On Update Rate-Limiting in BGP

Ahmed Elmokashfi
Simula Reserach Laboratory

Amund Kvalbein
Simula Reserach Laboratory

Tarik Cicic
University of Oslo

***Abstract*—In order to reduce the number of BGP updates that routers need to process, it is common to rate-limit such updates using a timer that specifies the minimum time between two consecutive updates for a given destination prefix. Rate-limiting plays an important role in determining the number of routing updates that are generated after a routing event, and the time it takes before the network converges to a new stable state. Still, there are few guidelines for how rate-limiting timers should be configured in order to achieve the desired convergence properties. This work takes a first step in this direction, by exploring how different rate-limiting implementations and configurations affect the resulting churn level in a live BGP session. Measurements are performed on multiple parallel BGP sessions to a stub AS, configured with and without rate-limiting timers. We find that the daily rate of updates is reduced by two thirds when configuring the timer to the default value recommended by BGP standards. We further investigate different rate-limiting implementations and configurations using the measured BGP update patterns on emulated BGP sessions, and find that increasing the rate-limiting timer gives a logarithmic decrease in churn. Finally, using BGP update traces from RouteViews, we present the first empirical model that quantifies the impact of rate-limiting in terms of churn reduction given the observed arrival pattern of BGP updates.**

I. Introduction

The Border Gateway Protocol (BGP) [14] is the protocol used for inter-domain routing in the Internet. BGP is a path-vector protocol; routers announce to their neighbors the AS-level path they can provide to reach a given destination prefix. When a BGP router learns about a new preferred path to a destination, it informs its neighbors using BGP UPDATE messages. Each update causes a processing load in the receiving router. The new path must be recorded, and the BGP decision process must be run to decide whether the preferred path should be updated. If so, the new preferred path must be installed in the forwarding table, and new updates must be sent to BGP neighbors whenever the used routing policies allow.

To limit the rate of updates that a router must process, referred to as the churn rate or simply churn, it is common to perform some type of rate-limiting in BGP sessions. When an event affects the best route to a destination prefix, this will often trigger a sequence of update messages before the network stabilizes on the new preferred route. By delaying the transmission of an update message for a configured amount of time, that message will often be invalidated by the subsequent update for the same destination prefix. This way, it is often possible to mask out intermediate states, and thus reduce the number of updates sent over the BGP session. A larger configured delay gives a stronger reduction in churn, but also increases the time used to converge to the new steady state.

Earlier work by Griffin and Premore [5] analyzed the impact of different rate-limiting settings on churn and convergence time by simulating single prefix announcements and withdrawals in small generic topologies. The main insight from their work is that it is possible to find a timer setting that minimizes convergence time while keeping the number of updates low, but that these settings vary depending on the size and structure of the topology. In addition, the convergence process will depend on the nature and location of the underlying routing events; routing changes that result in a complete withdrawal of a network prefix involve more path exploration than those ends with an alternative path [12]. Hence, it is difficult to find timer settings that work well in the Internet based on these results. To give practical guidelines for the use of rate-limiting timers, the update arrival pattern that is observed in the Internet must be characterized and taken into account. Analyzing the impact of different rate-limiting implementations and timer settings on real measured BGP sessions is the main goal of this paper.

Our starting point is the time series of BGP updates produced by border routers in a well-connected stub AS. This data is a product of the Internet topology (as seen from the monitored AS) and the mix of routing events that takes place during our measurement period. Starting from this data, we investigate the different ways in which major router vendors implement BGP rate-limiting. We explain how the two main approaches (MRAI timers and OutDelay) have very different effects on the churn rate and convergence time. Based on measurements on parallel monitoring sessions with and without rate-limiting, we quantify the churn reduction achieved with a given configuration setup. Further, we use the measured data to emulate different rate-limiting implementations and timer values and quantify the corresponding churn levels.

