Measures of Self-Similarity of BGP Updates and Implications for Securing BGP

Geoff Huston

Centre for Advanced Internet Architectures,
Swinburne University of Technology,
Melbourne, Australia
gih@swin.edu.au

Abstract. Techniques for authenticating BGP protocol objects entail the inspection of additional information in the form of authentication credentials that can be used to validate the contents of the BGP update message. The additional task of validation of these credentials when processing BGP messages will entail significant additional processing overheads. If the BGP validation process is prepared to assume that a validation outcome has a reasonable lifetime before requiring re-validation, then a local cache of BGP validation outcomes may provide significant leverage in reducing the additional processing overhead. The question then is whether we can quantify the extent to which caching of BGP updates and the associated validation outcome can reduce the validation processing load. The approach used to address this question is to analyze a set of BGP update message logs collected from a regional transit routing location within the public IPv4 Internet. This paper describes the outcomes of this study into the self-similarity of BGP updates and relates these self-similarity metrics to the size and retention time characteristics of an effective BGP update cache. This data is then related to the message validation activity, and the extent to which caching can reduce this validation processing activity is derived.

Keywords: BGP, Secure BGP, Validation Caching

1 Introduction

The scaling properties of BGP [1] have represented a long-term concern for the viability of BGP in the role of supporting the inter-domain routing system of the public Internet. These concerns were first raised in the early 1990's [2] and continue to the present time. The elements of this concerns are twofold: firstly, the 'size' of the routing space, as expressed by the number of discrete entries to be found in a default-free BGP routing table, and, secondly, the rate of BGP Updates messages to be processed, which relates to the 'processing load' of the routing system. A study of the characteristics of BGP update messages over the entire year of 2005 indicated that the number of discrete entries in the BGP routing table grew by 18% to a total at the end of the year of some 175,400 entries, the number of per-prefix updates per eBGP peer session grew by 49% over the same period, and the number of BGP prefix withdrawals grew by 112% [3]. The salient observation here is that the number of BGP updates, or the 'processing load' associated with BGP in the Internet is growing at a higher rate

Paper to be presented at Passive and Active Measurement Workshop, April 2007   http://pam2007.intel-research.net/

than the number of BGP entries, or the ‘size’ of the BGP routing table. The implication here is that ‘processor load’ appears to represent the more critical scaling factor for BGP in the context of the growth trends within the public Internet.

Adding additional attributes to the BGP protocol that are intended to improve the security of BGP also have the potential to impose higher processing loads per BGP update message. The significant factor in incremental processing loads is that associated with the task of validating the contents of the BGP update message. The basic security questions with respect to securing the BGP update payload include validation of the address prefix, validation of the origin AS, validation that the origin AS has the authority to originate an advertisement of this prefix, and validating the AS Path attribute of the update as being an accurate representation of the advertised path to reach this address prefix [4].

Irrespective of the manner by which the authentication credentials associated with a BGP update are propagated across the routing realm, the security consideration is that the BGP receiver should validate these credential as a precursor to accepting the BGP update as authentic. Approaches to validation of such information commonly rely on public key cryptography, where the original ‘author’ of the information can attach a signature to the information using their private key, and any recipient of the information can validate the authenticity of the information by validating the signature block through using the matching public key. Such a validation process implies some form of additional processing load.

One approach to minimize the additional processing overhead associated with validation of BGP updates is to use validation caching. In this approach the validation outcome of the BGP update is cached for a period of time, and if the update is repeated within this period, then the previous validation outcome is used without further validation checking.

The questions that this approach raise include: How effective could validation caching be in this context? How big a cache is appropriate in terms of size and hit-rate trade-offs? What time-period of validation caching would be appropriate in terms of a balance between cache hit rates and validation accuracy?

The remainder of this paper is structured as follows. Section 2 provides a description of the measurement and analysis methodology used in this study, and the data set used by this study. Section 3 describes the outcomes of this study, relating self-similarity results to per-update validation processing loads. Section 4 concludes the paper.

