Caching Popular BGP Prefixes with Grey Modeling Prediction

Wei Zhang[1,2] Jun Bi[1,3,4] Jianping Wu[1,3,4] Baobao Zhang[1,3]

1 Department of Computer Science, Tsinghua University, Beijing, China

2 Wuhan commanding communications Institution, Wuhan, China

3 Network Research Center, Tsinghua University, Beijing, China

4 Tsinghua National Laboratory for Information Science and Technology (TNList)

<zw,zbb>@netarchlab.tsinghua.edu.cn, <junbi,jianping>@tsinghua.edu.cn

***Abstract*—Internet core routers are facing challenges brought by the ever-increasing transit bandwidth and routing scale. In order to meet the requirements of highly efficient packet forwarding, some solutions propose to load a small portion of the BGP RIB entries into the FIB. Therefore the most popular prefixes, which contribute major traffic loads, need to be cached in the FIB as long as possible. In this paper, we try to propose a prediction based method to cache those popular prefixes in the FIB. The cache strategy is guided by the traffic prediction of a grey model. We also apply FIB aggregation techniques to suppress the number of overlapped sub-prefixes of the popular prefixes on cache/route updates. We evaluate our method with real traffic traces and find that our prediction-based cache replacement strategy outperforms other cache strategies and matches Internet traffic dynamics very well.**

Keywords:BGP, Popular Prefix, Route caching, grey mode

I. INTRODUCTION

Internet core routers are facing challenges brought by the ever-increasing transit bandwidth and routing scale. The Autonomous System border routers (ASBRs) may have to process packets on each port at a rate of 40 Gbps. This requirement can only be met by applying high-performance hardware, such as TCAM or SRAM. However, this kind of special-purpose memory is costly and energy eager while normally its size can not be very large. On the other hand, the size of BGP Route Information Base (RIB) increases inexorably. If we load every entry of the route table into the Forwarding Information Base (FIB), a router is compellingly under a great pressure of FIB inflation.

Recently there are some new proposals to address this routing scalability problem. Although there might be different solutions to tackle this problem, the most straightforward method is to load a small portion of RIB entries into the FIB. This method can be modeled as a route caching mechanism in a dynamic routing system.

Motivated by these considerations, we propose to take advantage of the features of the real traffics, and dynamically cache the popular prefixes based on an effective traffic prediction method.

The rest of the paper proceeds as follows. In section II, we review related works. In section III, we discuss route caching mechanism and relevant cache strategies. In section IV, we analyze the feature of Internet traffic and introduce the grey model prediction we use in our method. In section V, we propose our prediction based cache strategy in detail as well as the solution of prefix overlapping problem. In section VI, we evaluate our method via simulations on real traffics and explain our findings. In the end, we draw a conclusion.

II. RELATED WORKS

ViAggre [1] statically sets a few popular prefixes measured on the routers of an ISP/Point of Presence (PoP). Their experiments show that loading a relatively static popular prefix set can improve the performance undoubtedly. However they assume the set of popular prefixes is very stable and can be set statically.

Dynamically updating route cache on routers is not new though [2, 3]. Changhoon Kim et al. [4] argue that dynamic route caching is still viable because the Internet traffic exhibits high degrees of temporal locality .However, they measured these features only based on a uniform class prefix (i.e. with a fixed-length of </24). In their experiment, the routes are non-aggregatable and there are approximately 500k~1M routes to be stored in the cache for an acceptable route miss rate.

Another related work in [5] addresses ID/Locator mapping mechanism designed in ID/Locator split schemes, and discusses the cost of route caching on the granularity of BGP prefixes. Their research focuses on the efficiency of a pull-model-based ID/Locator mapping mechanism. In [5], the cache policy involves a timeout threshold for all cached entries. Their experiment shows the cache size scales with the traffic loads, which means the required cache has no boundary.

This paper try to highlight the features of working traffic on dynamics of AS level traffic loads over real BGP prefixes and apply them to facilitate pragmatic route caching mechanisms.

