Source Address Validation Using BGP UPDATEs, ASPA, and ROA (BAR-SAV)

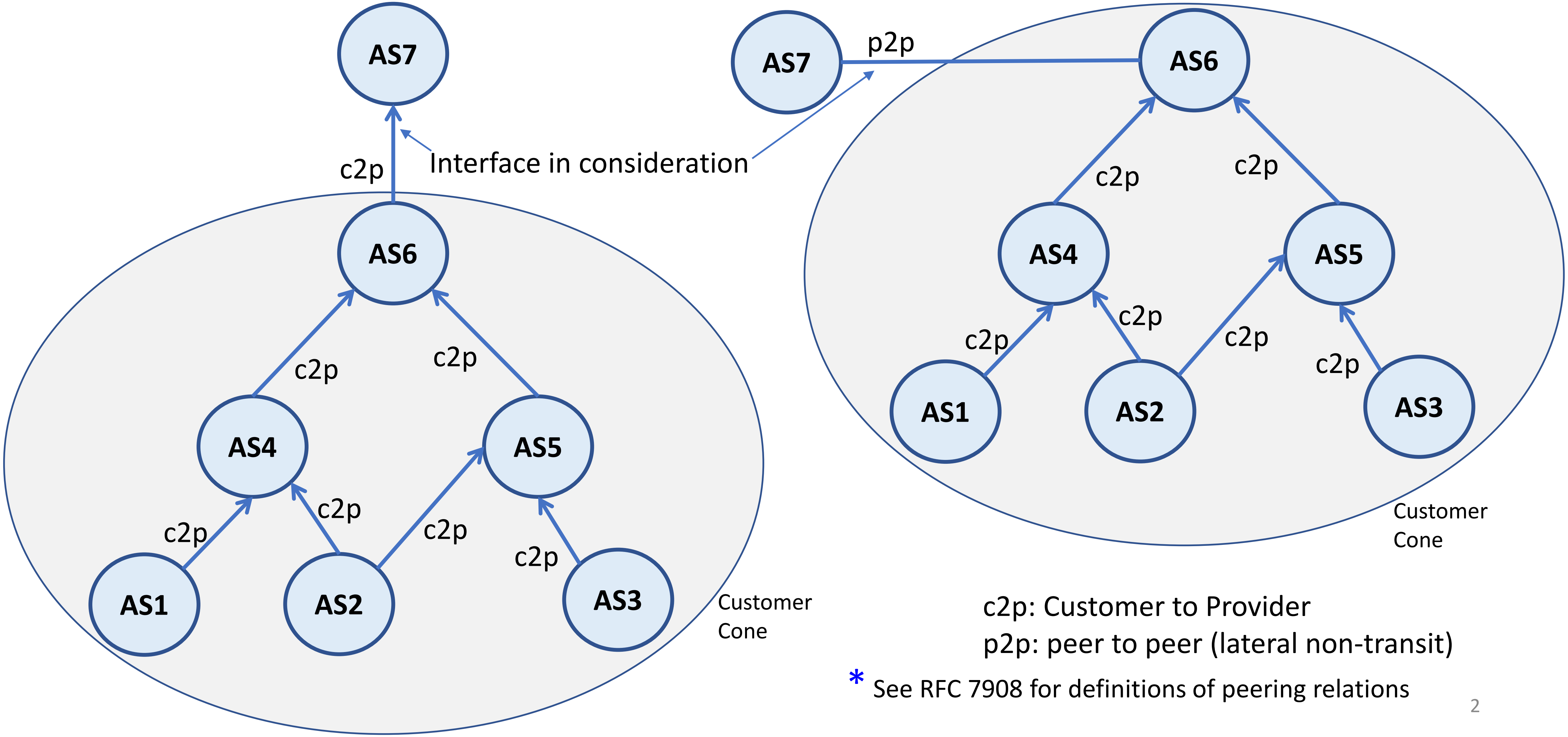
Kotikalapudi Sriram, Igor Lubashev, and Doug Montgomery

Email: ksriram@nist.gov ilubashe@akamai.com dougm@nist.gov

June 2022

Goal: Construct Permissible Ingress Prefix List for SAV (at AS7)

The methodology is the same for a Customer or Lateral (i.e., non-transit) Peer* Interface

