Internet Engineering Task Force (IETF)
Request for Comments: 9494
Updates: 6368
Category: Standards Track
ISSN: 2070-1721

J. Uttaro
Independent Contributor
E. Chen
Palo Alto Networks
B. Decraene
Orange
J. Scudder
Juniper Networks

November 2023

Long-Lived Graceful Restart for BGP

Abstract

This document introduces a BGP capability called the "Long-Lived Graceful Restart Capability" (or "LLGR Capability"). The benefit of this capability is that stale routes can be retained for a longer time upon session failure than is provided for by BGP Graceful Restart (as described in RFC 4724). A well-known BGP community called "LLGR_STALE" is introduced for marking stale routes retained for a longer time. A second well-known BGP community called "NO_LLGR" is introduced for marking routes for which these procedures should not be applied. We also specify that such long-lived stale routes be treated as the least preferred and that their advertisements be limited to BGP speakers that have advertised the capability. Use of this extension is not advisable in all cases, and we provide guidelines to help determine if it is.

This memo updates RFC 6368 by specifying that the LLGR_STALE community must be propagated into, or out of, the path attributes exchanged between the Provider Edge (PE) and Customer Edge (CE) routers.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Routing protocols in general, and BGP in particular, have historically been designed with a focus on "correctness", where a key part of correctness is for each network element's forwarding state to converge to the current state of the network as quickly as possible. For this reason, the protocol was designed to remove state advertised by routers that went down (from a BGP perspective) as quickly as possible. Over time, this has been relaxed somewhat, notably by BGP Graceful Restart (GR) [RFC4724]; however, the paradigm has remained one of attempting to rapidly remove stale state from the network.

Over time, two phenomena have arisen that call into question the underlying assumptions of this paradigm.

1. The widespread adoption of tunneled forwarding infrastructures (for example, MPLS). Such infrastructures eliminate the risk of some types of forwarding loops that can arise in hop-by-hop forwarding; thus, they reduce one of the motivations for strong consistency between forwarding elements.

2. The increasing use of BGP as a transport for data that is less closely associated with packet forwarding than was originally the case. Examples include the use of BGP for auto-discovery (Virtual Private LAN Service (VPLS) [RFC4761]) and filter programming (Flow Specification (FLOWSPEC) [RFC8955]). In these cases, BGP data takes on a character more akin to configuration than to conventional routing.

The observations above motivate a desire to offer network operators the ability to choose to retain BGP data for a longer period than has hitherto been possible when the BGP control plane fails for some reason. Although the semantics of BGP Graceful Restart [RFC4724] are close to those desired, several gaps exist, most notably in the maximum time for which stale information can be retained: Graceful Restart imposes a 4095-second upper bound.

In this document, we introduce a BGP capability called the "Long-Lived Graceful Restart Capability". The goal of this capability is that stale information can be retained for a longer time across a session reset. We also introduce two BGP well-known communities:

* LLGR_STALE to mark such information, and

* NO_LLGR to indicate that these procedures should not be applied to the marked route.

Long-lived stale information is to be treated as least preferred, and its advertisement limited to BGP speakers that support the capability. Where possible, we reference the semantics of BGP Graceful Restart [RFC4724] rather than specifying similar semantics in this document.

The expected deployment model for this extension is that it will only be invoked for certain address families. This is discussed in more detail in Section 5. The use of this extension may be combined with that of conventional Graceful Restart; in such a case, it is invoked after the conventional Graceful Restart interval has elapsed. When not combined, LLGR is invoked immediately. Apart from the potential to greatly extend the timer, the most obvious difference between LLGR and conventional Graceful Restart is that in LLGR, routes are "depreferenced"; that is, they are treated as least preferred. Contrarily, in conventional GR, route preference is not affected. The design choice to treat long-lived stale routes as least preferred was informed by the expectation that they might be retained for (potentially) an almost unbounded period of time; whereas, in the conventional Graceful Restart case, stale routes are retained for only a brief interval. In the case of Graceful Restart, the trade-off between advertising new route status (at the cost of routing churn) and not advertising it (at the cost of suboptimal or incorrect route selection) is resolved in favor of not advertising. In the case of LLGR, it is resolved in favor of advertising new state, using stale information only as a last resort.

Section 7 provides some simple examples illustrating the operation of this extension.

2. Terminology

2.1. Definitions

Depreference: A route is said to be depreferenced if it has its route selection preference reduced in reaction to some event.