III. ROUTE CACHING MECHANISM AND STRATEGIES

The route caching mechanism can be abstracted as the following conceptual model: to store a small fraction of the routes in a size-limited but efficient cache, and a full route table will be stored in a relatively slow but bigger memory and can be accessed with some overheads. A cache “miss” surely introduces considerable delay or other costs. The objective to optimize the performance of the cache is to minimize the cache miss rate.

In this model one great challenge is that the popularity of a prefix is dynamically determined by the interfering of two changing elements: the RIB entries and the working traffic loads on it. We have already known the Internet traffic exhibits locality properties (temporal and spatial) and the packet traffic is non-uniformly distributed [7,8]. We need to device proper cache replacement strategies to match the fundamental dynamics of the working traffic over CIDR prefixes.

We will discuss some relevant cache replacement strategies applied in the route caching mechanism before we devise new cache strategy by taking advantage of the features of working traffics.

Static strategy

This is the most straightforward strategy and it depends on the assumption that most of the popular prefixes are stable. The cached routes will not be updated for a long period of time. Normally the cached popular prefixes are chosen by hand (based on personal experience or preference). Any replacement requires administrators' involvement therefore the administrative cost can be very high.

Cache-miss-based Dynamic Strategies

For this kind of dynamic strategy, the cache replacement can happen at any time, only triggered by a cache miss. There are different algorithms. Basically Least Recently Used (LRU) and Least Frequently Used (LFU) are typical strategies. Another strategy is proposed in [5]. The TD maintains a timer of not being hit for every prefixes in the cache. Once the timer gets over a threshold, the prefix should be dumped. All above cache-miss-based algorithms are conducted on the packet level, and may require replacement when a cache miss happens. The overheads of this kind of strategies can be very high when numerous bursty traffics exist.

Optimal (Opt)

The Opt strategy is the theoretical limit as it requires future knowledge. The Opt algorithm dumps prefixes that will be least used in the future therefore it has the least cache miss rate. Note that the Opt can not be implemented in practice and is be used as a theoretical comparison.

All these replacement strategies tend to catch popular prefixes in a route caching mechanism. Except the static strategy, the others are adaptive to the traffic dynamics. Their efficiencies depend on how good they may fit the property of the traffic load distribution across prefixes. We need to have a general analysis on these features of the Internet traffic before we can justifiably propose our purposive contribution.

IV. INTERNET TRAFFIC PREDICTION

A. *Internet traffic features*

Many previous network measurement studies have found that the Internet traffic has a feature of power law distribution [9, 10, 11, 12] that a few prefixes contribute most traffic loads and the rest prefixes in the RIB have only trivial or non traffic in a certain time span. This power-law observation has held up in the past decades and hopefully will hold up in the future. This feature makes the route caching mechanism viable and beneficial. The power-law distribution of traffic loads over different prefixes means too many prefixes only contribute trivial traffic loads. There comes our basic understanding that to give up those trivial traffics is a good strategy on controlling the size of route cache. In this paper we focus on the popular prefixes which contribute most traffic loads.

More complicated is the dynamics of the popular prefixes. Our observation and other experimental research [7] show the popularity of a prefix only has short term stability. The traffic on a prefix can be very bursty, and a popular prefix can not have heavy traffic loads for all the time. Generally speaking the popularity of a prefix is most likely instable when observed on short term time scales. This empirical understanding is very helpful on designing a practical cache replacement method.

B. *Grey Model for traffic prediction*

On predicting the traffic over different prefixes, we apply a grey model [18], which is typically used in grey system control and prediction. This model assumes that although the elements that influence the traffic loads cannot be depicted with explicit functions, their general effect has been illustrated in the traffic traces. The traffic load is influenced by a "grey" system which we may by and large estimate its overall behavior in the next stage if we can effectively retrieve enough information from the traffic traces.

