Can Longest Prefix Matching Make The Path Length Shorter?

Xia Yin, Weiguo Zhang, Jianping Wu, Wei Zhang
Dept. Computer Science and Technology, Tsinghua University, Beijing, China, 100084
yxia@tsinghua.eud.cn, {zwg, wjp, zwei}@csnet1.cs.tsinghua.edu.cn
doi:10.4156/jdcta.vol4. issue8.19

Abstract

It has been recognized that the existing routing architecture of today's Internet is facing scalability problems. Although deployment of CIDR (Classless Inter-Domain Routing) slows the growth of global routing tables, LPM (longest prefix matching), which is greedily pursuing for more-specific prefixes, seriously restricts the advance of scalable routing. The prevalent LPM algorithm for routing lookup is considered the best path to the destination, which logically makes sense. However, we find that LPM algorithm does not necessarily provide shortest path length for routing lookup in most cases with comparison experiment. To find more general prefixes, we first present SPM (shortest prefix matching) algorithm, and then make a comparative evaluation between SPM and LPM by using the metric of average AS path length. In conclusion, we analyze the reasons that cause longer average AS path length for LPM algorithm in most cases and discuss the impact of LPM to scaling routing.

Keywords*: CIDR, LPM, SPM, Routing Scalability*

1. Introduction

Internet routing and addressing system is facing serious scaling problems and the most immediate concern is the rapid Default Free Zone (DFZ) routing table size inflation and increasing update churns [1]. To solve the scaling problem, CIDR and IPv6 were advanced in tandem: one for the immediate solution and the other for the long term. First of all, CIDR, proposed in 1993, was originally as a relatively short-term solution, with an expected lifespan of approximately three to five years, during which a more permanent addressing and routing architecture would be designed and implemented. Surprisingly CIDR has far outlasted its anticipated lifespan and has become the mid-term solution to combat growing routing table sizes and IP address space depletion [2]. Hopefully, the deployment of IPv6 will help solve the IP address shortage crisis. The fact is that IPv6 also utilizes CIDR routing technology in the same way as IPv4 so as to guarantee the utility efficiency of the IP address.

However, unfortunately CIDR, together with LPM routing, discourages route aggregation, and induces the announcement of more specific prefixes [3] when multi-homing or traffic engineering requirements come into consideration. De-aggregation is the major factor that contributes to the BGP routing table growth [4]. Moreover, there is an intensive concern that large-scale IPv6 deployment might result in a dramatic increase of the routing table size, which may exceed our ability to engineer the operational routing system [3].Therefore, an obvious question to ask is whether CIDR can continue to be viable to keep global routing state healthy and address space depletion efficiently [4]. To address this issue, several solutions have been proposed by some pioneer research organizations such as IRTF RRG [5] and IETF GROW [6] working groups. These proposals fall into two categories: separation or elimination [3]. Solutions in the separation category separate Internet into transit core and edge network and a mapping system must be inserted between transit core and edge networks [7]. Separation solutions adopt indirect ways to resolve the scalability of routing, which may invite new scalability problem such as the scalability problem of mapping system. Solutions in the elimination category remain existing routing scheme and require supporting the separation of identity and locator.

We perform a preliminary study to analyze the influence of LPM on routing scalability problem. CIDR scheme employs special LPM algorithm to pick the most-specific route from multiple matches. As a result, LPM has made routing lookups more complicated now than they were before the adoption of CIDR when only one exact match search operation was required. Moreover, LPM plays a key role in multihoming, traffic engineer and so forth, which induces routing scalability problem consequently. We first design and implement SPM algorithm to find more general prefixes. Then we make a comparative evaluation between SPM and LPM. The results show that in most cases, average length of AS path of SPM is less than LPM for the same destination prefix of FIB, the percentage of average AS path for SPM less than or equal to LPM is around 60-80% and the percentage for SPM more than LPM is around 10-20%, while the uncertain percentage is less than 10%. Overall, our conclusion is that LPM does not necessarily provide shortest path length for routing lookup in most cases.

The rest of the paper is organized as follows. In section 2, we briefly characterize CIDR and LPM. In section 3, we first present a SPM algorithm, and then compare RIB with FIB depending on the metric of average AS path length (APL). Finally we give a rough and accurate comparison to exactly evaluate LPM and SPM. Section 4 discusses the impact of LPM to scaling routing. Section 5 concludes the paper.

2. CIDR and LPM

CIDR is a method for assigning IP addresses without using the previously defined IP address classes like Class A, Class B or Class C. When originally proposed in [RFC1338] and [RFC1519], CIDR was designed to be a short-term solution to the problems of routing state and address depletion on the IPv4 Internet. It is expected to last approximately three to five years and a more permanent addressing and routing architecture can be designed and implemented. However, CIDR has far outlasted its anticipated lifespan and has become the mid-term solution to the problems described above.

