APNet 2024

TAR: Traffic Adaptive IPv6 Routing Lookup Scheme

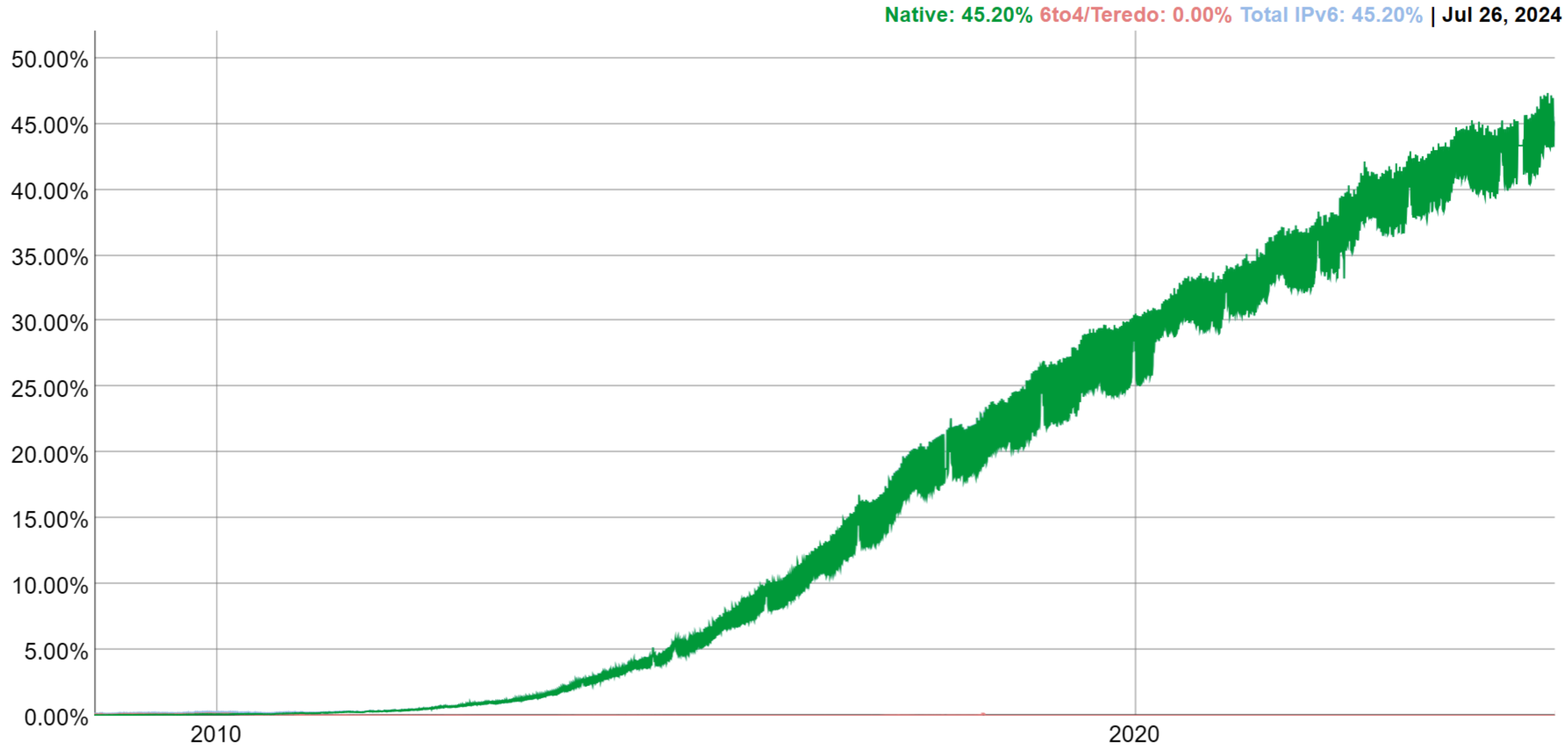
Xinyi Zhang[1,2], **Zhiyuan Xu[2]**, Huaiyi Zhao[2,3], Yanbiao Li[1,2], Gaogang Xie[1,2]

[1]Computer Network Information Center, Chinese Academy of Sciences, Beijing, China

[2]University of Chinese Academy of Sciences, Beijing, China

[3]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

IPv6 Adoption



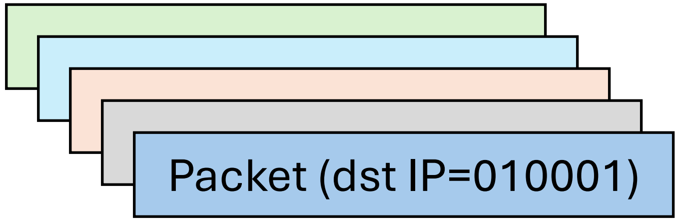
Google collects statistics about IPv6 adoption in the Internet on an ongoing basis. We hope that publishing this information will help Internet providers, website owners, and policymakers as the industry rolls out IPv6.

