Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: May 13, 2015

R. Shakir
BT

November 9, 2014

Operational Requirements for Enhanced Error Handling Behaviour in BGP-4
draft-ietf-grow-ops-reqs-for-bgp-error-handling-07

Abstract

BGP-4 is utilised as a key intra- and inter-Autonomous System routing protocol in modern IP networks. The failure modes as defined by the original protocol standards are based on a number of assumptions around the impact of session failure. Numerous incidents both in the global Internet routing table and within Service Provider networks have been caused by strict handling of a single invalid UPDATE message causing large-scale failures in one or more Autonomous Systems.

This memo describes the current use of BGP-4 within Service Provider networks, and outlines a set of requirements for further work to enhance the mechanisms available to a BGP-4 implementation when erroneous data is detected. Whilst this document does not provide specification of any standard, it is intended as an overview of a set of enhancements to BGP-4 to improve the protocol’s robustness to suit its current deployment.

Status of This Memo

Copyright Notice

Table of Contents

1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Problem Statement

BGP has developed into a key intra- and inter-domain routing protocol, deployed within both the Internet and private networks. The changing deployments of the protocol have resulted in increased demand for robustness of the routing system - with the error handling behaviour defined in [RFC4271] having been shown to have caused numerous incidents within live network deployments. This document intends to provide an overview of the current deployment cases for BGP-4, and define a set of requirements (from the perspective of a network operator) for enhancing error handling within the protocol.

2.1. Role of BGP-4 in Service Provider Networks

BGP was designed as an inter-autonomous system (AS) routing protocol. Many of the error handling mechanisms within the protocol are defined in order to be guarantee consistency and correctness of information between two neighbouring speakers. The assumption is made that each AS operates with many adjacencies, each propagating a relatively small amount of routing information. Through focusing on information consistency, the protocol specification prefers failure of an individual routing adjacency to maintaining reachability to all NLRI propagated through a particular neighbour, with the expectation that alternate, less direct, paths can be selected where a failure occurs. These assumptions resulted in the specification made in [RFC4271] whereby the receipt of an erroneous UPDATE message is reacted to by sending a NOTIFICATION message, and tearing down the adjacency with the remote speaker from whom the error was observed.

BGP’s deployments have evolved with the growth of IP-based services. Historically, a network would deploy an interior gateway protocol (IGP) to carry infrastructure and customer routes, and utilise an external gateway protocol (EGP) such as BGP to propagate routes to other autonomous systems. However, within modern deployments to ensure route convergence within an AS is within acceptable time bounds the amount of information within the IGP has been minimised (typically to only infrastructure routes). iBGP is then utilised to carry both internal, customer and external routes within an AS. As such, this has resulted in BGP having become an IGP, with traditional IGPs providing only reachability between nodes within the AS for packet forwarding, and to establish iBGP sessions. This change in role within the overall architecture of an AS has resulted in an increased robustness requirement for BGP, with the expectation of a similar level of robustness to that of an IGP being set. The loss of an iBGP session can result in significant levels of unreachability internally to an AS, especially since there are typically limited (when compared to the Internet) signalling and forwarding paths available.

The volume and nature of the information carried within BGP has also changed - it has become the ubiquitous means through which service information can be propagated between devices. For instance, being utilised to carry IP/MPLS service information such as Layer 3 IP VPN routes [RFC4364] , and Layer 2 Virtual Private LAN Service device membership [RFC4761]. Since these extensions to the protocol allow signalling of multiple services (represented by address families within BGP), and multiple customer topologies (i.e., subsets of routes within each address family) via the BGP protocol, the impact of session failure is increased. The tear down of a single BGP session can result in a complete outage to all customer services signalled via the session, even where the triggering event is related to only one service or topology being carried.

In addition, there has been significant growth in the volume of routing information carried in BGP. In numerous networks, the RIB size of individual BGP speakers can be of the order of millions of paths. Particularly large volumes are observed at BGP speakers performing aggregation and border roles (such as ASBR, or route reflector hierarchies). This increased volume of routes results not only in a significant number of services being impacted during a protocol failure, but also increases the time to recovery after re-establishing a BGP session. The time taken to learn, compute and distribute new paths increases the impact of failures on services carried by the network - adding further weight to the requirement to avoid failures, or limit the extent of their impact. Particularly, the impact of individual session failures is increased due to the existence of a relatively small number of highly-critical BGP sessions within Internet and multi-service network deployments. These sessions propagate a high-proportion of the reachability information - for instance, providing an Internet AS with the global routing table from upstream providers, or providing IP/MPLS Provider Edge devices adjacency with route reflector hierarchy providing signalling for elements of services connected elsewhere within the routing domain. In both cases, the failure of these sessions can result in a significant outage to customer services.

2.2. Service Requirements for Amended BGP Error Handling

Alongside the infrastructure requirements outlined above, service provider customer requirements continue to evolve. In particular, there are increasing requirements for robustness and fault isolation based on:

- o The increasing reliance on public IP service instead of private networks - resulting is requirements for greater availability of Internet services. The diversity of autonomous systems has resulted in individual BGP sessions within the Internet carrying

more routing information (e.g., IP transit, or large peering interconnections), which is originated from more individual networks - increasing both the impact of an individual session failure, and the number of different sources of error which can lead to its failure. To meet the requirement of high-availability Internet services, it is therefore an expectation that the error handling behaviour MUST affect only the those routes, or autonomous systems, that are are impacted by the erroneous messages, rather than all routes received by a particular session, such that the maximum service availability is maintained.

- o The requirement to support multiple services. In multi-service environments such as those that support L3VPNs, multiple customer VPNs are isolated from one another, and from other IP environments (such as the Internet). There is an expectation from a service perspective therefore that the customer service is within its own fault domain (even when carried via a shared set of signalling), hence an error on routes or BGP messages related to one VPN SHOULD NOT negatively impact other VPNs. Further to this, an error relating to another service (i.e., another address family, such as Internet or L2VPN services) SHOULD NOT impact the availability of the VPN service. Both of these principles of fault separation are required in order to support multiple services and segregated customer infrastructures over a common network infrastructure whilst meeting the availability required of them.

It should be noted that the requirements for fault isolation and high-availability do not imply that routing information that is potentially erroneous (through being carried in an UPDATE message that cannot be parsed for example) is always maintained despite questions as to its integrity, particularly as such routing information may result in leakage between services - but merely that there is a requirement to reconsider the balance between protocol correctness, and robustness.

In addition to these service requirements, an increasing requirement to minimise the time taken to recover from incidents exists. In some cases, this may require an operator to compromise on correctness in order to maintain integrity of a subset of routing information or services. To meet this requirement, mechanisms allowing an operator to ignore all errors or maintain "known good" routing information MAY be required. The implementation of such mechanisms is a business consideration of the service provider in question, and MUST consider the balance between the risk of incorrectness and the overall impact to a network platform. Such mechanisms are particularly of use where lack of routing information violates an operator’s policies (e.g., filtering rules distributed via BGP FlowSpec are no longer installed), or fault isolation requires significant external liaison

(such as contacting a third-party autonomous system to amend or filter route announcement).

3. Classes of Errors within UPDATE Messages

To meet the requirement to provide more targeted error handling, errors are therefore classified into the following scopes:

- o Attribute Scope - in this case, an error can be localised to a particular attribute within the message. For instance, such errors may occur when invalid flags are set within an individual attribute within a message, which is otherwise well-formed.

- o Message Scope - errors resulting in the inability to parse a single UPDATE message, but not affecting the ability of an implementation to parse subsequent BGP messages. For instance, where the overall length of an UPDATE message is correct, but the length of a single attribute contained within it is erroneously specified.

- o Session Scope - where errors occur such that an error in an UPDATE message results in the inability to the parse subsequent messages. In this case, attribute length errors may result in the inability for a BGP implementation to locate the bounds of an UPDATE, and hence the subsequent message from a peer.

For session-scope errors, the error handling approach implemented MUST conform with the requirements described in Section 5 of this document (generically referred to as "Critical" error handling mechanisms). Session-scope errors requiring Critical error handling MUST be the only case whereby the impact of error handling mechanisms should be allowed to impact entire BGP sessions between two BGP speakers.

For message- and attribute-level errors, "Non-Critical" error handling mechanisms SHOULD be used, which MUST meet the specification described in Section 4. In the case of attribute-scope errors, a BGP speaker MUST limit the impact of error-handling mechanisms to the NLRI carried within the message, and MAY (where applicable) limit to the scope of error handling to the individual attribute. Where a message-scope error occurs, a BGP speaker MUST limit the impact of error handling to the NLRI contained within the affected UPDATE.

3.1. Characteristics of Session Scope Errors

Based on analysis of existing BGP implementations, and incidents within the Internet and private network routing tables, it is expected that errors with a session level scope are restricted to:

- o UPDATE Message Length errors - where the specified UPDATE message length is inconsistent with the sum of the Total Path Attribute and Withdrawn Routes length. These errors relate to message packing or framing, and result in cases whereby the NLRI attribute cannot be correctly extracted from the message.

- o Errors parsing the NLRI attribute of an UPDATE message - where the contents of the IPv4 Unicast Advertised or Withdrawn Routes attributes, or multi-protocol BGP NLRI attributes (MP_REACH_NLRI and/or MP_UNREACH_NLRI as defined in [RFC2858]), cannot be successfully parsed.

