          Crankback Signaling Extensions for MPLS and GMPLS RSVP-TE

Status of This Memo

   This document specifies an Internet standards track protocol for the
   Internet community, and requests discussion and suggestions for
   improvements. Please refer to the current edition of the "Internet
   Official Protocol Standards" (STD 1) for the standardization state
   and status of this protocol. Distribution of this memo is unlimited.

Abstract

   In a distributed, constraint-based routing environment, the
   information used to compute a path may be out of date. This means
   that Multiprotocol Label Switching (MPLS) and Generalized MPLS
   (GMPLS) Traffic Engineered (TE) Label Switched Path (LSP) setup
   requests may be blocked by links or nodes without sufficient
   resources. Crankback is a scheme whereby setup failure information
   is returned from the point of failure to allow new setup attempts to
   be made avoiding the blocked resources. Crankback can also be
   applied to LSP recovery to indicate the location of the failed link
   or node.

   This document specifies crankback signaling extensions for use in
   MPLS signaling using RSVP-TE as defined in "RSVP-TE: Extensions to
   RSVP for LSP Tunnels", RFC 3209, and GMPLS signaling as defined in
   "Generalized Multi-Protocol Label Switching (GMPLS) Signaling
   Functional Description", RFC 3473. These extensions mean that the
   LSP setup request can be retried on an alternate path that detours
   around blocked links or nodes. This offers significant improvements

   in the successful setup and recovery ratios for LSPs, especially in
   situations where a large number of setup requests are triggered at
   the same time.

Table of Contents

Section A : Problem Statement

1. Introduction and Framework

1.1. Background

   RSVP-TE (RSVP Extensions for LSP Tunnels) [RFC3209] can be used for
   establishing explicitly routed LSPs in an MPLS network. Using RSVP-
   TE, resources can also be reserved along a path to guarantee and/or
   control QoS for traffic carried on the LSP. To designate an explicit
   path that satisfies Quality of Service (QoS) guarantees, it is
   necessary to discern the resources available to each link or node in
   the network. For the collection of such resource information,
   routing protocols, such as OSPF and Intermediate System to
   Intermediate System (IS-IS), can be extended to distribute additional
   state information [RFC2702].

   Explicit paths can be computed based on the distributed information
   at the LSR (ingress) initiating an LSP and signaled as Explicit
   Routes during LSP establishment. Explicit Routes may contain 'loose
   hops' and 'abstract nodes' that convey routing through a collection
   of nodes. This mechanism may be used to devolve parts of the path
   computation to intermediate nodes such as area border LSRs.

   In a distributed routing environment, however, the resource
   information used to compute a constraint-based path may be out of
   date. This means that a setup request may be blocked, for example,
   because a link or node along the selected path has insufficient
   resources.

   In RSVP-TE, a blocked LSP setup may result in a PathErr message sent
   to the ingress, or a ResvErr sent to the egress (terminator). These
   messages may result in the LSP setup being abandoned. In Generalized
   MPLS [RFC3473] the Notify message may additionally be used to
   expedite notification of failures of existing LSPs to ingress and
   egress LSRs, or to a specific "repair point" -- an LSR responsible
   for performing protection or restoration.

   These existing mechanisms provide a certain amount of information
   about the path of the failed LSP.

   Generalized MPLS [RFC3471] and [RFC3473] extends MPLS into networks
   that manage Layer 2, TDM and lambda resources as well as packet
   resources. Thus, crankback routing is also useful in GMPLS networks.

   In a network without wavelength converters, setup requests are likely
   to be blocked more often than in a conventional MPLS environment
   because the same wavelength must be allocated at each Optical Cross-

Connect on an end-to-end explicit path. This makes crankback routing
all the more important in certain GMPLS networks.

1.2. Control Plane and Data Plane Separation

Throughout this document, the processes and techniques are described
as though the control plane and data plane elements that comprise a
Label Switching Router (LSR) coreside and are related in a one-to-one
manner. This is for the convenience of documentation only.

It should be noted that GMPLS LSRs may be decomposed such that the
control plane components are not physically collocated. Furthermore,
one presence in the control plane may control more than one LSR in
the data plane. These points have several consequences with respect
to this document:

o The nodes, links, and resources that are reported as errors, are
   data plane entities.

o The nodes, areas, and Autonomous Systems (ASs) that report that
   they have attempted re-routing are control plane entities.

o Where a single control plane entity is responsible for more than
   one data plane LSR, crankback signaling may be implicit in just
   the same way as LSP establishment signaling may be.

The above points may be considered self-evident, but are stated here
for absolute clarity.

The stylistic convenience of referring to both the control plane
element responsible for a single LSR and the data plane component of
that LSR simply as "the LSR" should not be taken to mean that this
document is applicable only to a collocated one-to-one relationship.
Furthermore, in the majority of cases, the control plane and data
plane components are related in a 1:1 ratio and are usually
collocated.

1.3. Repair and Recovery

If the ingress LSR or intermediate area border LSR knows the location
of the blocked link or node, it can designate an alternate path and
then reissue the setup request. Determination of the identity of the
blocked link or node can be achieved by the mechanism known as
crankback routing [PNNI, ASH1]. In RSVP-TE, crankback signaling
requires notifying the upstream LSR of the location of the blocked
link or node. In some cases, this requires more information than is
currently available in the signaling protocols.

On the other hand, various recovery schemes for link or node failures
have been proposed in [RFC3469] and include fast re-routing. These
schemes rely on the existence of a protecting LSP to protect the
working LSP, but if both the working and protecting paths fail, it is
necessary to re-establish the LSP on an end-to-end basis, avoiding
the known failures. Similarly, fast re-routing by establishing a
recovery path on demand after failure requires computation of a new
LSP that avoids the known failures. End-to-end recovery for
alternate routing requires the location of the failed link or node.
Crankback routing schemes could be used to notify the upstream LSRs
of the location of the failure.

Furthermore, in situations where many link or node failures occur at
the same time, the difference between the distributed routing
information and the real-time network state becomes much greater than
in normal LSP setups. LSP recovery might, therefore, be performed
with inaccurate information, which is likely to cause setup blocking.
Crankback routing could improve failure recovery in these situations.

The requirement for end-to-end allocation of lambda resources in
GMPLS networks without wavelength converters means that end-to-end
recovery may be the only way to recover from LSP failures. This is
because segment protection may be much harder to achieve in networks
of photonic cross-connects where a particular lambda may already be
in use on other links: End-to-end protection offers the choice of use
of another lambda, but this choice is not available in segment
protection.

This requirement makes crankback re-routing particularly useful in a
GMPLS network, particularly in dynamic LSP re-routing cases (i.e.,
when there is no pre-establishment of the protecting LSP).

1.4. Interaction with TE Flooding Mechanisms

   GMPLS uses Interior Gateway Protocols (IGPs) (OSPF and IS-IS) to
   flood traffic engineering (TE) information that is used to construct
   a traffic engineering database (TED) which acts as a data source for
   path computation.

   Crankback signaling is not intended to supplement or replace the
   normal operation of the TE flooding mechanism, since these mechanisms
   are independent of each other. That is, information gathered from
   crankback signaling may be applied to compute an alternate path for
   the LSP for which the information was signaled, but the information
   is not intended to be used to influence the computation of the paths
   of other LSPs.

Any requirement to rapidly flood updates about resource availability
so that they may be applied as deltas to the TED and utilized in
future path computations are out of the scope of this document.

1.5. Terminology

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. Discussion: Explicit versus Implicit Re-Routing Indications

There have been problems in service provider networks when
"inferring" from indirect information that re-routing is allowed.
This document proposes the use of an explicit re-routing indication
that authorizes re-routing, and contrasts it with the inferred or
implicit re-routing indication that has previously been used.

Various existing protocol options and exchanges, including the error
values of PathErr message [RFC2205, RFC3209] and the Notify message
[RFC3473], allow an implementation to infer a situation where re-
routing can be performed. This allows for recovery from network
errors or resource contention.

However, such inference of recovery signaling is not always desirable
since it may be doomed to failure. For example, experience of using
release messages in TDM-based networks, for analogous implicit and
explicit re-routing indications purposes provides some guidance.
This background information is given in Appendix A.

