Update on the ASPA Verification Draft

https://datatracker.ietf.org/doc/draft-ietf-sidrops-aspa-verification/

K. Sriram
(in collaboration with co-authors, Claudio)

IETF SIDROPS Interim Meeting
June 2024

Requests for improvements from WG members (1/2)

- Expand explanations of the basic principles behind the algorithms (Matthias, Tassilo, Claudio)
  - Better understanding of how Invalid, Valid, Unknown are determined
  - Rationale for separate upstream/downstream algorithms

Requests for improvements from WG members (2/2)

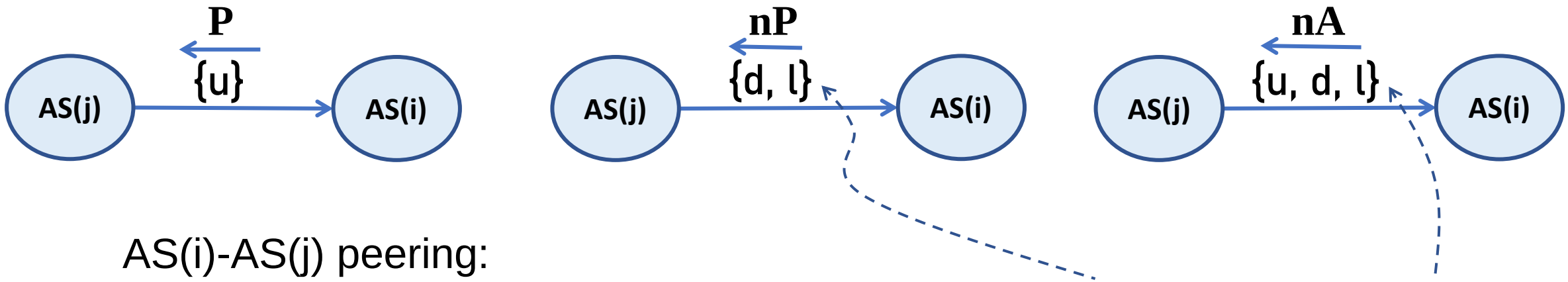
- Make the Security Considerations section more solid and substantial (Ruediger)
  - Now we have 3 types of shortcomings explained instead of one in v-17
- Mutual Transit and Complex peering -- make it clear if we are defining Roles or just BGP session types (Claudio)

ASPA Hop Check Function

Definition:

P: Provider
nP: not Provider
nA: no Attestation

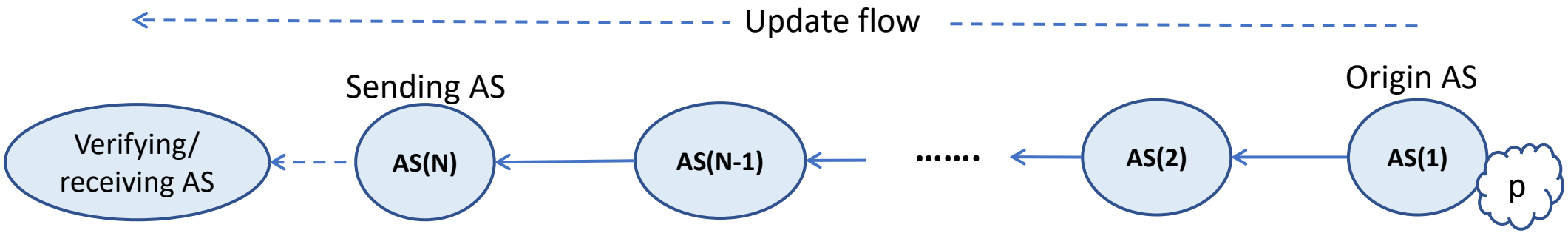
$$\text{hop}(AS(i), AS(j)) = \begin{cases} \mathbf{P} & \text{if AS(i) attests AS(j) is a provider} \\ \mathbf{nP} & \text{if AS(i) attests AS(j) is not a provider} \\ \mathbf{nA} & \text{if AS(i) does not have an ASPA} \end{cases}$$



AS(i)-AS(j) peering:
u = Up (customer to provider (C2P))
d = Down (provider to customer (P2C))
l = Lateral (peer to peer (p2p))

allowed peering relations

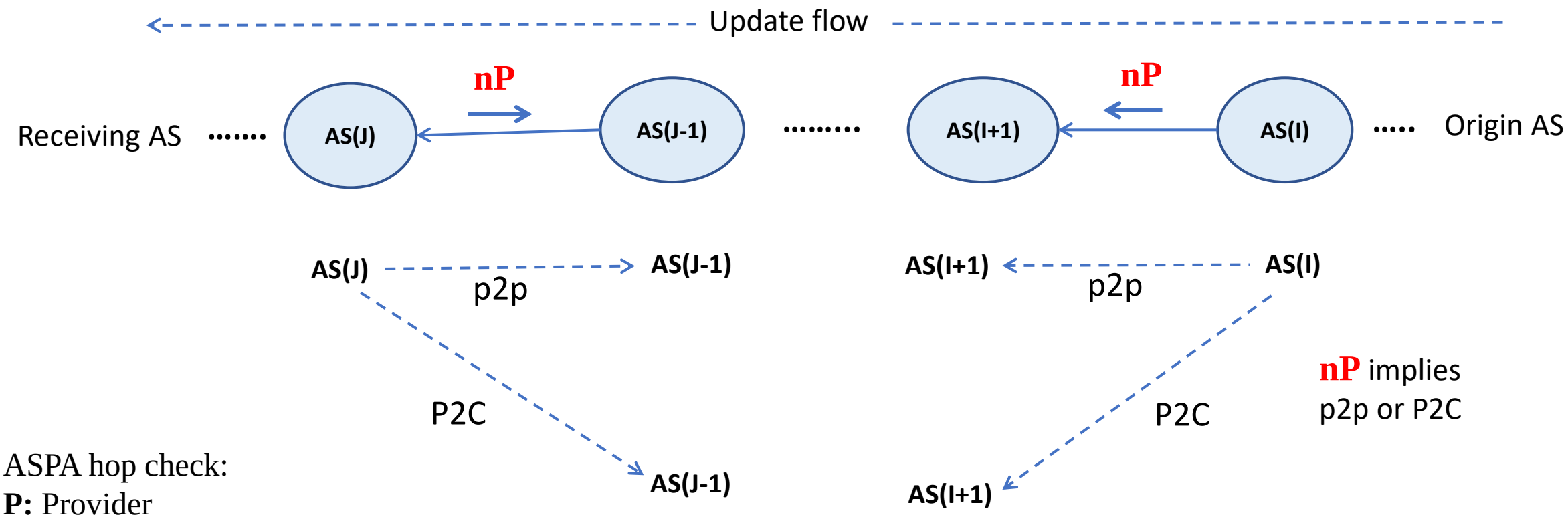
AS_PATH representation for ASPA Verification



