There's something about MRAI: Timing diversity can exponentially worsen BGP convergence

Alex Fabrikant
Google Research*
fabrikant@google.com

Umar Syed
University of Pennsylvania
usyed@cis.upenn.edu

Jennifer Rexford
Princeton University
jrex@cs.princeton.edu

***Abstract*—To better support interactive applications, individual network operators are decreasing the timers that affect BGP convergence, leading to greater diversity in the timer settings across the Internet. While decreasing timers is intended to improve routing convergence, we show that, ironically, the resulting timer heterogeneity can make routing convergence substantially *worse*. We examine the widely-used Min Route Advertisement Interval (MRAI) timer that rate-limits update messages to reduce router overhead. We show that, while routing systems with homogeneous MRAI timers have *linear* convergence time, diverse MRAIs can cause *exponential* increases in both the number of BGP messages and the convergence time (as measured in "activations"). We prove tight upper bounds on these metrics in terms of MRAI timer diversity in general dispute-wheel-free networks and economically sensible (Gao-Rexford) settings. We also demonstrate significant impacts on the *data plane*: blackholes sometimes last throughout the route-convergence process, and forwarding changes, at best, are only polynomially less frequent than routing changes. We show that these problems vanish in contiguous regions of the Internet with homogeneous MRAIs or with next-hop-based routing policies, suggesting practical strategies for mitigating the problem, especially when all routers are administered by one institution.**

I. Introduction

The Border Gateway Protocol (BGP) [1], the Internet's interdomain routing protocol, reacts slowly to topology changes. After a failure, BGP routing messages propagate through a network of tens of thousands of Autonomous Systems (ASes) that must update their routing tables and start forwarding traffic along new paths. In the meantime, data packets are lost, delayed, or delivered out of order. A growing body of anecdotal accounts and measurement studies [2]–[4] show that BGP convergence is too slow for interactive applications like VoIP, multi-player games, and financial transactions. In an effort to reduce convergence time, router vendors and network operators are reducing the timers that control the BGP convergence process.

A. Reducing the MRAI Timer

By limiting the rate of update messages between BGP neighbors, the Minimum Route Advertisement Interval (MRAI) timer plays a critical role in BGP convergence. The MRAI timer trades off convergence speed against bad behavior during convergence—transient routing and forwarding changes, as well as high overhead on the routers. Although the BGP RFC recommends a default timer of 30 seconds [1], router vendors and the IETF alike are moving toward lowering or removing this recommendation entirely [5], [6]. Early simulation work [7] supports these decisions, suggesting that smaller timers may indeed reduce convergence time.

In this paper, we consider what *could* potentially happen if the proposed MRAI "improvements" get *incrementally* deployed, and uncover a problem. We show that the worst-case convergence time of the network may not get better until lower MRAI settings are *universally* deployed, while the network's behavior during convergence may get exponentially worse. That is, *incrementally-deployed changes to MRAI, and especially the complete deregulation thereof, can be destructive to both sides of the tradeoff that MRAI addresses*—both to convergence time and to behavior during convergence. Based on our results, we advocate a more careful deployment of MRAI changes.

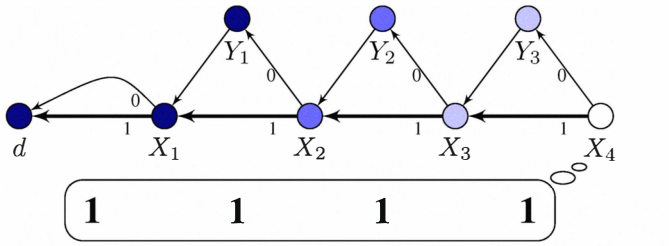
Studying convergence time only makes sense if the routing system converges in the first place. This property, *BGP safety*, has been studied in depth over the past decade [8]–[14]. Yet, oscillations rarely happen in practice, most likely because real BGP policies are constrained by the business and performance objectives. If we consider all networks that converge *eventually*, the worst-case convergence time looks hopeless—some BGP systems require exponential time to converge [12]. Instead, we focus on "reasonable" networks, constrained by the well-studied "No-Dispute-Wheel" (NDW) condition [10] or by the economic *Gao-Rexford conditions* (GR) [11]. We study convergence as a function of the number of nodes (in NDW networks) or the depth of the customer-provider hierarchy (in GR networks).

B. Example of Exponential Update Messages

Most analysis of convergence (e.g. [15], [16]) assumes routing changes happen in *fair phases*, where each BGP participant updates its routing state (i.e., "activates") at least once. Under the NDW and GR constraints, BGP converges after at most a *linear* number of phases [15], [16]. Yet, a network can also experience *exponentially* many routing changes [17], [18]. Both properties are evident in Figure 1, a variation on a construction in [17]. Each node X_i has 2^i possible routes to destination d. Node X_i's preferences are indicated by the binary number "spelled out" by the labels on the edges, read from left to right with higher numbers

* Work done while this author was a postdoc at Princeton University

978-1-4244-9921-2/11/$26.00 ©2011 IEEE



X_1	Y_1	X_2	Y_2	X_3	Y_3	X_4
1	1	11	11	111	111	1111
0	1	11	11	111	111	1111
0	0	10	11	111	111	1111
0	0	01	10	110	111	1111
0	0	01	10	101	110	1110
0	0	01	10	101	110	1101
0	0	01	10	101	101	1100
0	0	01	01	100	101	1011
0	0	01	01	011	100	1010
0	0	01	01	011	100	1001
0	0	01	01	011	011	1000
0	0	01	01	011	011	0111

- update is received, not yet processed for selection
- selected route is announced
- waiting to announce route, due to MRAI

Fig. 1. Sawtooth Gadget: exponential path exploration within a linear number of fair phases

preferred over lower ones. For example, X_3 prefers the path $X_3X_2X_1$ over the paths $X_3Y_2X_2X_1$ and $X_3Y_2X_2Y_1X_1$. With these preferences, the X_i nodes converge on paths using only the "1" edges.

In this example, a failure can lead to an exponential number of messages during convergence. Suppose that the nodes start in the stable state, with the X_i routing through "1" edges and each Y_i routing through its neighbor X_i. Now, suppose the leftmost "1" edge fails. The sole stable configuration after this failure has X_i using "1" edges all the way to X_1. However, with the given activation pattern, an exponential amount of path exploration will occur before convergence, as X_4 explores all nine routes from "1111" to "0111" in decreasing order. Meanwhile, the nodes activating least frequently, X_1 and Y_1, need activate only once, making the system converge in a single fair phase — well below the linear upper bound.