By looking at update traces from a large number of route monitors in the RouteViews project [2], we discover that the arrival pattern of BGP updates for single prefixes is remarkably stable across BGP sessions from a diverse set of ASes across the Internet. This allows us to derive a formulation that quantifies the expected reduction in churn for different rate-limiting implementations and configured timer values.

Our findings present a general framework for helping network operators that want to use rate-limiting in deciding which implementation to choose. Furthermore, it helps in finding the right balance between churn reduction and increased convergence times when setting the timer value. These findings can be particularly interesting for the networks wishing to receive full Internet routing information while using edge routers with limited processing power.

The rest of the paper is organized as follows. Section II discusses different rate-limiting implementations. Section III describes our measurement setup and takes a first look at the data. Section IV looks at churn reduction with different rate-limiting implementations and configurations. Section V presents a model that predicts churn reduction for different rate-limiting implementations. Section VI briefly discusses related work, before Sec. VII sums up our main findings.

II. RATE-LIMITING IMPLEMENTATIONS

To limit the rate of BGP updates, the BGP standard [14] recommends the use of a MinRouteAdvertisementInterval-Timer (MRAI timer) which specifies the minimum time interval between two consecutive updates for a destination prefix. The recommended value of this timer on eBGP sessions is 30 seconds. To avoid peaks in the number of sent updates, the standard recommends jittering the MRAI timer by multiplying its value with a random number between 0.75 an 1.

Some implementations (notably Cisco IOS and the Quagga software router), implement per-session timers rather than per-prefix timers in order to reduce overhead. When the MRAI timer is enabled on a certain session, the router queues updates for all prefixes and sends them out in a burst when the timer expires[1]. We refer to this rate-limiting implementation simply as *MRAI timers* in the sequel.

Another common BGP implementation (Juniper's JunOS) implements rate-limiting using the *out delay* parameter. Unlike the MRAI timer implementation described above, this delay is added to each update for each prefix individually. When a router changes its best path to a destination prefix, it will not inform its peer about the change unless the route has been present in its routing table for the specified out delay. The out delay parameter value is 0 seconds by default in Juniper routers (i.e., no rate-limiting). We refer to this rate-limiting implementation as *OutDelay* in the sequel.

The different implementation choices give different reductions in the number of updates and different increase in convergence time. Fig. 1 illustrates a sequence of updates for a prefix p arriving at a router. First, let us assume the router uses an MRAI timer with a value of 30 seconds, and that the timer expires at $t = 0, 30, 60$. Then, the router sends out an update informing about the change caused by U_1 at $t = 30$. The router queues an update that reflects the change caused by U_2 to be sent out at time $t = 60$. However, the queued update is invalidated by the arrival of U_3, and only a single update sent at $t = 60$. In this example, two updates are sent, and the convergence process on takes 55 seconds.

Second, let us consider a session with an OutDelay of 30 seconds. When the router receives U_1, it schedules an update to be sent at time $t = 35$. However, when receiving U_2 at $t = 31$, the scheduled update transmission is cancelled. Instead, a new update that reports the change caused by U_2 is scheduled to be sent at $t = 61$, which in turn is cancelled after receiving U_3. The receiving of U_3 finally results in scheduling a new update that is sent at $t = 71$. In this case, the router sends out only one update and converges in 66 seconds. This example shows that the OutDelay implementation gives a stronger reduction in churn than MRAI timers, but that it also converges slower, since updates are always delayed by a full timer interval. In this study, we use a combination of measurements and emulations to compare the two different implementations, and to evaluate the churn reduction achieved with different settings.

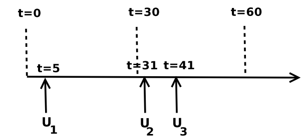

Fig. 1: Example of a convergence sequence

III. MEASUREMENT SETUP AND DATA

Our measurement setup consists of two collectors, each connected to three different routers (referred to as monitors) in a stub AS. The collectors that receive BGP updates from the monitors are implemented on computers that run the Quagga routing suite [1] using private AS numbers. The monitored stub AS is well connected to the Internet through multiple transit providers and direct peering, and has a geographical presence in several cities in both North America and Europe. The collectors use multi-hop eBGP sessions to peer with the monitors. A monitor sends a BGP update to the collector every time there is a change in the preferred path from the monitor to a destination prefix. In addition, we dump a snapshot of the routing table of each monitor every two hours.

