Routing and Addressing with Length Variable IP Address

Shoushou Ren, Delei Yu, Guangpeng Li, Shihui Hu

Huawei Technologies Co., Ltd.

Ye Tian, Xiangyang Gong

Beijing University of Posts and Telecommunications

Robert Moskowitz

HTT Consulting, LLC

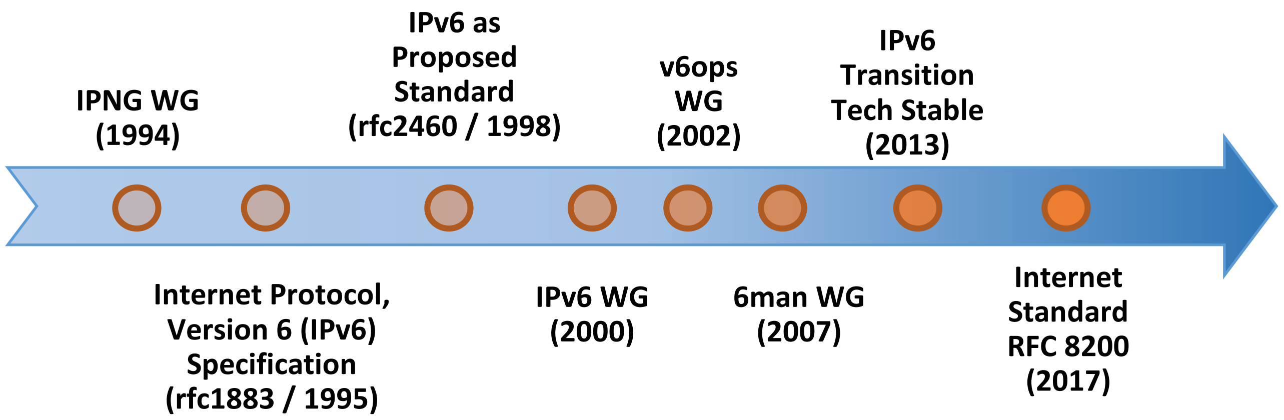
Backgrounds and Motivation

- Address exhaustion of IPv4
  - IPv4 address is 32 bits length
  - Provide limited address space (less than 4.3billion available addresses)
  - The last two /8 address blocks of IPv4 were assigned by IANA in 2011
- IPv6 was designed to solve the problem of address exhaustion
- However…

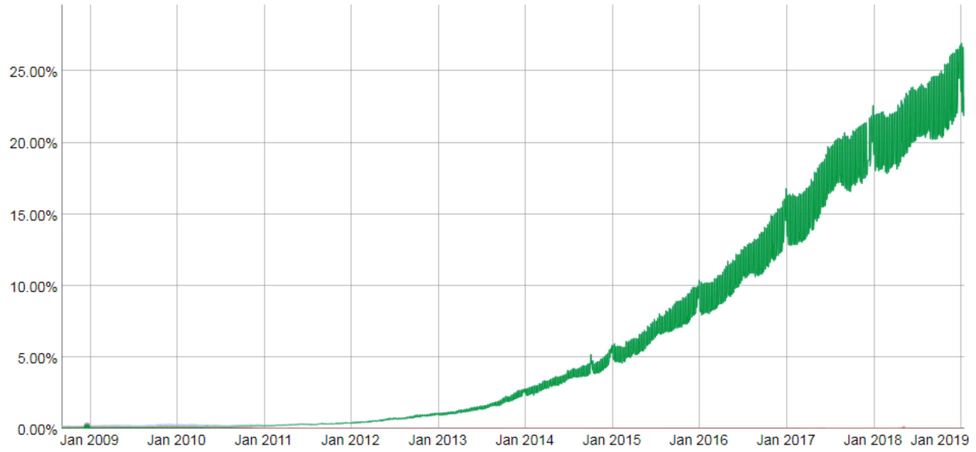
Backgrounds and Motivation

- **Poor compatibility with IPv4**
  - IPv4/6 are with separated address space and completely different protocol formats
  - More than one billion devices need to update, high costs
  - Information asymmetry with patching solution, like NAT64

- **IPv6 is NOT infinite**
  - Applied and assigned with prefix size of /32
  - By July 2017, 215959 blocks of /32 assigned （more than 1/20000 of all the available IPv6）



The percentage of users that access Google over IPv6.