The exponential path exploration is enabled by a particular timing pattern, with nodes X_i and Y_i activating more frequently for higher i. In Figure 1, and in the rest of the paper, darker nodes activate more slowly, and lighter nodes activate faster. This timing is precisely a distinction in MRAIs, with darker nodes having slower MRAIs than lighter nodes. With the MRAI timers for X_i and Y_i set to twice as long as those for X_{i+1} and Y_{i+1}, extending the above pattern ensures that X_n will explore 2^{n-1} paths before convergence. Both here and throughout, we defer the formal inductive proofs of counter-example behavior to the full version of the paper.

C. Modeling Heterogeneous MRAI Settings

As the recommended MRAI values change, the deployment will naturally proceed incrementally. Some AS will change immediately to a range of lower MRAI settings, while other ASes continue using the current default configuration. The uncoordinated deployment will lead to much more heterogeneous MRAI values. We study BGP convergence behavior as a function of two coarse parameters of MRAI heterogeneity:

- **Disparity:** The ratio r between the highest and lowest MRAI values in use.
- **Diversity**: The number v of different MRAI settings in use.

To evaluate the impact of MRAI disparity and diversity, we consider the following measures of convergence after a network event:

- **Convergence time**: The time until the BGP system is stable ($\hat{T}$).
- **Routing updates**: The maximum number of routing updates sent along a particular edge ($\hat{r}$), and the number of updates sent system-wide ($\hat{R}$).
- **Forwarding changes**: The global number of forwarding changes ($\hat{F}$), as a measure of the impact on data traffic.

To set the stage for our results, Section II briefly reviews our model of BGP, directly borrowed from [15] (a somewhat simplified version of the more detailed queue-based model of [10]), and augments it with MRAI timers. In Section III, we show the relationship between both disparity and diversity of MRAIs and convergence behavior. In particular, we show that convergence time may *not* improve as long as the slowest MRAIs don't change. Section IV extends these results to forwarding behavior. In Section V, we discuss possible strategies to mitigate these problems and other implications of this work, as well as the limitations of our results and open questions.

II. BGP Model & Notation

To model BGP dynamics, we start with the simple BGP model and notation used in [15]. We model the Internet as a graph $G = (N, L)$ of routers as nodes, interconnected by physical communication links, and independently treat the problem of routing to any particular destination d. Router i has a set P^i of possible simple paths to d, and a preference function λ^i that has no ties between paths with different next hops. It may also have import and export policies specifying which routes can be imported from, and exported to, which neighbor.

The key change is the introduction of timing, with a per-neighbor, per-destination MRAI timer t_i assigned to each node. This contrasts both the arbitrary, adversarial asynchronous activations measured in terms of fair phases, as in [15], and a uniform global timing, as in [16], [19].

The MRAI timer constrains *announcement events*: If a node i has changed its route selection, and this change will be reflected in an update to its neighbor j (i.e., the new route is non-empty and exportable to j), i sends a corresponding announcement to j just after t_i seconds elapse since the last

announcement to j, or immediately, if the last announcement to j was more than t_i seconds ago. Route changes triggering withdrawals (i drops the route itself, or picks a route that it will not export to j) always result in immediate withdrawal announcements to j. We use t^* to denote the slowest MRAI in the system, $t^* = \max_i t_i$, and t_* to denote the fastest MRAI, $t_* = \min_i t_i$.

Since we aim to study the convergence of a network after some specific event, it is reasonable to assume that, at the time of the initial event, all the MRAI timers have had time to expire since the previous oscillation, and thus the first updated route that any node wants to announce gets announced immediately. We call this the *clean phase* model, in contrast to the more general *dirty phase* model where, at the time of the initial event, some nodes may still be in mid-MRAI delay, due to having recently announced something before the event. A dirty phase model may also be helpful in studying per-neighbor MRAIs, discussed in Section V. Almost everywhere below, we use, implicitly, the stronger of the two models: all but one of our counter-example results demonstrate bad behavior even under the clean phase restriction; all the positive results (upper bounds on convergence) work even in the more general, dirty phase model.

The MRAI heterogeneity measures we study are:

- Disparity: $r = t^*/t_*$, and
- Diversity: $v = |\bigcup_i \{t_i\}|$

For the *selection events*, we use the approach of [16]. Assume that, once node i has received any particular combination of updates that changes its preferred route, it would notice this, select the route, and be ready to export it, all within some short span of time, s_i. Throughout the paper, we assume that s_i is non-zero. That is, if a node has been waiting to send an update to a neighbor due to an MRAI timer, and the timer expires right as the node receives a new update, the first, soon-to-be-outdated, announcement is still sent out since the node has yet to process the new update. On the other hand, we assume that s_i is fast enough to treat it as negligible relative to the t_i settings, since modern routers can process updates far more quickly than the typical MRAI values of 5 and 30 seconds frequently in use today.

As established in [15], the above model, captures the relevant features of BGP for studying convergence, while abstracting away some of the details of lower-level processing of update messages as modeled by the well-known Simple Path Vector Protocol model of [10]. In terms of convergence, the only impact of disregarding the lower-level details, and effectively setting $s_i = \epsilon$ is that we discard the possibility of the router's internal processing taking so long that it meanwhile has a chance to send out *spurious* updates that don't correspond to the best route currently available to it. These are outsider our scope here and are treated in depth in [20].

III. Slow routing convergence

A. MRAI disparity and convergence

We first consider routing convergence when the fastest MRAIs are much faster than the slowest ones, yielding a high MRAI disparity r. We start with what appear like crude upper bounds directly derived from [15], and then use a delicate combination of some previously known BGP gadgets and some new constructions to show that these bounds are actually tight.

In [15]'s terminology, each "fair phase" takes at most $\max_i s_i + t_i \approx t^*$ time. Any node i that, at the start of the phase, is about to change its route selection will do so within s_i after the start, and will announce it within t_i after that. Any other node may as well be "activated" at the beginning of the phase, both for selection and for announcement, and nothing will happen there.

