nsdi '25
22nd USENIX Symposium on Networked Systems Design and Implementation

From Address Blocks to Authorized Prefixes: Redesigning RPKI ROV with a Hierarchical Hashing Scheme for Fast and Memory-Efficient Validation

Zedong Ni#, Yinbo Xu#, Hui Zou, Yanbiao Li*, Guang Cheng, Gaogang Xie*

中国科学院
计算机网络信息中心
Computer Network Information Center,
Chinese Academy of Sciences

SOUTHEAST UNIVERSITY 東南大學 1902 南京 心於至善

中国科学院大学
University of Chinese Academy of Sciences

紫金山实验室
Purple Mountain Laboratories

BGP is important yet vulnerable

- Border Gateway Protocol (BGP) is one of the key building blocks of the global Internet.
- However, BGP lacks built-in security protection and thus is vulnerable to prefix hijacks.

Cloudflare DNS Resolver Hit by BGP Hijack

The free Cloudflare DNS resolver service 1.1.1.1 was hit by a pair of simultaneous BGP attacks, showing that BGP is vulnerable even to accidental attacks.

by Paul Shread — July 6, 2024 Reading Time: 5 mins read

KLAYswap

KLAYSWAP|KLAYSWAP-BGP-HIJACK

Catalin Cimpanu

February 14th, 2022

News Cybercrime Technology

KlaySwap crypto users lose funds after BGP hijack

Hackers have stolen roughly $1.9 million from South Korean cryptocurrency platform KLAYswap after they pulled off a rare and clever BGP hijack against the server infrastructure of one of the platform's providers.

The BGP hijack—which is the equivalent of hackers hijacking internet routes to bring users on malicious sites instead of legitimate ones—hit KakaoTalk, an instant messaging platform popular in South Korea.

Testing mistake triggered Telstra route 'hijacks'

By Juha Saarinen
Oct 5 2020
7:33AM

5 Comments

RELATED ARTICLES

Mandiant names APT43 group as North Korean operation

TIO seeks digital platform role as scams proliferate

Microsoft introduces AI-powered cyber security assistant

Gov kicks off mobile audit process

Routing mishaps difficult to prevent.

An erroneous bulk upload of static routes to a Telstra production network edge router was the cause of last Wednesday's internet-wide service disruption that saw data traffic take a long detour via Australia, causing performance degradation for other providers in the process.

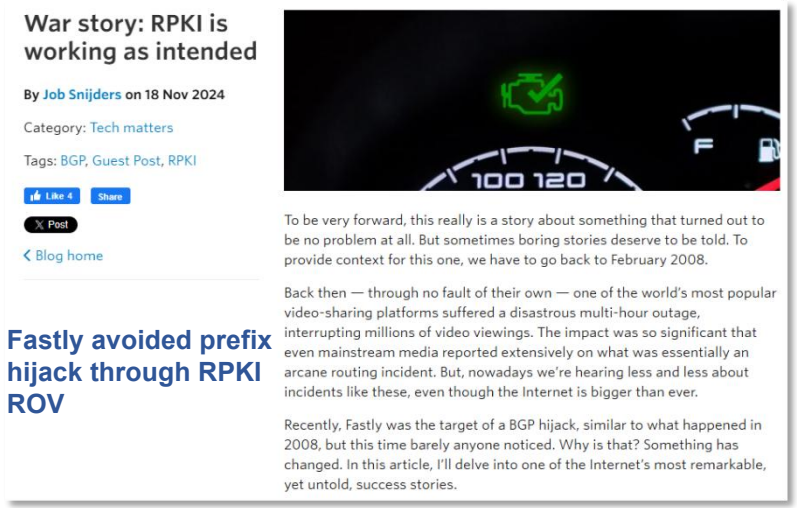
Telstra senior network engineer Mark Duffell apologised for the error, which meant that 500 internet protocol version 4 (IPv4) prefixes, or subnetworks, were advertised as belonging to Telstra.

The technical error occurred as part of post-verification testing to address a software bug in the Telstra Internet Direct provisioning tools.

After the incorrect configuration was deployed to a single edge router the hundreds of IPv4 prefixes were announced to the global internet through the border gateway protocol (BGP) that supplies route information for network providers.

RPKI ROV: Strong Protection, Limited Reach

- **Route Origin Authorization (ROA)**
  - Cryptographically binds an AS with the prefix(es) it is authorized to advertise in BGP.
- **Route Origin Validation (ROV)**
  - Validates BGP messages by veifying the oigin of route prefixes using ROAs.
  - Three kinds of results: Valid, Invalid, and NotFound.


War story: RPKI is working as intended

By Job Snijders on 18 Nov 2024

Category: Tech matters

Tags: BGP, Guest Post, RPKI

Like 4 Share

Post

Blog home

Fastly avoided prefix hijack through RPKI ROV

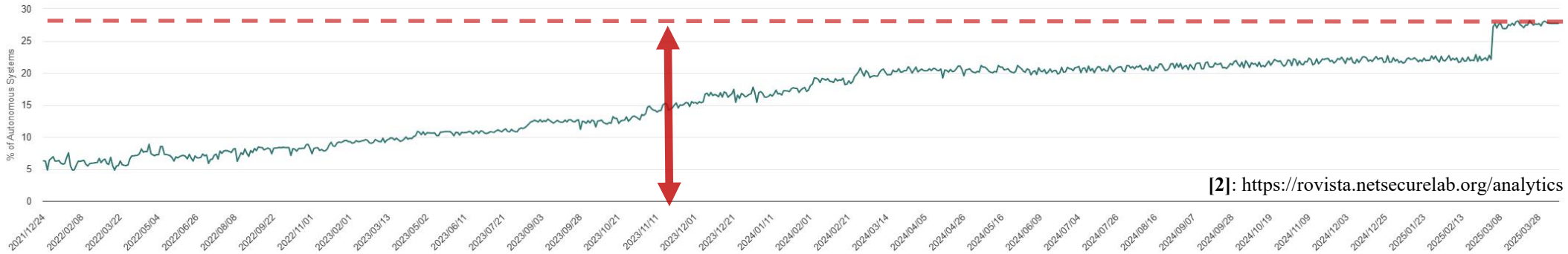
To be very forward, this really is a story about something that turned out to be no problem at all. But sometimes boring stories deserve to be told. To provide context for this one, we have to go back to February 2008.

Back then — through no fault of their own — one of the world's most popular video-sharing platforms suffered a disastrous multi-hour outage, interrupting millions of video viewings. The impact was so significant that even mainstream media reported extensively on what was essentially an arcane routing incident. But, nowadays we're hearing less and less about incidents like these, even though the Internet is bigger than ever.

Recently, Fastly was the target of a BGP hijack, similar to what happened in 2008, but this time barely anyone noticed. Why is that? Something has changed. In this article, I'll delve into one of the Internet's most remarkable, yet untold, success stories.

