An Update on Source Address Validation Using BGP Updates, ASPA, and ROA" (BAR-SAV)

https://datatracker.ietf.org/doc/html/draft-sriram-sidrops-bar-sav-01

Presenter: Igor Lubashev

Authors: K. Sriram, I. Lubashev, and D. Montgomery

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

SIDROPS WG Meeting, IETF 115
November 2022

Outline of the Talk

- Recap of the BAR-SAV method
- Changes in version -01

Recap of the BAR-SAV Method

- Version-00 was presented at IETF 114 in Philly:

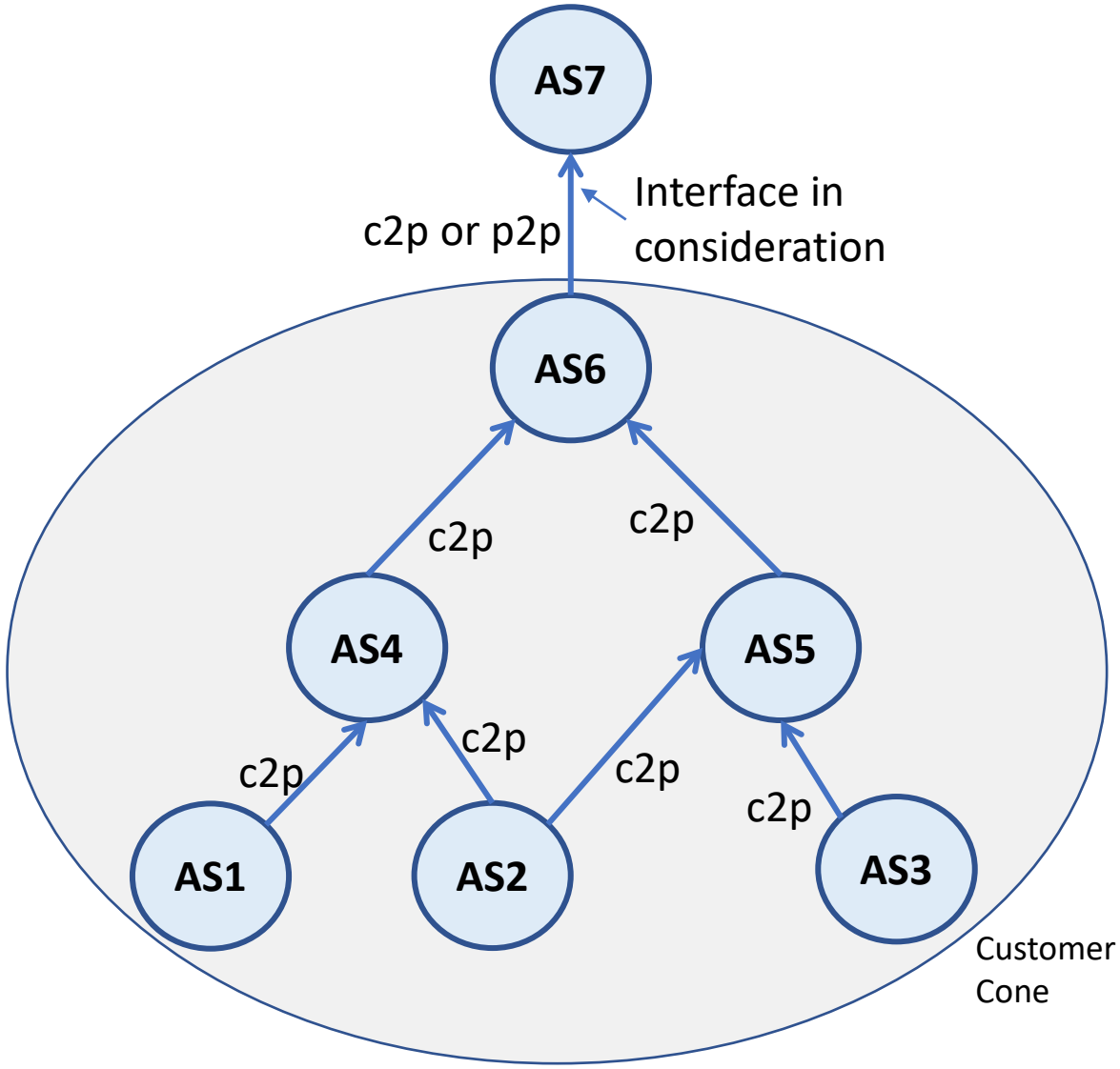
https://datatracker.ietf.org/meeting/114/materials/slides-114-sidrops-source-address-validation-using-bgp-updates-aspa-and-roa-bar-sav-00