These bounds follow directly:

Corollary 1 (from 4.2 and 4.3 of [15]): NDW systems converge in $\hat{T} = O(nt^*) = O(nrt_*)$ time. Each edge (i, j) will see at most $O(nrt_*/t_i) \leq O(nr)$ updates in that time, ensuring $\hat{r} = O(nr)$ and thus at most $\hat{R} = O(nmr)$ routing updates sent system-wide. Since node i can receive new information only $O(nr)$ times from each of its neighbors, it would only have $O(nr \deg(i))$ opportunities to change its forwarding, yielding $\hat{F} = O(nr \sum \deg(i)) = O(nmr)$. By parallel arguments for GR networks, $\hat{T} = O(\alpha r t_*)$, $\hat{r} = O(\alpha r)$, $\hat{R} = O(\alpha mr)$, and $\hat{F} = O(\alpha mr)$.

These bounds appear crude. The time bound derives from, effectively, just assuming that everyone's MRAI is as slow as the slowest one. The behavior during convergence bounds derive effectively from assuming that almost everyone's MRAI is as fast as the fastest one, and the maximum possible number of control and data plane events happens in the time allowed by the time bound.

Surprisingly, we find that these bounds are asymptotically tight. This establishes that the worst-case convergence duration in the network won't improve until even the slowest MRAI is sped up, while the number of control and data plane events can indeed be amplified linearly by a large disparity between the fastest and slowest nodes. Formally:

Theorem 2: In NDW networks with n nodes and m edges, in the worst case, $\hat{T} = \Theta(nrt_*)$, $\hat{r} = \Theta(nr)$, $\hat{R} = \Theta(nmr)$, and $\hat{F} = \Theta(nmr)$. In GR networks with α levels and m edges, the same bounds apply: $\hat{T} = \Theta(\alpha r t_*)$, $\hat{r} = \Theta(\alpha r)$, $\hat{R} = \Theta(\alpha mr)$, and $\hat{F} = \Theta(\alpha mr)$. The worst-case behaviors for all of these can occur in the same network.

Proof sketch: With the upper bounds already established above, we can demonstrate a family of GR networks parameterized by (1) α or n, and (2) $m = \Omega(n)$ that match all of the bounds simultaneously. Consider the "*Christmas tree*" gadget shown in Figure 2. The node set contains the destination d and 4 other groups of nodes:

- a "trunk" of $m/4n$ nodes T_j; and the remaining $n - m/4n$ nodes split equally into 5 groups of k:
- a "base" group of k pairs of nodes S_i and R_i,

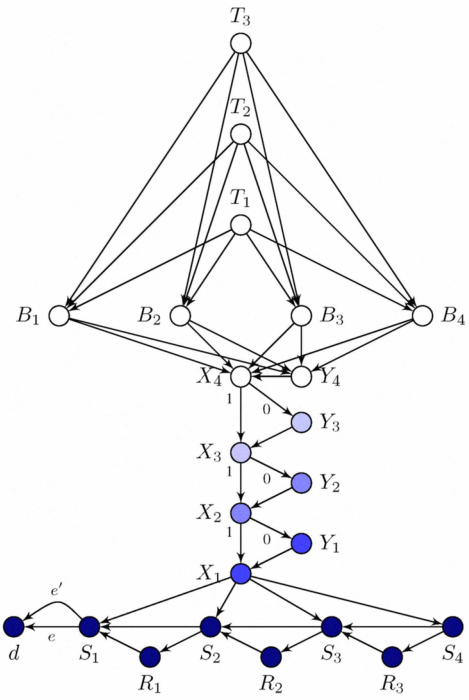


Fig. 2. A "Christmas Tree" gadget: worst-case behavior with disparity r

- a "stump" group of k pairs of left and right nodes X_i and Y_i, and
- a "branches" group of k nodes B_i.

The poor behavior of this system arises from the interaction between the "base" and the "stump" groups.

The "base" group uses a synchronous variation on the Sawtooth Gadget of Fig. 1. It guarantees that, after e fails, $\Theta(nt^*)$ time passes until all S_i's confirm that they can only route through e'. At it^* seconds after e fails, S_{i+1} will announce that it can only route through e', and then S_{i+2} announces a route through R_{i+1} leading to e, right as R_{i+1} announces to it that R_{i+1}, too, is now routing through e'. The MRAI of S_{i+2} will then require another t^* interval until the news spreads further right.

In the "stump" group, isomorphic to the Sawtooth Gadget of Figure 1 with its disparate MRAIs, the X_i's prefer any route through e over any route through e', and within each category prefer to go through the fewest S_i's, and, for each fixed S_i, use the same lexicographical ordering as in the Sawtooth Gadget. Each time a new base node announces that it now routes via e', the stump performs the full pattern of the Sawtooth Gadget. The fastest nodes, at the top, make $\Theta(r)$ forwarding changes, and send $\Theta(r)$ announcements out, for every one announcement that the "stump" receives from the "base".

The base thus ensures that $\Theta(n)$ phases will happen, and, crucially, that each phase will involve the worst possible behavior in the stump, yielding $\Theta(kr) = \Theta(nr)$ messages on, e.g., edge (X_{k-1}, X_k), and X_k performs $\Theta(nr)$ forwarding changes. The "branches" group yields a tight bound on the global count of routing updates and forwarding changes ($\hat{R}$ and $\hat{F}$) in sparse graphs, and the trunk lets us parameterize $\hat{F}$ and $\hat{R}$ by m.

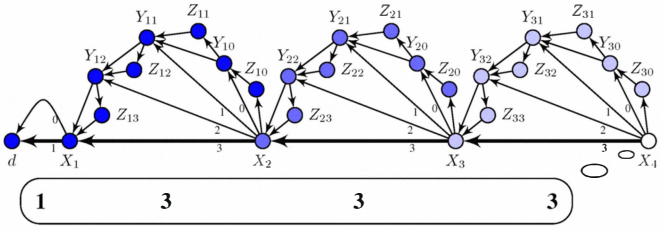


Fig. 3. A "Chain" gadget yields worst-case behavior with MRAI diversity v, shown here for $v = 4$, $n/v = 4$.

