Bicone Source Address Validation

Dan Li, **Lancheng Qin**, Li Chen, Libin Liu

November 2024

Background

- Bicone SAV is an inter-domain SAV solution which generates and uses a blocklist SAV filter
  - The blocklist is used on interfaces facing a customer AS or a lateral peer AS
  - It helps **avoid improper blocks** (i.e., blocking legitimate traffic) in the case of limited propagation of prefixes in customer cone (e.g., NO_EXPORT case [1, 2])
- Historical versions
  - draft-li-sidrops-bicone-sav-00, IETF 119 SAVNET WG & SIDROPS WG
  - draft-li-sidrops-bicone-sav-01, June, 2024
  - draft-li-sidrops-bicone-sav-02, IETF 120 SIDROPS WG
  - draft-li-sidrops-bicone-sav-03, July, 2024
  - draft-li-sidrops-bicone-sav-04, July, 2024
  - **Adoption call in SIDROPS WG, 19 August 2024 - 3 September 2024**
  - **Moved from SIDROPS to SAVNET by agreement of Chairs and ADs**
  - **draft-li-sidrops-bicone-sav-05, IETF 121 SAVNET WG**

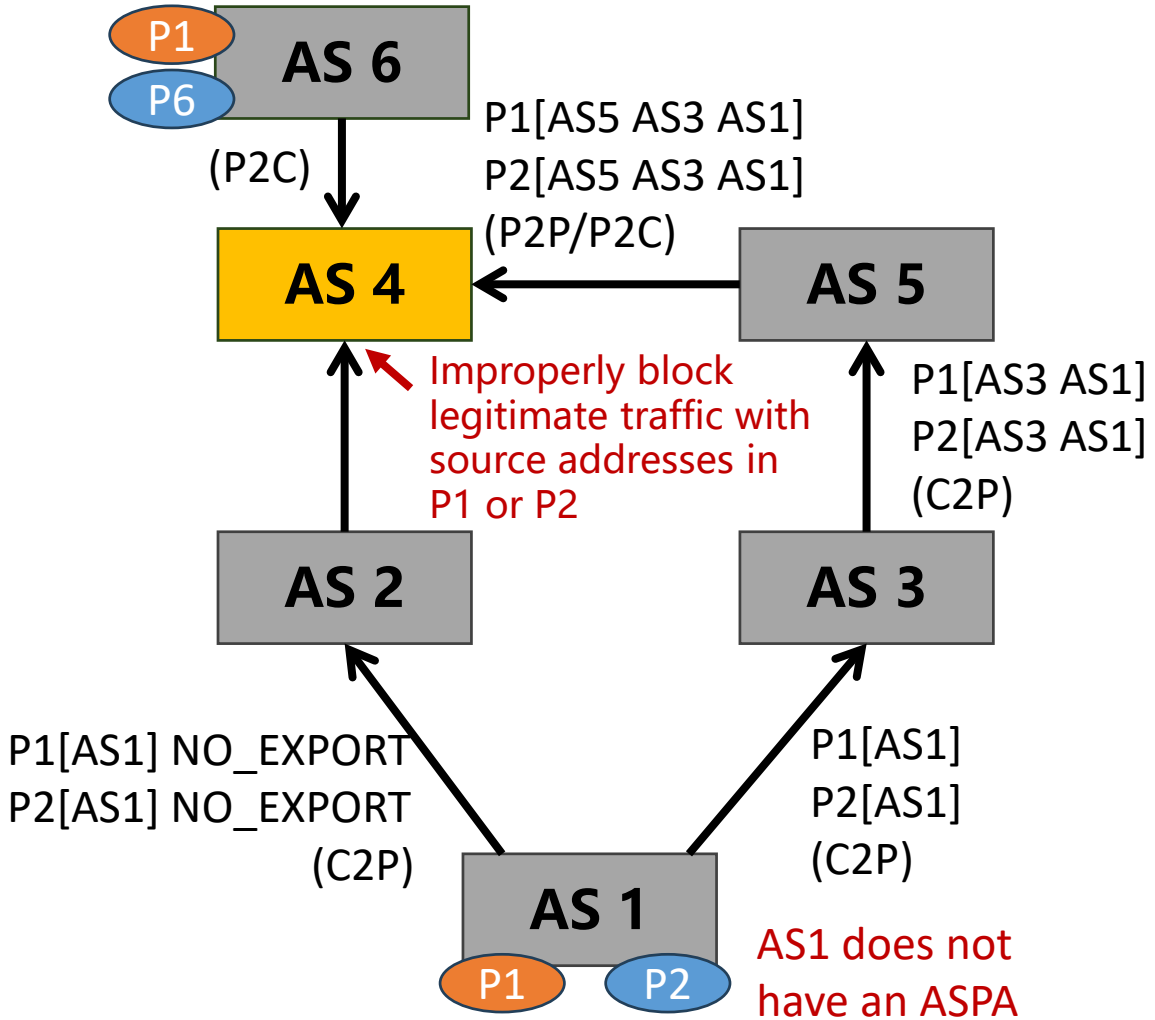
[1] Section 3.3 and Figure 4 in RFC 8704
[2] Section 4.1.1 and Figure 3 in draft-ietf-savnet-inter-domain-problem-statement-05

