

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

K. Sriram and Jakob Heitz

[ksriram@nist.gov](mailto:ksriram@nist.gov) [jheitz@cisco.com](mailto:jheitz@cisco.com)

March 2021

# Overview – Key Takeaways

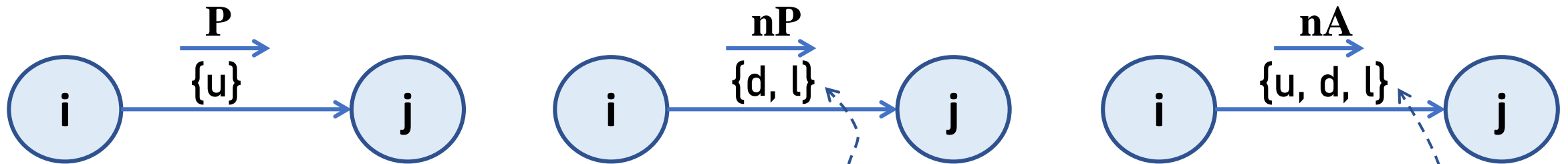
- Downstream AS path algorithm in the current draft (-06) has an oversight
  - Some Valid AS paths are misclassified as Unknown
- A correct algorithm exists with formal proof
  - Classifies all Valid, Invalid, and Unknown AS paths correctly
- We think the proposed new algorithm is correct and efficient
  - Minimizes ASPA look ups
- We recommend updating the algorithm in the draft

# ASPA Hop Check Function

Definition:

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

**P**: Provider  
**nP**: not Provider  
**nA**: no Attestation



AS-AS peering:

u = Up (provider to customer)

d = Down (customer to provider)

l = Lateral (peer to peer)

Note: It is well understood that ASPAs are AFI dependent, so AFI is not explicitly shown in function *g*.

# An error in the current draft-06 algorithm

## Example 1

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

ASPA hop check:

**P**: Provider

**nP**: not Provider

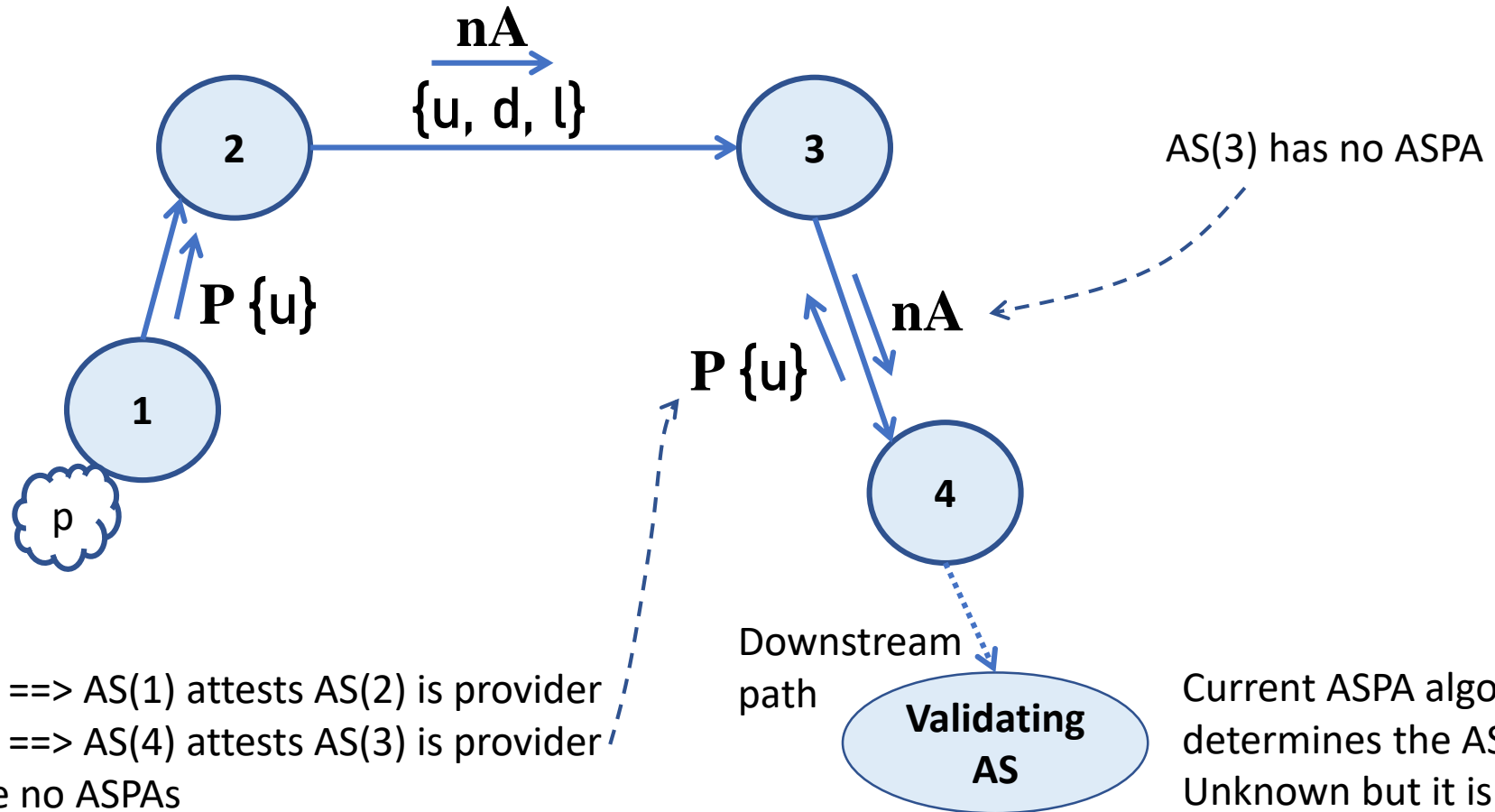
**nA**: no Attestation

AS-AS peering:

u = Up

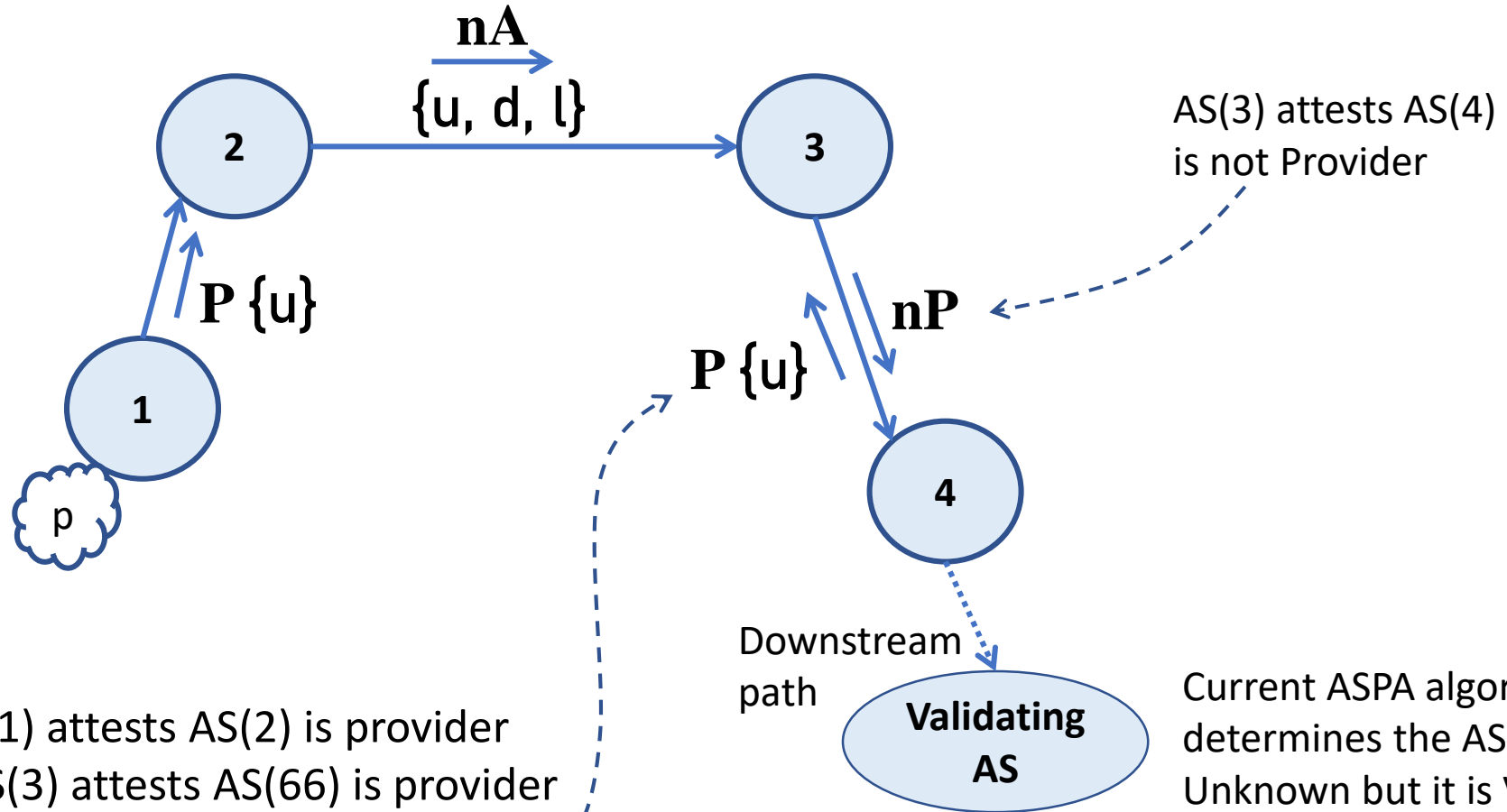
d = Down

l = Lateral



# An error in the current draft-06 algorithm

Example 2



ASPAs:

{AS(1), AS(2)} ==> AS(1) attests AS(2) is provider

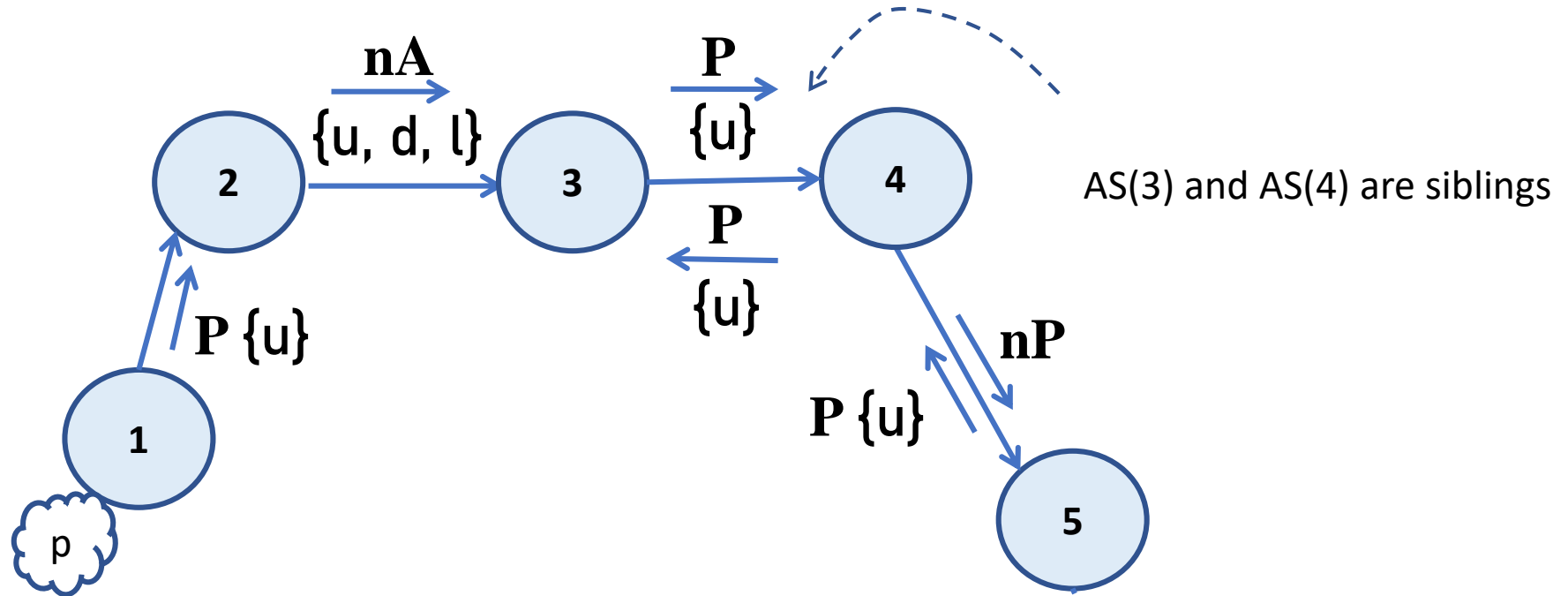
{AS(3), AS(66)} ==> AS(3) attests AS(66) is provider

{AS(4), AS(3)} ==> AS(4) attests AS(3) is provider

AS(2) and AS(3) have no ASPAs

# An error in the current draft-06 algorithm

Example 3



ASPAs:

{AS(1), AS(2)} ==> AS(1) attests AS(2) is provider

{AS(3), AS(4)}

{AS(4), AS(3)}

{AS(5), AS(4)}

AS(2) doesn't have ASPA

Downstream path



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 hops on the left,
  - there is a down-ramp of Valid P2C hops on the right, and
  - No other hops in the middle, or a single lateral hop in the middle which is **nP** or **nA**.
  - 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:
  - If there are opposing valley walls, i.e., an **nP** hop from left to right and a subsequent **nP** 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

ASPA hop check:

**P**: Provider

**nP**: not Provider

**nA**: no Attestation

AS-AS peering:

u = Up

d = Down

l = Lateral

ASPAs:

AS(1), AS(2)

AS(2), AS(3)

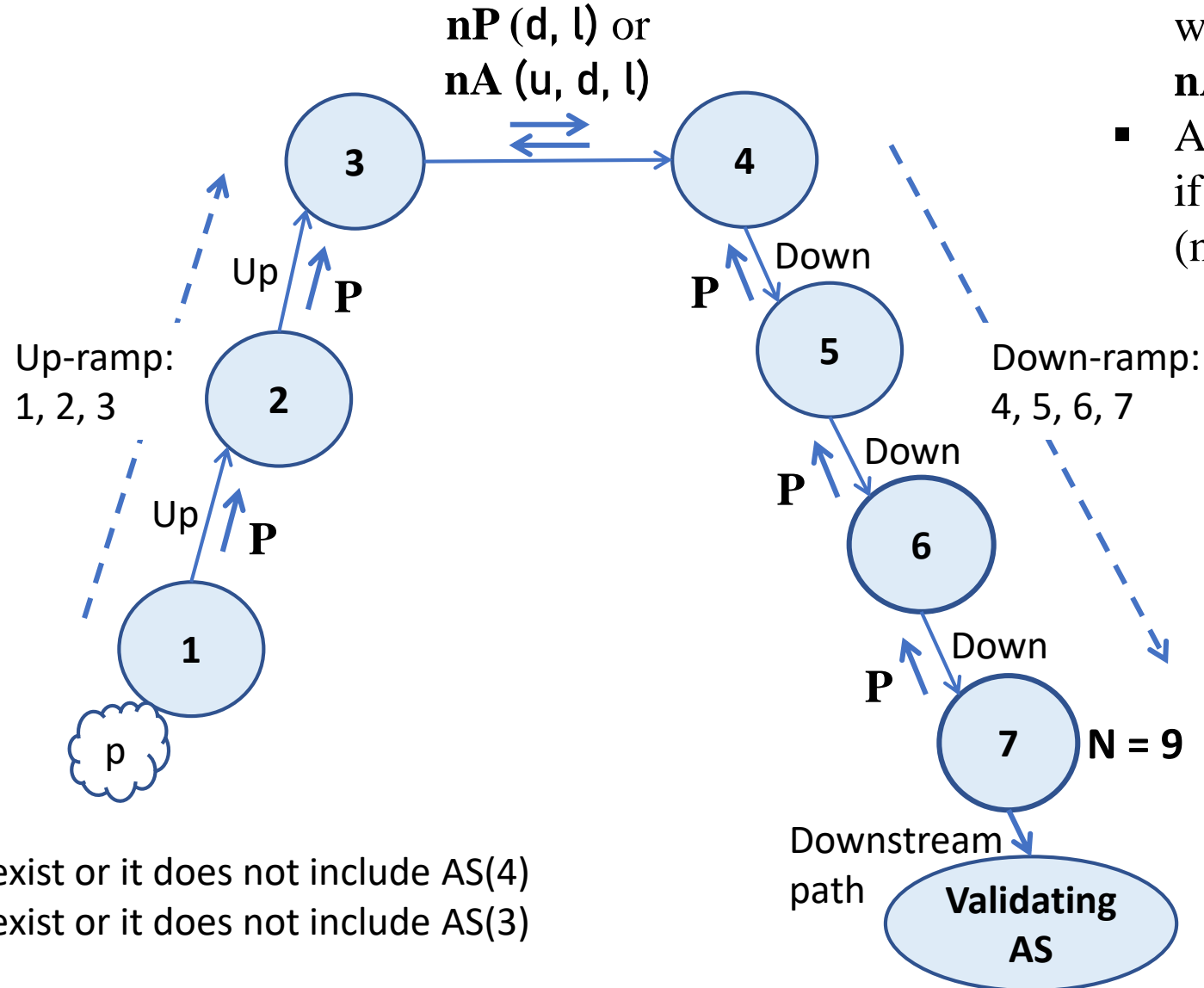
AS(5), AS(4)