It is certainly the case that with topology information distribution,
as performed with routing protocols such as OSPF, the ingress LSR
could infer the re-routing condition. However, convergence of
topology information using routing protocols is typically slower than
the expected LSP setup times. One of the reasons for crankback is to
avoid the overhead of available-link-bandwidth flooding, and to more
efficiently use local state information to direct alternate routing
to the path computation point.

[ASH1] shows how event-dependent-routing can just use crankback, and
not available-link-bandwidth flooding, to decide on the re-route path
in the network through "learning models". Reducing this flooding
reduces overhead and can lead to the ability to support much larger
AS sizes.

Therefore, the use of alternate routing should be based on an
explicit indication, and it is best to know the following information
separately:

   - where blockage/congestion occurred.

   - whether alternate routing "should" be attempted.

3. Required Operation

   Section 1 identifies some of the circumstances under which crankback
   may be useful. Crankback routing is performed as described in the
   following procedures, when an LSP setup request is blocked along the
   path or when an existing LSP fails.

3.1. Resource Failure or Unavailability

   When an LSP setup request is blocked due to unavailable resources, an
   error message response with the location identifier of the blockage
   should be returned to the LSR initiating the LSP setup (ingress LSR),
   the area border LSR, the AS border LSR, or some other repair point.

   This error message carries an error specification according to
   [RFC3209] -- this indicates the cause of the error and the node/link
   on which the error occurred. Crankback operation may require further
   information as detailed in Sections 3.2.1 and 6.

   A repair point (for example, an ingress LSR) that receives crankback
   information resulting from the failure of an established LSP may
   apply local policy to govern how it attempts repair of the LSP. For
   example, it may prioritize repair attempts between multiple LSPs that
   have failed, and it may consider LSPs that have been locally repaired
   ([RFC4090]) to be less urgent candidates for end-to-end repair.
   Furthermore, there is a likelihood that other LSRs are also
   attempting LSP repair for LSPs affected by the same fault which may
   give rise to resource contention within the network, so an LSR may
   stagger its repair attempts in order to reduce the chance of resource
   contention.

3.2. Computation of an Alternate Path

   In a flat network without partitioning of the routing topology, when
   the ingress LSR receives the error message, it computes an alternate
   path around the blocked link or node to satisfy QoS guarantees using
   link state information about the network. If an alternate path is
   found, a new LSP setup request is sent over this path.

   On the other hand, in a network partitioned into areas such as with
   OSPF, the area border LSR may intercept and terminate the error
   response, and perform alternate (re-)routing within the downstream
   area.

In a third scenario, any node within an area may act as a repair
point. In this case, each LSR behaves much like an area border LSR
as described above. It can intercept and terminate the error
response and perform alternate routing. This may be particularly
useful where domains of computation are applied within the
(partitioned) network, where such domains are not coincident on the
routing partition boundaries. However if, all nodes in the network
perform re-routing it is possible to spend excessive network and CPU
resources on re-routing attempts that would be better made only at
designated re-routing nodes. This scenario is somewhat like 'MPLS
fast re-route' [RFC4090], in which any node in the MPLS domain can
establish 'local repair' LSPs upon failure notification.

3.2.1. Information Required for Re-Routing

   In order to correctly compute a route that avoids the blocking
   problem, a repair point LSR must gather as much crankback information
   as possible. Ideally, the repair node will be given the node, link,
   and reason for the failure.

   The reason for the failure may provide an important discriminator to
   help decide what action should be taken. For example, a failure that
   indicates "No Route to Destination" is likely to give rise to a new
   path computation excluding the reporting LSR, but the reason
   "Temporary Control Plane Congestion" might lead to a simple retry
   after a suitable pause.

   However, even this information may not be enough to help with re-
   computation. Consider for instance an explicit route that contains a
   non-explicit abstract node or a loose hop. In this case, the failed
   node and link are not necessarily enough to tell the repair point
   which hop in the explicit route has failed. The crankback
   information needs to indicate where, within the explicit route, the
   problem has occurred.

3.2.2. Signaling a New Route

   If the crankback information can be used to compute a new route
   avoiding the failed/blocking network resource, the route can be
   signaled as an Explicit Route.

   However, it may be that the repair point does not have sufficient
   topology information to compute an Explicit Route that is guaranteed
   to avoid the failed link or node. In this case, Route Exclusions
   [RFC4874] may be particularly helpful. To achieve this, [RFC4874]
   allows the crankback information to be presented as route exclusions
   to force avoidance of the failed node, link, or resource.

3.3. Persistence of Error Information

   The repair point LSR that computes the alternate path should store
   the location identifiers of the blockages indicated in the error
   message until the LSP is successfully established by downstream LSRs
   or until the repair point LSR abandons re-routing attempts. Since
   crankback signaling information may be returned to the same repair
   point LSR more than once while establishing a specific LSP, the
   repair point LSR SHOULD maintain a history table of all experienced
   blockages for this LSP (at least until the routing protocol updates
   the state of this information) so that the resulting path
   computation(s) can detour all blockages.

   If a second error response is received by a repair point (while it is
   performing crankback re-routing) it should update the history table
   that lists all experienced blockages, and use the entire gathered
   information when making a further re-routing attempt.

   Note that the purpose of this history table is to correlate
   information when repeated retry attempts are made by the same LSR.
   For example, suppose that an attempt is made to route from A through
   B, and B returns a failure with crankback information, an attempt may
   be made to route from A through C, and this may also fail with the
   return of crankback information. The next attempt SHOULD NOT be to
   route from A through B, and this may be achieved by use of the
   history table.

   The history table can be discarded by the signaling controller for A
   if the LSP is successfully established through A. The history table
   MAY be retained after the signaling controller for A sends an error
   upstream, however the value this provides is questionable since a
   future retry as a result of crankback re-routing should not attempt
   to route through A. If the history information is retained for a
   longer period it SHOULD be discarded after a local timeout has
   expired. This timer is required so that the repair point does not
   apply the history table to an attempt by the ingress to re-establish
   a failed LSP, but to allow the history table to be available for use
   in re-routing attempts before the ingress declares the LSP as failed.

   It is RECOMMENDED that the repair point LSR discard the history table
   using a timer no larger than the LSP retry timer configured on the
   ingress LSR. The correlation of the timers between the ingress and
   repair point LSRs is typically by manual configuration of timers
   local to each LSR, and is outside the scope of this document.

   The information in the history table is not intended to supplement
   the TED for the computation of paths of other LSPs.

3.4. Handling Re-Route Failure

   Multiple blockages (for the same LSP) may occur, and successive setup
   retry attempts may fail. Retaining error information from previous
   attempts ensures that there is no thrashing of setup attempts, and
   knowledge of the blockages increases with each attempt.

   It may be that after several retries, a given repair point is unable
   to compute a path to the destination (that is, the egress of the LSP)
   that avoids all of the blockages. In this case, it must pass an
   error indication message upstream. It is most useful to the upstream
   nodes (and in particular to the ingress LSR) that may repair points
   for the LSP setup, if the error indication message identifies all of
   the downstream blockages and also the repair point that was unable to
   compute an alternate path.

3.5. Limiting Re-Routing Attempts

   It is important to prevent endless repetition of LSP setup attempts
   using crankback routing information after error conditions are
   signaled, or during periods of high congestion. It may also be
   useful to reduce the number of retries, since failed retries will
   increase setup latency and degrade performance by increasing the
   amount of signaling processing and message exchanges within the
   network.

   The maximum number of crankback re-routing attempts that are allowed
   may be limited in a variety of ways. This document allows an LSR to
   limit the retries per LSP, and assumes that such a limit will be
   applied either as a per-node configuration for those LSRs that are
   capable of re-routing, or as a network-wide configuration value.

   When the number of retries at a particular LSR is exceeded, the LSR
   will report the failure in an upstream direction until it reaches the
   next repair point where further re-routing attempts may be attempted,
   or it reaches the ingress which may act as a repair point or declare
   the LSP as failed. It is important that the crankback information
   this is provided indicates that routing back through this node will
   not succeed; this situation is similar to that in Section 3.4.

4. Existing Protocol Support for Crankback Re-Routing

   Crankback re-routing is appropriate for use with RSVP-TE.

