An Experimental Analysis of BGP Convergence Time

Timothy G. Griffin
AT&T Research
griffin@research.att.com

Brian J. Premore
Dartmouth College
beej@cs.dartmouth.edu

Abstract

The Border Gateway Protocol (BGP) is the routing protocol used to maintain connectivity between autonomous systems in the Internet. Empirical measurements have shown that there can be considerable delay in BGP convergence after routing changes. One contributing factor in this delay is a BGP-specific timer used to limit the rate at which routing messages are transmitted. We use the SSFNet simulator to explore the relationship between convergence time and the configuration of this timer. For each simple network topology simulated, we observe that there is an optimal value for the rate-limiting timer that minimizes convergence time.

1 Introduction

The Border Gateway Protocol (BGP) [17, 11, 18] is currently the only Internet routing protocol used to maintain connectivity between autonomous systems. BGP is a *path vector* protocol where each router selects best routes to destinations based on the routes advertised by neighboring routers. Labovitz *et al* [15, 16] measured routing changes in the Internet and showed that there can be considerable delay in BGP convergence. They also observed that high levels of route fluctuation during delayed convergence have an adverse effect on end-to-end traffic delay, resulting in packet loss and intermittent disruption of connectivity.

There are two types of update messages sent by BGP. An *advertisement* informs neighboring routers of a path to a given destination. Advertisements include an AS path that contains the autonomous system numbers of all ASes the route advertisement has traversed from the AS originating the destination. BGP avoids inter-AS routing loops by prohibiting a router from installing a route that has that router's AS number in its AS path. A *withdrawal* is an update indicating that a previously advertised destination is no longer available. Advertisements can function as *implicit* withdrawals if they contain a previously announced destination.

There are two primary causes of BGP delayed convergence. The first is that the distributed nature of BGP path selection can lead to a set of routers simultaneously enumerating multiple alternate paths that are repeatedly eliminated and replaced with other choices until every router finally arrives at a stable path. Second, BGP advertisements are rate-limited using timers associated with the value **Minimum Route Advertisement Interval** (MRAI). The value of MRAI is configurable, although the recommended default value is 30 seconds [17]. When a router sends a route advertisement for a given destination to a neighbor it starts an instance of this timer. It is not allowed to send another advertisement concerning this destination to that neighbor until the associated timer has expired after MRAI seconds. This rate limiting acts to dampen some of the oscillation inherent in the path vector approach to routing. While waiting for an MRAI timer to expire, a router may receive many messages and run the BGP decision process multiple times. This allows a router to privately enumerate many alternative choices of its best path without exposing its neighbors to every intermediate step. Thus rate limiting reduces the number of updates needed for convergence at the cost of adding some delay to the messages that are sent.

Labovitz *et al* [15] also conjecture that convergence delay is increased by two common implementation decisions: the lack of *sender side loop detection* (SSLD), and the use of *withdrawal rate limiting* (WRATE). SSLD refers to an optimization whereby a router detects AS path loops before an advertisement is sent to a neighbor. WRATE is the application of rate limiting to withdrawal messages as well as advertisements, even though RFC 1771 [17] states that this should not be done.

The open-ended and programmable nature of BGP routing policies provide a very flexible and adaptable protocol. However, a rather unpleasant consequence of this flexibility is that BGP is not guaranteed to converge [19, 9, 8]. In other words, there are no bounds on BGP convergence time, in general. This does not invalidate the study of convergence time, though, as long as we restrict ourselves to considering only those network configurations in which convergence is guaranteed.

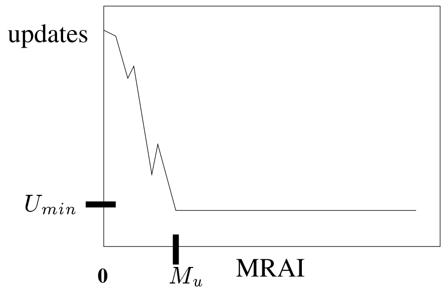


Figure 1. Optimal MRAI, M_u.

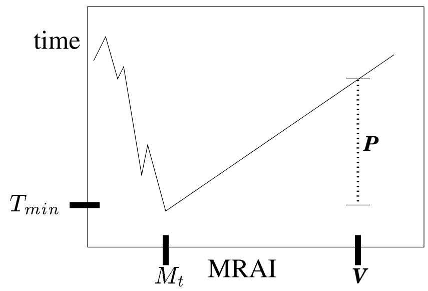


Figure 2. Optimal MRAI, M_t.

1.1 Summary of results

The goal of this work is to determine the impact of MRAI, SSLD, and WRATE on BGP convergence time. Our approach is to use simulation of BGP using SSFNet [1]. We study convergence in a small number of simple model topologies. For each topology considered, we perform two kinds of convergence experiments:

UP: An *origin* node advertises a single destination, and the system is allowed to converge.

DOWN: In the stable state produced by the UP experiment, the origin withdraws its destination, and the system is allowed to converge.

For each kind of experiment we perform multiple simulations using different network sizes and topologies, a range of values for the MRAI parameter, and all combinations of the SSLD and WRATE options. We briefly summarize the main observations.