With the arrows set to point from providers to customers, this network obeys the Gao-Rexford constraints, with $\alpha = \Theta(n)$, yielding the matching Gao-Rexford bounds, too. ■

B. MRAI diversity and convergence

The construction of the previous section notably requires a broad variety of values for the MRAI timer. Both the IETF and major router vendors could potentially maintain the status quo and keep the number of different MRAI values in common use relatively small, via recommendations and standardized default settings, respectively. We now establish that this is indeed advisable, to ensure good worst-case behavior under convergence. That is, we show that the number of control and data plane updates during convergence can skyrocket exponentially as a function of MRAI diversity v (the number of different MRAI values in use):

Theorem 3: In an NDW network with n nodes that use $v \leq n/2$ distinct MRAI values among them, the number of control and data plane updates ($\hat{F}$ and $\hat{R}$) can be $(n/v)^{\Omega(v)}$. If the network is Gao-Rexford, with α levels and $v \leq \alpha/2$, $\hat{F}$ and $\hat{R}$ can be $(\alpha/v)^{\Omega(v)}$. For higher diversity, $v \geq n/2$ in NDW networks and $v \geq \alpha/2$ in GR networks, the behavior remains exponential, with $\hat{F}$ and $\hat{R}$ both bounded by $2^{\Omega(n)}$ for NDW and $2^{\Omega(\alpha)}$ for GR.

Proof sketch: The diversity-based lower bounds are achieved by the "Chain" gadget in Fig. 3, which amalgamates the Sawtooth Gadget and the synchronous behavior of the Christmas Tree gadget's base in a different way, by replacing each "tooth" of the Sawtooth Gadget with a copy of the "base". Now, each "link" of the chain, like the "base" above, shares MRAI value t_i and spends n/v "ticks" of the t_i timer to count down its "digit" from n/v to 0. With t_i being n/v times slower than t_{i+1}, this lets the preference functions produce exponential path exploration by simulating counting down v-digit numbers in base n/v rather than base 2,

The fastest node, X_v, will receive $(n/v)^v$ routing updates, each of which will trigger a forwarding change; the routing and forwarding events at this node will asymptotically dominate the global sum over such events, ensuring that $\hat{F}$ and $\hat{R}$ are both $(n/v)^{\Omega(v)}$, and, since $\alpha = n$ here, $(\alpha/v)^{\Omega(v)}$.

In the degenerate case of almost unique MRAIs, $v > n/2$, or if $v > \alpha/2$ in a GR network, using the chain gadget for $v = n/2$, and adding miniscule noise to some of the t_i's will not change the performance of the system while keeping the $\Omega(2^n)$ and $\Omega(2^\alpha)$ bounds. ■

Can $\hat{R}$ and $\hat{F}$ get even worse than this with diverse MRAIs? We show that the above example is fairly tight. That is, NDW networks with MRAI diversity v won't get substantially asymptotically worse:

Theorem 4: In any NDW network with v distinct MRAIs, the number of routing updates and forwarding changes is no worse than $(n/v^{1/3})^{O(v)}$.

Let us first split the nodes into "groups" $G_1, \ldots, G_v$ by distinct MRAI value, in order, so that G_1 are the nodes with the fastest MRAI, G_v are those with the slowest, etc. For each i, we consider what happens in the interval between two adjacent announcements by any "slow" nodes, defined, relative to i, as the set $H_i = \{d\} \cup \bigcup_{j=i+1}^{v} G_j$. We call such an interval of time an $i+1$-interval. From the viewpoint of the nodes in all of the "fast" nodes, defined as the set $G_{1,i} = \bigcup_{j=1}^{i} G_j$, the slow nodes have announced a particular route before the $i+1$-interval, and don't announce anything again for the rest of the $i+1$-interval, as if that route is "fixed" as a stable selection.

Lemma 3.1: Within an $i+1$-interval, routing changes can happen for at most $|G_{1,i}|$ *i-fair phases*, defined as sequences where each node in $G_{1,i}$ selects and announces to each neighbor at least once.

Proof of lemma: The proof requires "steinerizing" the inductive argument of [15]. Their result was an induction that grew a routing tree from the destination. Here, we will consider *any* subsets of nodes with "fixed" routes, and have the induction grow a *forest of route prefixes* from the slow nodes, whose announced but possibly outdated routes are effectively arbitrary route suffixes steadily available for use by fast nodes.

Inductively, suppose some strict subset of fast nodes $\emptyset \subseteq S \subset G_{1,i}$ is "steady", i.e. (1) each $s \in S$ has picked a route that follows only other steady nodes to a slow node (i.e. a route that starts with $s, s^1, s^2, \ldots, s^k, z$, with $s^j \in S$ and $z \in H_i$), and (2) this route is consistent with the choices of all the steady nodes that appear before the first slow node on the path (i.e., for all j, s^j also picks $s^j, \ldots, s^k, z$), and with the route announced by the first slow node z.

Given a set of steady and slow nodes and their selected routes, we say that a route $v_1, v_2, \ldots, v_l = d$ that starts at *any* fast node has a *consistent fast prefix* if it is consistent with the routes selected by its first slow node and any steady nodes that come before it (if, for all $k < j$, $v_k \notin H_i$, and $v_j \in S \cup H_i$, then v_j must have selected $v_j, \ldots, v_l$). Note that, by construction, up until the first slow node, steady nodes may only be followed by other steady nodes.

Pick an arbitrary fast, non-steady node $v^1 \in G_{1,i} \setminus S$. Let $P^1 = (v^1 = v_1^1, v_2^1, \ldots, d)$ be v^1's highest-preferred path among all paths that have a consistent fast prefix, with v_j^1 being the first slow or steady node on that path. Let $v^2 = v_{j-1}^1$, and consider *its* most preferred path with a consistent fast suffix. Continue this process until reaching the sequence of v^js first loops, producing the first pair $v^a = v^b$, for $b < a$. The loop is guaranteed to happen since the node set is finite. If $a > b+1$, the loop has 2 or more nodes. Consider then each step in the resulting loop of v^j's. The suffix of P^j that starts at v^{j+1} is available to v^{j+1}, and has an (empty) consistent fast prefix, but v^{j+1} prefers a different path P^{j+1}. These suffixes thus form the spoke paths of a dispute wheel, with the matching prefixes forming the rims, violating the NDW condition.