It has more details than today's recap!

Overview of BAR-SAV

- History: BCP 38 → RFC 3704 (FP-RPF) → RFC 8704 (EFP-uRPF) → BAR-SAV
- BAR-SAV makes complementary use of BGP, ASPAs, and ROAs
  - Attempts to find all ASes in a customer cone (CC) using:
    AS PATH from BGP announcements and ASPA data
  - Attempts to find all prefixes in the CC using:
    PREFIX from BGP announcements and ROA data
- Overcomes barriers to accurate SAV filter design:
  - Hidden prefixes due to asymmetric routing, NO_EXPORT, etc.
- If a CC has full adoption of ASPA and ROA, BAR-SAV can provide a perfect SAV filter design using ASPA and ROA data alone

BAR-SAV Operation



1. **Customer Cone construction**
   Starting with the customer (or peer) ASN, iteratively obtain the set of ASNs using "customer-of" and "previous-AS" relationships in ASPAs and AS_PATHs.
2. **SAV Prefix List construction**
   a. Gather all prefixes in ROAs associated with the ASNs found in Step 1.
   b. Gather all prefixes in BGP UPDATE messages with originating ASN among ASNs found in Step 1.
   c. Combine sets found in Steps 2a and 2b. Keep only the unique prefixes. This is the permissible prefix list for SAV for the interface in consideration.

Complementary Nature of BGP, ASPA, and ROA

- There need not be widespread deployment of ROAs and ASPAs
- They help in cases when a CC AS or prefix is invisible in BGP
  - When an AS is not visible in BGP in a CC,
    a registered ASPA object can help locate that AS
  - When a prefix is not visible in BGP in a CC,
    a registered ROA object can help locate that prefix

Changes in Version-01

- There was good discussion and feedback at the mic at IETF 114
- Version -01 incorporates changes to address those comments

Key Version-01 Changes

- The following new sections have been added in -01

6. Operations and Management Considerations

    6.1. Applicability of ASPA and ROA

    6.2. BAR-SAV and Routing Policy

    6.3. Where to Deploy BAR-SAV

    6.4. Automation is the Key

    6.5. Implementation Guidelines

    6.5.1. Management of Local RPKI Repository Caches

    6.5.2. Management of Objects Temporarily Missing from RPKI Repositories

Applicability of ROA and ASPA Objects

- ROA and ASPA objects as currently specified seem sufficient
  - Both help uncover hidden prefixes (e.g., due to use of NO_EXPORT, DSR, other traffic engineering)
  - ROA max-length attribute is unused

- There was a suggestion to introduce SAV-specific ROA- and ASPA-like objects instead of using ROA and ASPA for SAV
  - The authors could not find examples that show why the current definitions of ROA and ASPA might be inadequate:
    - ✓ ASPA for path verification purposes fully meets SAV needs also
    - ✓ ROA “motivated by SAV needs” helps SAV and does not harm RPKI-ROV
  - We welcome further discussion and collaboration

Implementation Guidelines

- RPKI is not guaranteed to be 100% available or consistent
  - Implementations must fail open
  - RPKI-ROV use: fail → BGP works, but prefix hijacks are possible
  - SAV use: fail → data forwarding works, but src addr spoofing is possible

- If repository is unavailable, assume all previously valid signed objects are still valid (ignore expiration)
- If an unexpired signed object is no longer present in the repository, an implementation may still use it from a local cache till it expires (if it is not on a CRL)

Feedback welcome!

