Insights into the accuracy of algorithms for ASPA based route leak detection

K. Sriram and Jakob Heitz
ksriram@nist.gov   jheitz@cisco.com

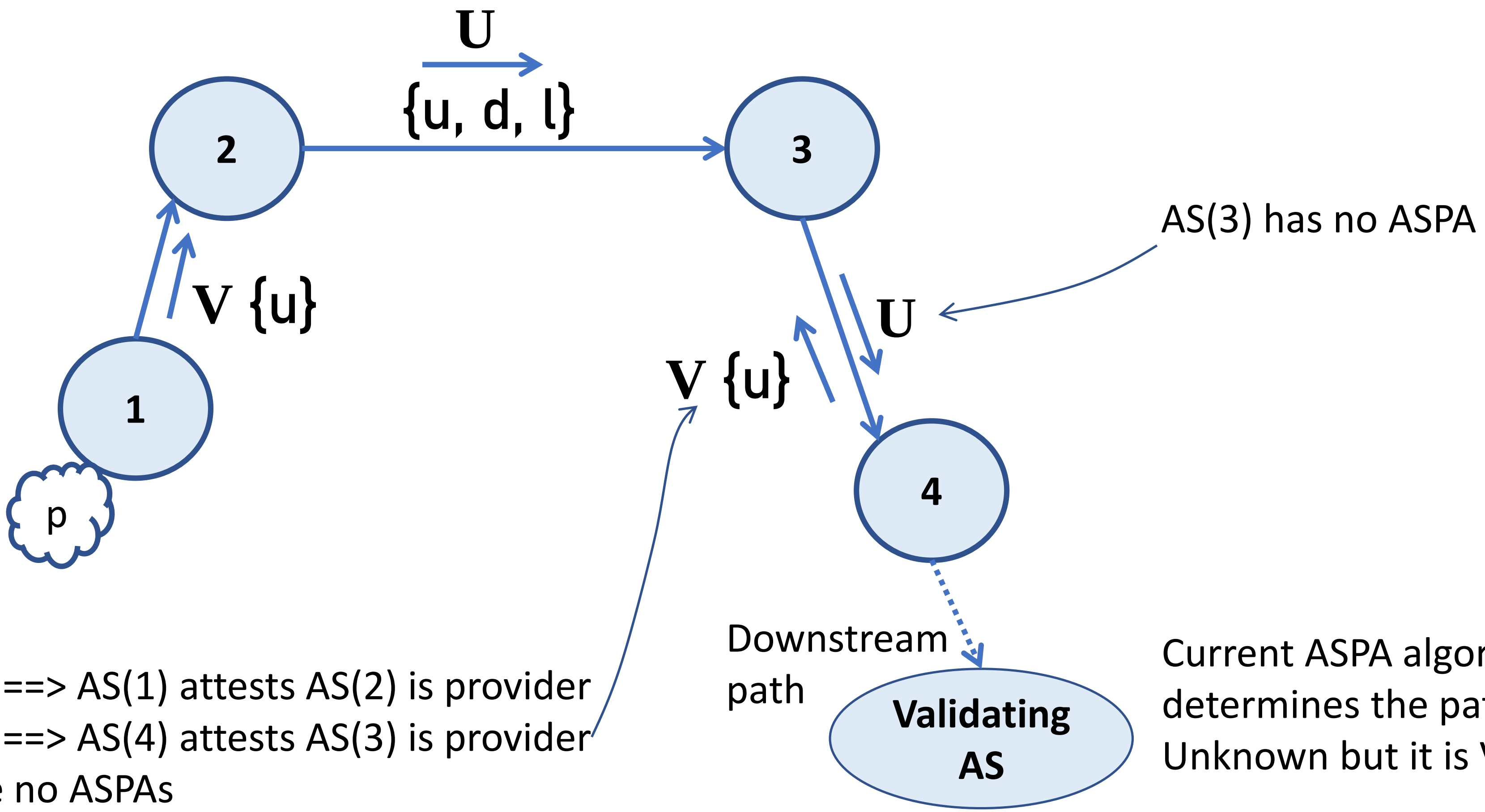
February 2021

An error in the current draft-06 algorithm

- The current algorithm classifies some Valid downstream AS paths as Unknown

Hop ASPA validity:
V: Valid
I: Invalid
U: Unknown

AS-AS peering:
u = Up
d = Down
l = Lateral



ASPA: {AS(1), AS(2)} ==> AS(1) attests AS(2) is provider
ASPA: {AS(4), AS(3)} ==> AS(4) attests AS(3) is provider
AS(2) and AS(3) have no ASPAs