It is well known that each entry in a routing table may specify a network; one destination address may match more than one routing table entry. With CIDR, it was no longer possible to identify the number of bits relevant for the forwarding decision from the address itself, but required a prefix match. LPM becomes part of CIDR and the de facto prefix match strategy. LPM algorithm determines a longest matching prefix for packet routing among all entries, and then many covered prefixes must be kept and leads to excessive size of RIB and FIB. Though CIDR offers a way to summarize routing information, which is one of the keys for routing scalability in today's Internet, unfortunately LPM algorithm doesn't make CIDR become a highly efficient route aggregation mechanism. With LPM algorithm, the benefits of CIDR are counteracted by disincentives to aggregate, leading to the announcement of more specific prefixes in addition to, or instead of, aggregated prefixes [1]. Recent measurements indicate that exponential growth has resumed and there is more requirement multi-homing and load balancing. An obvious question to ask is whether CIDR together with LPM can continue to be viable since it has a negative side effect on routing aggregation. Although efficient aggregation strategies and proper equipment configuration may mitigate the exponential growth, LPM and the pursuit of global route result in the announcement of more specific prefixes. Therefore, it is worthwhile to reevaluate CIDR with LPM mechanism.

As LPM seriously restricts the practice of route aggregation and consequently influences the scalability of routing systems, why is LPM algorithm prevalent? One of the possible reasons is that among multiple routes, the more specific path is considered more reliable. Therefore, LPM search is considered the best path to the destination, which logically makes sense. The other possible reason is that LPM is consistent across implementations and consistent with other routing protocols, such as OSPF. Another possible reason is incremental deployment. It is important that packets using the old address be forwarded correctly during the CIDR with LPM transition period.

3. Experiments and Evaluation

Can LPM algorithm make the path length shorter and better? Since LPM can provide more-specific prefixes, LPM search is considered the best path to the destination, which logically makes sense. In order to quantify the extent, we present a SPM (Shortest Prefix Matching) algorithm to find more general prefixes and compare LPM with SPM depending on the metric of average AS path length. We first implement a Patricia-trie-based RIB and FIB where IP prefixes, their next-hop information and AS path information are stored in certain trie nodes. Then we apply LPM and SPM algorithm to the prefix lookup of RIB and FIB collected from Route Views [8], RIPE [9] from 2000 to 2010. Finally, we compare LPM with SPM by experiments.

3.1. SPM Algorithm

Prefixes in the RIB and FIB can be divided into two kinds, covering prefixes and covered prefixes. Covered prefixes are contained by other prefixes and covering prefixes are in reverse. SPM algorithm performs a lookup to find the entry where the smallest number of leading address bits in the table entry matches those of the destination address. Contrary to LPM, SPM algorithm can find appropriate entry in shortest covering prefixes.

We implement a SPM algorithm by using a Patricia-trie, a tree-like data structure widely used for routing tables. Algorithm 1 shows the lookup function of SPM. Since Patricia-tries are binary trees in which each node represents a binary string that encodes the path from the tree's root to that node, as long as the first matching prefix node is found, SPM lookup is finished.

Algorithm 1: SPM Algorithm

```
-----------------------------------------------------------
  Patricia_SPM_lookup (Node, Prefix){
     Node=Patricia->head;
     While (Node→Prefix_length < Prefix→Prefix_length)
       {
         If (Node->Prefix)
          {
            If (Prefix Є Node->prefix && Node->prefix != "0.0.0.0") return (Node);
       }
      If (Case BIT_TEST(Prefix))
       {
        Node=Node->r;
         }
      Else Node=Node->l;
       }
  }
------------------------------------------------------------
```

3.2. Average APL for RIB and FIB

APL is the number of distinct ASes recorded in the AS path attribute of the BGP routing table entries. To avoid the effect of BGP AS-prepend, we count the AS only once in the AS path length if an AS appears in an AS path several times.