The mathematical analysis of the grey model can be found in [18]. Here we only deliver the GM(1,1) modeling process in our specific prediction application.

On applying GM(1,1) model, we should calculate the traffic loads of each prefix within every time of intervals. Then we have a series of traffic loads for each prefix in the RIB as $X^0(t)$, where t=0,1,2…; after an accumulation generation operation(AGO), we have $X^1(t)$, the cumulative series of $X^0(t)$, and formally $X^1(t)=\sum_{i=0}^{t} X^0(i)$, t=0,1,2,…

The GM(1,1) model mathematically assumes the cumulative series is continuous and abides with the following difference equation:

$$\frac{dX^1(t)}{dt}+aX^1(t)=u \qquad (1)$$

After Laplace transform and inverse Laplace transform, we have the solution of the differential equation:

$$X^1(t+1)=(X^0(1)-\frac{u}{a})e^{-at}+\frac{u}{a} \qquad (2)$$

In equation (1) and (2), a and u are parameters which can be estimated through least square method.

$$\hat{A}=(\hat{a},\hat{u})^T=(B^TB)^{-1}B^TY \qquad (3)$$

where

$$B=\begin{bmatrix} -0.5(x^1(2)+x^1(1)) & 1 \\ -0.5(x^1(3)+x^1(2)) & 1 \\ \dots & \dots \\ -0.5(x^1(t)+x^1(t-1)) & 1 \end{bmatrix} \quad Y=\begin{bmatrix} x^0(2) \\ x^0(3) \\ \dots \\ -x^0(t) \end{bmatrix}$$

After that, we substitute $\hat{a}$ and $\hat{u}$ into equation (2), and then inverse the AGO operation. The prediction of the traffic load will be

$$\hat{X}^0(t+1) = \hat{X}^1(t+1) - \hat{X}^1(t).$$

This model treats the accumulative traffic load as the single variable that its current variance is only subjected to its previous amount. This model establishes a first order differential equation shown in (1). The GM(1,1) model, namely a single variable $X^1(t)$ and its first order differential equation, has been widely used in grey prediction applications.

Technically, if $X^1(t)$ is high, we hope its variance accordingly low, which means a stable popular prefix ($X^1(t)$ has a high volume with low variance) matches the model better than otherwise. In our algorithm, we only accept the prediction with $a \in (-2,2)$. This method can extract useful information from a grey system efficiently.

The grey model has the following virtues. Firstly, the accumulative series depicts a general trend of the traffic with certain degree of confidence which is essential to our traffic prediction. Secondly, it is adaptive to burstness of the traffic loads over prefixes. If we estimate the popularity of a prefix by simply calculating an averaged traffic load over a period of time, the traffic burst of a non-popular prefix may influence our judgment especially when the traffic trace is insufficiently informative. Thirdly, the irrelevant prefixes can be differentiated easily (if the parameter *a* falls into the forbidden area). Lastly, the grey model requires not too much input information compared with other models. It is very suitable to the situation of sketchy forecasting.

V. Prediction Based Route Caching Strategy

Based on the GM(1,1) traffic prediction, our route caching strategy will dump non-popular routes out and load potential popular prefixes periodically.

In our scheme, we need a few sampled traffic traces to guide our prediction. Instead of triggering a cache replacement by a cache "miss", we can estimate the popular prefixes of the next stage beforehand. The cache loading/dumping is solely based on this "prediction" and takes place with a certain intervals. The size of the cache and the intervals of the cache replacement may vary according to the hardware capacity. In our strategy, at first we need to set a replacement interval (e.g. 5 minutes). A shorter interval may have higher cache efficiency but also introduces more computational overheads.

In order to reduce the overheads of frequent cache/route replacement, we suggest using two independent memories for forwarding lookups. And the forwarding lookups will be carried on in turns with these two memories. When one memory is used as a working cache, the other is used for loading the caching routes on the next stage. This mechanism guarantees there will be almost no delay of cache replacement which is extremely important for high speed routers.