Encourage ASPA Adoption

- ASPA helps detect route leaks and forged-origin hijacks
- ASPA helps SAV filter design (BAR-SAV)

- Updated ASPA profile (v-11) and ASPA-based AS path verification (v-11) drafts have been published recently

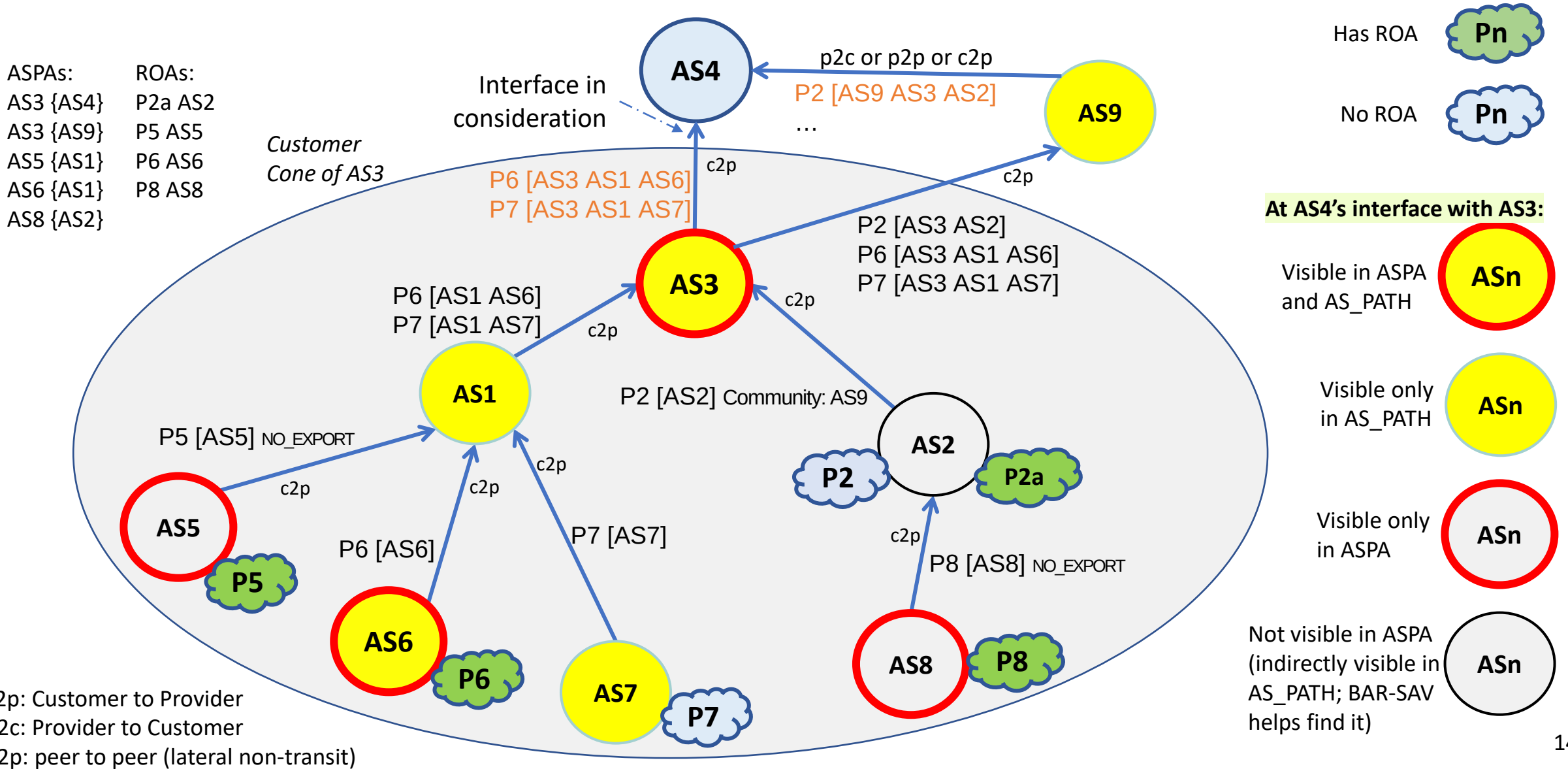
Conclusion – Requests for the WG

- Feedback is requested on the new section 6 "Operations and Management Considerations"
  - It was motivated by comments/feedback from IETF 114