3.2. Characteristics of Message Scope Errors

Message scope errors are restricted to those whereby erroneous encoding results in the ability to parse and determine the NLRI carried by the message - but the carried attributes are invalid. These errors (based on existing attributes) are limited to:

- o Errors where the length of all path attributes contained within the UPDATE does not correspond to the total path attribute length.

- o UPDATE messages missing mandatory attributes, unrecognised non-optional attributes, or those that contain duplicate or invalid attributes (be they unsupported, or unexpected).

- o Those messages where the NEXT_HOP, the MP_REACH_NLRI next-hop values are missing, zero-length, or invalid for the relevant address family.

3.3. Characteristics of Attribute Scope Errors

Attribute scope errors are defined to be those that relate to an individual attribute (not related to the NLRI) carried by an UPDATE message. Particularly, where:

- o Zero- or invalid-length errors in path attributes, excluding those containing NLRI.

- o Invalid data or flags are contained in a path attribute that does not relate to the NLRI.

3.4. Avoiding Session Scope Errors

In order to maximise the number of cases whereby the NLRI attributes can be reliably extracted from a received message, where a BGP speaker supports multi-protocol extensions, the MP_REACH_NLRI and MP_UNREACH_NLRI attributes SHOULD be utilised for all address

families (including IPv4 Unicast) and these attributes should be the first attribute contained within the UPDATE message. For these Non-Critical errors, the NLRI-targeted error handling requirements described in Section 4 should be followed.

3.5. Future Attributes introduced to BGP

Where attributes are introduced by future extensions to the BGP protocol error handling behaviour SHOULD be assumed to be be at a message- or attribute-scope, unless otherwise specified within the per-extension memo, or the attribute relates directly to carrying NLRI. It is recommended that authors of future BGP extensions SHOULD specify the error handling behaviour required on a per-attribute error basis.

4. Error Handling for Non-Critical Errors

4.1. NLRI-level Error Handling Requirements

When a Non-Critical error is detected within an UPDATE message a BGP speaker MUST NOT send a NOTIFICATION message to the remote neighbour. Instead, the NLRI contained within the message SHOULD be considered as being withdrawn by the neighbour (referred to as treat-as-withdraw), until they are updated by a subsequent UPDATE message. Where defined is acceptable by the relevant memo, for the specific-case of attribute-scope errors, the erroneous attribute MAY be discarded by an implementation. This attribute-discard approach MUST only be used for attributes that do not impact best-path selection within an implementation. An operator SHOULD consider the impact of implementing policies considering such attributes as part of the route selection algorithm, such that operator configuration does not result in unexpected consequences should such an attribute be discarded.

Network operators SHOULD recognise that where treat-as-withdraw behaviour is implemented black-holing or looping of traffic may occur in the period between the NLRI being treated as withdrawn, and subsequent updates, dependent upon the routing topology. It SHOULD be noted that such periods of RIB inconsistency (where one speaker has advertised a prefix, which has had treat-as-withdraw applied to it by the receiving speaker) may be relatively long lived, based on situations such as an erroneous implementation at the receiver, or the error occurring within an optional-transitive attribute not examined by the direct neighbour. In order to allow operators to select sessions on which this risk of inconsistency is acceptable, an implementation SHOULD provide means by which Non-Critical error handling can be disabled on a per-session basis.

Since the Non-Critical error handling required within this section results in no NOTIFICATION message being transmitted, the fact that an error has occurred, and there may be inconsistency between the local and remote BGP speaker MUST be flagged to the network operator through standard operational interfaces (e.g., SNMP, syslog). The information highlighted MUST include the NLRI identified to be contained within the error message, and SHOULD contain a exact copy of the received message for further analysis.

4.1.1. Notifying the Remote Peer of Non-Critical Errors

In order that the operator of the BGP speaker from whom an erroneous UPDATE message has been advertised is aware of the fact that some NLRI advertised to the remote speaker have been considered invalid, a BGP speaker SHOULD support mechanisms to report the occurrence of Non-Critical error handling to the remote speaker. The receiving speaker SHOULD transmit the NLRI contained within the erroneous message to the advertising speaker. An exact copy of the received UPDATE message SHOULD also be sent.

The exchange of such information related to events occurring as a result of BGP messages is not currently supported by any extension to the protocol. Clearly, where the two speakers reside within the same administrative domain, shared logging information can be utilised to identify the root cause of errors. However, in many cases these devices reside within separate administrative domains (e.g., are ASBRs for Internet or private networks). In this case, mechanisms allowing transmission in-band to the BGP session SHOULD be utilised (e.g., the OPERATIONAL message described in [I-D.ietf-idr-operational-message]). Such an in-band channel is preferred based on the BGP session representing a pre-established trusted source which is related to a specific BGP-speaking device within a network. It is expected that the overall system scalability of a BGP speaker is improved through utilising the existing channel, rather than incurring overhead for maintaining many additional sessions for relatively infrequent messaging events when errors occur. However, the extensions providing such a channel MUST consider their impact to base BGP protocol functions such as the transmission of UPDATE or KEEPALIVE messages, and SHOULD limit the volume of messaging to direct reactions to Non-Critical errors occurring. These considerations SHOULD be made in order to ensure that no compromise is made to the security, scalability and robustness of BGP. Where additional BGP monitoring information that is not suitable to be carried in-band is required, out-of-band mechanisms such as the BMP protocol described in [I-D.ietf-grow-bmp] could be utilised to provide further information relating to erroneous messages.

4.2. Recovering RIB Consistency following NLRI-level Error Handling

In order to recover consistency of Adj-RIBs following Non-Critical error handling, a means by which a validation and recovery of consistency can be achieved SHOULD be provided to an operator. This functionality MAY be provided through extension of the ROUTE-REFRESH [RFC2918] mechanism - providing means to identify the beginning and end of a replay of the entire Adj-RIB-Out of the advertising speaker (as per the suggestion in [I-D.ietf-idr-bgp-enhanced-route-refresh]).

As Non-Critical error handling is localised to the NLRI contained within the erroneous UPDATE message, a targeted recovery mechanism MAY be provided allowing a speaker to request re-advertisement of a particular subset of the Adj-RIB-Out. Where such targeted refresh functions are available, they SHOULD be preferred to mechanisms requesting re-advertisement of the whole Adj-RIB-Out based on their more limited use of CPU and network resources.

A BGP speaker may automatically trigger recovery mechanisms such as those described in this section following the receipt of an erroneous UPDATE message identified as Non-Critical to expedite recovery. It SHOULD be noted that if automatic recovery mechanisms trigger only re-advertisement of an identical erroneous message, they may be ineffective. Additionally, where the best-path to be advertised by remote speaker changes, this will be advertised directly, without a requirement for a request from the receiver. However, in some cases, RIB consistency recovery mechanisms may prompt alternate UPDATE message packing, and hence allow quicker recovery. Where such automatic mechanisms are implemented, those focused on smaller sets of NLRI SHOULD be preferred over those requesting the entire RIB. In addition, such mechanisms SHOULD have dampening mechanisms to ensure that their impact to computational and network resources is limited.

5. Error Handling for Critical Errors

Critical error handling MUST be used where session-scope errors occur. In such cases, a NOTIFICATION message MUST be sent to the remote peer. In order to limit the impact to network operation, during such events the mechanisms applied MUST allow for the paths NLRI received from the remote speaker to continue to be utilised during the session reset and re-establishment. It is envisaged that this requirement may be met through extension of the BGP Graceful Restart mechanism ([RFC4724]) to be triggered by NOTIFICATION messages indicating the occurrence of a Critical error. Such an extension allows a restart of the TCP and BGP sessions between two speakers, in a similar manner to the current session restart behaviour triggered by a NOTIFICATION message. In order to maximise the level of re-initialisation which occurs during such a restart

triggered by a Critical error, BGP speakers MAY re-initialise memory structures related to the RIB where possible.

Where such a restart event occurs, the continued liveliness of the remote device MAY be verified by BGP KEEPALIVE packets or other OAM functions such as Bidirectional Forwarding Detection ([RFC5880]). If the observed Critical BGP error is indicative of a wider device failure of the remote speaker, it is expected that a BGP sessions will not re-establish correctly. By default, each BGP speaker SHOULD maintain a limited time window in which session restart is expected in order to mitigate this possibility.

When a Critical error occurs, the network operator MUST be made aware of its occurrence through local logging mechanisms (e.g., SNMP traps or syslog). The BGP speaker receiving an UPDATE message identified as a Critical error MUST log its occurrence and a copy of the UPDATE message. Where a inter-device messaging mechanism is implemented (as discussed in Section Section 4.1) a copy of the erroneous UPDATE message SHOULD be transmitted to the remote speaker upon session-re-establishment (or via a separate session if implemented). Both BGP speakers MUST indicate to an operator the cause of a session restart was a Critical error in an UPDATE message.

Since repeated critical errors (and session restarts) may have an impact in overall device scaling if Critical error handling does not resolve the failure condition, a BGP speaker MAY choose to revert to the session tear down behaviour described in the base BGP specification. This reversion SHOULD only be utilised after a number of attempts which MUST be controllable by the network operator. Where a session is shut down, the implementation MAY utilise a back-off from session restart attempts (as per the IdleHoldTimer described in the BGP FSM [RFC4271]). Where reversion to tearing down the BGP session is performed, a speaker SHOULD limit the impact of withdrawing prefixes from downstream speakers where possible. It is envisaged that this can be achieved by utilising a mechanism such as the BGP Graceful Shutdown procedure as described in [I-D.ietf-grow-bgp-gshut].