Google IPv6

"IPv6 – Google." Accessed: Jul. 28, 2024. [Online]. Available: https://www.google.com/intl/en/ipv6/statistics.html#tab=ipv6-adoption

IPv6 Lookup

- **Traffic**: actual packets moving through the network.
- **Ruleset**: the rules for processing and forwarding packets based on their destination IPv6 addresses.
- Principle: Longest Prefix Matching (LPM)



Rule #	Prefix	Next Hop
A	0*	√ 1
B	01000*	√ 2
C	0111*	× 3
D	10*	× 4
E	11*	× 5

$$len(A) = 1 < len(B) = 5$$

NextHop:2

Existing Software-based Lookup Schemes

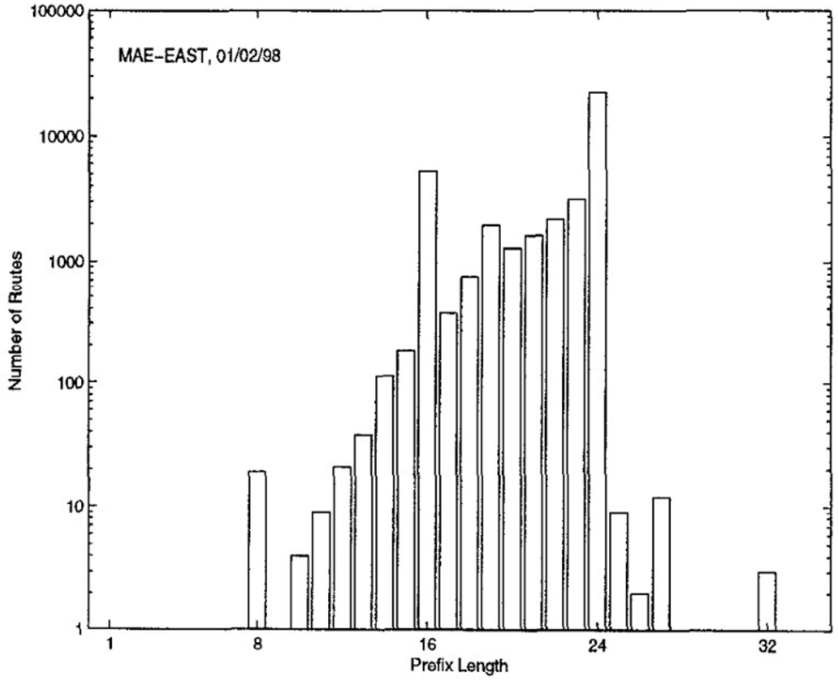
- Trie-based
  - DIR-24-8 (INFOCOM’98)
  - SAIL (SIGCOMM’14)
  - Poptrie (SIGCOMM’15)
  - Neurotrie (ICDCS’22)
- BST-based
  - ABST (SIGCOMM’97)
  - Hi-BST (GLOBECOM’18)

- When it comes to IPv6…
  - Longer prefix
  - Exploded address space
  - Heavier storage stress
  - Increased lookup complexity

[1] P. Gupta, S. Lin, and N. McKeown, “Routing lookups in hardware at memory access speeds,” in Proceedings. IEEE INFOCOM ’98, the Conference on Computer Communications. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Gateway to the 21st Century (Cat. No.98CH36169), San Francisco, CA, USA: IEEE, 1998, pp. 1240–1247. doi: 10.1109/INFCOM.1998.662938.
[2] T. Yang et al., “Guarantee IP lookup performance with FIB explosion,” in Proceedings of the 2014 ACM conference on SIGCOMM, Chicago Illinois USA: ACM, Aug. 2014, pp. 39–50. doi: 10.1145/2619239.2626297.
[3] H. Asai and Y. Ohara, “Poptrie: A Compressed Trie with Population Count for Fast and Scalable Software IP Routing Table Lookup,” SIGCOMM Comput. Commun. Rev., vol. 45, no. 4, pp. 57–70, Sep. 2015, doi: 10.1145/2829988.2787474.
[4] M. Waldvogel, G. Varghese, J. Turner, and B. Plattner, “Scalable high speed IP routing lookups,” in Proceedings of the ACM SIGCOMM ’97 conference on Applications, technologies, architectures, and protocols for computer communication, Cannes France: ACM, Oct. 1997, pp. 25–36. doi: 10.1145/263105.263136.
[5] T. Shen, X. Yu, G. Xie, and D. Zhang, “High-Performance IPv6 Lookup with Real-Time Updates Using Hierarchical-Balanced Search Tree,” in 2018 IEEE Global Communications Conference (GLOBECOM), Dec. 2018, pp. 1–7. doi: 10.1109/GLOCOM.2018.8647190.
[6] H. Chen, Y. Yang, M. Xu, Y. Zhang, and C. Liu, “Neurotrie: Deep Reinforcement Learning-based Fast Software IPv6 Lookup,” in 2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS), Bologna, Italy: IEEE, Jul. 2022, pp. 917–927. doi: 10.1109/ICDCS54860.2022.00093.

Ruleset characteristics

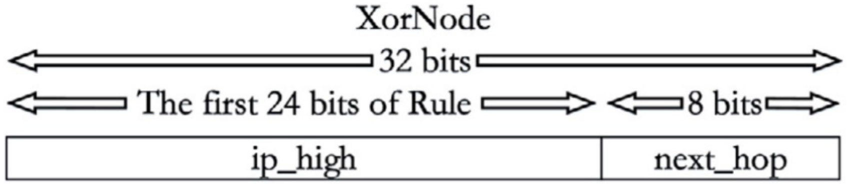
- Ruleset characteristics have been extensively studied to develop lookup algorithms



DIR-24-8 (INFOCOM'98)