   1) LSP establishment may fail because of an inability to route,
      perhaps because links are down. In this case a PathErr message is
      returned to the ingress.

2) LSP establishment may fail because resources are unavailable.
   This is particularly relevant in GMPLS where explicit label
   control may be in use. Again, a PathErr message is returned to
   the ingress.

3) Resource reservation may fail during LSP establishment, as the
   Resv is processed. If resources are not available on the required
   link or at a specific node, a ResvErr message is returned to the
   egress node indicating "Admission Control failure" [RFC2205]. The
   egress is allowed to change the FLOWSPEC and try again, but in the
   event that this is not practical or not supported (particularly in
   the non-PSC context), the egress LSR may choose to take any one of
   the following actions.

   - Ignore the situation and allow recovery to happen through Path
     refresh message and refresh timeout [RFC2205].

   - Send a PathErr message towards the ingress indicating "Admission
     Control failure".

   Note that in multi-area/AS networks, the ResvErr might be
   intercepted and acted on at an area/AS border router.

4) It is also possible to make resource reservations on the forward
   path as the Path message is processed. This choice is compatible
   with LSP setup in GMPLS networks [RFC3471], [RFC3473]. In this
   case, if resources are not available, a PathErr message is
   returned to ingress indicating "Admission Control failure".

Crankback information would be useful to an upstream node (such as
the ingress) if it is supplied on a PathErr or a Notify message that
is sent upstream.

4.1. RSVP-TE

   In RSVP-TE, a failed LSP setup attempt results in a PathErr message
   returned upstream. The PathErr message carries an ERROR_SPEC object,
   which indicates the node or interface reporting the error and the
   reason for the failure.

   Crankback re-routing can be performed explicitly avoiding the node or
   interface reported.

4.2. GMPLS-RSVP-TE

   GMPLS extends the error reporting described above by allowing LSRs to
   report the interface that is in error in addition to the identity of
   the node reporting the error. This further enhances the ability of a
   re-computing node to route around the error.

   GMPLS introduces a targeted Notify message that may be used to report
   LSP failures direct to a selected node. This message carries the
   same error reporting facilities as described above. The Notify
   message may be used to expedite the propagation of error
   notifications, but in a network that offers crankback routing at
   multiple nodes there would need to be some agreement between LSRs as
   to whether PathErr or Notify provides the stimulus for crankback
   operation. This agreement is constrained by the re-routing behavior
   selection (as listed in Section 5.4). Otherwise, multiple nodes
   might attempt to repair the LSP at the same time, because:

   1) these messages can flow through different paths before reaching
      the ingress LSR, and

   2) the destination of the Notify message might not be the ingress
      LSR.

Section B : Solution

5. Control of Crankback Operation

5.1. Requesting Crankback and Controlling In-Network Re-Routing

   When a request is made to set up an LSP tunnel, the ingress LSR
   should specify whether it wants crankback information to be collected
   in the event of a failure, and whether it requests re-routing
   attempts by any or specific intermediate nodes. For this purpose, a
   Re-routing Flag field is added to the protocol setup request
   messages. The corresponding values are mutually exclusive.

   No Re-routing   The ingress node MAY attempt re-routing
                            after failure. Intermediate nodes SHOULD
                            NOT attempt re-routing after failure.
                            Nodes detecting failures MUST report an
                            error and MAY supply crankback information.
                            This is the default and backwards
                            compatible option.

   End-to-end Re-routing   The ingress node MAY attempt re-routing
                            after failure. Intermediate nodes SHOULD
                            NOT attempt re-routing after failure.

                          Nodes detecting failures MUST report an
                          error and SHOULD supply crankback
                          information.

   Boundary Re-routing   Intermediate nodes MAY attempt re-routing
                          after failure only if they are Area Border
                          Routers or AS Border Routers (ABRs/ASBRs).
                          The boundary (ABR/ASBR) can either decide
                          to forward the error message upstream to
                          the ingress LSR or try to select another
                          egress boundary LSR. Other intermediate
                          nodes SHOULD NOT attempt re-routing. Nodes
                          detecting failures MUST report an error and
                          SHOULD supply crankback information.

   Segment-based Re-routing Any node MAY attempt re-routing after it
                          receives an error report and before it
                          passes the error report further upstream.
                          Nodes detecting failures MUST report an
                          error and SHOULD supply full crankback
                          information.

5.2. Action on Detecting a Failure

   A node that detects the failure to setup an LSP or the failure of an
   established LSP SHOULD act according to the Re-routing Flag passed on
   the LSP setup request.

   If Segment-based Re-routing is allowed, or if Boundary Re-routing is
   allowed and the detecting node is an ABR or ASBR, the detecting node
   MAY immediately attempt to re-route.

   If End-to-end Re-routing is indicated, or if Segment-based or
   Boundary Re-routing is allowed and the detecting node chooses not to
   make re-routing attempts (or has exhausted all possible re-routing
   attempts), the detecting node MUST return a protocol error indication
   and SHOULD include full crankback information.

5.3. Limiting Re-Routing Attempts

   Each repair point SHOULD apply a locally configurable limit to the
   number of attempts it makes to re-route an LSP. This helps to
   prevent excessive network usage in the event of significant faults,
   and allows back-off to other repair points which may have a better
   chance of routing around the problem.

5.3.1. New Status Codes for Re-Routing

   An error code/value of "Routing Problem"/"Re-routing limit exceeded"
   (24/22) is used to identify that a node has abandoned crankback re-
   routing because it has reached a threshold for retry attempts.

   A node receiving an error response with this status code MAY also
   attempt crankback re-routing, but it is RECOMMENDED that such
   attempts be limited to the ingress LSR.

5.4. Protocol Control of Re-Routing Behavior

   The LSP_ATTRIBUTES object defined in [RFC4420] is used on Path
   messages to convey the Re-Routing Flag described in Section 4.1.
   Three bits are defined for inclusion in the LSP Attributes TLV as
   follows. The bit numbers below have been assigned by IANA.

   Bit   Name and Usage
   Number

      1   End-to-end re-routing desired.
             This flag indicates the end-to-end re-routing behavior for an
             LSP under establishment. This MAY also be used for
             specifying the behavior of end-to-end LSP recovery for
             established LSPs.

      2   Boundary re-routing desired.
             This flag indicates the boundary re-routing behavior for an
             LSP under establishment. This MAY also be used for
             specifying the segment-based LSP recovery through nested
             crankback for established LSPs. The boundary ABR/ASBR can
             either decide to forward the PathErr message upstream to an
             upstream boundary ABR/ASBR or to the ingress LSR.
             Alternatively, it can try to select another egress boundary
             LSR.

      3   Segment-based re-routing desired.
             This flag indicates the segment-based re-routing behavior for
             an LSP under establishment. This MAY also be used to specify
             the segment-based LSP recovery for established LSPs.

6. Reporting Crankback Information

6.1. Required Information

   As described above, full crankback information SHOULD indicate the
   node, link, and other resources, which have been attempted but have
   failed because of allocation issues or network failure.

The default crankback information SHOULD include the interface and
the node address.

Any address reported in such crankback information SHOULD be an
address that was distributed by the routing protocols (OSPF and IS-
IS) in their TE link state advertisements. However, some additional
information such as component link identifiers is additional to this.

6.2. Protocol Extensions

[RFC3473] defines an IF_ID ERROR_SPEC object that can be used on
PathErr, ResvErr and Notify messages to convey the information
carried in the Error Spec Object defined in [RFC3209]. Additionally,
the IF_ID ERROR_SPEC Object has the scope for carrying TLVs that
identify the link associated with the error.

The TLVs for use with this object are defined in [RFC3471], and are
listed below. They are used in two places. In the IF_ID RSVP_HOP
object they are used to identify links. In the IF_ID ERROR_SPEC
object they are used to identify the failed resource which is usually
the downstream resource from the reporting node.

```
Type Length Format      Description
-------------------------------------------------------------------
 1      8   IPv4 Addr.  IPv4                     (Interface address)
 2     20   IPv6 Addr.  IPv6                     (Interface address)
 3     12   Compound    IF_INDEX                 (Interface index)
 4     12   Compound    COMPONENT_IF_DOWNSTREAM (Component interface)
 5     12   Compound    COMPONENT_IF_UPSTREAM   (Component interface)
```

Note that TLVs 4 and 5 are obsoleted by [RFC4201] and SHOULD NOT be
used to identify component interfaces in IF_ID ERROR_SPEC objects.