5.1. Long-Lived Critical Errors

Where Critical error handling mechanisms are required to be utilised, significant impact to an operator’s network or services may still be experienced. In order to allow an operator to avoid such scenarios:

- o An implementation MAY provide functionality whereby all future Critical errors result in UPDATE messages being discarded. Such functionality MUST be disabled by default, and SHOULD be configurable on a per-address-family basis. An operator MUST

consider such mechanisms as a tool of last-resort to maintain service for a subset of NLRI, whilst the root cause of a such errors is investigated and resolved. This MAY be achieved by filtering erroneous NLRI at an upstream peer.

- o Provide means by which a the restart timer for Graceful Restart can be configured to be a long period (order of days, or weeks) such that a critical failure can be resolved whilst maintaining operation for a subset of NLRI. This restart period MUST be configured separately to standard graceful-restart timers and MUST be configurable per-address-family. Long-lived restart mechanisms MAY be configurable to be utilised by default. An operator MUST configure the impact to forwarding correctness of such configuration, based on the expected rate of change of NLRI within a particular <AFI,SAFI>.

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

The requirements outlined in this document provide mechanisms which limit the forwarding impact of the response to an error in a BGP UPDATE message. This is of benefit to the security of a BGP speaker. Without these mechanisms, where erroneous UPDATE messages relating to a single NLRI entry can be propagated to a BGP speaker, all other NLRI carried via the same session are affected by the resulting session tear-down. This may result in a means by which an AS can be isolated from particular routing domains (such as the Internet) should an UPDATE message be propagated via targeted specific paths. It is envisaged by reducing the impact of the reaction of the receiving speaker to these messages, the isolation can be constrained to specific sets of NLRI, or a specific topology.

A number of the mechanisms meeting the requirements specified within the document (particularly those relating to operational monitoring) may raise further security concerns. Such concerns will be addressed during the specification of these mechanisms.

8. Acknowledgements

Many thanks are extended to Bruno Decraene and David Freedman for their numerous detailed reviews, and significant contribution towards the refinement of the requirements in this document.

In addition, the author would like to thank the following network operators for their insight, and valuable input into defining the

requirements for a variety of deployments of BGP: Shane Amante, Colin Bookham, Rob Evans, Wes George, Tom Hodgson, Sven Huster, Jonathan Newton, Neil McRae, Thomas Mangin, Tom Scholl and Ilya Varlashkin. Many thanks are extended to Jeff Haas, Wim Hendrickx, Tony Li, Alton Lo, Keyur Patel, John Scudder, Adam Simpson and Robert Raszuk for their expertise relating to implementations of the BGP protocol.

9. References

9.1. Normative References

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

[RFC2858] Bates, T., Rekhter, Y., Chandra, R., and D. Katz, "Multiprotocol Extensions for BGP-4", RFC 2858, June 2000.

[RFC2918] Chen, E., "Route Refresh Capability for BGP-4", RFC 2918, September 2000.

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.

[RFC4724] Sangli, S., Chen, E., Fernando, R., Scudder, J., and Y. Rekhter, "Graceful Restart Mechanism for BGP", RFC 4724, January 2007.

[RFC4761] Kompella, K. and Y. Rekhter, "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, January 2007.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, June 2010.

9.2. Informational References

[I-D.chen-ebgp-error-handling]
Chen, E., Mohapatra, P., and K. Patel, "Revised Error Handling for BGP Updates from External Neighbors", draft-chen-ebgp-error-handling-01 (work in progress), September 2011.

[I-D.ietf-grow-bgp-gshut]
Francois, P., Decraene, B., Pelsser, C., Patel, K., and C. Filsfils, "Graceful BGP session shutdown", draft-ietf-grow-bgp-gshut-06 (work in progress), August 2014.

[I-D.ietf-grow-bmp]
Scudder, J., Fernando, R., and S. Stuart, "BGP Monitoring Protocol", draft-ietf-grow-bmp-07 (work in progress), October 2012.

[I-D.ietf-idr-bgp-enhanced-route-refresh]
Patel, K., Chen, E., and B. Venkatachalapathy, "Enhanced Route Refresh Capability for BGP-4", draft-ietf-idr-bgp-enhanced-route-refresh-10 (work in progress), June 2014.

[I-D.ietf-idr-operational-message]
Freedman, D., Raszuk, R., and R. Shakir, "BGP OPERATIONAL Message", draft-ietf-idr-operational-message-00 (work in progress), March 2012.

Author’s Address

Rob Shakir
BT plc.
pp. C3L,
BT Centre,
81, Newgate Street,
London. EC1A 7AJ
UK

Email: rob.shakir@bt.com
URI: http://www.bt.com/

Global Routing Operations
Internet-Draft
Intended status: Informational
Expires: September 10, 2015

K. Sriram
D. Montgomery
US NIST
D. McPherson
E. Osterweil
Verisign, Inc.
March 9, 2015

Problem Definition and Classification of BGP Route Leaks
draft-ietf-grow-route-leak-problem-definition-01

Abstract

A systemic vulnerability of the Border Gateway Protocol routing system, known as ’route leaks’, has received significant attention in recent years. Frequent incidents that result in significant disruptions to Internet routing are labeled "route leaks", but to date we have lacked a common definition of the term. In this document, we provide a working definition of route leaks, keeping in mind the real occurrences that have received significant attention. Further, we attempt to enumerate (though not exhaustively) different types of route leaks based on observed events on the Internet. We aim to provide a taxonomy that covers several forms of route leaks that have been observed and are of concern to Internet user community as well as the network operator community.

Status of This Memo

Table of Contents

1. Introduction

Frequent incidents [Huston2012][Cowie2013][Cowie2010][Madory][Zmijewski][Paseka][LRL][Khare] that result in significant disruptions to Internet routing are commonly called "route leaks". Examination of the details of some of these incidents reveals that they vary in their form and technical details. Before we can discuss solutions to "the route leak problem" we need a clear, technical definition of the problem and its most common forms. In Section 2, we provide a working definition of route leaks, keeping in view many recent incidents that have received significant attention. Further, in Section 3, we attempt to enumerate (though not exhaustively) different types of route leaks based on observed events on the Internet. We aim to provide a taxonomy that covers several forms of route leaks that have been observed and are of concern to Internet user community as well as the network operator community.

2. Working Definition of Route Leaks

A proposed working definition of route leak is as follows:

A "route leak" is the propagation of routing announcement(s) beyond their intended scope. That is, an AS’s announcement of a learned BGP route to another AS is in violation of the intended policies of the receiver, the sender and/or one of the ASes along the preceding AS path. The intended scope is usually defined by a set of local redistribution/filtering policies distributed among the ASes involved. Often, these intended policies are defined in terms of the pair-wise peering business relationship between ASes (e.g., customer, provider, peer). For literature related to AS relationships and routing policies, see [Gao][Gill][Luckie]. For measurements of valley-free violations in Internet routing, see [Giotsas][Wijchers].

The result of a route leak can be redirection of traffic through an unintended path which may enable eavesdropping or traffic analysis, and may or may not result in an overload or black-hole. Route leaks can be accidental or malicious, but most often arise from accidental misconfigurations.

The above definition is not intended to be all encompassing. Perceptions vary widely about what constitutes a route leak. Our aim here is to have a working definition that fits enough observed incidents so that the IETF community has a basis for starting to work on route leak mitigation methods.

3. Classification of Route Leaks Based on Documented Events

As illustrated in Figure 1, a common form of route leak occurs when a multi-homed customer AS (such as AS1 in Figure 1) learns a prefix update from one provider (ISP1) and leaks the update to another provider (ISP2) in violation of intended routing policies, and further the second provider does not detect the leak and propagates the leaked update to its customers, peers, and transit ISPs.

```
                                     /\              /\
                                      \ route-leak(P)/
                                       \ propagated /
                                        \          /
             +------------+    peer    +------------+
      ______| ISP1 (AS2) |----------->|  ISP2 (AS3)|---------->
     /       ------------+  prefix(P) +------------+ route-leak(P)
    | prefix |          \   update      /\        \  propagated
     \  (P)  /           \              /          \
      -------   prefix(P) \            /            \
                   update  \          /              \
                            \        /route-leak(P)  \/
                             \/     /
                        +---------------+
                        | customer(AS1) |
                        +---------------+
```

Figure 1: Illustration of the basic notion of a route leak.

We propose the following taxonomy for classification of route leaks aiming to cover several types of recently observed route leaks, while acknowledging that the list is not meant to be exhaustive. In what follows, we refer to the AS that announces a route that is in violation of the intended policies as the "offending AS".

- o Type 1 "U-Turn with Full Prefix": A multi-homed AS learns a prefix route from one upstream ISP and simply propagates the prefix to another upstream ISP. Neither the prefix nor the AS path in the update is altered. This is similar to a straight forward path-poisoning attack [Kapela-Pilosov], but with full prefix. It should be noted that attacks or leaks of this type are often accidental (i.e. not malicious). The update basically makes a U-turn at the attacker’s multi-homed AS. The attack (accidental or deliberate) often succeeds because the second ISP prefers customer announcement over peer announcement of the same prefix. Data packets would reach the legitimate destination albeit via the offending AS, unless they are dropped at the offending AS due to its inability to handle resulting large volumes of traffic.

  - * Example incidents: Examples of Type 1 route-leak incidents are (1) the Dodo-Telstra incident in March 2012 [Huston2012], (2) the Moratel-PCCW leak of Google prefixes in November 2012 [Paseka], and (3) the VolumeDrive-Atrato incident in September 2014 [Madory].