AS(6), AS(5)

AS(7), AS(6)

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

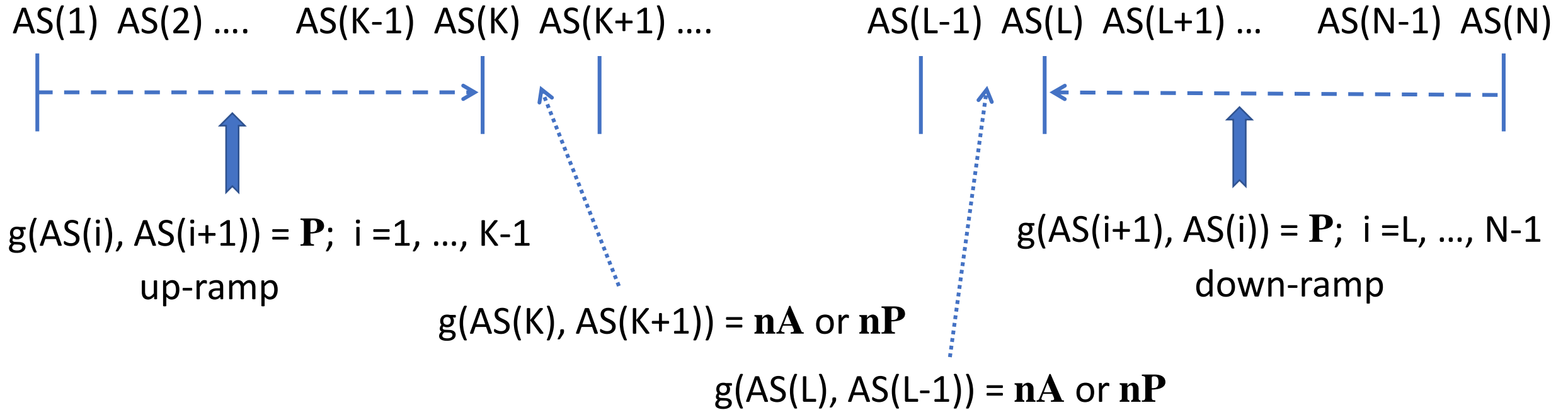
ASPA of AS(4) does not exist or it does not include AS(3)



- The AS path is Valid with/without the **nP** or **nA** 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



$g$  is the ASPA hop check function as defined on slide 3.

ASPA hop check:

**P**: Provider

**nP**: not Provider

**nA**: no Attestation

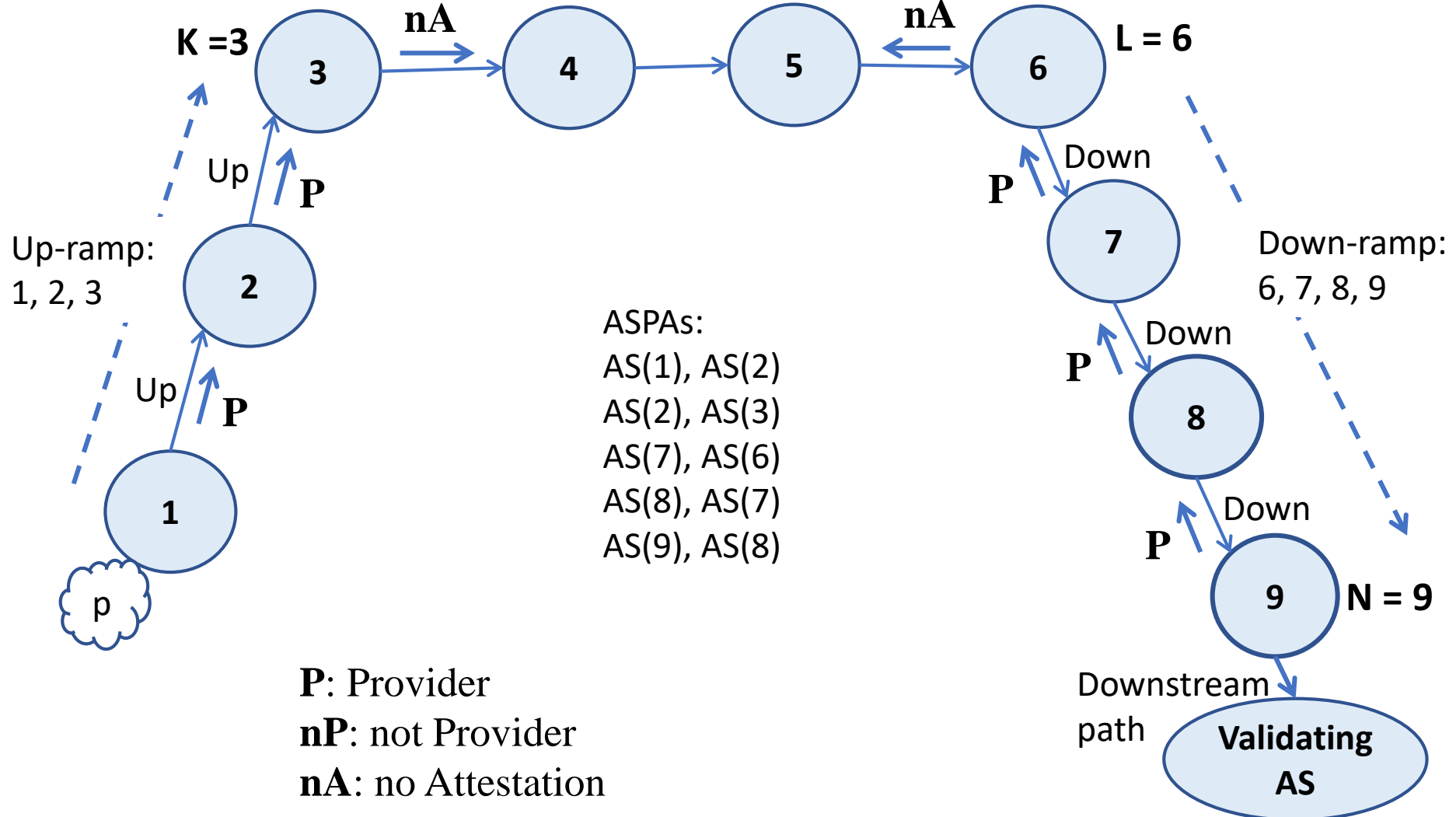
# (K, L) representation of Downstream AS path

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

**nP**  
**or**  
**nA**

**nP**  
**or**  
**nA**

ASPA of AS(6) does not exist  
or it does not include AS(5)



# When $L < K$

- $L < K$  means that the up-ramp and down-ramp overlap
  - Obvious that the AS path is Valid
- Happens when there are siblings in the middle of the path
  - Sibling: neighbors  $AS(i)$  and  $AS(j)$  each attest the other as provider
- During computation, the down-ramp determination can be halted when the top of the down-ramp touches the top of the up-ramp:
  - $AS(1)$  to  $AS(K)$  is regarded as the up-ramp
  - $AS(K+1)$  to  $AS(L)$  is regarded as the down-ramp

# Theorems that help design the algorithm

**Theorem 1:** The downstream AS path is “Valid” if and only if  $L-K \leq 1$ . 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  $AS(K), AS(K+1), \dots, AS(L-1), AS(L)$ . 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  $g(AS(u), AS(u+1)) = \mathbf{nP}$  and  $g(AS(v+1), AS(v)) = \mathbf{nP}$ . Otherwise, the partial path is Unknown.

Function  $g$  is defined on slide 3.

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

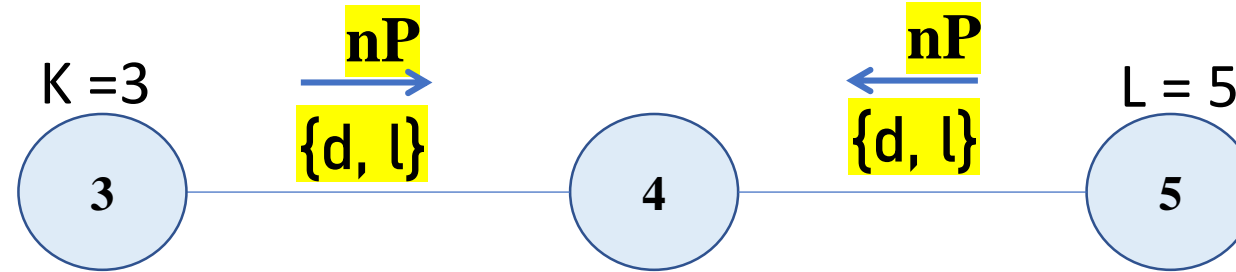
# For $L-K \geq 2$ , only Invalid or Unknown are possible