Comments during Adoption Call in SIDROPS WG

- During the adoption call, we have received **significant support from many people** [1]
  - This document is moved to SAVNET by agreement of Chairs and ADs
- Technical comments:
  - It will still have improper blocking – in some multihoming scenarios with prefixes that are invisible in the CC [from Sriram]. This situation would result in an **improper block** of the "invisible" prefix **by both bar-sav and bicone** [from Igor]
    - **Response**: We update the design of blocklist generation in Section 5 to **further avoid improper blocks**
  - It currently admits all special purpose prefixes, unallocated prefixes, and allocated but unannounced prefixes (not intended to be announced; having no ROAs) [from Sriram]
    - **Response**: We update Section 7.3 to provide implementation and operations recommendations to **reduce improper admits**

[1] https://mailarchive.ietf.org/arch/msg/sidrops/MTWoV6nv3pl0oIqSg_JwGLhYBJg/

Improper Block Problem of Using An Allowlist



An example of limited propagation of prefixes in the customer cone

Ingress SAV filtering on AS4-AS2 Interface

- AS1 (a multi-homed customer AS) attaches NO_EXPORT to all prefixes announced to AS2
- AS4 never receives routes for P1 and P2 from its customer AS2
- EFP-uRPF Algorithm A and Algorithm B on AS4-AS2 Interface have **improper block problems**
- More recent SAV solutions (e.g., BAR-SAV) additionally **use ASPAs and ROAs**
  - **If some ASes (e.g., AS1) do not have APSAs, it still has improper blocks**

Two Goals of Bicone SAV

☐ Goal #1: Avoid improper block

◆Existing allowlist-based SAV solutions may have improper block problems in NO_EXPORT and DSR cases, Bicone SAV aims to avoid improper blocks in these cases

