nodes.

To simplify the discussion, we call each group of nodes an abstract node. Thus, we can also say that a constraint-based route is a path including all of the abstract nodes, with the specified operations occurring along that path.

4.5 Strict and Loose ER-Hops

The L bit in the ER-Hop is a one-bit attribute. If the L bit is set, then the value of the attribute is "loose." Otherwise, the value of the attribute is "strict." For brevity, we say that if the value of the ER-Hop attribute is loose then it is a "loose ER-Hop." Otherwise, it's a "strict ER-Hop." Further, we say that the abstract node of a strict or loose ER-Hop is a strict or a loose node, respectively. Loose and strict nodes are always interpreted relative to their prior abstract nodes.

The path between a strict node and its prior node MUST include only network nodes from the strict node and its prior abstract node.

The path between a loose node and its prior node MAY include other network nodes which are not part of the strict node or its prior abstract node.

4.6 Loops

While the constraint-based route TLV is of finite length, the existence of loose nodes implies that it is possible to construct forwarding loops during transients in the underlying routing protocol. This may be detected by the originator of the constraint-based route through the use a path vector object as defined in [LDP].

4.7 ER-Hop semantics

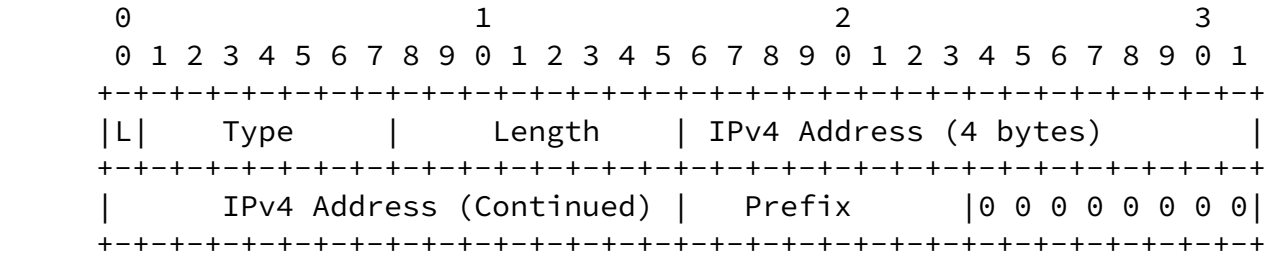
4.7.1. ER-Hop 1: The IPv4 prefix

The contents of an IPv4 prefix ER-Hop are a 4 byte IPv4 address, 1

byte of prefix length, and 1 byte of padding. The abstract node

represented by this ER-Hop is the set of nodes which have an IP address which lies within this prefix. Note that a prefix length of 32 indicates a single IPv4 node.

The length of the IPv4 prefix ER-Hop is 8 bytes. The contents of the 1 byte of padding must be zero on transmission and must not be checked on receipt.



Type

IPv4 Address 0x01

Length

A one byte field indicating the total length of the TLV in bytes. It includes the L-bit, the Type, Length, the IP Address, and the Prefix fields. The length is always 8 bytes.

IP Address

A four byte field indicating the IP Address.

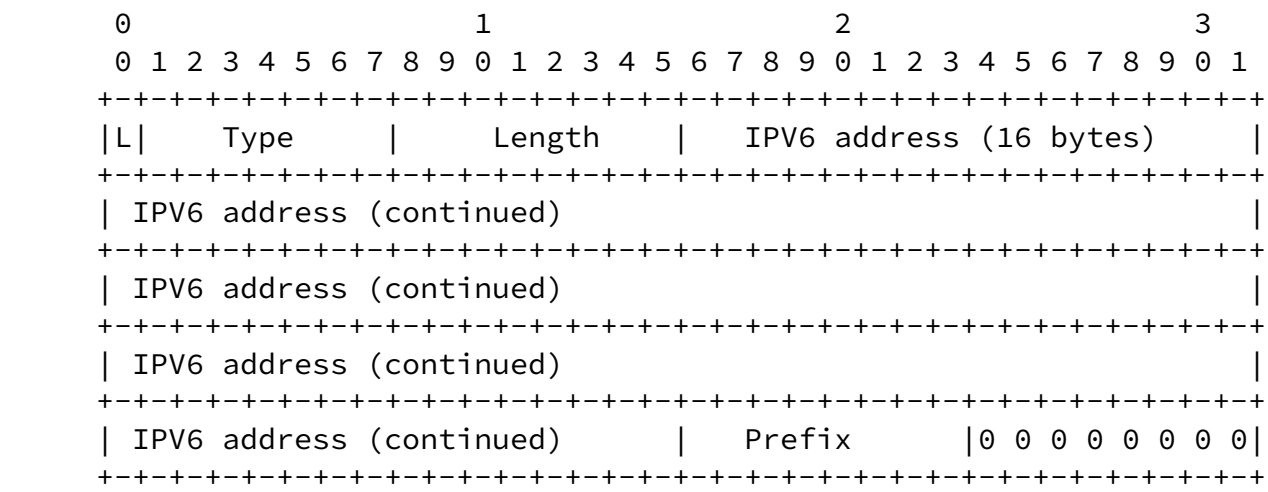
Prefix Length

1-32

Padding

Zero on transmission. Ignored on receipt.