Thus, $a = b+1$, that is, there is a node v^a which, with all the nodes in $S \cup H_i$ having selected their paths, most prefers a path that goes directly to a node in $S \cup H_i$, and has a consistent fast prefix, thus allowing us to add v^a to S to complete the induction. ∎

Proof of Theorem 4: Denote by m_i the number of edges with an endpoint in G_i, ensuring $\sum_i m_i \leq 2m$; we disregard the degenerate case of $m_i = 0$. The global bounds ensure at most n fair phases before convergence, so the slowest nodes, those in G_v, will each make at most $n = |G_{1,v}|$ announcements on each of their edges, yielding at most $m_v \cdot |G_{1,v}| + 1$ v-intervals. Within each $i+1$-interval, at most $m_i \cdot |G_{1,i}|$ announcements by the G_i nodes will yield at most that many, plus 1 i-intervals. For $v \geq 2$, by induction, the number of route announcements, upper bounded by the number of 1-intervals, is:

$$\hat{R} \leq \prod_{i=1}^{v} (1 + m_i |G_{1,i}|) \leq (n+1)^v \prod_{i=1}^{v} m_i$$
$$\leq (n+1)^v \left(\frac{2m}{v}\right)^v \leq \left(\frac{n}{v^{1/3}}\right)^{O(v)}$$

For the remaining case of $v = 1$, the claim comes directly from Corollary 1. ∎

What about convergence behavior in diverse Gao-Rexford networks? In completely homogeneous networks, with $v = r = 1$, Corollary 1 guarantees that $\hat{R} = O(\alpha m)$. This is sensible: it represents an average load of $O(\alpha)$ routing messages on any given edge (in regard to routes to any given destination d), a very moderate bound. However, with more diversity, just carrying over the proof from Theorem 4 yields a bound of $\hat{R} \leq (\alpha m / v)^{O(v)}$. This threatens to disrupt one of the most optimistic previous bounds known on BGP convergence. Could it be that, with any amount of diversity, Gao-Rexford networks can have average control-plane traffic *per edge, per-destination* scale with m, not just with α as in Theorem 3? BGP is already infamous for scaling linearly with the number of destination prefixes, but such a scenario would create a threat of *superlinear control-plane load growth* as the network scales. Or is there a tighter upper bound that prevents this?

The following result gives some concrete cause for worry, at least in the so-called dirty phase model (defined in Section II):

Theorem 5: For a fixed α and v, there are Gao-Rexford networks of arbitrary size n where per-edge average number of routing messages until convergence can scale as $n^{\Omega(\min\{\alpha, v\})}$, when converging in response to a network event occurring while some MRAI timers are still running.

Proof sketch: The counter-example is the Sharktooth gadget in Fig. 4, with a fixed number, $k = \min\{2\alpha - 2, v - 1\}$, of "tooth columns" ($k = 3$ above), all of which grow linearly with n grows. Let $t_i = t_{i+1} \cdot n/k$. If, when the leftmost 1

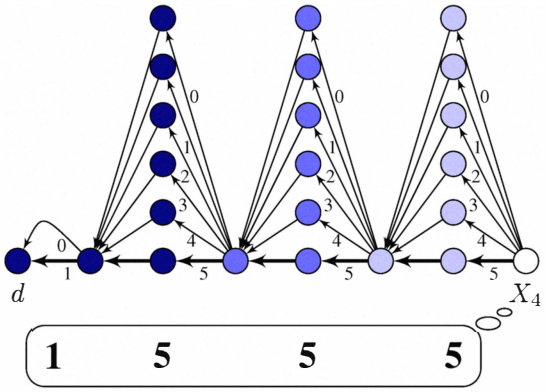


Fig. 4. Sharktooth gadget: per-edge average number of routing messages scales with n in a dirty phase setting.

edge fails, each tooth column has its teeth's timers spread out evenly, with the MRAI of tooth j in column i expiring in jt_i/k seconds, we can show that all possible routes ending in the 1 edge will be explored, while the MRAIs remain "evenly out of phase". ■

The dirty phase model is admittedly not as strong for modeling the response to a single event, since it requires some recent previous event to have set the MRAI timers in motion. On the other hand, it is more reasonable in the setting of per-edge MRAIs, addressed in Section V. Also, no upper-bound techniques we have found thus far succeed at differentiating worst-case dirty-phase behavior from worst-case clean-phase behavior, further indicating that this problem may indeed extend to clean-phase counterexamples. We leave this question as one of the important open problems of this work.

Lastly, in our discussion of MRAI diversity, we have thus far omitted its impact on convergence time. In a sense, convergence time, too, grows "exponentially" with v: the Chain gadget has $\hat{T} = t_* \cdot (n/v)^{\Omega(v)}$. But the exponential dependence on v in terms of the *fastest* MRAI is not as relevant to the practical concern of incrementally deploying *speedups* of MRAIs. As a function of "unimproved", slowest MRAIs, convergence time is still linear with no exponential dependence on v. Thus, the conclusion here is that, with incrementally deployed diversification of MRAIs, convergence time won't necessarily get *better*, while the other side of the MRAI tradeoff, control- and data-plane activity, can get exponentially worse.

C. From artificial gadgets to real networks

There is no doubt that the gadgets above are not typical network designs. As with any worst-case analysis, we need to consider whether the worst case behavior is limited to artificial situations, with "reasonable" networks exhibiting none of the worst-case problems. Since our goal is to capture the asymptotic worst-case problems that may arise as the network evolves, we should not fix our attention on a particular measured network graph, or a particular synthetic network model, and we must instead consider what *properties* all realistic networks, current and future, are expected to have.

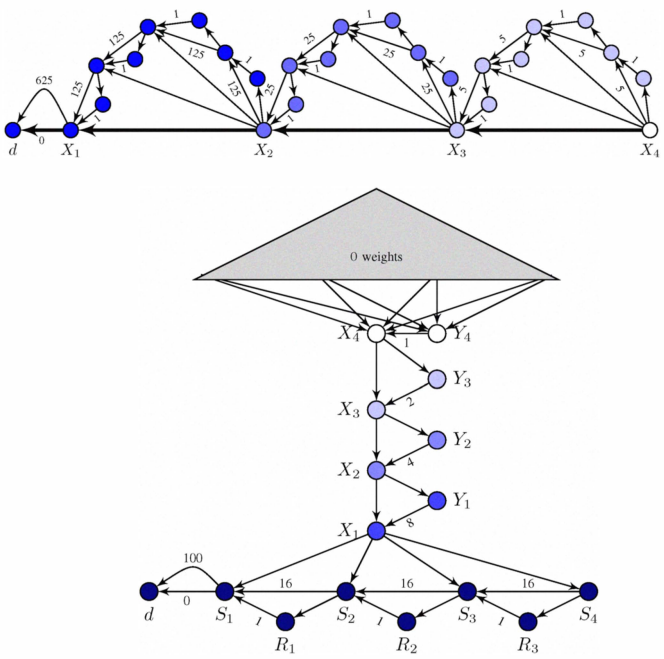


Fig. 5. Weights for the Christmas Tree and Chain gadgets.