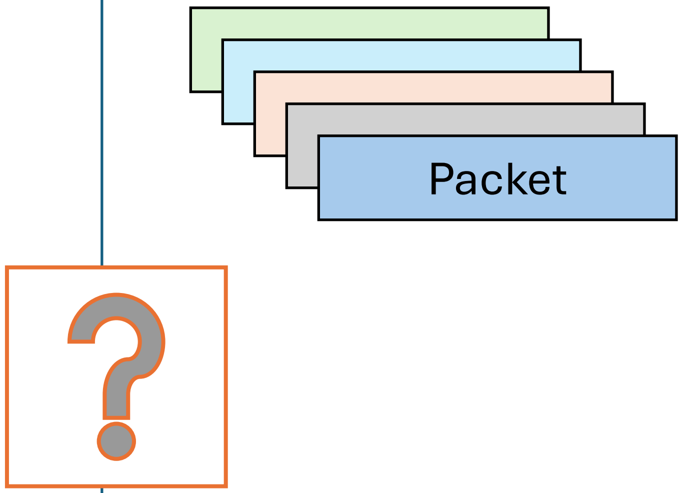
	Finding prefix length	Finding next hop
Prefix length 0~24	B_{16} , C_{16}, B_{24}	N_{16}, N_{24}
Prefix length 25~32	C_{24}	N_{32}

SAIL (SIGCOMM'14)



XorOffsetTrie (Computer Communications, 2022)

Neglected Factor: Traffic



Traffic: actual packets moving through the network.

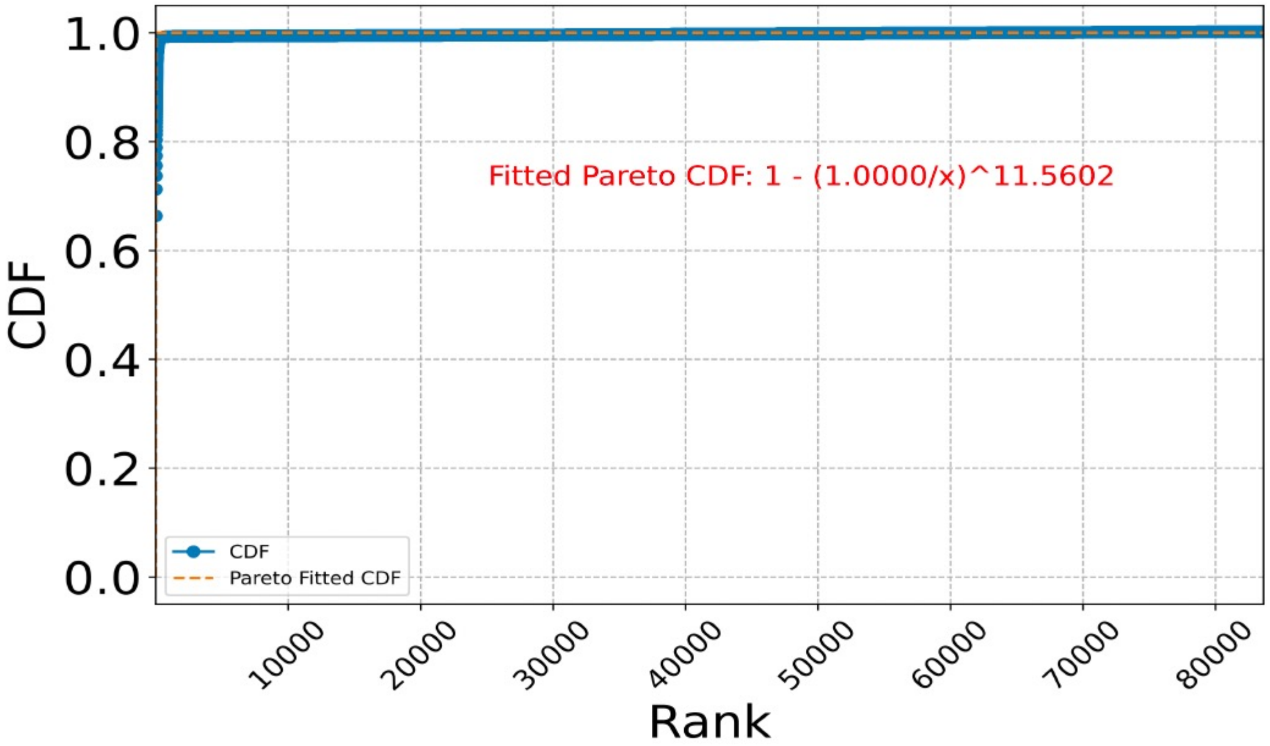
Longest Prefix Matching

Rule #	Prefix	Next Hop
A	0*	1
B	01000*	2
C	0111*	3
D	10*	4
E	11*	5

Ruleset: the rules for processing and forwarding packets based on their destination IPv6 addresses.

Traffic Distribution

- Observation: Traffic distribution is *skewed*!



CDF of IPv6 traffic with Pareto Fitted Curve

*MAWI. [n. d.]. MAWI Working Group Traffic Archive. https://mawi.wide.ad.jp/mawi/.