Observation 1: For each network topology and each kind of experiment (UP or DOWN), there is an optimal value for MRAI, denoted as M_u, beyond which the average total number of updates required for convergence is stable.

The value of M_u may be 0 in some networks. For example, in a simple chain of n one-router ASes the number of updates required for convergence does not depend on MRAI. But in networks with multiple alternate paths, the total number of updates sent before convergence often depends on the value of MRAI. In these networks we observe a relationship similar to the one depicted in Figure 1, which presents the total number of updates produced in the network before convergence as a function of MRAI (averaged over several experiments with different random seeds). As MRAI increases from 0 (no rate limiting), convergence requires fewer and fewer updates until MRAI is M_u, after which the average number of updates remains stable at U_{min} with a very small variance.

This observation suggests that the MRAI rate limiting effectively dampens some of the route oscillations inherent in a path-vector protocol such as BGP. However, U_{min} should not be taken as the minimal *possible* number of updates required for convergence. It only represents the minimum that can be achieved with the use of a rate limiting timer.

Observation 2: For each network topology and each kind of experiment (UP or DOWN), there is an optimal value for MRAI, denoted as M_t, where average convergence time is minimized to T_{min}. Beyond M_t, average convergence time increases linearly.

Figure 2 shows the penalty P incurred by using an MRAI value V such that $V > M_t$. The averages are again taken from several experiments using different random seeds. Note that the values of M_t and M_u are usually very close, but may differ due to the fact that M_u is defined with respect to the network-wide sum of updates, while M_t is the maximum router convergence time over all routers.

Observation 3: The optimal MRAI value M_t increases with average router workload while U_{min} remains stable.

Observation 4: An optimal value for MRAI can dramatically decrease convergence time. However, this optimal value varies from network to network, and may be difficult to approximate in practice.

Observation 5: In terms of convergence time, WRATE can result in either a gain or a loss, depending on the network and the experiment type (UP or DOWN). If MRAI is close to M_t, then WRATE has little effect.

Observation 6: SSLD never increases convergence time, and may decrease it by a small amount.

1.2 Outline

The SSFNet simulator and its BGP implementation is described in Section 2. We describe our experimental framework in Section 3 and introduce the results for a family of network configurations called CLIQUE, which was singled out as a worst case in [15]. Section 4 discusses results for three other model network configurations. Open questions and future directions are discussed in Section 5.

2 SSFNet

SSFNet is a Java-based simulation package capable of modeling large, complex networks [3, 1]. The package includes a simulation kernel, an open source suite of network component models, a random number management suite, and a configuration language (DML). DML is used to build models by composing network components in a hierarchical fashion. Included components, such as routers and protocols, are configurable and can also be integrated with new components written by the user.

In SSFNet, simulation is performed at the IP packet level. Routers and hosts may contain multiple protocols, which are organized as a protocol graph in the spirit of the *x*-kernel [14]. Among the protocols provided are IP, TCP, Sockets, BGP, OSPF, and client applications. Running a simulation requires two parameters: the length of the simulation in seconds, and a DML specification. Before initiating a simulation, all components query the user-specified DML database and self-configure. The modeler may choose to leave certain configuration tasks up to the simulator. For example, SSFNet will perform automatic CIDR-compliant IP address assignment on a network unless the DML configuration explicitly names the IP addresses for all interfaces on all nodes. Several such automated tasks exist for the BGP module.

The BGP implementation is compliant with the BGP-4 specification in RFC 1771 [17], although currently it does not include some optional extensions which are common in commercial implementations. The module does include some behavioral toggles not found in commercial implementations, such as those for SSLD and WRATE.

A suite of tests is included with the BGP model covering basic peering session maintenance, route advertisement and withdrawal, route selection, route reflection, and internal BGP. More complex networks are used to test the general behavior of the protocol in terms of proper end-to-end data delivery.

Each BGP instance may be configured using DML in a fashion resembling the configuration of commercial routers. Alternatively, a default auto-configuration mode may be used to simplify configuration. The behavior of protocol instances may also be supplemented with the addition of hand-coded pseudo-protocols. For example, to force a router to inject advertisements for a new test destination at a particular time, we add a pseudo-protocol (in the form of a Java class) above BGP in the router's protocol stack. This protocol waits until the desired simulation time, composes an update message, and then calls BGP's method for sending updates.

Observing the behavior of a simulation requires configuring additional attributes (in DML) that indicate which events are of interest. SSFNet's BGP has an array of more than 50 switches for recording particular events, protocol states, or routing tables. This event-reporting facility is the method by which statistics about model execution are gathered. SSFNet also provides reproducibility so that any simulation can be repeated with identical results, even if additional event-reporting switches are turned on.

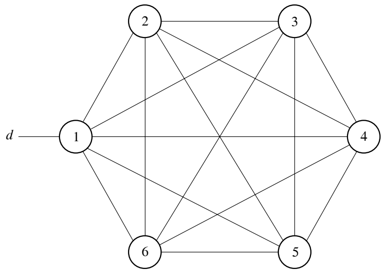


Figure 3. Topology of CLIQUE, size 6.