Helper: Sometimes referred to as "helper router". During Graceful Restart or Long-Lived Graceful Restart, the router that detects a session failure and applies the listed procedures. [RFC4724] refers to this as the "receiving speaker".

Route: In this document, "route" means any information encoded as BGP Network Layer Reachability Information (NLRI) and a set of path attributes. As discussed above, the connection between such routes and the installation of forwarding state may be quite remote.

Further note that, for brevity, in this document when we reference conventional Graceful Restart, we cite its base specification, [RFC4724]. That specification has been updated by [RFC8538]. The citation to [RFC4724] is not intended to be limiting.

2.2. Abbreviations

CE: Customer Edge (See [RFC4364] for more information on Customer Edge routers.)

EoR: End-of-RIB (See Section 2 of [RFC4724] for more information on End-of-RIB markers.)

GR: Graceful Restart (See [RFC4724] for more information on GR.) This term is also sometimes referred to herein as "conventional Graceful Restart" or "conventional GR" to distinguish it from the "Long-Lived Graceful Restart" or "LLGR" defined by this document.

LLGR: Long-Lived Graceful Restart

LLST: Long-Lived Stale Time

PE: Provider Edge (See [RFC4364] for more information on Provider Edge routers.)

VRF: VPN Routing and Forwarding (See [RFC4364] for more information on VRF tables.)

2.3. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in

BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Protocol Extensions

A BGP capability and two BGP communities are introduced in the subsections that follow.

3.1. Long-Lived Graceful Restart Capability

The "Long-Lived Graceful Restart Capability", or "LLGR Capability", (value: 71) is a BGP capability [RFC5492] that can be used by a BGP speaker to indicate its ability to preserve its state according to the procedures of this document. If the LLGR capability is advertised, the Graceful Restart capability [RFC4724] MUST also be advertised; see Section 4.1.

The capability value consists of zero or more tuples <AFI, SAFI, Flags, LLST> as follows:

```
+--------------------------------------------------+
| Address Family Identifier (16 bits)              |
+--------------------------------------------------+
| Subsequent Address Family Identifier (8 bits)    |
+--------------------------------------------------+
| Flags for Address Family (8 bits)                |
+--------------------------------------------------+
| Long-Lived Stale Time (24 bits)                  |
+--------------------------------------------------+
| ...                                              |
+--------------------------------------------------+
| Address Family Identifier (16 bits)              |
+--------------------------------------------------+
| Subsequent Address Family Identifier (8 bits)    |
+--------------------------------------------------+
| Flags for Address Family (8 bits)                |
+--------------------------------------------------+
| Long-Lived Stale Time (24 bits)                  |
+--------------------------------------------------+
```

The meaning of the fields are as follows:

Address Family Identifier (AFI), Subsequent Address Family Identifier (SAFI):
The AFI and SAFI, taken in combination, indicate that the BGP speaker has the ability to preserve its forwarding state for the address family during a subsequent BGP restart. Routes may be either:

- explicitly associated with a particular AFI and SAFI if using the encoding described in [RFC4760], or

- implicitly associated with <AFI=IPv4, SAFI=Unicast> if using the encoding described in [RFC4271].

Flags for Address Family:

This field contains bit flags relating to routes that were advertised with the given AFI and SAFI.

```
 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+
|F|   Reserved  |
+-+-+-+-+-+-+-+-+
```

The most significant bit is used to indicate whether the state for routes that were advertised with the given AFI and SAFI has indeed been preserved during the previous BGP restart. When set (value 1), the bit indicates that the state has been preserved. This bit is called the "F bit" since it was historically used to indicate the preservation of forwarding state. Use of the F bit is detailed in Section 4.2. The remaining bits are reserved and MUST be set to zero by the sender and ignored by the receiver.

Long-Lived Stale Time:

This time (in seconds) specifies how long stale information (for this AFI/SAFI) may be retained by the receiver (in addition to the period specified by the "Restart Time" in the Graceful Restart Capability). Because the potential use cases for this extension vary widely, there is no suggested default value for the LLST.

3.2. LLGR_STALE Community

The well-known BGP community LLGR_STALE (value: 0xFFFF0006) can be used to mark stale routes retained for a longer period of time (see [RFC1997] for more information on BGP communities). Such long-lived stale routes are to be handled according to the procedures specified in Section 4.

An implementation MAY allow users to configure policies that accept, reject, or modify routes based on the presence or absence of this community.

3.3. NO_LLGR Community

The well-known BGP community NO_LLGR (value: 0xFFFF0007) can be used to mark routes that a BGP speaker does not want to be treated according to these procedures, as detailed in Section 4.