From https://www.google.com/intl/en/ipv6/statistics.html

Backgrounds and Motivation

☐ Why IPv4 was designed with 32bits

- only for experiments, easy for processing, and no special reason

At the Google IPv6 Conference 2008, he hosted a panel discussion titled *What will the IPv6 Internet look like?* during which he recounted:

> The decision to put a 32-bit address space on there was the result of a year's battle among a bunch of engineers who couldn't make up their minds about 32, 128 or variable length. And after a year of fighting I said — I'm now at ARPA, I'm running the program, I'm paying for this stuff and using American tax dollars — and I wanted some progress because we didn't know if this is going to work. So I said 32 bits, it is enough for an experiment, it is 4.3 billion terminations — even the defense department doesn't need 4.3 billion of anything and it couldn't afford to buy 4.3 billion edge devices to do a test anyway. So at the time I thought we were doing a experiment to prove the technology and that if it worked we'd have an opportunity to do a production version of it. Well — [laughter] — it just escaped! — it got out and people started to use it and then it became a commercial thing.

Then we said "**how many countries are there?**" (two networks per country, how many networks?) and we didn't have Google to ask, so **we guessed at 128** and that would be 2 times 128 is 256 networks (that's 8 bits) and then we said "**how many computers will there be on each network?**" and we said "**how about 16 million?**" (that's another 24 bits) so we had a 32-bit address which allowed 4.3 *billion* terminations — which I thought in 1974/3 was enough to do the experiment!

By Vint Cerf, @ Google's IPv6 conference in 2008

☐ The address size was once discussed as a key issue in 1990s

- IETF IPng
  - 16 bytes scheme, typical proposal SIPP[1] (finally became IPv6)
  - 20 bytes scheme [2]
  - Variable length address scheme, typical proposal PIP[3]
- OSI (Open System Interconnect)
  - Variable length address scheme, typical proposal NSAP

[1]Robert Hinden. 1994. Simple internet protocol plus white paper
[2]S. Bradner and A. Mankin. 1995. The Recommendation for the IP Next Generation Protocol
[3]Paul F Tsuchiya. 1992. Pip Overview and Examples. Technical Report.

Backgrounds and Motivation

- None of the above proposals were wrong or others were right
- Some failed just because of the technique limitation at that time
- However, time is different now…

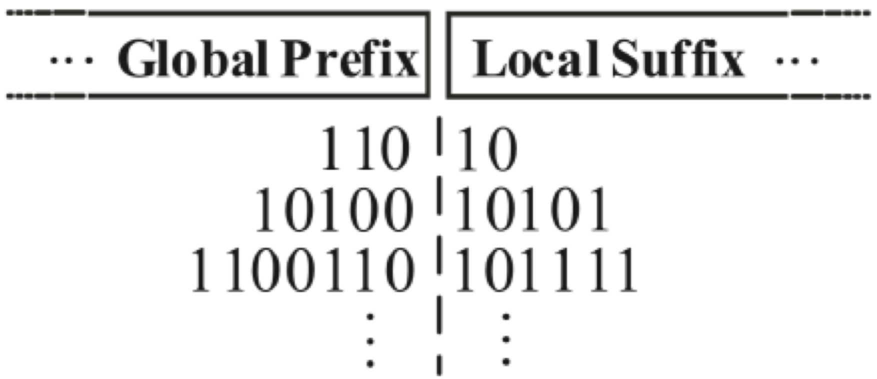
Backgrounds and Motivation

- Stronger and more powerful devices
- Demands for different address length are real
  - The number of Internet devices is growing faster than we think
    - 32bits -> 128bits, a long and hard transition way
    - 128bits -> longer ?
  - Many scenarios (especially IoT) requires shorter addresses
    - Closed scenario with short address
    - Light packet header in LLN
    - Many patching methods have been proposed to compress the 128bits, e.g. 6LoWPAN
- Why not to design a flexible address space that can
  - evolve smoothly itself, avoiding updates and transition
  - accommodate to different demands, avoiding patching