2 Methodology

The Zebra implementation of BGP [5] was configured as a BGP update message collector, and this collector was configured with a single eBGP session to AS4637. The motivation for this simple BGP configuration was to isolate the update message sequence that correspond to a single eBGP peering session, and for the update message sequence to reflect the changes that occur in the Local Routing Information Base (LOC-RIB) of the peer AS.

The configuration was set up to collect the complete set of BGP update messages, place a timestamp on each update and save then on a rolling 24 hour basis. Also, a snapshot was taken of the BGP routing table at the end of each 24 hour cycle. This collection process commenced on the 3rd March 2006, and the data set, continuously updated on a daily basis, has been published as a data resource.

For this study, the data corresponding to a two week period from midnight 10th September 2006 to 23:59 23rd September 2006[1] [2] was used. The update log data from the BGP collection unit was translated to a time-sequence of per-prefix update transactions, reflecting the sequence and time of changes to the LOC-RIB of the BGP speaker in the peer AS.

This sequence of BGP transactions was used as input into a multi-dimension cache simulator. This simulator simultaneously simulates a set of fix-length caches ranging in size from 1 through to the number of unique updates, and reports on the cache hit rate for each possible size of the cache.

3 Measurement Results

In the 14 day analysis period there were a total of 656,339 BGP Update messages. The profile of the BGP state across this 14 day period is shown in Table 1.

Table 1. BGP Profile for the period 10-September 2006 00:00 to 23-September-2006 23:59

Metric	Value
Number of BGP Update Messages	656,339
Prefix Updates	1,632,900
Prefix Withdrawals	223,616
Average BGP Update Messages per second	0.54
Average Prefix Updates per second	1.53
Peak Prefix Update Rate per second	4999
Prefix Count	202,769
Updated Prefix Count	111,769
Stable Prefix Count	91,000
Origin AS Count	23,233
Updated Origin AS Count	15,501
Stable Origin AS Count	7,732
Unique AS Path Count	87,238
Updated Path Count	75,529
Stable AS Paths	11,709

The distribution of updates is not uniform across the set of prefixes, nor is it uniform across the set of origin ASs. The most unstable 1% of prefixes generated 18% of the total number of BGP prefix updates, and the top 50 prefixes (0.025% of the prefixes in the BGP routing table) generated 3.5% of all prefix updates. A similar skewed distribution is evident for autonomous systems, where the busiest 1% of the Origin ASs (232 ASs) were affected by 38% of the total updates, and the top 50 Origin ASs (0.2% of the ASs) were involved in 24% of all the BGP prefix updates.

In considering the potential to cache validation outcomes of BGP updates, it is noted that the validation of a BGP Update has a time component, in that the validation outcome does not remain useable indefinitely. The potential effectiveness of validation caching depends on both the time characteristics of self-similar BGP updates (the elapsed time between identically keyed updates), as well as their space characteristics (the number of different updates between identically keyed updates).

[1] The times quoted here refer to the local time at UTC+10 hours.

[2] The BGP data set is published at http://www.potaroo.net/papers/phd/pam-2007/data

In this study four types of self-similarity are considered, namely the recurrence of updates that specify the same address prefix, secondly, updates that share the same address prefix and origin AS, thirdly, the same address prefix and AS Path, and finally updates that share the same address prefix and the same 'normalized' AS Path (in this context 'normalization' of an AS Path implies the removal of duplicate AS numbers from the AS Path that are the result of path pre-pending).

The occurrence of self-similar updates across the 14 day period is indicated in Table 2 ("Norm-Path" in this table refers to equivalence using the 'Normalized' AS Path). The daily totals are plotted in Figure 1.