In the case of route updates, the FIB loading can be triggered immediately without waiting for the next due time of a cache replacement. There is a copy of the current route cache in the FIB calculating device therefore any route update that involves the current cached routes can be recalculated immediately and update the cache without too much delay. After loading the updated cache, the standby memory can be activated as the working cache which means a cache memory switch has been triggered by route updates.

Another issue that needs to be taken into consideration is the problem of loading overlapped prefixes. In order to guarantee correct forwarding with longest prefix match algorithm, when we load a less specific prefix as popular prefix, all its more specific covered prefixes (sub-prefixes) have to be loaded as well. Otherwise there will be a routing problem. These overlapped prefixes make the cache size bigger than the popular prefix set that we estimate. Basically the size of the popular prefix set is proportional to the final cache size. Sometimes a prefix may have as many as more than 2000 sub-prefixes. In this case we adapt a FIB suppressing algorithm, Optimal Route Table Constructor (ORTC) [19], to reduce the number of sub-prefixes and consequently to reduce the ratio of the cache size and the accommodated popular prefix set. The gain of ORTC depends on the number of forwarding next hops and the forwarding redundancy of routes. Our simulation shows that ORTC can effectively reduce the number of sub-prefixes for all popular prefixes. The only limitation of ORTC is the problem of route update. Because the next hop of a prefix may be different with the corresponding entry in the RIB and even new prefixes can be generated, each route update requires recalculation of the entire tree. The incremental route update is not possible in this case. However, in our scheme the recalculation will be limited to relevant popular prefixes. Moreover a standby memory is always ready for route updates and popular prefix swaps. Generally the overheads of ORTC recalculation and route updates are acceptable.

The overall implementation of our scheme can be illustrated in a conceptual router model as shown in Figure 1.

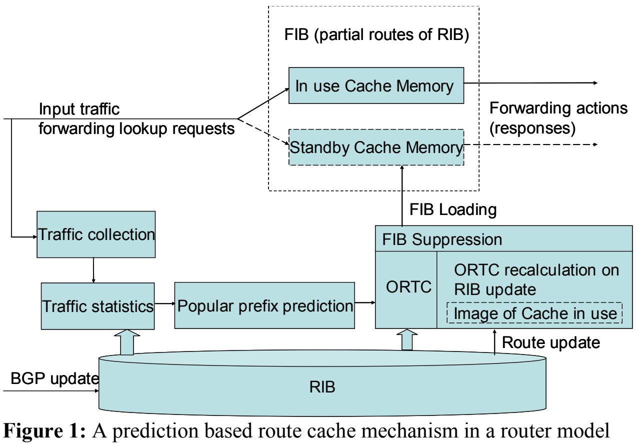

Figure 1: A prediction based route cache mechanism in a router model

We evaluated our method with the working traffic traces collected from the China Telecom backbone networks and our campus networks gateway accessed to the CERNET.

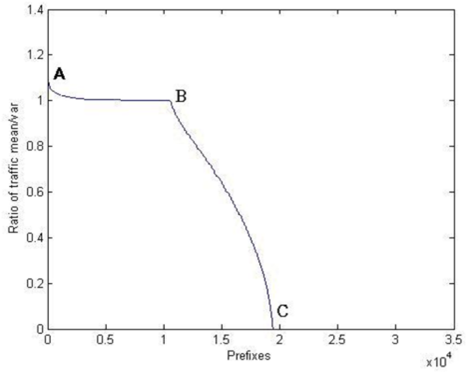

Figure 2: Traffic dynamics over prefixes

VI. EVALUATION

A. Methodology

We collected traffic trace and corresponding route tables provided by CHINANET [14], the backbone of China Telecom. We used selected traffic traces from two routers in Shanghai and Guangzhou respectively to generate per-prefix traffic statistics. We collected sampled flow-level Netflow traces in a period of two months (from 14th Nov 2009 to 16th Jan 2010) at different backbone routers located on two PoPs of CHINANET. In CHINANET, China Telecom IP network services are delivered over the international 40Gbps MPLS-enabled IP backbone. Since the volume of the traffic transited by those routers was extremely large, the Netflow sampling rate was set as 1:5000 with the sampling techniques specified in [13]. We observed that a small fraction of Internet prefixes carried a large majority of ISP traffic. This observation was also found in previous studies [1,12,15,16,17]. Our measurements confirmed this property which is the prerequisite of route caching.