In order to facilitate reporting of crankback information, the
following additional TLVs are defined.

```
  Type Length Format      Description
  -------------------------------------------------------------------
   6    var   See below   DOWNSTREAM_LABEL         (GMPLS label)
   7    var   See below   UPSTREAM_LABEL           (GMPLS label)
   8     8    See below   NODE_ID                  (TE Router ID)
   9     x    See below   OSPF_AREA                (Area ID)
  10     x    See below   ISIS_AREA                (Area ID)
  11     8    See below   AUTONOMOUS_SYSTEM        (Autonomous system)
  12    var   See below   ERO_CONTEXT              (ERO subobject)
  13    var   See below   ERO_NEXT_CONTEXT         (ERO subobjects)
  14     8    IPv4 Addr.  PREVIOUS_HOP_IPv4        (Node address)
  15    20    IPv6 Addr.  PREVIOUS_HOP_IPv6        (Node address)
  16     8    IPv4 Addr.  INCOMING_IPv4            (Interface address)
  17    20    IPv6 Addr.  INCOMING_IPv6            (Interface address)
  18    12    Compound    INCOMING_IF_INDEX        (Interface index)
  19    var   See below   INCOMING_DOWN_LABEL      (GMPLS label)
  20    var   See below   INCOMING_UP_LABEL        (GMPLS label)
  21     8    See below   REPORTING_NODE_ID        (Router ID)
  22     x    See below   REPORTING_OSPF_AREA      (Area ID)
  23     x    See below   REPORTING_ISIS_AREA      (Area ID)
  24     8    See below   REPORTING_AS             (Autonomous system)
  25    var   See below   PROPOSED_ERO             (ERO subobjects)
  26    var   See below   NODE_EXCLUSIONS          (List of nodes)
  27    var   See below   LINK_EXCLUSIONS          (List of interfaces)
```

For types 1, 2, and 3 the format of the Value field is already
defined in [RFC3471].

For types 14 and 16, the format of the Value field is the same as for
type 1.

For types 15 and 17, the format of the Value field is the same as for
type 2.

For type 18, the format of the Value field is the same as for type 3.

For types 6, 7, 19, and 20, the length field is variable and the
Value field is a label as defined in [RFC3471]. As with all uses of
labels, it is assumed that any node that can process the label
information knows the syntax and semantics of the label from the
context. Note that all TLVs are zero-padded to a multiple of four
octets so that if a label is not itself a multiple of four octets, it
must be disambiguated from the trailing zero pads by knowledge
derived from the context.

For types 8 and 21, the Value field has the format:

```
  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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                           Router ID                           |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
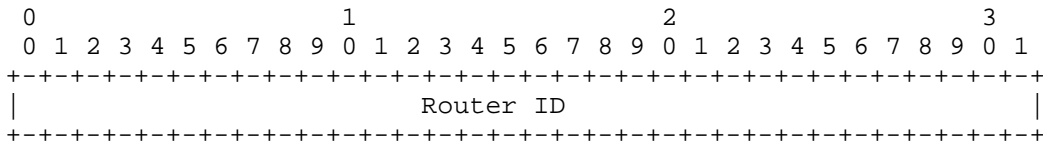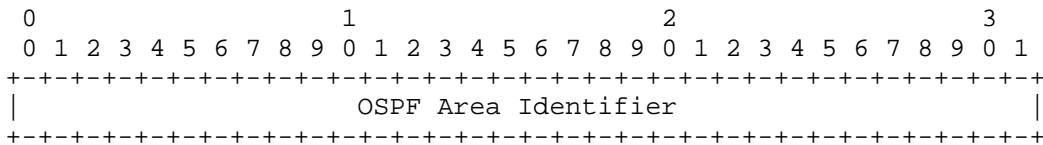
   Router ID: 32 bits

      The TE Router ID (TLV type 8) or the Router ID (TLV type 21)
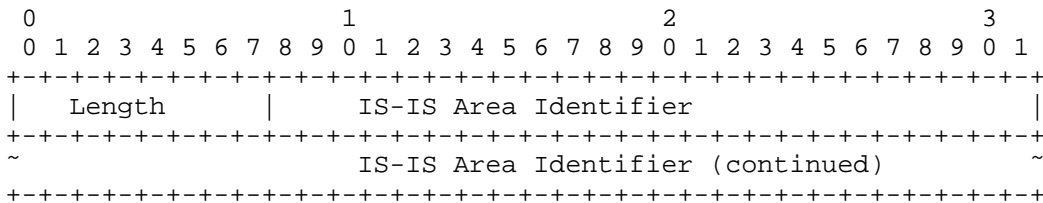   used to identify the node within the IGP.

For types 9 and 22, the Value field has the format:

```
  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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                     OSPF Area Identifier                      |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   OSPF Area Identifier

      The 4-octet area identifier for the node. This identifies the
      area where the failure has occurred.