  - Please, read this section and let us know if the comments have been addressed adequately

- **Working group adoption call request!**

Backup Sides

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



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

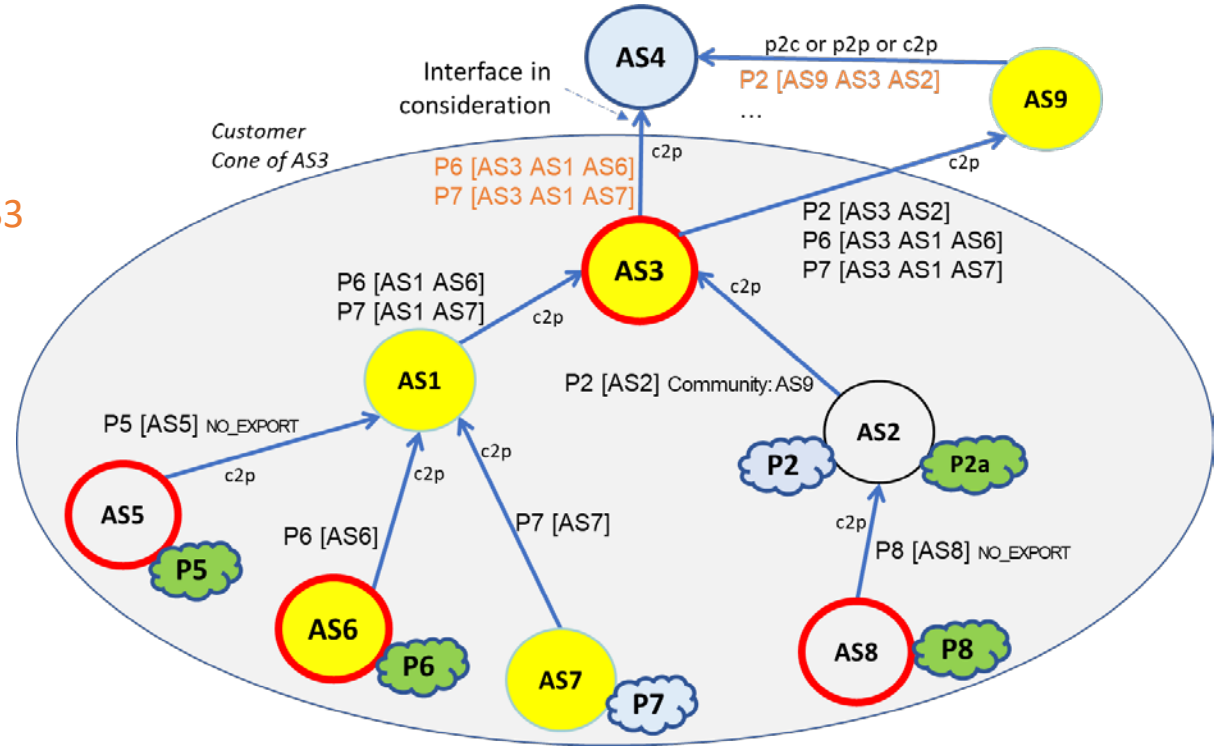
INPUTS

ASPAs:	ROAs:
AS3 {AS4, AS9}	P2a AS2
AS5 {AS1}	P5 AS5
AS6 {AS1}	P6 AS6
AS8 {AS2}	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, AS7, AS8	None	None