The three monitors belong to the *Default Free Zone* (DFZ), meaning that their routing tables contain an entry for practically all destination networks in the Internet. They are located in different POPs, two of them in Europe and the third in North America. All three monitors are Juniper routers that run JunOS. Each monitor runs a separate BGP session with each of the two collectors. In one of the sessions, the default OutDelay value of 0 seconds is used. We refer to the measured time series of updates for this session as M_0. The other session uses an OutDelay of 30 seconds. We refer to this time series as M_{OD30}. The two time series contain all BGP updates received in the period from March 7 2009 to July 6 2009[2].

If the BGP session between a monitored router and the collector is broken and re-established, the monitor will re-announce all its routing table. Such table transfers are a local artifact of the measurement infrastructure, and does not represent genuine routing dynamics. In order to remove these updates, we use the algorithm described in[15]. We verify the inferred table transfers against those identified using the BGP session logs, and find that the algorithm is able to identify all table transfers but with a mismatch in the starting time of the transfer in some cases, up to one minute in length. We therefore use the collectors' logs to identify the start of each

[1]This is the most common form of rate-limiting used in the Internet today, since it is turned on by default in Cisco routers.

[2]One of the collectors was unavailable for a few days in the beginning of May.

Monitor	M_0	M_{OD30}	M_{OD30}/M_0
A	302415	105084	0.35
B	286234	104553	0.37
C	257550	91550	0.36

TABLE I: Measured median daily churn

table transfer and use the reported length of the transfer by the algorithm plus one minute to decide the transfer period. This extra minute is added to the transfer duration to assure we do not include updates that belong to a table transfer in our filtered time series.

After filtering the updates caused by session resets, we record more than 161 million updates in M_0 and more than 56 million updates in M_{OD30} from the respective three monitors collectively. The total number of updates in M_{OD30} is about one third of those in M_0. However, both time series for each monitor contain spikes in the total number of daily updates, which exceed one million updates.

We further examine the measured daily churn across the three monitors in both configurations for similarity. Since the churn time series include outliers and that can affect the accuracy of a parametric correlation test such as Pearson's, we use instead the Spearman's rank correlation. This is a non-parametric test that does not assume a conformance between the underlying data and any probability distribution, and hence is less affected by outliers. The pairwise rank correlation coefficients between different monitors vary between 0.75 and 0.85 for both M_0 and M_{OD30}. The close association between our monitors is not surprising because they belong to the same AS and are connected in a full mesh of iBGP sessions. Because of space limitations and the close relation between our three monitors, we present results from only one monitor in the rest of this paper, unless stated otherwise.

The left and middle plots in Fig. 2 show the number of updates per day for M_0 and M_{OD30} respectively. A main observation from these plots is that the use of OutDelay for rate-limiting significantly reduces the number of BGP updates. Across the three monitors, churn levels were reduced by 64% over the measured period. This can also be observed in Tab. I, which shows that the reduction in the median daily churn rate is consistent across all three monitors.

There are significant spikes in the daily update rate in both M_0 and M_{OD30}. On closer inspection of the data, we find that these are normally caused by underlying events in the transit paths that affect a large number of destination prefixes simultaneously. Rate-limiting will reduce the number of updates for each individual prefix after such events, but at least one update still has to be sent for each affected prefix.

In the following section we investigate the impact of different rate-limiting implementations and timer values on churn reduction.

IV. Emulating rate-limiting

Our next goal is to quantify the different reduction in churn when performing rate-limiting with OutDelay and MRAI timers and the impact of the timer value.