For types 10 and 23, the Value field has the format:

```
  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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |    Length     |        IS-IS Area Identifier                  |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 ~                  IS-IS Area Identifier (continued)            ~
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
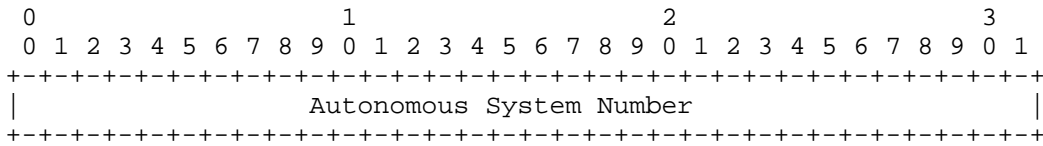
   Length

      Length of the actual (non-padded) IS-IS Area Identifier in
      octets. Valid values are from 2 to 11 inclusive.

   IS-IS Area Identifier

      The variable-length IS-IS area identifier. Padded with
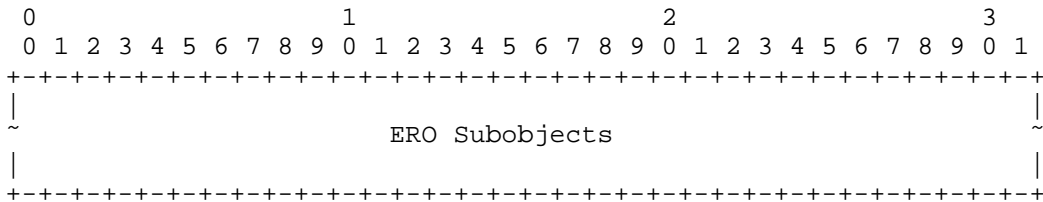   trailing zeroes to a four-octet boundary.

   For types 11 and 24, the Value field has the format:

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                    Autonomous System Number                   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

      Autonomous System Number: 32 bits

         The AS Number of the associated Autonomous System. Note that
         if 16-bit AS numbers are in use, the low order bits (16
         through 31) should be used and the high order bits (0 through
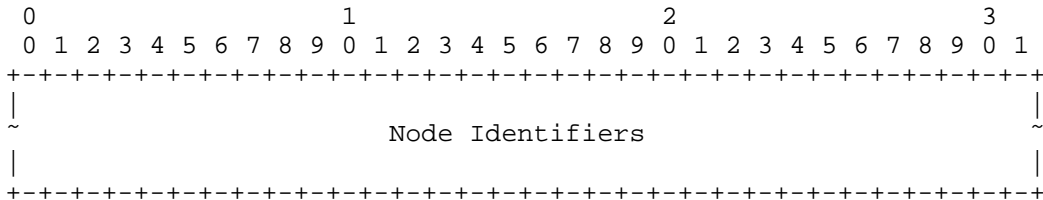   15) should be set to zero.

   For types 12, 13, and 25, the Value field has the format:

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   ~                       ERO Subobjects                          ~
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

      ERO Subobjects:

         A sequence of Explicit Route Object (ERO) subobjects. Any ERO
         subobjects are allowed whether defined in [RFC3209],
         [RFC3473], or other documents. Note that ERO subobjects
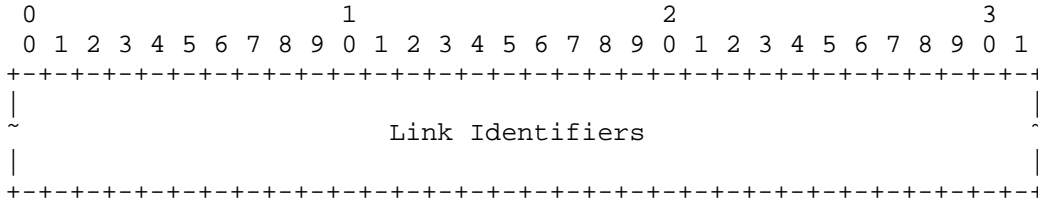   contain their own types and lengths.

   For type 26, the Value field has the format:

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   ~                       Node Identifiers                        ~
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   Node Identifiers:

      A sequence of TLVs as defined here of types 1, 2, or 8 that
      indicates downstream nodes that have already participated in
      crankback attempts and have been declared unusable for the
      current LSP setup attempt. Note that an interface identifier
      may be used to identify a node.

   For type 27, the Value field has the format:

    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
   +-+
   | |
   ~   Link Identifiers   ~
   | |
   +-+

   Link Identifiers:

      A sequence of TLVs as defined here of the same format as type
      1, 2 or 3 TLVs that indicate incoming interfaces at downstream
      nodes that have already participated in crankback attempts and
      have been declared unusable for the current LSP setup attempt.

6.3. Guidance for Use of IF_ID ERROR_SPEC TLVs

6.3.1. General Principles

   If crankback is not being used, inclusion of an IF_ID ERROR_SPEC
   object in PathErr, ResvErr, and Notify messages follows the
   processing rules defined in [RFC3473] and [RFC4201]. A sender MAY
   include additional TLVs of types 6 through 27 to report crankback
   information for informational/monitoring purposes.

   If crankback is being used, the sender of a PathErr, ResvErr, or
   Notify message MUST use the IF_ID ERROR_SPEC object and MUST include
   at least one of the TLVs in the range 1 through 3 as described in
   [RFC3473], [RFC4201], and the previous paragraph. Additional TLVs
   SHOULD also be included to report further information. The following
   section gives advice on which TLVs should be used under different
   circumstances, and which TLVs must be supported by LSRs.

   Note that all such additional TLVs are optional and MAY be omitted.
   Inclusion of the optional TLVs SHOULD be performed where doing so
   helps to facilitate error reporting and crankback. The TLVs fall
   into three categories: those that are essential to report the error,
   those that provide additional information that is or may be

fundamental to the utility of crankback, and those that provide
additional information that may be useful for crankback in some
circumstances.

Note that all LSRs MUST be prepared to receive and forward any TLV as
per [RFC3473]. This includes TLVs of type 4 or 5 as defined in
[RFC3473] and obsoleted by [RFC4201]. There is, however, no
requirement for an LSR to actively process any but the TLVs defined
in [RFC3473]. An LSR that proposes to perform crankback re-routing
SHOULD support receipt and processing of all of the fundamental
crankback TLVs, and is RECOMMENDED to support the receipt and
processing of the additional crankback TLVs.

It should be noted, however, that some assumptions about the TLVs
that will be used MAY be made based on the deployment scenarios. For
example, a router that is deployed in a single-area network does not
need to support the receipt and processing of TLV types 22 and 23.
Those TLVs might be inserted in an IF_ID ERROR_SPEC object, but would
not need to be processed by the receiver of a PathErr message.

6.3.2. Error Report TLVs

Error Report TLVs are those in the range 1 through 3. (Note that the
obsoleted TLVs 4 and 5 may be considered in this category, but SHOULD
NOT be used.)

As stated above, when crankback information is reported, the IF_ID
ERROR_SPEC object MUST be used. When the IF_ID ERROR_SPEC object is
used, at least one of the TLVs in the range 1 through 3 MUST be
present. The choice of which TLV to use will be dependent on the
circumstance of the error and device capabilities. For example, a
device that does not support IPv6 will not need the ability to create
a TLV of type 2. Note, however, that such a device MUST still be
prepared to receive and process all error report TLVs.

6.3.3. Fundamental Crankback TLVs

Many of the TLVs report the specific resource that has failed. For
example, TLV type 1 can be used to report that the setup attempt was
blocked by some form of resource failure on a specific interface
identified by the IP address supplied. TLVs in this category are 1
through 11, although TLVs 4 and 5 may be considered to be excluded
from this category by dint of having been obsoleted.

These TLVs SHOULD be supplied whenever the node detecting and
reporting the failure with crankback information has the information
available. (Note that some of these TLVs MUST be included as
described in the previous two sections.)

   The TLVs of type 8, 9, 10, and 11 MAY, however, be omitted according
   to local policy and relevance of the information.

6.3.4. Additional Crankback TLVs

   Some TLVs help to locate the fault within the context of the path of
   the LSP that was being set up. TLVs of types 12, 13, 14, and 15 help
   to set the context of the error within the scope of an explicit path
   that has loose hops or non-precise abstract nodes. The ERO context
   information is not always a requirement, but a node may notice that
   it is a member of the next hop in the ERO (such as a loose or non-
   specific abstract node) and deduce that its upstream neighbor may
   have selected the path using next hop routing. In this case,
   providing the ERO context will be useful to the upstream node that
   performs re-routing.

   Note the distinction between TLVs 12 and 13 is the distinction
   between "this is the hop I was trying to satisfy when I failed" and
   "this is the next hop I was trying to reach when I failed".

   Reporting nodes SHOULD also supply TLVs from the range 12 through 20
   as appropriate for reporting the error. The reporting nodes MAY also
   supply TLVs from the range 21 through 27.

   Note that in deciding whether a TLV in the range 12 through 20 "is
   appropriate", the reporting node should consider amongst other
   things, whether the information is pertinent to the cause of the
   failure. For example, when a cross-connection fails, it may be that
   the outgoing interface is faulted, in which case only the interface
   (for example, TLV type 1) needs to be reported, but if the problem is
   that the incoming interface cannot be connected to the outgoing
   interface because of temporary or permanent cross-connect
   limitations, the node should also include reference to the incoming
   interface (for example, TLV type 16).

   Four TLVs (21, 22, 23, and 24) allow the location of the reporting
   node to be expanded upon. These TLVs would not be included if the
   information is not of use within the local system, but might be added
   by ABRs relaying the error. Note that the Reporting Node ID (TLV 21)
   need not be included if the IP address of the reporting node as
   indicated in the ERROR_SPEC itself, is sufficient to fully identify
   the node.

   The last three TLVs (25, 26, and 27) provide additional information
   for recomputation points. The reporting node (or a node forwarding
   the error) MAY make suggestions about how the error could have been
   avoided, for example, by supplying a partial ERO that would cause the
   LSP to be successfully set up if it were used. As the error

propagates back upstream and as crankback routing is attempted and
fails, it is beneficial to collect lists of failed nodes and links so
that they will not be included in further computations performed at
upstream nodes. These lists may also be factored into route
exclusions [RFC4874].

Note that there is no ordering requirement on any of the TLVs within
the IF_ID Error Spec, and no implication should be drawn from the
ordering of the TLVs in a received IF_ID Error Spec.

The decision of precisely which TLV types a reporting node includes
is dependent on the specific capabilities of the node, and is outside
the scope of this document.

6.3.5. Grouping TLVs by Failure Location

Further guidance as to the inclusion of crankback TLVs can be given
by grouping the TLVs according to the location of the failure and the
context within which it is reported. For example, a TLV that reports
an area identifier would only need to be included as the crankback
error report transits an area boundary.