AS_PATH: {AS(N), AS(N-1), …., AS(2), AS(1)}

- Unique ASes
- AS(1) is the origin AS
- AS(N) is the most-recently added/sending AS

AS Path Verification: Invalid Outcome (any AS Path)

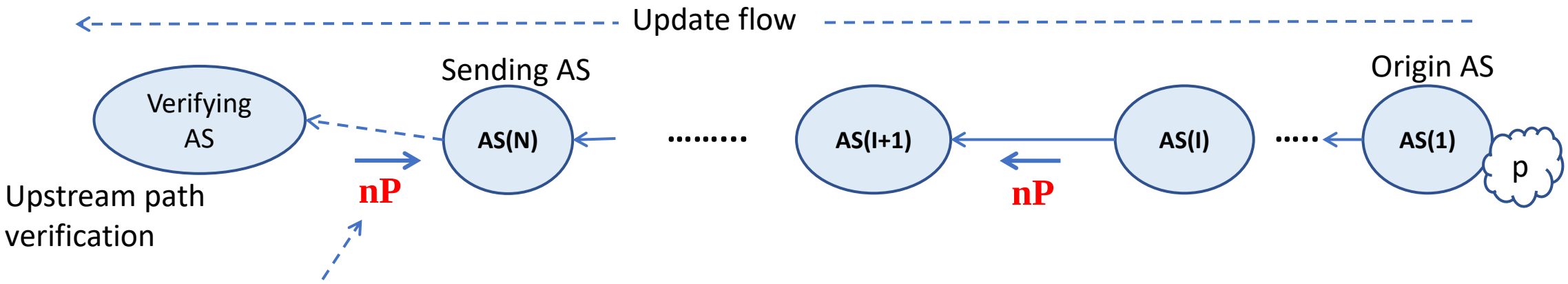


ASPA hop check:
P: Provider
nP: not Provider
nA: no Attestation

P2C = Provider-to-customer
C2P = Customer-to-provider
p2p = peer-to-peer (lateral peers)

- AS path is **Invalid** if any two hops in opposite directions (facing each other) are **nP** per ASPA ($J - I \geq 2$)
- Else, the AS_PATH is not Invalid (i.e., it is Valid or Unknown)

Verification of Upstream Paths: Invalid Outcome

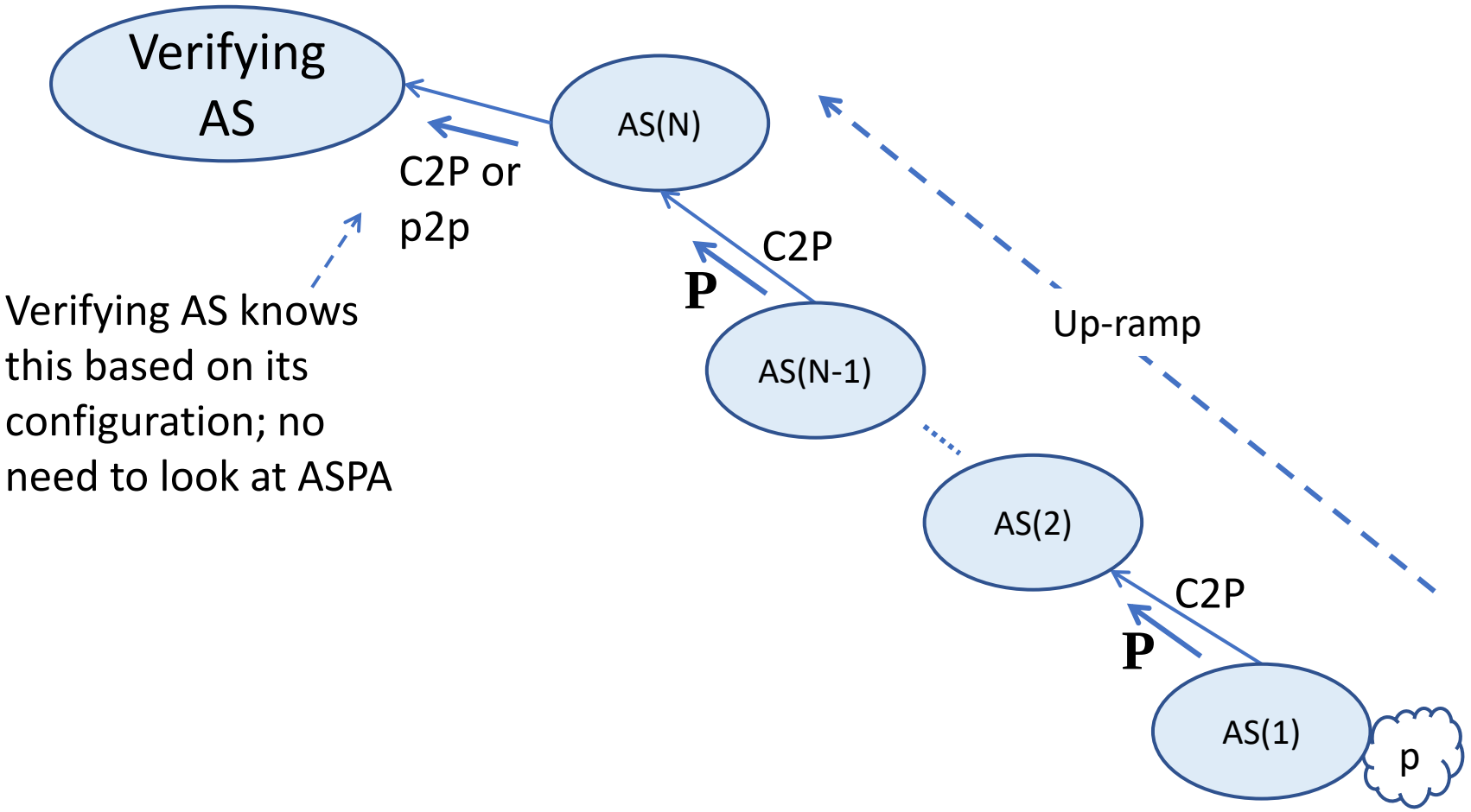


ASPA hop check:
P: Provider
nP: not Provider
nA: no Attestation

P2C = Provider-to-customer
C2P = Customer-to-provider
p2p = peer-to-peer (lateral peers)

- AS_PATH is **Invalid** if any one hop AS(I) to AS(I+1) is **nP** per ASPA for I = 1, 2, …, N-1
- Else, the AS_PATH is not Invalid (i.e., it is Valid or Unknown).