Current ASPA algorithm determines the path to be Unknown but it is Valid

Design principles for ASPA based for route leak detection:

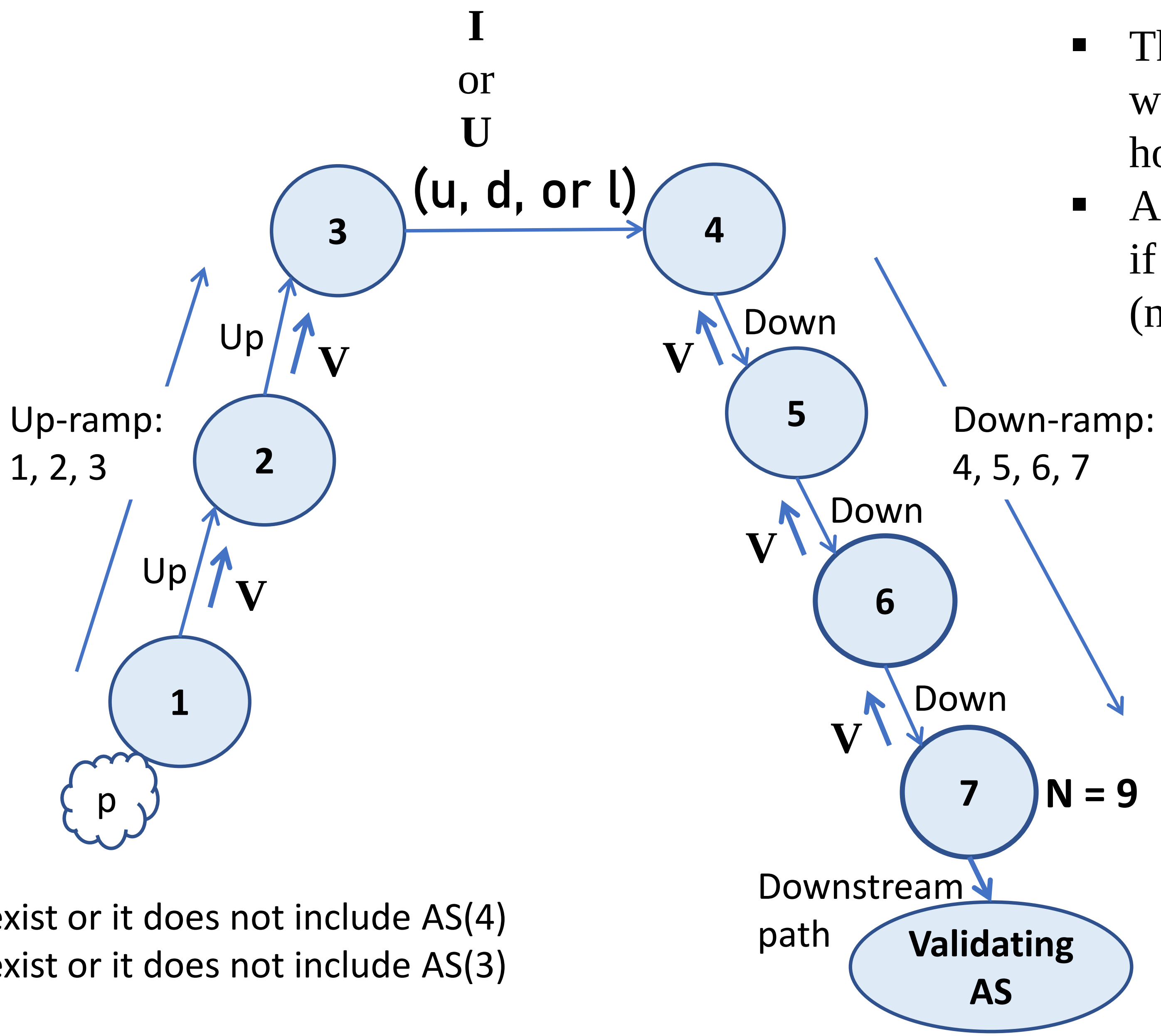
Focusing on Downstream Path only

- AS path is Valid if
  - there is a up-ramp of Valid C2P segments on the left,
  - there is a down-ramp of Valid P2C segments on the right, and
  - No other hops in the middle, or a single lateral hop in the middle which is Unknown/Invalid.
  - Up-ramp or down-ramp or both can be absent
    - One of those cases is when the AS path is simply two lateral peers.
- In effect, the above can be also stated as follows: If every transit AS has at least one neighbor that attests it a provider, then the AS path is valid.
- If the AS path segment in the middle (between the up-ramp and the down-ramp) is 2 or more hops long, then the AS path can be only Invalid or Unknown.
- The algorithm must work as best as possible if there are ASes not making attestations. To find an Invalid path in the presence of Unknown AS hops:
  - If there are opposing valley walls, i.e., an Invalid hop from left to right and a subsequent Invalid hop from right to left, then no matter what is in between, there must be at least one valley in the AS path and hence it is Invalid.
  - Otherwise, the AS path is Unknown.

Valid AS path

Hop ASPA validity:
V: Valid
I: Invalid
U: Unknown

AS-AS peering:
u = Up
d = Down
l = Lateral



ASPAs:
AS(1), AS(2)
AS(2), AS(3)
AS(5), AS(4)
AS(6), AS(5)
AS(7), AS(6)

ASPA of AS(3) does not exist or it does not include AS(4)
ASPA of AS(4) does not exist or it does not include AS(3)

- The AS path is Valid with/without the **I** or **U** hop in the middle
- AS path is trivially Valid if the AS path length is 2 (no ASPA needed)

(K, L) representation of Downstream AS path

AS(1) AS(2) …. AS(K-1) AS(K) AS(K+1) …. AS(L-1) AS(L) AS(L+1) … AS(N-1) AS(N)