4.7.2. ER-Hop 2: The IPv6 address



Type

0x02 IPv6 address

Length

The Length contains the total length of the ER-Hop TLV in bytes, including the Type and Length fields. The Length is always 20.

IPv6 address

A 128-bit unicast host address.

Prefix Length

1-128

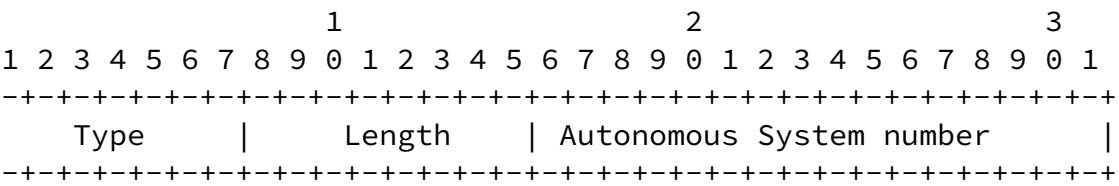
Padding

Zero on transmission. Ignored on receipt.

4.7.3. ER-Hop 32: The autonomous system number

The contents of an autonomous system (AS) number ER-Hop are a 2 byte autonomous system number. The abstract node represented by this ER-Hop is the set of nodes belonging to the autonomous system.

The length of the AS number ER-Hop is 4 bytes.



Type

AS Number 0x20

Length

A one byte field indicating the total length of the TLV in bytes. It includes the L-bit, the Type, and Length, and the AS number fields. The length is always 4 bytes.

AS number

A two byte field indicating the AS number.

4.8. Processing of the Constraint-Based Route TLV

4.8.1. Selection of the next hop

A Label Request message containing a constraint-based route TLV must determine the next hop for this path. Selection of this next hop may involve a selection from a set of possible alternatives. The mechanism for making a selection from this set is implementation dependent and is outside of the scope of this specification. Selection of particular paths is also outside of the scope of this specification, but it is assumed that each node will make a best effort attempt to determine a loop-free path. Note that such best efforts may be overridden by local policy.

To determine the next hop for the path, a node performs the following steps:

1) The node receiving the Label Request message must first evaluate the first ER-Hop. If the L bit is not set in the first ER-Hop and if the node is not part of the abstract node described

by the first ER-Hop, it has received the message in error, and should return a "Bad initial ER-Hop" error. If the L bit is set and the local node is not part of the abstract node described by the first ER-Hop, the node selects a next hop that is along the path to the abstract node described by the first ER-Hop. If there is no first ER-Hop, the message is also in error and the system should return a "Bad Constraint-Based Routing TLV" error.

2) If there is no second ER-Hop, this indicates the end of the constraint-based route. The constraint-based route TLV should be removed from the Label Request message. This node may or may not be the end of the LSP. Processing continues with section 4.8.2, where a new constraint-based route TLV may be added to the Label Request message.

3) If the node is also a part of the abstract node described by the second ER-Hop, then the node deletes the first ER-Hop and continues processing with step 2, above. Note that this makes the second ER-Hop into the first ER-Hop of the next iteration.

4) The node determines if it is topologically adjacent to the abstract node described by the second ER-Hop. If so, the node selects a particular next hop which is a member of the abstract node. The node then deletes the first ER-Hop and continues processing with section 4.8.2.

5) Next, the node selects a next hop within the abstract node of the first ER-Hop that is along the path to the abstract node of the second ER-Hop. If no such path exists then there are two cases:

5a) If the second ER-Hop is a strict ER-Hop, then there is an error and the node should return a "Bad strict node" error.

5b) Otherwise, if the second ER-Hop is a loose ER-Hop, then the node selects any next hop that is along the path to the next abstract node. If no path exists, then there is an error, and the node should return a "Bad loose node" error.

6) Finally, the node replaces the first ER-Hop with any ER-Hop that denotes an abstract node containing the next hop. This is necessary so that when the constraint-based route is received by

the next hop, it will be accepted.

7) Progress the Label Request Message to the next hop.

4.8.2. Adding ER-Hops to the constraint-based route TLV

After selecting a next hop, the node may alter the constraint-based route in the following ways.

If, as part of executing the algorithm in section 4.8.1, the constraint-based route TLV is removed, the node may add a new constraint-based route TLV.

Otherwise, if the node is a member of the abstract node for the first ER-Hop, then a series of ER-Hops may be inserted before the first ER-Hop or may replace the first ER-Hop. Each ER-Hop in this series must denote an abstract node that is a subset of the current abstract node.

Alternately, if the first ER-Hop is a loose ER-Hop, an arbitrary series of ER-Hops may be inserted prior to the first ER-Hop.

4.8.3. Error subcodes

In the processing described above, certain errors need to be reported as part of the Notification message. This section defines the status codes for the errors described above.

Status Code	Type
Bad Constraint-Based Routing TLV Error	0x04000001
Bad Strict Node Error	0x04000002
Bad Loose Node Error	0x04000003
Bad Initial ER-Hop Error	0x04000004
Resource Unavailable	0x04000005
Service Class Unavailable	0x04000006
Traffic Parameters Unavailable	0x04000007

5.0 CRLSP Service Classes and Traffic Parameters

The following sections describe the CRLSP Service Classes (SCs), and their associated traffic parameters.

The CRLSP Service Class is signaled in the SC Field of the CR-TLV defined in Section 4.1.