- o Type 2 "U-Turn with More Specific Prefix": A multi-homed AS learns a prefix route from one upstream ISP and announces a sub-prefix

(subsumed in the prefix) to another upstream ISP. The AS path in the update is not altered. Update is crafted by the attacker to have a subprefix to maximize the success of the attack while reverse path is kept open by the path poisoning techniques as in [Kapela-Pilosov]. Data packets reach the legitimate destination albeit via the offending AS.

* Example incidents: An example of Type 2 route-leak incident is the demo performed at DEFCON-16 in August 2008 [Kapela-Pilosov]. An attacker who deliberately performs a Type 1 route leak (with full prefix) can just as easily perform a Type 2 route leak (with subprefix) to achieve a greater impact.

o Type 3 "Prefix Hijack with Data Path to Legitimate Origin": A multi-homed AS learns a prefix route from one upstream ISP and announces the prefix to another upstream ISP as if it is being originated by it (i.e. strips the received AS path, and re-originates the prefix). This amounts to straightforward hijacking. However, somehow (not attributable to the use of path poisoning trick by the attacker) a reverse path is present, and data packets reach the legitimate destination albeit via the offending AS. But sometimes the reverse path may not be there, and data packets get dropped following receipt by the offending AS.

* Example incidents: Examples of Type 3 route leak include (1) the China Telecom incident in April 2010 [Hiran][Cowie2010][Labovitz], (2) the Belarusian GlobalOneBel route leak incidents in February-March 2013 and May 2013 [Cowie2013], (3) the Icelandic Opin Kerfi-Simmin route leak incidents in July-August 2013 [Cowie2013], and (4) the Indosat route leak incident in April 2014 [Zmijewski].

o Type 4 "Leak of Internal Prefixes and Accidental Deaggregation": An offending AS simply leaks its internal prefixes to one or more of its transit ASes and/or ISP peers. The leaked internal prefixes are often deaggregated subprefixes (i.e. more specifics) of already announced aggregate prefixes. Further, the AS receiving those leaks fails to filter them. Typically these leaked announcements are due to some transient failures within the AS; they are short-lived, and typically withdrawn quickly following the announcements.

* Example incidents: Leaks of internal prefix-routes occur frequently (e.g. multiple times in a week), and the number of prefixes leaked range from hundreds to thousands per incident. One highly conspicuous and widely disruptive leak of internal prefixes happened recently in August 2014 when AS701 and AS705

leaked about 22,000 more specifics of already announced aggregates [Huston2014][Toonk].

- o Type 5 "Lateral ISP-ISP-ISP Leak": This type of route leak typically occurs when, for example, three sequential ISP peers (e.g. ISP-A, ISP-B and ISP-C) are involved, and ISP-B receives a prefix-route from ISP-A and in turn leaks it to ISP-C. The typical routing policy between laterally (i.e. non-hierarchically) peering ISPs is that they should only propagate to each other their respective customer prefixes.

  - * Example incidents: In [Mauch-nanog][Mauch], route leaks of this type are reported by monitoring updates in the global BGP system and finding three or more very large ISP ASNs in a sequence in a BGP update’s AS path. Mauch [Mauch] observes that these are anomalies and potentially route leaks because very large ISPs such as ATT, Sprint, Verizon, and Globalcrossing do not in general buy transit services from each other. However, he also notes that there are exceptions when one very large ISP does indeed buy transit from another very large ISP, and accordingly exceptions are made in his detection algorithm for known cases.

- o Type 6 "Leak of Provider Prefixes to Peer": This type of route leak occurs when an offending AS leaks prefix-routes learned from its provider to a lateral peer.

  - * Example incidents: The incidents reported in [Mauch] include the Type 6 leaks.

- o Type 7 "Leak of Peer Prefixes to Provider": This type of route leak occurs when an offending AS leaks prefix-routes learned from a lateral peer to its (the AS’s) own provider. These leaked prefix-routes typically originate from the customer cone of the lateral peer.

  - * Example incidents: Some of the example incidents cited for Type 1 route leaks above are also inclusive of Type 7 route leaks. For instance, in the Dodo-Telstra incident [Huston2012], the leaked routes from Dodo to Telstra included routes that Dodo learned from its providers as well as lateral peers.

4. Summary

We attempted to provide a working definition of route leak. We also presented a taxonomy for categorizing route leaks. It covers not all but at least several forms of route leaks that have been observed and are of concern to Internet user and network operator communities. We

hope that this work provides the IETF community a basis for pursuing possible BGP enhancements for route leak detection and mitigation.

5. Security Considerations

No security considerations apply since this is a problem definition document.

6. IANA Considerations

No updates to the registries are suggested by this document.

7. Acknowledgements

The authors wish to thank Danny McPherson and Eric Osterweil for discussions related to this work. Also, thanks are due to Jared Mauch, Jeff Haas, Warren Kumari, Brian Dickson, Amogh Dhamdhere, Jakob Heitz, Geoff Huston, Randy Bush, Ruediger Volk, Andrei Robachevsky, Chris Morrow, and Sandy Murphy for comments, suggestions, and critique. The authors are also thankful to Padma Krishnaswamy, Oliver Borchert, and Okhee Kim for their comments and review.

8. Informative References

[Cowie2010]
Cowie, J., "China’s 18 Minute Mystery", Dyn Research/ Renesys Blog, November 2010, <http://research.dyn.com/2010/11/ chinas-18-minute-mystery/>.

[Cowie2013]
Cowie, J., "The New Threat: Targeted Internet Traffic Misdirection", Dyn Research/Renesys Blog, November 2013, <http://research.dyn.com/2013/11/ mitm-internet-hijacking/>.

[Gao]
Gao, L. and J. Rexford, "Stable Internet routing without global coordination", IEEE/ACM Transactions on Networking, December 2001, <http://www.cs.princeton.edu/~jrex/papers/ sigmetrics00.long.pdf>.

[Gill]
Gill, P., Schapira, M., and S. Goldberg, "A Survey of Interdomain Routing Policies", ACM SIGCOMM Computer Communication Review, January 2014, <https://www.cs.bu.edu/~goldbe/papers/survey.pdf>.

[Giotsas] Giotsas, V. and S. Zhou, "Valley-free violation in Internet routing - Analysis based on BGP Community data", IEEE ICC 2012, June 2012, <http://www0.cs.ucl.ac.uk/staff/V.Giotsas/files/giotsas.icc.2012.pdf>.

[Hiran] Hiran, R., Carlsson, N., and P. Gill, "Characterizing Large-scale Routing Anomalies: A Case Study of the China Telecom Incident", PAM 2013, March 2013, <http://www3.cs.stonybrook.edu/~phillipa/papers/CTelecom.html>.

[Huston2012] Huston, G., "Leaking Routes", March 2012, <http://labs.apnic.net/blabs/?p=139/>.

[Huston2014] Huston, G., "What’s so special about 512?", September 2014, <http://labs.apnic.net/blabs/?p=520/>.

[Kapela-Pilosov] Pilosov, A. and T. Kapela, "Stealing the Internet: An Internet-Scale Man in the Middle Attack", DEFCON-16 Las Vegas, NV, USA, August 2008, <https://www.defcon.org/images/defcon-16/dc16-presentations/defcon-16-pilosov-kapela.pdf/>.

[Khare] Khare, V., Ju, Q., and B. Zhang, "Concurrent Prefix Hijacks: Occurrence and Impacts", IMC 2012, Boston, MA, November 2012, <http://www.cs.arizona.edu/~bzhang/paper/12-imc-hijack.pdf/>.

[LRL] Khare, V., Ju, Q., and B. Zhang, "Large Route Leaks", Project web page, 2012, <http://nrl.cs.arizona.edu/projects/lsrl-events-from-2003-to-2009/>.

[Labovitz] Labovitz, C., "Additional Discussion of the April China BGP Hijack Inciden", Arbor Networks IT Security Blog, November 2010, <http://www.arbornetworks.com/asert/2010/11/additional-discussion-of-the-april-china-bgp-hijack-incident/>.

[Luckie] Luckie, M., Huffaker, B., Dhamdhere, A., Giotsas, V., and kc. claffy, "AS Relationships, Customer Cones, and Validation", IMC 2013, October 2013, <http://www.caida.org/~amogh/papers/asrank-IMC13.pdf>.

[Madory] Madory, D., "Why Far-Flung Parts of the Internet Broke Today", Dyn Research/Renesys Blog, September 2014, <http://research.dyn.com/2014/09/why-the-internet-broke-today/>.

[Mauch] Mauch, J., "BGP Routing Leak Detection System", Project web page, 2014, <http://puck.nether.net/bgp/leakinfo.cgi/>.

[Mauch-nanog] Mauch, J., "Detecting Routing Leaks by Counting", NANOG-41 Albuquerque, NM, USA, October 2007, <https://www.nanog.org/meetings/nanog41/presentations/mauch-lightning.pdf/>.

[Paseka] Paseka, T., "Why Google Went Offline Today and a Bit about How the Internet Works", CloudFare Blog, November 2012, <http://blog.cloudflare.com/why-google-went-offline-today-and-a-bit-about/>.

[Toonk] Toonk, A., "What Caused Today’s Internet Hiccup", August 2014, <http://www.bgpmon.net/what-caused-todays-internet-hiccup/>.

[Wijchers] Wijchers, B. and B. Overeinder, "Quantitative Analysis of BGP Route Leaks", RIPE-69, November 2014, <https://ripe69.ripe.net/presentations/157-RIPE-69-Routing-WG.pdf>.