An implementation MAY allow users to configure policies that accept, reject, or modify routes based on the presence or absence of this community.

4. Theory of Operation

If a BGP speaker is configured to support the procedures of this document, it MUST use BGP Capabilities Advertisement [RFC5492] to advertise the Long-Lived Graceful Restart Capability. The setting of the parameters for an AFI/SAFI depends on the properties of the BGP speaker, network scale, and local configuration.

In the presence of the Long-Lived Graceful Restart Capability, the procedures specified in [RFC4724] continue to apply unless explicitly revised by this document.

4.1. Use of the Graceful Restart Capability

If the LLGR Capability is advertised, the Graceful Restart capability MUST also be advertised. If it is not so advertised, the LLGR Capability MUST be disregarded. The purpose for mandating this is to enable the reuse of certain base mechanisms that are common to both "flavors" notably: origination, collection, and processing of EoR as well as the finite-state-machine modifications and connection-reset logic introduced by GR.

We observe that, if support for conventional Graceful Restart is not desired for the session, the conventional GR phase can be skipped by omitting all AFIs/SAFIs from the GR Capability, advertising a Restart Time of zero, or both. Section 4.2 discusses the interaction of conventional and LLGR.

4.2. Session Resets

BGP Graceful Restart [RFC4724] defines conditions under which a BGP session can reset and have its associated routes retained. If such a reset occurs for a session in which the LLGR Capability has also been exchanged, the following procedures apply:

- If the Graceful Restart Capability that was received does not list all AFIs/SAFIs supported by the session, then the GR Restart Time shall be deemed zero for those AFIs/SAFIs that are not listed.

- Similarly, if the received LLGR Capability does not list all AFIs/SAFIs supported by the session, then the Long-Lived Stale Time shall be deemed zero for those AFIs/SAFIs that are not listed.

The following text in Section 4.2 of [RFC4724] no longer applies:

> If the session does not get re-established within the "Restart Time" that the peer advertised previously, the Receiving Speaker MUST delete all the stale routes from the peer that it is retaining.

and the following procedures are specified instead:

After the session goes down, and before the session is re-established, the stale routes for an AFI/SAFI MUST be retained. The interval for which they are retained is limited by the sum of the Restart Time in the received Graceful Restart Capability and the Long-Lived Stale Time in the received Long-Lived Graceful Restart Capability. The timers received in the Long-Lived Graceful Restart Capability SHOULD be modifiable by local configuration, which may impose an upper bound, a lower bound, or both on their respective values.

If the value of the Restart Time or the Long-Lived Stale Time is zero, the duration of the corresponding period would be zero seconds. For example, if the Restart Time is zero and the Long-Lived Stale Time is nonzero, only the procedures particular to LLGR would apply.

Conversely, if the Long-Lived Stale Time is zero and the Restart Time is nonzero, only the procedures of GR would apply. If both are zero, none of these procedures would apply, only those of the base BGP specification [RFC4271] (although EoR would still be used as detailed in [RFC4724]). And finally, if both are nonzero, then the procedures would be applied serially: first those of GR and then those of LLGR. During the first interval, we observe that, while the procedures of GR are in effect, route preference would not be affected. During the second interval, while LLGR procedures are in effect, routes would be treated as least preferred as specified elsewhere in this document.

Once the Restart Time period ends (including the case in which the Restart Time is zero), the LLGR period is said to have begun and the following procedures MUST be performed:

* For each AFI/SAFI for which it has received a nonzero Long-Lived Stale Time, the helper router MUST start a timer for that Long-Lived Stale Time. If the timer for the Long-Lived Stale Time for a given AFI/SAFI expires before the session is re-established, the helper MUST delete all stale routes of that AFI/SAFI from the neighbor that it is retaining.

* The helper router MUST attach the LLGR_STALE community to the stale routes being retained. Note that this requirement implies that the routes would need to be readvertised in order to disseminate the modified community.

* If any of the routes from the peer have been marked with the NO_LLGR community, either as sent by the peer or as the result of a configured policy, they MUST NOT be retained and MUST be removed as per the normal operation of [RFC4271].

* The helper router MUST perform the procedures listed in Section 4.3.

Once the session is re-established, the procedures specified in [RFC4724] apply for the stale routes irrespective of whether the stale routes are retained during the Restart Time period or the Long-Lived Stale Time period. However, in the case of consecutive restarts, the previously marked stale routes MUST NOT be deleted before the timer for the Long-Lived Stale Time expires.