Flexible Address System (FAS)

Flexible Address Model

- FAS is composed of Global Prefix (GP) and Local Suffix (LS)
- GP is a centralized allocated unbounded-hierarchy, facilitating packet forwarding between multiple public connectivity providers
- Can be simply treated as a natural number and its size varies accordingly
- LS is a flexible address, assigned by the local network
- Both GP and LS are with infinite space



Flexible address space

Flexible Address System (FAS)

Textual Representation

- Flexible address can be uniformly divided into several units
- Flexible address can be represented textually with the conventional *dotted decimal form*
  - Auxiliary zeros can be added on the most left side，enabling the longest matching algorithm
  - $\underbrace{\underline{10}}_{\text{GP}}\ \underbrace{0100\ 0110}_{\text{LS}}$ $(\underline{2}.4.6) \rightarrow$ $\underbrace{\underline{0010}}_{\text{GP}}\ \underbrace{0100\ 0110}_{\text{LS}}$ $(\underline{0.2}.4.6)$
- As to flexible routing prefixes, denoted as *flexible-address/prefix-length*
  - 10 0100 0000, with 4 bits as mask number → 2.4.0/6 or 0.2.4.6/10
- Effective address v.s. expression address
  - $\underline{10}$ 0100 0000: effective address, 10 bits effective length
  - $\underline{0010}$ 0100 0000: expression address, 12 bits expression length

Flexible Address System (FAS)