We use a set of Perl scripts to automate some of the tasks required to run a large number of SSFNet experiments. These scripts allow us to specify a range of parameters to explore, perform a run of SSFNet for each point in this space, and then collect, filter, and aggregate the SSFNet output. For example, the results covered in this paper required over 200,000 runs of SSFNet and more than a week of CPU time.

3 Experimental Framework and the Model CLIQUE

In our network configurations, each AS consists of a single router. This simplification ignores effects due to internal routing and internal BGP. In each network configuration, AS 1 is the *origin*, which announces and later withdraws a single destination prefix *d*. This is the only destination prefix processed by BGP in the simulations. These and other simplifications are employed in order to simplify analysis, and they are discussed in Section 5.

To compensate for having just one destination prefix in the system, we estimate and impose additional synthetic router workload. However, MRAI timer behavior is affected only by messages regarding *d*, and no jitter is added to MRAI.

Our experiments examine four families of network configurations. Each family is defined by its topology and routing policies. Each network configuration within a family is identified by its size. In this section, we use the family CLIQUE as a running example.

CLIQUE**:** A network configuration of size n in the CLIQUE family is made up of n autonomous systems in a full mesh. Route selection is based on shortest AS path. Figure 3 illustrates a CLIQUE of size 6. This network configuration was singled out as a model *worst case* network in Labovitz *et al* [15], since in the DOWN phase each router can potentially enumerate a large number of alternate paths to the origin.

The *workload* on a router (actually, its CPU) is a measure of the amount of work performed. When a BGP update message arrives at a router, the processing of that update may be delayed because the CPU is busy with other tasks, such as handling other BGP messages, OSPF calculations, and servicing SNMP requests.

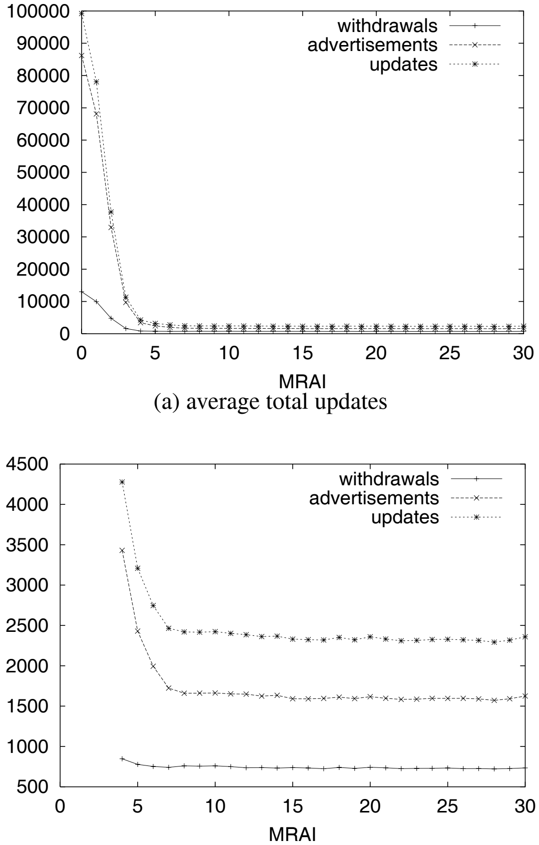
We model delays induced by router workload in a simple manner. Each router uses a FIFO queue for incoming BGP messages. When a message is removed from the queue it is assigned a delay value. This is the delay induced by the router workload and used to model CPU processing time. For our models, this delay is chosen randomly from a specified range. The total CPU delay induced by workload on an update message is thus the sum of its workload-induced delay and the workload-induced delay of all other BGP messages that were in the queue upon its arrival. These delays provide the only source of randomness in our simulations.

For each family F of network models (for example, CLIQUE), we have a function I_F that constructs an SSFNet model for an instance of this family. These functions take the following parameters $\vec{p}$:

n: size (number of ASes),
m: MRAI (in seconds),
p_{min}: minimum update processing delay (in seconds),
p_{max}: maximum update processing delay (in seconds),
ld: link delay (in seconds),
w: WRATE (boolean),
l: SSLD (boolean),
s: random number seed.

We simplify construction and analysis by applying parameters uniformly over each model. That is, all routers have the same values for m, p_{min}, p_{max}, w, and l, and all links have the same value for ld.

Given parameters $\vec{p}$, we run SSFNet on the model $I_F(\vec{p})$. For each experiment we record several measurements. We define the convergence time for a given phase of a given experiment to be the total amount of simulation time which elapsed from the time that the origin router sent out its first update message to the earliest time after which no router spends any more time processing update messages that resulted from the original update. This implies that a router hasn't converged until it processes all of the update messages it receives, even if these messages do not cause it to select a new best route. We express this as $time(f, \vec{p})$, where f is either UP or DOWN. We define the number of updates in a phase as the total number of advertisements plus withdrawals which were sent, by any router, during the phase. This is expressed similarly as $updates(f, \vec{p})$.