[1]: https://blog.apnic.net/2024/11/18/war-story-rpki-is-working-as-intended/

Only 27% of ASes have deployed ROV, and the deployment rate is **growing slowly.**

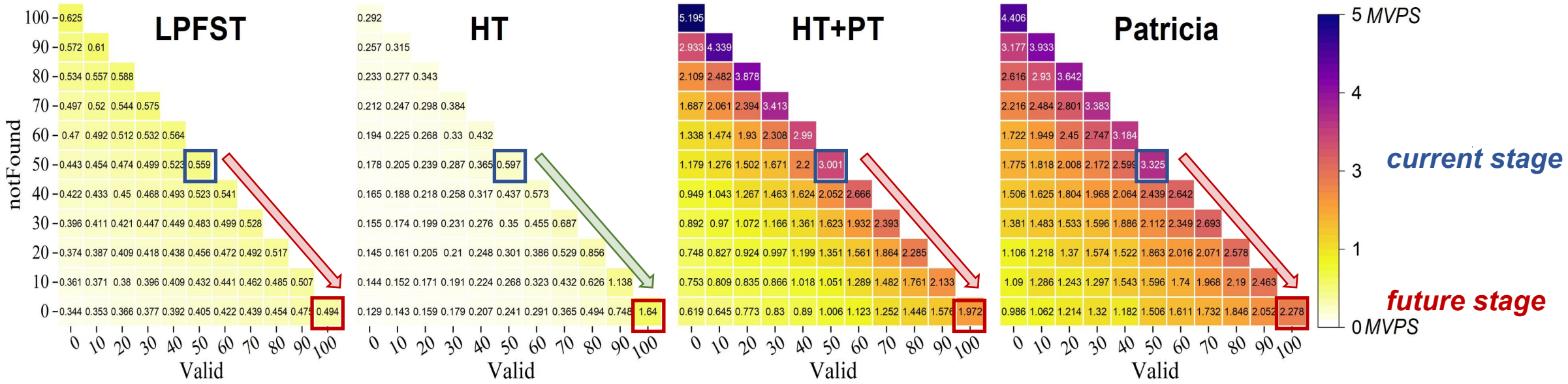


[2]: https://rovista.netsecurelab.org/analytics

Concerns about ROV efficiency

- **Operational efficiency concerns** has been identified as one of the main barrier to ROV adoption. [NDSS 24]

Our Obseveration: the more the valid routes, the lower the validation speed.



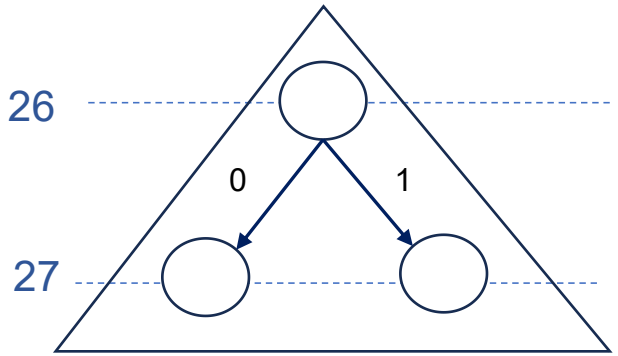
With the promotion of RPKI, the proportion of valid routes will increase, the ROV efficiency issue will become more severe.

ROV with Address Block

- ROV validates BGP messages by veifying the oigin of route prefixes using ROAs.

Address Block

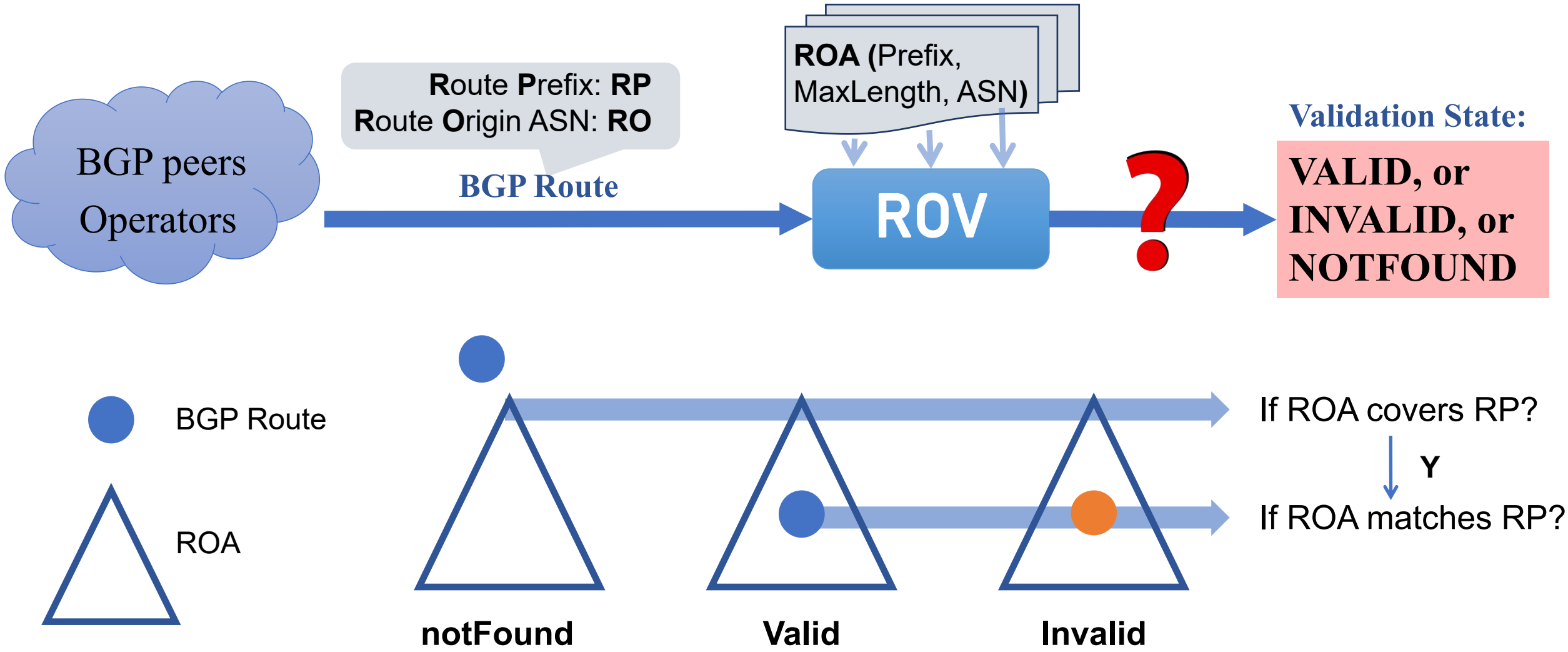
e.g. <192.0.2.64/26, 27>



L26	192.0.2.64/26
L27	192.0.2.64/27 192.0.2.96/27

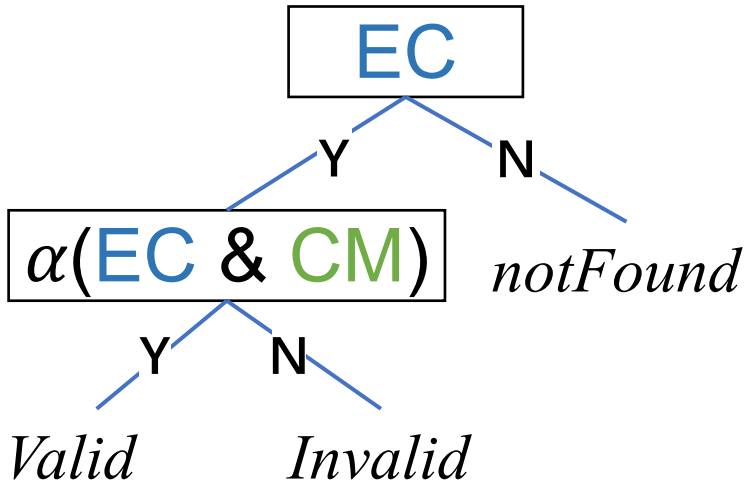
Authorized Prefixes

ROV with Address Block



Motivation

Address Block (AB) **Model**

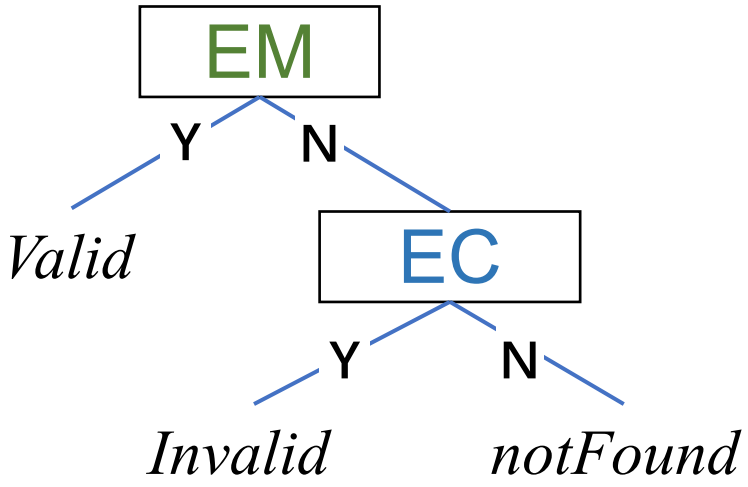


[1]: https://www.rfc-editor.org/info/rfc6811

Unfriendly to **valid** routes!