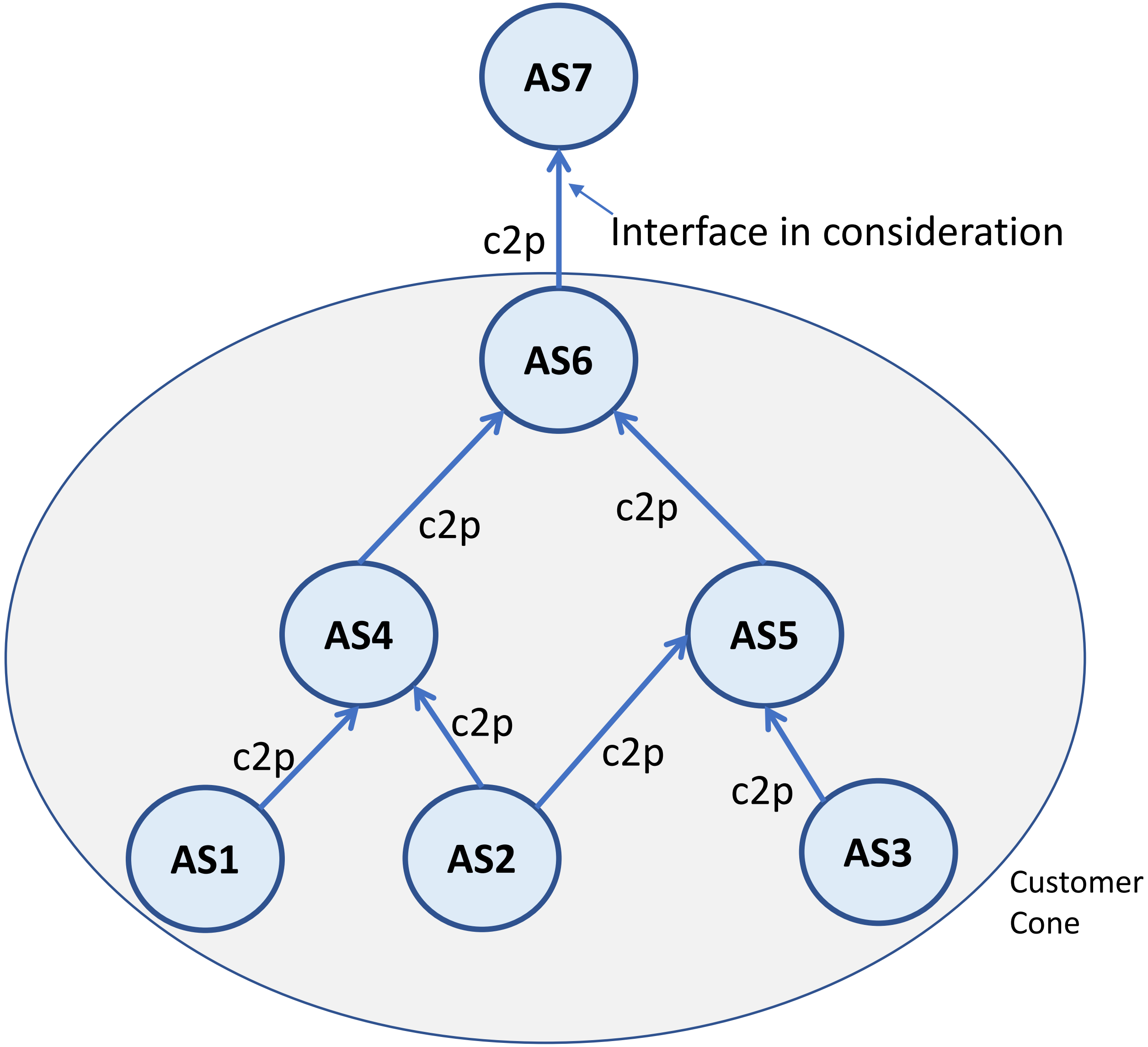


c2p: Customer to Provider
p2p: peer to peer (lateral non-transit)

* See RFC 7908 for definitions of peering relations

SAV Using Only ASPA and ROA (Procedure X)

Construction of Permissible Ingress Prefix List for SAV (at AS7)



When ASPA and ROA adoption is ubiquitous (in the future)

A. Obtain the set of ASNs in the Customer’s customer cone (CC) using ASPAs
B. Gather all prefixes in ROAs associated with the ASNs found in Step A. Keep only the unique prefixes.
C. The set computed in Step B is the permissible prefix list for SAV for the interface in consideration.

But there will be…

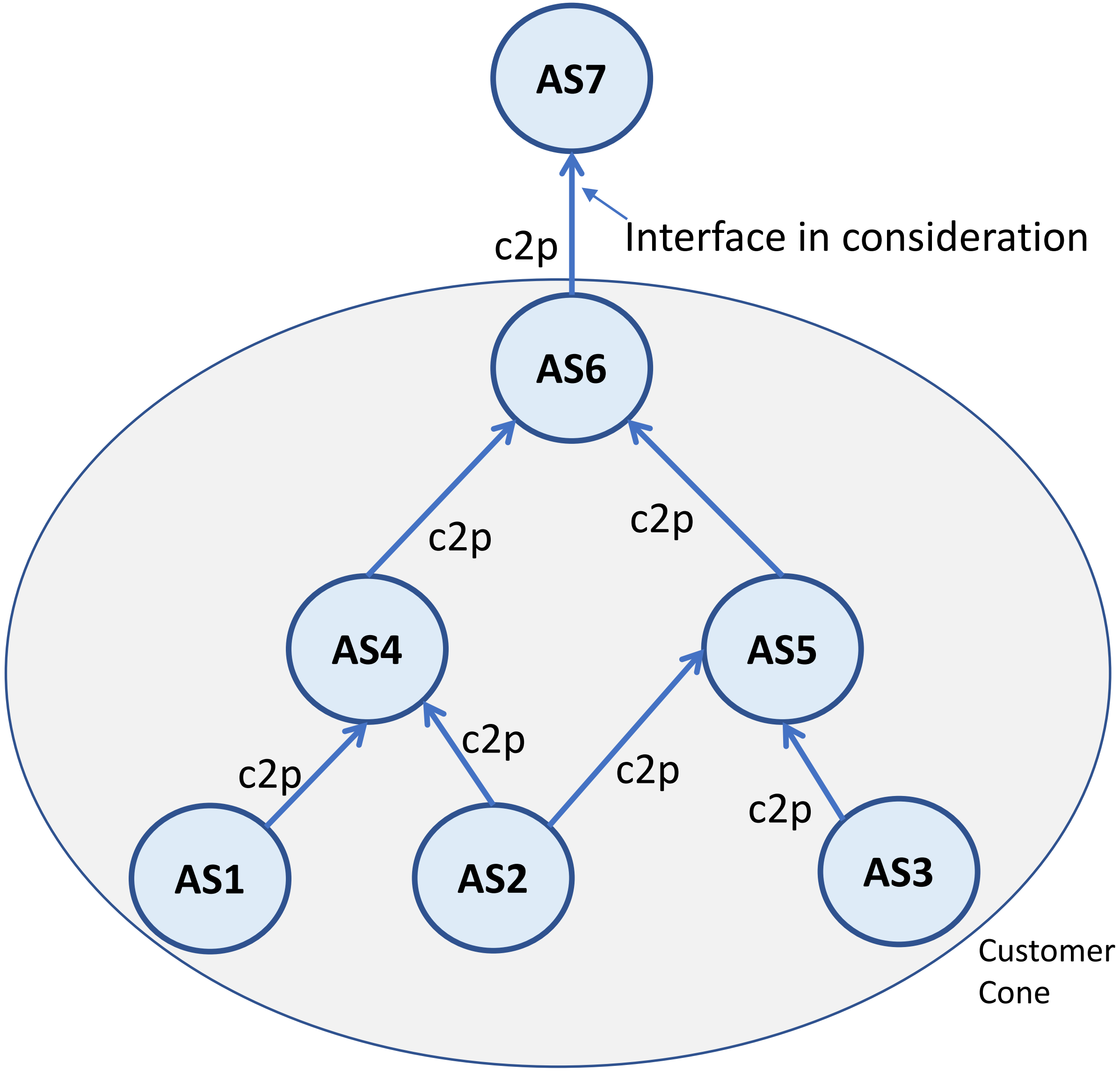
Partial deployment of ROAs and ASPAs for some time