Then we measured the dynamics of popular prefixes by means of variance analysis. We found the dynamics of popular prefix can be commonly featured at all AS border routers.

At last we evaluated different cache replacement strategies over working traffic traces (from 14th to 16th of Jan 2010). A reasonable range of reference history against different sizes of caches had also been measured. Basically our interests focused on the tradeoff among cache size, route miss rate and corresponding overheads.

B. Observations

According to our observation of the traffic distribution across BGP prefixes on 14th JAN 2010, of all the 336390 prefixes we calculate the number of packets hitting on each prefixes. Our observation is that 143950 prefixes (account for 42.8% of RIB) had no traffic at all that day. We find that 95% traffic is contributed by no more than 5% BGP prefixes. But the popularity of each individual prefix is not stable in a large time scale.

In order to illustrate the dynamics of the variable popularity, we calculate the traffic on a smaller time scale (at an interval of 5 minutes). That is to say we count the packets on each prefixes for every 5 minutes in a day, and consequently we have a discrete series x representing the traffic loads at 288 discrete time slots of a day for each prefix. In this N*288 matrix (N is the number of RIB entries), we find the traffic loads on a given prefix at successive time slots may change drastically from time to time which is the burstness observed as a common understanding.

On observing the stability of a prefix, we calculate the mean value and variance of x. Then we give a generalized metric to measure the popularity and stability of a prefix simultaneously: the ratio R between averaged traffic loads and variance of x, as shown in formula (4).

$$R = \begin{cases} Mean(x)/Var(x) & \text{if } mean(x) > 0 \quad Var(x) \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

(4)

In which,

$$Mean(x) = \sum_{i=1}^{288} x_i \Big/ 288; \qquad Var(x) = \sum_{i=1}^{288} (x_i - Mean(x))^2$$

Obviously, a stable popular prefix will have substantially high traffic loads and relatively low variance, and its value of R generally should be higher than otherwise. We plot the R value of each prefixes in figure 2, and find that a small portion (2~3% of RIB size) of prefixes can be both “popular” and relatively stable (from point A to B) all day long. These prefixes are rather stable and contribute major traffic loads and should be kept in the cache as long as possible. Other relatively instable popular prefixes (from point B to C) have changing popularities but still contribute considerable traffic loads. As the variance increasing, the value of R will decrease accordingly. We see in Figure 2 from point B to C a declining track which implies the increasing instability of such popular prefixes or a declining of contribution to traffic loads. These prefixes can only be caught effectively by dynamic cache strategies. Our prediction based method can differentiate them from normal traffic bursts. Beyond point C are non-popular prefixes. Remember among 336390 prefixes being measured, almost half of them have extremely trivial traffic for all 288 time slots in a day. The above observation illustrates the fact that in BGP RIB, we have stable popular prefixes, instable popular prefixes and non popular prefixes. Assuming the popularity of any prefix is stable and only adapt static strategy could be too ideal and far from optimal.