The key reason why the ROV efficiency issue will become more severe.

Authorized Prefix (AP) **Model**



equivalence of validation results

acceleration of validation speed

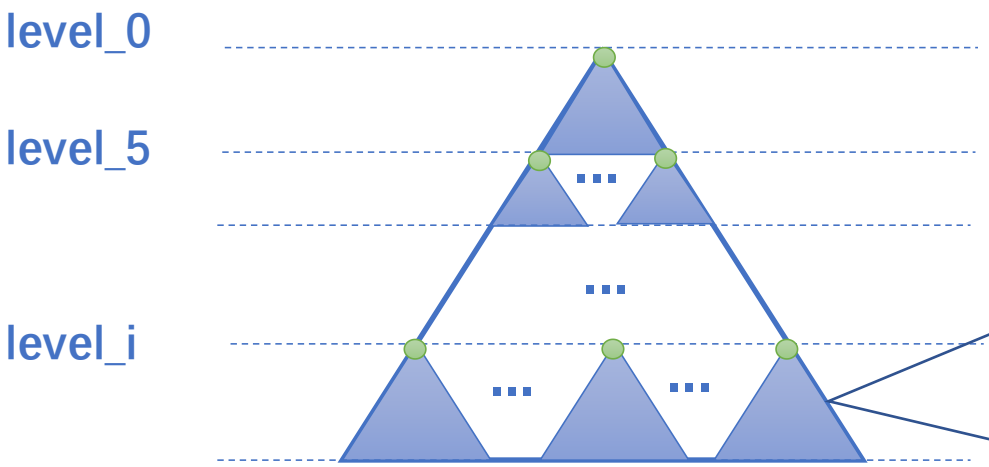
Redesigning ROV with the AP model

1. how to maintain ROAs at the granularity of Authorized Prefixes ?

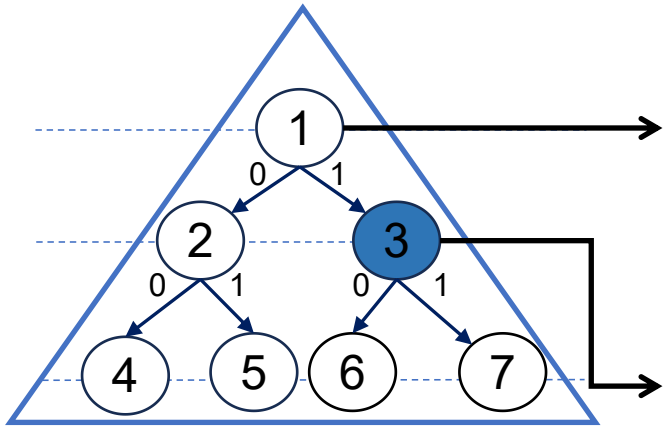
We adopt the bitmap-based encoding scheme [INFOCOM 22].

Bitmap-based encoding for authorized prefix(1)

Split an IP Prefix tree into sub-trees with a fixed-depth of 5



Each prefix can be located by **Subtree Identifier** and **Index**.



Subtree Identifier:

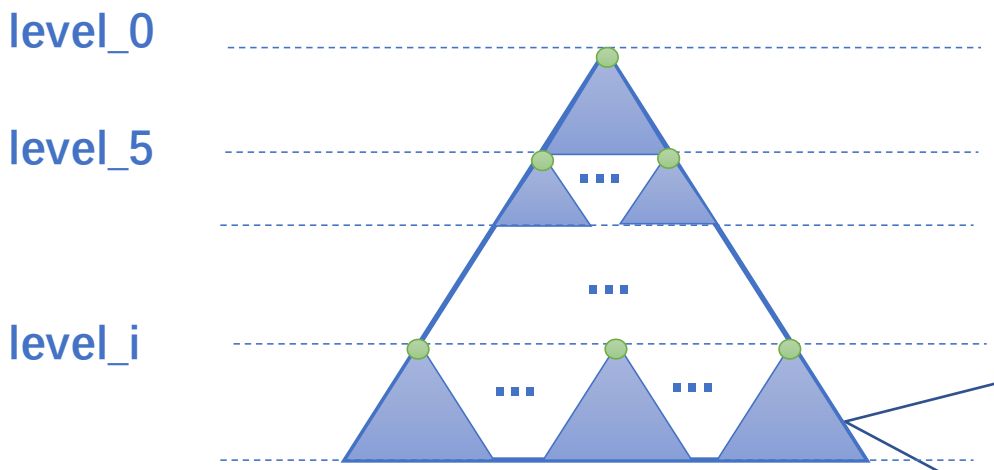
the root prefix of the sub-tree.

Index:

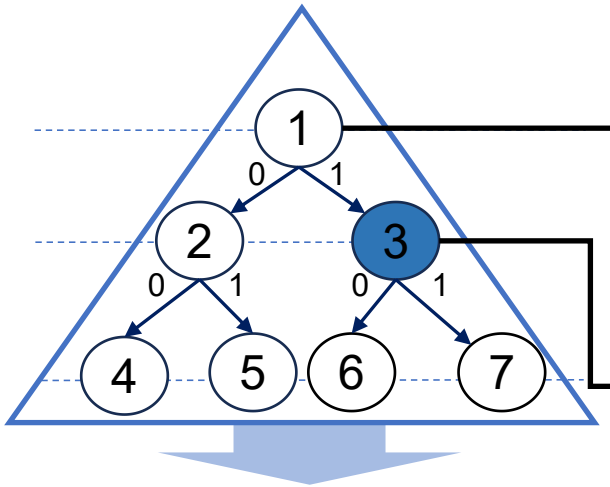
the Position of the prefix.

Bitmap-based encoding for authorized prefix(2)

Split an IP Prefix tree into sub-trees with a fixed-depth of 5



Each sub-tree can be represented by **Subtree Identifier** and **Encoded Subtree**.

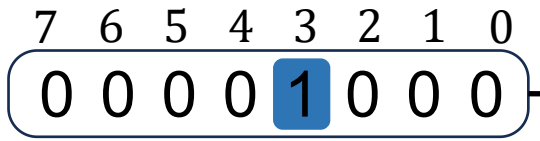


Subtree Identifier:

the root prefix of the sub-tree.

Index:

the Position of the prefix.



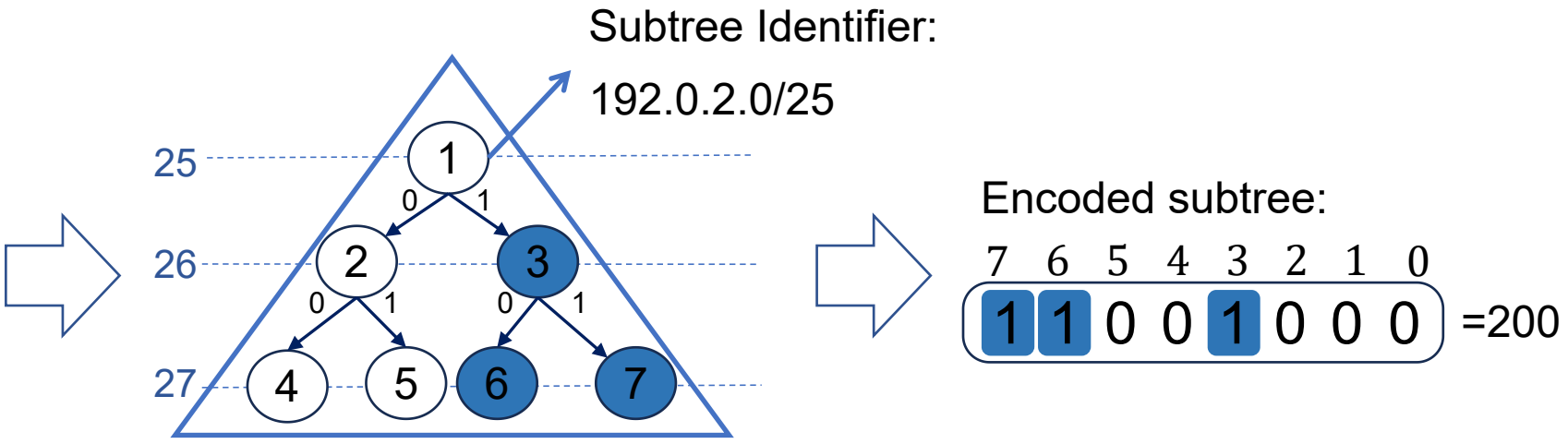
Encoded Subtree:

encodes nodes that are authorized.

