Deprecation of AS_SET in BGP
IETF 115

draft-ietf-idr-deprecate-as-set-confed-set-09

W. Kumari, K. Sriram, L. Hannachi, **J. Haas**

Reminders About Aggregation

- Aggregation is creating a less-specific IP route from one or more more-specific routes.
  - A key property of this feature is that this causes the less-specific route to come into being conditionally.
  - Providers often want to create such aggregates, or similar routes, to provide for stable route announcements for a destination.
- BGP's rules about the properties of aggregate routes are covered in the base BGP RFC, RFC 4271. Most importantly for this discussion, it covers how to form the AS_PATH of the aggregate. In particular, it governs the creation of AS_SETs.

What are AS_SETs?

- When forming aggregates according to the rules in RFC 4271, Section 9.2.2.2, the algorithm roughly is:
  - For all contributors to the aggregate, find the longest common leading set of AS_SEQ segment ASes and use this to construct the left-hand-side of the AS_PATH.
  - Take everything else and throw it into an AS_SET; prune duplicate ASes in the AS_SET.
  - Minimize the number of BGP segments.
- Section 5.1.6 vaguely discusses the ATOMIC_AGGREGATE attribute. The implication is you're permitted (as long as you think it's safe) to discard the AS_SET.
  - Original intention is "don't make this route more specific".

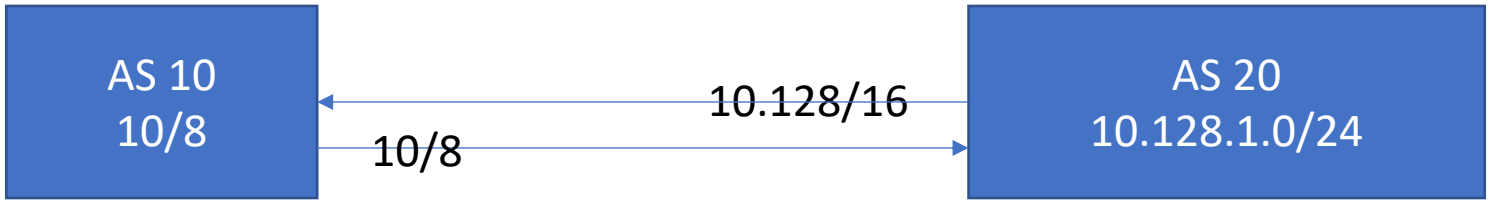
Example of why AS_SETs are needed when aggregating.

AS 10 owns address space 10.0.0.0/8 and has delegated 10.128.0.0/16 to AS 20.

AS 10 receives a route 10.128.0.0/16 from AS 20 with AS_PATH “20”
AS 10 creates the less specific route 10.0.0.0/8 via aggregation with AS_PATH “20” (no set!).
AS 10 then could advertise this aggregate to AS 20, which would drop it as a loop.

What happens if AS 20’s AS wasn’t in the AS_PATH and it accepted the less specific?

Example of why AS_SETs are needed when aggregating.



AS 20 may not be using all of its address space. It may simply be announcing its own address space, 10.128.0.0/16 to AS 10.

If AS 20 accepts a route covering its own address space, and internally has no route that covers it, forwarding loops can happen for the “un-covered” space.

- Normally the covering route with a discard is created by AS20 itself aggregating its internal prefixes for 10.128/16.
- But perhaps this route is not fully available within the AS for some reason.

If the 10/8 route had AS 20 in the path, it’d be a BGP loop and the less specific wouldn’t be accepted.

- Alternatively, if AS 20 is a stub AS and won’t learn the route from the Internet, AS 10 can simply not announce the aggregate to AS 20.

This scenario simply becomes more interesting when more than one route contributes to the aggregate and AS_SETs are created.

Best Current Practices

- Filter your own address space.
  - Don’t accept more specific routes to address spaces you are using from the outside world! This is easy in BGP policy.
  - Not accepting all less-specific routes is not easy to do in BGP policy.
- Best current practice is to *null-route* your own internal address space at your network borders. This functionality is normally provided by BGP implementations’ “aggregate” feature.
- If you do these things, loops covering your own address space when aggregates are created are less important.

Why do we care about AS_SETs?

- RPKI-Route Origin Validation needs a deterministic origin AS.
- AS_SETs, typically at the right-hand side of the AS_PATH, make that ambiguous.
- It can also make other BGP policies challenging. Most BGP policy algebras don’t support operations on SETs.
- RFC 6472 (BCP 172) say, “don’t create AS_SETs or CONFED_AS_SETs”.
  - They’re not common (300 at time of authoring), and increasingly becoming uncommon as RPKI-ROV is deployed.
  - Aggregation, however, is quite common! (About 10% of the Internet table has an AGGREGATOR attribute which suggests it’s being used as a tool.)

What needs to change in core BGP procedure?

- If you receive an AS_SET in a BGP route, you SHOULD do RFC 7606 “treat-as-withdraw”.
- Implementations SHOULD NOT create them when aggregating.
  - This is the part that needs most of the normative text vs. RFC 4271.
  - RFC 4271 didn’t cleanly describe “brief” aggregation. It’s clarified in our document.
  - “brief” aggregation isn’t good enough. What’s needed for RPKI-ROV purposes is a stable origin AS that would match a RPKI ROA. In most cases, that’d be the AS of the router that is doing the aggregation. ASes to the right of this stable origin AS would be pruned. **THIS IS NEW!!!**
  - Yes, keep putting ATOMIC_AGGREGATE on it.

Do we care about CONFED_AS_SETs?

- RPKI-ROV doesn’t care since the origin AS in a route that only has confederation segments in it is the confederation identifier and is thus known and stable.
- BGPsec does have normative procedure vs. confederation eBGP sessions. BGPsec can’t deal with SETs:
  - Either you apply similar procedures to not generate sets, or such routes are not BGPsec signed.
  - However, the bgpsec RFC itself suggests in Section 8.4 that its own procedures might matter less in confederations since they’re a zone of trust anyway.
- draft-09 removed CONFED_AS_SETs. Suggest in -10 to simply state that the procedures for removal of AS_SETs SHOULD apply to it as well.

What's next?

- Refine text vs. RFC 4271 changes.
- "Don't do that" (RFC 6472/BCP 172) isn't best text. We've got too much work with conformance testers to worry about loose wording changes.
- Get vendors to implement the new AS_PATH procedure for aggregation.