Table 2. BGP Self-Similarity Total Counts per day

Day	Prefix	Duplicates:	Duplicates:	Duplicates	Duplicates
	Updates	Prefix	Prefix +	Prefix +	Prefix + Norm-
			Origin AS	AS Path	Path
1	72,934	60,105 (82%)	54,924 (75%)	34,822 (48%)	35,312 (48%)
2	79,361	71,714 (90%)	67,942 (86%)	49,290 (62%)	50,974 (64%)
3	104,764	93,708 (89%)	87,835 (84%)	65,510 (63%)	66,789 (64%)
4	107,576	94,127 (87%)	87,275 (81%)	64,335 (60%)	66,487 (62%)
5	139,483	110,994 (80%)	99,171 (71%)	68,096 (49%)	69,886 (50%)
6	100,444	92,944 (92%)	88,765 (88%)	70,759 (70%)	72,108 (72%)
7	75,519	71,935 (95%)	69,383 (92%)	56,743 (75%)	58,212 (77%)
8	64,010	60,642 (95%)	57,767 (90%)	49,151 (77%)	49,807 (78%)
9	94,944	89,777 (95%)	86,517 (91%)	71,118 (75%)	72,087 (76%)
10	81,576	78,245 (96%)	75,529 (93%)	63,607 (78%)	64,696 (79%)
11	95,062	91,144 (96%)	87,486 (92%)	72,678 (76%)	74,226 (78%)
12	108,987	103,463 (95%)	99,662 (91%)	80,720 (74%)	82,290 (76%)
13	91,732	87,998 (96%)	85,030 (93%)	72,660 (79%)	74,116 (81%)
14	78,407	76,174 (97%)	74,035 (94%)	64,994 (83%)	65,509 (84%)

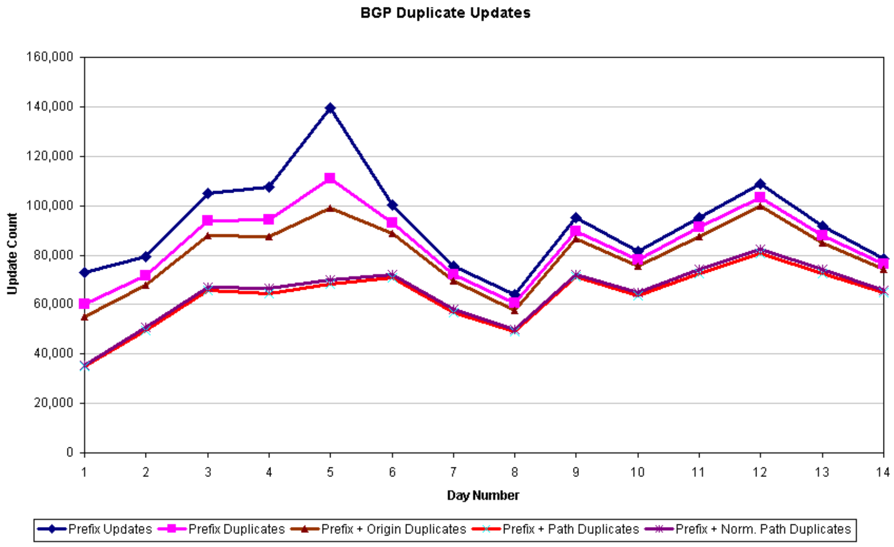

Fig. 1. Daily Duplicate Update Profile

3.1 Time Distribution of BGP Updates

The first question concerns the time spread of self-similar BGP updates and concerns the distribution of time intervals between pairs of similar BGP updates, using the four types of prefix update similarity noted in the previous section. While the time resolution of the collected update log data is in units of milliseconds, the eBGP session used for data collection uses a 30 second value for the min-route-advertisement timer, so that all updates are passed to the collection unit in 30 second intervals. Accordingly, in this examination of the time spread of self-similar BGP updates, the time intervals are aggregated into 30 second increments.

A cumulative histogram of the proportion of recurring update messages by varying recurrence intervals of these four types of update self-similarity are shown in Figure 2. The actual number of duplicate updates are shown in Figure 3. Some 49% of all prefix-recurring updates occur within 90 seconds of the previous update. Some of these high frequency updates could be due to BGP convergence behaviour following a prefix withdrawal, where BGP will explore a number of alternative routes before the withdrawal has been propagated across the entire network. When looking at recurrence of prefix + path only 8% of all similar updates occur within the same 90 second interval. As BGP withdrawal-triggered convergence should not explore the same route twice, the relative difference the two figures indicate that up to 40% of the recurring BGP updates that refer to the same prefix may be attributable to short term updates generated during the process of BGP convergence.