- During that period…
  - ✓ BAR-SAV compensates
  - ✓ Makes complementary use of BGP UPDATEs, ASPA, and ROA
    - Incorporates a refined version of EFP-uRPF*

* Enhanced Feasible Path uRPF (EFP-uRPF) [RFC 8704]

SAV Using ASPA, ROA, and BGP UPDATE (BAR-SAV)

Construction of Permissible Ingress Prefix List for SAV (at AS7)



Applicable in the period when ASPA and ROA adoption is not ubiquitous

A. Obtain the set of ASNs in the Customer's customer cone (CC) using ASPAs and AS_PATHs

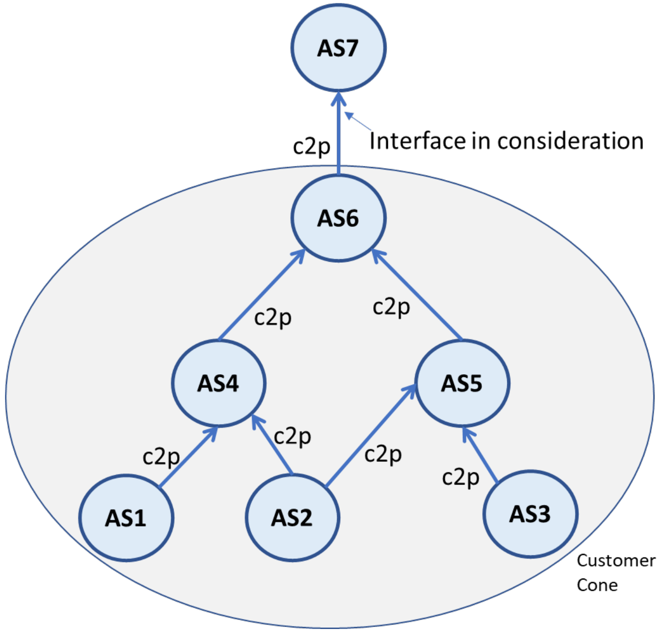
B. Gather all prefixes in ROAs associated with the ASNs found in Step A.

C. Gather all prefixes in BGP UPDATE messages with originating ASN among ASNs found in Step A.

D. Combine sets found in Steps B and C. Keep only the unique prefixes. This is the permissible prefix list for SAV for the interface in consideration.

A Note on Customer Cone Computation

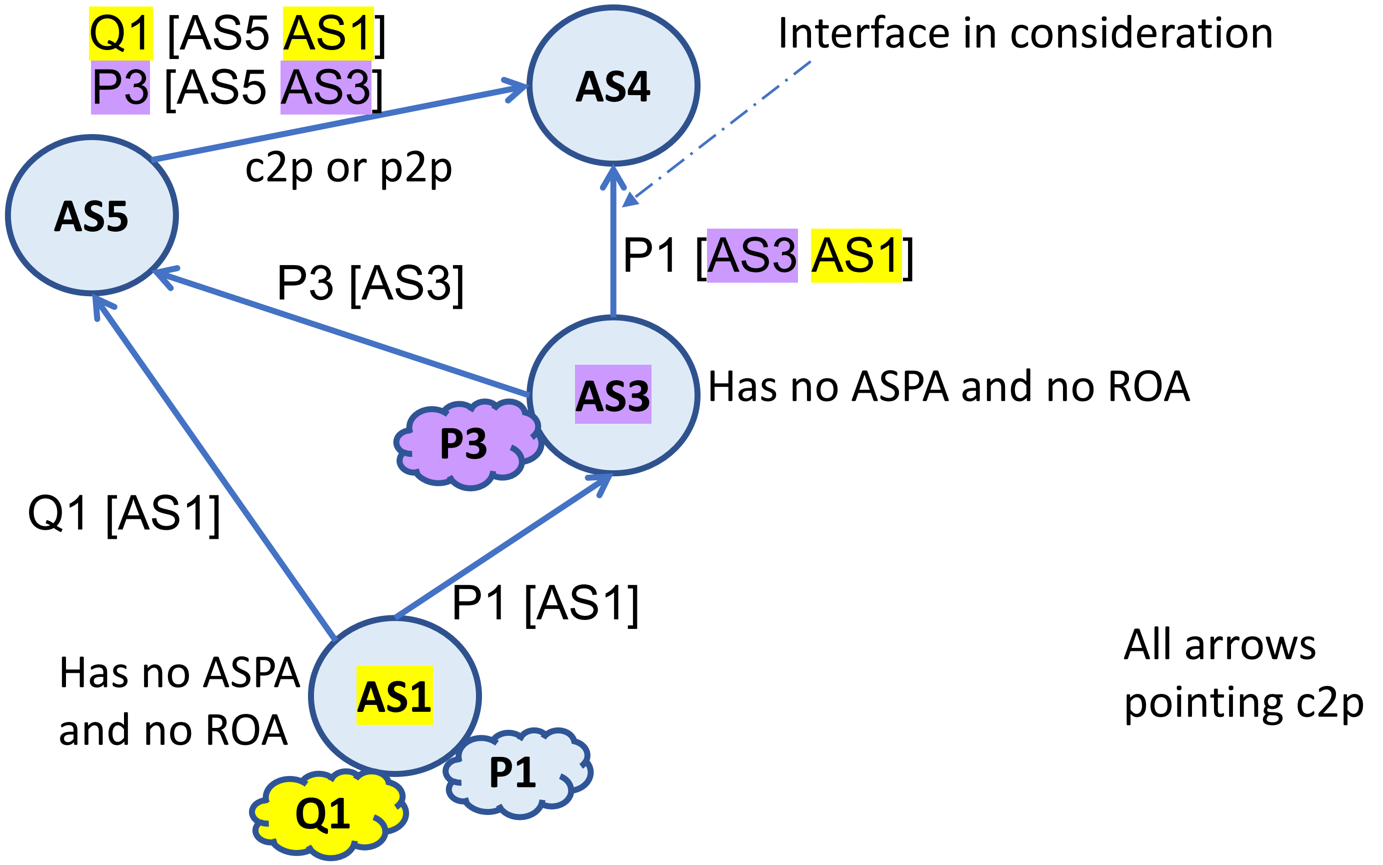
- One should *not* compute a customer cone by separately processing ASPA data and AS_PATH data and then merging the two sets of ASes at the end. Doing so is likely to miss ASes from the customer cone.