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

INPUTS

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

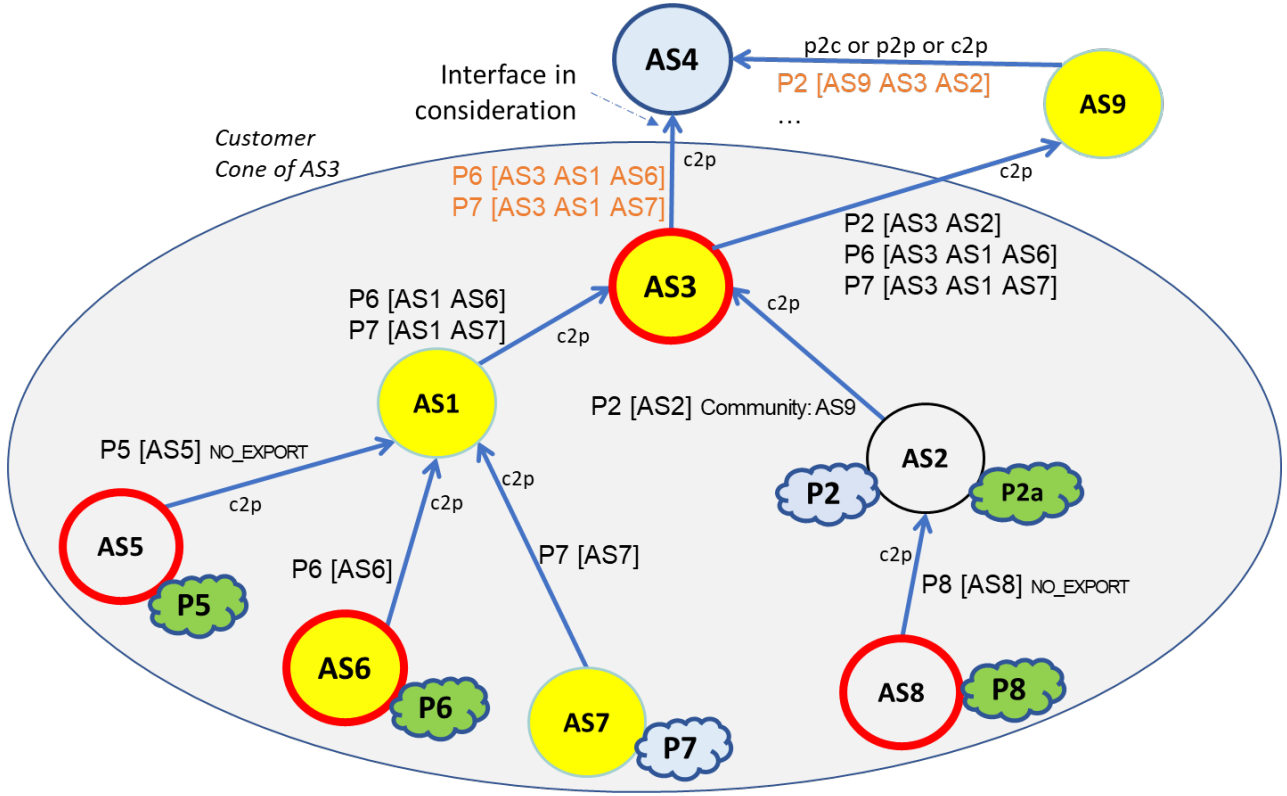
ROAs:
P2a AS2
P5 AS5
P6 AS6
P8 AS8

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

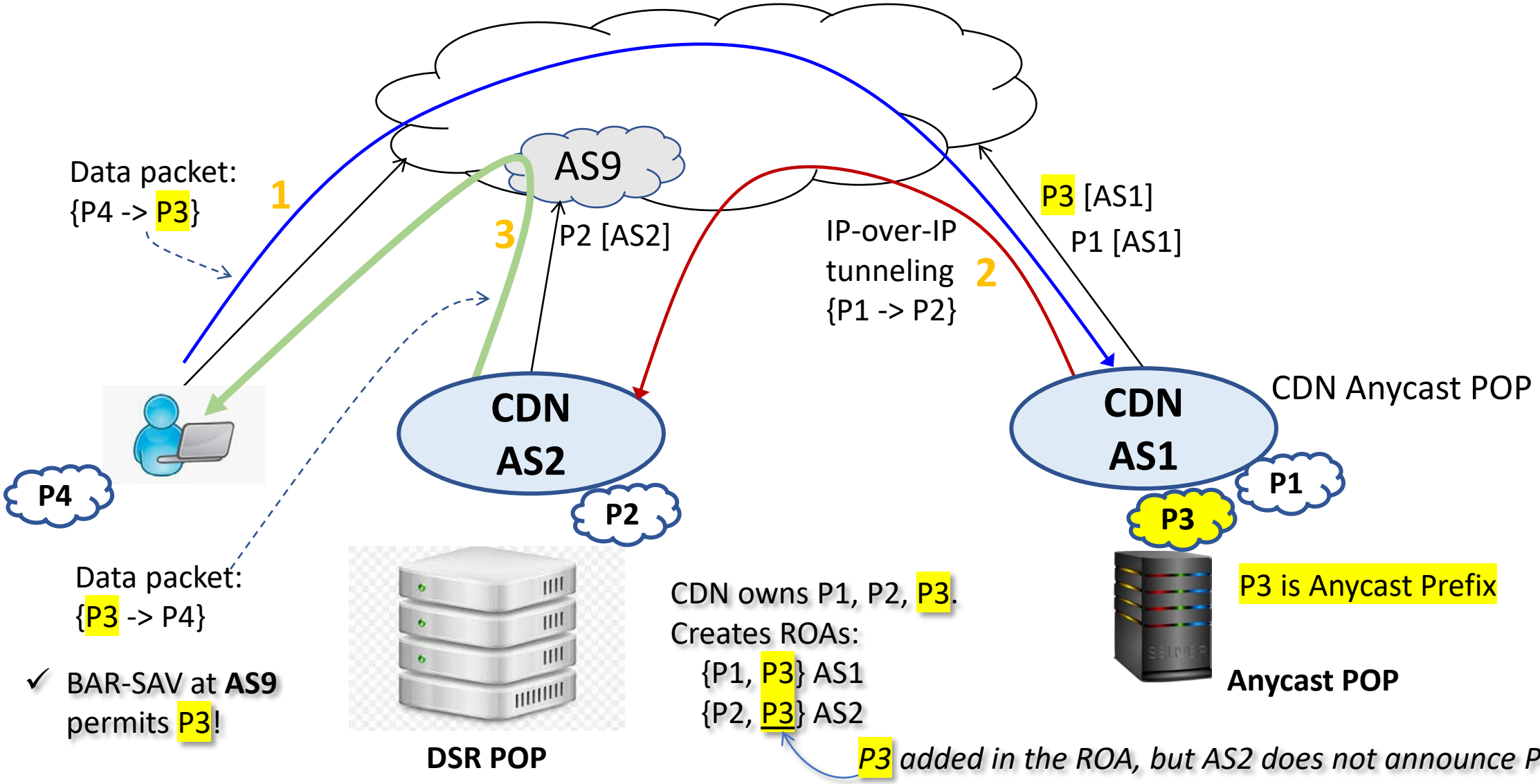
ASN	Prefixes from ROA	Prefixes from BGP
AS1		
AS2	(P2a AS2) → P2a	P2 [AS9 AS3 AS2] → P2
AS3		
AS5	(P5 AS5) → P5	
AS6	(P6 AS6) → P6	P6 [AS3 AS1 AS6] → P6
AS7		P7 [AS3 AS1 AS7] → P7
AS8	(P8 AS8) → P8	



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

Content Delivery Network (CDN) Application

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



✓ BAR-SAV at **AS9** permits P3!