A. OutDelay vs MRAI

Our measurement set-up captures only OutDelay of 30 seconds. Therefore, we use emulations for evaluating other configurations . The emulation script takes a series of BGP updates as an input. This series is collected from a BGP session that is configured with no rate-limiting. Furthermore, for emulating the MRAI timer, it identifies the timestamps at which the timer is supposed to expire. Then, it loops through the input workload and groups all updates for the same prefix that arrive between two consecutive timer instances. All grouped updates are invalidated but the last arriving update. This results in a new time series that reflects the effect of the timer. We also develop another script that emulates OutDelay.

For our purpose, we run the M_0 churn time series through the emulation scripts to emulate both MRAI and OutDelay timers of 30 seconds. The output from the scripts is two new time series that reflect the impact of the chosen timer implementation. We denote these emulated time series as E_{OD30} and E_{MRAI30} respectively.

To validate the sanity of the emulation scripts, we first apply them on M_0 using an OutDelay of 30 seconds, to obtain E_{OD30}. This time series is directly comparable to the measured time series M_{OD30}. For the two time series M_{OD30} and E_{OD30}, we count the number of updates in every hour in our measurement period. One hour granularity is chosen because it gives a reasonably large sample size, which improves our validation process. Three different statistical measures are then used for comparing M_{OD30} and E_{OD30}. Spearman rank correlation coefficient is used to measure the statistical dependence as it changes temporally. The correlation coefficient gives an idea about the relative dependence between two random variable, but it does not report similarities between absolute values. Therefore, we use Kolmogorov-Smirnov [11] and Kullback-Leibler divergence [7] tests to examine the similarity in churn distributions. The two-samples Kolomgorov-Smirnov (K-S) test examines the difference between two samples in order to check whether they come from the same population, while the Kullback-Leibler(KL) divergence test also measures the divergence between two samples but from information theoretic approach.

Table II shows the results of the statistical tests. M_{OD30} and E_{OD30} are strongly correlated with a Spearman's correlation coefficient ρ of 0.96 in all three monitors. The Kolmogorov-Smirnov (K-S) test reports relatively small divergence in all three monitors. Kullback-Leibler (K-L) divergence on the other hand, shows strong similarity between M_{OD30} and E_{OD30} in monitors A and C, but somewhat lower similarity in monitor B. The observed differences between M_{OD30} and E_{OD30} can be explained by the fact that the emulation is based on a different BGP session that operates independently from the measured timer-enabled session. Each session experiences independent session resets, and the timers in M_{OD30} and E_{OD30} are not synchronized.

The right plot in Fig. 2 shows the number of updates per day in one of our monitors for E_{MRAI30}. We observe that the

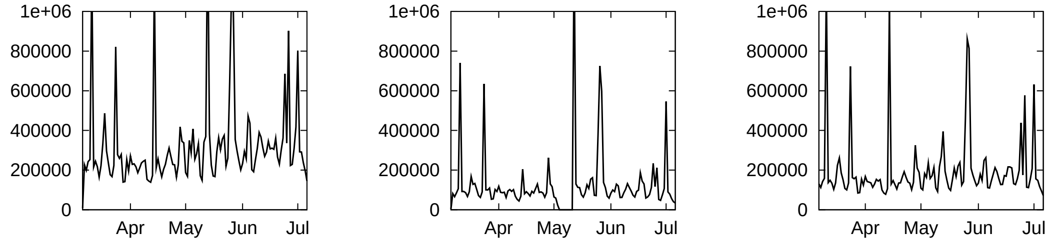

Fig. 2: Daily BGP churn: M_0 (left), M_{OD30} (middle), E_{MRAI30} (right)

Monitor	Spearman ρ	K-S	K-L
A	0.96	0.1863	0.06
B	0.96	0.1855	0.26
C	0.96	0.1315	0.03

TABLE II: Measurements vs Emulation

Monitor	M_0	E_{MRAI30}	E_{MRAI30}/M_0
A	302415	189377	0.63
B	286234	180339	0.63
C	257550	151472	0.59

TABLE III: Emulated median daily churn using MRAI timer

reduction in churn is smaller with MRAI timers than when using OutDelay with the same timer setting. This can also be seen in Tab. III, which shows the same ratio as presented in Tab. I for M_{OD30}.