- Instead, both ASPAs and AS_PATHs should be used to iteratively expand the discovered customer cone. When new ASes are discovered, both ASPA and AS_PATH data should be used to discover customers of those ASes. This process is repeated for newly discovered customer ASes until there are no new ASes to be found.

Refined Algorithm A of EFP-uRPF [RFC 8704] Incorporated into BAR-SAV

- Only Q1 is detected by Alg. A of RFC 8704
- Both Q1 and P3 are detected by BAR-SAV



EFP-uRPF = Enhanced Feasible Path uRPF

- Much better detection of “Hidden” prefixes in multihoming scenarios by BAR-SAV

Description of the BAR-SAV Procedure

- **Step A:**
  - Start with the Customer's or Peer's ASN and call it the Level 1 Set of ASNs.
  - Find Customer ASNs for all ASNs in the previous level's Set (Level 1 for the 1st iteration) using ASPAs and BGP AS_PATH data. (Note: The unique AS_PATHs in Adj-RIBs-In [RFC4271] of all interfaces at the BGP speaker computing the SAV filter are considered.) ASNs found at this step that do not belong to any of the prior Sets are called this level's Set (Level 2 for the 1st iteration).
  - Repeat the previous step until an empty Set of ASNs is computed.
  - The union of all thus computed Sets is the CC AS set. Call it AS-set A.

- **Step B:** Derive the permissible prefix list for SAV by two complementary methods (B.1 and B.2):

  ❖ **Step B.1:** From ROA data, compute the set of prefixes that are authorized to be announced by each ASN in AS-set A.

  ❖ **Step B.2:** Using the routes in Adj-RIBs-In of all interfaces, compute the set of unique prefixes that are originated from each ASN present in AS-set A.

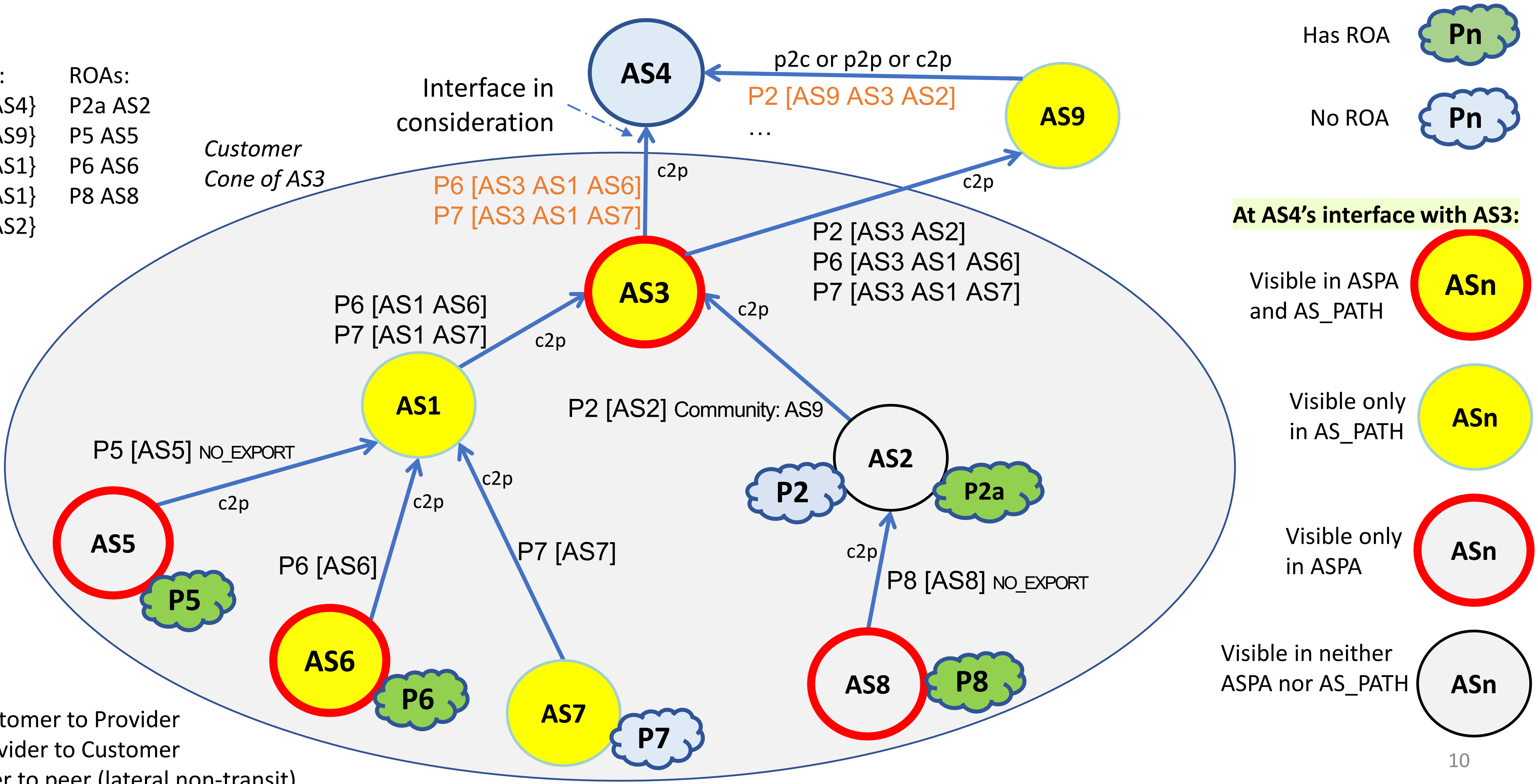
- **Step C:** Construct the permissible prefix list for SAV by forming the union of the sets obtained in steps B.1 and B.2.