(a) average total updates

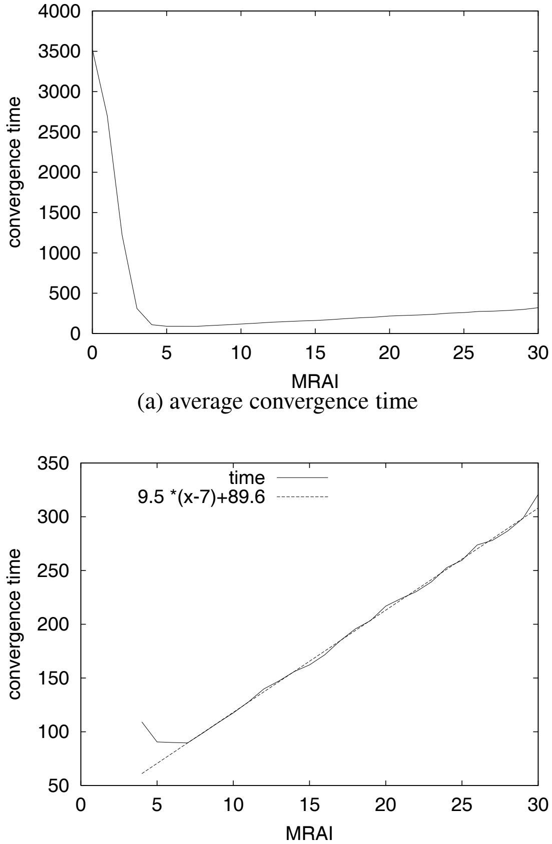
(b) average total updates (MRAI greater than 3)

Figure 4. Updates for CLIQUE of size 15, DOWN phase ($p_{min} = 0.1$, $p_{max} = 1$, no WRATE, no SSLD).

We now describe results for DOWN experiments using CLIQUE of size 15. For this set of experiments the CPU delay range was from $p_{min} = 0.01$ to $p_{max} = 1$, link delays were $ld = 0.01$, and WRATE and SSLD were not used. We note that our CPU delay values are quite high, and argue later (in the discussion of Figure 6) that the results we present scale with the average CPU delay. In order to explore variability due to the random CPU delays (which impart their randomness to the ordering of update messages), each experiment was repeated using 30 unique seeds.

Figure 4 (a) presents the average (over all seeds) total number of updates needed for convergence in these experiments. Figure 4 (b) presents this graph for MRAI values greater than 3. Figures 5 (a) and (b) present average convergence times for MRAI values ranging from 0 to 30. The average optimal MRAI value is $\overline{M_t} = 7$.

When MRAI is 0, there is no rate limiting and a large number of updates are needed for convergence (on average 100,000). About 3524 seconds (nearly 10 hours) is the average time required for convergence. Each router, not includ-



(a) average convergence time

(b) average convergence time (MRAI greater than 3)

Figure 5. Average convergence time for CLIQUE of size 15, DOWN phase ($p_{min} = 0.1$, $p_{max} = 1$, no WRATE, no SSLD).

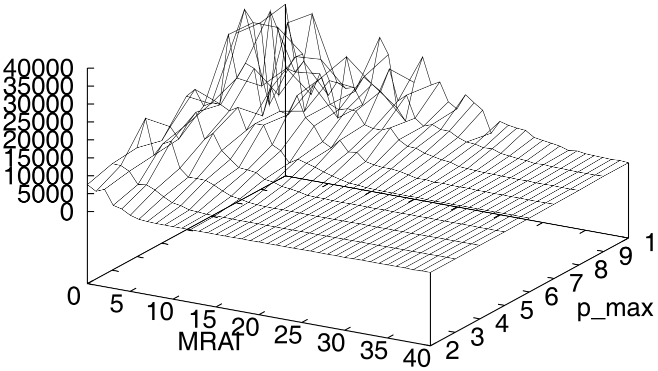


Figure 6. Average convergence time for CLIQUE of size 15, DOWN phase ($p_{min} = 0.01$, no WRATE, no SSLD). M_t depends on processor delay.

ing the origin, has to process about $100000/14 \approx 7143$ updates, each using on average 0.505 seconds ($7143 \times 0.505 \approx 3607$).

As MRAI increases, the number of updates decreases, until convergence time reaches an average minimum $\overline{T_{min}}$ of 89.6 seconds at an MRAI of $\overline{M_t} = 7$. After this point, the convergence time increases while the total number of updates remains relatively stable at about 2300.

Intuitively, each router has a number of "rounds" with each neighbor, where a round is the period of suppression of advertisements corresponding to one invocation of the MRAI timer. This bounds the number of advertisements that a router can receive during any interval of length MRAI seconds. However, the router's workload may be so large that it cannot process all of these advertisements within MRAI seconds. The MRAI value M_t is just large enough for all such updates to be processed.