The most prominent potential culprit for the "artificiality" of both of the above gadgets is the "counting in k-ary" preference function required for the combinatorial path exploration. But, it turns out, this exotic-sounding preference function is isomorphic to a number of rather reasonable BGP preferences, involving each node *optimizing a quantity similar to weighted shortest path over some globally-known measure*. As discussed in [21], this family of "semi-ring" preference functions can correspond to a number of relevant network phenomena: optimizing expected latency, optimizing packet loss rates, etc. Formally:

Theorem 6: The local preferences of all nodes in the Christmas Tree and Chain gadgets as shown in Theorems 2 and 3 are identical to an environment where each node's local preference over paths optimize a combination of per-edge quantities that form a semi-ring.

Proof sketch: The relevant weights are shown in Figure 5 (unspecified weights are zero). These can be thought of as, e.g., per-edge latencies, the sum of which each node tries to optimize, but this construction is also isomorphic to other policies that optimize over semi-rings. See [21] for a wide range of common examples. ■

Even though the preference functions are reasonable, there are other atypical features that enable the exponential path exploration: the paths explored contain many hops and the preference function is very "fine-grained", i.e. some nodes make very fine distinctions among a very large set of options. However, neither of these features prevents exponentially bad dependence on v:

Theorem 7: Even a GR network with only a constant number of allowed paths, all of constant length can produce an exponential number of routing updates and forwarding changes as a function of v before convergence, in terms of both n and α. That is, $\hat{F}$ and $\hat{R}$ are both lower bounded by $2^{\Omega(v)}$, where

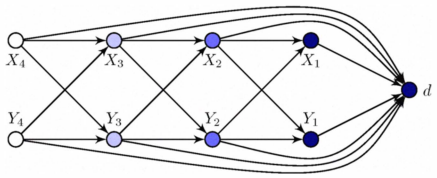


Fig. 6. A "Braid" gadget for $v = 4$. Each node has only 4 permitted paths, all of length 3, but the network can still generate exponential control and data plane activity.

$v = \Theta(n) = \Theta(\alpha)$.

Proof sketch: We form the Braid gadget (shown in Figure 6) by linking v copies of a cross-like subnetwork into a chain. For each permitted path at node X_i (and Y_i), its next-hop is either X_{i-1} or Y_{i-1}, its "next-next-hop" is either X_{i-2} or Y_{i-2}, and its final hop is to the destination, for a total of 4 permitted paths, all of length 3. With the arrows set to point from, e.g., providers to customers, the gadget obeys the Gao-Rexford constraints, with $\alpha = \Theta(n)$. To accommodate the update sequences we describe below, each pair (X_i, Y_i) has a different MRAI timer value, and thus $v = \Theta(n) = \Theta(\alpha)$.

The exponential number of routing updates and forwarding changes that can occur in the Braid gadget are the result of a particular sequence of updates that exploits an "amplification" property of the network. More precisely, there is a sequence of updates at each pair of nodes (X_i, Y_i) such that, whenever the sequence occurs, it causes the sequence of updates for the pair (X_{i+1}, Y_{i+1}) to occur *twice* consecutively, resulting in $2^{\Omega(v)}$ total updates throughout the network. Moreover, each update actually changes some node's next-hop, so both control and data plane activity are exponential in v. The Sawtooth and Chain gadgets each have a similar amplification property, but in those gadgets the sequence of updates could be described by the incrementing of a k-ary number. The situation for the Braid gadget is more complicated, and is best described by introducing some specialized terminology.

We say that X_i is using a *straight* path if the next-hop of its current path is X_{i-1}, and say it is using a *cross* path if the next-hop is Y_{i-1}. We define straight and cross paths for Y_i similarly. A *braid* is a sequence of updates at a pair of nodes (X_i, Y_i) such that the path types for these nodes change according to the following pattern:

Time step	Path type	
	X_i	Y_i
t_1	straight	straight
t_2	cross	straight
t_3	cross	cross
t_4	cross	straight
t_5	straight	straight

These path changes, visualized, resemble the braiding of a rope or ponytail. The key to the proof is establishing the following fact: It is possible to assign preference functions to each node so that, whenever a braid occurs at the pair (X_i, Y_i), it generates a sequence of update messages which cause a braid to occur twice consecutively at the pair (X_{i+1}, Y_{i+1}).

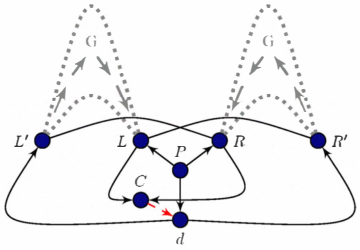


Fig. 7. The "Frog Eyes" gadget

There is one important complication that we have not addressed: The pair (X_2, Y_2) have no braids occurring "downstream" from their position, so to complete the gadget, we must insert a subnetwork at the rightmost end of the gadget, between the pair (X_1, Y_1) and the destination d, that causes a single braid to occur at (X_2, Y_2) upon the failure of some link to the destination. However, in the interests of a cleaner diagram, and because it adds little to the understanding of the gadget, the description of this subnetwork has been omitted. ■

IV. Slow forwarding convergence

A. Long-lasting blackholes

Forwarding changes can have a substantial impact on the system by causing out-of-order packet delivery, and sometimes introducing transient packet drops. However, the worst-case scenario for the data plane while the system is converging in response to a network event is far worse:

Theorem 8: After a network change, even in a Gao-Rexford network, even if no node ever becomes physically disconnected from d, there exists a network which may still forward a node's traffic to a black hole for the entire worst possible duration of control-plane convergence, $\Theta(nrt_*)$.

Proof sketch: The gadget in Fig. 7 is actually more general. It allows us to transform *any* gadget G like the ones above[1], and create black holes that last as long as the worst-case control-plane convergence in G does.

Arrows point from providers to customers, and edges without arrows are peer arrows, in the Gao-Rexford sense [11]. G permits a path from L' to L and from R' to R that follows provider-to-customer links only. Thus, at first, L' and R' route through their copies of G, and down through L and R, then through C. If the (C, D) link goes down, L and R no longer have a customer route, and prefer to take the peer route to R' and L' respectively, over their provider routes via P or d. After that, L' and R' will perform whatever path exploration G creates before exhausting their options for customer routes, and switching to a provider-learned route, which they do not export over their peer links to R and L, forcing the latter to take a provider route. For as long as G's path exploration is ongoing, L forwards packets to R', R', via G, to R, R to L', and L', via G, to L, creating a black hole.