The next 3 slides illustrate the details of how BAR-SAV works

How BAR-SAV Works
Finding **All** ASes and Prefixes in Customer's (or Peer's) Customer Cone
Using BGP Announcements (as seen at AS4), ASPA, and ROA

ASPAs:
AS3 {AS4}
AS3 {AS9}
AS5 {AS1}
AS6 {AS1}
AS8 {AS2}

ROAs:
P2a AS2
P5 AS5
P6 AS6
P8 AS8



Interface in consideration

Customer Cone of AS3

AS4

p2c or p2p or c2p

P2 [AS9 AS3 AS2]

…

AS9

c2p

P6 [AS3 AS1 AS6]
P7 [AS3 AS1 AS7]

c2p

AS3

P2 [AS3 AS2]
P6 [AS3 AS1 AS6]
P7 [AS3 AS1 AS7]

P6 [AS1 AS6]
P7 [AS1 AS7]

c2p

c2p

AS1

P2 [AS2] Community: AS9

P5 [AS5] NO_EXPORT

c2p

c2p

c2p

AS2

P2

P2a

AS5

P5

P6 [AS6]

P7 [AS7]

c2p

P8 [AS8] NO_EXPORT

AS6

P6

AS7

P7

AS8

P8

Has ROA — Pn

No ROA — Pn

At AS4's interface with AS3:

Visible in ASPA and AS_PATH — ASn

Visible only in AS_PATH — ASn

Visible only in ASPA — ASn

Visible in neither ASPA nor AS_PATH — ASn

c2p: Customer to Provider
p2c: Provider to Customer
p2p: peer to peer (lateral non-transit)

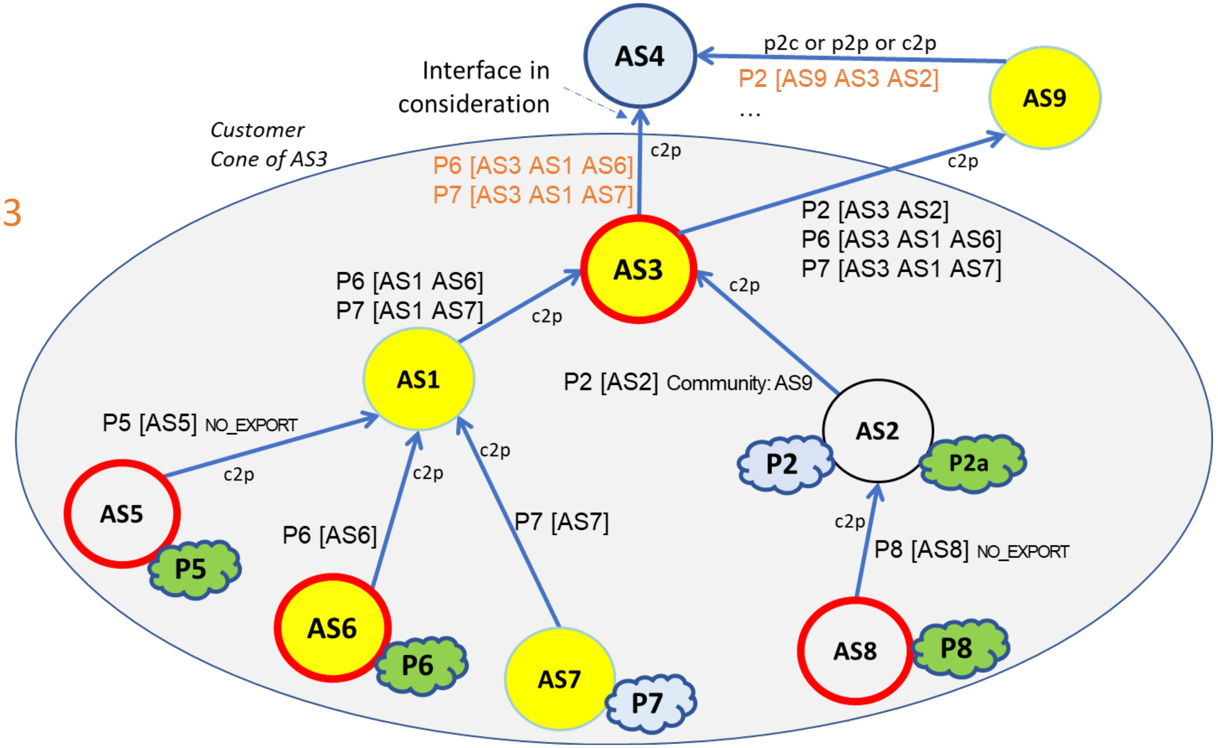
Finding **All** ASes in the CC using BGP AS_PATH and ASPA