➢Blocklist should be better than Allowlist in achieving Goal #1

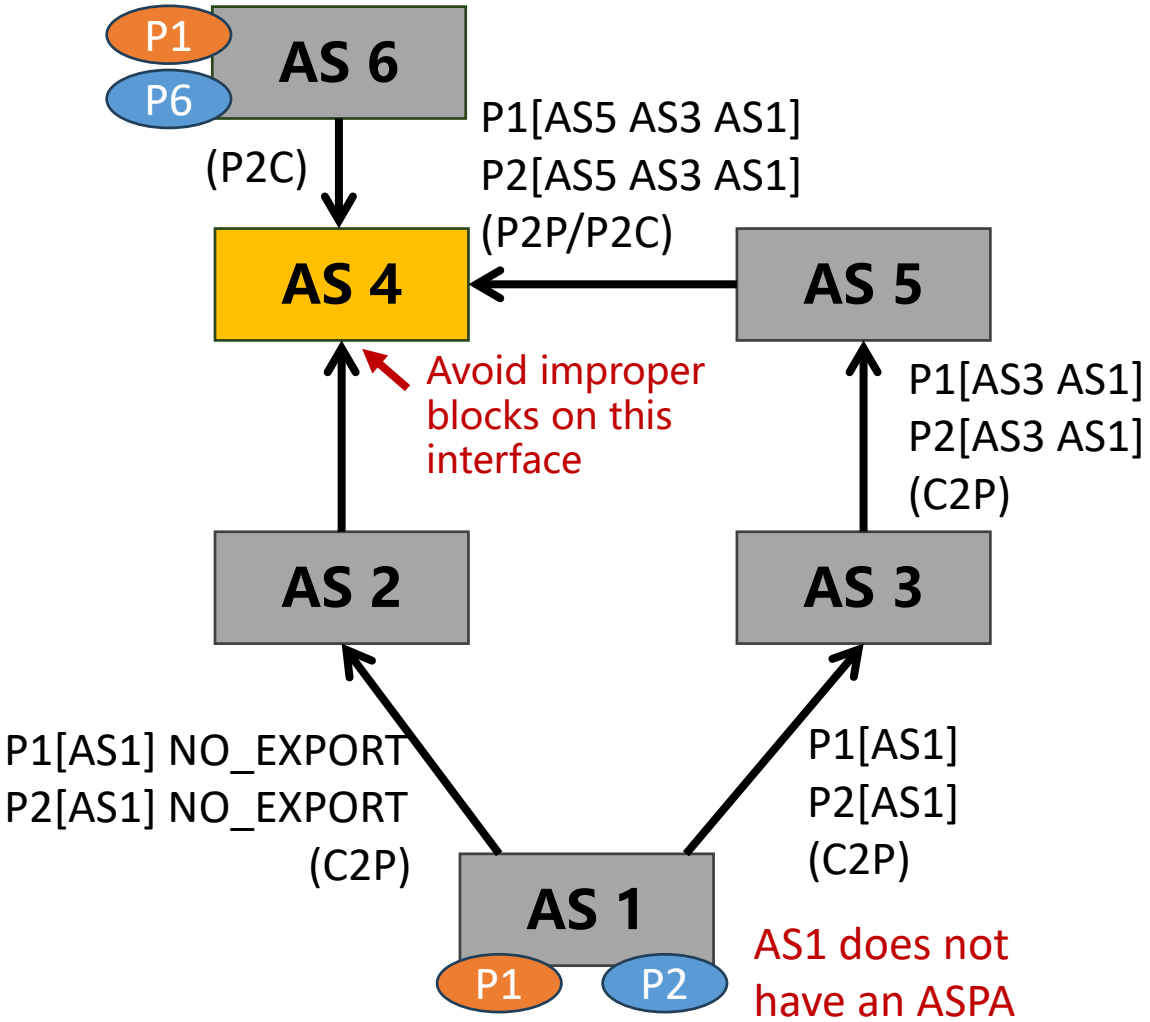
☐ Goal #2: Maintain directionality

◆Unlike Loose uRPF [RFC3704] which completely loses directionality, Bicone SAV aims to identify more source-spoofed data packets than Loose uRPF by maintaining directionality

➢Allowlist should be better than Blocklist in achieving Goal #2

Goal #1 is more important because for many network operators, improperly blocking legitimate traffic is unacceptable

Key Idea of Bicone SAV



An example of limited propagation of prefixes in the customer cone

- Blocklist contains **prefixes only belonging to the Provider Cone**
  - These prefixes should not be used as source addresses in data packets received from any customer AS or lateral peer AS unless there is a route leak
- Use Loose uRPF and blocklist together on interfaces facing a customer AS or a lateral peer AS
  - Loose uRPF blocks data packets using **source addresses of unallocated prefixes, and allocated but unannounced prefixes**
  - Blocklist blocks data packets using **source addresses only belonging to Provider Cone** (e.g., P6 in the example)

