AIGP Last Call Issues

- After almost 5 years, 5 implementations, and significant deployment, draft finally reaches WG last call
- So folks not directly involved read the draft for the first time
- Some interesting issues raised during LC, some controversy about how to address those issues
- Some F2F discussion seems worthwhile before finalizing
- Note: no objections raised during LC to “meat” of draft, i.e. to rules for computing and using the value of the AIGP attribute (semantics)
- Objections raised to error handling, encoding, “leakage protection” at admin boundaries, i.e., stuff that might impact “somebody else”
- Want to focus discussion on LC issues …
AIGP

- BGP Path Attribute: Accumulated IGP Metric of path to prefix
- Allows IGP metric to be major determinant of bestpath selection for BGP-distributed internal routes
  - Provisioning determines the set of prefixes to which AIGP gets attached
  - BGP becomes a sort of IGP for those prefixes
- **Must not leak** out past administrative boundary
  - **Not** an inter-provider metric
  - AIGP is non-transitive attribute, discarded when not recognized
  - By default, even if recognized, AIGP treated as unrecognized (discarded) on EBGP sessions
    - All admin boundaries are EBGP sessions (converse not true)
- For possible future expansion, attribute coded as list of TLVs, but only type 1 (IGP distance) defined
Error Handling for Malformed AIGP Attribute

• Not clearly specified in draft
• What’s best: *treat as withdraw*, or *discard attribute*?
• *Treat as withdraw* is default for attributes affecting bestpath selection
  • But AIGP is only to be used in scenarios where there is tunneling to the next hop; complete consistency not needed
• *Discard attribute* is therefore less disruptive way to handle malformed attribute
• *Discard attribute* is also very like what is done with an unrecognized transitive attribute
• Proposed resolution: use *discard attribute* as error handling method
Can the Non-Transitivity Break?

• R1---(ibgp)---ASBR1----(ebgp)----ASBR2

• AS containing ASBR2 uses AIGP
  • ASBR2 mistakenly sets the transitive bit on the AIGP attribute
  • ASBR2 mistakenly sends AIGP attribute to ASBR1

• ASBR1 does not understand attribute, sees transitive bit, forwards to R1 when really the attribute ought to be discarded

• R1 understands AIGP attribute and is provisioned to use it.
  • But now it mistakenly has received the attribute from across an admin boundary
  • Should R1:
    • Clear the transitive bit and forward the attribute (local repair)? Or
    • Discard attribute as malformed

• Proposed resolution: discard attribute as malformed
  • Attribute isn’t supposed to be processed by R1 or forwarded any further
  • Restores the proper non-transitive behavior
TLV Encoding Issues

• Length field not specified “correctly”, shouldn’t include length of type and length fields
  • Too late
  • Sorry 😞

• What if attribute contains multiple type 1 TLVs?
  • Is this malformed, or should one of the type 1 TLVs be used and the others ignored?
  • Proposed resolution: do not treat as malformed, use the first one.
  • Other TLV types to be ignored if not recognized, of course.
Disabled By Default

• Default per-session settings:
  • Do not originate routes with AIGP
  • On EBGP sessions, discard attribute if received
  • So:
    • On EBGP sessions, attribute shouldn’t pass unless enabled on both sides
    • On IBGP sessions, attribute will pass if enabled on one side

• Enough protection against leakage?
  • Think so; but controversial on mailing list.

• Enough protection against errors?
  • Can’t protect against all errors
Capability Needed?

• Capability needed?
  • No, shouldn’t need a capability for every new (optional) attribute