Similar to [RFC4724], once the LLGR Period begins, the Helper MUST immediately remove all the stale routes from the peer that it is retaining for that address family if any of the following occur:

* the F bit for a specific address family is not set in the newly received LLGR Capability, or

* a specific address family is not included in the newly received LLGR Capability, or

* the LLGR and accompanying GR Capability are not received in the re-established session at all.

If a Long-Lived Stale Time timer is running for routes with a given

AFI/SAFI received from a peer, it MUST NOT be updated (other than by manual operator intervention) until the peer has established and synchronized a new session. The session is termed "synchronized" for a given AFI/SAFI once the EoR for that AFI/SAFI has been received from the peer or once the Selection_Deferral_Timer discussed in [RFC4724] expires.

The value of a Long-Lived Stale Time in the capability received from a neighbor MAY be reduced by local configuration.

While the session is down, the expiration of a Long-Lived Stale Time timer is treated analogously to the expiration of the Restart Time timer in [RFC4724], other than applying only to the AFI/SAFI it accompanies. However, the timer continues to run once the session has re-established. The timer is neither stopped nor updated until the EoR marker is received for the relevant AFI/SAFI from the peer. If the timer expires during synchronization with the peer, any stale routes that the peer has not refreshed are removed. If the session subsequently resets prior to becoming synchronized, any remaining routes (for the AFI/SAFI whose LLST timer expired) MUST be removed immediately.

4.3. Processing LLGR_STALE Routes

A BGP speaker that has advertised the Long-Lived Graceful Restart Capability to a neighbor MUST perform the following upon receiving a route from that neighbor with the LLGR_STALE community or upon attaching the LLGR_STALE community itself per Section 4.2:

- Treat the route as the least preferred in route selection (see below). See Section 5.2 for a discussion of potential risks inherent in doing this.

- The route SHOULD NOT be advertised to any neighbor from which the Long-Lived Graceful Restart Capability has not been received. The exception is described in Section 4.6. Note that this requirement implies that such routes should be withdrawn from any such neighbor.

- The LLGR_STALE community MUST NOT be removed when the route is further advertised.

4.4. Route Selection

A least preferred route MUST be treated as less preferred than any other route that is not also least preferred. When performing route selection between two routes when both are least preferred, normal tiebreaking applies. Note that this would only be expected to happen if the only routes available for selection were least preferred; in all other cases, such routes would have been eliminated from consideration.

4.5. Errors

If the LLGR Capability is received without an accompanying GR Capability, the LLGR Capability MUST be ignored, that is, the implementation MUST behave as though no LLGR Capability has been received.

4.6. Optional Partial Deployment Procedure

Ideally, all routers in an Autonomous System (AS) would support this specification before it were enabled. However, to facilitate incremental deployment, stale routes MAY be advertised to neighbors that have not advertised the Long-Lived Graceful Restart Capability under the following conditions:

* The neighbors MUST be internal (Internal BGP (IBGP) or Confederation) neighbors.

* The NO_EXPORT community [RFC1997] MUST be attached to the stale routes.

* The stale routes MUST have their LOCAL_PREF set to zero. See Section 5.2 for a discussion of potential risks inherent in doing this.

If this strategy for partial deployment is used, the network operator should set the LOCAL_PREF to zero for all long-lived stale routes throughout the Autonomous System. This trades off a small reduction in flexibility (ordering may not be preserved between competing long-lived stale routes) for consistency between routers that do, and do not, support this specification. Since the consistency of route selection can be important for preventing forwarding loops, the latter consideration dominates.

4.7. Procedures When BGP Is the PE-CE Protocol in a VPN

4.7.1. Procedures When EBGP Is the PE-CE Protocol in a VPN

In VPN deployments (for example, [RFC4364]), External BGP (EBGP) is often used as a PE-CE protocol. It may be a practical necessity in such deployments to accommodate interoperation with peer routers that cannot easily be upgraded to support specifications such as this one. This leads to a problem: the procedures defined elsewhere in this document generally prevent LLGR stale routes from being sent across EBGP sessions that don't support LLGR, but this could prevent the VPN routes from being used for their intended purpose.

We observe that the principal motivation for restricting the propagation of "stale" routing information is the desire to prevent it from spreading without limit once it exits the "safe" perimeter. We further observe that VPN deployments are typically topologically constrained, making this concern moot. For this reason, an implementation MAY advertise stale routes over a PE-CE session, when explicitly configured to do so. That is, the second rule listed in Section 4.3 MAY be disregarded in such cases. All other rules continue to apply. Finally, if this exception is used, the implementation SHOULD, by default, attach the NO_EXPORT community to the routes in question, as an additional protection against stale routes spreading without limit. Attachment of the NO_EXPORT community MAY be disabled by explicit configuration in order to accommodate exceptional cases.