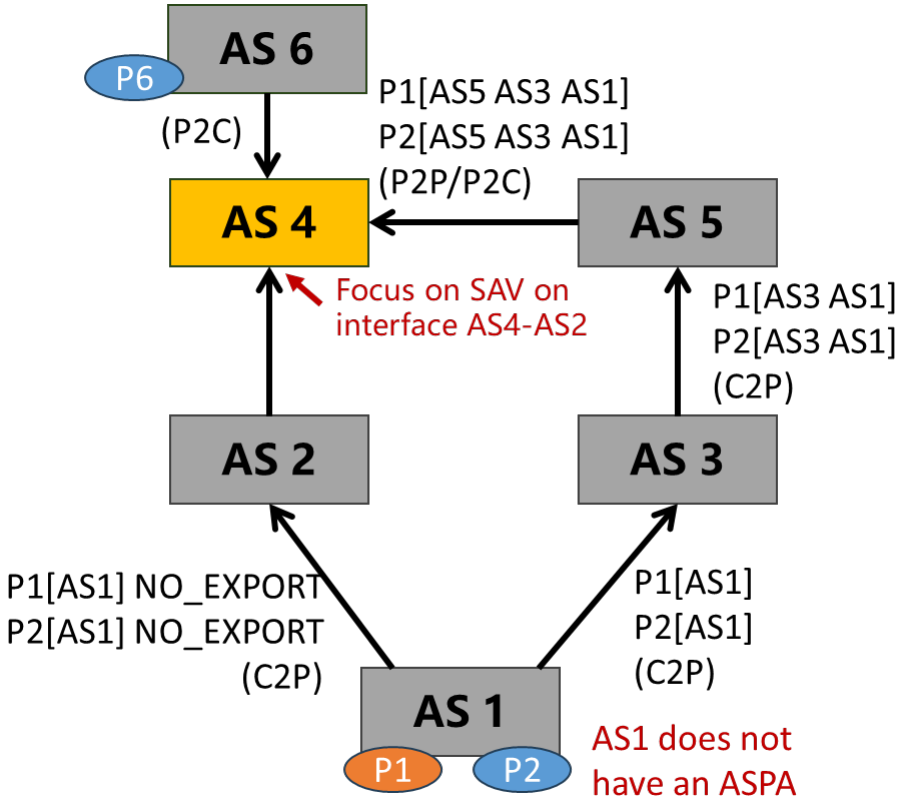
Blocklist Generation

- Step 1: Identify as many ASes in the provider cone as possible by using BGP UPDATES (AS_PATH) and ASPAs
- Step 2: Identify prefixes belonging to these ASes by using BGP UPDATES (ORIGIN and PREFIXES) and ROAs
- Step 3: Remove prefixes that **MAY** also belong to any customer cone
  - Since an AS may not know if a prefix is belonging to its customer cone, to avoid any potential improper blocks, bicone SAV removes prefixes that are belonging to both ASes in the provider cone and ASes not in the provider cone

More details can be found in Section 5 of this document

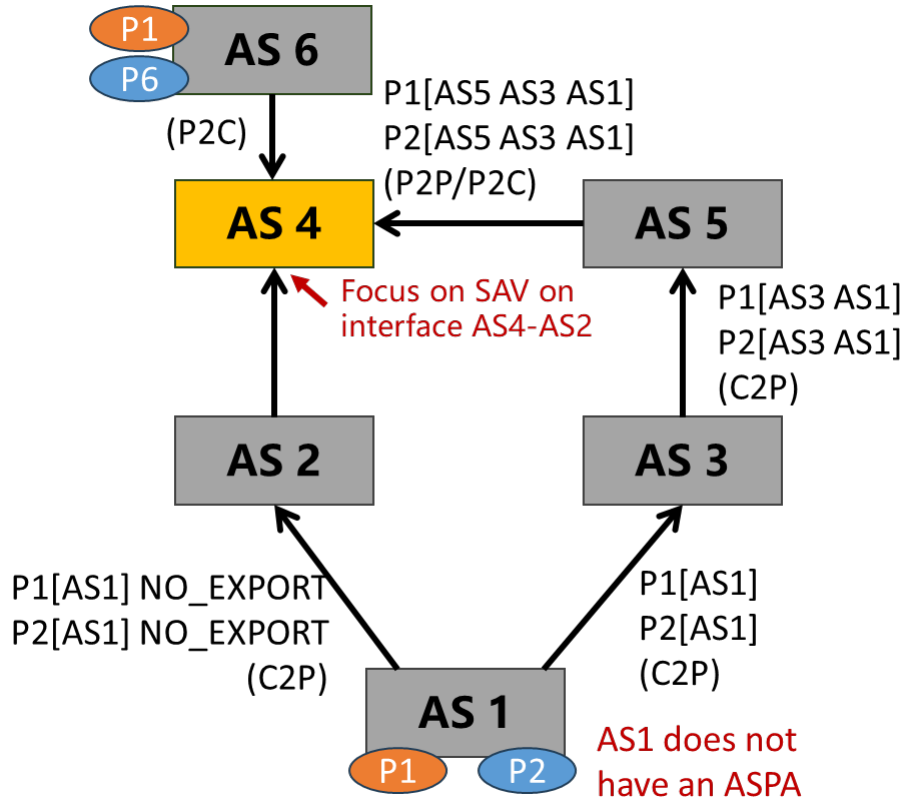
Compare Bicone SAV with Existing SAV Solutions

Scenario #1: Prefix P1 is single-homed to AS1



Allowlist-based SAV	**Improper blocks**
Loose uRPF	**Too many improper admits**
Bicone SAV + Loose uRPF	**No improper block** and **less improper admits** than Loose uRPF ✓