We first apply the metric of APL to the evaluation of RIB and FIB by using LPM algorithm. Figure 1 shows the trend of average APL of RIB and FIB from 2000 to 2010. Average APL of RIB shows a relatively slow increasing trend, varying from 3.6 to 3.8. While average APL of is relatively stable, about 2.6, which means a flattening Internet. This also shows that FIB has shorter APL than RIB, suggesting that with LPM algorithm, FIB choose shorter paths among the RIB for packet forwarding. Figure 2 shows the distribution of average APL for RIB and FIB. We see that the great majority of APL, over 70%, occurs within APL 3 to 4 for RIB, while over 75% APL is between 2 and 3. This means that in most cases LPM can choose slightly shorter AS path, but not necessarily the shortest one. This is mainly because that the shortest AS path routing policy are not typical routing policies used, and AS path inflation is more prevalent than expected in the current Internet. At least 45% of AS pairs choose a longer AS path than the shortest AS path [10]. Therefore, Whether LPM has shorter APL than SPM is not known yet. In the next section, we confirm this by measuring average APL for LPM and SPM.

3.3. Rough Comparison

In this section, we give a rough comparison between LPM and SPM by the metric of average APL. For LPM or SPM evaluation, every IP prefix in the FIB performs a LPM or SPM search to find a suitable IP prefix, and then APL is calculated and counted up. The comparison result between LPM and SPM with the metric of APL can be longer, shorter or equal, denoted by *LPM>SPM*, *LPM<SPM* and *LPM=SPM*. Finally, average APL is calculated and accumulated. In order to carefully evaluate SPM, we introduce Compact SPM (CSPM). For CSPM evaluation, FIB must first eliminate all covered prefixes by SPM algorithm, and then calculate average APL for the rest IP prefixes in the FIB. The results are presented in Figure 4. We can make the following observations: (1) the changing trend of three categories is roughly the same, but the changing extent is different; (2) LPM shows the longest average APL, followed by SPM, while CSPM shows the shortest. (3) CSPM can reduce the FIB size and provide a shorter APL.

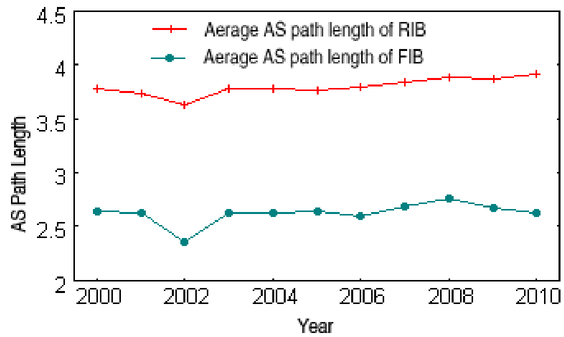

Figure 1. Average APL for RIB and FIB

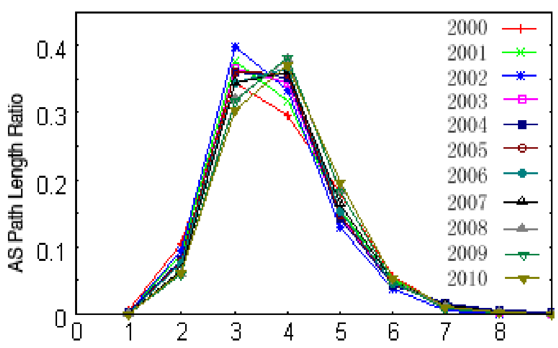

Figure 2. Distribution of average APL for RIB

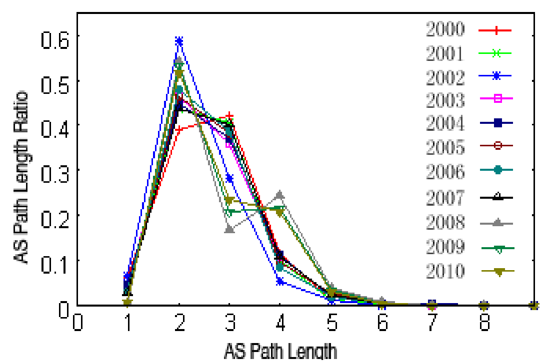

Figure 3. Distribution of average APL for FIB

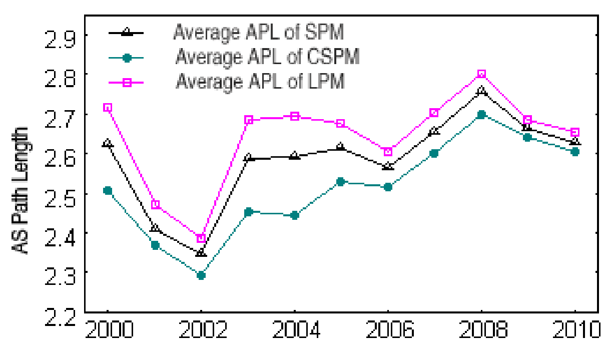

Figure 4. A comparison of LPM, SPM and CSPM by the metric of average APL

It seems that average APL of SPM is superior to LPM from above results. However, for the same IP prefix, AS path may be different for SPM and LPM algorithm. In other words, the conclusion above is not enough to be convincing. Therefore, an exact comparison between LPM and SPM must be based on same source and destination AS. In the next section, we propose a new way to exactly evaluate LPM and SPM.