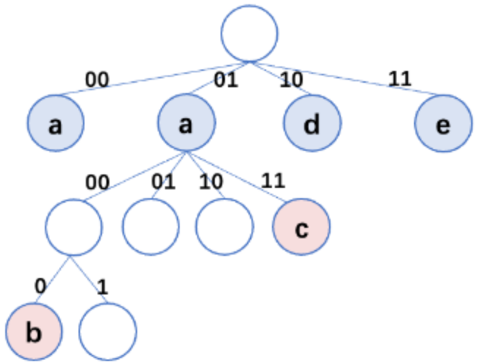
Traffic Adaptive Algorithm Design

- Trie-based
  - **Stride**: number of bits examined concurrently at a node
  - Nodes with higher traffic density: increase the stride
  - Nodes with lower traffic density: reduce the stride

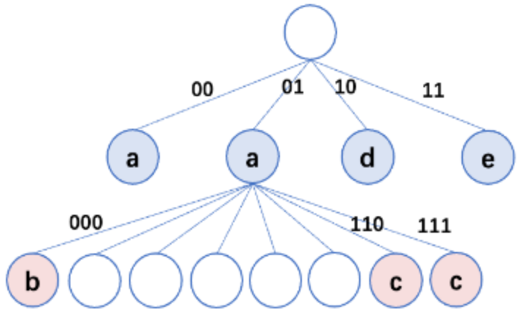
$$E(A(N)) = \begin{cases} 0 & R(N) = 1 \\ \sum\limits_{s \in sons(N, S(N))} E(A(s)) \cdot F(s) + 1 & \text{otherwise} \end{cases}$$

$$\begin{aligned} arg\min_{c} \quad & E(A(N_{\text{root of } T})) \\ s.t. \quad & M(T) = \sum_{N \in T} M(N, S(N), R(N)) \leq M_{\max} \end{aligned}$$

N: A node in the trie
$E(A(N))$: Expected number of subsequent memory accesses
F(s): Frequency of visiting node s
M(T): Memory consumption of a multi-bit trie T



Multibit trie



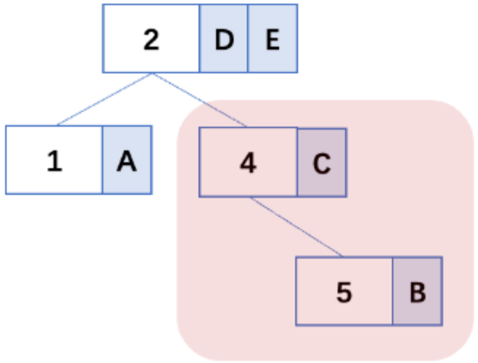
Traffic adaptive multibit trie

Traffic Adaptive Algorithm Design

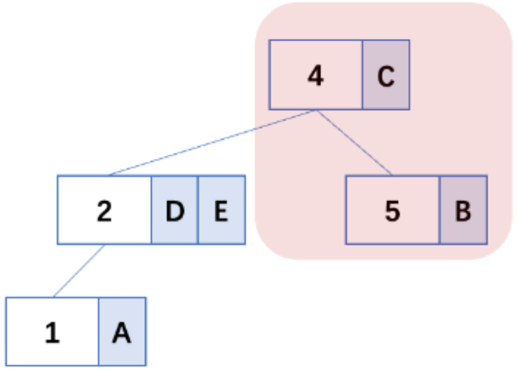
- BST-based
  - Place nodes with a higher probability of matching closer to the root

$$P(N) = \alpha \cdot \frac{R(N)}{\sum_{N \in Nodes} R(N)} + \beta \cdot F(N),$$

$R(N)$: the number of rules at node N
$F(N)$: the percentage of packets that finally find a match at node N
α and β: weights adjusting the impact of rule counts and hit counts