Verification of Upstream Paths: Valid Outcome



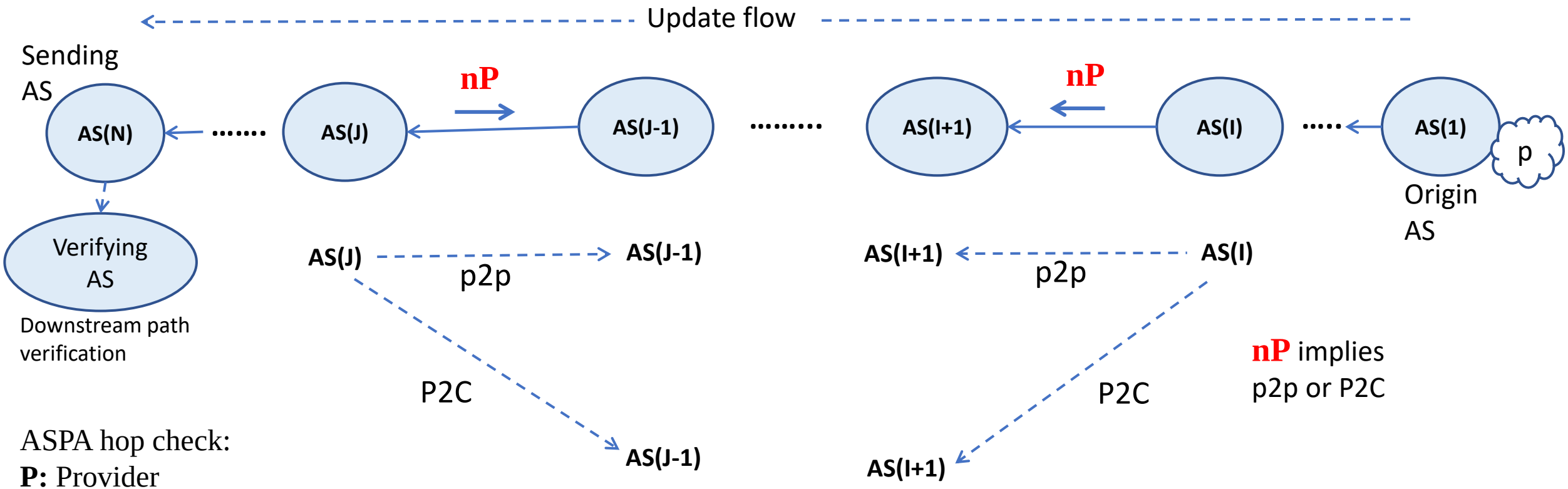
ASPA hop check:
P: Provider
nP: not Provider
nA: no Attestation

- AS_PATH is **Valid** only if all hops AS(I) to AS(I+1) are **P** (i.e., C2P) per ASPA for I = 1, 2, …, N-1

Verification of Upstream Paths: Unknown Outcome

In partial deployment, an Unknown outcome occurs when the available ASPA's do not produce an Invalid (slide 4) or Valid (slide 5) outcome for the Upstream AS_PATH.

Verification of Downstream Paths: Invalid Outcome



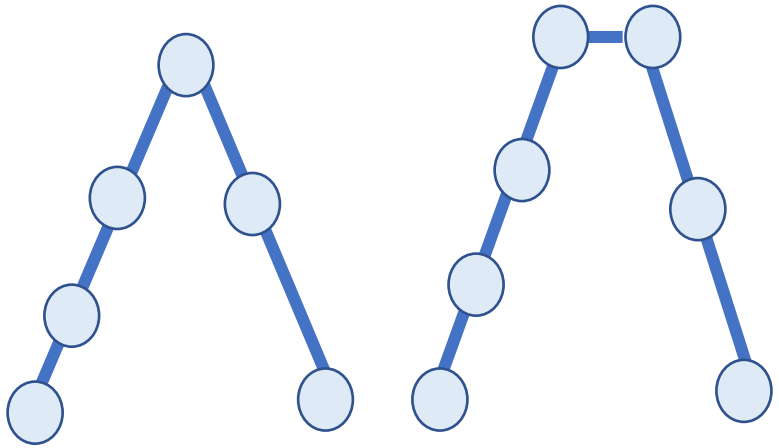
ASPA hop check:
P: Provider
nP: not Provider
nA: no Attestation

P2C = Provider-to-customer
C2P = Customer-to-provider
p2p = peer-to-peer (lateral peers)

- AS_PATH is **Invalid** if any two hops in opposite directions (facing each other) are **nP** per ASPA ($J - I \geq 2$).
- Else, the AS_PATH is not Invalid. Proceed to slide 2.