B. Rate-limiting timer value

The extent of churn reduction achieved by rate-limiting is dependent on the arrival pattern of updates for each prefix, and the timer value. A longer timer helps in invalidating more intermediate states, but it is also increases convergence time [5]. We use M_0 as input to our emulation scripts in order to determine the churn reduction for different OutDelay and MRAI timer values in the range between 5 and 300 seconds.

The left panel in Fig. 3 shows how churn is reduced for increasing timer values. The y-axis shows the total churn in the measurement period with a timer value x, as a fraction of the total churn in M_0. A first observation from the figure is that the OutDelay implementation gives a stronger reduction in churn than the MRAI implementation, as explained in Sec. II. Recalling that rate-limiting timers delay routing convergence by up to one timer interval for each BGP session that an update traverses, this figure also illustrates the tradeoff between churn and the configured timer value.

Using regression analysis, we find a logarithmic decrease in churn for increasing timer values. For both the OutDelay and MRAI approaches, the fraction of churn in the rate-limited case scales as $R(T) = \alpha - \beta ln(T)$, where T is the timer value. For our data set, we find that $R(T)_{OutDelay} = 0.86 - 0.11 ln(T)$ with coefficient of determination 98.4% in the OutDelay case, while $R(T)_{MRAI} = 0.98 - 0.17 ln(T)$ with coefficient of determination 99.3% in the MRAI case.

We observe that a timer value of 5 seconds results in a 31% reduction in the level of churn in the OutDelay implementation, while a timer value of 10 seconds cuts down the level of churn by 39%. This shows that a significant reduction in churn can be achieved even with a relatively low timer value. This effect can be further understood by looking at the distribution of update inter-arrival times for individual prefixes, shown in the middle panel of Fig. 3. This figure shows that a significant fraction of updates arrive shortly after the previous update for the same prefix, and will hence be filtered out by the timer even at low values. Note that there are peaks in the inter-arrival time distribution around multiples of 30 seconds; these peaks can be explained by the timers employed on the incoming BGP updates to the monitored router. These observations suggest that the recommended default MRAI timer given in the BGP standard is often too conservative, as pointed out also in [6].

On the other hand, the right plot in Fig. 3 illustrates the convergence delay introduced due to rate-limiting. OutDelay delays convergence by one timer interval, while MRAI results in a delay of one half the timer value. The reported delay is the difference between the convergence time in M_0 and the respective implementation time series.

V. A MODEL FOR CHURN REDUCTION USING RATE-LIMITING TIMERS

The middle plot in figure 3 illustrates the importance of the update inter-arrival pattern for individual prefixes for determining the effect of rate-limiting. In this section, we characterize the distribution of update inter-arrival times, and use this information to develop a model for churn reduction using rate-limiting timers.

Let $f(t)$ denote the probability density function of the inter-arrival times for updates concerning a single destination prefix (the middle plot in Fig. 3 shows this function for one of the monitors in our dataset), and let $F(t)$ denote the corresponding CDF. With rate-limiting using OutDelay, all updates that arrive less than one timer interval T before the subsequent update for the same prefix will be invalidated. In this case, the remaining churn (as a fraction of the non rate-limited churn) is

$$R(T)_{OutDelay} = 1 - F(T) \quad 1 \leq T \tag{1}$$

With rate-limiting using MRAI timers, an update is invalidated if the subsequent update for the same prefix arrives within the same MRAI interval (i.e., before the MRAI timer expires). Hence, on average, an update is invalidated if the subsequent update for the same prefix arrives within $T/2$ seconds. Taking into account that the MRAI timer is jittered by multiplying with a random number in [0.75, 1], the remaining churn with an MRAI timer value of T is given by