BST

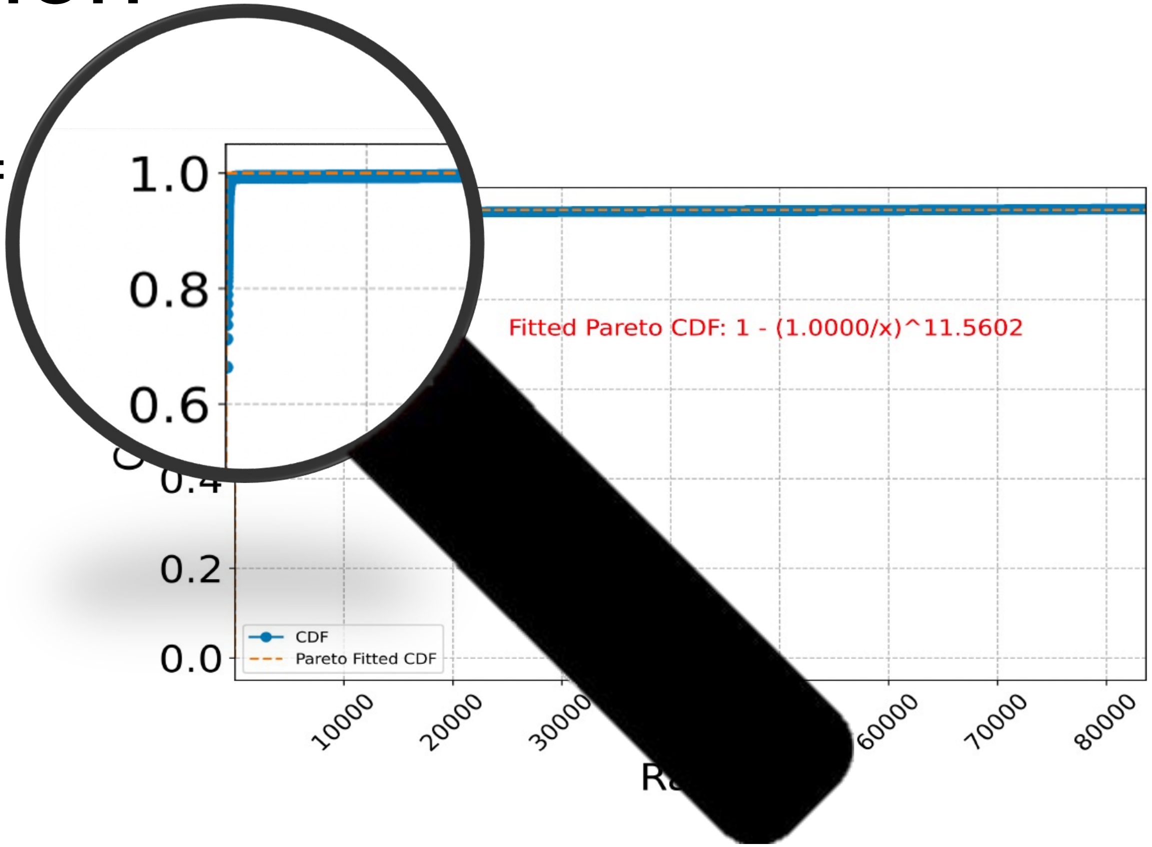


Traffic adavptive BST

Revisit Traffic Distribution

- Observation: A small number of destination IP addresses contributed to most of the network traffic.

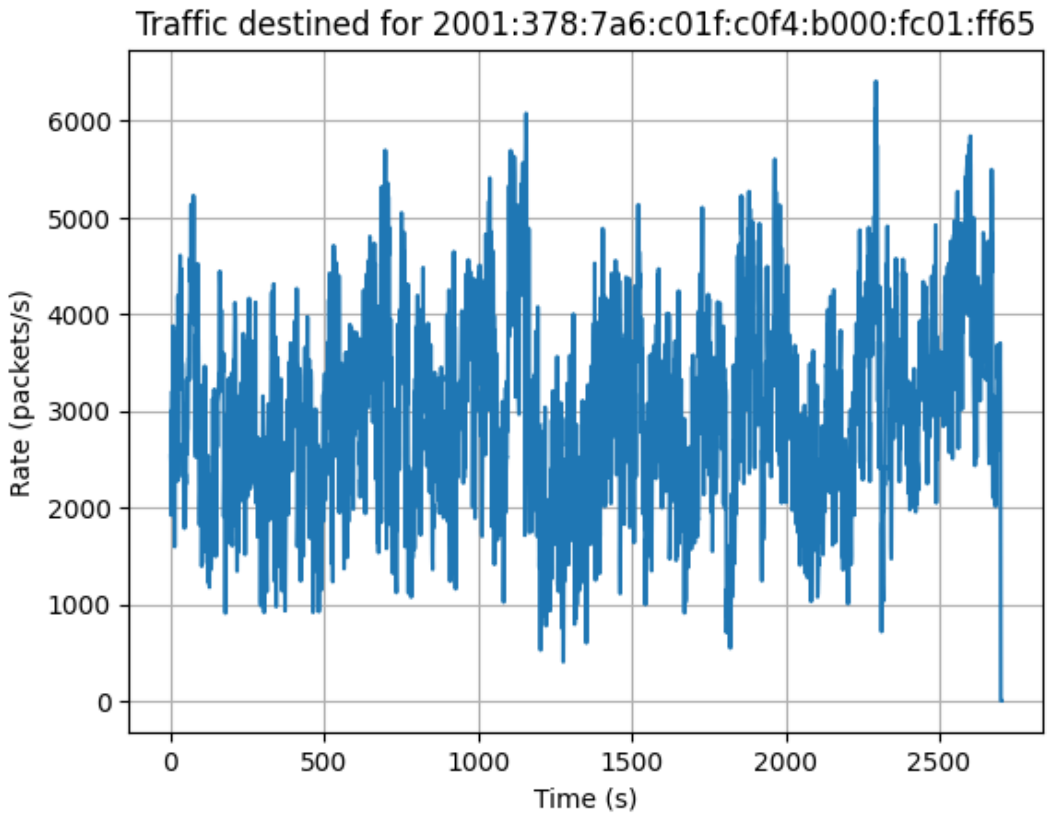
- Hint: Why not cache lookup results of heavy hitters?