[Zmijewski] Zmijewski, E., "Indonesia Hijacks the World", Dyn Research/Renesys Blog, April 2014, <http://research.dyn.com/2014/04/indonesia-hijacks-world/>.

Authors’ Addresses

Kotikalapudi Sriram
US NIST

Email: ksriram@nist.gov

Doug Montgomery
US NIST

Email: dougm@nist.gov

Danny McPherson
Verisign, Inc.

Email: dmcpherson@verisign.com

Eric Osterweil
Verisign, Inc.

Email: eosterweil@verisign.com

Interdomain Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 24, 2015

S. Litkowski
Orange Business Service
K. Patel
Cisco Systems
J. Haas
Juniper Networks
March 23, 2015

Timestamp support for BGP paths
draft-litkowski-idr-bgp-timestamp-02

Abstract

BGP is more and more used to transport routing information for critical services. Some BGP updates may be critical to be received as fast as possible : for example, in a layer 3 VPN scenario where a dual-attached site is loosing primary connection, the BGP withdraw message should be propagated as fast as possible to restore the service. The same criticity exists for other address-families like multicast VPNs where "join" messages should also be propagated very fast.

Experience of service providers shows that BGP path propagation time may vary depending on network conditions (especially load of BGP speaker on the path) and too long propagation time are affecting customer service.

It is important for service providers to keep track of BGP updates propagation time to monitor quality of service for the customers. It is also important to be able to identify BGP Speakers that are slowing down the propagation.

This document presents a solution to transport timestamps of a BGP path. The solution is targeted to be used using special identified beacon prefixes that are single-homed.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Status of This Memo

Copyright Notice

Table of Contents

1. Problem statement

```
CE3----PE3               PE4 --- CE4 (Source)
          \              /
           RR3       RR4
              \     /
                RR5
               /    \
            RR1     RR2
           / |         \
          /  |          \
CE1----PE1  PE5          PE2 --- CE2
             |
             CE5
```

Figure 1

The figure 1 describes a typical hierarchical RR design where PEs are meshed to local RRs and local RRs are meshed to more centric RRs. We consider a single multicast VPN between all CEs. CE4 is the source, all others may be receivers. The BGP controlplane also supports some other BGP service like L3VPN service.

We consider an event in L3VPN service leading to RR1 being temporarily overloaded (for example, RR1 is processing massive updates due to a router failure or formatting updates for a route-refresh). In the same timeframe, CE1 wants to join the multicast flow from CE4. PE1 propagates the C-multicast route to RR1, but RR1 fails to propagate the route to RR5 because it is busy processing L3VPN. When RR1 finishes the L3VPN job, it would send the C-multicast route to RR5 and updates would be imported by PE4. The

long time to join the flow may cause CE4 to miss part of the multicast flow.

All BGP implementations are different in term of internal processing within an address family or between address family. The issue described above is just given as an example, and the document does not presume that all implementations are suffering from this exact issue. But whatever the implementation, their always be cases where BGP path propagation could be delayed.

Service providers currently lack of efficient solution to keep track of BGP path propagation time as well as solution to identify the BGP speakers causing issues.

BMP (BGP Monitoring Protocol) may be a solution but as several drawbacks (see Section 6).

2. Requirements for monitoring BGP path propagation time

2.1. Architecture

```
                    ---------             -------
                   /         \          /       \
    RTR_SRC1 ----- |   AS1     | -----  |    AS2   |  ---- RTR_DST1
        |          \         /          \       /           |
      Inject         ---------             ---------          Sink point
       point             |                     |
                         |                     |
                    ---------             -------
                   /         \          /       \
    RTR_DST2 ---- |    AS4     |        |    AS3   |  ---- RTR_SRC2_DST2
        |          \         /          \       /             |
      Sink point    ---------             ---------         Inject/Sink
                                                              point
```

Figure 2

```
                    Single AS
      -------------------------------------------
     /                                           \
    |          RR1 ---------- RR2                 |
    |         /   \               \               |
    | RTR_SRC1     \               RTR_DST2       |
    |    |          \                  |          |
    |   Inject      RR3               Sink point  |
    |     point      |                            |
    |               RTR_DST1                     |
    |                |                            |
     \              Sink point                   /
       -------------------------------------------
```

Figure 3

Figure 2 and Figure 3 describes an interAS and a single AS scenario where a service provider wants to monitor BGP path propagation time from a router to multiple routers. In Figure 2, multiple probing routers are attached to multiple ASes. In Figure 3, all probing routers are in the same AS.

The architecture requires some BGP Speaker to originate some NLRI within the BGP controlplane. In the diagram above, they are identified as "Inject point". In order to provide information about propagation delays, the architecture requires introduction of timestamp information. Architecture also needs to identify BGP Speaker causing high propagation delays. As only, specific advertisement will serve for measurement, the architecture requires BGP Speaker to identify NLRIs that must be timestamped. The architecture also requires some BGP Speaker to serve as sink point where a timestamp vector information can be retrieved. The timestamp vector must contain propagation time information for all BGP Speaker that participated in the BGP path. It is so required that each BGP Speaker along the path to add timestamp information. There may be multiple sink points in the network to perform measurement at different location and also different inject points. An external tool may be connected to Sink Points to retrieve the timestamp information. But this is out of scope of the document.

In case of interAS, for security reason, the architecture MUST support hiding detailed timestamp information to the other AS.

Example of usage :

An external tool should command RTR_SRC to originate a probing BGP NLRI. All the BGP Speakers are configured to measure timestamp for this NLRI. The BGP path would propagate across BGP Speakers. Each BGP Speaker may provide timestamp informations. An external tool

connected to sink points will retrieve timestamp vector information for the NLRI.

2.2. Measurement accuracy

2.2.1. Clock synchronization

For the solution to be accurate, it is mandatory for BGP Speaker to be synchronized. This could be ensured easily within a single AS but in a inter domain scenario, it is hard to ensure that all Speakers are synchronized to a good clock source.

The solution MUST include synchronization information associated with the timestamp in order to be able to compare timestamps between them.

2.2.2. Beacon accuracy

In order to be accurate, an implementation SHOULD :

- o ensure that the timestamped NLRIs are processed with the same priority as non timestamped NLRIs.

- o ensure that the processing of adding timestamp information is as lightweight as possible. If some limitation exists, the vendor SHOULD document them.

Using a unique special prefix advertisement from a single location to evaluate propagation time will not provide a detail view of min/max propagation time values as the user will not know where the path for the prefix may be located in a processing queue. Considering a BGP Speaker handling high churn, the advertisement of the path for the special prefix may have a specific place in the long processing queue of the churn depending on the implementation : it may be first, last or somewhere in the middle.

It is required from user to perform sampling to establish propagation time boundaries based on multiple advertisements. Repeated operations of advertisement then withdraw may help in this. See Section 7 for more details.

2.3. Churn

The target solution MUST NOT create more churn in the BGP controlplane.

2.4. Path propagation complexity

When a NLRI is originated in BGP from a point, a BGP path is created. Nothing ensures that all nodes within the BGP controlplane will receive this BGP path. When a concurrent path already exists from the NLRI, the concurrent path may be prefered by some BGP Speaker leading to hiding of the new path. Moreover, even if the NLRI is originated in BGP from a single point, multiple paths may be created within the BGP controlplane, this is inherent to the BGP meshing in place.

As soon as multiple BGP paths are involved, controlplane convergence may be done in multiple steps in order to find the final best path. This convergence may involve multiple BGP path advertisement (replacing each other) between peers.

The goal of our proposal is not to measure the convergence time but to focus on the path propagation time. In a controlplane convergence involving multiple paths for a NLRI, the solution MUST identify timestamp for the event where the NLRI was seen for the first time on a BGP Speaker.

Example :

```
                  Single AS
    -------------------------------------------
   /                             RTR_SRC2- 10/8 \
  |                             /                 |
  |          RR1 ---------- RR2                   |
  |         /   \               \                 |
  | RTR_SRC1     \               RTR_DST2         |
  |     |         \                               |
  |   10/8        RR3                             |
  |                |                              |
  |               RTR_DST1                        |
  |                                               |
   \                                             /
     -------------------------------------------
```

Figure 4

In the figure above, consider that the service provider is keep tracking of propagation time for real NLRIs (corresponding to customer routes). All the BGP Speakers in our figure are configured to inspect the NLRI 10/8 which is multihomed. We consider that the network is starting and the NLRI has not been propagated yet.

RTR_SRC1 starts to propagate 10/8 within the BGP controlplane. All BGP Speakers considers the path as best and this path will be propagated within the whole controlplane. Each BGP Speaker would add its timestamp information and RTR_DST1 and RTR_DST2 would be able to record the timestamp vector. In this case, the timestamp vector is quite accurate because it represents an end to end propagation.

Now RTR_SRC2 starts to propagate its own path. RR2 has two paths for 10/8 and will choose the best one, let’s consider that RTR_SRC2 path is the best one, RTR_SRC2 path will so be propagated and timestamp vector will be updated. RR1 will also have two paths, and we consider that RR1 prefers RTR_SRC1 path, so RTR_SRC2 path will not be propagated by RR1. In this situation, RTR_DST2 will receive the path from RR2 with accurate timestamp (end to end propagation) but RTR_DST1 will never receive it.

