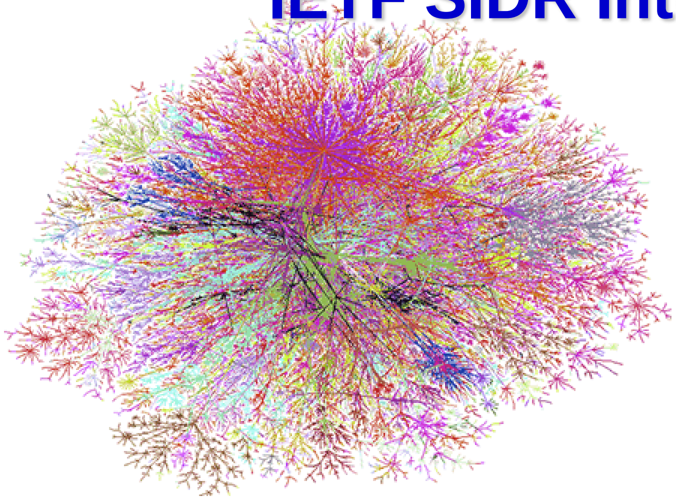
Trustworthy Networking Program

Some BGPSEC Protocol Design & Performance Issues for Discussion

K. Sriram

(ksriram@nist.gov)

IETF SIDR Interim Meeting, June 6, 2012



Some Performance Questions

- Q1: What is an acceptable convergence time after a BGPSEC peering session reset? (See Wes George's question (in Paris) and some comments shared by Geoff with me – later in these slides.)
- Q2: What degree of peering is reasonable to assume for various router scenarios such as Provider Edge (PE), Route Reflector (RR), Route Server (RS), etc.? (The convergence time and the RIB size would naturally depend on the number of BGPSEC peering sessions that the PE, RR, or RS needs to handle.)

Some Performance Questions

- Q3: Is there any difference in how the RIBs (RIB-in, RIB-out, etc.) are managed at a RR or RS as compared to how it is done at the PE router?
- Q4: Is a large percentage of the clients of an RS consists of stub routers, i.e., they do not require signed updates in receive direction?

Convergence Time Requirement

- Wes George (question at the Paris SIDR meeting): Are the 30 sec and 70 sec type of numbers (for convergence after BGPSEC peering reset) small enough? May not be. (This question followed my presentation: http://www.ietf.org/proceedings/83/slides/slides-83-sidr-7.pdf)