Three Service Classes are currently supported by CR-LDP:

Service Class		Value
Best Effort	(BE)	0x0
Throughput Sensitive	(TS)	0x1
Delay Sensitive	(DS)	0x2

These service classes are specified in the following sections.

5.1 Best Effort (BE)

The request of the BE SC implies that there are no expected service guarantees from the network. The service provided by the network is the familiar best effort service.

The Peak Date Rate (PDR) is the only traffic parameter that may be specified with the BE SC. The specification of the PDR allows the network to perform traffic shaping and policing functions.

5.2 Throughput Sensitive (TS)

In the service model for the Throughput Sensitive SC, the network commits to deliver with high probability user datagrams at a rate of at least CDR (Committed Data Rate). The user may transmit at a rate higher than CDR but datagrams in excess of CDR would have a lower probability of being delivered. If the user sends at a rate of CDR or lower the network commits to deliver with high probability all the user datagrams.

The TS SC has an associated tolerance to the burstiness of arriving

user datagrams. This tolerance is defined by the traffic parameter Committed Burst Tolerance (CBT).

Ideally, a TS CRLSP request carries with it a rich set of three traffic parameters (PDR, CDR, and CBT) that accurately describe its traffic characteristics. This allows the network to perform resource reservation, traffic shaping, and traffic policing.

However, for the sake of simplicity of the service definition, the CDR is the only parameter that MUST always be specified for a TS CRLSP. A peak data rate parameter (PDR) and a CBT are optional traffic parameters for the TS SC.

The network should make every effort to preserve ordering of the delivered datagrams of a TS CRLSP.

Network traffic that requires a low packet loss ratio at a given CDR but is not particularly sensitive to delay and jitter (e.g., network control traffic) is suited to the TS SC. The selection of the TS SC is used to signal to the various nodes along the path that the queuing and scheduling mechanisms used to handle the CRLSP should provide a low packet loss ratio.

5.3 Delay Sensitive (DS)

In the service model for the Delay Sensitive SC, the network commits to deliver with high probability user datagrams at a rate of CDR (Committed Data Rate) with minimum delay and delay variation. The user MUST transmit data at a rate of CDR or lower in order to be eligible for DS service. Datagrams in excess of CDR may be discarded by the network. If the user sends at a rate of CDR or lower the network commits to deliver with high probability all user datagrams with low delay and delay variation. If the user sends at a rate higher than CDR the network does not provide any guarantees on the excess traffic.

The Delay Sensitive SC has an associated tolerance to the burstiness of arriving user datagrams. This tolerance is defined by the traffic parameter Committed Burst Tolerance (CBT).

Ideally, a DS CRLSP request carries with it a rich set of three traffic parameters (PDR, CDR, and CBT) that accurately describe its traffic characteristics. This allows the network to perform resource reservation, traffic shaping and policing.

However, for the sake of simplicity of the service definition, the CDR is the only parameter that MUST always be specified for a DS CRLSP. A peak data rate parameter (PDR) and a CBT are optional traffic parameters for the DS SC.

The network should make every effort to preserve ordering of the

delivered datagrams of a DS CRLSP.

Network traffic that requires a low delay and delay variation at a given CDR (e.g., voice traffic) is suited to the DS SC. The selection of the DS SC is used to signal to the various nodes along the path that the queuing and scheduling mechanisms used to handle the CRLSP should provide low delay and delay variation.

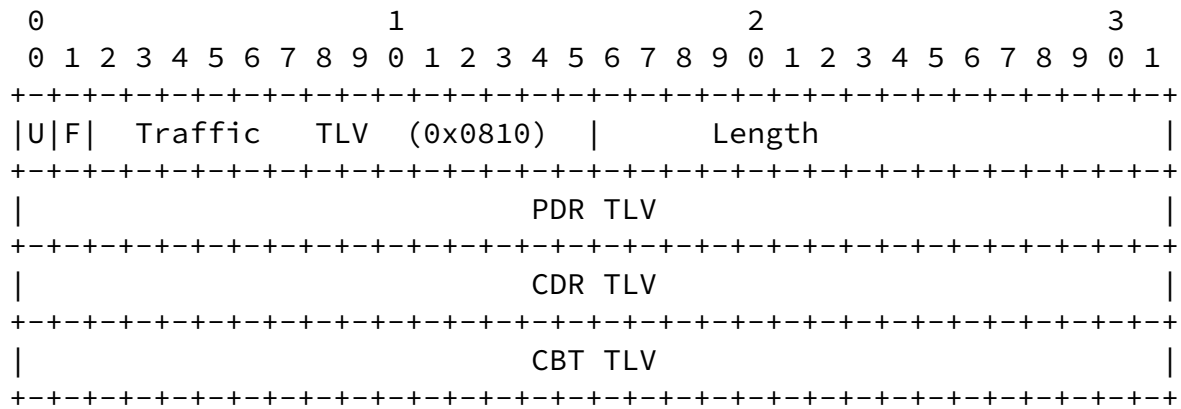
5.4 Traffic Parameters

The CRLSP traffic parameters are defined in this section.