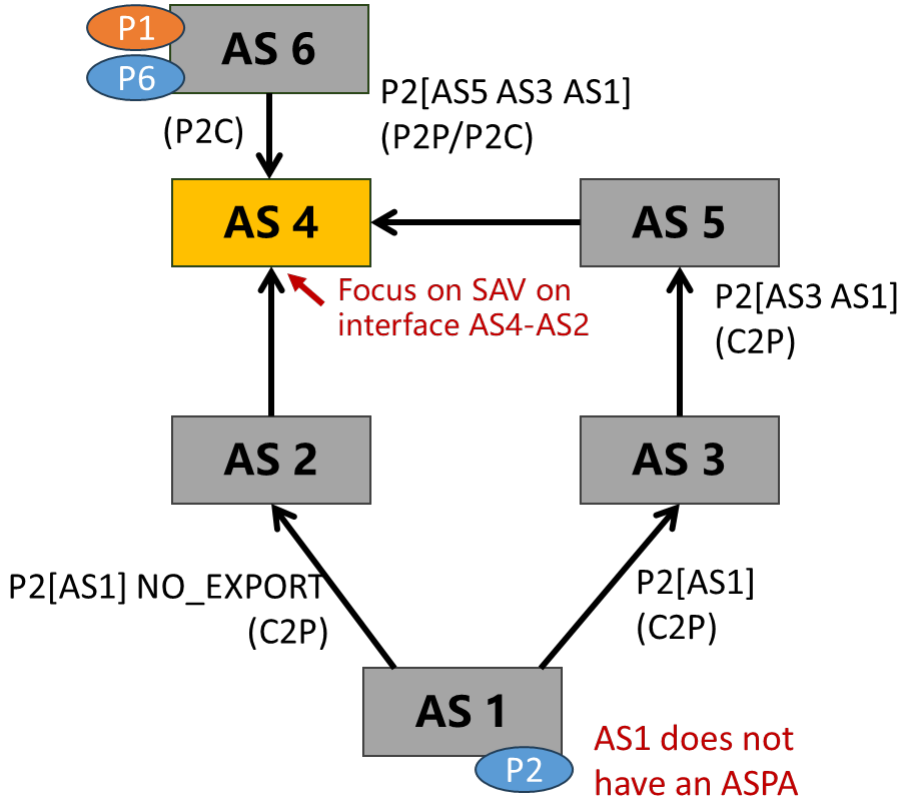
Scenario #2: Prefix P1 is multi-homed to AS1 and AS6



Allowlist-based SAV	**Improper blocks**
Loose uRPF	**Too many improper admits**
Bicone SAV + Loose uRPF	**No improper block** and **less improper admits** than Loose uRPF ✓

Compare Bicone SAV with Existing SAV Solutions

Challenging Scenario #3: Prefix P1 is single-homed to AS6 in BGP UPDATES and ROAs, but AS1 will use source addresses of prefix P1 (e.g., DSR case)



Allowlist-based SAV	**Improper blocks**
Loose uRPF	**Too many improper admits**
Bicone SAV + Loose uRPF	**Improper blocks**

Possible solution

Manually add DSR prefixes into allowlist or remove DSR prefixes from blocklist

Recommendations

- If the network operator can determine that the allowlist is complete, it is recommended to use an allowlist because the complete allowlist would have neither improper blocks nor improper admits
- If the network operator cannot, it is recommended to use blocklist + Loose uRPF to avoid improper blocks
- Network operators are allowed to **manually modify or configure the blocklist** according to their local knowledge
  - ◆Add special purpose prefixes that will not be used as source addresses by data packets or remove prefixes that would be used for anycast and DSR