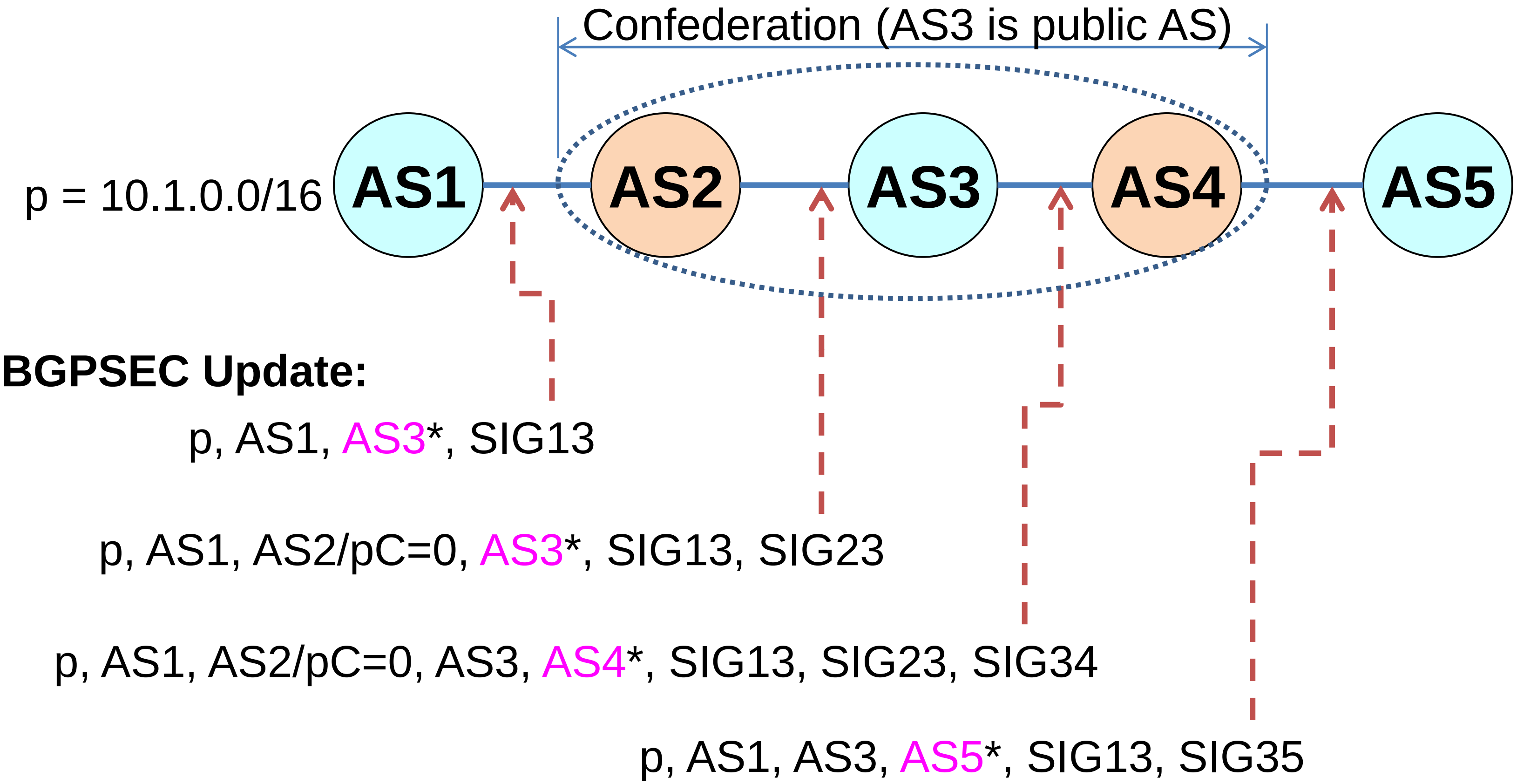
Geoff Huston’s Comment

- As a counter point, Geoff shared the following with me during a coffee break (in Paris):

  The way to think about those delays is that the data packets would traverse a possibly suboptimal path for the period of T seconds or minutes -- whatever the BGPSEC re-convergence delay number is (after a session reset). They still get to their destinations possibly with some added delay. That is a property of BGP. And most of the time users don't even notice that extra data packet delay at the application layer.

Some Slides to Assist
Further Discussion of AS_PATH, Confeds, AS migration

Confederation Sequence – w/o AS_PATH



BGPSEC Update:

p, AS1, AS3*, SIG13

p, AS1, AS2/pC=0, AS3*, SIG13, SIG23

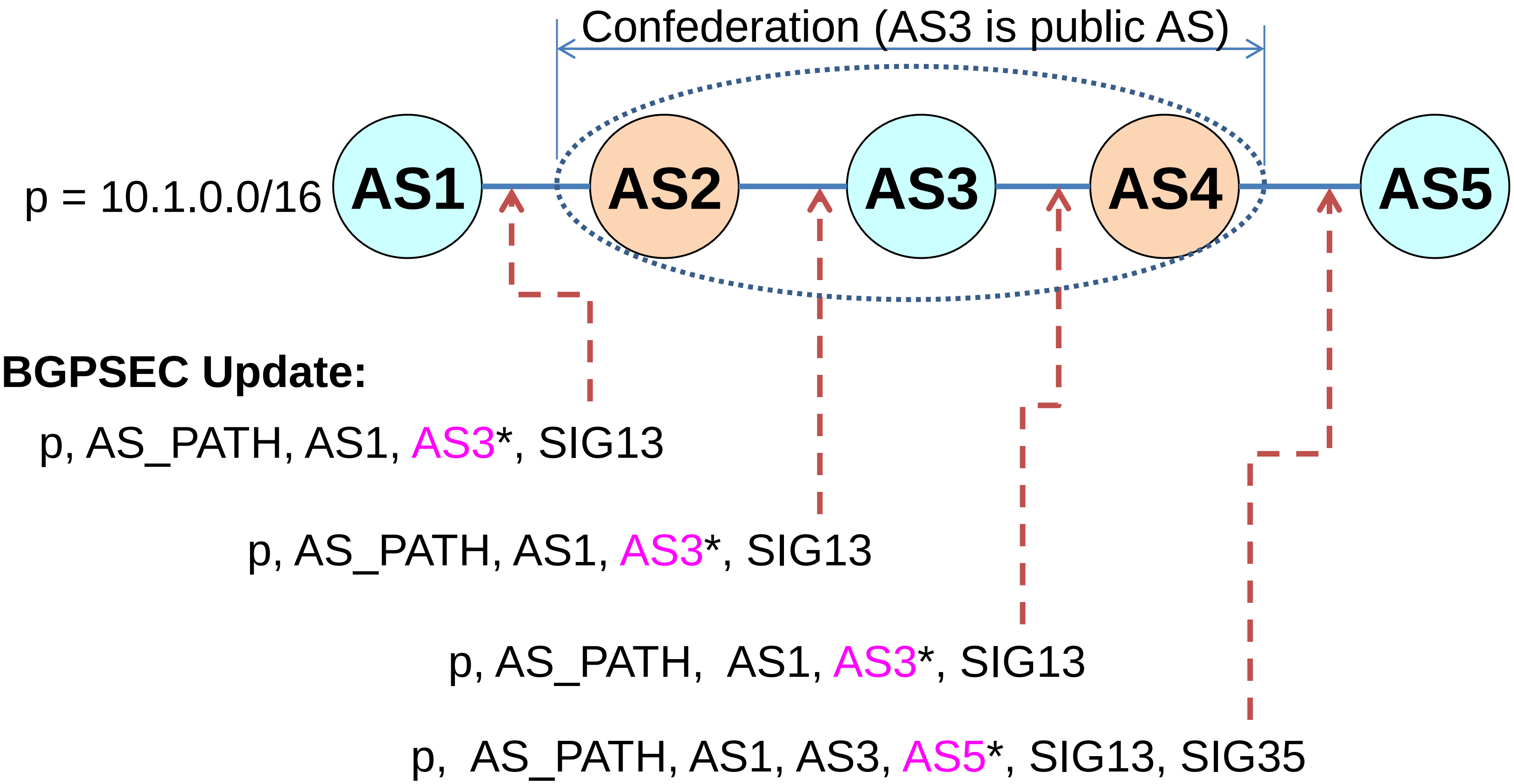
p, AS1, AS2/pC=0, AS3, AS4*, SIG13, SIG23, SIG34

p, AS1, AS3, AS5*, SIG13, SIG35

* Signed over next AS, but not included in the update on wire

- AS3 is the public identity of the confed; AS2 and AS4 are known as AS3 to their external peers
- Signatures within the confed are removed by AS4 before forwarding to external peer AS5
- ASs within the confed set their pC = 0, except one that is the public ID
- pCount (or pC) is not shown in the figure except when it is 0

Confederation Sequence – w/ AS_PATH



BGPSEC Update:

p, AS_PATH, AS1, AS3*, SIG13

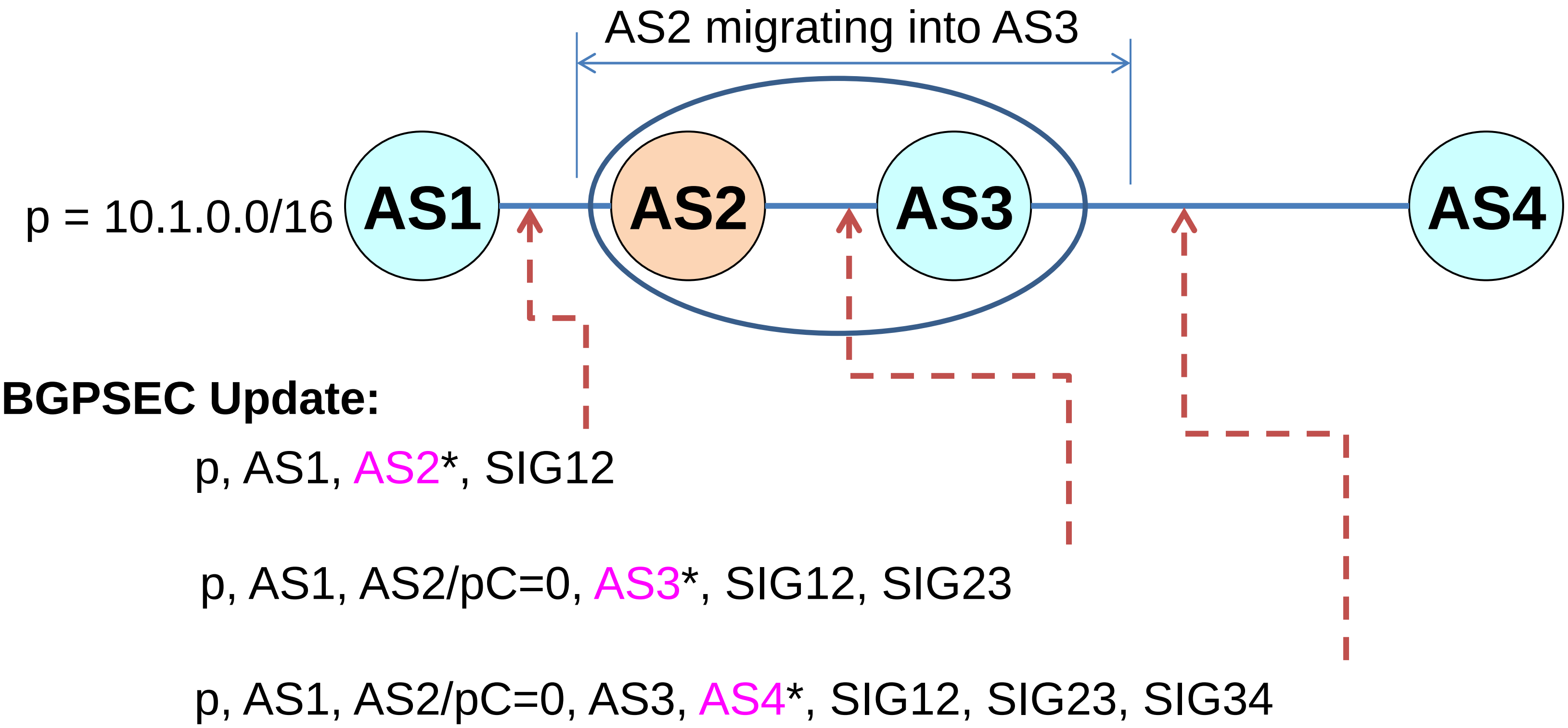
p, AS_PATH, AS1, AS3*, SIG13

p, AS_PATH, AS1, AS3*, SIG13

p, AS_PATH, AS1, AS3, AS5*, SIG13, SIG35

- ASs within the confed do loop detection using the BGP-4 AS_PATH
- Confed ASs do not add signature except when the update is being sent to external peer
- pCount (or pC) not shown in the figure (it is 1 for each AS in Secure_Path in this example)

AS Migration



AS2 migrating into AS3

p = 10.1.0.0/16

AS1

AS2

AS3

AS4

BGPSEC Update:

p, AS1, AS2*, SIG12

p, AS1, AS2/pC=0, AS3*, SIG12, SIG23

p, AS1, AS2/pC=0, AS3, AS4*, SIG12, SIG23, SIG34

- This above approach is trivially supportable.
- Other approach would be that {AS2, AS3} behave like a confed, and then we use an approach that is used for confeds