The traffic parameters CDR, CBT and PDR are defined in terms of a TOKEN_BUCKET_TSPEC as specified in [RFC2215]. The following mapping of parameters in the TOKEN_BUCKET_TSPEC is used:

Token rate,	r = CDR
Bucket depth,	b = CBT
Peak traffic rate,	p = PDR
Minimum policed unit,	m = 1
Maximum packet size,	M = MTU

The Traffic Parameters TLV is used to signal the traffic characteristics of the CRLSP. These traffic parameters are used to perform functions such as resource reservation, Shaping, and Policing. See [SIN] for more details. The encoding for the Traffic Parameters TLV is:

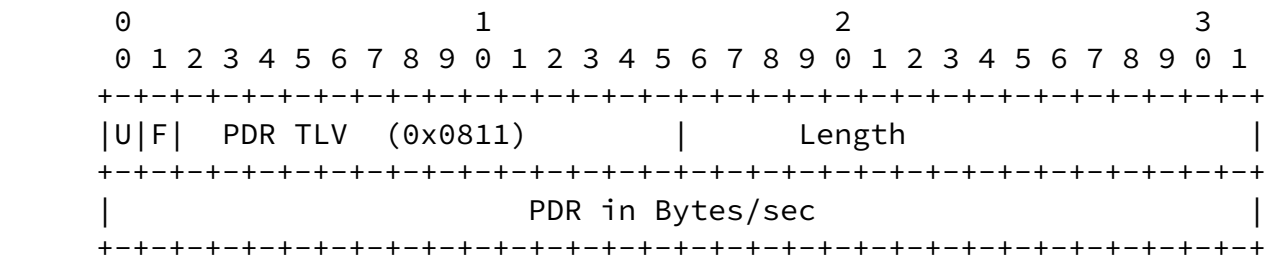


5.4.1 Peak data rate (PDR) TLV

The value of traffic parameter PDR is given as a positive integer in

bytes per second. Zero is not a valid value of PDR.

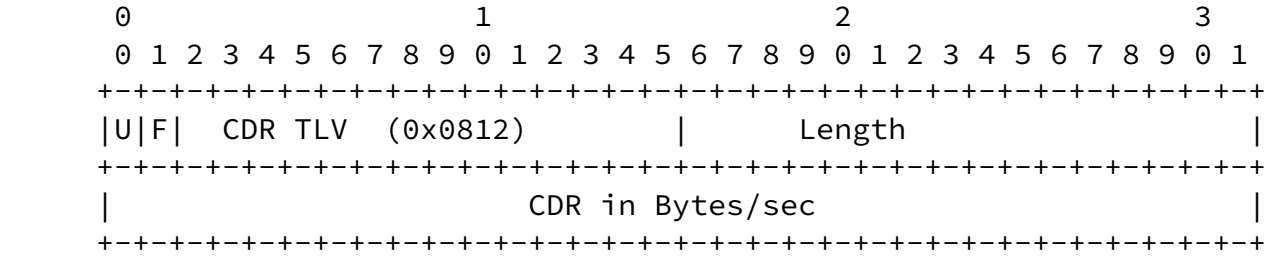
The user may specify the value of PDR depending the SC of the CRLSP. Specifying the PDR allows the network to use traffic management functions such as shaping.



5.4.2. Committed Data Rate (CDR)

The value of traffic parameter CDR is given as a positive integer in bytes per second. Zero is not a valid value of CDR.

The user may provide a requested value of CDR in the CRLSP request depending on the SC of the CRLSP.

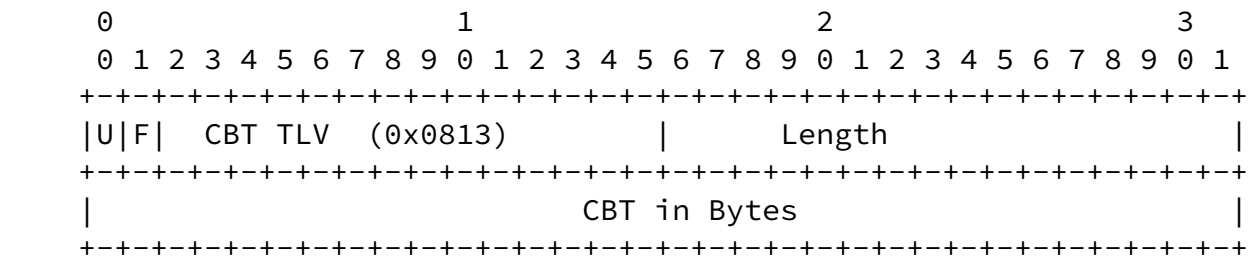


5.4.3. Committed Burst Tolerance (CBT)

The value of traffic parameter CBT is given in bytes. Zero is not a valid value of CBT.

The requested value of CBT MUST be no smaller than the MTU of the originating interface.

The user may provide a requested value of CBT in the CRLSP request. If the user chooses not to specify a requested value of CBT and the network is policing the traffic, then any excess traffic will be dropped by the network.



6. Open Issues

This section captures the issues that need further study.

1) Review the FSM described in Appendix B and extend it by the CR-TLV processing defined in Sections 4.8.1 and 4.8.2.

2) Consider if all three traffic parameters have to be signaled at all times and if the network should supply default values for the missing parameters.

3) Consider the following extensions to the CR-TLV:

3.1) Changing the 'P' bit to "next hop flag" and making it a 2-bit wide field with the following values:

- 00 "local repair", which means if it belongs to a loosely routed segment, and the LSR detects a next hop change, the LSR will try to establish a new LSP from this point on and switch it over to the new LSP when it is setup.