On analyzing the dynamic strategies of route caching mechanism, we first apply optimal strategy as a background to show the theoretical limits at different cache size and replacement frequencies. On plotting the limits of different replacement frequencies and cache sizes in figure 3 we can get some basic understandings of the Internet traffics. Firstly, at a given cache miss rate and replacement interval, we can estimate the least cache size which is the vital parameter in our method. Secondly, as we can see in Figure 3, there is a diminishing marginal utility of caching more prefixes: soon after the size of the cache exceeded 5%of the RIB size (about 18000 entries), the miss rate decreases slowly with the increasing size. After loading 10% of the RIB (33000

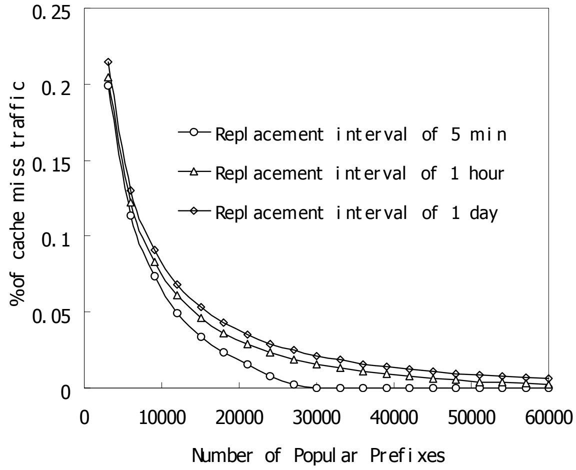

Figure 3: Comparison of different cache replacement intervals.

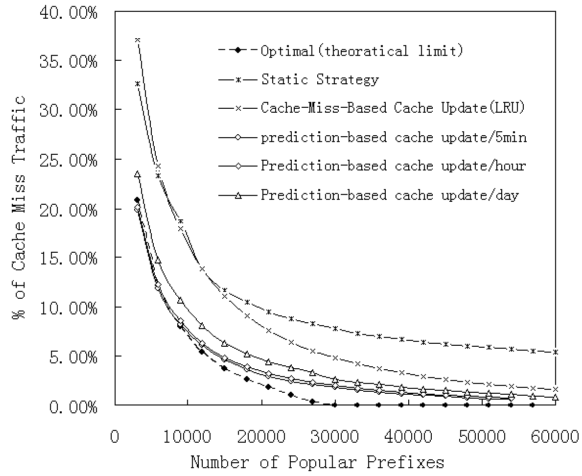

Figure 4: Comparison of different strategies.

prefixes) as popular prefixes, the miss rate decreases rather slow and almost has no obvious gains if the replacement takes place frequently. This fact implies that numerous prefixes contribute trivial traffic loads and suggests an efficient cache size will be around the turning points. Based on this observation we can tune a better tradeoff between the cache size, cache miss rate and cache update intervals (overheads).

C. Performance Evaluation

On evaluating the performance of our method, we simulated different strategies as a comparison. We plot in Figure 4 the performances of static cache strategy, cache-miss-based strategy (LRU) and prediction-based strategy respectively. Static cache has the highest miss rate. The popular prefixes set we used in static method was measured on a given day with the optimal strategy at the same router. In static strategy, we assume the most popular prefixes that day (account for approximately 2% of RIB) are in the cache and will not be replaced for a long time, but the other prefixes in the cache are randomly chosen if the cache size is more than 2% of RIB. To some extent, the static strategy can be viewed as the extreme situation where we set the replacement interval as infinity. In this way, we have long-term stable popular prefixes in the cache, but they are not necessarily stable on smaller time scales. Since there are only very few popular prefixes have short-term stability, most short-term instable popular prefixes can only be caught effectively by dynamic cache strategies. However, when the caches size is extremely small, it outperforms LRU. In most cases, the cache size is not so constraint, and static method gains less than the other dynamic cache strategies.

Cache-miss-based strategies (such as LRU/LFU) can catch the instable popular prefixes, but some bursty traffic will influence the route caching efficiency from time to time and might incur much higher overheads when replace cache items on each cache miss. Since we do not have a packet-level trace, we simulate LRU strategy based on our flow-level Netflow traces.

Our prediction based method will replace the cached prefixes periodically with determined intervals. It can catch most of the popular prefixes solely guided by traffic traces. It is almost immune to the traffic bursts if we set a proper range of reference traffic traces. Figure 5 illustrates the ideal range of reference is about 50 minutes when we set the cache size as 3% of the RIB.

