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.
1. Introduction

There is only a single BGP attribute that is carried from AS to AS and updated at every AS hop: the AS path. The AS path length is therefore the only real inter-AS metric that BGP has.

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. A similar situation can occur in more complex types of connectivity. With a finer-grained value that is communicated across ASes this problem would be reduced.

This memo proposes such a finer-grained inter-AS metric: the Inter-AS Cost (IAC). With this metric, it will be possible for both destinations of traffic and intermediate ASes to make precise adjustments to the metrics seen by the sources of traffic and thus make it possible to arrive at more favorable load sharing ratios between multiple links to different ASes without having to resort to the advertisement of more specific prefixes.

In the past, efforts somewhat similar to this have been undertaken. In 1995, [I-D.antonov-bgp-metrics] proposed new per-hop BGP metrics. However, this proposal suffered from high complexity and a resulting risk of unforeseen consequences. A year later, [I-D.chen-bgp-dpa] proposed a new inter-AS metric for the purpose of allowing symmetric routing and load sharing. This proposal wasn’t fleshed out in much detail.

Neither proposal specifically addressed the issue of granularity in an inter-AS metric.

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 memo 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.

The choice for the numbers in question is made based on the assumption that there will never be more than 255 AS hops in an AS path (as each AS has at least one router and the IP TTL / hop limit field will not allow for more than 255 routers in the path) so the maximum will fit in a 16-bit signed integer.

3. Load balancing

To promote load balancing for traffic engineering purposes, the IAClocal may be pseudo-randomized over a small range of values. The details of this process are left up to implementers. An example would be the addition of a value in the range 0 - 3 that is stable across routers in an AS, session reconnects and reboots, but is otherwise random.

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 equation below shows how to calculate IAClocal:

\[ IAClocal = 16 \times ASPathLen + IAC \times IACscale \]

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 for prefixes if the resulting IAC value is different from zero.

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 optional transitive 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. However, this is a general problem with BGP and other routing protocols. For instance, if a BGP implementation were to remove AS numbers from the AS path, similar problems would arise.

It is highly recommended that implementers include a mechanism to remove the IAC attribute in incoming or outgoing BGP updates. This mechanism MUST be disabled by default.

8. References

8.1. Normative References


8.2. Informational References


Author’s Address

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

Email: iljitsch@muada.com
Full Copyright Statement

Copyright (C) The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.