- 01 "global repair", which means when the LSR detects a next hop change, the LSR will tear down the LSP, the ingress LSR will try to reestablish another LSP through the new path.

- 10 "pinned", which means that the loosely routed segments must remain pinned down.

- 11 Reserved.

3.2) Adding one more field "LSPID" before ER-Hop TLV. LSPID can be used to identify a network wide unique CRLSP.

- The first 4 bytes carrying the ingress LSR IP address

- The second 4 bytes carrying the unique ID value assigned by the ingress LSR.

4) Consider the following extension to the ER-Hop TLV:

For Type field, add one more type, LSPID, which means the current CRLSP will go through another CRLSP which is identified with this LSPID value:

Value	Type
4	LSPID

Extend processing the LSPID ER-Hop as follows: If the type of ER-Hop is LSPID, and the other end of this CRLSP is not part of the constraint-based route TLV, add it to the constraint-based TLV with L bit turned off.

5) Consider traffic parameter negotiation and the ability to change the traffic parameters associated with an already established path

without tearing the old path down.

7. Security

No security issues are discussed in this version of the draft.

8. Acknowledgments

The messages used to signal the CRLSP setup are based on the work done by the [LDP] team. The Explicit Route object and procedures used in this specification are based on [ER].

The authors would also like to acknowledge the careful review and

comments of Osama Aboul-Magd, Ken Hayward, Greg Wright, Geetha Brown, Brian Williams, Peter Ashwood-smith, Paul Beaubien, Matthew Yuen, Liam Casey, and Ankur Anand.

9. References

[FRAME] Callon et al, "Framework for Multiprotocol Label Switching", work in progress (draft-ietf-mpls-framework-02), November 1997.

[ARCH] Rosen et al, "Multiprotocol Label Switching Architecture", work in progress (draft-ietf-mpls-arch-02), July 1998.

[LDP] Andersson et al, "Label Distribution Protocol Specification" work in progress (draft-ietf-mpls-ldp-02.txt), November 1998.

[ER] Guerin et al, "Setting up Reservations on Explicit Paths using RSVP", work in progress (draft-guerin-expl-path-rsvp-01.txt, November 1997.

[TER] Awduche et al, "Requirements for Traffic Engineering Over MPLS", work in progress (draft-awduche-mpls-traffic-eng-00), April 1998.

[VPN1] Heinanen et al, "MPLS Mappings of Generic VPN Mechanisms", work in progress (draft-heinanen-generic-vpn-mpls-00), August 1998.

[VPN2] Jamieson et al, "MPLS VPN Architecture" work in progress (draft-jamieson-mpls-vpn-00), August 1998.

[RFC2215] S. Shenker and J. Wroclawski, General Characterization Parameters for Integrated Service Network Elements, RFC 2215, Sep 1997.

[SIN] B. Jamoussi, N. Feldman, and L. Andersson, "MPLS Ships in the Night with ATM", (draft-jamoussi-mpls-sin-00.txt), August 1998.

10. Author Information

Loa Andersson

Director Bay Architecture Lab, EMEA
Kungsgatan 34, PO Box 1788
111 97 Stockholm, Sweden
phone: +46 8 441 78 34
mobile +46 70 522 78 34
e-mail: loa_andersson@baynetworks.com

Ross Callon
IronBridge Networks
55 Hayden Avenue,
Lexington, MA 02173
Phone: +1-781-402-8017
Email: rcallon@ironbridgenetworks.com

Ram Dantu
Alcatel USA Inc.
IP Competence Center
1201 E. Campbell Road.,446-315
Richadson, TX USA., 75081-2206
Phone: 972 996 2938
Fax: 972 996 5902
Email: ram.dantu@aud.alcatel.com

Paul Doolan
Ennovate Networks
330 Codman Hill Rd
Marlborough MA 01719
Phone: 978-263-2002
email: pdoolan@ennovatenetworks.com

Nancy Feldman
IBM Corp.
17 Skyline Drive
Hawthorne NY 10532
Phone: 914-784-3254
email: nkf@us.ibm.com

Andre Fredette
Nortel Networks
3 Federal Street
Billerica, MA 01821
email: fredette@baynetworks.com

Eric Gray
Lucent Technologies, Inc
1600 Osgood St.
North Andover, MA 01847
email: ewgray@lucent.com

Joel M. Halpern
Newbridge Networks Inc.
593 Herndon Parkway
Herndon, VA 20170
email: jhalpern@newbridge.com
phone: 1-703-736-5954
fax: 1-703-736-5959

Juha Heinanen
Telia Finland, Inc.
Myyrmaentie 2
01600 VANTAA
Finland
Tel: +358 303 944 808
Email: jh@telia.fi

Bilel Jamoussi
Nortel Networks
P O Box 3511 Station C
Ottawa, ON K1Y 4H7
Canada
phone: +1 613 765-4814
email: jamoussi@NortelNetworks.com

Timothy E. Kilty
Northchurch Communications
5 Corporate Drive,
Andover, MA 018110
phone: 978 691-4656
Email: tkilty@northc.com

Andrew G. Malis
Ascend Communications, Inc.
1 Robbins Road
Westford, MA 01886
phone: 978 952-7414
fax: 978 392-2074
Email: malis@ascend.com

Muckai K Girish
SBC Technology Resources, Inc.
4698 Willow Road
Pleasanton, CA 94588