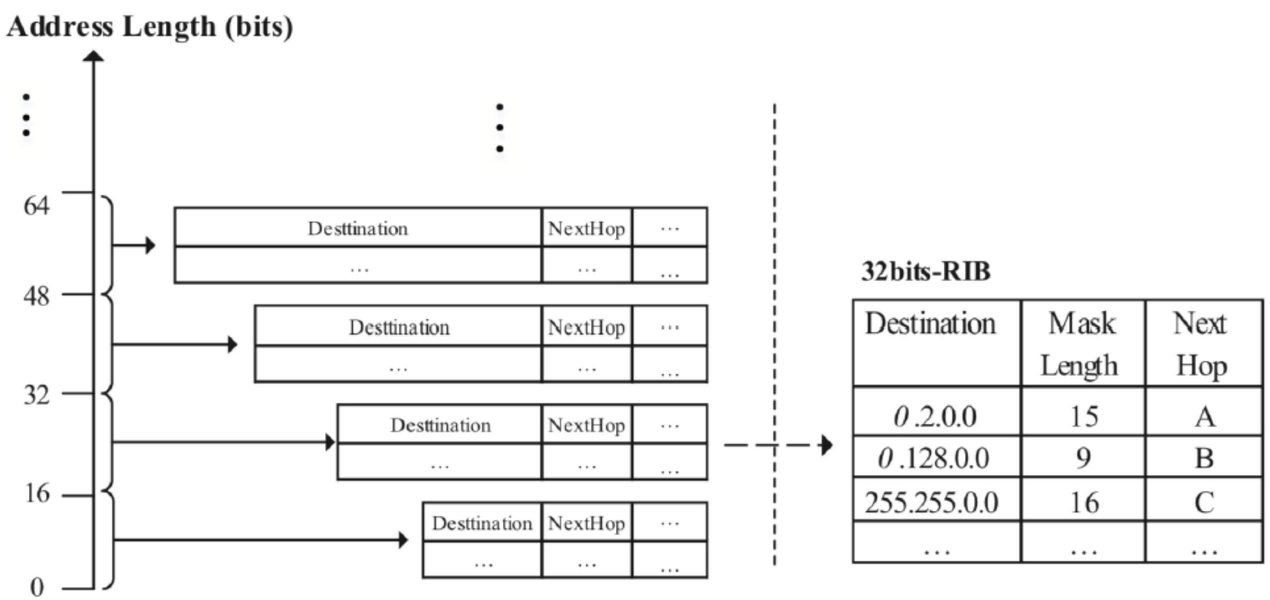
Basic Allocation Principles

- Centralized allocated by organization like IANA
- The value of *GP* and the size of *LS* need to be prescribed
- *LS* can be further divided into smaller subnets or assigned as specified local addresses
- A *GP* can be assigned to multiple organizations with different size of *LS*
  - *e.g.*, $\underbrace{\underline{2}}_{\text{GP}}.\underbrace{0.0}_{\text{LS}}$ → organization A $\underbrace{\underline{2}}_{\text{GP}}.\underbrace{0.0.0}_{\text{LS}}$ →organization B
- One address block can NOT be assigned twice, even with different *GP*
  - *e.g.*, if $\underbrace{\underline{2.2}}_{\text{GP}}.\underbrace{\underline{0.0}}_{\text{LS}}$ → *organization A*, <u>2.2.0</u>.0 or <u>2</u>.2.0.0 cannot be assigned

Routing and Addressing

Storing Routing Entries with Multiple Routing Information Bases (RIBs)

- FAS space is divided into continuous and disjoint intervals according to address length
- Each interval is bound to a RIB, routing entries belonging to same interval will be stored into a same RIB
- RIB-width: maximum address length of each interval

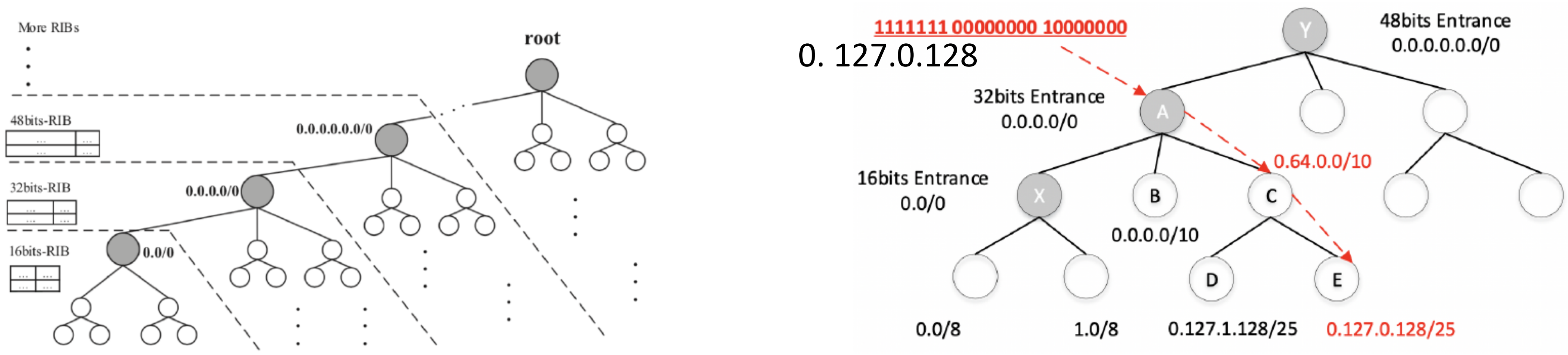


An example for multiple RIBs

Routing and Addressing

Lookup with *Multi-Entrance-Trie*

- RIBs are organized based on a new data structure, i.e., ***Multi-Entrance-Trie (METrie)***
- Multiple entrance nodes, represented by solid circles
- Each part stores routing information of its corresponding RIB, while the solid circle is the lookup entrance to that RIB
- Descendant nodes stores searching keys with the same RIB-width, each part is organized in Trie
  Lookup procedure is relatively straightforward
  - Effective length determination →RIB selection with RIB-width → longest prefix matching