See further discussion of using an explicitly configured policy to mitigate this issue in Section 5.1.

4.7.2. Procedures When IBGP Is the PE-CE Protocol in a VPN

If IBGP is used as the PE-CE protocol, following the procedures of [RFC6368], then when a PE router imports a VPN route that contains the ATTR_SET attribute into a destination VRF and subsequently advertises that route to a CE router:

* If the CE router supports the procedures of this document (in other words, if the CE router has advertised the LLGR Capability):

    In addition to including the path attributes derived from the ATTR_SET attribute in the advertised route as per [RFC6368], the PE router MUST also include the LLGR_STALE community if it is present in the path attributes of the imported route, even if it is not present in the ATTR_SET attribute.

* If the CE router does not support the procedures of this document:

    Then the optional procedures of Section 4.6 MAY be followed, attaching the NO_EXPORT community and setting the value of LOCAL_PREF to zero, overriding the value found in the ATTR_SET.

Similarly, when a PE router receives a route from a CE into its VRF and subsequently exports that route to a VPN address family:

* If the PE router supports the procedures of this document (in other words, if the PE router has advertised the LLGR Capability):

    In addition to including in the VPN route the ATTR_SET derived from the path attributes as per [RFC6368], the PE router MUST also include the LLGR_STALE community in the VPN route if it is present in the path attributes of the route as received from the CE.

* If the PE router does not support the procedures of this document:

    There exists no ideal solution. The CE could advertise a route with LLGR_STALE, with the understanding that the LLGR_STALE marking will only be honored by the provider network if appropriate policy configuration exists on the PE (see Section 5.1). It is at least guaranteed that LLGR_STALE will be propagated when the route is propagated beyond the provider network, or the CE could refrain from advertising the LLGR_STALE route to the incapable PE.

5. Deployment Considerations

The deployment considerations discussed in [RFC4724] apply to this document. In addition, network operators are cautioned to carefully consider the potential disadvantages of deploying these procedures for a given AFI/SAFI. Most notably, if used for an AFI/SAFI that conveys conventional reachability information, the use of a long-lived stale route could result in a loss of connectivity for the covered prefix. This specification takes pains to mitigate this risk where possible by making such routes least preferred and by restricting the scope of such routes to routers that support these procedures (or, optionally, a single Autonomous System, see Section 4.6). However, if a stale route is chosen as best for a given prefix, then according to the normal rules of IP forwarding, that route will be used for matching destinations, even if a non-stale less specific matching route is also available. Networks in which the deployment of these procedures would be especially concerning include those that do not use "tunneled" forwarding (in other words, those using conventional hop-by-hop forwarding).

Implementations MUST NOT enable these procedures by default. They MUST require affirmative configuration per AFI/SAFI in order to enable them.

The procedures of this document do not alter the route resolvability requirement of Section 9.1.2.1 of [RFC4271]. Because of this, it will commonly be the case that "stale" IBGP routes will only continue to be used if the router depicted in the next hop remains resolvable, even if its BGP component is down. Details of IGP fault-tolerance strategies are beyond the scope of this document. In addition to the foregoing, it may be advisable to check the viability of the next hop through other means, for example, Bidirectional Forwarding Detection (BFD) [RFC5880]. This may be especially useful in cases where the next hop is known directly at the network layer, notably EBGP.

As discussed in this document, after a BGP session goes down and before the session is re-established, stale routes may be retained for up to two consecutive periods, controlled by the Restart Time and the Long-Lived Stale Time, respectively:

* During the first period, routing churn would be prevented, but with potential persistent packet loss.

* During the second period, potential persistent packet loss may be reduced, but routing churn would be visible throughout the network.

The setting of the relevant parameters for a particular application should take into account trade-offs, network dynamics, and potential failure scenarios. If needed, the first period can be bypassed either by local configuration or by setting the Restart Time in the Graceful Restart Capability to zero and/or not listing the AFI/SAFI in that capability.