Some 80% of recurring prefix updates occur within 1 hour of the preceding prefix update. The same 80% threshold occurs within 35 hours when considering the recurrence of BGP updates that contain the same prefix and AS Path. There is little difference between Path and Compressed Path similarity measures in either short (1 hour) or longer (36 hour) timeframes.

If a benchmark value for potential cache efficiency is set to 80%, and the validation cache process is intended to cache the outcome of both prefix origination and AS Path validity, then a validation retention time period of 36 hours would be a minimum value to achieve this performance metric.

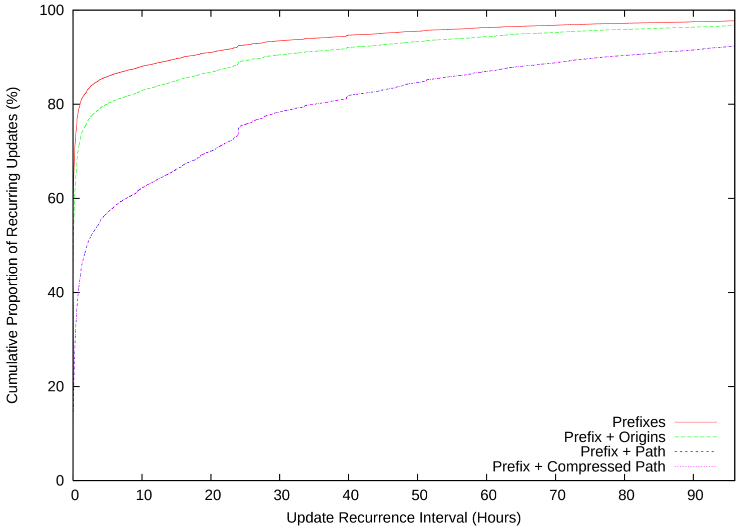

Fig. 2. Cumulative Proportion of Recurring Updates for Address Prefixes

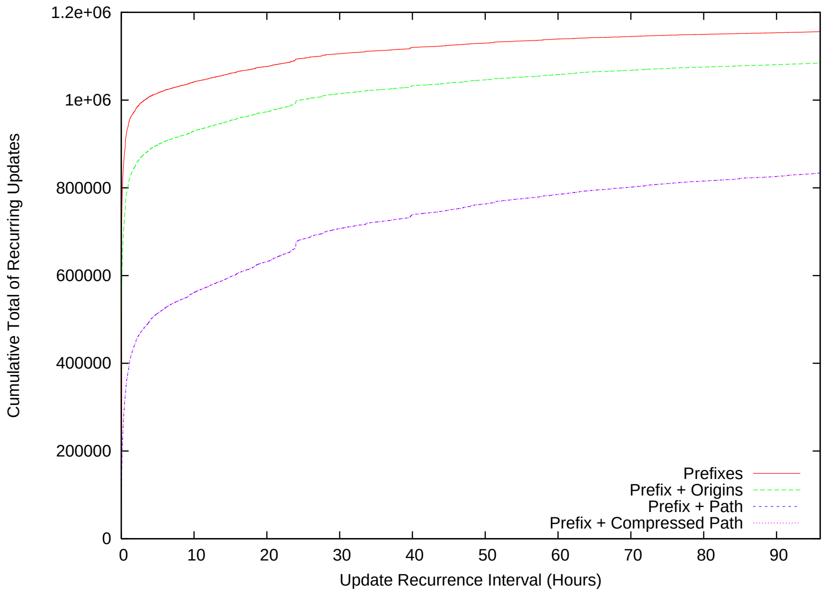

Fig. 3. Cumulative Volume of Recurring Updates for Address Prefixes

3.2 Validation Cache Simulations

The next metric concerns the spatial dispersion of similar BGP updates, where the number of intervening unique updates between two matching updates is considered. This dispersion relates to the size of a cache that would be able to generate a cache hit for the matching updates when using a simple Least Recently Used (LRU) cache management regime.

Figure 4 shows the cache hit rate per day for each of the 14 days in the study period, using a simple LRU cache scheme. This data indicates that a cache of 200 prefixes would provide an average hit rate of between 50 to 70%. Improvement in caching efficiency per incremental unit of cache size appears to drop once the cache size exceeds 1000. The outlier 24 hour period sequence correlates to a BGP reset of a transit peering session which, in turn, generated a large number of one-off prefix updates, which caused the lower than average cache hit rate for this 24 hour period.