[1]Specifically, any Gao-Rexford compliant gadget where (1) worst-case control-plane convergence explores paths of provider-to-customer links only, and (2) the worst-case behavior happens in response to a change in the last hop, with the new option last hop less preferred than any option involving the previous last hop.

Inserting the "Christmas Tree base" pattern for G, for instance, yields a blackhole lasting for $\Theta(nrt_*) = \Theta(\alpha rt_*)$ time. ■

B. Forwarding plane never much better than control plane

Recall that a routing update implies that a node has changed its path, while a forwarding change implies something stronger: a node has actually switched its next-hop. Since only forwarding changes impact the data plane, exponential lower bounds on the number of routing updates in a network are not necessarily cause for concern. However, all of the exponential bounds in the previous sections applied to both routing updates and forwarding changes. Was this a coincidence, or indicative of a general phenomenon? The next theorem shows that the number of routing updates and forwarding changes are indeed tightly related, and can never be exponentially far apart; specifically, they are always within a factor of n^2 of each other. Thus any exponential characterization of control plane convergence (a bound on $\hat{R}$) also exponentially bounds forwarding plane convergence ($\hat{F}$), and vice-versa.

Theorem 9: In a network with n nodes, $\hat{F} \leq \hat{R} \leq n^2\hat{F}$.

Proof: Since a routing update occurs whenever there is a forwarding change, we immediately have $\hat{F} \leq \hat{R}$. Now define a *frozen phase* to be any sequence of routing updates in which no node changes its next-hop. We will prove that no node can change paths more than n times during a frozen phase. Since there are n nodes, this implies the theorem.

Fix a frozen phase, and consider a node v_1. Let F be the longest acyclic path $(v_1, v_2, \cdots, v_k)$ such that v_{i+1} is the next-hop of v_i during the phase, for all i. So F is (a prefix of) the forwarding path of packets that originate at v_1. Because F is acyclic, the length of F is at most n. Define the *agreement length* of F to be the length of the longest prefix F' of F such that the path used by each node v on F' is a suffix of some common path P. For example, after the first time v_1 changes paths during the phase, since we know v_1 and v_2 do not change their next-hops during the phase, the agreement length of F will be at least 2. In general, every time v_1 changes paths during the phase, the agreement length of F increases by at least 1. And the agreement length of F cannot exceed n. ■

C. Restricted policies do not help

One approach to limiting forwarding changes is to place global restrictions on the policies that nodes can use to select and filter routes, even beyond what is needed to ensure BGP safety. One commonly studied type of restriction is to demand that each node use a *next-hop* import policy, i.e. each node's preference for a path depends only on the path's next-hop. In limited cases, this can help: During instabilities in which there are only path announcements, and if all nodes have a next-hop import policy, the number of forwarding changes is upper bounded by a polynomial in n [22].

However, during instabilities in which there are path withdrawals, the situation is essentially hopeless: There can be exponentially many forwarding changes even if every node uses a *shortest-path-first* rule [17] to select routes. In fact, we can extend this result to the case where every node uses a next-hop import policy. The proof, which is omitted for lack of space, is another example of our "amplification" technique for leveraging MRAI diversity to construct exponentially long update sequences, but applied to the "Trapezoid" gadget described in [17].

V. Discussion

Mitigation recommendations: The results above clearly show that network-wide consistency in MRAIs significantly improves the worst-case guarantees we can make about BGP convergence. We believe that our **theoretical worst-case examples are worth considering as a practical matter**. This is *not* because we expect these structures to appear verbatim in practice, but because they may suggest plausible patterns in the network that could substantially degrade performance, even if the real network never approaches the true worst case. To that extent, we believe it worthwhile to evaluate the counterexamples in detailed simulation to measure the likelihood of the worst-case chain of events. Also, we have yet to identify *network properties that curtail bad convergence, **and** are characteristic of the real Internet, but not our counterexamples*. The search for such properties would benefit from examining worst-case examples, with an eye to evaluating the realism of proposed properties via measurement studies.

We agree that the status quo of MRAI defaults is untenable, since the current default values are clearly large enough to cause substantial convergence delays. The deployment of any MRAI changes will necessarily be incremental, but, in light of Theorems 4 and 5, **we recommend** *updating the recommended MRAI default, rather than de-standardizing it completely*. If router vendors and operators do not substantially deviate from the new recommended default, the difference in worst-case convergence behavior may be like the difference between, e.g., quadratic scaling of path exploration versus exponential scaling, or perhaps, even more dramatically, the difference between BGP message loads scaling as a function of customer-provider hierachy height (≈ 5), versus scaling linearly or worse as a function of the network size (≈ 40000).

More practical is the question of **MRAI settings within a single autonomous system**, or multiple ASes run by the same institution. In regard to convergence, the iBGP route reflector hierarchy effectively mirrors the behavior of Gao-Rexford eBGP systems [13], [14], so we fully expect that the same bounds would apply to route information propagating through an iBGP system with route reflectors. But a single AS, unlike the IETF, can indeed enforce homogeneous timer settings by "flag day" changes within its network, and our results are thus a strong and practical recommendation that any single AS update its MRAI settings homogeneously.

We have thus far cast the motivations in terms of default MRAI settings. Anecdotal accounts from operators suggest that, indeed, most operators do not alter the vendor-specified default timer settings, and the small number of popular router vendors thus roughly upper-bounds the diversity of timer settings. We thus leave open the consequently relevant question

of what can happen to convergence if a just a **small fraction of individual router operators change the timers**, while the vast majority of the network retains the defaults. Conversely, also open is **whether a small group of routers or ISPs can collaboratively set their timings to improve network-wide behavior**, assuming the collaborators are in positions of influence in the network (e.g., some or all Tier 1s).

Limitations: Our analysis pertains to MRAI timers as recommended in the RFC [1]: a separate timer for announcements about each destination prefix over each BGP session with a neighbor. In practice, many implementations only permit **cruder MRAI timers, which apply to *any* announcements sent to a particular next hop**. This means that the routing convergence processes for two different destinations are no longer independent, with slow convergence for one destination producing MRAI delays for all the others. The single-destination, dirty-phase model, as used in Theorem 5 is a solid starting point for analyzing how two destinations' convergence processes may interfere with each other in the worst case. But it's entirely possible that even more dire worst case behavior will arise from fully general interactions between convergence processes to several destinations. As noted in Section II, we also do not treat the corner case of **MRAI being set to zero**, as currently done by default by some vendors, or MRAI set to be faster than the time it takes for incoming updates to be processed. The convergence in that case can be modeled by asynchronous results like [15], [17], or, worse, if updates can be released mid-processing, the system may engage in qualitatively different long-term behaviors, treated in [20].