In general, we see that the penalty for using an MRAI value $V > M_t$ can be closely approximated by $\overline{k} \times (V - \overline{M_t})$ for some value of $\overline{k}$ representing the average slope of this line, which corresponds to the average number of rounds per router required for convergence. If we let T_V be the convergence time when MRAI is V, then the line from $(M_t,\ T_{min})$ to (V, T_V) can be closely approximated by

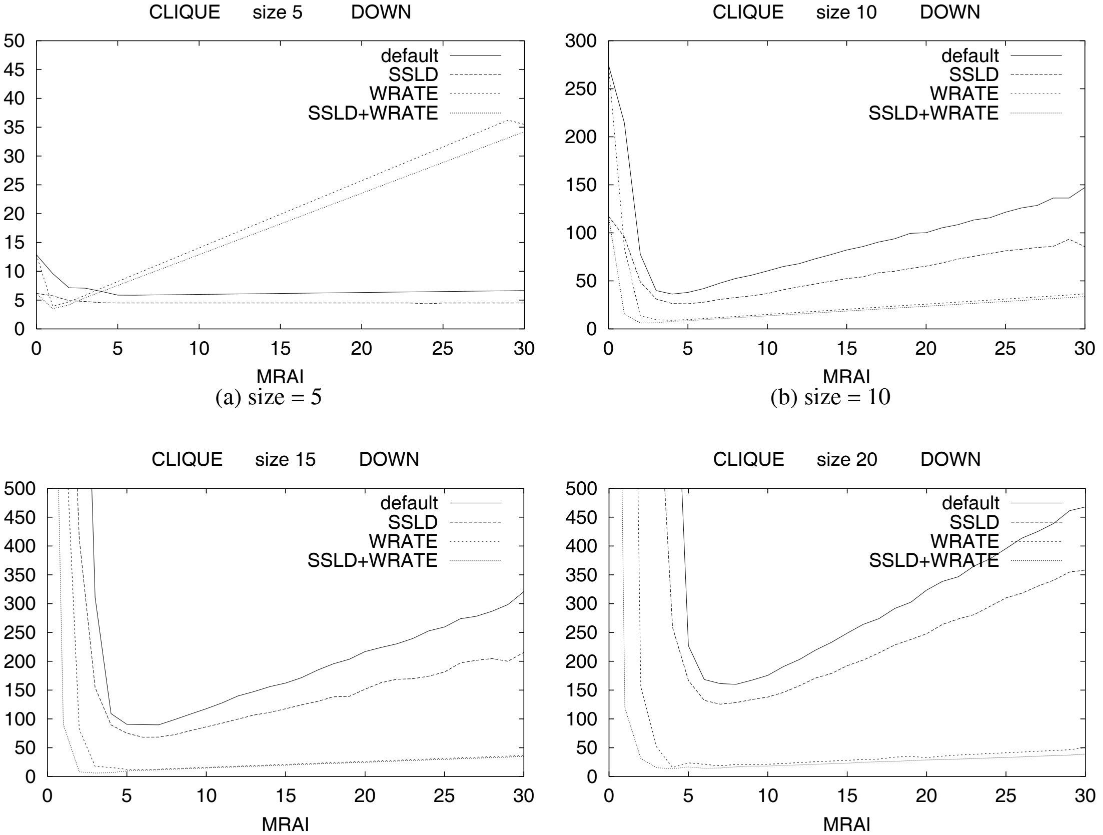
$$t = \overline{k} \times (x - \overline{M_t}) + \overline{T_{min}}.$$

For our CLIQUE experiments, we measured $\overline{k} = 9.5$, and Figure 5 (b) also plots the line $9.5 \times (x - 7) + 89.6$, which fits the measured convergence time after MRAI of $\overline{M_t}$.

The values that we have chosen for p_{min} and p_{max} are arbitrary, and a maximum value of 1 second may seem too high. However, there exist few studies detailing average router workload for BGP update processing. We can vary the value of p_{max} and examine the resulting changes in M_t and T_{min}. We found that as router workload increases, so do both M_t and T_{min}. This is because fewer messages can be processed within MRAI seconds, and a larger number of rounds may be required as workload increases. What would the maximum router workload have to be in order for M_t to be close to the default value of 30 seconds? Figure 6 shows convergence times for a CLIQUE of size 15, in the DOWN phase, as MRAI ranges from 0 to 40 and p_{max} ranges from 2 to 10. A maximum router workload of about 10 seconds (average of about 5 seconds) is needed before M_t exceeds 30 seconds.

The average value of M_t, denoted $\overline{M_t}$, also depends on the size of the CLIQUE, and on the values of the implementation options WRATE and SSLD. Figure 7 shows the plots of average convergence time for CLIQUE, sizes 5, 10, 15, and 20. Table 1 shows the values of $\overline{M_t}$ for different combinations of size and implementation options. For these experiments, $p_{min} = 0.01$ and $p_{max} = 1$.