We could also consider a stable network situation, where both paths have been advertised for a long time. A network event may occur (e.g. IGP metric change) that would cause a BGP Speaker within a path vector to change its best path. In Figure 10, an IGP event, may cause RR1 to change its decision and prefers the path originated by RTR_SRC2 as best, the path will be propagated with previous received timestamp information that are no more accurate. RTR_DST1 will receive a BGP timestamp vector containing stale (old) timestamp informations as well as new ones.

3. Proposal

Our proposal is based on tagging NLRI with timestamp values along its BGP path propagation. Each BGP Speaker along the path will add timestamp values, so creating a timestamp vector. An ordered list of timestamps would so be built along the path.

```
   BGP Update      BGP Update       BGP Update      BGP Update
   10.0.0.0/8      10.0.0.0/8       10.0.0.0/8      10.0.0.0/8
   Timestamp:      Timestamp:       Timestamp:      Timestamp:
   R1:T1           R1:T1            R1:T1           R1:T1
                   R2:T2            R2:T2           R2:T2
                                    R3:T3           R3:T3
                                                    R4:T4
  R1 ------------> R2 ------------> R3 ------------> R4 ------------> R5
```

Using this mechanism, we can easily identify if a hop within a path is slowing down the propagation.

We propose to use a new BGP attribute, BGP timestamp attribute to encode timestamps information.

4. BGP timestamp attribute

The BGP timestamp (BGP-TS) Attribute is an optional transitive BGP Path Attribute. The attribute type code is TBD.

The value field of the BGP timestamp attribute is defined as an ordered list of timestamp entries, the first entry being the first timestamp entry added (origin):

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Timestamp #1  (variable)                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Timestamp #2  (variable)                       |
...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                Timestamp #n  (variable)                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The timestamps entries are encoded as follows :

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Receive Timestamp #x                          |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Send Timestamp #x                             |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 ASN                                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|T|   Rsvd      |   SyncType    |   EntryType   |               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
|                                                               |
|                        Optional variable field                |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- o Receive timestamp : the time at which the BGP path was received. When originating a path in BGP, the timestamp is the originating time. Expressed in seconds and microseconds since midnight (zero hour), January 1, 1970 (UTC). If zero, the time is unavailable. Precision of the timestamp is implementation- dependent.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Timestamp (seconds)                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Timestamp (microseconds)                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- o Send timestamp : the time at which the BGP path was exported to the peer. Expressed in seconds and microseconds since midnight (zero hour), January 1, 1970 (UTC). If zero, the time is unavailable. Precision of the timestamp is implementation-dependent.

- o ASN : AS Number of the local node creating the timestamp entry.

- o Flags :

  - * T : Synchronized, if set, the BGP speaker clock is synchronized to an external system.

- o SyncType : defines the stratum as defined in [RFC5905].

- o EntryType : defines the type of Timestamp entry, the following types are defined :

  - * Type 0 : empty. There is no following variable field. This type is to be used in case of timestamp summarization.

  - * Type 1 : IPv4 address, the following variable field will be 4 bytes long and will contain the IPv4 router ID of the local node.

  - * Type 2 : IPv6 address, the following variable field will be 16 bytes long and will contain the IPv6 router ID of the local node.

  - * Type 3 : Stale Indicator, Stale indicates that previous timestamp entries are old. There is no following variable field. The receive timestamp and send timestamp should be set to zero. The ASN is set to the ASN of the local BGP Speaker.

5. Processing the BGP timestamp attribute

5.1. Inspection list

A BGP Speaker supporting the BGP-TS can decide to timestamp only some specific NLRIs. An inspection list may be configured by the user (filter) to apply timestamping on a specific set of BGP NLRIs. By

default, we suggest that a BGP Speaker supporting BGP-TS SHOULD NOT timestamp any BGP NLRIs.

User of our proposal must be aware that using a complex policy to express inspection list may result in more processing that will influence the end to end propagation time. It is expected that the inspection list policy should be kept as simple as possible.

5.2. Originating a timestamped route in BGP

When a BGP Speaker supporting BGP-TS originates a new path in BGP that matches the inspection list, it MUST add the BGP-TS attribute to the BGP path and MUST set the receive timestamp field to the time the path was originated in BGP. At this time of processing, the send timestamp will be set to 0. If the BGP Speaker is synchronized to an external system when originating the route, the S-bit MUST be set in the attribute and the SyncType MUST be set to the current stratum. As mentioned above, the BGP path of the originated route will have a send timestamp value of zero in the BGP LOC-RIB.

5.3. Receiving a timestamped route in BGP

When a BGP Speaker supporting BGP-TS receives a BGP path that matches the inspection list, the implementation MUST record the current time associated with the received path.

The time recording MUST append before the inbound routing policies.

```
                            Inspection
                                      List
        +------------+      +---+    No match    +------------+
-->     | Adj-RIB-in | -->  | I | -------------> | Rtg pol in |
        | Peer#1     |      | n |                | Peers#1    | ----->
        +------------+      | s |     +-------+  |            |
                            | p | --> | AddTS |->|            |
                            | e |     +-------+  +------------+
                            | c |   If match
                            | t |
                            |   |
                            | l |
        +------------+      | i |    No match    +------------+
-->     | Adj-RIB-in | -->  | s | -------------> | Rtg pol in |
        | Peer#2     |      | t |                | Peers#2    | ----->
        +------------+      |   |     +-------+  |            |
                            |   | --> | AddTS |->|            |
                            |   |     +-------+  +------------+
                            |   |   If match
                            +---+
```

If the path that matches the inspection list and does not contains a BGP-TS attribute, it MUST add a BGP-TS attribute with a timestamp entry :

- o The receive timestamp MUST be set to the recorded time for this BGP path.

- o If the BGP Speaker is synchronized to an external system when receiving the route, the S-bit MUST be set in the attribute and the SyncType MUST be set to the current stratum.

- o The send timestamp MUST be set to zero.

If the path that matches the inspection list and contains a BGP-TS attribute, it MUST append a new timestamp entry in the existing attribute :

- o The receive timestamp MUST be set to the recorded time for this BGP path.

- o If the BGP Speaker is synchronized to an external system when receiving the route, the S-bit MUST be set in the attribute and the SyncType MUST be set to the current stratum.

- o The send timestamp MUST be set to zero.

The process of adding a timestamp entry or adding BGP-TS attribute SHOULD be as light as possible in order to influence the propagation time as lowest as possible.

When a BGP Speaker supporting BGP-TS receives a BGP path that does not the inspection list and contains a BGP-TS attribute, it MUST NOT change the existing attribute.

When a BGP Speaker not supporting BGP-TS receives a BGP path that contains a BGP-TS attribute, it MUST follow the standard BGP procedures described in [RFC4271].

5.4. Sending a timestamped route in BGP

5.4.1. Propagating the BGP Timestamp attribute

For a manageability/security purpose, the authors suggest that BGP timestamp attribute MAY NOT be sent to a peer unless it was explicitly configured for. This would prevent timestamp and internal address informations to be propagated to some external peers for example. See Section 5.7 for more information.

If a BGP path containing a BGP-TS attribute must be sent to be peer not configured with BGP timestamp option, the BGP-TS attribute should be dropped when the update message is sent to the peer.

5.4.2. Setting the send timestamp

If sending timestamp attribute is authorized for a specific peer, and path has a BGP-TS attribute, the outgoing BGP processing MUST fill the send timestamp field when exporting the path to a peer. The time recording MUST occur after all BGP filtering policies (outgoing routing policies, ORF, ...) and after placing path in Adj-RIB-Out. An implementation SHOULD set timestamp at the nearest possible step before sending the BGP Update to the peer. Depending of the implementation, the timestamping may occur at different stage of the outgoing BGP processing. Each implementer SHOULD document their timestamping process in order to make users understand correctly timestamp values. As most of implementations are using the concept of peer-groups, in case, timestamp is set too early in the BGP outgoing processing, all peers within a group may have the same timestamp value. Implementation should avoid this.

The process of adding the send timestamp must be as light as possible in order to influence the propagation time as lowest as possible.

```
+------+
|      |     +--------+     +-----+     +---+   +-------+     No TS
|      | --> | Rtgpol | --> | ORF | --> |...|-->|Adj-RIB|-------------->
|      |     | Out    |     |P#1  |     |   |   |Out    |       Send to peer
|      |     | Peer#1 |     |     |     |   |   |Peer#1 |   +-----+
|      |     |        |     |     |     |   |   |       |-->|AddTS| --->
|      |     +--------+     +-----+     +---+   +-------+   +-----+
|      |                                                     TS present
| BGP  |
| LOC  |
| RIB  |
|      |
|      |     +--------+     +-----+     +---+   +-------+     No TS
|      | --> | Rtgpol | --> | ORF | --> |...|-->|Adj-RIB|-------------->
|      |     | Out    |     |P#2  |     |   |   |Out    |       Send to peer
|      |     | Peer#2 |     |     |     |   |   |Peer#2 |   +-----+
|      |     |        |     |     |     |   |   |       |-->|AddTS| --->
|      |     +--------+     +-----+     +---+   +-------+   +-----+
|      |                                                     TS present
+------+
```

5.5. Limiting churn

Adding timestamp informations to BGP path will make all received paths to be unique.

```
            RR1
          /    \
10/8 - R1        RR3 --- R3
          \    /
            RR2
```

In the figure above, we consider that RR1 and RR2 are part of the same cluster (cluster ID : 1). RR3 is client of RR1 and RR2. R3 is client from RR3, R1 is client from RR1 and RR2.