Phone: (925) 598-1263
Fax: (925) 598-1321
Email: mgirish@tri.sbc.com

Kenneth Sundell
Ericsson
SE-126 25 Stockholm
Sweden

email: kenneth.sundell@etx.ericsson.se

Pasi Vaananen
Nokia Telecommunications
3 Burlington Woods Drive, Suite 250
Burlington, MA 01803
Phone: +1-781-238-4981
Email: pasi.vaananen@ntc.nokia.com

Tom Worster
General DataComm, Inc.
5 Mount Royal Ave.
Marlboro MA 01752
Email: tom.worster@gdc.com

Liwen Wu
Alcatel U.S.A
44983 Knoll Square
Ashburn, Va. 20147
USA
Phone: (703) 724-2619
FAX: (703) 724-2005
Inet: liwen.wu@adn.alcatel.com

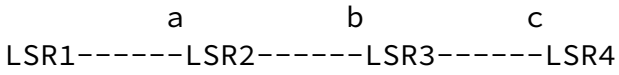
Appendix A: CRLSP Establishment Examples

A.1 Strict Constraint-Based Route Example

This appendix provides an example for the setup of a strictly routed CRLSP. In this example, each abstract node is represented by a specific node.

The sample network used here is a four node network with two edge

LSRs and two core LSRs as follows:



LSR1 generates a Label Request Message as described in Section 3.1 of this draft and sends it to LSR2. This message includes the CR-TLV.

The CR-TLV is composed by a vector of three ER-Hop TLVs <a, b, c>. The ER-Hop TLVs used in this example are of type 0x01 (IPv4 prefix) with a prefix length of 32. Hence, each ER-Hop TLV identifies a specific node as opposed to a group of nodes.

At LSR2, the following processing of the CR-TLV per Section 4.8.1 of this draft takes place:

1) The first hop <a> is part of the abstract node LSR2. Therefore, the first step passes the test. Go to step 2.

2) There is a second ER-Hop, <b>. Go to step 3.

3) LSR2 is not part of the abstract node described by the second ER-Hop <b>. Go to Step 4.

4) LSR2 determines that it is topologically adjacent to the abstract node described by the second ER-Hop <b>. LSR2 selects a next hop (LSR3) which is the abstract node. LSR2 deletes the first ER-Hop <a> from the CR-TLV which now becomes <b , c>. Go to Section 4.8.2.

At LSR2, the following processing of Section 4.8.2 takes place:

Executing algorithm 4.8.1 did not result in the removal of the CR-TLV.

Also, LSR2 is not a member of the abstract node described by the first ER-Hop <b>.

Finally, the first ER-Hop <b> is a strict hop.

Therefore, processing section 4.8.2 does not result in the insertion of new ER-Hops. The selection of the next hop has been

already done is step 4 of Section 4.8.1 and the processing of the CR-TLV is completed at LSR2. In this case, the Label Request Message including the CR-TLV <b, c> is progressed by LSR2 to LSR3.

At LSR3, a similar processing to the CR-TLV takes place except that the incoming CR-TLV = <b, c> and the outgoing CR-TLV is <c>.

At LSR4, the following processing of section 4.8.1 takes place:

1) The first hop <c> is part of the abstract node LSR4. Therefore, the first step passes the test. Go to step 2.

2) There is no second ER-Hop, this indicates the end of the CRLSP. The CR-TLV is removed from the Label Request Message. Processing continues with Section 4.8.2.

At LSR4, the following processing of Section 4.8.2 takes place:

Executing algorithm 4.8.1 resulted in the removal of the CR-TLV. LSR4 does not add a new CR-TLV.

Therefore, processing section 4.8.2 does not result in the insertion of new ER-Hops. This indicates the end of the CRLSP and the processing of the CR-TLV is completed at LSR4.

At LSR4, processing of Section 3.2 is invoked. The first condition is satisfied (LSR4 is the egress end of the CRLSP and upstream mapping has been requested). Therefore, a Label Mapping Message is generated

by LSR4 and sent to LSR3.

At LSR3, the processing of Section 3.2 is invoked. The second condition is satisfied (LSR3 received a mapping from its downstream next hop LSR4 for a CRLSP for which an upstream request is still pending). Therefore, a Label Mapping Message is generated by LSR3 and sent to LSR2.

At LSR2, a similar processing to LSR 3 takes place and a Label Mapping Message is sent back to LSR1 which completes the end-to-end CRLSP setup.

A.2. Node Groups and Specific Nodes Example

A request at an ingress LSR to setup a CRLSP might originate from a management system or an application, the details are implementation specific.

The ingress LSR uses information provided by the management system or the application and possibly also information from the routing database to calculated the constraint-based route and to create the Label Request Message.

The Label request message carries together with other necessary information a CR-TLV defining the constraint-based routed path. In our example the list of hops in the ER-Hop TLV is supposed to contain an abstract node representing a group of nodes, an abstract node representing a specific node, another abstract node representing a group of nodes, and an abstract node representing a specific egress point.

In--{Group 1}--{Specific A}--{Group 2}--{Specific Out: B}

The CR-TLV contains four ER-Hop TLVs:

1. An ER-Hop TLV that specifies a group of LSR valid for the first abstract node representing a group of nodes (Group 1).

2. An ER-Hop TLV that indicates the specific node (Node A).

3. An ER-Hop TLV that specifies a group of LSRs valid for the second abstract node representing a group of nodes (Group 2).