Figure 8 presents average convergence times for these experiments. Figure 8 (a) shows the convergence times



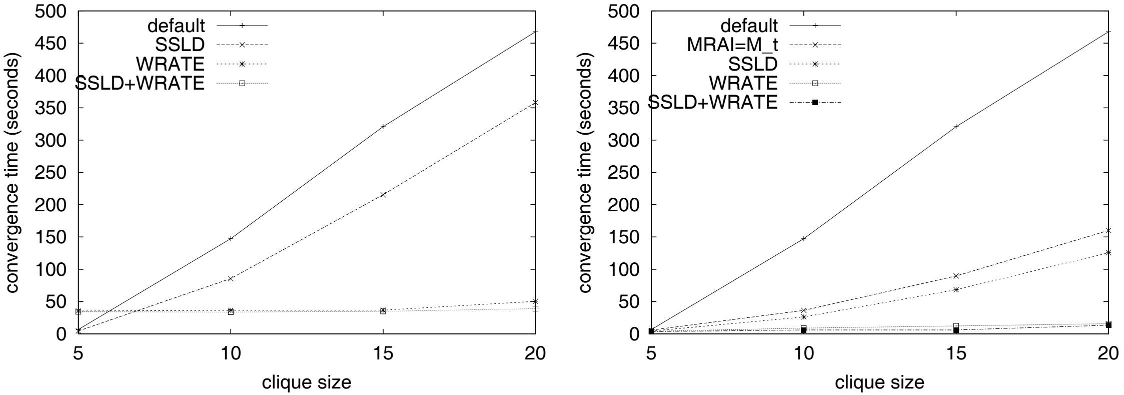
(a) size = 5

(b) size = 10

(c) size = 15

(d) size = 20

Figure 7. Average convergence times for CLIQUE, DOWN phase.



(a) MRAI = 30

(b) MRAI = $M_t(SSLD, WRATE)$

Figure 8. Average convergence times for CLIQUE, DOWN phase.

size	*default*	SSLD	WRATE	WRATE & SSLD
5	6	4	1	1
10	4	5	4	2
15	7	6	6	3
20	8	7	4	4

Table 1. Average M_t values for CLIQUE, DOWN phase.

when the default value of MRAI (30 seconds) is used. Figure 8 (b) shows convergence time when the value $\overline{M_t}$ from Table 1 is used. The plot for the default setting of MRAI is added to this graph for comparison.

In this model network, using WRATE improves convergence time dramatically. This is the exception among our experiments. Also, this is only true for the DOWN phase. For the UP phase, neither WRATE or SSLD have a significant impact on convergence time, while optimizing MRAI does. MRAI impacts the UP phase of CLIQUE because, with high probability, some node will hear about a two-hop path before it hears about the one-hop path to the origin. This path will be selected and advertised, only to be quickly replaced by another selection, whose advertisement is delayed by MRAI timers.

4 An Overview of the Experimental Results

In this section we summarize our experiments for the following additional models. The data for these experiments is presented in the full version of this document [10].

RING: A network configuration of size n in the RING family has n ASes in a ring (Figure 9 (a)). Route selection is based on shortest AS path.

FOCUS: A network configuration of size n in the FOCUS family has $n-2$ parallel paths of length two, all terminating at AS n (Figure 9 (b)). Route selection is based on shortest AS path. This type of topology corresponds to a customer (AS 1) multihoming to $n-2$ upstream providers, while AS n might be thought of as a Tier I provider [5, 12, 13].

P-CLIQUE: A network configuration of size n in the P-CLIQUE family has the same topology as a CLIQUE. The routing policy is as follows. For a given AS i where $i \neq 1$, AS 1 is treated as a customer network and all other ASes are treated as peer networks. That is to say that AS i will pass on advertisements to any of its peer ASes (any AS j where $i \neq j$ and $j \neq 1$) regarding routes that it learns from its customer, AS 1. Furthermore, AS i will refrain from passing on any advertisements that it receives from its peers. These types of policies are common in practice [5, 12, 13].

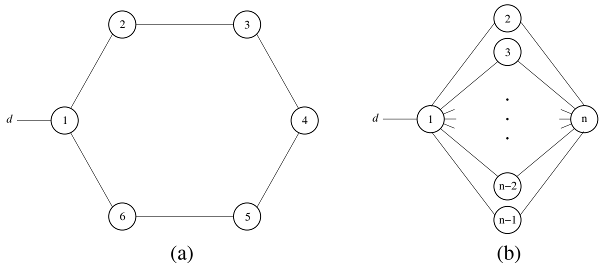


Figure 9. Topologies for a RING of size 6 and a FOCUS of size n.

While we observe a wide variety of idiosyncratic behaviors between different families and phases of experiments, there are consistent trends across all simulations.

Each phase of the experiments with these three models shows a clearly defined optimal MRAI value with respect to updates (refer to Figure 1). That is, there is an MRAI value beyond which the total number of updates required for convergence remains stable (near-constant with very small variance). This holds true without exception. Below the optimal (if it is not 0), the total number of updates increases. This increase is most dramatic in models with a greater number of alternative paths, such as FOCUS. In all experiments, the value increases proportionally with the model size.

A similar optimal MRAI value with respect to convergence times is also apparent in each of these models (refer to Figure 2). Analogously, the convergence times below this optimum increase, often rapidly. Above it, however, the times always increase linearly in proportion to the number of rounds—the approximate number of MRAI timer invocations per router—in the simulation.

The use of an optimized MRAI value proves highly beneficial during the UP phases in those networks with large numbers of alternative paths to the destination. Reduction in convergence times is typically greater than 50%. FOCUS is a good example which demonstrates this property (Figure 10), as is CLIQUE. When there are few alternative paths, the use of an optimal MRAI has little or no effect on convergence times. The UP phases of RING and P-CLIQUE fit this criterion.