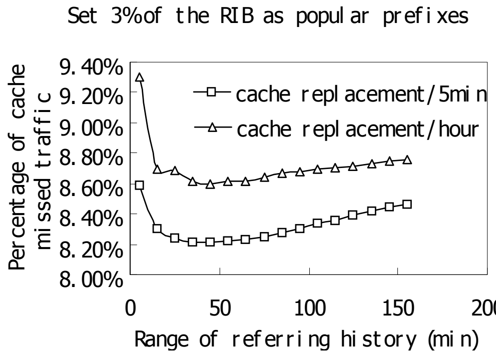

Figure 5: Reasonable range of reference for a valid prediction

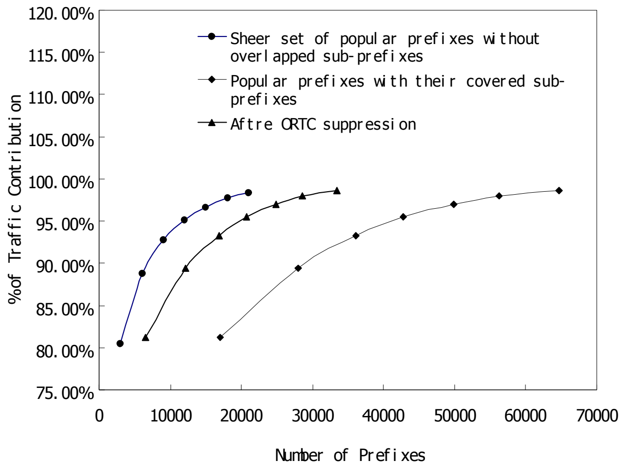

Figure 6: Comparison of caching popular prefixes with and without their covered non-popular sub-prefixes, as well as the FIB suppression effects of ORTC.

The last important measurement of our study is the overlapped sub-prefixes of the popular prefixes. As plotted in Figure 6, without ORTC, in the worst cases, the cache size can be almost 5 times as big as the sheer popular prefix set, which means we have to load many non-popular sub-prefixes into the cache. With ORTC, the number of sub-prefixes reduces considerably and the overall cache size is about a double size of the sheer popular prefix set which is affordable when the popular prefix set is only 2~5% of the RIB. Hopefully we can reduce the requirement of the FIB size an order of magnitude with no more than 5% cache miss rate.

On evaluating the overheads of our method, the traffic trace collection requires about 10MB in a relatively slow memory, the prediction process will take no more than 30 seconds and FIB suppression algorithm will take no more than 40 seconds. The standby cache memory will be ready on time for cache replacement. The route update only involves the updated popular routes. An ORTC recalculation of the updated popular prefix is optional but it surely will have no significant delay because the other cached routes will not change.

With the experiments on the working Internet traffics, our method can dynamically cache 5~10% of BGP RIB with 2 independent caches, the cache miss rate is less than 5%; the ideal cache replacement interval is 5 minutes and a less frequent replacement strategy may introduce corresponding higher miss rate(there always exists a tradeoff between performance and overhead); the reasonable range of reference traffic trace is no more than one hour; the sampling statistic of the traffic loads can be optional and the sampling ratio can be adjusted according to the general traffic load on the router. The GM(1,1) prediction is not very much sensitive to the sampling ratio.

VII. CONCLUSION

Our study gives an in depth research on the features of the Internet traffic distribution across BGP prefixes. The popularity of a prefix may be instable. On the other hand the BGP prefixes are heavily overlapped with each other. These two facts introduce two major challenges on the route caching mechanism: how to catch the instable popular prefixes efficiently and ameliorate the side effect of loading their non-popular sub-prefixes effectively.

Our method addresses these two problems by apply a grey model prediction and the FIB aggression technique. The GM(1,1) model can retrieve information efficiently from limited traffic traces and adaptive to the bursts over non-popular prefixes very well. The ORTC algorithm can suppress the number of sub-prefixes effectively when loading a popular prefix into the cache.