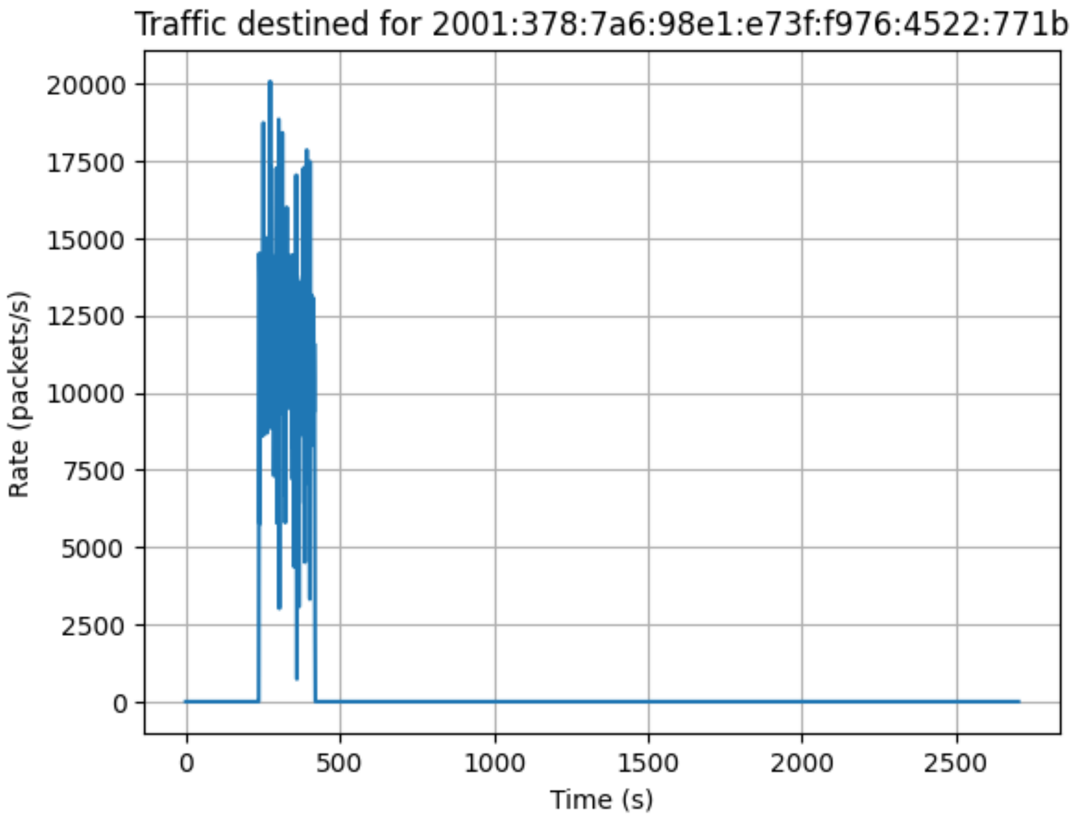


CDF of IPv6 traffic with Pareto Fitted Curve

*MAWI. [n. d.]. MAWI Working Group Traffic Archive. https://mawi.wide.ad.jp/mawi/.

Heavy Hitters





Heavy-Hitters in real-world traffic

*MAWI. [n. d.]. MAWI Working Group Traffic Archive. https://mawi.wide.ad.jp/mawi/.

TA-Cache Design

Desiderata of this cache table:

- Only caches heavy hitters
- Avoid cache table thrashing
- Promptly accommodates newly-emerged heavy-hitters
- Compact size

Destination IP Address	Counter	Next Hop
2001:378:f1c:...	18213	5
2001:378:7a6:...	15213	3

TA-Cache Table

Update procedure:

- $idx = hash(ip_{dst})\ mod\ TABLE\_SIZE$
- If $tb[idx]$ is empty, insert a new entry.
- Otherwise:

$$tb[idx].counter = \begin{cases} tb[idx].counter + 1 & \text{if ip matches} \\ tb[idx].counter \times \alpha & \text{if ip does not match} \end{cases}$$

- Removes an entry when its counter falls below 1.

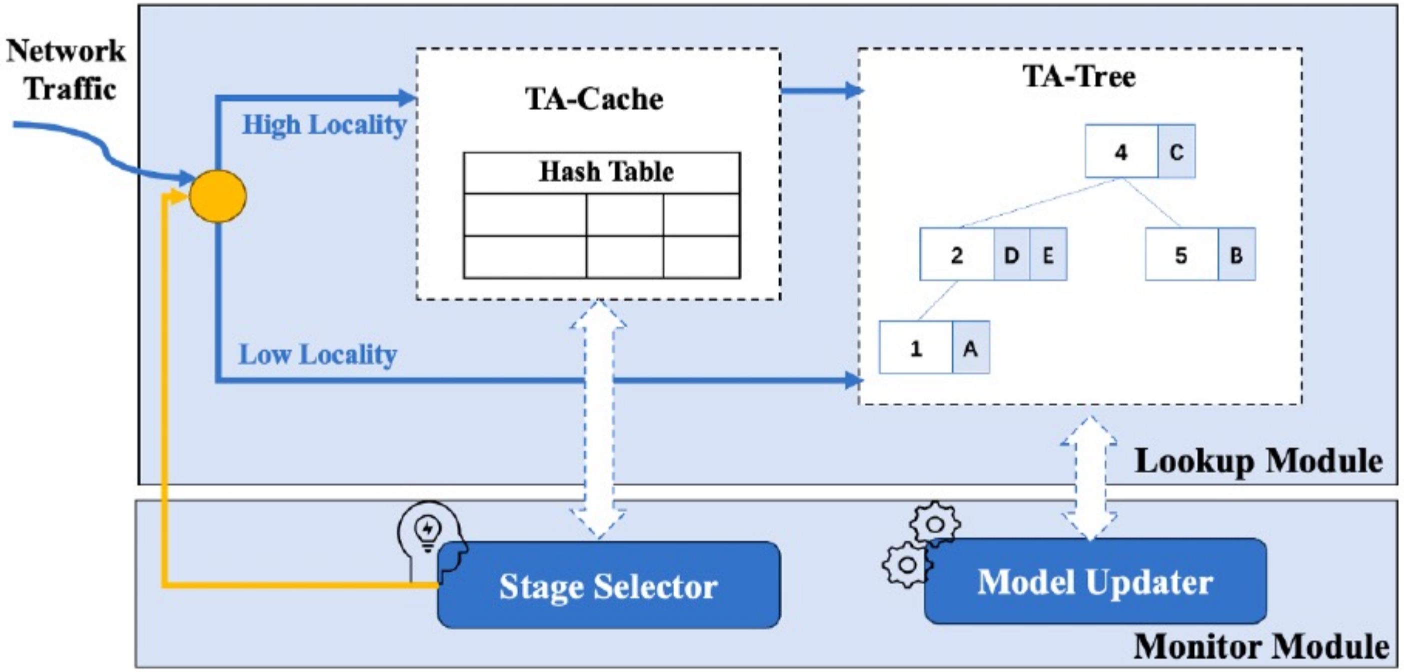
System Implementation

Lookup Module:

- Two-stage lookup process
  - ***TA-Cache:*** Deal with heavy-hitters and net bursts
  - ***TA-Tree:*** Complete lookup structure, optim for statistical traffic patterns.