INPUTS

ASPAs:
AS3 {AS4}
AS3 {AS9}
AS5 {AS1}
AS6 {AS1}
AS8 {AS2}

ROAs:
P2a AS2
P5 AS5
P6 AS6
P8 AS8

BGP UPDATE AS_PATHs:
Interface in Consideration: AS3
P6 [AS3 AS1 AS6]
P7 [AS3 AS1 AS7]
Other Interfaces:
P2 [AS9 AS3 AS2]



OUTPUT

Iteration	Customer Cone	New ASes from ASPA	New ASes from AS_PATH
1	AS3	None	P6 [AS3 AS1 AS6] → AS1 P7 [AS3 AS1 AS7] → AS1 P2 [AS9 AS3 AS2] → AS2
2	AS3, AS1, AS2	AS5 {AS1} → AS5 AS6 {AS1} → AS6 AS8 {AS2} → AS8	P6 [AS3 AS1 AS6] → AS6 P7 [AS3 AS1 AS7] → AS7
3	AS3, AS1, AS2, AS5, AS6, AS8, AS7	None	None

Finding **All** Prefixes in the CC using BGP Routes and ROA

INPUTS

ASPAs:
AS3 {AS4}
AS3 {AS9}
AS5 {AS1}
AS6 {AS1}
AS8 {AS2}

ROAs:
P2a AS2
P5 AS5
P6 AS6
P8 AS8

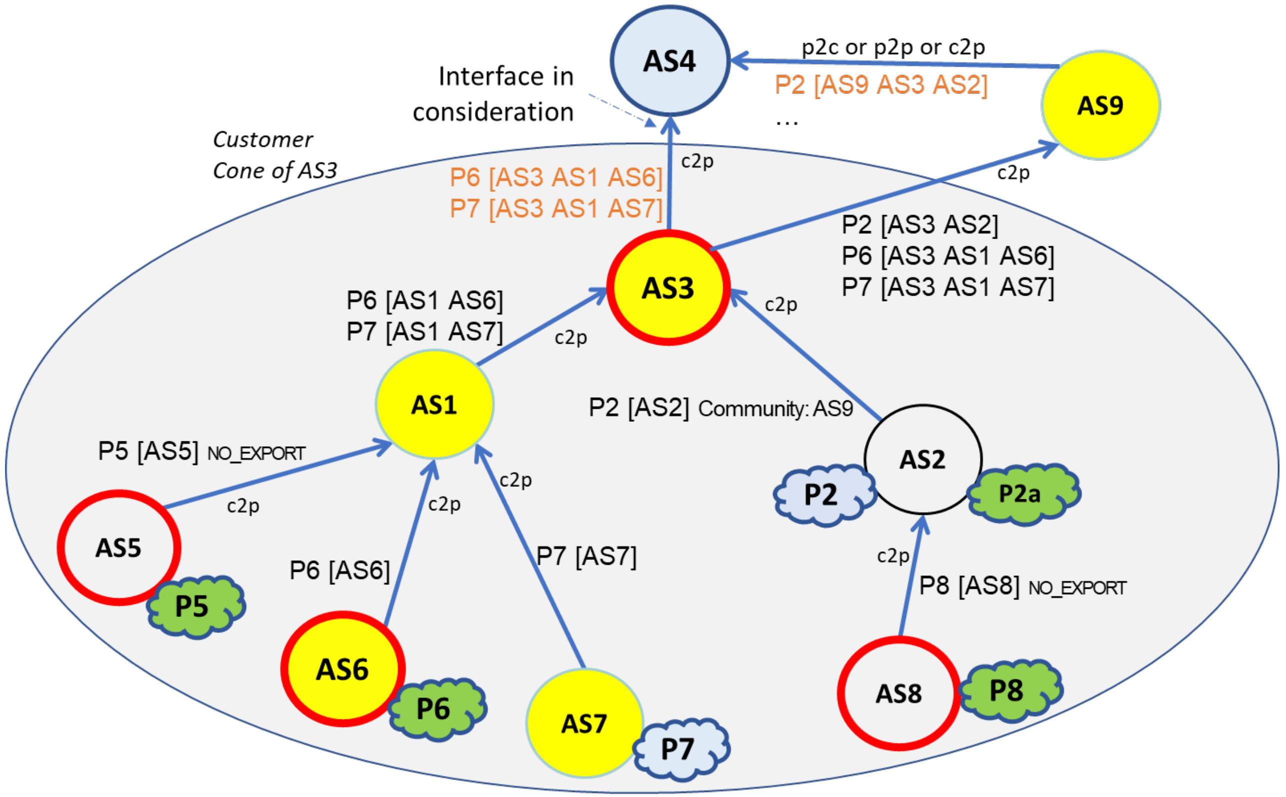
BGP UPDATE AS_PATHs:
Interface in Consideration: AS3
P6 [AS3 AS1 AS6]
P7 [AS3 AS1 AS7]
Other Interfaces:
P2 [AS9 AS3 AS2]