On empirically analyzing the dynamics of popular prefixes in the working traffics, we evaluate our prediction based route caching strategy that matches the Internet traffic at AS border routers better than other cache strategies. The simulations have shown that the general overheads of our method are acceptable and the performance is close to the ideal limits.

ACKNOWLEDGMENT

This work was supported by National Science Foundation of China under Grant 61073172, Program for New Century Excellent Talents in University, Specialized Research Fund for the Doctoral Program of Higher Education of China under Grant 200800030034 and 20090002110026, and National Basic Research Program ("973" Program) of China under Grant 2009CB320501.

REFERENCES

[1] H. Ballani, P. Francis, T. Cao, J. Wang, "Making Routers Last Longer with ViAggre", In: *Proceedings of USENIX NSDI*, 2009.

[2] How to Choose the Best Router Switching Path for Your Network. Cisco Systems (August 2005), <http://www.cisco.com/warp/public/105/20.pdf>

[3] C. Partridge, P. Carvey, "A 50-Gb/s IP router, IEEE/ACM" Transactions on Networking 6 (3) (1998) 237-248.

[4] C. Kim, M. Caesar, A. Gerber, and J. Rexford, "Revisiting Route Caching: The World Should Be Flat", in: Proceedings of PAM, 2009.

[5] L. Iannone, O. Bonaventure, "On the Cost of Caching Locator/ID Mappings", in: proceedings of CoNEXT, 2007.

[6] S. Brim, D. Farinacci, V. Fuller, D. Lewis, and Meyer, "LISP-CONS: A Content distribution Overlay Network Service for LISP", Internet Draft, < http://tools.ietf.org/id/draft-meyer-lisp-cons-00.txt >, 2007.

[7] C. Park ,F.H. Campos, J.S. Marron, F.D. Smith, "Long-range dependence in a changing Internet traffic mix", Computer Networks 48 (3) (2005) 401−422.

[8] C. Williamson, "Internet Traffic Measurement", IEEE Internet Computing 5 (6) (2001) 70-74.

[9] M. Arlitt, C. Williamson, Internet Web Servers: "Workload Characterization and Performance Implications", IEEE/ACM Transactions on Networking, 5 (5) (1997) 815-826.

[10] L. Breslau, P. Cao, L. Fan, G. Phillips, S. Shenker, "Web Caching and Zipf-like Distributions: Evidence and Implications", in: Proceedings of IEEE INFOCOM, 1999.

[11] M. Crovella, A. Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes", IEEE/ACM Transactions on Networking, 5 (6) (1997) 835-846.

[12] W. FANG, L. PETERSON, "Inter-As traffic patterns and their implications". in: Proceedings of Global Internet, 1999.

[13] Sampled NetFlow, Cisco Systems.

[14] CHINANET, <http://en.chinatelecom.com.cn/products/t20060116_48406.html>

[15] A. FELDMANN, A. GREENBERG, C. LUND, N. REINGOLD, J. REXFORD, F. TRUE, "Deriving traffic demands for operational IP networks: methodology and experience", IEEE/ACM Transactions on Networks, 9 (3) (2001) 265-279 .

[16] N. TAFT, S. BHATTACHARYYA, J. JETCHEVA, C. DIOT, Understanding traffic dynamics at a backbone PoP, in: Proceedings of Scalability and Traffic Control and IP Networks SPIE ITCOM, 2001.

[17] J. REXFORD, J. WANG, Z. XIAO, Y. ZHANG, "BGP routing stability of popular destinations", in: Proceedings of Internet Measurement Workshop, 2002.

[18] J. Deng, "Control problems of grey systems", systems & control letters, 1 (5) (1982) 288-294.

[19] R.P. Draves, C. King, S. Venkatachary, B.N. Zill, "Constructing Optimal IP routing tables", in: Proceedings of IEEE INFOCOM, 1999.