Without BGP timestamp, when R1 originates the BGP prefix 10/8, it sends it to RR1 and RR2. Consider that RR3 receives path from RR1 first, it will reflect it to R3. When it will receive the path from RR2, it may consider that path from RR2 is best (lowest router ID) but as BGP attributes of the path are exactly the same as for RR1 path, there is no need to send an update to R3.

With BGP timestamp, when R1 originates the BGP prefix 10/8, it sends it to RR1 and RR2. Consider that RR3 receives path from RR1 first,

it will reflect it to R3. When it will receive the path from RR2, it may consider that path from RR2 is best (lowest router ID) but as BGP attributes of the two paths are not more equal due to the timestamp difference, RR3 may need to advertise an update to R3.

In order to prevent introducing more churn, we propose to modify the behavior described in Section 9.2. of [RFC4271]. An implementation MUST NOT consider BGP-TS attribute when evaluating the need to send a new update. As the BGP-TS attribute is purely informational, even if BGP Speakers have a different view of the timestamp attribute, there will be no impact on routing.

Considering our example, when RR3 will receive the path from RR2, even if it considers RR2 path as best, it will not send an update to R3 as all the attributes, except BGP-TS are equal.

5.6. Marking stale entries

Section 2.4 describes some cases where advertised timestamp information is no more relevant because it is old and also requires identification of first propagation timestamps.

In order to do this, we propose to mark old entries by adding a Stale Indicator within the timestamp vector. The presence of Stale Indicator must be interpreted as all previous timestamp entries need to be considered as old and not considered as a first propagation.

BGP-TS attribute example :

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  +
|                Timestamp #1  (IPv4)                           |  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  | Old
|                Timestamp #2   (IPv4)                          |  | entries
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |
|                Timestamp #3   (IPv4)                          |  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  +
|                Timestamp #4   (Stale Indicator)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  +
|                Timestamp #5   (IPv4)                          |  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  | Usable
...                                                             ...entries
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |
|                Timestamp #n  (variable)                       |  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  +
```

Insertion of Stale Indicator in a BGP-TS attribute may happen in the following conditions :

- o A path is received from a peer containing BGP-TS attribute or originated locally, the path matches the inspection list, and the decision process does not select the path as best path. Then the Stale Indicator SHOULD be inserted after decision process happened.

- o A path is received from a peer containing BGP-TS attribute or originated locally, the path matches the inspection list, and the decision process does select the path as best path. The path is exported to peers and then the Stale Indicator MUST be inserted. The path MUST NOT be repropagated as per Section 5.5.

When inserting a Stale indicator, if a Stale Indicator already exists in the timestamp vector, the implement SHOULD remove it before adding the new one.

```
BGP Update                                                BGP Update
10/8                                                      10/8
NH R2                                                     NH=R1
ASP : 2                                                   ASP : 1,2
Origin IGP                                                Origin IGP
BGP-TS :                                                  BGP-TS :
 [TS_entry1:IPv4]                                          [TS_entry1:IPv4]
 [TS_entry2:IPv4]                                          [TS_entry2:IPv4]
 [TS_entry3:Stale]                                         [TS_entry3:Stale]
 [TS_entry4:IPv4]                                          [TS_entry4:IPv4]
 [TS_entry5:IPv4]                                          [TS_entry5:IPv4]
                                                           [TS_entry6:IPv4]

BGP                        BGP Speaker                            BGP Speaker
Speaker                    R1                                     R3
R2                       +---------------------------+
  ----------------->     |                           | ------------>
                         |   BGP Path                |
                         |   At reception            |
                         | +-----------------------+ |
                         | | 10/8, from R2         | |
                         | |   BGP-TS :            | |
                         | |     [TS_entry1:IPv4]  | |
                         | |     [TS_entry2:IPv4]  | |
                         | |     [TS_entry3:Stale] | |
                         | |     [TS_entry4:IPv4]  | |
                         | |     [TS_entry5:IPv4]  | |
                         | |     [TS_entry6:IPv4]<-| | New timestamp entry
                         | +-----------------------+ | created by R1
                         |                           |
                         |   BGP Path                |
                         |   after sending to peer   |
                         |   Stale state is added    |
                         | +-----------------------+ |
                         | | 10/8, from R2         | |
                         | |   BGP-TS :            | |
                         | |     [TS_entry1:IPv4]  | |
                         | |     [TS_entry2:IPv4]  | |
                         | |     [TS_entry4:IPv4]  | |
                         | |     [TS_entry5:IPv4]  | |
                         | |     [TS_entry6:IPv4]<-| | New timestamp entry
                         | |     [TS_entry7:Stale] | | created by R1
                         | +-----------------------+ |
                         +---------------------------+
```

In the example above, R2 sends a BGP path with some existing stale timestamps. When R1 receives the route, it creates a new timestamp

entry in the BGP-TS attribute. We consider that the decision process decides that the path is best, the path is exported with the new timestamp entry and old timestamps coming from R2. Then R1 will update its local path by removing the previous Stale Indicator and replace a new one at the latest position to mark that it is no more the first propagation.

```
                Single AS
     ----------------------------
    /               RTR_SRC2- 10/8 \
   |              /                  |
   |          RR1                    |
   |         /   \                   |
   | RTR_SRC1     \                  |
   |     |         \                 |
   |   10/8        RR3               |
   |                |                |
   |               RTR_DST1          |
    \                               /
      ----------------------------
```

In the figure above, we consider that all BGP Speaker apply timestamp for prefix 10/8. RTR_SRC1 originates 10/8 in BGP, the decision process will decide that the path is best. RTR_SRC1 will export path to RR1 and then it will add locally the Stale Indicator within the timestamp vector. The path exported does not have the Stale Indicator. RR1 will receive the path and add a timestamp entry, the path is considered as best, RR1 will export it to RTR_SRC2 and RR3 and then it will add a stale indicator. RR3 will proceed in the same way.

When RTR_SRC2 will originate a new path for 10/8, if this new path is best on RTR_SRC2, it will export the path to RR1 and then it will add locally the Stale Indicator to the path. When RR1 will receive the route :

- o If the path from RTR_SRC2 is best, RR1 will export the new path to RTR_SRC1 and RR3 and then will add Stale indicator to the path.If RTR_SRC2 fails after some time, RR1 will pick up RTR_SRC1 path as best, and will export it to RR3. RR3 will know that the received timestamp entries are stale thanks to the stale indicator.

- o If the path from RTR_SRC2 is not best, RR1 will add Stale indicator to the path. If RTR_SRC1 fails after some time, RR1 will pick up RTR_SRC2 path as best, and will export it to RR3. RR3 will know that the received timestamp entries are stale thanks to the stale indicator.

5.7. Inter-AS considerations

```
BGP update
10.0.0.0/8
TS:
 AS3;CE1:rT1,sT2
```

```
CE1--------->R1 ------------> R2 ------------> R3 ------------> R4 -------> CE2
            |                   |             |                   |
            |                   |             |                   |
        AS3                  AS1                                 AS2
  AS4
```

In the figure above, we consider that customer wants to monitor BGP updates propagation time between its two sites.

If AS1 and AS2 BGP Speakers does not support BGP-TS, the attribute will be transported transparently accross AS1 without any processing. CE2 will so receive the BGP path with only a single timestamp entry from CE1.

If AS1 and AS2 BGP Speakers does support BGP-TS, four different options are offered : drop, drop-as, summarize, propagate. It must be noted that using drop-as or summarize options may involve more processing and so may impact the end to end propagation time.

5.7.1. Drop option

If AS1 and/or AS2 BGP Speakers support BGP-TS, they may not want to expose any timestamp information between each other. If a service does not want to propagate timestamp information to external peers, it can decide to not activate the "timestamp" option on the peer configuration , as explained in Section 5.4.

```
   BGP update       BGP update            BGP update        BGP update
   10.0.0.0/8       10.0.0.0/8            10.0.0.0/8        10.0.0.0/8
   TS:              TS:                                     TS:
    AS3;CE1:rT1,sT2  AS3;CE1:rT1,sT2                         AS2;R3:rT5,sT6
                     AS1;R1:rT3,sT4

CE1------------->R1 -----------------> R2 ---------------> R3 ------------> R4
                 |                     | no TS            |                    |
                 |                     |                  |                    |
        AS3                  AS1                                     AS2
```

In the example above, CE1 is configured to send timestamp to R1, as well as R1 to R2. But R2 does not want to send timestamp to R3.

When sending BGP route for 10/8, CE1 adds timestamp attribute and a timestamp entry (AS3, entry type : IPv4=CE1_IP, receive timestamp = T1, send timestamp=T2). R1 receives the path, we suppose that the inspection list matches, so R1 adds a timestamp entry. When sending to R2, R1 will send the following information in its timestamp entry : AS1,entry type : IPv4=R1_IP, receive timestamp T3, send timestamp T4. As R2 is configured to not send timestamp information to R3, it will drop the BGP attribute when sending to R3.

5.7.2. Drop AS option

If AS1 and/or AS2 BGP Speakers support BGP-TS, they may not want to expose their timestamps or internal BGP topology to other ASes. If a service does not want to propagate local AS related timestamp information to external peers, it can decide to use the "drop-as" option towards the peer.