3.4. Accurate Comparison

For a particular prefix *p*, suppose that path $P_{LPM}(S, D)$ denotes the AS path from source AS *S* to destination AS *D* by using LPM algorithm, and $P_{SPM}(S, D')$ denotes the AS path from source AS S to destination AS *D'* by using SPM algorithm. An exact comparison between LPM and SPM must be based on same source and destination AS, namely, an exact comparison between LPM and SPM should occur between $P_{LPM}(S, D)$ and $P_{SPM}(S, D)$. When the destination AS for LPM and SPM is different, namely $D \neq D'$, the minimal AS path between *D'* and *D*, denoted by *MP(D',D)* should be calculated. If *MP(D',D)* is beyond calculation, called as Iffy *MP(D',D)*, we will give up the comparison between LPM and SPM. Therefore, the comparison result between LPM and SPM with the metric of APL can be longer, shorter, equal or uncertain. To distinguish from above representation for rough comparison, we denote them by *P(L>S)*, *P(L<S)*, *P(L=S)* and *P(L?S)*, and $P(L \geq S)$ can also represent the condition of *P(L>S)* and *P(L=S)*.

According to many factors such as destination AS, the relationship between $P_{LPM}(S, D)$ and $P_{SPM}(S, D')$ can be divided into three categories:

- First category: *D=D'*;
- Second category: $D \neq D'$ and $P_{SPM}(S, D')$ contains the destination AS *D* of $P_{LPM}(S, D)$;
- Third category: $D \neq D'$ and $P_{SPM}(S, D')$ does not contain the destination AS D of $P_{LPM}(S, D)$.

To understand the possible reasons for the AS path changes, we compare the destination AS and first AS for LPM and SPM. This leaves us with more specific cases, as summarized in Table 1.

Table 1. Classification of AS path match between LPM and SPM

Category	Subcategory	APL	AS path example (LPM‖SPM)
First category	Same	*LPM=SPM*	3561 577‖3561 577
	Parallel	*LPM=SPM*	1239 12179 714‖1239 10911 714
		LPM>SPM	1239 701 7843‖1239 7843
		LPM<SPM	3561 11423 6192‖3561 11422 1852 6192
	Converged	*LPM=SPM*	293 701‖2914 701
		LPM>SPM	5409 6427 6461 7018‖7018
		LPM<SPM	3549‖3561 1239 3549
Second category		*LPM=SPM*	2914 6453 4755 ‖ 3561 4755 2697
		LPM>SPM	3333 286 209 1740‖1740 226
		LPM<SPM (1) $P_{LPM}(S, D) \geq P_{SPM}(S, D)$ (2) $P_{LPM}(S, D) < P_{SPM}(S, D)$	1239‖1239 7843 1239‖5056 701 1239 6528
Third category	Merged	*LPM=SPM*	3561 5483 8579‖2914 6453 5483
		LPM>SPM	2914 2685 5676 ‖ 2914 2685
		LPM<SPM	1 1299 6746 9005‖1 8220 12878 5606 6746
	No Merged	*LPM=SPM*	7018 19694‖7018 17311
		LPM>SPM (1) Iffy *MP(D',D)* (2) $P_{LPM}(S,D) \geq P_{SPM}(S,D)+MP(D',D)$ (3) $P_{LPM}(S,D) < P_{SPM}(S,D)+MP(D',D)$	1 701 17169‖2914 8011
		LPM<SPM	3257 4766‖7018 9318 3608 7563

Figure 5 shows an example topology for three categories. First category means that $P_{LPM}(S, D)$ and $P_{SPM}(S, D')$ have same destination AS while second and third categories are on the contrary. Second category means that the destination AS *D* of $P_{LPM}(S, D)$ is one AS hop of $P_{SPM}(S, D')$. Third category means that $P_{SPM}(S, D')$ does not contain the destination AS D of $P_{LPM}(S, D)$, in other words, for prefix *p*, $P_{SPM}(S, D)=P_{SPM}(S, D')+P_{SPM}(D', D)$.

For every IP prefix in the FIB, we first respectively perform a LPM and SPM search, then compare the AS path for $P_{LPM}(S, D)$ and $P_{SPM}(S, D')$, finally count the number of every category. We plot the distribution trend for three categories in Figure 6. The plot shows that first category and third category account for the majority of the total, while second category is in the minority. Moreover, first category shows an overall increasing trend, from 30% by 2000 to 70% by 2010. While third category is on the contrary, suggesting that the FIB has become more amenable to aggregation over the years. One possible explanation is that the increasing practice of prefix splitting due to multi-homing and traffic engineering has made a larger percentage of FIB entries aggregatable.