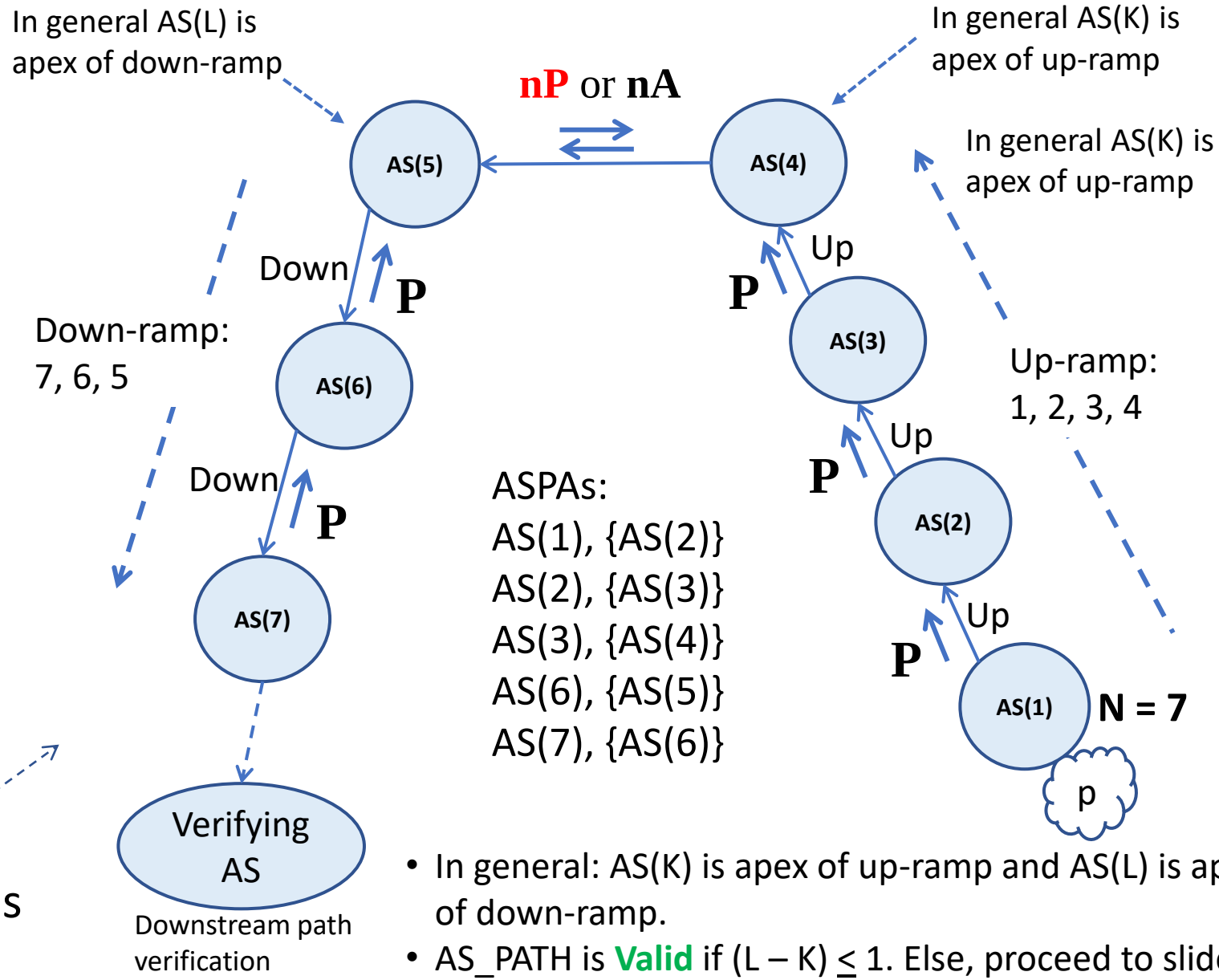
Verification of Downstream Paths: Valid Outcome

The only permissible path trajectories for **Valid** outcome are an inverted V or inverted V with a one hop p2p at the apex



details

ASPA hop check:
P: Provider
nP: not Provider
nA: no Attestation



- In general: AS(K) is apex of up-ramp and AS(L) is apex of down-ramp.
- AS_PATH is **Valid** if (L – K) ≤ 1. Else, proceed to slide 3.

Verification of Downstream Paths: Unknown Outcome

In partial deployment, an Unknown outcome occurs when the available ASPA's do not produce an Invalid (slide 7) or Valid (slide 8) outcome for the Downstream AS_PATH.

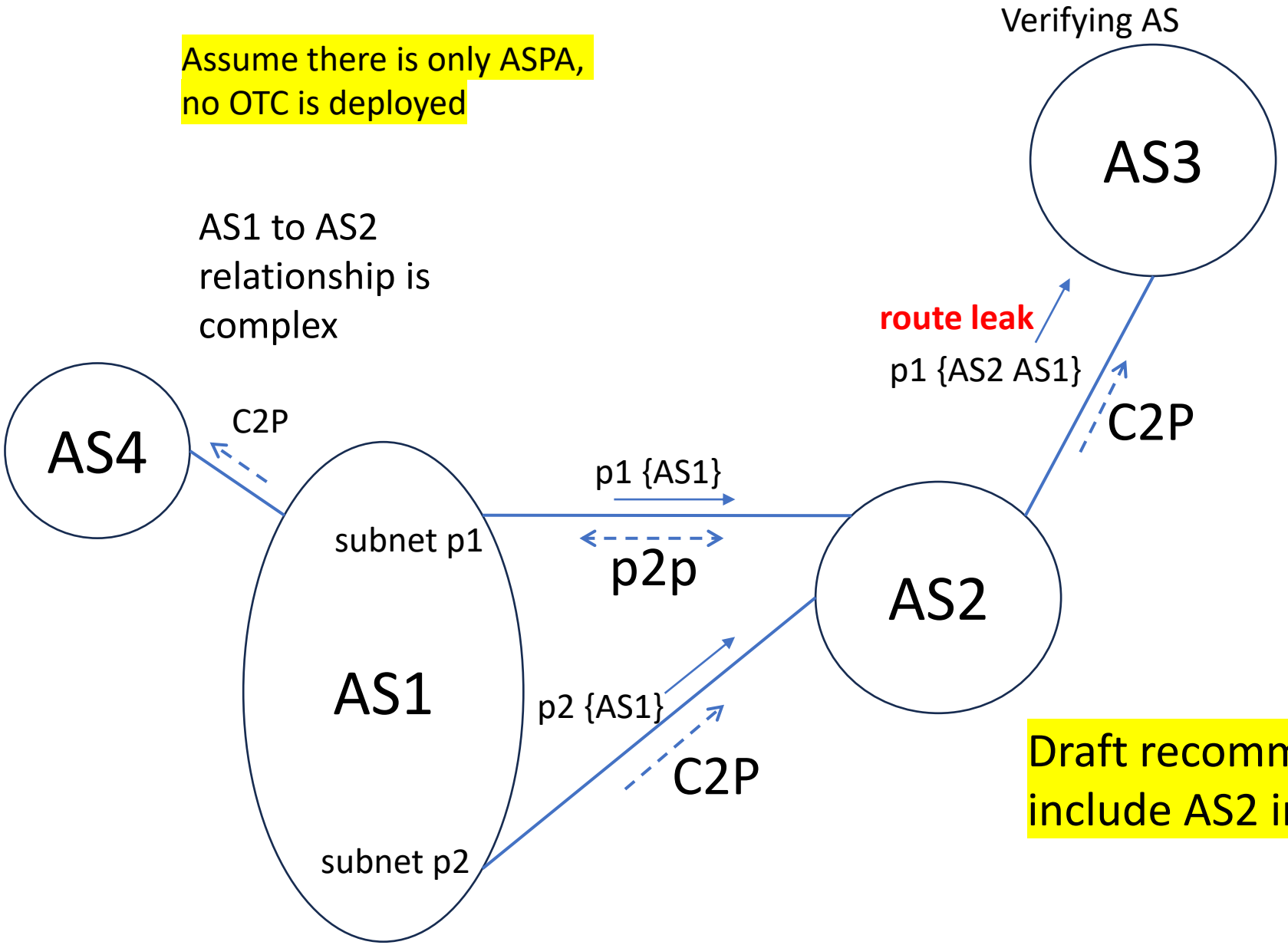
Requirements for ASes in Mutual Transit (MT) Relationship