4. An ER-Hop TLV that indicates the specific egress point for the CRLSP (Node B).

All the ER-Hop TLVs are strictly routed nodes.

The setup procedure for this CRLSP works as follows:

1. The ingress node sends the Label Request to a node that is a member the group of nodes indicated in the first ER-Hop TLV, following normal routing for the specific node (A).

2. The node that receives the message identifies itself as part of the group indicated in the first ER-Hop TLV, and that it is not the specific node (A) in the second. Further it realizes that the specific node (A) is not one of its next hops.

3. It keeps the ER-Hop TLVs intact and sends a Label Request Message to a node that is part of the group indicated in the first ER-Hop TLV (Group 1), following normal routing for the specific node (A).

4. The node that receives the message identifies itself as part of the group indicated in the first ER-Hop TLV, and that it is not the specific node (A) in the second ER-Hop TLV. Further it realizes that the specific node (A) is one of its next hops.

5. It removes the first ER-Hop TLVs and sends a Label Request Message to the specific node (A).

6. The specific node (A) recognizes itself in the first ER-Hop TLV. Removes the specific ER-Hop TLV.

7. It sends a Label Request message to a node that is a member of the group (Group 2) indicated in the ER-Hop TLV.

8. The node that receives the message identifies itself as part of the group indicated in the first ER-Hop TLV, further it realizes that the specific egress node (B) is one of its next hops.

9. It sends a Label Request message to the specific egress node (B).

10. The specific egress node (B) recognizes itself as the egress for the CRLSP, it returns a Label Mapping Message, that will traverse the same path as the Label Request Message in the opposite direction.

Appendix B. CR-LDP Finite State Machine

In this description of the CR-LDP FSM, behavior relating to the state of LDP messages is assumed to be defined (implicitly or explicitly) in [LDP]. In particular, LDP is assumed to retain state information relating a Label Request made of a downstream neighbor to the Label Request message(s) of upstream neighbors (downstream-on-demand mode) which the (downstream) Label Request is meant to satisfy. This will be true of many potential applications of LDP, of which CR-LDP is an example. Minimally, this state should include message IDs of Label Requests (both sent and received) and the LSR(s) from which pending Label Request(s) were received.

The FSM describes CR-LDP behavior in the following operations:

- Start of CRLSP setup (in which a Label Request is sent);

- Processing the CR-TLV portion of Label Requests;

- Completion of CRLSP setup (via Label Mapping messages);

- Notification of originator when:

  - a loop is detected in a loose constraint-based route segment,

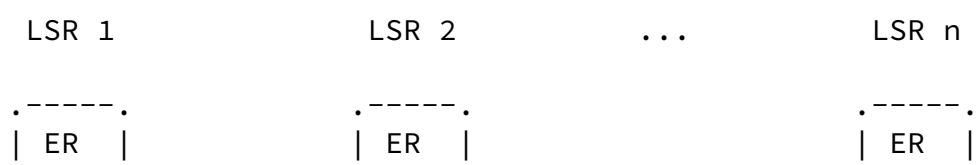
  - an ER-Hop is not reachable from a previous ER-Hop,

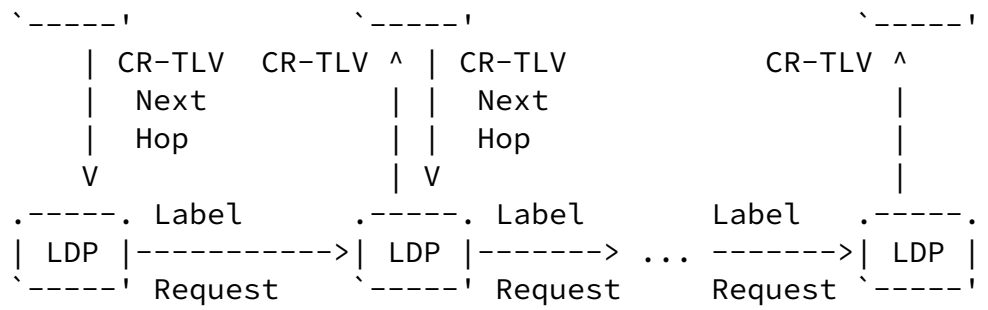
  - a next ER-Hop is strict and not directly connected to the current LSR or

  - the current LSR is strict and is not (part of the abstract node in) the first ER-Hop in the CR-TLV;

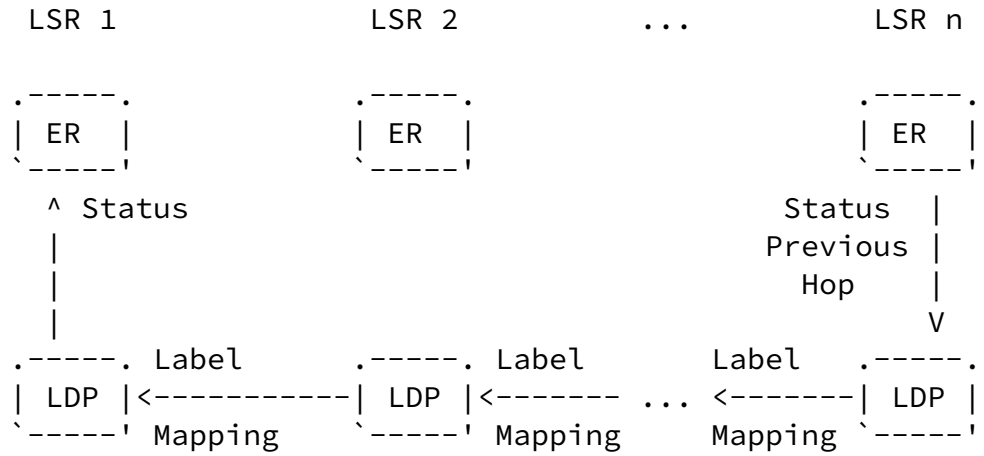
- Withdrawing a CRLSP.

