BGP Monitoring Protocol (BMP) Revision -07

John Scudder

November 9, 2012

Status!

- Last time I presented BMP to GROW was at IETF-75 (really?!?)
- Numerous suggestions for changes/additions since then
  - Incorporated (with credit, I hope),
  - Not incorporated on purpose,
  - Or conceivably dropped though I hope not. Please check.

Changes between -02 and -07

- Version bumped to 3
- Length field added to fixed header
- Clarified error handling
- Send relevant FSM event in Peer Down msg
- Local addr, local+remote ports in Peer Up msg
- Require EoR per peer after initial dump
- Add Initiation Message
- Add Termination Message

Changes between -02 and -07

- Permit muxing pre-, post-policy feeds
  - L-bit used as multiplexor
- New stat types –
  - 5 (number of updates invalidated due to ORIGINATOR_ID), 6 (number of updates invalidated due to AS_CONFED_SEQUENCE/AS_CONFED_SET), 7 (number of routes in Adj-RIB-In) and 8 (number of routes in Loc-RIB)
- Let monitoring station be active party
- Session establishment must be rate-controlled

Not Adopted

- Instrumentation of next hops
  - Why? Nice idea but complex, would hinder completion. Possibly other solutions available.
- Raw inbound stream option
  - Why? Unlikely to be implemented.
- Instrumentation of treat-as-withdraw
  - Why? Not as easy as it sounds.
- All “why” are IMHO of course.

State Compression

- BMP, like BGP, is (or anyway can be) state-compressed –
  - Peer sends (announce A, withdraw A, announce A, withdraw A, announce A)
  - Monitoring station may see (announce A)… depending on dynamics
  - Converges to correct final state but compresses out intermediate states when under load
- State-compression bounds storage required to O(size of Adj-RIBs-In)
- In short, it's scalable (memory, but also CPU)

Raw Inbound Stream

- Echo each received BGP message to monitoring station as it is read from the wire
- Attractive. But fundamentally not compressible
- Consequence of slow monitoring station – router builds ever-deeper backlog of messages to echo until it runs out of buffer. Then what?
  - Could drop messages,
  - Could drop the session,
  - Could flow-block real BGP peers
- All of these would be unacceptable in most deployments

Treat-As-Withdraw

- draft-ietf-idr-error-handling recommends “treat as withdraw” handling for many errored BGP updates (that would otherwise have reset the session)
- Suggestion was to replicate these into BMP
- Sounds nice to have for debugging!

Treat-As-Withdraw

- But! Implementations will discard (not just hide) “treat as withdraw” updates
  - Can’t keep negative state around forever, don’t know when to free it
  - Can’t bound number of bad updates you’d have to keep
- Without data in Adj-RIB-In, there is nothing to feed into BMP
  - Reduces to a previously unsolved problem (Raw Inbound Stream)

Next Steps

- Authors believe draft -07 is done, or close to done
  - Please review!
- Advancement in 2013
- Questions/comments/discussion?