Bitmap-based encoding for authorized prefix(3)

Given authorized prefixes → Locate the belonging subtree → Set the bit in the bitmap

AS **Y** originates:
192.0.2.64/26
192.0.2.64/27
192.0.2.96/27



Subtree Identifier	Encoded Subtree	ASN
192.0.2.0/25	200	AS Y

Redesigning ROV with the AP model

1. how to maintain ROAs at the granlarity of Authorized Prefixes ?

We adopt the bitmap-based encoding scheme [INFOCOM 22].

2. how to validate BGP routes with bitmap-encoded ROAs?

We propose a hierachical hashing scheme with two hash tables: SOT and STT.

ROV Match Operation with SOT

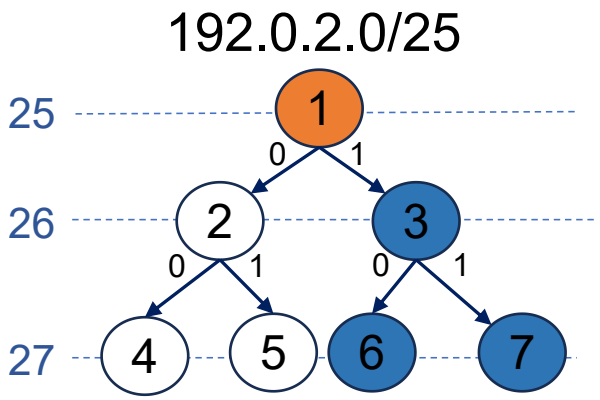
Subtree-Origin Table(SOT) records all authorized prefixes and their corresponding ASNs.

KEY: Subtree Identifier and the ASN

VALUE: Encoded Subtree

AS **X** originates:
192.0.2.0/25

AS **Y** originates:
192.0.2.64/26
192.0.2.64/27
192.0.2.96/27



Subtree-Origin Table

KEY(Subtree Identifier, ASN)	VALUE(Encoded Subtree)
<192.0.2.0/25, X>	0 0 0 0 0 0 1 0
<192.0.2.0/25, Y>	1 1 0 0 1 0 0 0
..	..

ROV Match Operation with SOT

With **SOT,** match operation can be conducted by a simple hash probe.

1. Calculate the **Subtree Identifier(ID)** and **Index** of Route Prefix.
2. Get the **Encoded Subtree(SOT[ID,ASN])**, if the **index-th** bit is set, the result is valid.

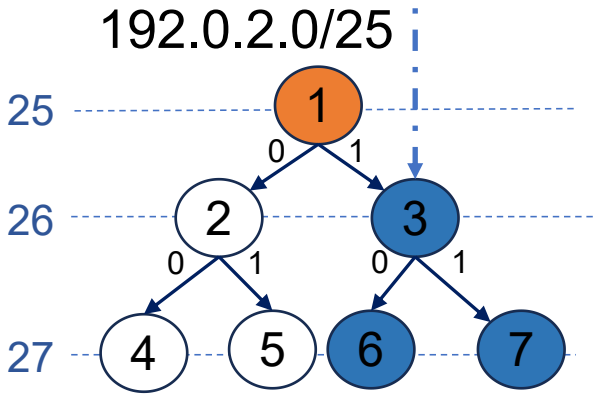
BGP Route:
<192.0.2.64/26, AS Y>

Subtree-ID=192.0.2.0/25
Index=3

Query Key = <192.0.2.0/25, Y>

AS **X** originates:
192.0.2.0/25

AS **Y** originates:
192.0.2.64/26
192.0.2.64/27
192.0.2.96/27



Subtree-Origin Table

KEY(Subtree Identifier, ASN)	VALUE(Encoded Subtree)
<192.0.2.0/25, X>	0 0 0 0 0 0 1 0
<192.0.2.0/25, Y>	1 1 0 0 1 0 0 0
..	..

Valid

ROV Cover Operation with STT

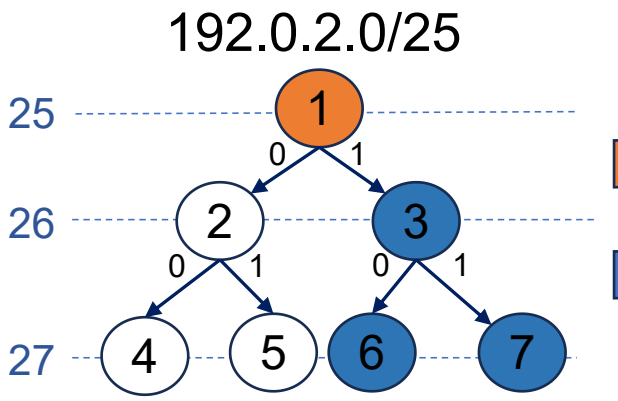
SubTree Table(STT) records all authorized prefixes(without their corresponding ASNs).

KEY: Subtree Identifier

VALUE: the merged Encoded Subtree for all ASNs

AS **X** originates:
192.0.2.0/25

AS **Y** originates:
192.0.2.64/26
192.0.2.64/27
192.0.2.96/27



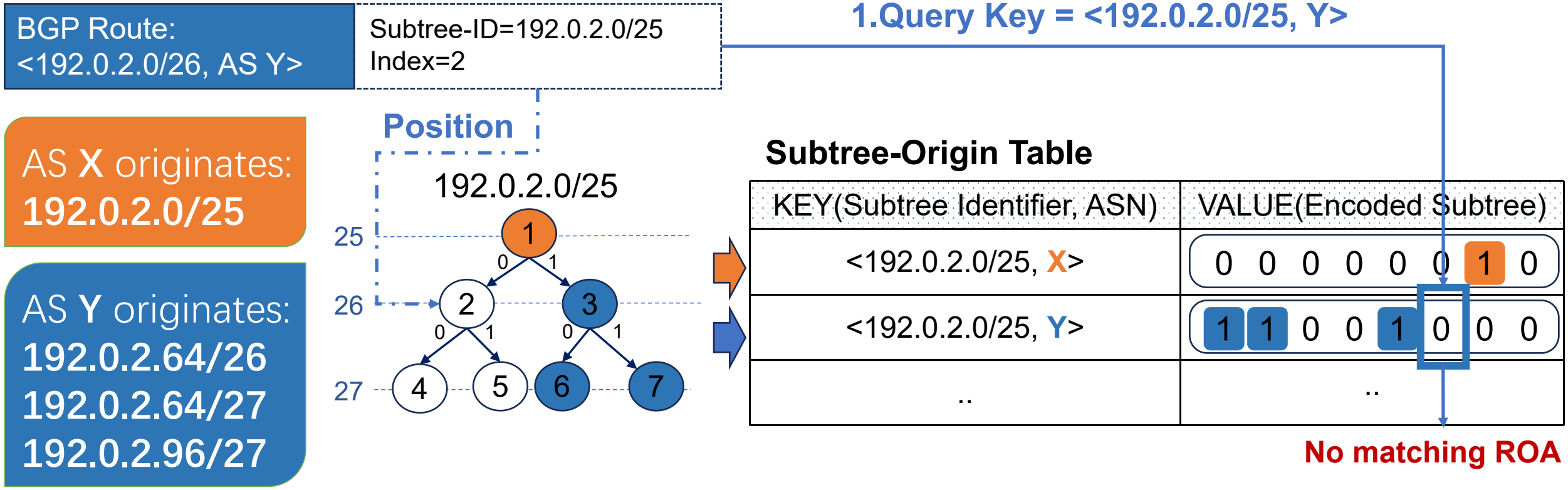
SubTree Table

KEY(Subtree Identifier)	VALUE(Encoded Subtree)
<192.0.2.0/25>	1 1 0 0 1 0 1 0
..	..