$$R(T)_{MRAI} = 1 - F(\frac{0.875T}{2}) \quad 1 \leq T \tag{2}$$

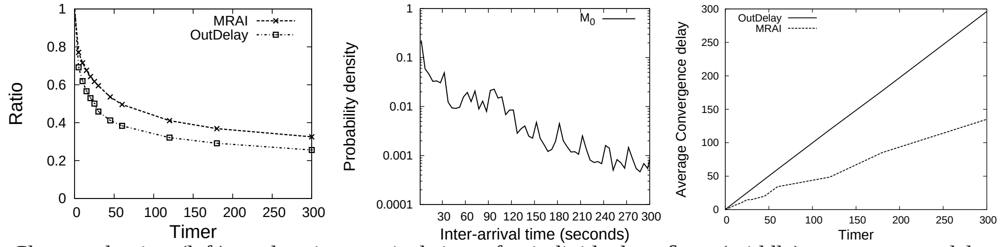

Fig. 3: Churn reduction (left), update inter-arrival times for individual prefixes (middle), convergence delay (right)

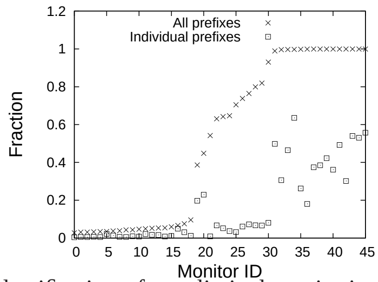

Fig. 4: Identification of rate-limited monitoring sessions.

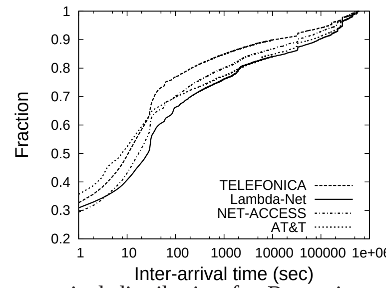

Fig. 5: Inter-arrival distribution for Routeviews monitors.

To formulate a model for $R(T)$, we need to characterize $F(T)$ using empirical data. A main question is how universal $F(T)$ is across different BGP sessions in the Internet[3]. To answer this, we look at update traces from a large number of monitoring sessions operated by the RouteViews project. However, we are only interested in monitoring sessions that are not rate-limited by MRAI or OutDelay, and this information is not available from the RouteViews repository.

To determine whether a monitoring session is rate-limited, we look at the time series of updates for that monitoring session. If a monitor uses MRAI, we expect to see a pattern where updates arrive in bursts every time the timer expires. In other words, we should see very few inter-arrival times in the range [1-22] seconds, assuming a jittered default MRAI timer value. On the other hand, if a monitor uses OutDelay to perform rate-limiting, we do not expect the same bursty pattern of updates. Instead, we should see a pattern where updates arrive in a steady flow, but where two updates for the same prefix are always spaced by at least the OutDelay timer value.

Figure 4 shows the fraction of update inter-arrival times for individual prefixes and across all prefixes that is less than 23 seconds, for all 45 monitors that peer with the RouteViews Oregon-IX collector. We have excluded all inter-arrival times of 0 seconds. The figure shows data from the first week of 2008; we have repeated the same exercise for the first week in each year from 2006 to 2009.

We observe a low fraction of inter-arrivals below 23 seconds for both individual prefixes and across all prefixes in first 20 monitors, which indicates that these monitors apply MRAI timers. The next 10 monitors show a low fraction of inter-arrivals below 23 seconds only when looking at each prefix individually, while there is a high fraction when looking at inter-arrivals across all prefixes. This indicates that these monitors perform rate-limiting using OutDelay. The last 15 monitors show a large fraction of inter-arrivals below 23 seconds for both individual prefixes and across all prefixes, indicating no rate-limiting. The set of non rate-limiting monitors include monitors in tier-1, large regional providers, and stub ASes.