g(AS(i), AS(i+1)) = **V**; i =1, …, K-1

up-ramp

g(AS(i+1), AS(i)) = **V**; i =L, …, N-1

down-ramp

g(AS(K), AS(K+1)) = **U** or **I**

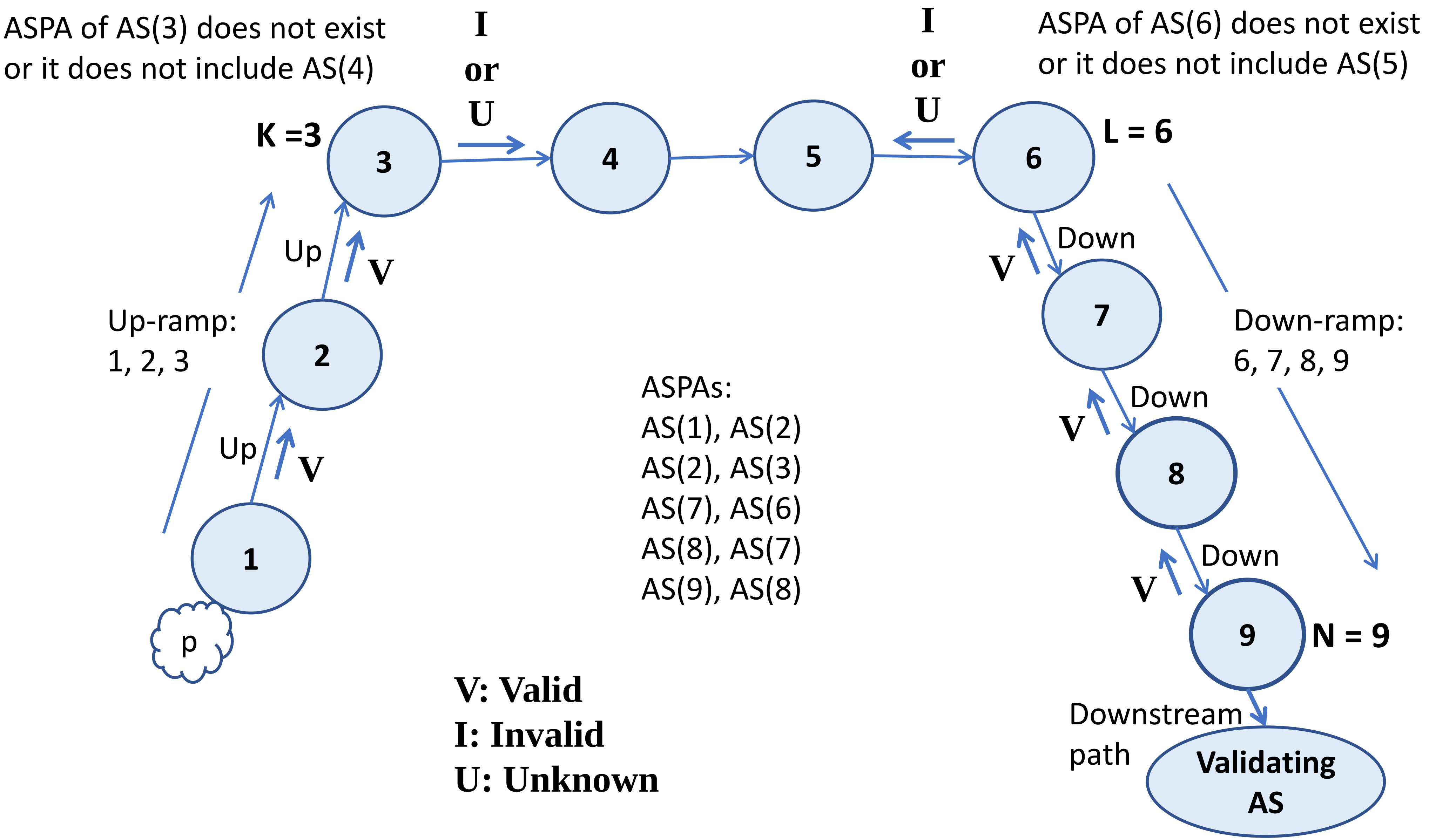
g(AS(L), AS(L-1)) = **U** or **I**

Definition:

g(AS(i), AS(j)) =
- **V** if AS(i) attests AS(j) is a provider
- **I** if AS(i) attests AS(j) is not a provider
- **U** if AS(i) does not have an ASPA