ROV Cover Operation with STT

With **STT,** cover operation can be conducted by several hash probes.

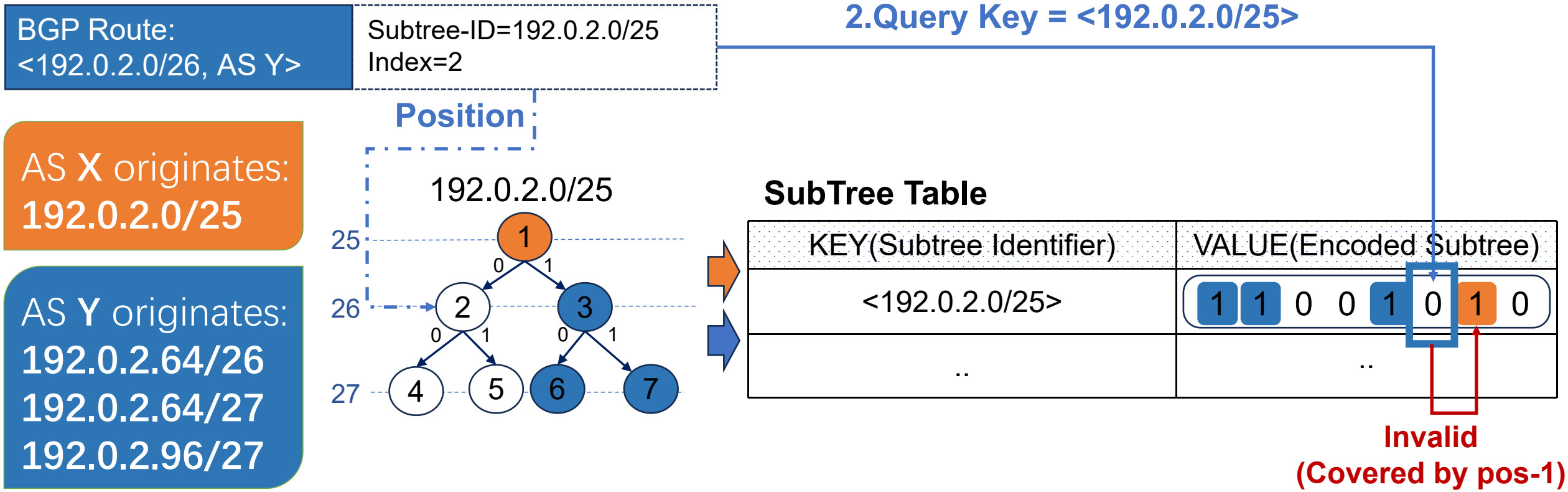
1. If query in SOT fails, get the merged encoded subtree(**STT[ID]**).
2. If there exists **a bit covering RP is set** in the merged encoded subtree, the result is Invalid.



ROV Cover Operation with STT

With **STT,** cover operation can be conducted by several hash probes.

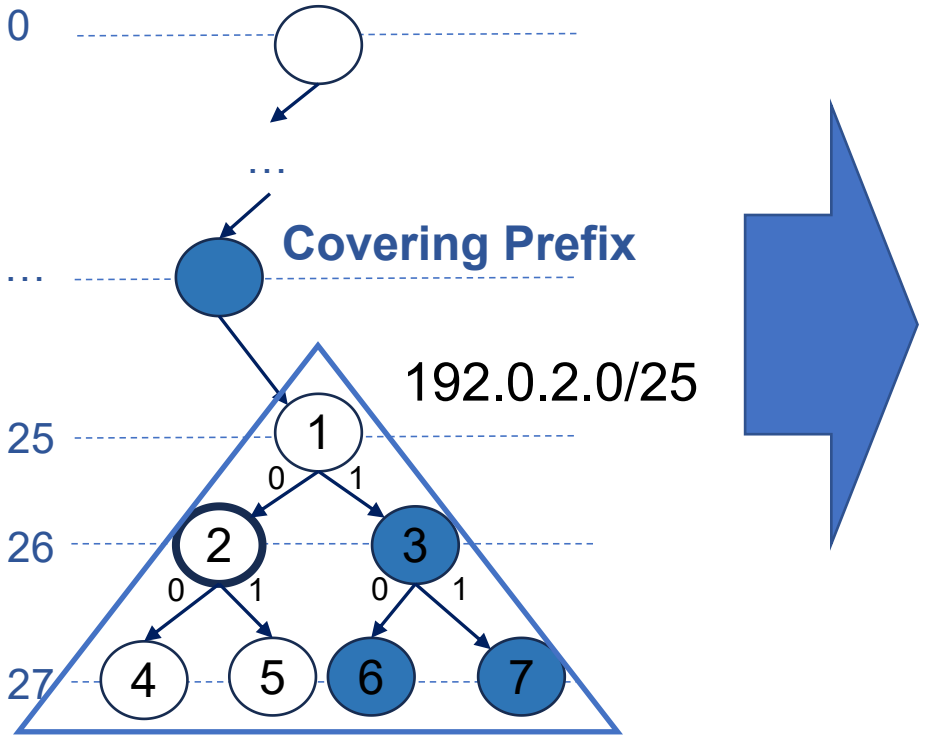
1. If query in SOT fails, get the merged encoded subtree(**STT[ID]**).
2. If there exists **a bit covering RP is set** in the merged encoded subtree, the result is Invalid.



ROV Cover Operation with STT

With **SubTree Table,** cover operation can be conducted by several hash probes.

1. If query in SOT fails, get the merged encoded subtree(**STT[ID]**).
2. If there exists **a bit covering RP is set** in the merged encoded subtree, the result is Invalid.
3. If no bit covering RP is set in the encoded subtree, find the covering prefix by **linear search**.



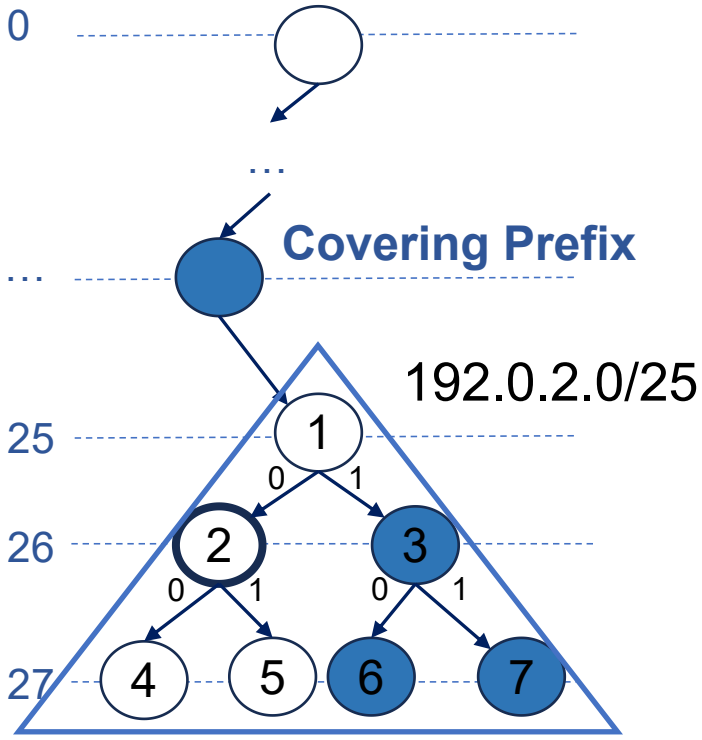
SubTree Table

KEY(Subtree Identifier)	VALUE(Encoded Subtree)
<192.0.2.0/25>	1 1 0 0 1 0 1 0
<192.0.0.0/**20**>	..
<192.0.0.0/**15**>	…
linear search…	…

ROV Cover Operation with STT -- Optimization

With **SubTree Table,** cover operation can be conducted by several hash probes.