Illustration for  $L-K = 2$

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

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

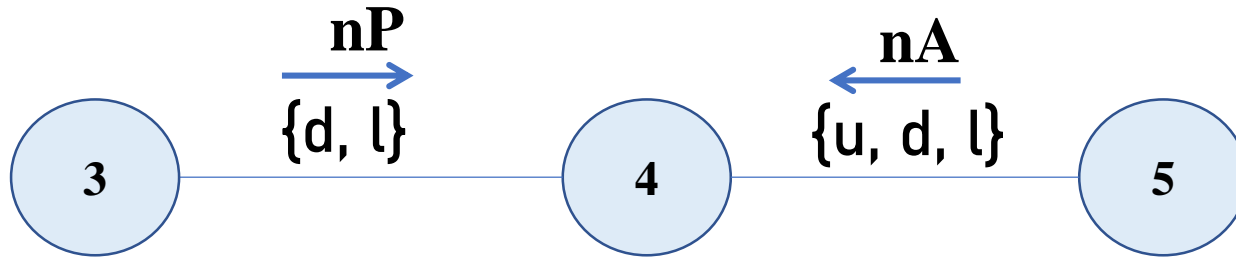
→ Arrows indicate direction of ASPA hop check



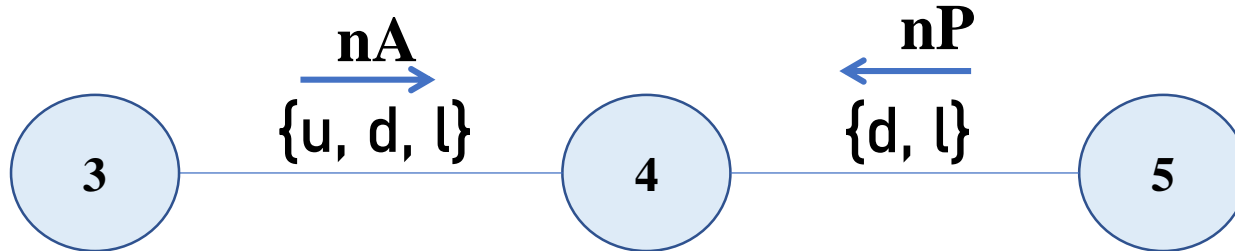
AS Path Validation



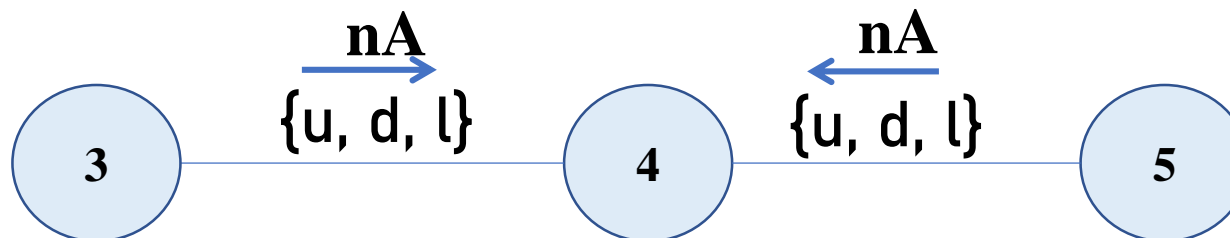
**Invalid**



**Unknown**



**Unknown**



**Unknown**

# Algorithm for ASPA based downstream AS path validation

## Crisp Description

Formulate the AS path using the  $(K, L)$  representation\*.

If the AS path length  $N \leq 2$ , then the path is Valid and the procedure halts.

If  $L-K \leq 1$ , then the AS path is Valid 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  $g(\text{AS}(u), \text{A}(u+1)) = \mathbf{nP}$ , and

$g(\text{AS}(v+1), \text{A}(v)) = \mathbf{nP}$ ,

then the AS path is Invalid and the procedure halts.

Else, the AS path is Unknown.

\* Collapsing mutual transit ASes (siblings) and keeping only one of them in the AS path (when they each have ASPA attesting the other as provider) is not necessary.

# 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  $N \leq 2$ , then the update is "Valid" and go to step 11.
6. At this step,  $N \geq 3$ . Evaluate sequentially starting from  $i = 1$  ( $1 \leq i \leq N-2$ ) and determine the largest  $i$  (=  $i_{\max}$ ) such that  $g(\text{AS}(i), \text{AS}(i+1)) = \mathbf{P}$  for each  $i \leq i_{\max}$ . If there is no such  $i_{\max}$ , then set  $i_{\max} = 0$ . Let  $K = i_{\max} + 1$ . If  $K = N-1$ , then the AS path is "Valid" and go to step 11.
7. Evaluate sequentially starting from  $j = 1$  ( $1 \leq j \leq N-K-1$ ) and determine the largest  $j$  (=  $j_{\max}$ ) for which  $g(\text{AS}(N-j+1), \text{AS}(N-j)) = \mathbf{P}$  for each  $j \leq j_{\max}$ . If there is no such  $j_{\max}$ , then set  $j_{\max} = 0$ . Let  $L = N - j_{\max}$ .
8. If  $L-K \leq 1$ , then the AS path is "Valid" and go to step 11.
9. At this step,  $L-K \geq 2$ . Record the lowest value of  $i$  ( $K \leq i \leq L-2$ ) for which  $g(\text{AS}(i), \text{AS}(i+1)) = \mathbf{nP}$  and set  $u$  equal to that lowest value. If no such value for  $u$  exists, then go to step 10. Else, find a value of  $j$  ( $u+1 \leq j \leq L-1$ ) for which  $g(\text{AS}(j+1), \text{AS}(j)) = \mathbf{nP}$ . If such  $j$  is found, then the AS path is "Invalid" and the procedure halts.
10. If AS\_SET\_FLAG = 0, then the Update is "Unknown". The procedure halts.
11. If AS\_SET\_FLAG = 1, then the Update is "Unverifiable". The procedure halts.