V: Valid
I: Invalid
U: Unknown

(K, L) representation of AS path



Theorems that help design the algorithm

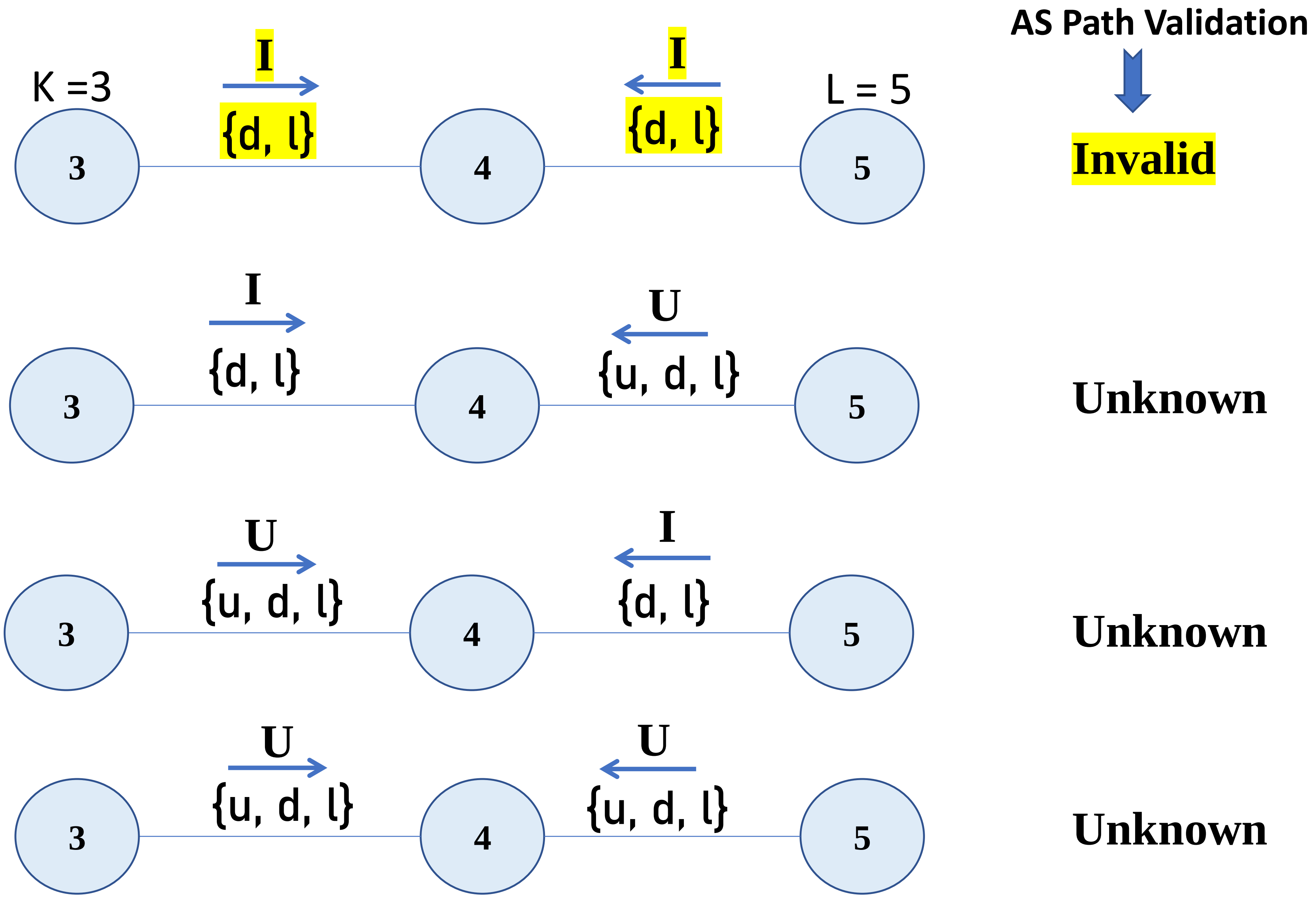
Theorem 1: The downstream AS path is "Valid" if and only if L-K $= 0$ or 1. If L-K $\geq$ 2, then the AS path can be "Unknown" or "Invalid" but never "Valid".

Theorem 2: For L-K $\geq$ 2, the validity of the whole AS path is the same as that of the partial path AS(K), AS(K+1), …., AS(L-1), AS(L). The partial path can only be either Invalid or Unknown. It is Invalid if there exist u and v (u and v in the range from K to L-1) such that u $<$ v and g(AS(u), AS(u+1)) = Invalid and g(AS(v+1), AS(v)) = Invalid. Otherwise, the partial path is Unknown.

Proofs exist; discussed in the next slide and backup slides.

For L-K ≥ 2, only Invalid or Unknown are possible

Illustration for L-K = 2



Hop ASPA validity:
V: Valid
I: Invalid
U: Unknown

AS-AS peering:
u = Up
d = Down
l = Lateral

Arrows indicate direction of hop validity

Algorithm for ASPA based downstream AS path validation

Crisp Description

Formulate the AS path using the (K, L) representation*.
The AS path is Valid if L-K $\leq$ 1 and the procedure halts.
(Note: For L-K $\geq$ 2, to determine whether the AS path is Invalid or Unknown, we only need to focus on the portion of the path from AS(K) to AS(L).)
Consider the partial path represented by AS(K), AS(K+1), …, AS(L-1), AS(L).
For L-K $\geq$ 2, if there exist u and v in the range from K to L-1 such that u $<$ v and
g(AS(u), A(u+1)) = Invalid, and
g(AS(v+1), A(v)) = Invalid,
then the AS path is Invalid and the procedure halts.
Else, the AS path is Unknown.

* Collapsing mutual transit ASes and keeping only one of them in the AS path (when they each have ASPA attesting the other as provider) is not necessary because the algorithm looks at each hop in both directions. Particularly, where it may matter -- in the middle portion, AS(K) to AS(L).