If authentication data were applied to prefixes, and a BGP receiver validated this prefix data, then it is possible to look at the potential improvements that a prefix validation cache could provide. This is shown in Figure 5. It is noted that this data set uses a validation lifetime of 36 hours, such that a cached entry will be considered stale and re-validation required once the entry is more than 36 hours old. Within each 24 hour period some 15% to 20% of announced prefixes are announced more than 36 hours after their previous announcement, while some 80% of announced prefixes have been previously announced less than 36 hours previously. The overall majority of announced prefixes in BGP are short lived announcements that are refined by subsequent updates within the ensuing 36 hours. This data indicates that a per-eBGP peer validation cache of 1000 prefixes, managed on an LRU basis with a 36 hour validation period would provide a reduction in validation processing of BGP updates by between 60% to 80% on a daily basis. Increasing this cache to 10,000 entries could improve this validation processing load reduction to between 80% to 90%.

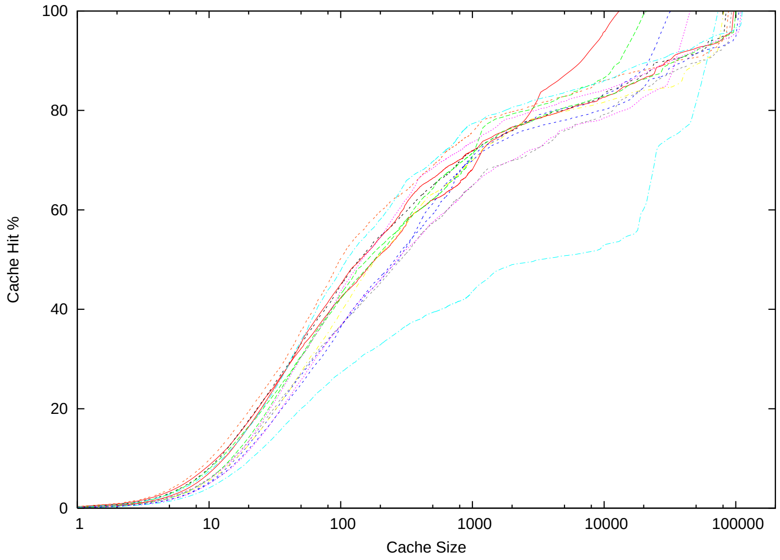

Fig. 4. Cache Hit rates per Day for a range of Cache Sizes. The cache algorithm is LRU, using a lookup key of the prefix

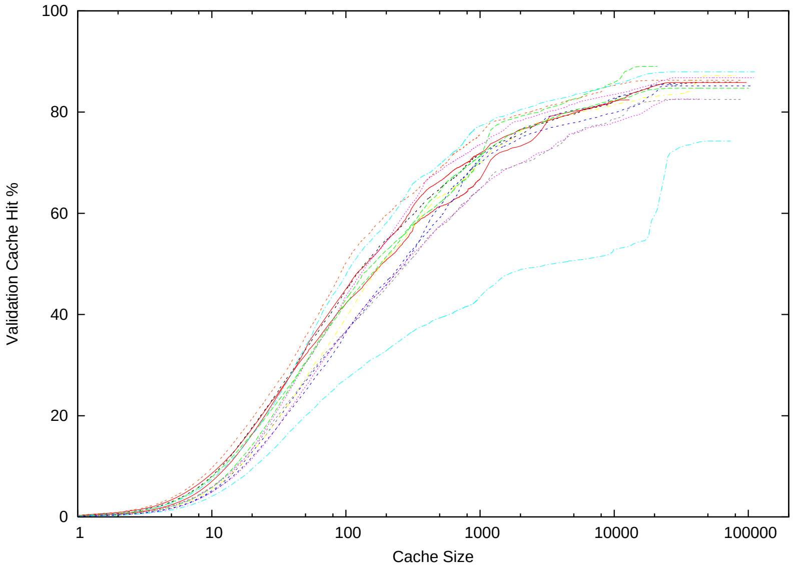

Fig. 5. Validation Cache Hit rates per Day for a range of Cache Sizes. The cache algorithm is LRU, using a lookup key of the prefix, with a validation re-use period of 36 hours

