A BGP Inter-AS Cost Attribute
draft-van-beijnum-idr-iac-00

Abstract

Although BGP implementations have extensive path selection algorithms, in practice operators have trouble performing satisfactory traffic engineering of incoming traffic based on BGP attributes that are taken into account in the path selection algorithm alone. For this reason, many ASes deaggregate their address range(s) into smaller blocks and announce these blocks differently to different neighboring ASes in order to arrive at the desired traffic flow. This practice contributes to the growth of the global routing table, which drives up capital expenditures for
networks engaging in inter-domain routing. This memo introduces a new inter-domain metric that supports finer-grained traffic engineering than current BGP attributes, most notably, the AS path.

1. Introduction

It’s easy to see how comparing AS paths lengths is problematic in today’s flat AS hierarchy. Assume 10 tier-1 ISPs that can reach all destinations connected to the internet through peering, and assume that the local AS buys transit service from two tier-1 ISPs. The traffic to the customers of those ISPs will normally flow through the respective ISP. However, for all destinations reachable over the 8 other tier-1s, the AS paths will have the same length over both transit ISPs. This means that prepending the AS path towards one ISP has a very dramatic effect: as much as 80% of all traffic may subsequently flow over the non-prepended ISP. With a finer-grained value that is communicated across ASes this problem would be reduced.

2. IAC and IAClocal

The new metric is named Inter-AS Cost (IAC). The content of the IAC is a 16-bit signed value that represents the cost or distance towards the source of the associated prefix. The intent of this document is that BGP implementations that support the IAC compare the IAC rather than the AS path length as part of the path selection algorithm. As such, it’s necessary that the IAC is increased for every AS in the path, even if the AS in question doesn’t support the IAC. To accomplish this, the IAC received over BGP is turned into IAClocal by adding 16 * AS_path_length. The IAClocal is then used in path selection.

In order let operators set finer-grained preferences on paths, an AS may add to or subtract from the IAC attribute when receiving a path over eBGP or when advertising a path over eBGP. In both cases, a router may subtract a maximum of 7, or add a maximum of 56. As a result, the IAC gets modified by any value between -14 and 112 by any given AS. This means that the IAClocal in each AS is between 2 and 128 higher than in the previous AS, increasing the granularity of the value that is compared when considering the IAC by more than a factor 120 over the situation where only the AS path length is considered.

3. Load balancing

To promote load balancing for traffic engineering purposes, the IAClocal is pseudo-randomized over a small range of values. The size
The minimum is 1 (no randomization), the maximum 7. The IAClocal is constructed from the IAC and Rt temporary value as follows:

$$Rt = IAC + (A_\text{origin} \text{ bit-AND } 0xffff) + (A_\text{local} \text{ bit-AND } 0xffff)$$

$$IAClocal = 16 \times A_\text{PathLen} + IAC + (Rt \mod R)$$

Note that IAC is signed, while OriginAS and LocalAS are unsigned; Rt must be large enough to hold 18-bit unsigned values. IAClocal must be positive and no higher than 255 * 128 = 32640. 255 is the upper bound on the number of AS hops imposed by the IP TTL/hop limit field, 128 the maximum increase of the IAC + AS path length per hop. So the IAClocal can be stored in a 15-bit unsigned field or a 16-bit signed field.

4. Backward compatibility

It would be undesirable to see a large shift in traffic flow when the IAC capability becomes available after a software update or when it’s administratively enabled for the first time. To minimize this problem, implementations SHOULD employ an IACscale variable, which can be set system-wide, but overridden through neighbor-specific or prefix-specific mechanisms such as route maps or policy filters. Conceptually, the IACscale variable takes a value between 0 and 1, although an approximation using integer math is acceptable. 0 means the behavior is identical to comparing just AS path lengths, 1 means the IAC is fully taken into account. Increasing the IACscale from 0 to 1 over time allows for the gradual introduction of the mechanism in existing networks. However, the IAC is propagated without taking the IACscale factor into account, so downstream networks can take advantage of the mechanism even when an intermediate network uses an IACscale of 0.

The steps below show how to calculate the IAC, taking all previously defined steps as well as IACscale into account:

$$Rt = IAC + (A_\text{origin} \text{ bit-AND } 0xffff) + (A_\text{local} \text{ bit-AND } 0xffff)$$

$$IAClocal = 16 \times A_\text{PathLen} + (IAC + (Rt \mod R)) \times IACscale$$

Implementers are encouraged to set IACscale to 0 by default when the IAC capability becomes available on previously configured systems but set the IACscale to 1 by default on systems without a previous configuration.
5. The IAC attribute

The new IAC path attribute is an optional transitive attribute that can take two forms: over eBGP, the attribute only contains the IAC. When communicated through iBGP, the attribute both contains the IAC and the IAClocal, in that order. Both attributes are 16-bit signed values encoded as two octets in network byte order.

Generating the IAC attribute:

The attribute is only included when routes are generated if it includes a non-zero value.

Updating of the IAC attribute:

1. When receiving a path with the IAC over eBGP: adjust IAC if desired, compute IAClocal.

2. When transmitting a path with the IAC over iBGP: add IAClocal to the IAC attribute.

3. When receiving a path with the IAC over iBGP: no action, unless the system is specifically configured to compute the IAClocal from the IAC. The IAC is NOT updated in the latter case.

4. When receiving a path without the IAC over iBGP: compute IAClocal.

5. When transmitting a path over eBGP: adjust IAC if desired. If there was no IAC attribute but the system is configured to adjust the attribute, create the attribute with a value that reflects the configured adjustment. If, after adjustment, the IAC is zero, the IAC attribute SHOULD be removed. Include IAC attribute with just the IAC value (not the IAClocal value) in eBGP updates.

6. IANA considerations

IANA is requested to allocate a BGP attribute type code.

7. Security considerations

When negative values outside of what’s allowed by this specification are used in the IAC, this may allow for unexpected and/or problematic path selection.
Appendix A. Document and discussion information

The latest version of this document will always be available at http://www.muada.com/drafts/. Please direct questions and comments to the idr mailinglist or directly to the author.

Author’s Address

Iljitsch van Beijnum
IMDEA Networks
Avda. del Mar Mediterraneo, 22
Leganes, Madrid 28918
Spain

Email: iljitsch@muada.com