Customer Cone
AS1, AS2, AS3, AS5, AS6, AS7, AS8

OUTPUT

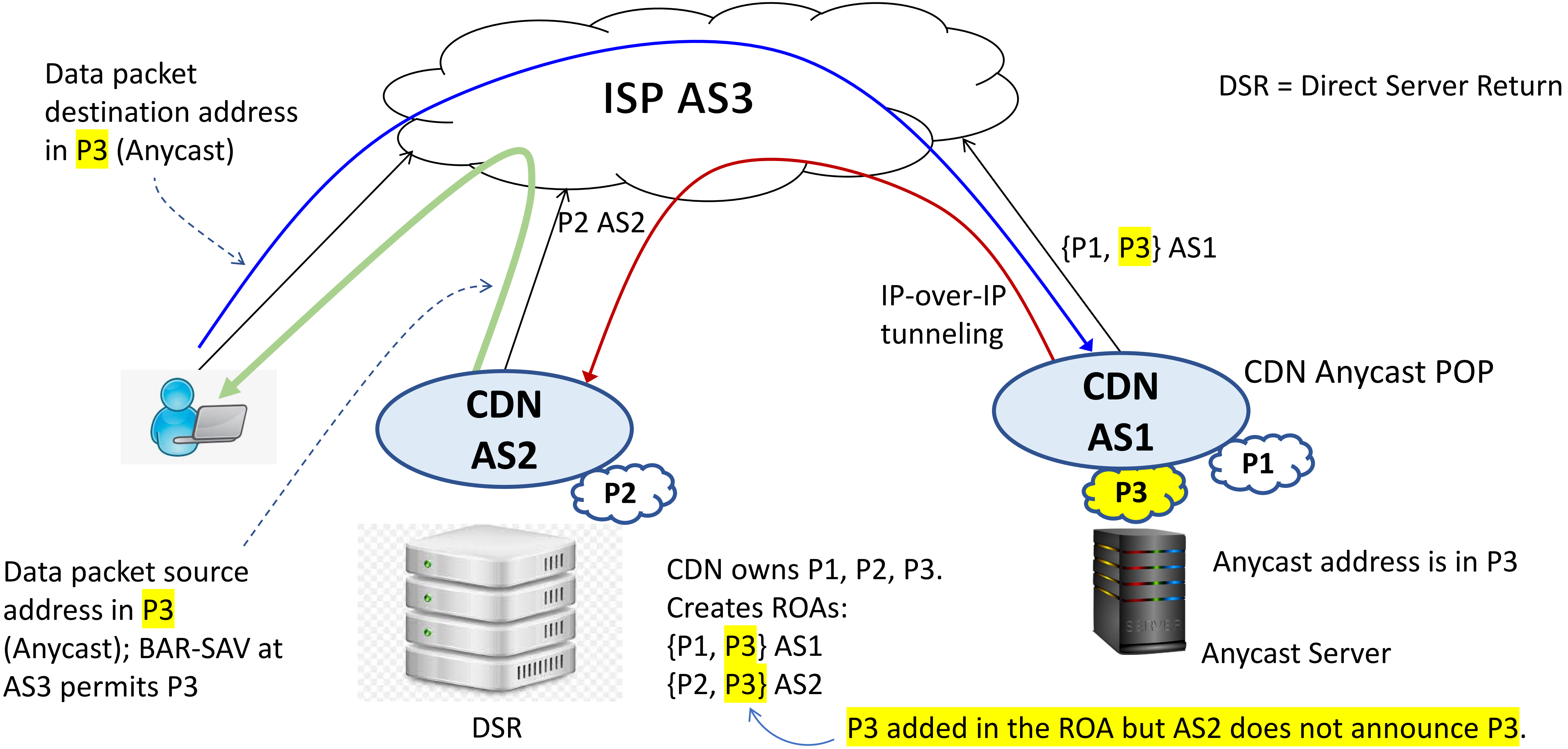
ASN	Prefixes from ROA	Prefixes from BGP
AS1		
AS2	(P2a AS2) → P2a	P2 [AS9 AS3 AS2] → P2
AS3		
AS5	(P5 AS5) → P5	
AS6	(P6 AS6) → P6	P6 [AS3 AS1 AS6] → P6
AS7		P7 [AS3 AS1 AS7] → P7
AS8	(P8 AS8) → P8	

SAV Prefixes
P2, P2a, P5, P6, P7, P8



Content Delivery Network (CDN) Application

Example of how the BAR-SAV method solves the DSR blocking problem



ISP AS3

DSR = Direct Server Return

Data packet destination address in P3 (Anycast)

P2 AS2

{P1, P3} AS1

IP-over-IP tunneling

CDN AS2

P2

CDN AS1

CDN Anycast POP

P1

P3

Anycast address is in P3

Anycast Server

Data packet source address in P3 (Anycast); BAR-SAV at AS3 permits P3

DSR

CDN owns P1, P2, P3.
Creates ROAs:
{P1, P3} AS1
{P2, P3} AS2

P3 added in the ROA but AS2 does not announce P3.

Backup slides

Detailed Procedure X

Creating the Permissible Prefix List for SAV for a Customer or Lateral Peer using only ASPA and ROA

1. Let the Customer or Lateral Peer ASN be denoted as AS-k.
2. Let i = 1. Initialize: AS-set S(1) = {AS-k}.
3. Increment i to i+1.
4. Create AS-set S(i) of all ASNs whose ASPA data declares at least one ASN in AS-set S(i-1) as a Provider.
5. If AS-set S(i) is null, then set i_max = i - 1 and go to Step 6.
   Else, go to Step 3.
6. Form the union of the sets, S(i), i = 1, 2, ..., i_max, and name this union as AS-set A.
7. Select all ROAs in which the authorized origin ASN is equal to any ASN in AS-set A. Form the union of the sets of prefixes listed in the selected ROAs. Name this union set of prefixes as P-set.
8. Apply P-set as the list of permissible prefixes for SAV.