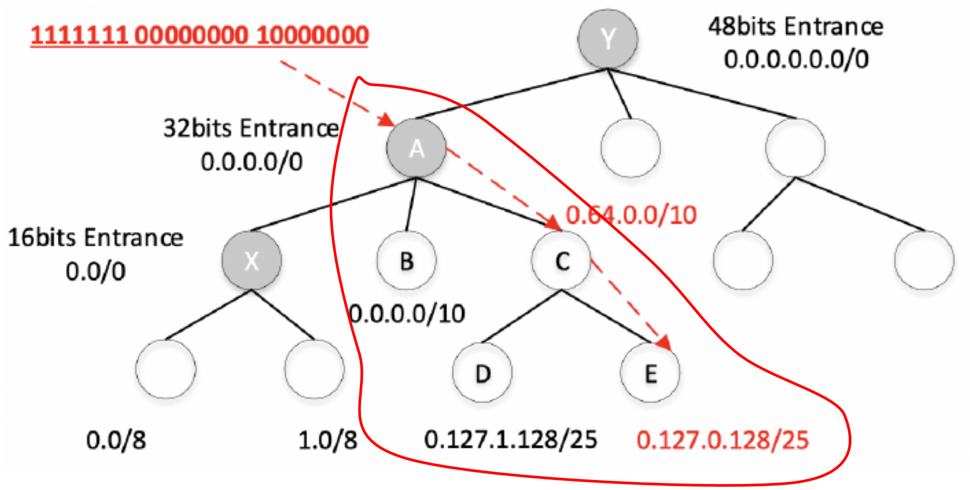


An example for Multi-Entrance-Trie

Routing and Addressing

Dynamically Pruning and Expanding *METrie*

- Pruning obsoleted branches
  - In some cases, routing entries stored in an existing RIB might be obsoleted
  - Corresponding branches need to be deleted

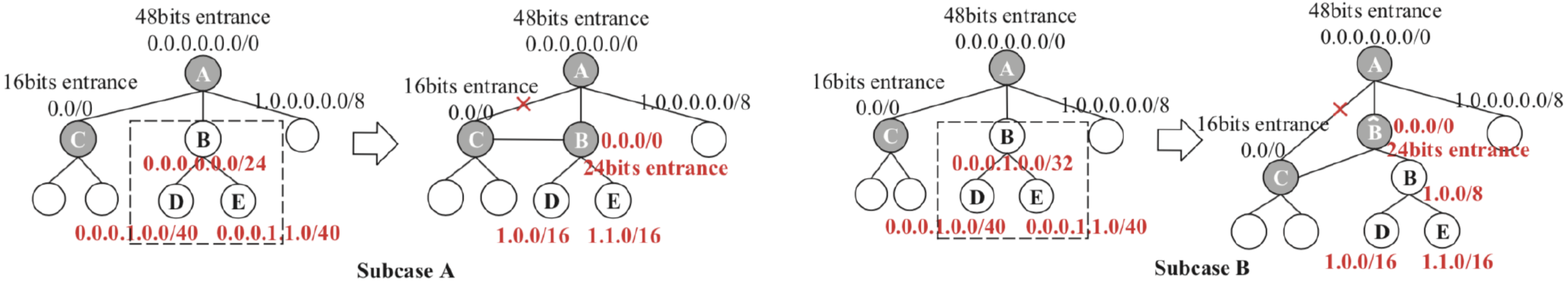


- Expanding upper branches
  - If an address or prefix is longer than the biggest RIB-width, existing *METrie* will be expanded with a new root

Routing and Addressing

- Inserting new entrance node in the middle of *METrie*
  - a router may need to create a new RIB which is 'smaller' than its biggest RIB-width for lookup efficiency reasons
  - a new entrance node needs to be added on the mountainside of the current *METrie*