Monitoring Module:

  - ***Stage Selector:*** For low network traffic loc Stage Selector deactivates the TA-cache ar reroute packet queries directly to the TA-Tr
  - ***Model Updater:*** Update the lookup structu when the traffic distribution changes to maximize performance.



Evaluation Setup

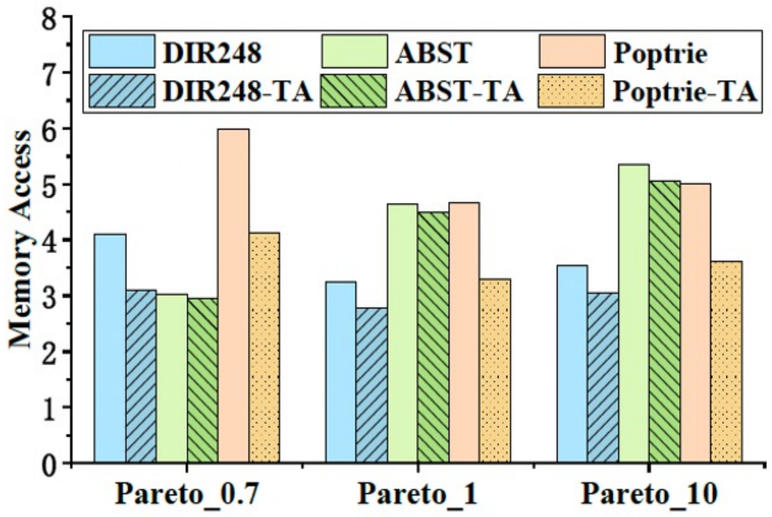
- Platform
  - Intel Xeon Gold 6338, Mellanox ConnectX-5 100G NICs
- Ruleset
  - Real FIBs from RIPE RRC00*
- Trace
  - Simulated Trace: Sample from Pareto distribution
  - Real Traffic: Derived from MAWI Dataset
- For algorithm comparison:
  - One dedicated server running algorithm prototypes. Simulated Trace.
- For system evaluation:
  - One server running the TAR system, another running a packet sender. Both programs are implemented under DPDK. Real Traffic.

*RIPE. [n. d.]. Route collectors. https://ris.ripe.net/docs/route-collectors/#bgptimer-settings. ([n. d.]).

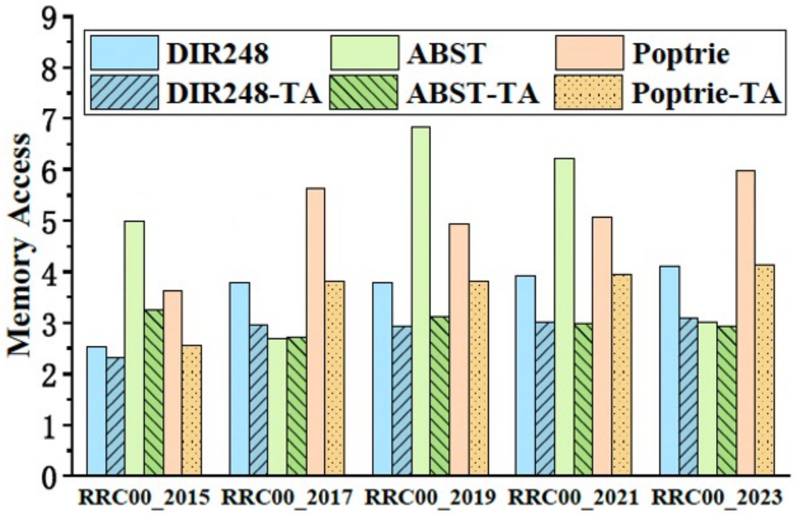
Algorithm Evaluation (1/2)

- Memory Access

Traffic adaptive methods for DIR248, ABST, and Poptrie respectively improve performance by 1.1~1.3 X, 1.5~2.2X, and 1.3 ~1.5 X compared to the original algorithms.



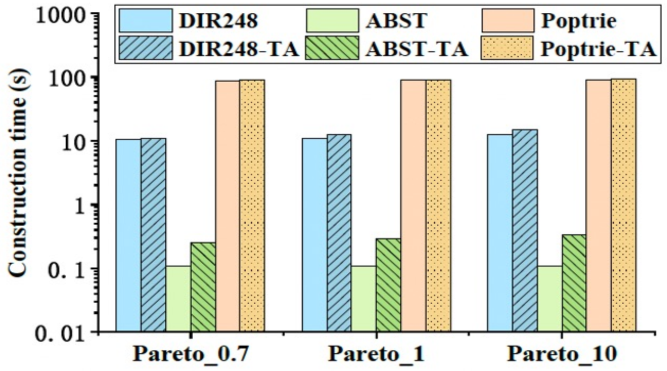
with different pareto b parameters (a=1)



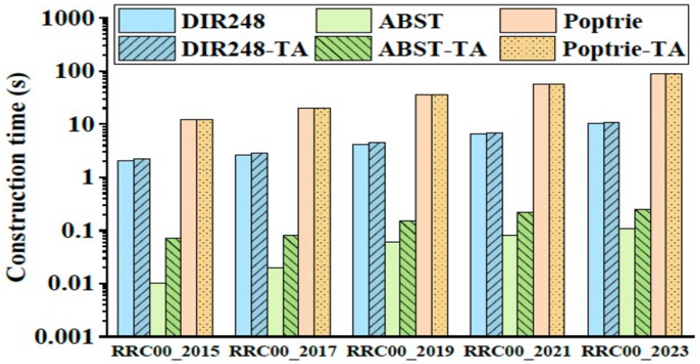
with different dataset (pareto: a=1, b=0.7)