The setting of the F bit (and the Forwarding State bit of the accompanying GR Capability) depends, in part, on deployment considerations. The F bit can be understood as an indication that the Helper should flush associated routes (if the bit is left clear). As discussed in Section 1, an important use case for LLGR is for routes that are more akin to configuration than to conventional routing. For such routes, it may make sense to always set the F bit, regardless of other considerations. Likewise, for control-plane-only entities, such as dedicated route reflectors that do not participate in the forwarding plane, it makes sense to always set the F bit. Overall, the rule of thumb is that if loss of state on the restarting router can reasonably be expected to cause a forwarding loop or persistent packet loss, the F bit should be set scrupulously according to whether state has been retained. Specifics of whether or not the F bit is set are implementation dependent and may also be controlled by configuration. Also, for every AFI/SAFI represented in the LLGR Capability that is also represented in the GR Capability, there will be two corresponding F bits: the LLGR F bit and the GR F bit. If the LLGR F bit is set, the corresponding GR F bit should also be set, since to do otherwise would cause the state to be cleared on the Receiving Router per the normal rules of GR, violating the intent of the set LLGR bit.

5.1. When BGP Is the PE-CE Protocol in a VPN

As discussed in Section 4.7, it may be necessary for a PE to advertise stale routes to a CE in some VPN deployments, even if the CE does not support this specification. In that case, the operator configuring their PE to advertise such routes should notify the operator of the CE receiving the routes, and the CE should be configured to depreference the routes.

Similarly, it may be necessary for a CE to advertise stale routes to a PE, even if the PE does not support this specification. In that case, the operator configuring their CE to advertise such routes should notify the operator of the PE receiving the routes, and the PE should be configured to depreference the routes.

Typical BGP implementations will be able to be configured to depreference routes by matching on the LLGR_STALE community and setting the LOCAL_PREF for matching routes to zero, similar to the procedure described in Section 4.6.

5.2. Risks of Depreferencing Routes

Depreferencing EBGP routes is considered safe, no different from the common practice of applying a routing policy to an EBGP session. However, the same is not always true of IBGP.

Consistent route selection is a fundamental tenet of IBGP correctness and safe operation in hop-by-hop routed networks. When routers within an AS apply different criteria in selecting routes, they can arrive at inconsistent route selections. This can lead to the formation of forwarding loops unless some form of tunneled forwarding is used to prevent "core" routers from making a (potentially inconsistent) forwarding decision based on the IP header.

This specification uses the state of a peering session as an input to the selection criteria, depreferencing routes that are associated with a session that has gone down but that have not yet aged out. Since different routers within an AS might have different notions as to whether their respective sessions with a given peer are up or down, they might apply different selection criteria to routes from that peer. This could result in a forwarding loop forming between such routers.

For an example of such a forwarding loop, consider the following simple topology:

```
A ---- B ---- C ------------------------- D
^                                         ^
|                                         |
R1                                        R2
```

Figure 1

In this example, A - D are routers with a full mesh of IBGP sessions between them (the sessions are not shown). The short links have unit cost, the long link has cost 5. Routers A and D are AS border routers, each advertising some route, R, with the same LOCAL_PREF into the AS: denoted R1 and R2 in the diagram. In ordinary operation, it can be seen that routers B and C will select R1 for forwarding and will forward toward A.

Suppose that the session between A and B goes down for some reason, and it stays down long enough for LLGR processing to be invoked on B. Then, on B, route R1 will be depreferenced, leading to the selection of R2 by B. However, C will continue to prefer R1. In this case, it can be seen that a forwarding loop for packets destined to R would form between B and C. (We note that other forwarding loop scenarios can be constructed for conventional GR, but these are generally considered less severe since GR can remain in effect for a much more limited interval.)

The potential benefits of this specification can outweigh the risks discussed above, as long as care is exercised in deployment. The cardinal rule to be followed is that if a given set of routes is being used within an AS for hop-by-hop forwarding, enabling LLGR procedures is not recommended. If tunneled forwarding (such as MPLS) is used within the AS, or if routes are being used for purposes other than hop-by-hop forwarding, less caution is needed; however, the operator should still carefully consider the consequences of enabling LLGR.

6. Security Considerations

The security implications of the LLGR mechanism defined in this document are akin to those incurred by the maintenance of stale routing information within a network. However, since the retention time may be much longer, the window during which certain attacks are feasible may substantially increase. This is particularly relevant when considering the maintenance of routing information that is used for service segregation, such as MPLS label entries.

For MPLS VPN services, the effectiveness of the traffic isolation between VPNs relies on the correctness of the MPLS labels between ingress and egress PEs. In particular, when an egress PE withdraws a label L1 allocated to a VPN1 route, this label must not be assigned to a VPN route of a different VPN until all ingress PEs stop using the old VPN1 route using L1.