Intuitively, having multiple alternative paths decreases the likelihood that a router will learn of the best path first, thereby increasing the likelihood that it will send more than one advertisement within the MRAI, thus causing an MRAI-delayed update. In P-CLIQUE, although topologically there are many possible routes, the numbers are highly restricted by the filtering policy in use, thus negating the impact of MRAI.

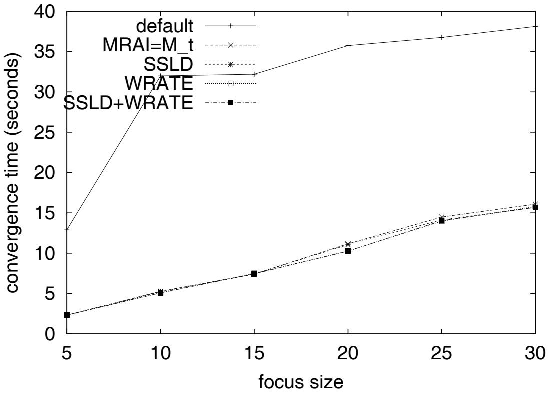


Figure 10. Average convergence time for FOCUS, UP phase. Using M_t has a significant impact, while the benefits of SSLD and WRATE are negligible.

SSLD consistently reduces convergence time during DOWN phases, though the benefits are small. P-CLIQUE (and CLIQUE) produce the best results with reductions of about 25%, although smaller values are more common. SSLD has little effect on convergence time in the UP phases. While SSLD consistently reduces the number of updates sent and thus the amount of processing time spent by routers, this almost always represents only a tiny fraction of the total processing time required by the system.

WRATE, when applied with MRAI $= 30$, incurs heavy penalties in nearly all cases. The negative impact of WRATE diminishes as network size increases, because the number of times that the MRAI timer is invoked grows very slowly in n. Using MRAI $= M_t$ in conjunction with WRATE usually minimizes the negative effects, and it becomes clear that in this case WRATE is most effective when the number of alternative routes is very large, as in CLIQUE in the previous section. As the number of alternative routes diminishes, so does the impact of WRATE.

Convergence times for SSLD combined with WRATE, using MRAI $= 30$, are highly network-dependent, and have unpredictable but explicable results. When SSLD is ineffectual by itself, it also has no effect when combined with WRATE. WRATE cannot introduce any detectable loops which would not have already been present before. However, it is possible for small improvements in SSLD to completely neutralize large negative effects of WRATE, as in the UP phase of P-CLIQUE (MRAI $= 30$ case). This suggests that SSLD prevents many routes with loops from being advertised, thus implicitly preventing any later withdrawals for those routes as well. This avoids possible invocations of the timer due to withdrawals, which can have a large impact on convergence time. In at least one case (the FOCUS DOWN phase), SSLD is only able to partially compensate for the impact of WRATE. In this case there are too many withdrawals of valid (non-looping) routes for SSLD to overcome. When using optimized MRAI values, combining SSLD and WRATE provide nearly identical results as SSLD by itself in all cases, with the lone exception of CLIQUE.

There are many behaviors specific to given network configurations which are evident in our observations. The explanations for many of these behaviors is not immediately clear. To illustrate this general phenomenon, consider one example. In the DOWN phase of the RING experiments, we see small improvements with the use of SSLD. This initially appears to be counterintuitive. After the UP phase, at least two routers will know of two paths to the origin, and in the DOWN phase at least one router will have its best path withdrawn before hearing of the withdrawal for its backup choice. Upon receiving the first such withdrawal, it initiates a "chain" of advertisements for the backup route going back towards the origin. The final link in this chain is the advertisement back to the origin of the path which traverses the entire loop. SSLD prevents this advertisement.

In general we see that the effects of SSLD and WRATE are negligible when using the optimal value for MRAI. This suggests that the MRAI value is the most dominant of these factors affecting the convergence times. For all phases of all of our experiments, the optimal MRAI value with respect to convergence time (M_t) was 10 seconds or less over 90% of the time, and was 15 seconds or less over 95% of the time. However, it should be noted that in many cases, converge time increased very rapidly for values just below the optimal (see Figure 5 (a)), suggesting that underestimating the optimal value could prove harmful.

5 Discussion and Directions for Future Work

We have presented simulation results on the relationship between the BGP rate-limiting timer, various BGP implementation techniques, and router workload delay, and how they affect convergence time in simple network topologies. Our models incorporate several simplifications, and we now discuss what can be done in the future to make them more realistic.

One simplification in our models is the lack of another widely used means of suppressing BGP route oscillations, the *route flap dampening* mechanism of RFC 2439 [20]. The MRAI timer can be thought of as a means of dealing with the short-lived oscillations that are *inherent* in a distance vector protocol. In contrast, route flap dampening is a complex technique aimed at suppressing long-term oscillations induced by network failures and misconfigurations. The MRAI value is fixed and applies to all routes, while route flap dampening requires a history to be maintained for each misbehaving route, and punishment accumulates, or is shed, depending on past behavior. The MRAI timers and route flap dampening can potentially interact in complex ways. We need to understand if route flap dampening is being invoked by oscillations *inherent* in the BGP protocol, as opposed to oscillations due to network failures. This may play an important role in the ultimate explanation of observed delays in Internet routing convergence.