We also forego entirely the question of **jitter**, the required randomization of MRAI values by each router [1]. The full paper shows that all of our upper bounds remain intact under jitter. The lower-bound counter examples also remain valid, in that stochastic jitter *could* end up be effectively equal at all routers, simulating a jitter-less network. That is exponentially improbable, but we expect that a stochastic analysis of our worst-case examples will retain the asymptotic behavior, even in expectation. We conjecture that, roughly, all our counterexamples (except Sharktooth) require at most "one direction" of exact timing (if jitter causes event A to occur before B, the counterexample may "skip a beat", but if B is before A, it will perform as above); and that combining these effects will at most halve the exponent of the exponential results, but confirming this rigorously is an open question.

Why MRAI?: Lastly, a worthwhile question to consider is, **why MRAI**? This timer was introduced to enable a practicable and desirable tradeoff, allowing operators to lower control-plane load by potentially reducing convergence time. But MRAI is clearly a simple and crude way to implement this tradeoff, with a single timer value for a node, or for an edge, with no regard to what update is being delayed by it. There has been some implementation and simulation work proposing various more sophisticated alternatives [23], [24]. We believe that some of our negative results may well be a peculiar artifact of MRAI timers as such, not a general property of all tools that enable the control-plane load vs time tradeoff. We thus think that, before deploying any of the new proposals, it's worth evaluating them in the worst-case framework we presented, starting with the many badly-behaved examples above.

It may well be that even a simple adjustment to **make MRAI timers adaptive**, based on the path being announced, may mitigate much of the worst-case convergence problems. We note in particular that all of our examples of bad behavior require nodes further from the destination to be updating much faster than those that are closer. If each node was aware of the timer settings of others on the path, it could resolve to not update faster than any node ahead of it on the path being announced. We conjecture that such an approach may yield a polynomial upper bound on control and data plane activity.

REFERENCES

[1] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)." RFC 4271, January 2006.

[2] N. Kushman, S. Kandula, and D. Katabi, "Can you hear me now?!: It must be BGP," *ACM SIGCOMM CCR*, vol. 37, no. 2, pp. 75–84, 2007.

[3] F. Wang, Z. M. Mao, J. Wang, L. Gao, and R. Bush, "A measurement study on the impact of routing events on end-to-end Internet path performance," in *ACM SIGCOMM*, pp. 375–386, September 2006.

[4] D. Pei, L. Wang, D. Massey, S. F. Wu, and L. Zhang, "A study of packet delivery performance during routing convergence," in *International Conference on Dependable Systems and Networks*, pp. 183–192, 2003.

[5] P. Jakma, "Revisions to the BGP 'Minimum Route Advertisement Interval'." Internet Draft `draft-ietf-idr-mrai-dep-02`, 2010.

[6] Juniper, "Out-delay." https://www.juniper.net/techpubs/software/junos/junos57/swconfig57-routing/html/bgp-summary32.html . Accessed 2010-07-30.

[7] T. G. Griffin and B. J. Premore, "An experimental analysis of BGP convergence time," in *Proceedings of ICNP*, 2001.

[8] K. Varadhan, R. Govindan, and D. Estrin, "Persistent route oscillations in inter-domain routing," *Computer Networks*, vol. 32(1), pp. 1–16, 2000.

[9] T. G. Griffin and G. Wilfong, "An analysis of BGP convergence properties," in *ACM SIGCOMM*, pp. 277–288, September 1999.

[10] T. G. Griffin, F. B. Shepherd, and G. Wilfong, "The stable paths problem and interdomain routing," *Trans. Netw.*, vol. 10(2), pp. 232–243, 2002.

[11] L. Gao and J. Rexford, "Stable Internet routing without global coordination," *IEEE/ACM Trans. on Networking*, vol. 9(6), pp. 681–692, 2001.

[12] A. Fabrikant and C. H. Papadimitriou, "The complexity of game dynamics: BGP oscillations, sink equilibria, and beyond," in *ACM-SIAM Symposium on Discrete Algorithms*, pp. 844–853, 2008.

[13] T. G. Griffin and G. Wilfong, "On the correctness of IBGP configuration," in *ACM SIGCOMM*, vol. 32, pp. 17–29, August 2002.

[14] A. Basu, C.-H. L. Ong, A. Rasala, F. B. Shepherd, and G. Wilfong, "Route oscillations in I-BGP with route reflection," in *ACM SIGCOMM*, pp. 235–247, August 2002.

[15] R. Sami, M. Schapira, and A. Zohar, "Searching for stability in inter-domain routing," in *IEEE INFOCOM*, pp. 549–557, April 2009.

[16] D. Obradovic, "Real-time model and convergence time of BGP," in *Proc. of INFOCOM*, vol. 2, pp. 893–901, 2002.

[17] H. J. Karloff, "On the convergence time of a path-vector protocol," in *ACM-SIAM Symposium on Distributed Algorithms*, pp. 605–614, 2004.

[18] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed Internet routing convergence," *Trans. Netw.*, vol. 9(3), pp. 293–306, 2001.

[19] C. Labovitz, A. Ahuja, R. Wattenhofer, and V. Srinivasan, "The impact of internet policy and topology on delayed routing convergence," in *INFOCOM*, pp. 537–546, 2001.

[20] M. Suchara, A. Fabrikant, and J. Rexford, "BGP safety with spurious updates." In submission.

[21] T. G. Griffin, "The stratified shortest-paths problem," in *Proceedings of COMSNETS*, pp. 1–10, 2010.

[22] Anonymous, "Putting BGP on the right path: A case for next-hop routing." In submission.

[23] A. Sahoo, K. Kant, and P. Mohapatra, "Improving BGP convergence delay for large-scale failures," *In Proc. of DSN*, pp. 323–332, 2006.

[24] A. Lambert, M.-O. Buob, and S. Uhlig, "Improving internet-wide routing protocols convergence with MRPC timers," in *CoNEXT'09*, pp. 325–336.