Validation of BGP updates can apply to more than just the validity of the advertised address prefix. Using a signed attestation, where the address holder explicitly permits an Autonomous System to originate a BGP advertisement, the combination of the address prefix and originating AS can be validated. The extent to which caching the validation outcome of the combination of address prefix and origin AS in indicated in Figure 6. The number of prefixes that show short term instability in their origin AS is very low, so the validation cache outcomes when using a key of prefix plus origin AS are very similar. A cache of 1000 prefix + Origin AS validation entries can reduce the validation processing load by an average of 60% within a 24 hour period, while a cache size of 10,000 entries offers an average hit rate of 75%. This cache size, 10,000 entries, is some 5% of the total number of distinct entries in the BGP routing table as of September 2006.

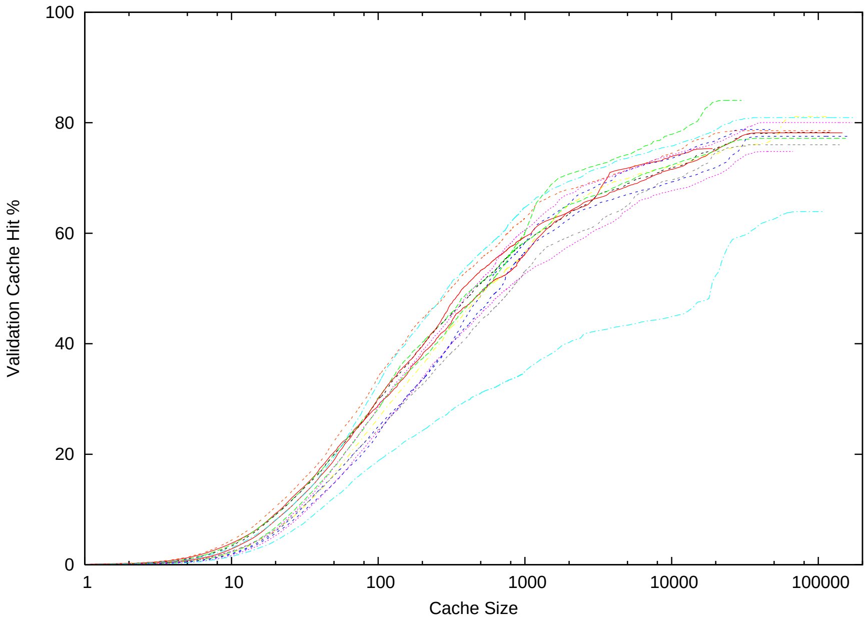

Fig. 6. Prefix + Origin AS Validation Cache Hit rates per Day for a range of Cache Sizes. The cache algorithm is LRU, using a lookup key of the prefix and the Origin AS, with a validation re-use period of 36 hours

The next step is to include consideration of the AS Path into the cache key. In this case the routing assertion that is being authenticated when validating the AS Path is that the BGP update was processed by each of the ASs in the AS Path in sequence, and that the AS Path has not been altered in any way. To replicate this load the validation cache key used the compressed AS path, where instances of AS prepending with duplicate AS number values in the path were removed The validation cache rate is indicated in Figure 7. In this case the validation cache is not as effective, and a cache of 10,000 entries will reduce the total validation load by some 30% to 50%. A potential explanation of this difference lies in the nature of the protocol operation of BGP. Withdrawal of a route may not propagate at a uniform rate through the network, and partial withdrawal information may generate transient routing states during the convergence period. These transient routing states share a common origin and AS number, but differ in the AS Path. The sequence of transient routing path states may be highly variable, which in turn causes a low level of cacheability of the path-based information.