It would be worthwhile to perform simulations with network models that more closely resemble the actual topology of the Internet [6, 2, 21, 4, 7]. This should include an attempt to closely model the routing policies used by ISPs [16, 5] and the impact of these policies on convergence time. In addition, the complexities of the internal topologies of autonomous systems and the related configuration of internal BGP may prove to be important factors in delayed convergence. Empirical measurements are needed to better estimate the CPU delay incurred in update processing, and the transit delays of BGP messages, especially in the context of internal BGP. One point worth noting is that workload is actually a function of MRAI, though here we have treated it as an independent parameter. In fact, increasing MRAI influences router workload by limiting the number of messages in the system at any one time. Also, we have considered only a single destination, and did not simulate the route fluctuations of multiple destinations.

Another simplification in our models is that we consider only a single prefix originated by a particular AS in a network. In the global Internet today there are more than 100,000 prefixes announced by more than 11,000 ASes. Assuming that the optimal value for MRAI is the same for all prefixes originated by a given AS, then there could be more than 22,000 distinct optimal MRAI values (two per AS, corresponding to the UP and DOWN phases). It is not clear how to define a globally optimal value for MRAI. Should it be defined as an average? Or a maximum? Furthermore, should the MRAI value be the same in all locations of the Internet? For example, in those locations with very few alternate paths, perhaps MRAI could be set to 0. One thing which is clear, though, is that MRAI's default value of 30 seconds is somewhat arbitrary while its impact on BGP convergence time is tremendous.

Acknowledgments. We would like to thank kc claffy, Madanlal Musuvathi, Srdjan Petrovic, Jennifer Rexford, and Aman Shaikh for many helpful comments.

References

[1] SSFNet: Scalable Simulation Framework—Network Models. `http://www.ssfnet.org/`. See `http://www.-ssfnet.org/publications.html` for links to related publications.

[2] K. Calvert, M. Doar, and E. Zegura. Modeling Internet topology. *IEEE Communication Magazine*, June 1997.

[3] James H. Cowie, David M. Nicol, and Andy T. Ogielski. Modeling the global Internet. *Computing in Science & Engineering*, 1(1):30–38, January-February 1999.

[4] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the Internet topology. In *Proc. ACM SIGCOMM*, 1999.

[5] Lixin Gao. On inferring autonomous system relationships in the Internet. In *Proc. IEEE Global Internet Symposium*, November 2000.

[6] Ramesh Govindan and Anoop Reddy. An analysis of Internet inter-domain topology and route stability. In *Proc. IEEE INFOCOM*, April 1997.

[7] Ramesh Govindan and Hongsuda Tangmunarunkit. Heuristics for Internet map discovery. In *Proc. IEEE INFOCOM*, March 2000.

[8] T. G. Griffin, F. B. Shepherd, and G. Wilfong. Policy disputes in path-vector protocols. In *Proc. Inter. Conf. on Network Protocols*, November 1999.

[9] T. G. Griffin and G. Wilfong. An analysis of BGP convergence properties. In *Proc. ACM SIGCOMM*, September 1999.

[10] Timothy G. Griffin and Brian J. Premore. An analysis of BGP convergence time simulations. Technical report, Dartmouth College Department of Computer Science, Hanover, New Hampshire, 2001.

[11] Bassam Halabi. *Internet Routing Architectures*. Cisco Press, 1997.

[12] Geoff Huston. Interconnection, peering and settlements: Part I. *Internet Protocol Journal*, 2(1), June 1999.

[13] Geoff Huston. Interconnection, peering and settlements: Part II. *Internet Protocol Journal*, 2(2), June 1999.

[14] Norman C. Hutchinson and Larry L. Peterson. The *x*-kernel: An architecture for implementing network protocols. *IEEE Transactions on Software Engineering*, 17(1):64–76, January 1991.

[15] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed Internet routing convergence. In *Proc. ACM SIGCOMM*, August/September 2000.

[16] C. Labovitz, R. Wattenhofer, S. Venkatachary, and A. Ahuja. The impact of Internet policy and topology on delayed routing convergence. In *Proc. IEEE INFOCOM*, April 2001.

[17] Y. Rekhter and T. Li. A Border Gateway Protocol. RFC 1771 (BGP version 4), March 1995.

[18] John W. Stewart. *BGP4: Inter-Domain Routing in the Internet*. Addison-Wesley, 1998.

[19] K. Varadhan, R. Govindan, and D. Estrin. Persistent route oscillations in inter-domain routing. ISI technical report 96-631, USC/Information Sciences Institute, 1996.

[20] C. Villamizar, R. Chandra, and R. Govindan. BGP route flap damping. RFC 2439, 1998.

[21] E. Zegura, K. Calvert, and M. Donahoo. A quantitative comparison of graph-based models for Internet topology. *IEEE/ACM Trans. Networking*, December 1997.