Algorithm Evaluation (2/2)

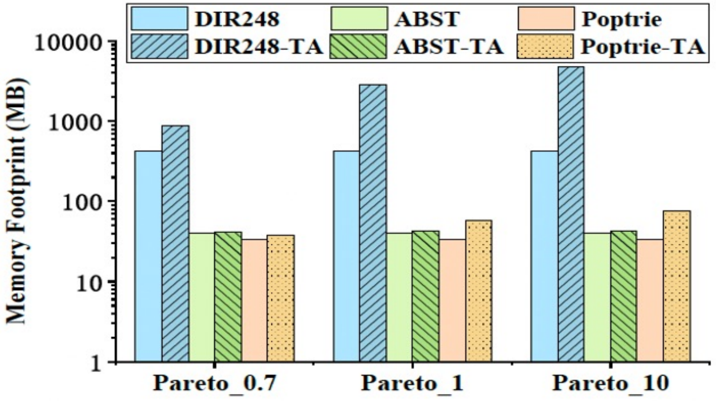
- Construction time and memory footprint are kept at a reasonable level



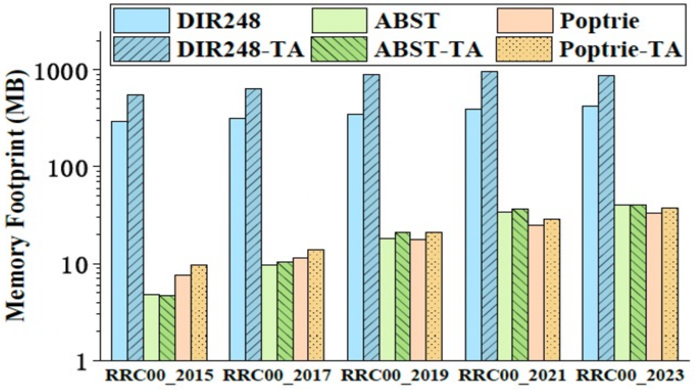
with different pareto b parameters (a=1)



with different dataset (pareto: a=1, b=0.7)



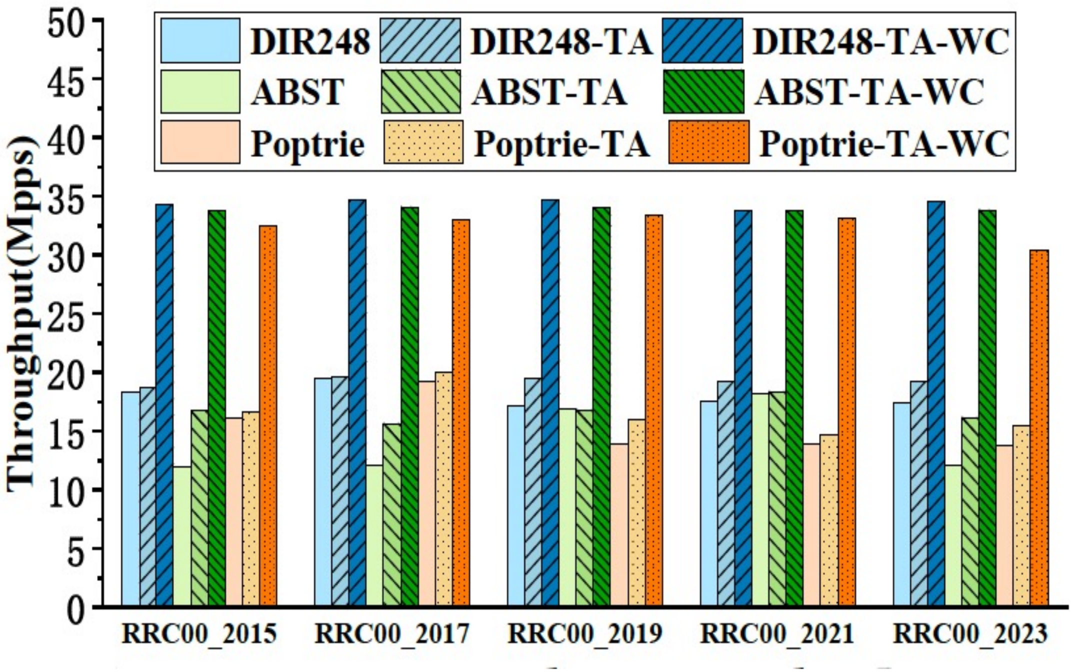
with different pareto b parameters (a=1)



with different dataset (pareto: a=1, b=0.7)

System Evaluation

- Throughput of TAR systems is 1.8 ~2.0X, 1.9 ~2.8X, and 1.7~2.4X that of systems using only basic algorithms.



System evaluation with DPDK

Conclusion

- TAR
  - The *first* IPv6 routing lookup scheme that utilizes the *characteristics of network traffic*.
  - Performance improvement of 1.7~2.8X in terms of system throughput.
  - Proof-of-concept: leveraging traffic characteristics could be beneficial

- Future work
  - Deeper analysis into real-world traffic distribution
  - Refine methods of utilizing traffic characteristics

TAR: Traffic Adaptive IPv6 Routing Lookup Scheme

Thank you!