Such a corner case may happen today if the propagation of VPN routes by BGP messages between PEs takes more time than the label reallocation delay on a PE. Given that we can generally bound the worst-case BGP propagation time to a few minutes (for example, 2-5 minutes), the security breach will not occur if PEs are designed to not reallocate a previously used and withdrawn label before a few minutes.

The problem is made worse with BGP GR between PEs because VPN routes can be stalled for a longer period of time (for example, 20 minutes).

This is further aggravated by the LLGR extension specified in this document because VPN routes can be stalled for a much longer period of time (for example, 2 hours, 1 day).

In order to exploit the vulnerability described above, an attacker needs to engineer a specific LLGR state between two PE devices and also cause the label reallocation to occur such that the two topologies overlap. To avoid the potential for a VPN breach, the operator should ensure that the lower bound for label reuse is greater than the upper bound on the LLST before enabling LLGR for a VPN address family. Section 4.2 discusses the provision of an upper bound on LLST. Details of features for setting a lower bound on label reuse time are beyond the scope of this document; however, factors that might need to be taken into account when setting this value include:

* The load of the BGP route churn on a PE (in terms of the number of VPN labels advertised and the churn rate).

* The label allocation policy on the PE, which possibly depends upon the size of the pool of the VPN labels (which can be restricted by hardware considerations or other MPLS usages), the label allocation scheme (for example, per route or per VRF/CE), and the reallocation policy (for example, least recently used label).

Note that [RFC4781], which defines the Graceful Restart Mechanism for BGP with MPLS, is also applicable to LLGR.

7. Examples of Operation

For illustrative purposes, we present a few examples of how this specification might be used in practice. These examples are neither exhaustive nor normative.

Consider the following scenario: A border router, ASBR1, has an IBGP peering with a route reflector, RR1, from which it learns routes. It has an EBGP peering with an external peer, EXT, to which it advertises those routes. The external peer has advertised the GR and LLGR Capabilities to ASBR1. ASBR1 is configured to support GR and LLGR on its sessions with RR1 and EXT. RR1 advertises a GR Restart Time of 1 (second) and an LLST of 3600 (seconds):

```
    +==========+======================================================+
```

Time	Event
t	ASBR1's IBGP session with RR fails. ASBR1 retains RR's routes according to the rules of GR [RFC4724].
t+1	GR Restart Time expires. ASBR1 transitions RR's routes to long-lived stale routes by attaching the LLGR_STALE community and depreferencing them. However, since it has no backup routes, it continues to make use of them. It re-announces them to EXT with the LLGR_STALE community attached.
t+1+3600	LLST expires. ASBR1 removes RR's stale routes from its own RIB and sends BGP updates to withdraw them from EXT.

Table 1

Next, imagine the same scenario, but suppose RR1 advertised a GR Restart Time of zero, effectively disabling GR. Equally, ASBR1 could have used a local configuration to override RR1's offered Restart Time, setting it to a locally configured value of zero:

Time	Event
t	ASBR1's IBGP session with RR fails. ASBR1 transitions RR's routes to long-lived stale routes by attaching the LLGR_STALE community and depreferencing them. However, since it has no backup routes, it continues to make use of them. It re-announces them to EXT with the LLGR_STALE community attached.
t+0+3600	LLST expires. ASBR1 removes RR's stale routes from its own RIB and sends BGP updates to withdraw them from EXT.

Table 2

Next, imagine the original scenario, but consider that the ASBR1-RR1 session comes back up and becomes synchronized 180 seconds after the failure was detected:

Time	Event
t	ASBR1's IBGP session with RR fails. ASBR1 retains RR's routes according to the rules of GR [RFC4724].
t+1	GR Restart Time expires. ASBR1 transitions RR's routes to long-lived stale routes by attaching the LLGR_STALE community and depreferencing them. However, since it has no backup routes, it continues to make use of them. It re-announces

	them to EXT with the LLGR_STALE community attached.
t+1+179	Session is re-established and resynchronized. ASBR1 removes the LLGR_STALE community from RR1's routes and re-announces them to EXT with the LLGR_STALE community removed.

Table 3

Finally, imagine the original scenario, but consider that EXT has not advertised the LLGR Capability to ASBR1:

Time	Event
t	ASBR1's IBGP session with RR fails. ASBR1 retains RR's routes according to the rules of GR [RFC4724].
t+1	GR Restart Time expires. ASBR1 transitions RR's routes to long-lived stale routes by attaching the LLGR_STALE community and depreferencing them. However, since it has no backup routes, it continues to make use of them. It withdraws them from EXT.
t+1+3600	LLST expires. ASBR1 removes RR's stale routes from its own RIB.