For the identified non rate-limiting monitors, we calculate the inter-arrival distribution of updates for each prefix. For each monitor, we look at updates for the first two months in each year from 2006 to 2009. This gives us a set of 48 distributions that is diverse across both time and (topological) space. Figure 5 shows the CDF of inter-arrival times for single prefixes[4]. To keep the plot readable, we show results for only a subset of the monitors and only one year. Results of other monitors and years are similar.

The CDF plot shows a clear similarity between the inter-arrival distributions for the different monitors. To confirm this similarity we use the two-samples Kolomgorov-Smirnov (K-S) test, which confirms that all 48 inter-arrival distributions computed from the RouteViews data come from the same population at a confidence level of 95%.

Using non-linear regression on this data, we find that the CDF of the inter-arrival times for individual prefixes is on the form $F(T) = \alpha + \beta ln(T)$ for $1 \leq T \leq 300$. Averaging across the selected RouteViews time series, we find that $\alpha = 0.18$ with a standard deviation $\sigma_\alpha = 0.14$, while $\beta = 0.10$ with a standard deviation $\sigma_\beta = 0.01$. The parameter $\alpha = F(1)$ corresponds to the fraction of inter-arrival times that is less than or equal to 1 second. This parameter shows quite large variation across the time series. Looking closer at the data, we observe that monitors in stub networks show smaller α values, while monitors in large tier-1 ISPs show larger α values. This indicates that the value of α might be depending on path diversity and the connectivity at the monitor AS. Looking closer at this is part of our plans for future work.

Returning to our original goal, we can now give a for-

[3]We only consider eBGP sessions in this work.

[4]Since we are mainly interested in inter-arrival times in the order of a rate-limiting timer, we have imposed a maximum inter-arrival time of 1 week.

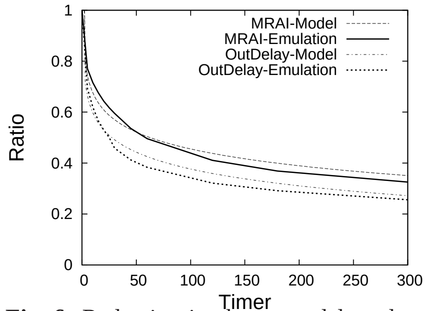

Fig. 6: Reduction in churn: model vs data

Monitor	M_{OD30}	$Model$
A	22.99	22.71
B	18.54	16.91
C	15.22	16.50

TABLE IV: Total number of updates (million), model vs data

mulation for the expected churn level as a function of the rate-limiting timer, based on the empirical data from the RouteViews monitors. Substituting $F(T)$ in (1) and (2), we get

$$R(T)_{OutDelay} = 0.82 - 0.10ln(T) \quad 1 \leq T \leq 300 \quad (3)$$
$$R(T)_{MRAI} = 0.90 - 0.10ln(T) \quad 1 \leq T \leq 300 \quad (4)$$

Figure 6 shows our model for $R(T)$ along with the churn reduction from our emulated rate-limiting in Sec. IV. Recall that using regression, we estimated $R(T)_{MRAI} = 0.98 - 0.17ln(T)$ and $R(T)_{OutDelay} = 0.86 - 0.11 * ln(T)$ based on our measurement data in that section. Table IV shows the total number of updates measured in M_{OD30} during our study period along with numbers approximated using our model. The model and the emulated data are in a good accordance; the model is able to predict the reduction in churn within a few percent for our dataset.

VI. Related work

Labovitz et. al [8] studied the impact of topology and routing policies on BGP convergence and suggested re-thinking the MRAI implementation due to its impact on BGP convergence time. Griffin and Premore [5] showed through simulation that there is an optimal MRAI value that minimizes convergence delay and network-wide number of updates which differs from a network to another. Later work [13] followed in the same direction and demonstrated through formal analysis, simulation, and PlanetLab experiment that the optimal MRAI value can be 5-10 times lower than the current recommended value. In addition, several other papers [4], [3], [10], [9] proposed modifying the MRAI timer to reduce the convergence delay.