1. If query in SOT fails, get the merged encoded subtree(**STT[ID]**).
2. If there exists **a bit covering RP is set** in the merged encoded subtree, the result is Invalid.
3. If no bit covering RP is set in the encoded subtree, find the covering prefix by **linear search**.



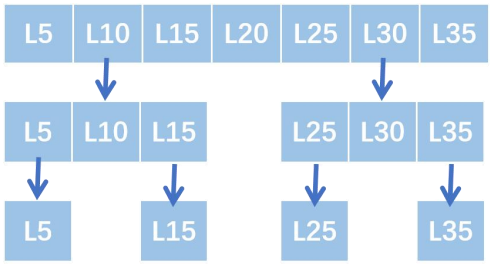
IPv4

We **record the coverage status for key prefixes**. Most prefixes can find the coverage status with time complexity **O(1)**.

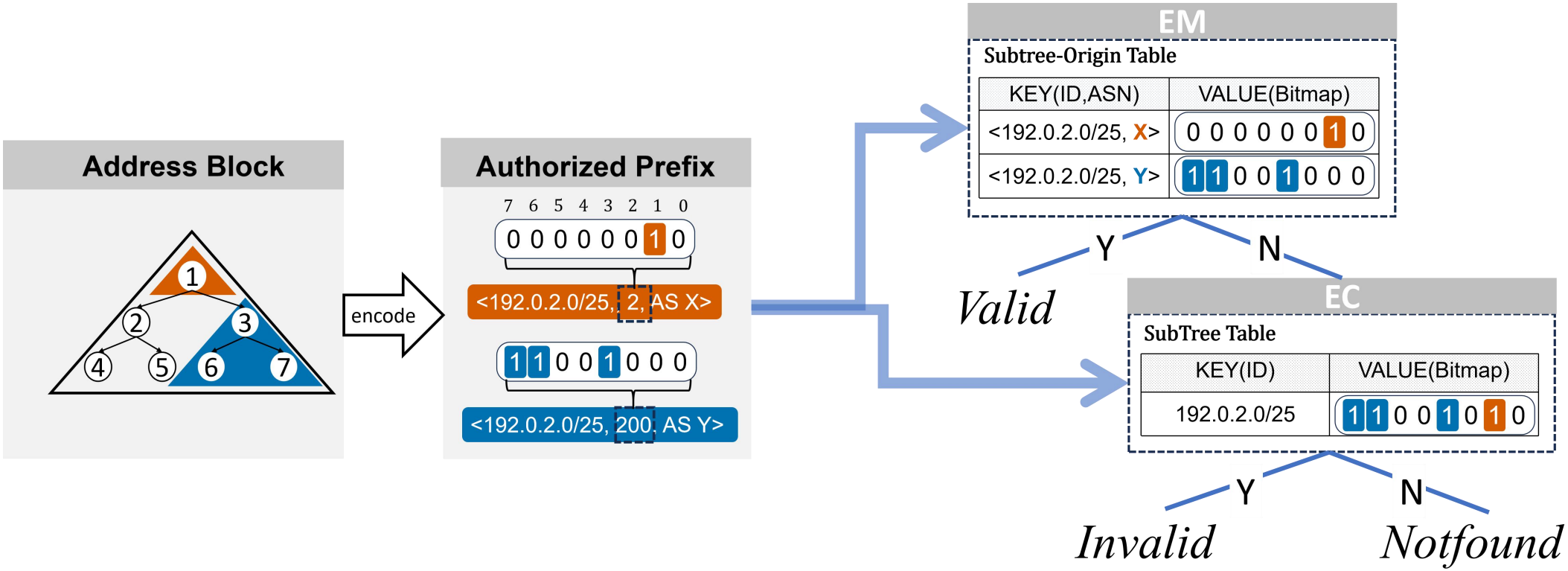
Prefix	status
Prefix1	covered
...	...

IPv6

We design a **binary search** strategy with time complexity **O(log(l))**.



h²ROV: a Hierarchical Hashing ROV Scheme



Redesigning ROV with the AP model

1. how to maintain ROAs at the granlarity of Authorized Prefixes ?

We adopt the bitmap-based encoding scheme [INFOCOM 22].

2. how to validate BGP routes with bitmap-encoded ROAs?

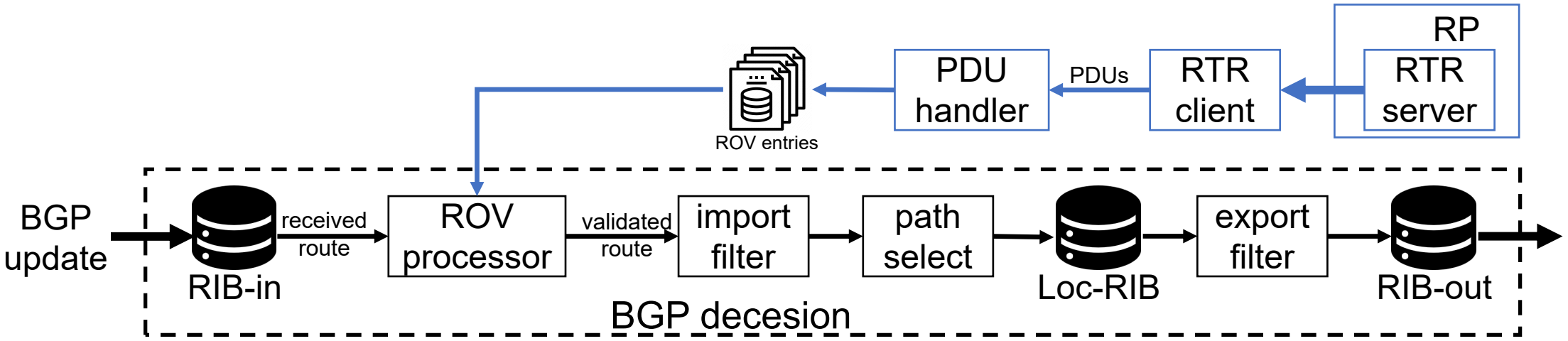
We propose a hierachical hashing scheme with two hash tables: SOT and STT.

3. how to implement the proposed scheme with BGP routers?