Algorithm for ASPA based downstream AS path validation

Implementation procedure

1. If there is an AS_SET present in the AS path, then set AS_SET_FLAG = 1, else AS_SET_FLAG = 0.
2. Collapse prepends in the AS_SEQUENCE in the AS path so that each AS number (ASN) in the path is unique. Call this path (after collapsing the prepends) as the AS path for this algorithm.
3. If the AS path in step 2 is empty, then go to step 11*.
4. Let the AS path be represented as AS(1), AS(2), …, AS(N-1), AS(N), where N is the AS path length and AS(N) is the most recently added AS in the AS path and neighbor to the receiving/validating AS.
5. If N =1, then the update is "Valid" and go to step 11.
6. Evaluate sequentially starting from i = 1 and determine the largest i (= i_max) such that g(AS(i), AS(i+1)) is "Valid" for each i ≤ imax. If there is no such i_max, then set i_max = 0. Let K = i_max + 1.
7. Evaluate sequentially starting from j = 1 and determine the largest j (j_max) for which g(AS(N-j+1), AS(N-j)) is "Valid" for each j ≤ jmax. If there is no such j_max, then set j_max = 0. Let L = N - j_max.
8. If L-K ≤ 1, then the Update is "Valid" and go to step 11.
9. At this step, L-K ≥ 2. Compute P(i) = g(AS(i), AS(i+1)) sequentially starting at i = K and incrementing by 1 each time. Record the lowest value of i (K ≤ i ≤ L-2) for which P(i) = "Invalid" and set u equal to that lowest value. Compute Q(j) = g(AS(j+1), AS(j)) sequentially starting at j = L-1 and decrementing by 1 each time. Record the highest value of j (K+1 ≤ j ≤ L-1) for which Q(j) = "Invalid" and set v equal to that highest value. If u < v, then the AS path is "Invalid" and the procedure halts.
10. If AS_SET_FLAG = 0, then the Update is "Unknown" and the procedure halts.
11. If AS_SET_FLAG = 1, then the Update is "Unverifiable" and the procedure halts.

* Totally empty AS_PATH (no AS_SEQUENCE, no AS_SET) would be an error in eBGP.