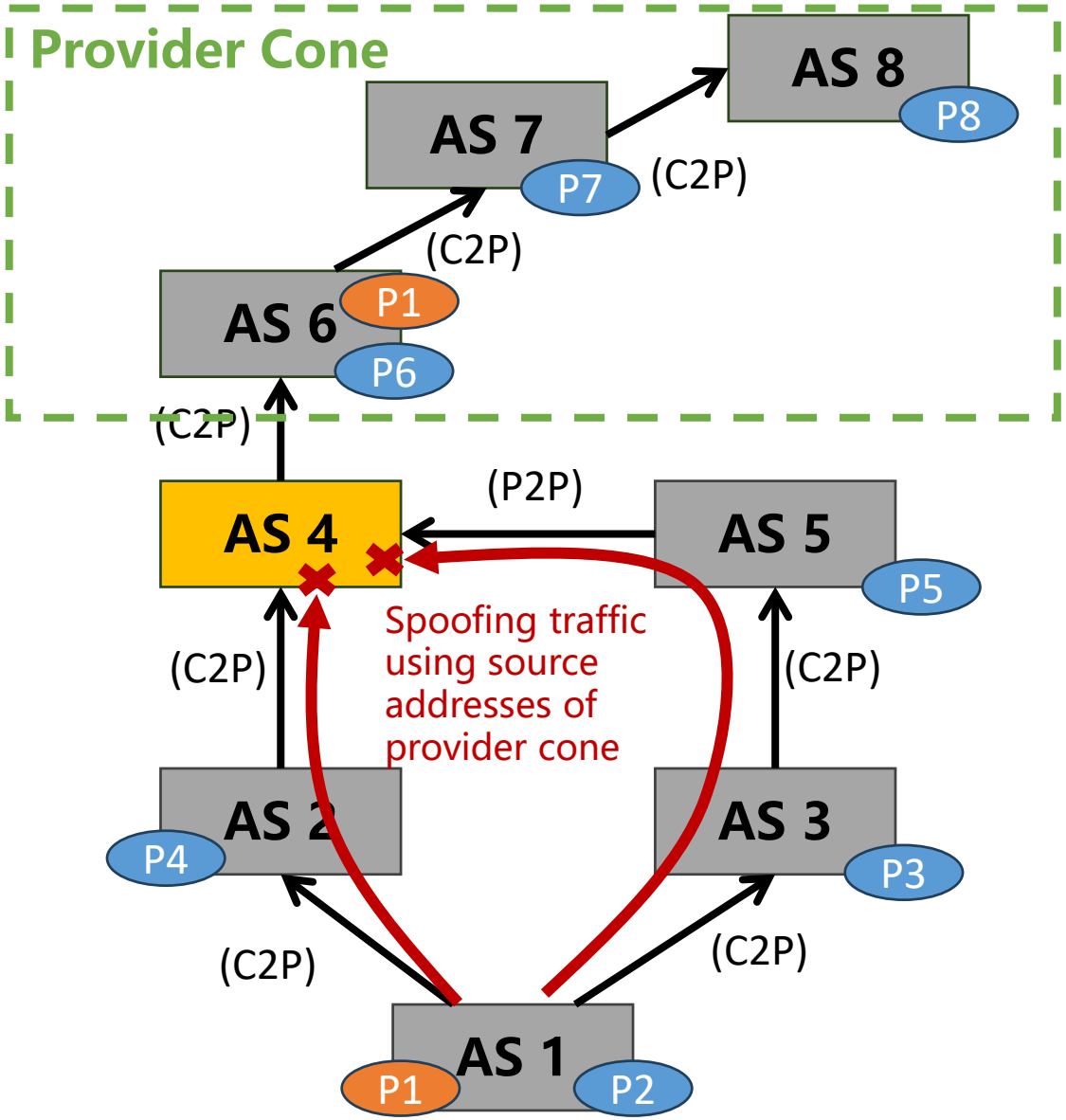
Thanks!

Backup

Adoption Call in SIDROPS WG

- Conclusion of adoption call
  - Chairs of SIDROPS WG: "There is **significant support for this work**, but several people have spoken against the work being done in the SIDROPS WG. Therefore, this document is **not being adopted in this WG (SIDROPS WG)**. We have started a conversation with two Area Directors to find a home for this work."
  - Chairs of SAVNET WG: "after consulting with both the ADs and Chairs of the SAVNET/SIDROPS WG, we determined that these two documents **should be relocated and discussed within the SAVNET WG**. The Bicone-SAV draft will be kept as one individual document, and **will be waited later opportunity to be adopted**, or combined with other documents to be forwarded together."

Example: Ingress SAV Blocklist Filter



Ingress SAV filtering on AS4

- Assume provider cone of AS4 includes AS6, AS7, and AS8
- If AS4 identifies all prefixes (i.e., P6, P7, and P8) only belonging to the provider cone
  - ◆Block data packets received from AS2 and AS5 with source addresses in P6, P7 and P8
- If AS4 only identifies partial prefixes (e.g., P6, P7) only belonging to the provider cone
  - ◆Block data packets received from AS2 and AS5 with source addresses in P6 and P7
  - ◆**Avoid improper block**