Table 4

8. IANA Considerations

This document defines a BGP capability called the "Long-Lived Graceful Restart Capability". IANA has assigned a value of 71 from the "Capability Codes" registry.

This document introduces two BGP well-known communities:

* the first called "LLGR_STALE" for marking long-lived stale routes, and

* the second called "NO_LLGR" for marking routes that should not be retained if stale.

IANA has assigned these well-known community values 0xFFFF0006 and 0xFFFF0007, respectively, from the "BGP Well-known Communities" registry.

IANA has established a registry called the "Long-Lived Graceful Restart Flags for Address Family" registry under the "Border Gateway Protocol (BGP) Parameters" group. The registration procedures are Standards Action (see [RFC8126]). The registry is initially populated as follows:

Bit Position	Name	Short Name	Reference
0	Preservation of state	F	RFC 9494
1-7	Unassigned		

Table 5

9. References

9.1. Normative References

[RFC1997] Chandra, R., Traina, P., and T. Li, "BGP Communities Attribute", RFC 1997, DOI 10.17487/RFC1997, August 1996, <https://www.rfc-editor.org/info/rfc1997>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <https://www.rfc-editor.org/info/rfc4271>.

[RFC4724] Sangli, S., Chen, E., Fernando, R., Scudder, J., and Y. Rekhter, "Graceful Restart Mechanism for BGP", RFC 4724, DOI 10.17487/RFC4724, January 2007, <https://www.rfc-editor.org/info/rfc4724>.

[RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, DOI 10.17487/RFC4760, January 2007, <https://www.rfc-editor.org/info/rfc4760>.

[RFC5492] Scudder, J. and R. Chandra, "Capabilities Advertisement with BGP-4", RFC 5492, DOI 10.17487/RFC5492, February 2009, <https://www.rfc-editor.org/info/rfc5492>.

[RFC6368] Marques, P., Raszuk, R., Patel, K., Kumaki, K., and T. Yamagata, "Internal BGP as the Provider/Customer Edge Protocol for BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 6368, DOI 10.17487/RFC6368, September 2011, <https://www.rfc-editor.org/info/rfc6368>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8538] Patel, K., Fernando, R., Scudder, J., and J. Haas, "Notification Message Support for BGP Graceful Restart", RFC 8538, DOI 10.17487/RFC8538, March 2019, <https://www.rfc-editor.org/info/rfc8538>.

9.2. Informative References

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <https://www.rfc-editor.org/info/rfc4364>.

[RFC4761] Kompella, K., Ed. and Y. Rekhter, Ed., "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, DOI 10.17487/RFC4761, January 2007, <https://www.rfc-editor.org/info/rfc4761>.

[RFC4781] Rekhter, Y. and R. Aggarwal, "Graceful Restart Mechanism for BGP with MPLS", RFC 4781, DOI 10.17487/RFC4781, January 2007, <https://www.rfc-editor.org/info/rfc4781>.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <https://www.rfc-editor.org/info/rfc5880>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8955] Loibl, C., Hares, S., Raszuk, R., McPherson, D., and M. Bacher, "Dissemination of Flow Specification Rules", RFC 8955, DOI 10.17487/RFC8955, December 2020, <https://www.rfc-editor.org/info/rfc8955>.

Acknowledgements

We would like to thank Nabil Bitar, Martin Djernaes, Roberto Fragassi, Jeffrey Haas, Jakob Heitz, Daniam Henriques, Nicolai Leymann, Mike McBride, Paul Mattes, John Medamana, Pranav Mehta, Han Nguyen, Saikat Ray, Valery Smyslov, and Bo Wu for their valuable input and contributions to the discussion and solution.

Contributors

Clarence Filsfils
Cisco Systems
1150 Brussels
Belgium
Email: cf@cisco.com

Pradosh Mohapatra
Sproute Networks
Email: mpradosh@yahoo.com

Yakov Rekhter

Eric Rosen
Email: erosen52@gmail.com

Rob Shakir
Google, Inc.
1600 Amphitheatre Parkway
Mountain View, CA 94043
United States of America
Email: robjs@google.com

Adam Simpson
Nokia
Email: adam.1.simpson@nokia.com

Authors' Addresses

James Uttaro
Independent Contributor
Email: juttaro@ieee.org

Enke Chen
Palo Alto Networks
Email: enchen@paloaltonetworks.com

Bruno Decraene
Orange
Email: bruno.decraene@orange.com

John G. Scudder
Juniper Networks
Email: jgs@juniper.net