```
Resource Failure
        6         DOWNSTREAM_LABEL
        7         UPSTREAM_LABEL
Interface Failures
        1         IPv4
        2         IPv6
        3         IF_INDEX
        4         COMPONENT_IF_DOWNSTREAM (obsoleted)
        5         COMPONENT_IF_UPSTREAM   (obsoleted)
       12         ERO_CONTEXT
       13         ERO_NEXT_CONTEXT
       14         PREVIOUS_HOP_IPv4
       15         PREVIOUS_HOP_IPv6
       16         INCOMING_IPv4
       17         INCOMING_IPv6
       18         INCOMING_IF_INDEX
       19         INCOMING_DOWN_LABEL
       20         INCOMING_UP_LABEL
Node Failures
        8         NODE_ID
       21         REPORTING_NODE_ID
Area Failures
        9         OSPF_AREA
       10         ISIS_AREA
       22         REPORTING_OSPF_AREA
       23         REPORTING_ISIS_AREA
       25         PROPOSED_ERO
       26         NODE_EXCLUSIONS
       27         LINK_EXCLUSIONS
AS Failures
       11         AUTONOMOUS_SYSTEM
       24         REPORTING_AS
```

   Although discussion of aggregation of crankback information is out of
   the scope of this document, it should be noted that this topic is
   closely aligned to the information presented here. Aggregation is
   discussed further in Section 6.4.5.

6.3.6. Alternate Path Identification

   No new object is used to distinguish between Path/Resv messages for
   an alternate LSP. Thus, the alternate LSP uses the same SESSION and
   SENDER_TEMPLATE/FILTER_SPEC objects as the ones used for the initial
   LSP under re-routing.

6.4. Action on Receiving Crankback Information

6.4.1. Re-Route Attempts

   As described in Section 2, a node receiving crankback information in
   a PathErr must first check to see whether it is allowed to perform
   re-routing. This is indicated by the Re-routing Flags in the
   LSP_ATTRIBUTES object during an LSP setup request.

   If a node is not allowed to perform re-routing it should forward the
   PathErr message, or if it is the ingress report the LSP as having
   failed.

   If re-routing is allowed, the node should attempt to compute a path
   to the destination using the original (received) explicit path and
   excluding the failed/blocked node/link. The new path should be added
   to an LSP setup request as an explicit route and signaled.

   LSRs performing crankback re-routing should store all received
   crankback information for an LSP until the LSP is successfully
   established or until the node abandons its attempts to re-route the
   LSP. On the next crankback re-routing path computation attempt, the
   LSR should exclude all the failed nodes, links and resources reported
   from previous attempts.

   It is an implementation decision whether the crankback information is
   discarded immediately upon a successful LSP establishment or retained
   for a period in case the LSP fails.

6.4.2. Location Identifiers of Blocked Links or Nodes

   In order to compute an alternate path by crankback re-routing, it is
   necessary to identify the blocked links or nodes and their locations.
   The common identifier of each link or node in an MPLS network should
   be specified. Both protocol-independent and protocol-dependent
   identifiers may be specified. Although a general identifier that is
   independent of other protocols is preferable, there are a couple of
   restrictions on its use as described in the following subsection.

   In link state protocols such as OSPF and IS-IS, each link and node in
   a network can be uniquely identified, for example, by the context of
   a TE Router ID and the Link ID. If the topology and resource
   information obtained by OSPF advertisements is used to compute a
   constraint-based path, the location of a blockage can be represented
   by such identifiers.

Note that when the routing-protocol-specific link identifiers are
used, the Re-routing Flag on the LSP setup request must have been set
to show support for boundary or segment-based re-routing.

In this document, we specify routing protocol specific link and node
identifiers for OSPFv2, OSPFv3, and IS-IS for IPv4 and IPv6. These
identifiers may only be used if segment-based re-routing is
supported, as indicated by the Routing Behavior flag on the LSP setup
request.

6.4.3. Locating Errors within Loose or Abstract Nodes

The explicit route on the original LSP setup request may contain a
loose or an Abstract Node. In these cases, the crankback information
may refer to links or nodes that were not in the original explicit
route.

In order to compute a new path, the repair point may need to identify
the pair of hops (or nodes) in the explicit route between which the
error/blockage occurred.

To assist this, the crankback information reports the top two hops of
the explicit route as received at the reporting node. The first hop
will likely identify the node or the link, the second hop will
identify a 'next' hop from the original explicit route.

6.4.4. When Re-Routing Fails

When a node cannot or chooses not to perform crankback re-routing, it
must forward the PathErr message further upstream.

However, when a node was responsible for expanding or replacing the
explicit route as the LSP setup was processed, it MUST update the
crankback information with regard to the explicit route that it
received. Only if this is done will the upstream nodes stand a
chance of successfully routing around the problem.

6.4.5. Aggregation of Crankback Information

When a setup blocking error or an error in an established LSP occurs
and crankback information is sent in an error notification message,
an upstream node may choose to attempt crankback re-routing. If that
node's attempts at re-routing fail, the node will accumulate a set of
failure information. When the node gives up, it MUST propagate the
failure message further upstream and include crankback information
when it does so.

   Including a full list of all failures that have occurred due to
   multiple crankback failures by multiple repair point LSRs downstream
   could lead to too much signaled information using the protocol
   extensions described in this document. A compression mechanism for
   such information is available using TLVs 26 and 27. These TLVs allow
   for a more concise accumulation of failure information as crankback
   failures are propagated upstream.

   Aggregation may involve reporting all links from a node as unusable
   by flagging the node as unusable, flagging an ABR as unusable when
   there is no downstream path available, or including a TLV of type 9
   which results in the exclusion of the entire area, and so on. The
   precise details of how aggregation of crankback information is
   performed are beyond the scope of this document.

6.5. Notification of Errors

6.5.1. ResvErr Processing

   As described above, the resource allocation failure for RSVP-TE may
   occur on the reverse path when the Resv message is being processed.
   In this case, it is still useful to return the received crankback
   information to the ingress LSR. However, when the egress LSR
   receives the ResvErr message, per [RFC2205] it still has the option
   of re-issuing the Resv with different resource requirements (although
   not on an alternate path).

   When a ResvErr carrying crankback information is received at an
   egress LSR, the egress LSR MAY ignore this object and perform the
   same actions that it would perform for any other ResvErr. However,
   if the egress LSR supports the crankback extensions defined in this
   document, and after all local recovery procedures have failed, it
   SHOULD generate a PathErr message carrying the crankback information
   and send it to the ingress LSR.

   If a ResvErr reports on more than one FILTER_SPEC (because the Resv
   carried more than one FILTER_SPEC) then only one set of crankback
   information should be present in the ResvErr and it should apply to
   all FILTER_SPEC carried. In this case, it may be necessary per
   [RFC2205] to generate more than one PathErr.

6.5.2. Notify Message Processing

   [RFC3473] defines the Notify message to enhance error reporting in
   RSVP-TE networks. This message is not intended to replace the
   PathErr and ResvErr messages. The Notify message is sent to
   addresses requested on the Path and Resv messages. These addresses
   could (but need not) identify the ingress and egress LSRs,
   respectively.

   When a network error occurs, such as the failure of link hardware,
   the LSRs that detect the error MAY send Notify messages to the
   requested addresses. The type of error that causes a Notify message
   to be sent is an implementation detail.

   In the event of a failure, an LSR that supports [RFC3473] and the
   crankback extensions defined in this document MAY choose to send a
   Notify message carrying crankback information. This would ensure a
   speedier report of the error to the ingress and/or egress LSRs.

6.6. Error Values

   Error values for the Error Code "Admission Control Failure" are
   defined in [RFC2205]. Error values for the error code "Routing
   Problem" are defined in [RFC3209] and [RFC3473].

   A new error value is defined for the error code "Routing Problem".
   "Re-routing limit exceeded" indicates that re-routing has failed
   because the number of crankback re-routing attempts has gone beyond
   the predetermined threshold at an individual LSR.

6.7. Backward Compatibility

   It is recognized that not all nodes in an RSVP-TE network will
   support the extensions defined in this document. It is important
   that an LSR that does not support these extensions can continue to
   process a PathErr, ResvErr, or Notify message even if it carries the
   newly defined IF_ID ERROR_SPEC information (TLVs).

   This document does not introduce any backward compatibility issues
   provided that existing implementations conform to the TLV processing
   rules defined in [RFC3471] and [RFC3473].

7. LSP Recovery Considerations

   LSP recovery is performed to recover an established LSP when a
   failure occurs along the path. In the case of LSP recovery, the
   extensions for crankback re-routing explained above can be applied
   for improving performance. This section gives an example of applying
   the above extensions to LSP recovery. The goal of this example is to
   give a general overview of how this might work, and not to give a
   detailed procedure for LSP recovery.

   Although there are several techniques for LSP recovery, this section
   explains the case of on-demand LSP recovery, which attempts to set up
   a new LSP on demand after detecting an LSP failure.

7.1. Upstream of the Fault

   When an LSR detects a fault on an adjacent downstream link or node, a
   PathErr message is sent upstream. In GMPLS, the ERROR_SPEC object
   may carry a Path_State_Remove_Flag indication. Each LSR receiving
   the message then releases the corresponding LSP. (Note that if the
   state removal indication is not present on the PathErr message, the
   ingress node MUST issue a PathTear message to cause the resources to
   be released.) If the failed LSP has to be recovered at an upstream