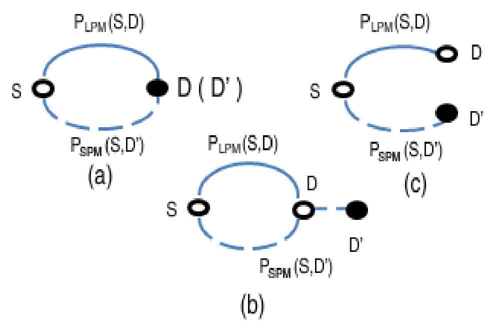

Figure 5. Example topology of three category. (a) First category. (b) Second category. (c) Third category.

Figure 6. Distribution trends for three categories

a) First category evaluation

According to many factors such as first AS and destination AS, first category can be subdivided into three subcategories: same subcategory, parallel subcategory and converged subcategory (see Figure 7). Same subcategory denotes that P_{LPM} *(S, D)* and P_{SPM} *(S, D')* are the same. Parallel subcategory denotes that first AS and destination AS are the same for $P_{LPM}(S, D)$ and $P_{SPM}(S, D')$. Converged subcategory denotes that destination AS is the same while the first AS is different for P_{LPM} *(S, D)* and P_{SPM} *(S, D')*. Figure 8 shows that same and parallel subcategories make principal contributions and both puts up a relation of ebb and flow.

Due to $P_{LPM}(S, D)$ and $P_{SPM}(S, D')$ are the same for same subcategory, APL apparently is equal, while APL for parallel and converged subcategories can be different. The result may be longer, shorter or equal. Figure 9 and 10 respectively show the distribution of three results for parallel and converged subcategories. In Figure 9, for the condition of *LPM=SPM*, we observe a sharp increase in 2008. Later, the improvement of the metric becomes persistent. This also further confirms the problem of no-aggregation. The results also shows that the increase for *LPM<SPM* is marginal.

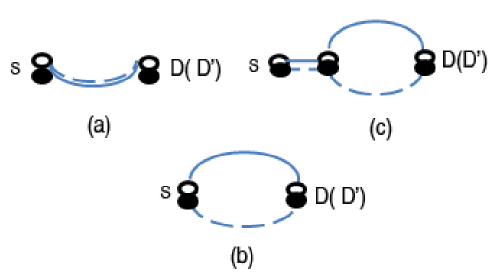

Figure 7. Example topology of first category
(a) Same sub category. (b) Parallel subcategory.
(c) Converged subcategory.

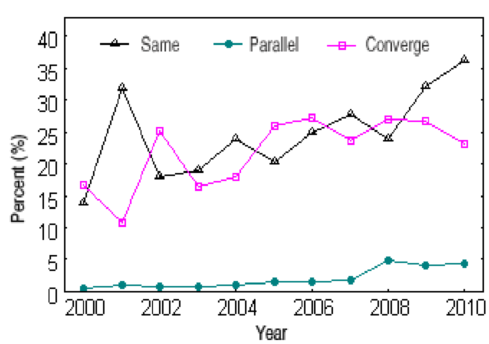

Figure 8. Distribution trends for first category

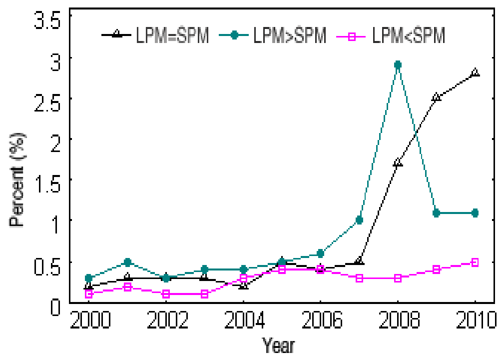

Figure 9. Distribution trends for the parallel subcategory of first category

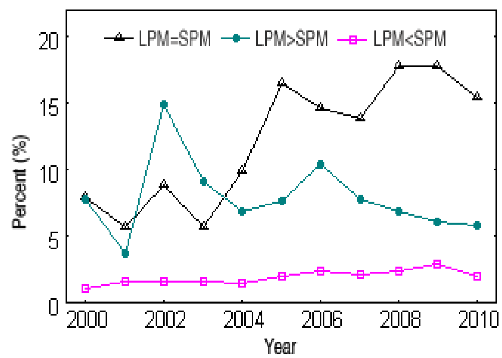

Figure 10. Distribution trends for the converged subcategory of first category

b) Second category evaluation

Second category, an abnormal category, occupies only a small proportion of the whole. Usually more-specific prefix is announced by low-level Internet service provider (ISP) and less-specific prefix is announced by their provider ISP. However, second category is on the contrary. As presented in [11], one possible cause is sub prefix hijacking caused by a configuration mistake or a malicious attack. Another cause could be a configuration mistake of router filter that is supposed to block small address blocks announced by one of its customer ASes.