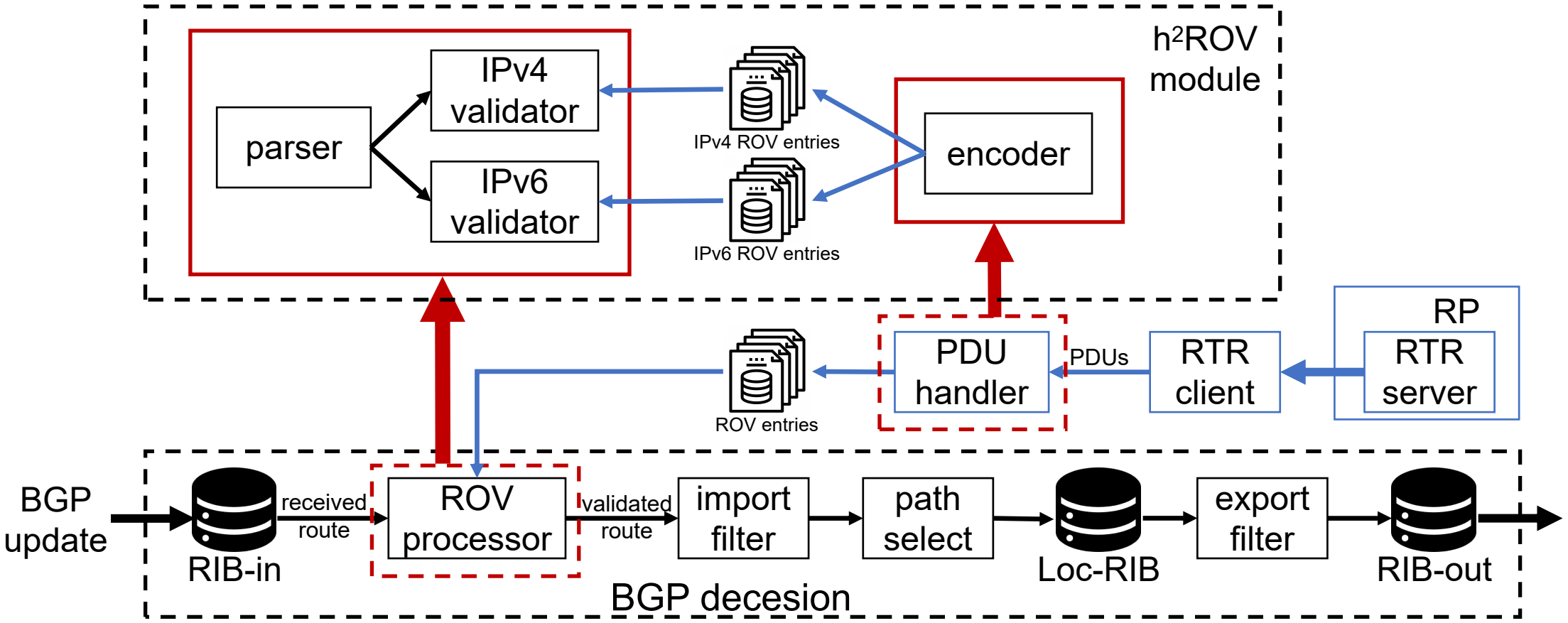
We refactor two system modules and verify the implementation with two software routers: FRRouting[1] and BIRD[2].

[1]. https://frrouting.org
[2]. https://bird.network.cz

System Implementation



System Implementation



- Implemented in two software routers: FRRouting and BIRD.
- Source Code: https://github.com/FIRLab-CNIC/h-2ROV.

Experiment Setup

- Experimental Environment
  - CPU: AMD EPYC 7742(2.2GHz, 128cores)
  - RAM: DRAM 128G

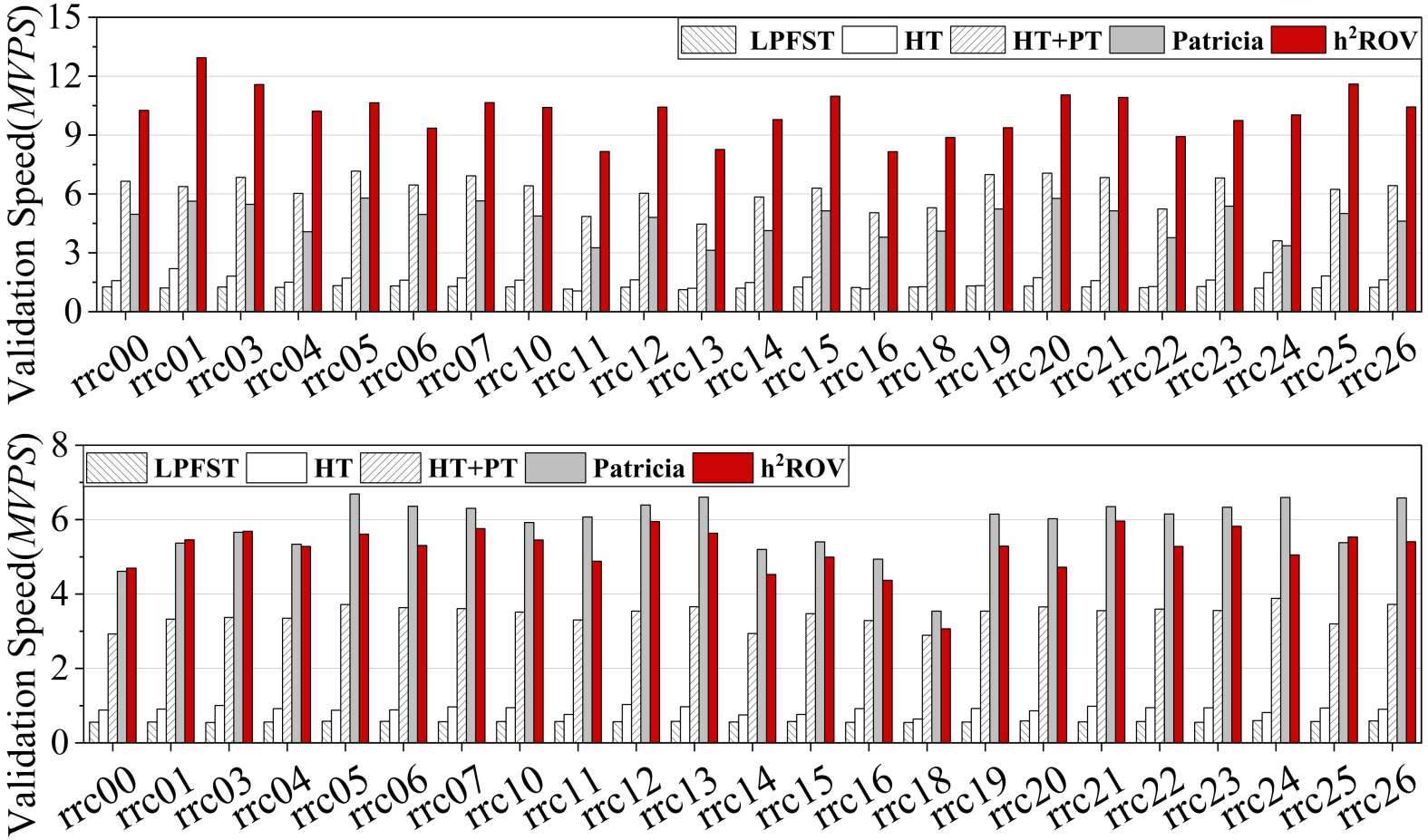
- Compared Schemes

Core Data Structure	name	organization
Longest Prefix First Search Tree	LPFST	RTRLib[1]
HashTable with Compressed Trie	HT+PT	BIRD[2]
Patricia	Patricia	BGP-SRx[3]
HashTable	HT	BIRD[2]

[1]. https://rtrlib.realmv6.org/
[2]. https://bird.network.cz
[3]. https://www.nist.gov/services-resources/software/bgp-secure-routing-extension-bgp-srx-software-suite