   LSR, the IF_ID ERROR SPEC that includes the location information of
   the failed link or node is included in the PathErr message. The
   ingress, intermediate area border LSR, or indeed any repair point
   permitted by the Re-routing Flags, that receives the PathErr message
   can terminate the message and then perform alternate routing.

   In a flat network, when the ingress LSR receives the PathErr message
   with the IF_ID ERROR_SPEC TLVs, it computes an alternate path around
   the blocked link or node satisfying the QoS guarantees. If an
   alternate path is found, a new Path message is sent over this path
   toward the egress LSR.

   In a network segmented into areas, the following procedures can be
   used. As explained in Section 5.4, the LSP recovery behavior is
   indicated in the Flags field of the LSP_ATTRIBUTES object of the Path
   message. If the Flags indicate "End-to-end re-routing", the PathErr
   message is returned all the way back to the ingress LSR, which may
   then issue a new Path message along another path, which is the same
   procedure as in the flat network case above.

   If the Flags field indicates Boundary re-routing, the ingress area
   border LSR MAY terminate the PathErr message and then perform
   alternate routing within the area for which the area border LSR is
   the ingress LSR.

If the Flags field indicates segment-based re-routing, any node MAY
apply the procedures described above for Boundary re-routing.

7.2. Downstream of the Fault

This section only applies to errors that occur after an LSP has been
established. Note that an LSR that generates a PathErr with
Path_State_Remove Flag SHOULD also send a PathTear downstream to
clean up the LSP.

A node that detects a fault and is downstream of the fault MAY send a
PathErr and/or Notify message containing an IF_ID ERROR SPEC that
includes the location information of the failed link or node, and MAY
send a PathTear to clean up the LSP at all other downstream nodes.

However, if the reservation style for the LSP is Shared Explicit (SE)
the detecting LSR MAY choose not to send a PathTear -- this leaves
the downstream LSP state in place and facilitates make-before-break
repair of the LSP re-utilizing downstream resources. Note that if
the detecting node does not send a PathTear immediately, then the
unused state will timeout according to the normal rules of [RFC2205].

At a well-known merge point, an ABR or an ASBR, a similar decision
might also be made so as to better facilitate make-before-break
repair. In this case, a received PathTear might be 'absorbed' and
not propagated further downstream for an LSP that has an SE
reservation style. Note, however, that this is a divergence from the
protocol and might severely impact normal tear-down of LSPs.

8. IANA Considerations

8.1. Error Codes

IANA maintains a registry called "RSVP Parameters" with a subregistry
called "Error Codes and Globally-Defined Error Value Sub-Codes".
This subregistry includes the RSVP-TE "Routing Problem" error code
that is defined in [RFC3209].

IANA has assigned a new error value for the "Routing Problem" error
code as follows:

    22   Re-routing limit exceeded.

8.2. IF_ID_ERROR_SPEC TLVs

   The IF_ID_ERROR_SPEC TLV type values defined in [RFC3471] are
   maintained by IANA in the "Interface_ID Types" subregistry of the
   "GMPLS Signaling Parameters" registry.

   IANA has made new assignments from this subregistry for the new TLV
   types defined in Section 6.2 of this document.

8.3. LSP_ATTRIBUTES Object

   IANA maintains an "RSVP TE Parameters" registry with an "Attributes
   Flags" subregistry. IANA has made three new allocations from this
   registry as listed in Section 5.4.

   These bits are defined for inclusion in the LSP Attributes TLV of the
   LSP_ATTRIBUTES. The values shown have been assigned by IANA.

9. Security Considerations

   The RSVP-TE trust model assumes that RSVP-TE neighbors and peers
   trust each other to exchange legitimate and non-malicious messages.
   This assumption is necessary in order that the signaling protocol can
   function.

   Note that this trust model is assumed to cascade. That is, if an LSR
   trusts its neighbors, it extends this trust to all LSRs that its
   neighbor trusts. This means that the trust model is usually applied
   across the whole network to create a trust domain.

   Authentication of neighbor identity is already a standard provision
   of RSVP-TE, as is the protection of messages against tampering and
   spoofing. Refer to [RFC2205], [RFC3209], and [RFC3473] for a
   description of applicable security considerations. These
   considerations and mechanisms are applicable to hop-by-hop message
   exchanges (such as used for crankback propagation on PathErr
   messages) and directed message exchanges (such as used for crankback
   propagation on Notify messages).

   Key management may also be used with RSVP-TE to help to protect
   against impersonation and message content falsification. This
   requires the maintenance, exchange, and configuration of keys on each
   LSR. Note that such maintenance may be especially onerous to
   operators, hence it is important to limit the number of keys while
   ensuring the required level of security.

   This document does not introduce any protocol elements or message
   exchanges that change the operation of RSVP-TE security.

However, it should be noted that crankback is envisaged as an inter-
domain mechanism, and as such it is likely that crankback information
is exchanged over trust domain borders. In these cases, it is
expected that the information from within a neighboring domain would
be of little or no value to the node performing crankback re-routing
and would be ignored. In any case, it is highly likely that the
reporting domain will have applied some form of information
aggregation in order to preserve the confidentiality of its network
topology.

The issue of a direct attack by one domain upon another domain is
possible and domain administrators should apply policies to protect
their domains against the results of another domain attempting to
thrash LSPs by allowing them to set up before reporting them as
failed. On the whole, it is expected that commercial contracts
between trust domains will provide a degree of protection.

A more serious threat might arise if a domain reports that neither it
nor its downstream neighbor can provide a path to the destination.
Such a report could be bogus in that the reporting domain might not
have allowed the downstream domain the chance to attempt to provide a
path. Note that the same problem does not arise for nodes within a
domain because of the trust model. This type of malicious behavior
is hard to overcome, but may be detected by use of indirect path
computation requests sent direct to the falsely reported domain using
mechanisms such as the Path Computation Element [RFC4655].

Note that a separate document describing inter-domain MPLS and GMPLS
security considerations will be produced.

Finally, it should be noted that while the extensions in this
document introduce no new security holes in the protocols, should a
malicious user gain protocol access to the network, the crankback
information might be used to prevent establishment of valid LSPs.
Thus, the existing security features available in RSVP-TE should be
carefully considered by all deployers and SHOULD be made available by
all implementations that offer crankback. Note that the
implementation of re-routing attempt thresholds are also particularly
useful in this context.

10. Acknowledgments

We would like to thank Juha Heinanen and Srinivas Makam for their
review and comments, and Zhi-Wei Lin for his considered opinions.
Thanks, too, to John Drake for encouraging us to resurrect this
document and consider the use of the IF_ID ERROR SPEC object. Thanks
for a welcome and very thorough review by Dimitri Papadimitriou.

11. References

11.1. Normative References

   [RFC2119]   Bradner, S., "Key words for use in RFCs to Indicate
               Requirement Levels", BCP 14, RFC 2119, March 1997.

   [RFC2205]   Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S.
               Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1
               Functional Specification", RFC 2205, September 1997.

   [RFC3209]   Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V.,
               and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP
               Tunnels", RFC 3209, December 2001.

   [RFC3471]   Berger, L., Ed., "Generalized Multi-Protocol Label
               Switching (GMPLS) Signaling Functional Description", RFC
               3471, January 2003.

   [RFC3473]   Berger, L., Ed., "Generalized Multi-Protocol Label
               Switching (GMPLS) Signaling Resource ReserVation
               Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC
               3473, January 2003.

   [RFC4420]   Farrel, A., Ed., Papadimitriou, D., Vasseur, J.-P., and A.
               Ayyangar, "Encoding of Attributes for Multiprotocol Label
               Switching (MPLS) Label Switched Path (LSP) Establishment
               Using Resource ReserVation Protocol-Traffic Engineering
               (RSVP-TE)", RFC 4420, February 2006.

11.2. Informative References

   [ASH1]   G. Ash, ITU-T Recommendations E.360.1 --> E.360.7, "QoS
               Routing & Related Traffic Engineering Methods for IP-,
               ATM-, & TDM-Based Multiservice Networks", May, 2002.

   [RFC2702]   Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., and J.
               McManus, "Requirements for Traffic Engineering Over MPLS",
               RFC 2702, September 1999.

   [RFC3469] Sharma, V., Ed., and F. Hellstrand, Ed., "Framework for
              Multi-Protocol Label Switching (MPLS)-based Recovery", RFC
              3469, February 2003.

   [RFC4090] Pan, P., Ed., Swallow, G., Ed., and A. Atlas, Ed., "Fast
              Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090,
              May 2005.

   [RFC4201] Kompella, K., Rekhter, Y., and L. Berger, "Link Bundling
              in MPLS Traffic Engineering (TE)", RFC 4201, October 2005.

   [RFC4655] Farrel, A., Vasseur, J.-P., and J. Ash, "A Path
              Computation Element (PCE)-Based Architecture", RFC 4655,
              August 2006.

   [RFC4874] Lee, CY., Farrel, A., and S. De Cnodder, "Exclude Routes -
              Extension to Resource ReserVation Protocol-Traffic
              Engineering (RSVP-TE)", RFC 4874, April 2007.

   [PNNI]   ATM Forum, "Private Network-Network Interface
              Specification Version 1.0 (PNNI 1.0)", <af-pnni-0055.000>,
              May 1996.

Appendix A.   Experience of Crankback in TDM-Based Networks

   Experience of using release messages in TDM-based networks for
   analogous repair and re-routing purposes provides some guidance.

   One can use the receipt of a release message with a Cause Value (CV)
   indicating "link congestion" to trigger a re-routing attempt at the
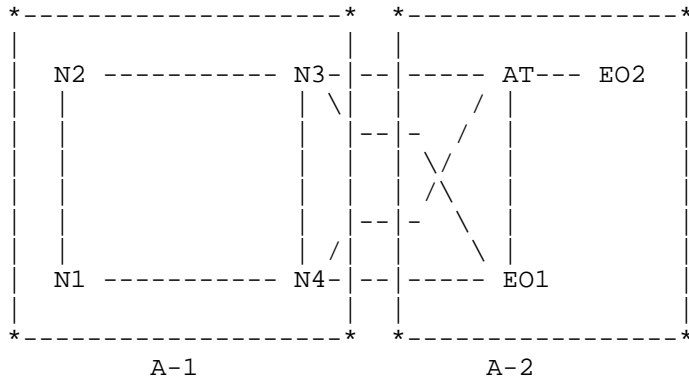   originating node. However, this sometimes leads to problems.