Due to destination D and D' are different for second subcategory, the comparison between $P_{LPM}(S, D)$ and $P_{SPM}(S, D')$ needs to take into account the location of destination *D* and *D'*. For the condition of *LPM=SPM* or *LPM>SPM*, $P_{LPM}(S, D)$ must be longer than $P_{SPM}(S, D)$. However, for the condition of LPM<SPM, two cases need to be consider, one is $P_{LPM}(S, D) \geq P_{SPM}(S, D)$, the other is $P_{LPM}(S, D) < P_{SPM}(S, D)$. Figure 11 shows the distribution trends for above four conditions. The condition of $P_{LPM}(S, D) \geq P_{SPM}(S, D)$ for LPM<SPM is in the majority while other three conditions play only a small part in the whole thing.

c) Third category evaluation

Third category means that $P_{SPM}(S, D')$ contains the destination AS D of $P_{LPM}(S, D)$ and destination AS is different for LPM and SPM. According to whether destination AS D' for SPM is contained by $P_{LPM}(S, D)$, third category can be subdivided into two subcategories: merged subcategory and no-merged subcategory (see Figure 12). Figure 10 shows the distribution trends for merged subcategory and no merged subcategory. Both have similar percentage and show an overall decreasing trend.

For merged subcategory, there exist three cases: *LPM=SPM*, *LPM>SPM* and *LPM<SPM*. For the case of *LPM=SPM* or *LPM<SPM*, it is obvious that $P_{LPM}(S, D) < P_{SPM}(S, D)$. Theoretically for the case of *LPM>SPM*, the comparison results between $P_{LPM}(S, D)$ and $P_{SPM}(S, D')$ can be longer, shorter, equal or uncertain. However, the test results show that $P_{LPM}(S, D) \geq P_{SPM}(S, D')$ is the only result for the case of *LPM=SPM* or *LPM<SPM*. Figure 14 shows the distribution trends for three cases of merged subcategory. The case of *LPM>SPM* is in the majority while other two cases play only a small part in the whole thing.

For no merged subcategory, the division into *LPM=SPM*, *LPM>SPM* and *LPM<SPM* are first considered. For the case of *LPM=SPM* or *LPM<SPM*, $P_{LPM}(S, D) < P_{SPM}(S, D)$ is obvious. For the case of *LPM>SPM*, *MP(D',D)* must be considered since $P_{SPM}(S, D')$ does not contain the destination AS D of $P_{LPM}(S, D)$. *MP(D',D)* calculation is composed of the following steps: (1) All AS pairs between D' and D are searched and stored in the FIB; (2) If AS pairs between *D'* and *D* exist, *MP(D',D)* can be determined by the minimal APL among all AS pairs between D' and D; If AS pairs between D' and D do not exist, *MP(D',D)* is iffy *MP(D',D)*. There are three possible cases for the condition of *LPM>SPM*. One is uncertain comparison result due to iffy *MP(D',D)*, another is $P_{LPM}(S, D) \geq P_{SPM}(S, D)$, the other is $P_{LPM}(S, D) < P_{SPM}(S, D)$. Figure 15 shows the distribution trends for three cases of no merged subcategory. The case of *LPM=SPM* take the greatest proportion. Following is the case of *LPM>SPM* and *LPM<SPM*. All the conditions show an overall decreasing trend, suggesting that third category keeps the decreasing tendency in the recent years.

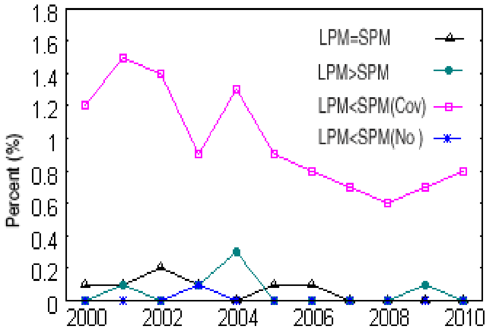

Figure 11. Distribution trends for second category

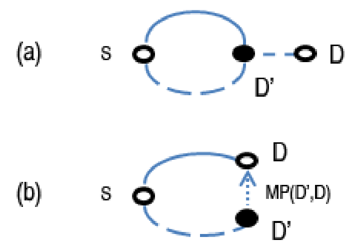

Figure 12. Example topology of third category (a) Merged subcategory. (b) No merged subcategory.