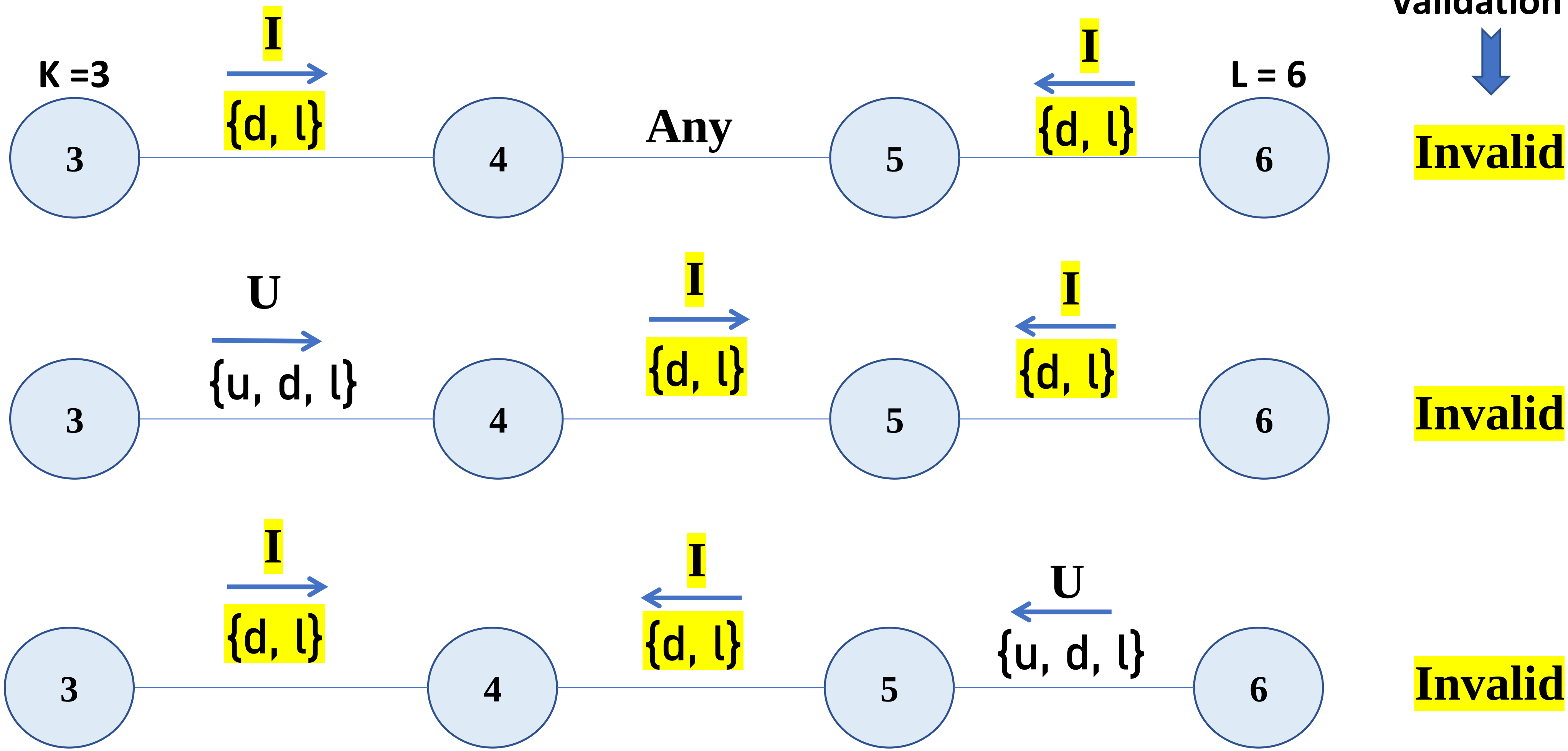
Backup slides

Proof of the Theorems

Proof: For L-K ≥ 2, only Invalid or Unknown are possible

Illustration for L-K = 3

AS Path Validation



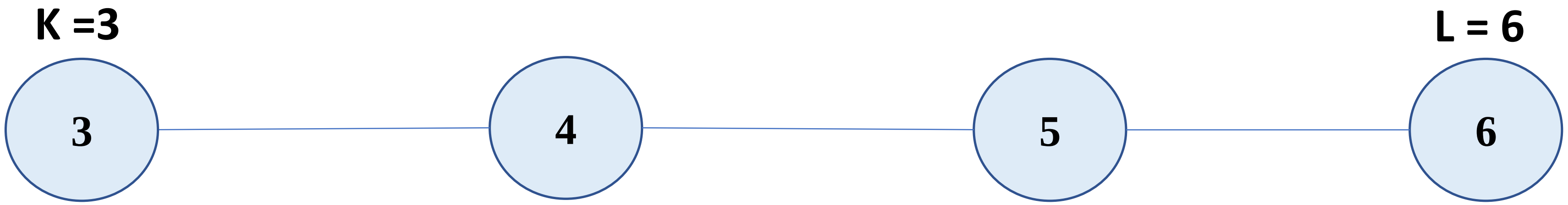
Hop ASPA validity:
V: Valid
I: Invalid
U: Unknown

AS-AS peering:
u = Up
d = Down
l = Lateral

Arrows indicate direction of hop validity

Proof: For L-K ≥ 2, only Invalid or Unknown are possible

Illustration for L-K = 3



Hop 3-4	Hop 4-5	Hop 5-6	AS path
→ I {d, l}	Any: **V, I, or U**	← I {d, l}	Invalid
→ I {d, l}	← I {d, l}	← U {u, d, l}	Invalid
→ I {d, l}	← U {u, d, l}	← U {u, d, l}	Unknown
→ I {d, l}	← V {u}	← U {u, d, l}	Unknown
→ U {u, d, l}	→ I {d, l}	← I {d, l}	Invalid
→ U {u, d, l}	→ I {d, l}	← U {u, d, l}	Unknown
→ U {u, d, l}	→ U {u, d, l}	← I {d, l}	Unknown
→ U {u, d, l}	→ U {u, d, l}	← U {u, d, l}	Unknown
→ U {u, d, l}	→ V {u}	← I {d, l}	Unknown
→ U {u, d, l}	→ V {u}	← U {u, d, l}	Unknown

Hop ASPA validity:
V: Valid
I: Invalid
U: Unknown

AS-AS peering:
u = Up
d = Down
l = Lateral

→ ← Arrows indicate direction of hop validity

Proof of Theorem 2

Theorem 2 has been shown to be correct by enumeration for L-K = 2 and L-K = 3 (see slides 8, 12, 13). Now the proof can be completed by the method of induction. It can be shown that if the assertion is true for L-K = n, then it also true for L-K = n+1, for any value of n. This will be described in a paper.