Performance Evaluation

ROA: from real world PRKI data
BGP updates: from RIPE RIS BGP data

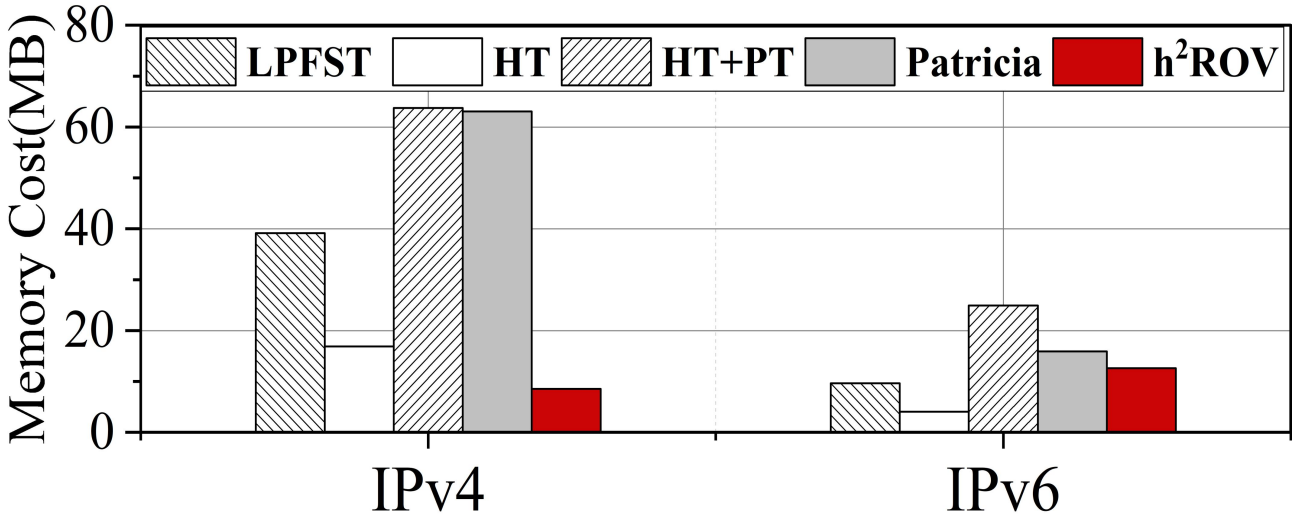


In IPv4, h²ROV outperforms all other schemes.

In IPv6, h²ROV outperforms LPFST, HT and HT+PT, while slower than Patricia.

Performance Evaluation

ROA: from real world PRKI data



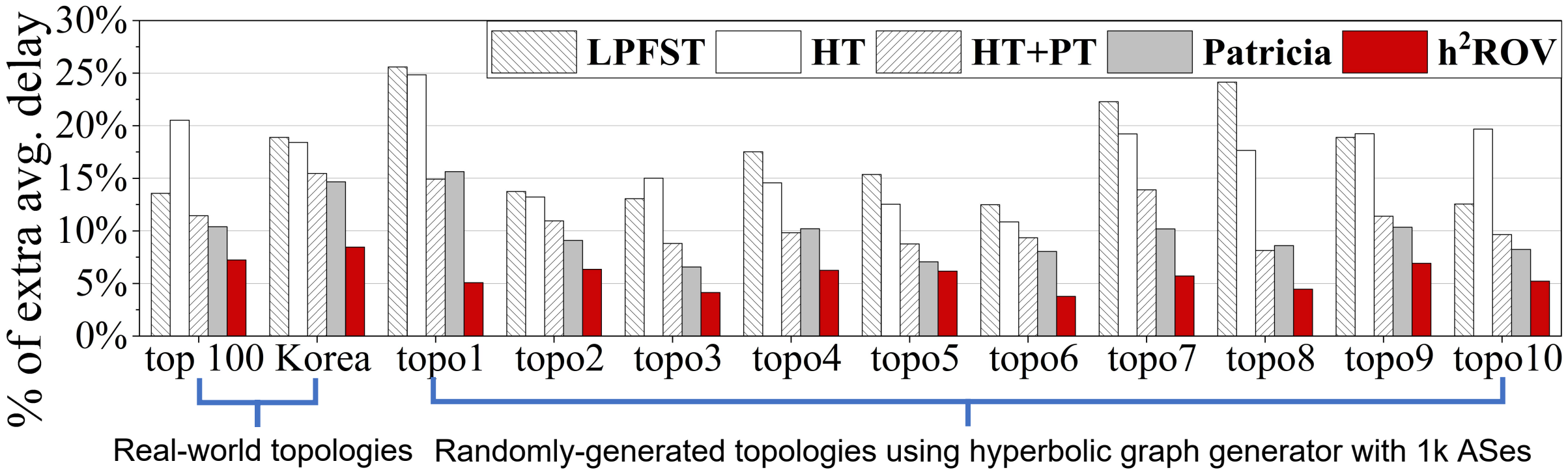
In IPv4, h²ROV consumes the least memory cost.

In IPv6, h²ROV consumes less memory than HT+PT and Patricia, while more than LPFST and HT.

System Evaluation

ROA: from real world PRKI data
BGP updates: from RIPE RIS BGP data

Emulation Platform: https://www.sernes.cn



h²ROV reduces the ROV-induced delays in BGP convergence by 30.4% ~ 64.7% compared to other schemes.

Conclusion

- The performance bottleneck of current ROV schemes lies in the ROV model using **Address Blocks**.To tackle this problem, We propose a brand-new ROV model based on **Authorized Prefixes.**
- Based on the new model, We present **h²ROV**, which archives Fast and Memory-Efficient validation.
- h²ROV considerably **mitigates ROV-induced routing convergence delays**.

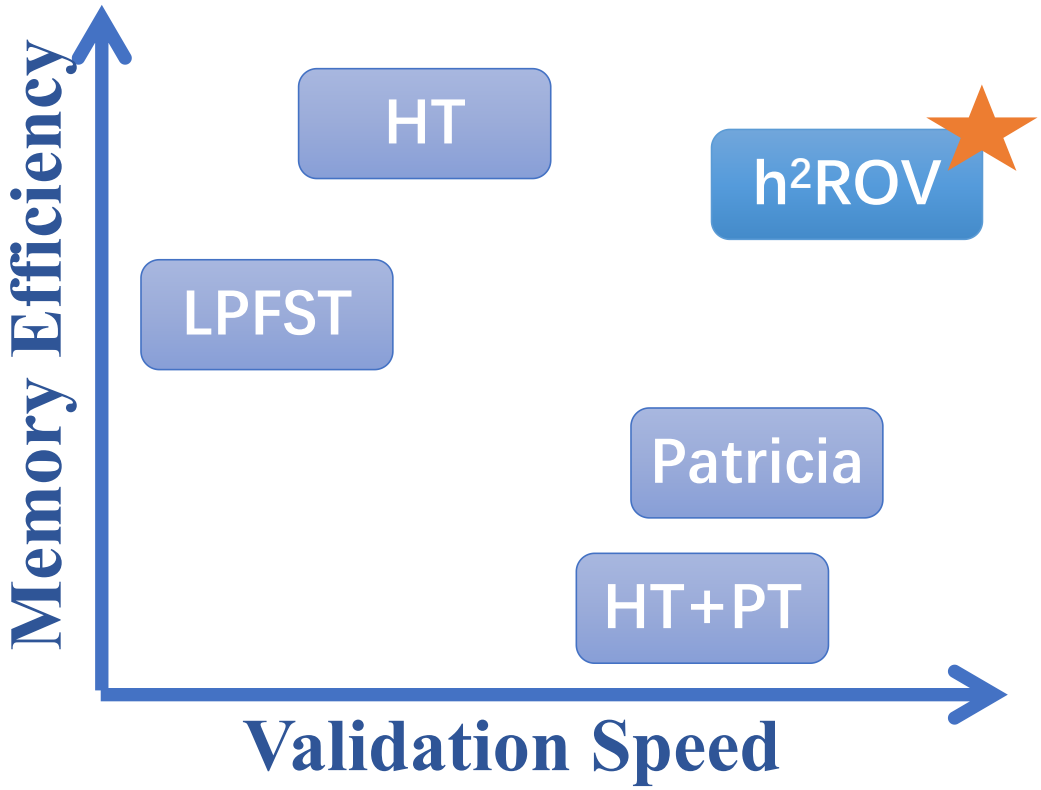
Thank you!

For more details, please read our paper:
From Address Blocks to Authorized Prefixes: Redesigning RPKI ROV with a Hierarchical Hashing Scheme for Fast and Memory-Efficient Validation

Contact: lybmath@cnic.cn

Emulation platform: https://www.sernes.cn

Code: https://github.com/FIRLab-CNIC/h-2ROV



中国科学院
计算机网络信息中心
Computer Network Information Center,
Chinese Academy of Sciences

SOUTHEAST UNIVERSITY
東南大學
1902
南京

中国科学院大学
University of Chinese Academy of Sciences

紫金山实验室
Purple Mountain Laboratories