- Each of the two ASes in a mutual-transit pair MUST register its ASPA including the other AS in its SPAS
- The implementation of [RFC9234] procedures is RECOMMENDED to complement the ASPA-based AS_PATH verification. However, if implementing [RFC9234] procedures, they MUST NOT be applied for the mutual-transit relationship.
  - Reason (for the MUST NOT): Mutual-transit (MT) ASes MAY export everything (both customer and non-customer routes) to each other.

ASPA and Complex / Mutual-Transit Peering Relationship Discussion

Complex relationship (excluding mutual transit)

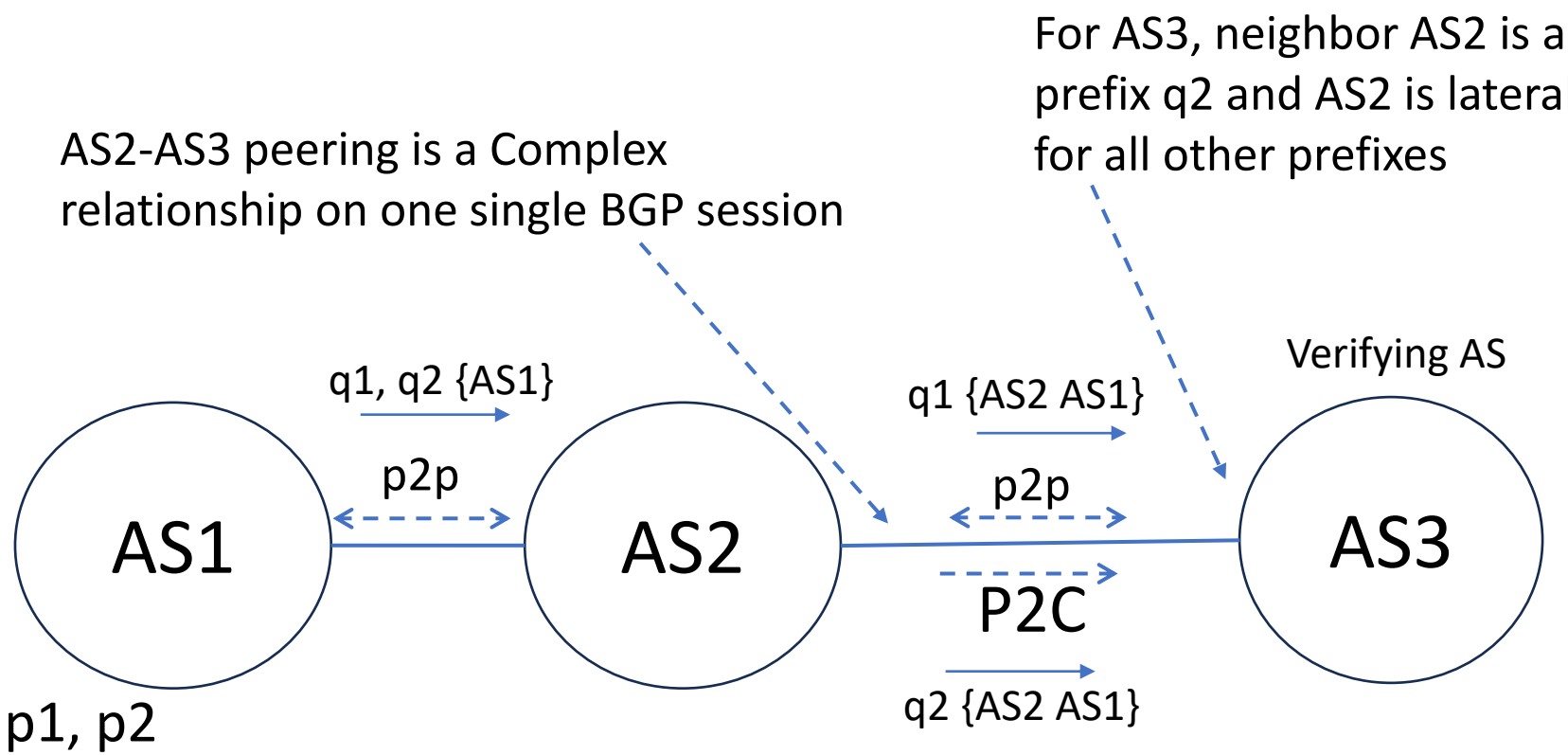
Assume there is only ASPA, no OTC is deployed



If AS1 includes AS2 in its ASPA, then the route p1 {AS2 AS1} will be Valid (false negative) at AS3 per ASPA. If AS1 does not include AS2 in its ASPA, then route p2 {AS2 AS1} will be Invalid (false positive) which is worse.

Draft recommendation: AS1 MUST include AS2 in its ASPA.

Complex relationship on one single BGP session (i.e., not segregated BGP sessions)