\* Totally empty AS\_PATH (no AS\_SEQUENCE, no AS\_SET) would be an error in eBGP.

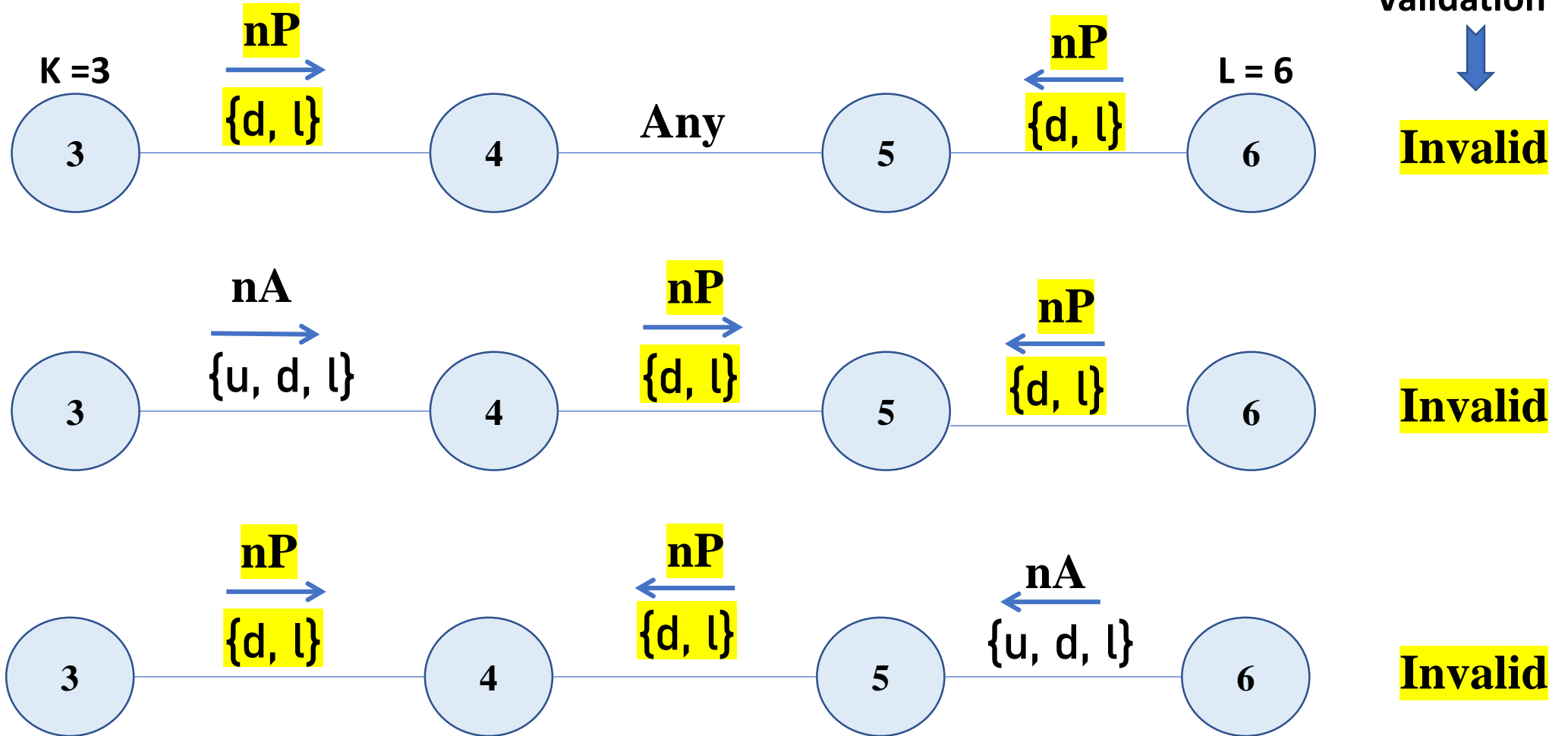
# Backup slides

## Proof of the Theorems



# Proof: For $L-K \geq 2$ , only Invalid or Unknown are possible

Illustration for  $L-K = 3$



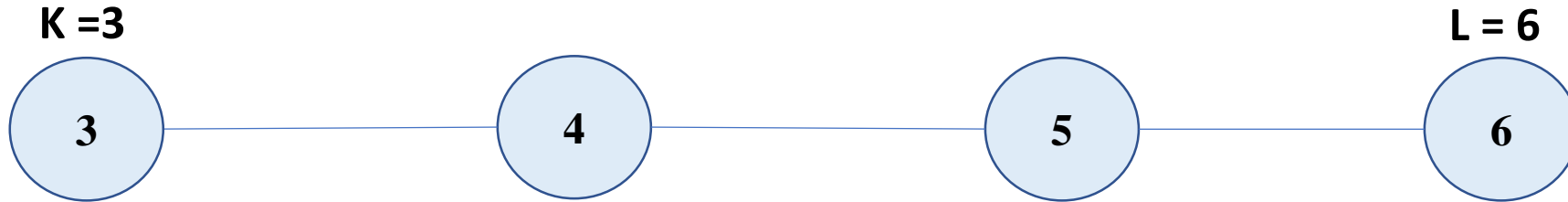
ASPA hop check:  
**P:** Provider  
**nP:** not Provider  
**nA:** no Attestation