```
   BGP update       BGP update            BGP update         BGP update
   10.0.0.0/8       10.0.0.0/8            10.0.0.0/8         10.0.0.0/8
   TS:              TS:                   TS:                TS:
    AS3;CE1:rT1,sT2  AS3;CE1:rT1,sT2       AS3;CE1:rT1,sT2    AS3;CE1:rT1,sT2
                     AS1;R1:rT3,sT4                           AS2;R3:rT5,sT6

CE1------------->R1 -----------------> R2 ---------------> R3 ------------> R4
                 |                     | no TS            |                    |
                 |                     |                  |                    |
        AS3                  AS1                                     AS2
```

In the example above, CE1 is configured to send timestamp to R1, as well as R1 to R2. But R2 does not want to send AS1 internal timestamp to R3. "Drop-as" option is configured on R2 towards R3.

When sending BGP route for 10/8, CE1 adds timestamp attribute and a timestamp entry (AS3, entry type : IPv4=CE1_IP, receive timestamp = T1, send timestamp=T2). R1 receives the path, we suppose that the inspection list matches, so R1 adds a timestamp entry. When sending to R2, R1 will send the following information in its timestamp entry : AS1,entry type : IPv4=R1_IP, receive timestamp T3, send timestamp T4. As R2 is configured with "drop-as" option to R3, it will remove all timestamp entries where the ASN is equal to its autonomous system number and then send the update to R3.

5.7.3. Summary option

If AS1 and/or AS2 BGP Speakers support BGP-TS, they may want to offer timestamp service to their customers but they want to hide their internal topology. In order to achieve the expected behavior, AS1/AS2 can activate a timestamp summary option on the external peer.

```
BGP update       BGP update            BGP update          BGP update
10.0.0.0/8       10.0.0.0/8            10.0.0.0/8          10.0.0.0/8
TS:              TS:                   TS:                 TS :
 AS3;CE1:rT1,sT2  AS3;CE1:rT1,sT2       AS3;CE1:rT1,sT2     AS3;CE1:rT1,sT2
                  AS1;R1:rT3,sT4        AS1;rT3,sT5         AS1;rT3,sT5
                                                            AS2;R3,rT6,sT7

CE1------------->R1 -----------------> R2 ---------------> R3 ------------> R4
                 |                     | TS summary       |                |
                 |                     |                  |                |
        AS3                   AS1                                      AS2
```

When using summary option, the BGP-TS attribute is modified as follows when exporting the route :

- o All timestamp entries containing the local AS in AS field are removed.
- o A new timestamp entry is created and inserted in place of removed entries (n entries replaced by 1).
- o The new timestamp entry will use an entry type zero.
- o The new timestamp entry MUST have the S bit set.

- o The new timestamp entry MUST NOT have any EntryType.

- o The receive timestamp of the new timestamp entry is the receiving timestamp of the first timestamp entry that has been removed.

- o The send timestamp of the new timestamp entry will be added as usual.

In the example above, CE1 is configured to send timestamp to R1, as well as R1 to R2. But R2 wants summarize timestamp information to AS2.

When sending BGP route for 10/8, CE1 adds timestamp attribute and a timestamp entry (AS3, entry type : IPv4=CE1_IP, receive timestamp = T1, send timestamp=T2). R1 receives the path, we suppose that the inspection list matches, so R1 adds a timestamp entry. When sending to R2, R1 will send the following information in its timestamp entry : AS1,entry type : IPv4=R1_IP, receive timestamp T3, send timestamp T4. As R2 is configured with "summarize" option to R3, it will remove all timestamp entries where the ASN is equal to its autonomous system number and add a new timestamp entry with an entry type zero. The receive timestamp will be retrieved from R1 timestamp entry.

5.7.4. Propagate option

If AS1 and/or AS2 BGP Speakers support BGP-TS, they may want to offer timestamp service to their customers with a full view. This MUST be the default behavior when timestamp is activated on a peer.

```
BGP update       BGP update            BGP update          BGP update
10.0.0.0/8       10.0.0.0/8            10.0.0.0/8          10.0.0.0/8
TS:              TS:                   TS:                 TS :
 AS3;CE1:rT1,sT2  AS3;CE1:rT1,sT2       AS3;CE1:rT1,sT2     AS3;CE1:rT1,sT2
                  AS1;R1:rT3,sT4        AS1;R1:rT3,sT4      AS1;R1:rT3,sT4
                                        AS1;R2:rT5,sT6      AS1;R2,rT5,sT6
                                                            AS2;R3,rT6,sT7

CE1------------->R1 -----------------> R2 ---------------> R3 ------------> R4
                 |                     | TS propagate     |                    |
                 |                     |                  |                    |
        AS3                 AS1                                        AS2
```

5.8. Retrieving timestamp vector

Authors suggest to implementers to use a local wrapping buffer on each node and record entries in the buffer each time a BGP path is timestamped. An external tool should then retrieve timestamps information from sink points. How the information is retrieved is out of scope of the document but we can imagine using :

- o BMP from the external tool to the sink point.
- o NetConf get to retrieve wrapping buffer information.
- o SNMP get to retrieve wrapping buffer information.
- o CLI command to retrieve wrapping buffer information.

5.9. Handling malformed attribute

When receiving a BGP Update message containing a malformed BGP-TS attribute, an "attribute discard" action MUST be applied as defined in [I-D.ietf-idr-error-handling].

5.10. Impact on update packing

Introducing timestamps information will make update packing less efficient for the timestamps path. In the deployment we are targeting (Section 7), this is not considered as an issue. In the case where a site is generating a special prefix with path timestamped and others not timestamped, these prefixes will not be packed together, so two update messages will be generated. Even if two updates are generated, we do not consider, that the propagation time will be highly affected.

6. Compared to BMP

BMP (BGP Monitoring Protocol) [I-D.ietf-grow-bmp] is a solution to monitor BGP sessions and provides a convenient interface for obtaining route views. BMP is a complete suite of messages to exchange informations regarding a BGP session.

We can imagine to use BMP as a solution to monitor BGP update propagation time but there is multiple drawbacks associated with such solution :

- o BMP provides dump of all received BGP update (per peer). If we are interested only in probing BGP routes, a strong filtering of information may be needed in BMP messages.

- o BMP does not mandate timestamping of messages (as per [I-D.ietf-grow-bmp] Section 5) : "If the implementation is able to provide information about when routes were received, it MAY provide such information in the BMP timestamp field. Otherwise, the BMP timestamp field MUST be set to zero, indicating that time is not available."

- o BMP may provide (if implementation available) timestamps information only for a single router point of view. If we want to retrieve timestamps of all BGP Speakers on a path, a BMP session is required to all BGP speakers. Correlation (based on known design) is also required at the external tool to order timestamps from each BMP session.

- o If BMP provides timestamp information, it does not provide information on how the router clock is synchronized (free run, NTP, GPS ...).

- o BMP only provides Adj-RIB-in view and does not provide outgoing information.

Using BMP to monitor BGP update propagation may complexify the design of the monitor solution. But as mentioned in Section 1, BMP can be used on specific sink routers to retrieve BGP TS vector.

7. Deployment considerations

This solution is not intended to perform timestamp imposition on all BGP prefixes.

The deployment scenario we are targeting is really to monitor some specific single-homed NLRIs identified by the service provider (see Section 2 as an example).

These NLRIs may be advertised at some injection point in the network, and timestamp vector will be retrieved at some sink points. As pointed in Section 2.2.2 , multiple samples of measurement will be necessary in order to evaluate the propagation time.

These NLRIs should be single-homed in order to ensure an end to end propagation from injection point to sink point. A coordination between injection and sink points based on an external tool is necessary : once a NLRI to be monitored has been advertised, the tool would retrieve the timestamp vector from the sink point.

Service provider may use real prefixes (used for routing) or special prefixes (standard IP prefix but allocated for beaconing). In case of special prefix used, the tool can at regular interval command the

advertisement and withdrawal of the prefix. The tool must ensure that it has retrieved the timestamp vector before withdrawing the prefix and also wait for convergence after withdrawal before advertising back the prefix.

The inspection list should be kept as small as possible by users in order to not introduce processing overhead and as a consequence slow down propagation.

8. Security considerations

Depending of the implementation and router capacity, adding timestamps to BGP path may consume some router resources. As proposed in Section 5.1, by default a BGP Speaker will not timestamp any path and inspection list should be configured to activate timestamping on a subset of paths. Using this approach, we consider that overhead that may be introduced by timestamping BGP paths is well controlled by operators. An external router cannot force an internal router to timestamp.

Providing detailed timestamps information to other ASes may introduce security issues by exposing internal datas (part of BGP topology, IP addresses, internal performance) to external entities. The proposal we make in Section 5.7 solves this security issue by giving flexibility to operators on the level of information he wants to expose to external peers.

9. Acknowledgements

10. IANA Considerations

IANA shall assign a codepoint for the BGP Timestamp attribute. This codepoint will come from the "BGP Path Attributes" registry.

11. Normative References

[I-D.ietf-grow-bmp]
Scudder, J., Fernando, R., and S. Stuart, "BGP Monitoring Protocol", draft-ietf-grow-bmp-07 (work in progress), October 2012.

[I-D.ietf-idr-error-handling]
Chen, E., Scudder, J., Mohapatra, P., and K. Patel, "Revised Error Handling for BGP UPDATE Messages", draft-ietf-idr-error-handling-18 (work in progress), December 2014.

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

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC5905] Mills, D., Martin, J., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.

Authors' Addresses

Stephane Litkowski
Orange Business Service

Email: stephane.litkowski@orange.com

Keyur Patel
Cisco Systems

Email: keyupate@cisco.com

Jeff Haas
Juniper Networks

Email: jhaas@juniper.net