For the description, the following pictorial representations may be used as an aid to understanding:

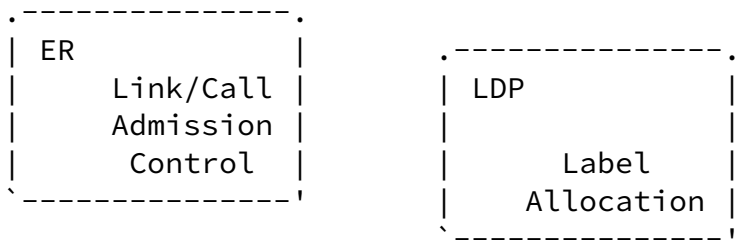




CRLSP Setup propagation



CRLSP Status propagation



Related Tasks

B.1. CR-LDP Primitives

The following sections describe the logical interactions between Constrain-based Route and LDP state machines in terms of primitives that describe the minimal information exchange required. These assume an asynchronous exchange model involving locally significant IDs that is used to tie status of a request to the initial setup and to allow LDP to relate incoming/outgoing Label Request messages. A synchronous model - possibly based on multiple threads - is also possible and would eliminate the need for IDs.

B.1.1. CR to LDP Primitives

LDP_SEND_REQ(TLV_List, To_LSR, Identifier)

TLV_List

TLVs to be sent to a neighboring LSR; includes at least an

CR-TLV and may contain additional TLVs (i.e. QoS TLVs).

To_LSR

The neighbor LSR to which a Label Request is to be sent.

Identifier

Locally significant unique identifier. May be used to associate the Label Request to be sent either with a Label Request that was previously received (e.g. - LSR 2 above) or a subsequent CRLSP Status (e.g. - LSR 1 above).

LDP_SEND_RSP(Status, Identifier)

Status

Status of a specific CRLSP Setup Request. A Status of zero indicates success; other Status values are given in Error Subcodes section. This Status is carried in Label Mapping or Notification messages to the originator of the CRLSP setup.

Identifier

Locally significant unique identifier used to associate the Label Mapping to be sent with a Label Request received (e.g. LSR n above).

B.1.2. LDP to CR Primitives

CR_RECEIVED_REQ(TLV_List, Identifier)

TLV_List

TLVs to be processed by the local constraint-based route function.

Identifier

Locally significant unique identifier used to associate the received request either with a subsequent further request or a response. For example, the identifier provided here would be used in a subsequent LDP_SEND_REQ or LDP_SEND_RSP.

CR_LSP_STATUS(Status, Identifier)

Status

Status of a specific CRLSP Setup Request. A Status of zero indicates success; other Status values are given in section Error Subcodes. This Status originated at the remote LSR

which either completed the CRLSP setup or determined that CRLSP setup could not be done.

Identifier

Locally significant unique identifier used to associate the received response with the original request. For example, this identifier would be the same as was used in the initial LDP_SEND_REQ.

B.2. CR-LDP States

This document defines 3 states relative to any one specific CRLSP.

They are:

CR_Non_Existant - no state information exists relative to this CRLSP;

CR_In_Progress - LDP_SEND_REQ has been called in result of external input (e.g. - management);

CR_Established - a successful status has been received from an earlier setup.

These states are defined such that no additional state is required to support CRLSPs using LDP at intermediate LSRs than is already required in LDP.

B.3. CR-LDP Events

This document defines 4 events impacting any one specific CRLSP. They are:

CR_Start - a CRLSP is required based on an external stimulus (e.g. - management);

CR_Req_Received - further CRLSP setup processing is required based on CR_RECEIVED_REQ (i.e. - from an upstream LSR's CRLSP Label Request);

CR_Setup_Complete - CRLSP setup has been successfully completed based on CR_LSP_STATUS (with success status);

CR_LSP_Failure - Either a CRLSP could not be established as requested, or a setup CRLSP has dropped; based on CR_LSP_STATUS (with error status).

B.4. CR-LDP Transitions

State transitions are defined as follows:

State	Event	Action	New State

CR_Non_Existant	CR_Start	1	CR_In_Progress
CR_Non_Existant	CR_Req_Rec	2	CR_Non_Existant
CR_In_Progress	CR_Setup_Complete		CR_Established
CR_In_Progress	CR_LSP_Failure	3	CR_Non_Existant
CR_Established	CR_LSP_Failure	3	CR_Non_Existant

Actions:

1) Establish CRLSP state, create CR-TLV information, LDP_SEND_REQ.
2) Process CR-TLV (as described in "Processing of the Constraint-Based Route TLV" section) and either LDP_SEND_REQ or LDP_SEND_RSP.
3) Remove state information relative to this CRLSP (may notify management, other external source initially requiring setup).

For the purposes of this transition table, illegal transitions (not included in the table) are ignored.