```
       *--------------------*  *----------------*
       |                    |  |                |
       |  N2 ---------- N3-|--|----- AT--- EO2  |
       |  |             | \|  |     / |         |
       |  |             |  |--|- /   |          |
       |  |             |  |  | \/   |          |
       |  |             |  |  | /\   |          |
       |  |             |  |--|- \   |          |
       |  |             | /|  |   \  |          |
       |  N1 ---------- N4-|--|----- EO1         |
       |                  |  |                  |
       *--------------------*  *----------------*
             A-1                     A-2
```

             Figure 1.   Example of network topology

   Figure 1 illustrates four examples based on service-provider
   experiences with respect to crankback (i.e., explicit indication)
   versus implicit indication through a release with CV.   In this
   example, N1, N2,N3, and N4 are located in one area (A-1), and AT,
   EO1, and EO2 are in another area (A-2).

   Note that two distinct areas are used in this example to clearly
   expose the issues. In fact, the issues are not limited to multi-area
   networks, but arise whenever path computation is distributed
   throughout the network, for example, where loose routes, AS routes,
   or path computation domains are used.

   1. A connection request from node N1 to EO1 may route to N4 and then
      find "all circuits busy". N4 returns a release message to N1 with
      CV34 indicating all circuits busy. Normally, a node such as N1 is
      programmed to block a connection request when receiving CV34,
      although there is good reason to try to alternately route the
      connection request via N2 and N3.

   Some service providers have implemented a technique called Route
   Advance (RA), where if a node that is RA capable receives a
   release message with CV34, it will use this as an implicit re-
   route indication and try to find an alternate route for the
   connection request if possible. In this example, alternate route
   N1-N2-N3-EO1 can be tried and may well succeed.

   2. Suppose a connection request goes from N2 to N3 to AT while trying
      to reach EO2 and is blocked at link AT-EO2. Node AT returns a
      CV34 and with RA, N2 may try to re-route N2-N1-N4-AT-EO2, but of
      course this fails again. The problem is that N2 does not realize
      where this blocking occurred based on the CV34, and in this case
      there is no point in further alternate routing.

   3. However, in another case of a connection request from N2 to E02,
      suppose that link N3-AT is blocked. In this case N3 should return
      crankback information (and not CV34) so that N2 can alternate
      route to N1-N4-AT-EO2, which may well be successful.

   4. In a final example, for a connection request from EO1 to N2, EO1
      first tries to route the connection request directly to N3.
      However, node N3 may reject the connection request even if there
      is bandwidth available on link N3-EO1 (perhaps for priority
      routing considerations, e.g., reserving bandwidth for high
      priority connection requests). However, when N3 returns CV34 in
      the release message, EO1 blocks the connection request (a normal
      response to CV34 especially if E01-N4 is already known to be
      blocked) rather than trying to alternate route through AT-N3-N2,
      which might be successful. If N3 returns crankback information,
      EO1 could respond by trying the alternate route.

      It is certainly the case that with topology exchange, such as
      OSPF, the ingress LSR could infer the re-routing condition.
      However, convergence of routing information is typically slower
      than the expected LSP setup times. One of the reasons for
      crankback is to avoid the overhead of available-link-bandwidth
      flooding, and to more efficiently use local state information to
      direct alternate routing at the ingress-LSR.

   [ASH1] shows how event-dependent-routing can just use crankback, and
   not available-link-bandwidth flooding, to decide on the re-route path
   in the network through "learning models". Reducing this flooding
   reduces overhead and can lead to the ability to support much larger
   AS sizes.

   Therefore, the alternate routing should be indicated based on an
   explicit indication (as in examples 3 and 4), and it is best to know
   the following information separately:

   a) where blockage/congestion occurred (as in examples 1-2)

      and

   b) whether alternate routing "should" be attempted even if there
      is no "blockage" (as in example 4).

Authors' Addresses

   Adrian Farrel (Editor)
   Old Dog Consulting
   Phone: +44 (0) 1978 860944
   EMail: adrian@olddog.co.uk

   Arun Satyanarayana
   Cisco Systems, Inc.
   170 West Tasman Dr.
   San Jose, CA 95134
   Phone: +1 408 853-3206
   EMail: asatyana@cisco.com

   Atsushi Iwata
   NEC Corporation
   System Platforms Research Laboratories
   1753 Shimonumabe Nakahara-ku,
   Kawasaki, Kanagawa, 211-8666, JAPAN
   Phone: +81-(44)-396-2744
   Fax: +81-(44)-431-7612
   EMail: a-iwata@ah.jp.nec.com

   Norihito Fujita
   NEC Corporation
   System Platforms Research Laboratories
   1753 Shimonumabe Nakahara-ku,
   Kawasaki, Kanagawa, 211-8666, JAPAN
   Phone: +81-(44)-396-2091
   Fax: +81-(44)-431-7644
   EMail: n-fujita@bk.jp.nec.com

   Gerald R. Ash
   AT&T
   EMail: gash5107@yahoo.com