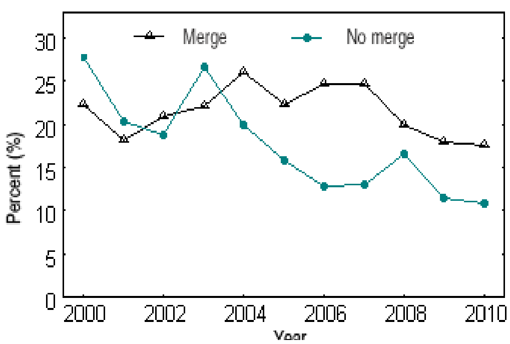

Figure 13. Distribution trends for third category

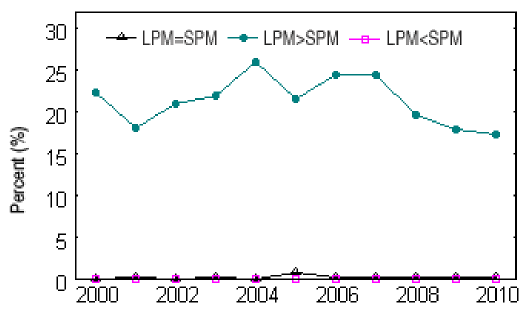

Figure 14. Distribution trends for merged subcategory of third category

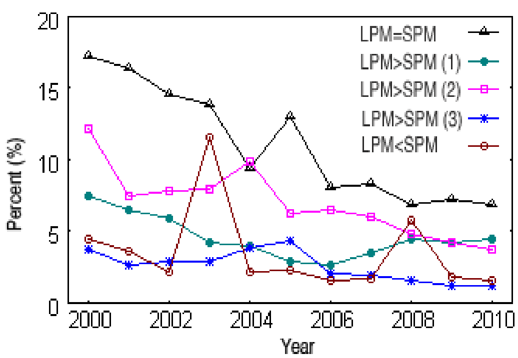

Figure 15. Distribution trends for no-merged subcategory of third category

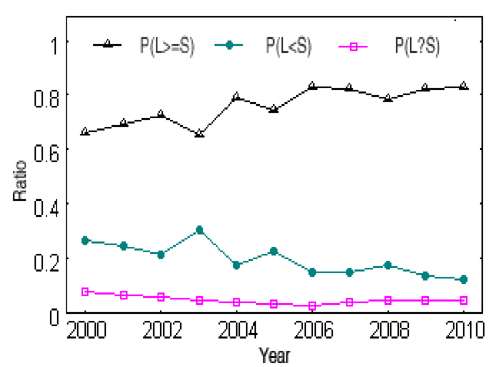

Figure 16. The APL comparison between SPM and LPM

d) Comprehensive Comparison

For above categories and subcategories, we present a joint analysis with the metric of APL. As has been argued, the comparison result between LPM and SPM with the metric of APL can be denoted by *P(L≥S)*, *P(L<S)*, *P(L=S)* and *P(L?S)*. We first calculate the percentage of three conditions respectively for above categories and subcategories, and then calculate the sum. Figure 16 shows the APL comparison result between LPM and SPM. As the APL ratio curve shows, the condition of *P(L>=S)* is in the majority, from 60% to over 80% and keeps an overall increasing trend, meaning that in most cases SPM is superior to LPM by the metric of average APL. While the condition of *P(L<S)* accounts for below 30% and shows an overall decreasing trend, meaning that in most cases LPM is inferior to SPM by the metric of average APL. The condition of *P(L?S)* has the smallest proportion of total, below 10%.

4. Discussion

The experiment results unexpectedly show that in most cases, average length of AS path of SPM is less than LPM for the same destination prefix of FIB. This means that LPM can't provide the best path for specific prefixes. Therefore, it is worthwhile to reevaluate CIDR with LPM mechanism.

LPM seriously restricts the practice of route aggregation and consequently influences the scalability of routing systems. Internet routing is the process of moving packets through an internetwork and an ideal routing path is to find the shortest path. However, the shortest path must require the information of the whole network topology, which is conflicted with the scalability of routing. Scalable routing, on the contrary, tries to find relatively short path based on partial topology information. Route aggregation is a feasible way for scalable routing, which can provide an indirect routing path by aggregating a large span of addresses into a less specific prefix. Packets must first be moved to the aggregation point, and then a recursive routing lookup is performed to find a more precise route in numerous more-specific prefixes. However, improving the aggregation quality is hampered by the employment of LPM. Customer pursues more specific prefixes to meet the requirements of multihoming or traffic engineering because of the LPM strategy in their packet forwarding implementation. Therefore providers are obliged to propagate each full length site prefixes into the DFZ, which is the major reason for routing scalability problems in BGP.