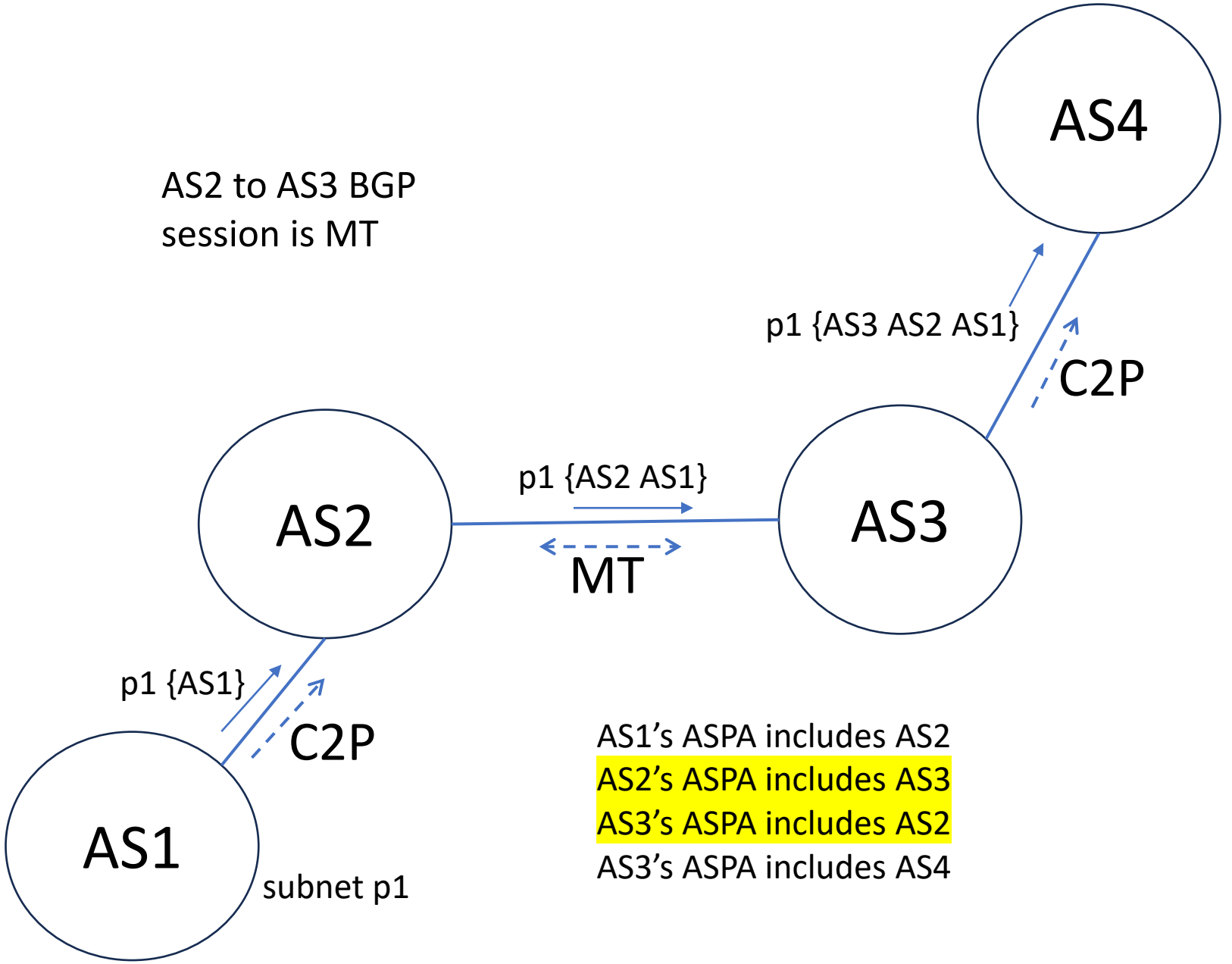


AS1 has ASPA and it does not include AS2

Proposed text in the ASPA Draft:
It is NOT RECOMMENDED to perform ASPA-based path verification on a complex BGP session. Configuring per-prefix ASPA-based AS path verification on such sessions is left to operator discretion.

AS2 leaks q1 to AS3 over the p2p link.
If AS3 applies the alg. for downstream paths for all prefixes, it cannot catch the route leak.
If AS3 applies the alg. for upstream paths for all prefixes, it will consider q2 {AS2 AS1} Invalid (false positive) .

Easy to make ASPA work in the case of Mutual Transit (MT) session



The route p1 {AS3 AS2 AS1} will be Valid per ASPA at AS4 which is correct.
Also, if route for p4 is advertised by AS4, and received at AS1:
p4 {AS2, AS3, AS4},
that route will also be Valid per ASPA (also correct).

- Mutual-transit (MT) ASes MAY export everything (both customer and non-customer routes) to each other.
- An AS in a MT relationship MUST NOT send BGP Role Capability to its counterpart (RFC9234 does not cover it)
- On a MT BGP connection, the sending AS MUST NOT perform the egress OTC Attribute processing and the receiving AS MUST NOT perform the ingress OTC Attribute processing. However, any existing OTC Attribute is retained.

Example:

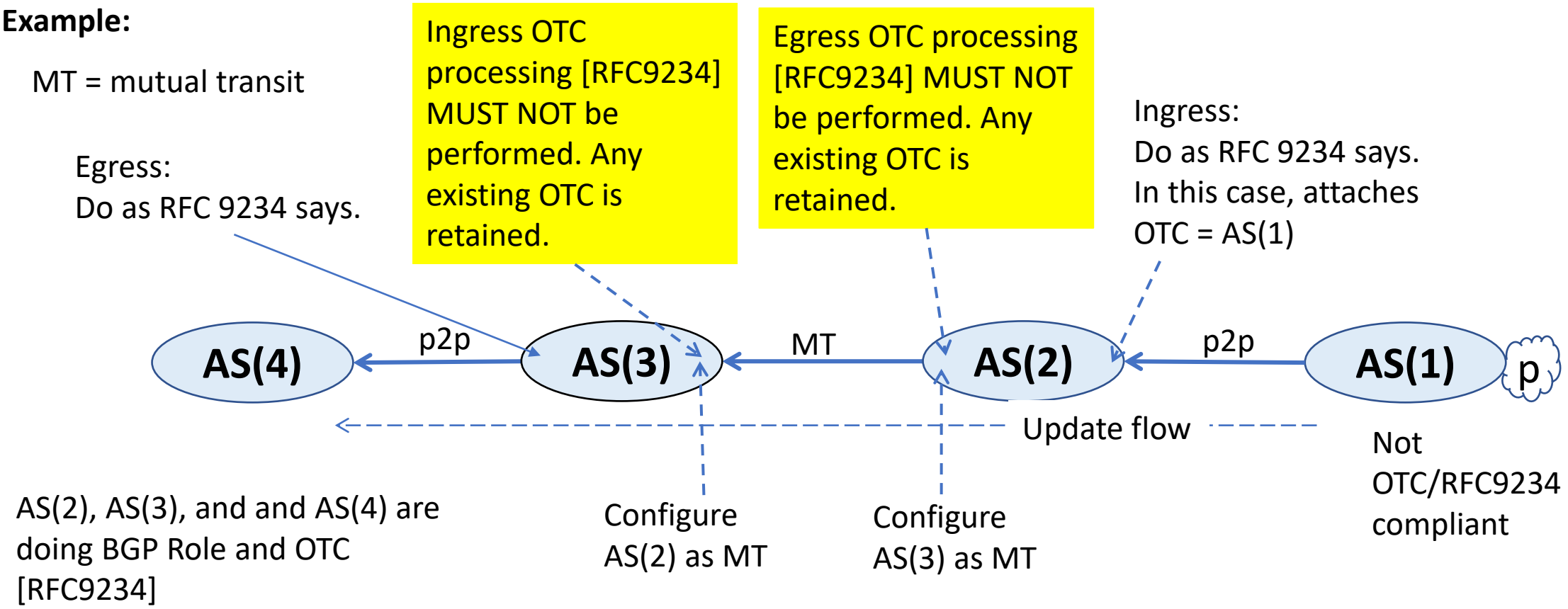


Fig. 1

Example:

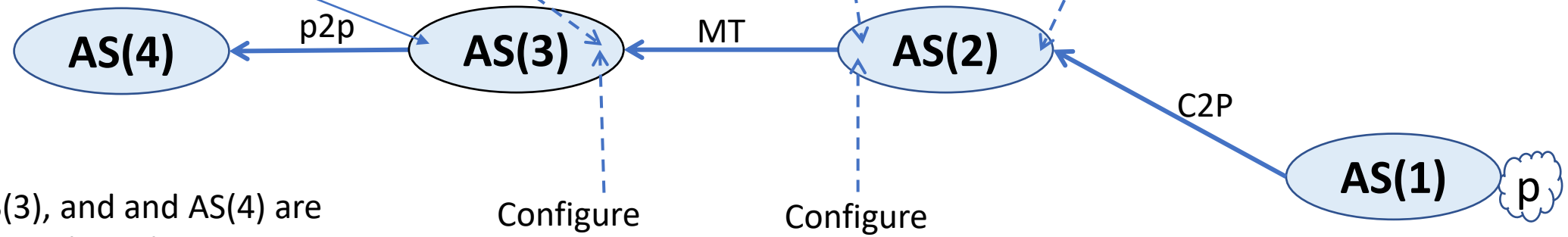
MT = mutual transit

Egress:
Do as RFC 9234 says.

Ingress OTC processing [RFC9234] MUST NOT be performed. Any existing OTC is retained.

Egress OTC processing [RFC9234] MUST NOT be performed. Any existing OTC is retained.

Ingress:
Do as RFC 9234 says.



AS(2), AS(3), and and AS(4) are doing BGP Role and OTC [RFC9234]

Configure AS(2) as MT

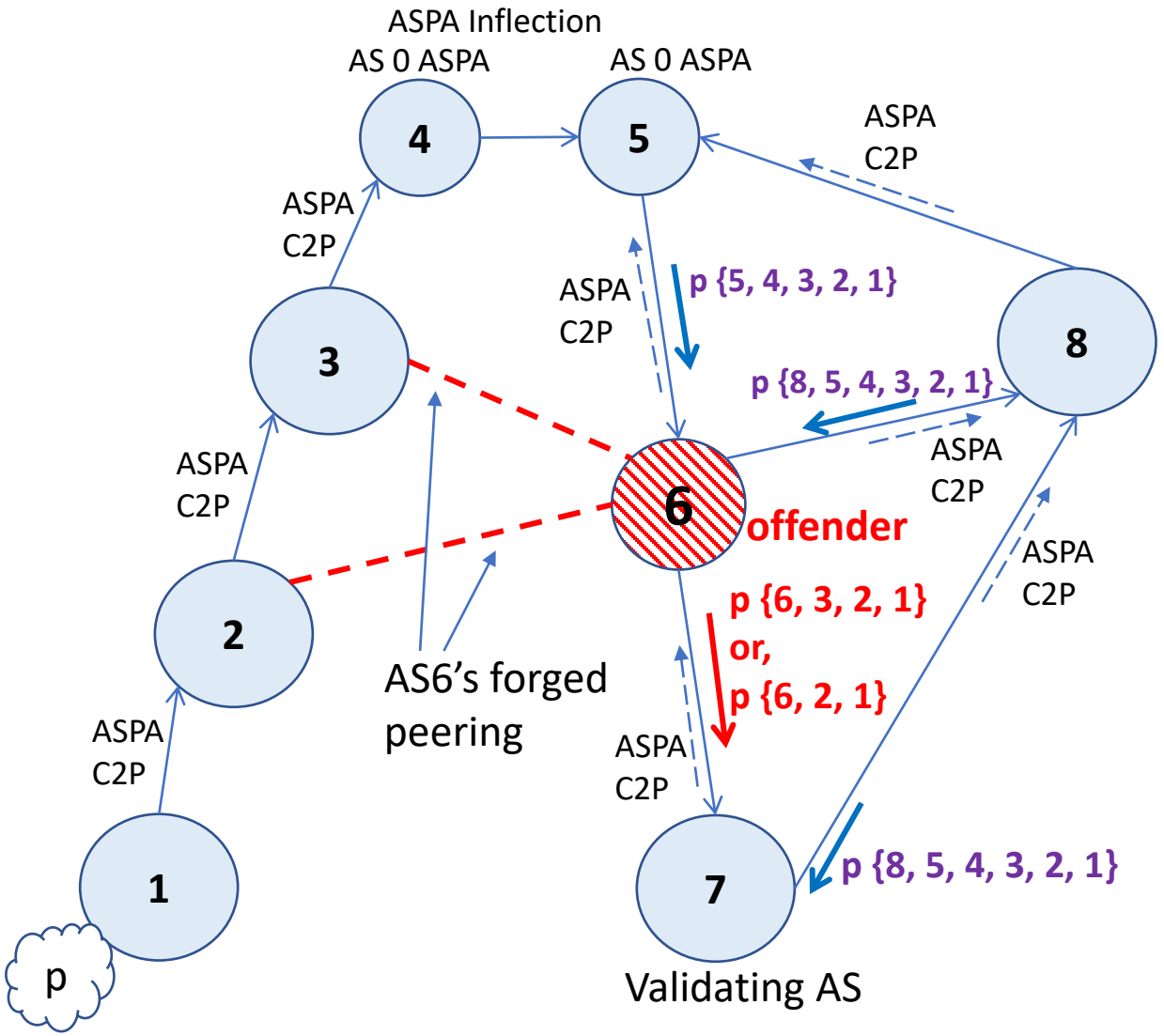
Configure AS(3) as MT

Update flow

Fig. 2

Make the Security Considerations section more solid and substantial (Ruediger)

AS_PATH maliciously shortened by a Provider



C2P = Customer to Provider

- All ASes are doing ASPA
- AS6 (provider) wants AS7 (customer) to prefer its path
- AS6 shortens the AS_PATH
- AS6 does not intend to drop data traffic from AS7 (if it does, that can be addressed only at a higher layer)
- Consider path validation at AS 7

ASPA deficiency: AS6 can shorten the path as shown w/o detection. ASPA verification cannot verify the physical existence of the AS3 to AS6 (or AS2 to AS6) hop* (the presumed inflection hop).