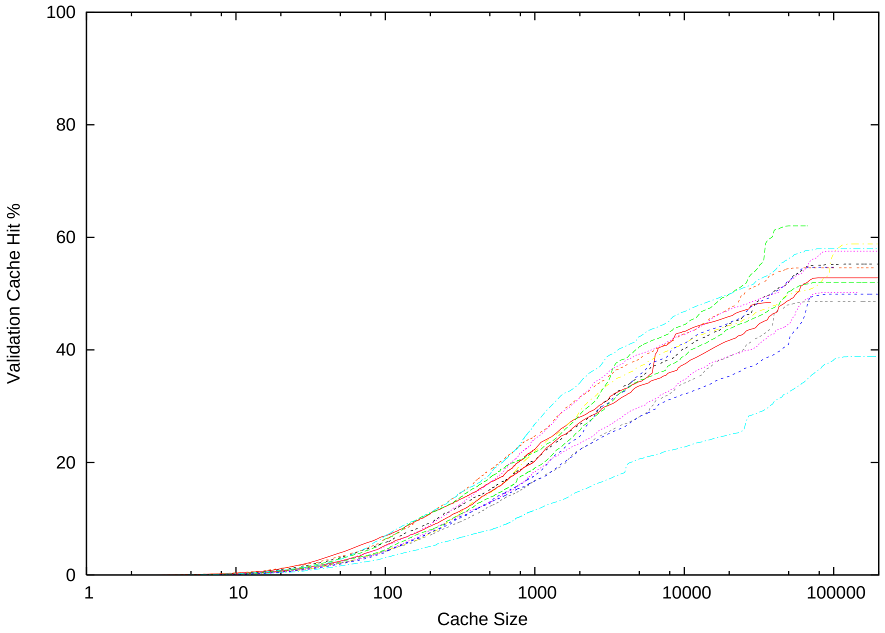

Fig. 7. Prefix + Path AS Validation Cache Hit rates per Day for a range of Cache Sizes. The cache algorithm is LRU, using a lookup key of the prefix and the Compressed AS Path, with a validation re-use period of 36 hours

This data suggests that caching can offer the highest processing load reduction when the cache reflects the validation of the prefix and the origin AS, and that a cache size of between 1,000 to 10,000 entries can offer a reduction in processing overheads of between 60% to 80%.

4 Conclusion

BGP updates exhibit a distribution structure where a relatively small number of prefixes are the subject of a significant number of similar BGP updates, and these self-similar updates appear to be strongly clustered in both time and space.

A significant concern when examining proposals to modify BGP processing to include validating the security credential material supplied with BGP updates is the incremental processing load that may be imposed on BGP speakers.

If a BGP speaker is willing to cache a validation outcome for a period of up to 36 hours, and retain these outcomes using a cache size of 10,000 entries (or some 5% of the total number of distinct BGP table entries) with a LRU replacement cache management algorithm, then the cache is capable of performing at an average 75% hit rate for prefix origination. Similar parameters for prefix plus path validation indicate that a similar set of cache parameters, namely a cache size of 10,000 entries and 36 hours cache hold time, is capable of sustaining an average of a 30% to 50% cache hit rate.

This analysis indicates that the incremental load imposed by adding validation of credentials associated with BGP updates can be significantly mitigated by using caching of validation outcomes for subsequent reuse.

This work reflects a study of the BGP update traffic for a single peer. Future work on a study on self-similarity and cache parameters for multiple eBGP sessions, decoupled origination and path validation caching, and also the effects of delayed validation on the validation work load are logical extensions of this study.

References

1. Y. Rekhter, T.Li, S. Hares: A Border Gateway Protocol 4 (BGP-4), RFC 4271, Internet Engineering Task Force, January 2006.
2. Internet Architecture Board: Minutes of Meeting, January 1991 (online at http://www.iab.org/documents/iabmins/IABmins.1991-01-08.arch.html).
3. G. Huston: 2005 – BGP Updates, presentation to Global Routing Operations Working Group, IETF 65, March 2006 (online at http://www3.ietf.org/proceedings/06mar/slides/grow-3.pdf).
4. S. Kent, C. Lynn, and K. Seo: Secure border gateway protocol (s-bgp), IEEE Journal on Selected Areas in Communication, vol. 18, no. 4, 2000.
5. GNU Zebra. (online at http://www.zebra.org)