On the other hand, LPM makes the forwarding mechanism complex and difficult since prefixes can be overlapped, while traditional lookup algorithms (such as hash lookup, binary search) practically cannot be applied. Especially IPv6 addressing scheme are about to be adopted, the address length is expanded from 32 bits to 128 bits and the size of RIB/FIB can be extremely large. Routing/forwarding with LPM will face rigorous challenges.

Finally, there exit conflicting routing strategies between different ASs. Service providers apply 'hot potato' routing policy and customers may prefer the policy of 'avoid being dumped on' model. Service providers frequently have conflicting operational objectives for handling traffic [12]. Service providers adopt 'hot potato' routing to minimize internal costs, therefore, they look to hand off traffic as quickly as possible, while customers look to avoid being dumped on as long as possible. The main reason is that customer is attempting to influence incoming routes by multi-homed site prefix as far as possible. At the same time, the holders of the packets on the other side of the DFZ would prefer to filter any long prefixes so they can simply dump packets as quickly as possible [12]. As a result, the 'dump early' strategy wants to hear short prefix lengths, while the 'avoid being dumped on' model drives wide distribution of the longest possible prefix.

To avoid direct influence of multihoming and traffic engineer, LPM with the same next-hop can effectively take into account both routing scalability and functional requirement. In fact, FIB aggregation [13] and Real aggregation [14] adopt the way of LPM with the same next-hop.

5. Conclusion and Future work

In this paper, we analyze LPM in CIDR environment that seriously hinders the development of scalable routing. The prevalent LPM algorithm for routing lookup is considered the best path to the destination, which logically makes sense. However, we find that LPM algorithm does not necessarily provide shortest path length for routing lookup in most cases with comparison experiment. To find more general prefixes, we first present SPM algorithm, and then make a comparative evaluation between SPM and LPM by using the metric of average AS path length.

In our ongoing work, we plan to study CIDR based on optimized LPM algorithm. Our long-term goal is to propose a long-term solution based on optimized LPM.

6. Acknowledgements

The authors are grateful to former peer's devoting in this area. This research was under the support of National Basic Research Program of China (Grant No. 2009 CB3 20502), National Key Technology R&D Program of China (Grant No. 2008BAH37B03). The authors also would like to thank all the anonymous reviewers for their hard works.

7. References

[1] D. MEYER, L. ZHANG, K. FAL. Report from the IAB workshop on routing and addressing. IETF RFC4984, 2007.
[2] V. Fuller, T. Li. Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan. IETF RFC4632, 2006.
[3] L. Wang, D. Jen, M. Meisel, et al. Towards A New Internet Routing Architecture: Arguments for Separating Edges from Transit Core. In Proc. of HotNets-VII, pp. 13-18, 2008.
[4] T. Bu, L. Gao, D. Towsley. On characterizing BGP routing table growth. In Proc. of IEEE Global, pp. 363-364, 2002.
[5] IRTF Routing Research Group. http://www.irtf.org/ charter?gtype=rg\&group=rrg.
[6] IETF Global Routing Operations (GROW). http://www.ietf.org/dyn /wg/charter/growcharter.html.
[7] D. Jen, M. Meisel, D. Massey, et al. APT: A Practical Tunneling Architecture for Routing Scalability. Technical Report 080004, UCLA, 2008.
[8] Route Views Project Page. http://www.routeviews.org.
[9] RIPE Route Information Service. http://www.ripe.net/ris/index.html.
[10] L. Gao, F. Wang. The extent of AS path inflation by routing policies. In Proc. of the GLOBECOMM'02, pp. 2180-2184, 2002.
[11] Y. Zhu, J. Rexford, S. Sen, et al. Impact of Prefix-Match Changes on IP Reachability. In Proc. of Internet Measurement Conference, pp. 235-241, 2009.
[12] T. Hain. An IPv6 Provider-Independent Global Unicast Address Format. IETF draft draft-hain-ipv6-pi-addr-use-10.txt, 2006.
[13] X. Zhao, Y. Liu, L. Wang, et al. On the Aggregatability of Router Forwarding Tables. IEEE INFOCOM, pp. 1-9, 2010.
[14] W. Zhang, X. Yin, J. Wu, et al. Real Aggregation for Reducing Routing Information Base Size. Journal of Convergence Information Technology (JCIT), vol.5, pp. 47-53, 2010.