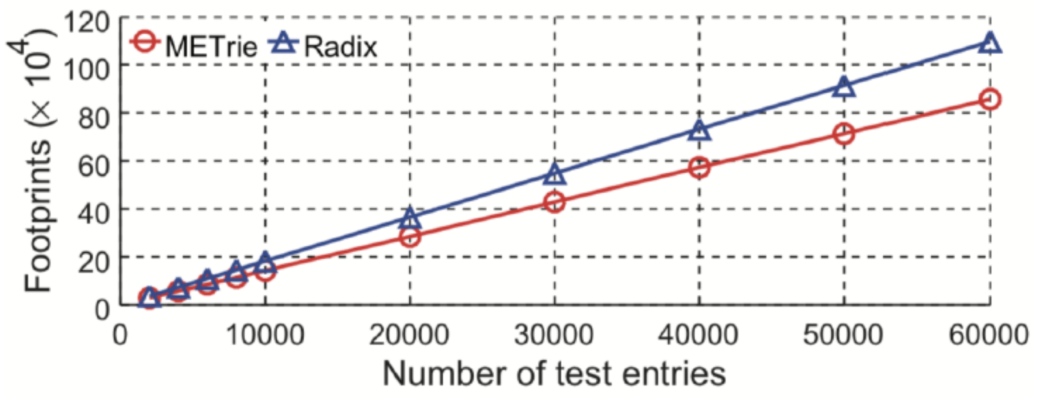
- Case A: the new entrance node already exists as a child node
- Case B: the new entrance node doesn't exist on current *METrie*



Inserting a new entrance node on METrie

Performance Evaluation

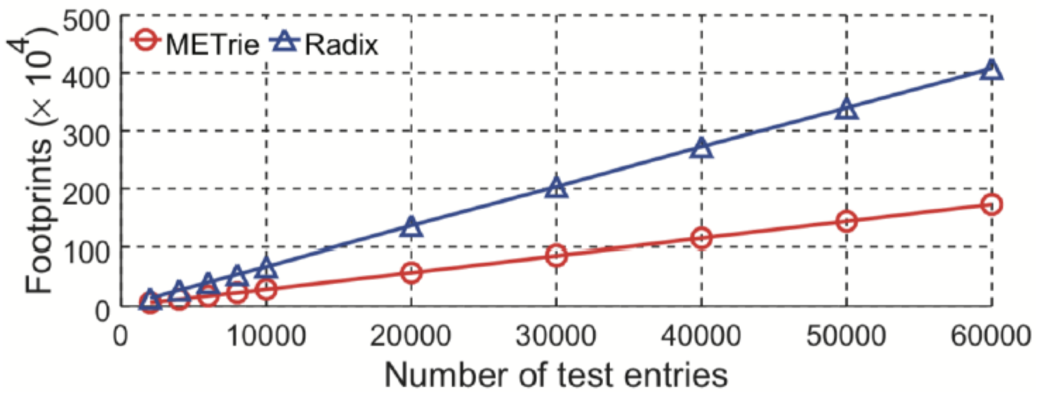
- Setup – for comparison with IPv4
  - 50,000 routing entries are generated randomly to initiate the METrie with length in 16 ~32 bits
  - *METrie* is designed with four entrance nodes while the corresponding RIB-width are 20bits, 24bits, 28bits and 32bits
  - The radix tree for IPv4 is also initiated and constructed with 50,000 randomly generated IPv4 routing entries
- Criterion
  - The number of nodes on *METrie* and radix tree that have been visited when searching routing entries



lookup performance of FAS and IPv4

Performance Evaluation

- Setup – for comparison of IPv6
  - 50,000 routing entries are generated randomly to initiate the METrie with length in 32 ~128 bits
  - *METrie* is designed with four entrance nodes while the corresponding RIB-width 32bits, 64bits, 96bits and 128bits
  - The radix tree for IPv6 is also initiated and constructed with 50,000 randomly generated IPv6 routing entries
- Criterion
  - The number of nodes on *METrie* and radix tree that have been visited when searching routing entries



lookup performance of FAS and IPv6

Conclusion & Future work

- Many applications propose different demands on IP addresses, while fixed length addresses cannot accommodate to them well

- A flexible address space benefits the long-term evolution of the Internet in the following aspects, this paper only proposes a candidate solution on this direction
  - Evolve and extend smoothly, without defining new address space and protocol formats
  - Allows people to apply for an address space with suitable length
  - Addresses with different length can communicate with each other

- We are now working on a flexible Internet protocol than can provide
  - Length variable IP address
  - More flexible and power functions than conventional IP

- A demonstration is also under developing

THANK YOU