Most of the previous work focused mainly on quantifying and reducing the impact of rate-limiting on BGP convergence times. Our work is the first to compare common rate-limiting implementations and to model their impact on churn.

VII. Conclusions

This work explores how different BGP rate-limiting implementations affect the level of churn. Measurements were performed on two parallel BGP sessions, with and without rate-limiting respectively, to three routers in a stub AS. Our measurements show that the sustained level of churn is strongly reduced when enabling the rate-limiting timer, and we explain how the OutDelay implementation (used in Juniper routers) gives a stronger reduction than MRAI timers (used in Cisco routers). Using emulation on the measured churn time series, we show that *the reduction is significant for both implementations already at low timer values* (which keep the convergence delay acceptable). With an OutDelay of 30 seconds, churn is reduced by as much as two thirds.

Using data from a large number of RouteViews monitors, we investigate the update inter-arrival pattern in BGP sessions that are not rate limited. We find a strong similarity in the distribution of inter-arrival times across monitoring sessions in different parts of the Internet, and across different years. This observed universality allows us to formulate an expression that quantifies the expected reduction in churn levels for different rate-limiting implementations and timer values. This expression is able to predict the churn reduction observed in our measurements within a few percent.

The expression for churn reduction given in this work will be useful for network operators in deciding a rate-limiting configuration that gives the right balance between churn reduction and convergence delay.

References

[1] Quagga project website. http://www.quagga.net/.
[2] Routeviews project page. http://www.routeviews.org.
[3] Y. Afek, A. Bremler-Barr, and S. Schwarz. Improved BGP convergence via ghost flushing. *IEEE Journal on Selected Areas in Communications*, 22(10):1933–1948, 2004.
[4] S. Deshpande and B. Sikdar. On the impact of route processing and MRAI timers on BGP convergence times. In *Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE*, 2004.
[5] T. Griffin and B. Premore. An experimental analysis of BGP convergence time. In *Proceedings ICNP*, 2001.
[6] P. Jakma. Revised default values for the BGP 'minimum route advertisement interval'. IETF Internet Draft, November 2008.
[7] S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 1951.
[8] C. Labovitz, A. Ahuja, R. Wattenhofer, and S. Venkatachary. The impact of Internet policy and topology on delayed routing convergence. In *Proceedings IEEE INFOCOM*, Anchorage, AK, USA, 2001.
[9] A. Lambert, M. Buob, and S. Uhlig. Improving internet-wide routing protocols convergence with MRPC timers. In *CoNEXT '09: Proceedings of the 5th international conference on Emerging networking experiments and technologies*, pages 325–336, New York, NY, USA, 2009. ACM.
[10] N. Laskovic and L. Trajkovic. BGP with an adaptive minimal route advertisement interval. In *IPCCC*, 2006.
[11] A. M. Law and D. M. Kelton. *Simulation Modeling and Analysis*. McGraw-Hill Higher Education, 1999.
[12] R. Oliveira, B. Zhang, D. Pei, R. Izhak-Ratzin, and L. Zhang. Quantifying path exploration in the Internet. In *Proceedings IMC*, Rio de Janeiro, Brazil, oct 2006.
[13] J. Qiu, R. Hao, and X. Li. The optimal rate-limiting timer of BGP for routing convergence. *IEICE Transactions*, 88-B(4):1338–1346, 2005.
[14] Y. Rekhter, T. Li, and S. Hares. A border gateway protocol 4 (BGP-4). RFC4271, January 2006.
[15] B. Zhang, V. Kambhampati, M. Lad, D. Massey, and L. Zhang. Identifying BGP routing table transfers. In *MineNet '05: Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data*, pages 213–218, New York, NY, USA, 2005. ACM.