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

→  
←  
Arrows indicate direction of ASPA hop check

# Proof: For $L-K \geq 2$ , only Invalid or Unknown are possible

Illustration for  $L-K = 3$



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

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

→ Arrows indicate  
 ← direction of  
 ASPA hop check

Hop 3-4	Hop 4-5	Hop 5-6	AS path
→ nP {d, l}	Any: P, nP, or nA	← nP {d, l}	Invalid
→ nP {d, l}	← nP {d, l}	← nA {u, d, l}	Invalid
→ nP {d, l}	← nA {u, d, l}	← nA {u, d, l}	Unknown
→ nP {d, l}	← P {u}	← nA {u, d, l}	Unknown
→ nA {u, d, l}	→ nP {d, l}	← nP {d, l}	Invalid
→ nA {u, d, l}	→ nP {d, l}	← nA {u, d, l}	Unknown
→ nA {u, d, l}	→ nA {u, d, l}	← nP {d, l}	Unknown
→ nA {u, d, l}	→ nA {u, d, l}	← nA {u, d, l}	Unknown
→ nA {u, d, l}	→ P {u}	← nP {d, l}	Unknown
→ nA {u, d, l}	→ P {u}	← nA {u, d, l}	Unknown

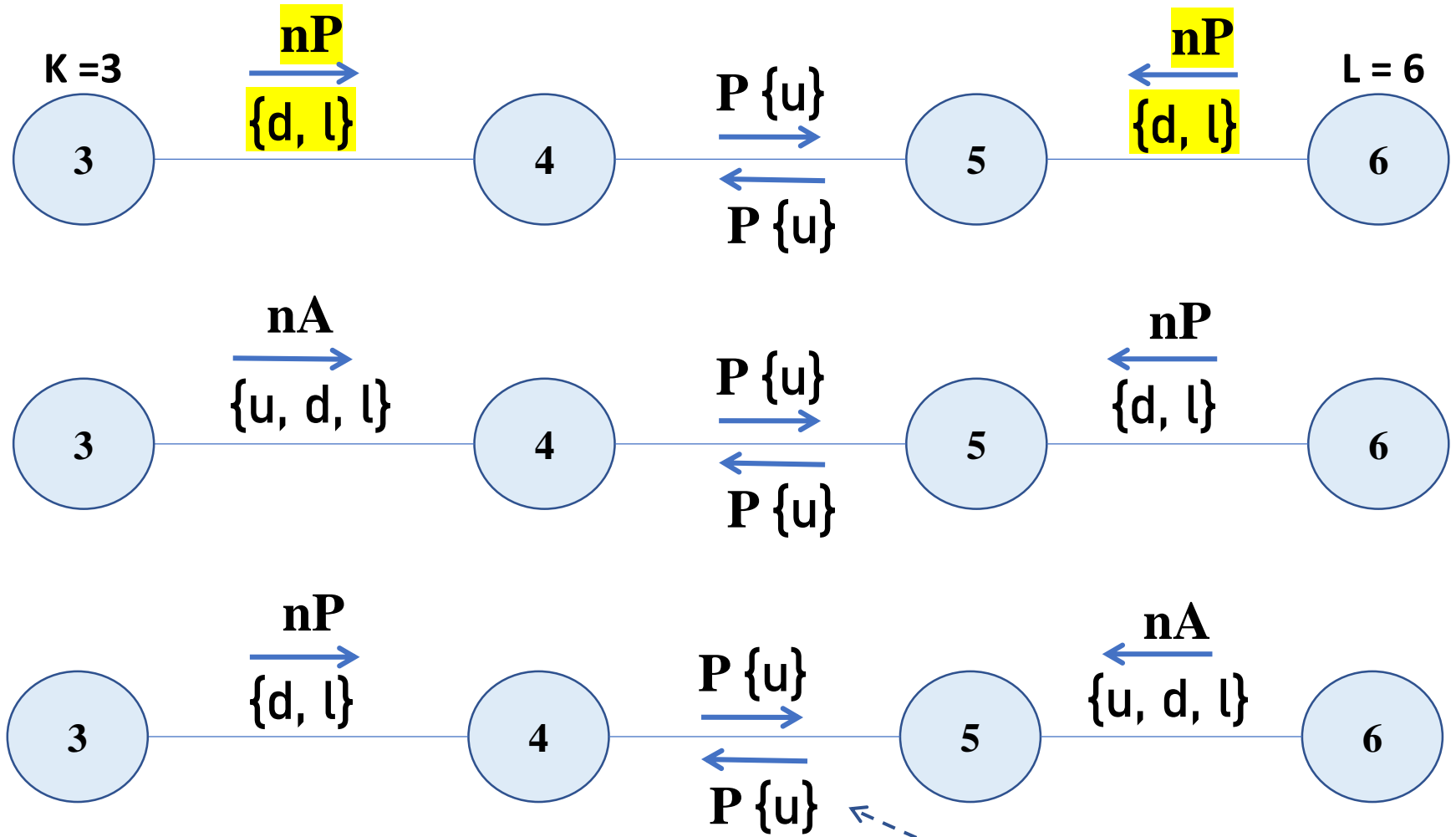
Proof: For  $L-K \geq 2$ , only Invalid or Unknown are possible  
**Theorems stay true in the presence of sibling hops**

Illustration for  $L-K = 3$

AS Path Validation



**Invalid**



**Unknown**

**Unknown**

Arrows indicate direction of ASPA hop check

AS(4) and AS(5) are siblings

# Proof of Theorem 2

Theorem 2 has been shown to be correct by enumeration for  $L-K = 2$  and  $L-K = 3$  (see slides 13, 17, 18, 19). 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.