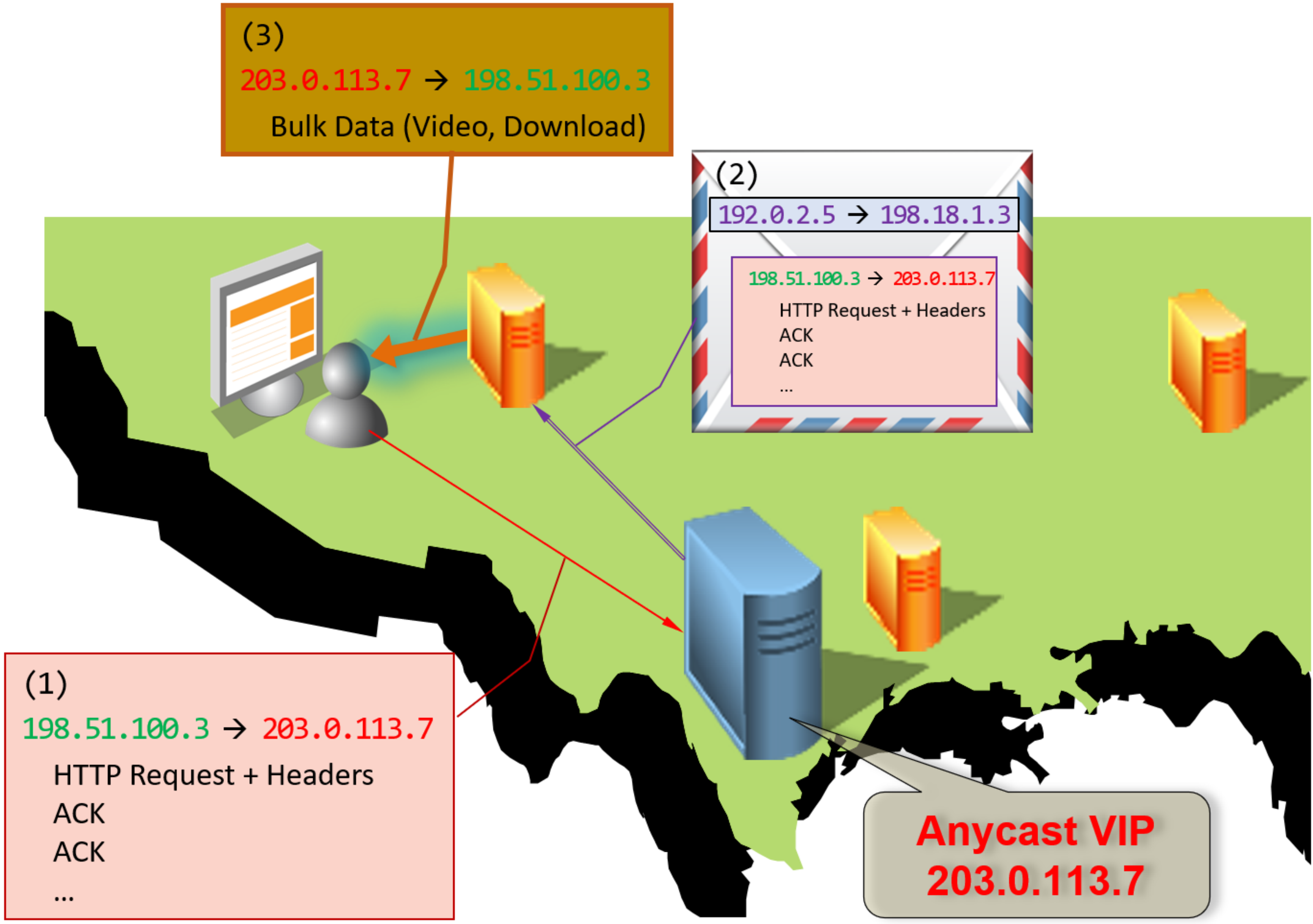
Note: Algorithm X is for future use when the deployment of ASPA and ROA is ubiquitous.

Detailed BAR-SAV Procedure

1. Let the Customer or Lateral Peer ASN be denoted as AS-k.
2. Let i = 1. Initialize: AS-set Z(1) = {AS-k}.
3. Increment i to i+1.
4. Create AS-set A(i) of all ASNs whose ASPA data declares at least one ASN in AS-set A(i-1) as a Provider.
5. Create AS-set B(i) of all such ASNs where each member is an immediate downstream neighbor (customer) AS of at least one ASN in AS-set Z(i-1) based on information in the unique AS_PATHs present in Adj-RIBs-In of all interfaces.
6. Form the union of AS-sets A(i) and B(i) and call it AS-set X. From AS-set X, remove any ASNs that are present in Z(j), for j=1 to j=(i-1). Call the resulting set Z(i).
7. If AS-set Z(i) is null, then set i_max = i - 1 and go to Step 8. Else, go to Step 3.
8. Form the union of the AS-sets, Z(i), i = 1, 2, ..., i_max, and name this union as AS-set C.
9. Select all ROAs in which the authorized origin ASN is in AS-set C. Form the union of the sets of prefixes listed in the selected ROAs. Name this union set of prefixes as Prefix-set P1.
10. Using the routes in Adj-RIBs-In of all interfaces, create a list of all prefixes originated by any ASN in AS-set C. Name this set of prefixes as Prefix-set P2.
11. Form the union of Prefix-sets P1 and P2. Apply this union set as the list of permissible prefixes for SAV.

Anycast/Edge Hybrid – Direct Server Return

1. Anycast POPs lookup "best" edge POP for each new connection (using the actual user IP)
2. Anycast POPs tunnel packets to edge POPs
3. Edge servers send data to users directly – Direct Server Return (DSR)



(3)
203.0.113.7 → 198.51.100.3
Bulk Data (Video, Download)

(2)
192.0.2.5 → 198.18.1.3
198.51.100.3 → 203.0.113.7
HTTP Request + Headers
ACK
ACK
…

(1)
198.51.100.3 → 203.0.113.7
HTTP Request + Headers
ACK
ACK
…

Anycast VIP
203.0.113.7