In ASPA's defense:

(a) In the given scenario, any physically connected data path from AS6 to p is route-leak free in BGP. Both AS paths {5, 4, 3, 2, 1} and {8, 5, 4, 3, 2, 1} are route-leak free. That benefits AS7 for its data path to destination p.

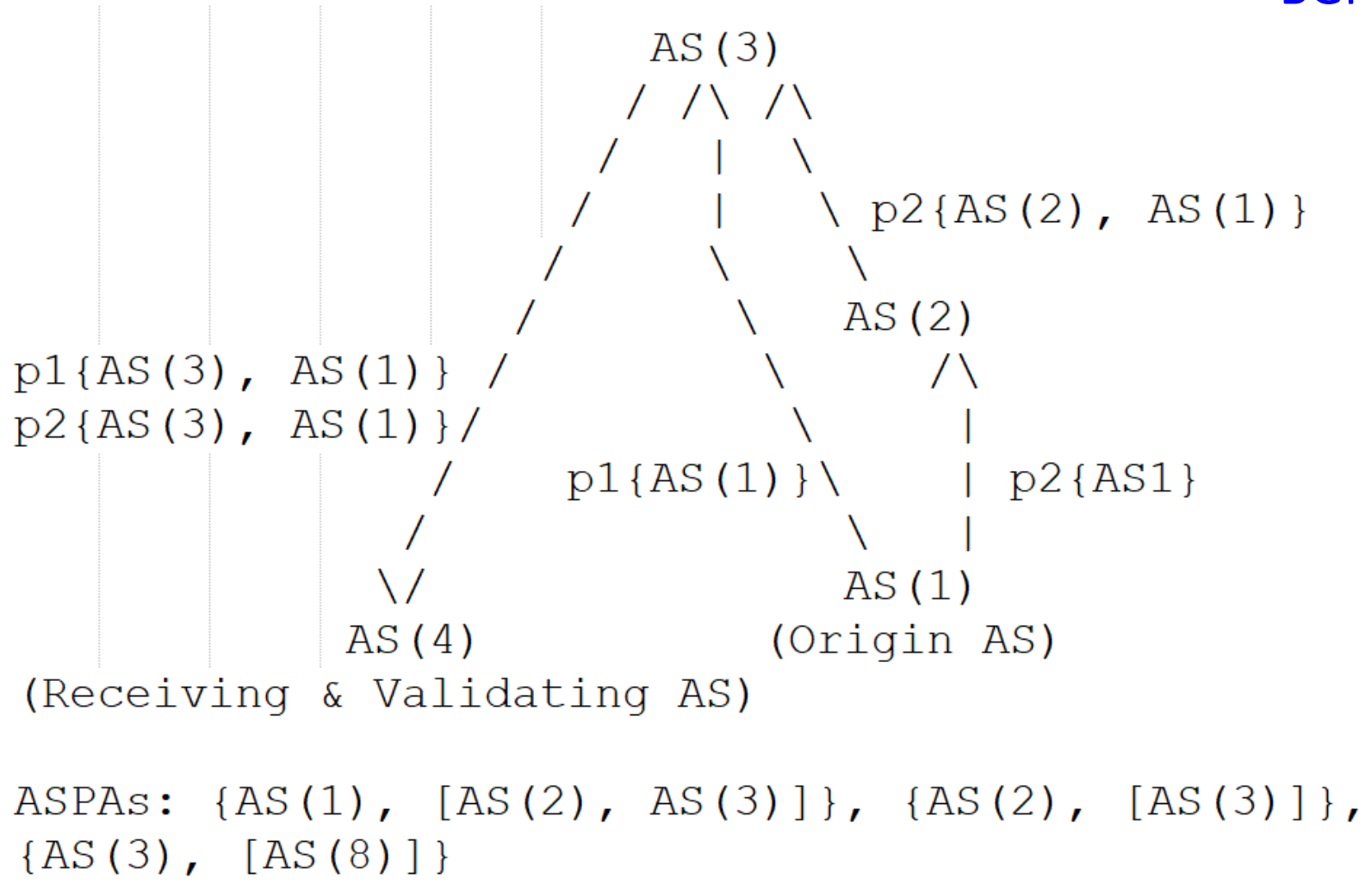
(b) It is an AS path modification attack, but not a route leak anomaly/attack (see slide 2).

* Strict verification is possible if lateral peers are also required to be registered in RPKI. But that would complicate things.

Malicious Removal/Addition of AS Prepended

- ASPA cannot protect against malicious removal/addition of AS prepends
- BGPsec can

- The path for p2 is manipulated by AS(3)
- ASPA method fails to detect
- Attack too devious for a 2nd Tier ISP?
  - BGPsec can help



Two Versions of the Draft Update under discussions (GitHub)

PR24 (https://github.com/QratorLabs/ASPA/pull/24/commits) -- Sriram (in collaboration with Claudio, Alexander), May 12
xml : https://github.com/ksriram25/ASPA/blob/patch-17/draft-ietf-sidrops-aspa-verification.xml

Current Master: -- Alexander, June 14
xml: https://github.com/QratorLabs/ASPA/blob/master/draft-ietf-sidrops-aspa-verification.xml

Interested WG members please review and comment to help with final convergence.

Questions for the WG

- WG OK with a more detailed explanations about how the algorithms work (per request from TT, MW, CJ)?
- WG OK with more detailed explanation/illustrations of the types of path manipulations ASPA cannot detect (per Ruediger's request)?
- Define Mutual Transit and Complex as separate types of sessions rather than saying Mutual Transit is a special case of Complex
- Keep some existing terminology unchanged (already whetted through WG LC in March 2023)
  - Validated ASPA Payload (VAP) -- X.509 validated
  - Provider+ (connotes Provider, RS, Mutual Transit)
- Keep "Early Adoption Benefits" section where it is currently in the main body (rather than moved to the Appendix)