

Effect of MRAI Timers and Routing Policies on BGP Convergence Times

Rajvir Gill, Ravinder Paul, and Ljiljana Trajković

Simon Fraser University
Vancouver, British Columbia
Canada

e-mail: {rajvirg, rpa28, ljilja}@sfu.ca

Abstract—The Minimal Route Advertisement Interval (MRAI) plays a prominent role in convergence of the Border Gateway Protocol (BGP). Previous studies have suggested using adaptive MRAI and reusable timers to reduce the BGP convergence time. The adaptive MRAI timers perform well in the case of normal load of BGP updates. However, a large number of BGP updates may flood the Internet routers. We propose a new algorithm called MRAI with Flexible Load Dispersing (FLD-MRAI) that reduces the router's overhead by dispersing the load in case of a large number of BGP updates. We examine the MRAI timers in the case of normal load of BGP updates. Since BGP routing policies play a significant role in preserving the Internet routing stability, we evaluate their impact on BGP convergence time and Route Flap Damping (RFD) algorithms. The proposed algorithm and policies are evaluated using the ns-BGP network simulator.

Keywords—Communication networks, routing protocols, BGP, MRAI, RFD, routing policies.

I. INTRODUCTION

Among routing protocols, the Border Gateway Protocol (BGP) is the only protocol that operates in a network of the Internet's size. BGP provides a set of mechanisms for supporting Classless Inter-Domain Routing (CIDR). It supports any policy conforming to the “hop-by-hop” paradigm [1].

Minimal Route Advertisement Interval (MRAI) is the interval limitation that affects BGP convergence. Its default value is 30 s, which is efficient for a variety of network topologies and under many network conditions [1]. Continuous MRAI timers control the MRAI value and may be per-destination or per-peer timers. Each network destination is associated with one per-destination MRAI timer that independently limits advertisements to various destinations. However, per-destination MRAI timers may not be used because of the Internet size. Each peer in the network is associated with one per-peer MRAI timer. The timer starts ticking when the source router sends a route advertisement to peers. Per-peer MRAI timers adversely affect advertisements to each destination. For example, if an advertisement establishes a connection relying on the per-peer MRAI timer of a different AS, all subsequent advertisements sent to that AS would be delayed. Optimal MRAI values depend on the network size, topology, traffic volume, and network conditions [2].

The processing delay of an update performed by a BGP router significantly affects BGP convergence time. This is the total time of an update waiting in the queue and the time required for a BGP router to process it. Most proposed

solutions use the uniform processing delay for evaluating BGP convergence time [3]–[5]. They assume that a BGP router processes update messages sequentially one by one. If an update message is being processed, the update message that follows has to wait in the queue. Hence, the delay in processing updates affects the processing time of update messages that follow. Measurements show that the majority of update messages are processed within 200 ms. The average processing time for updates is 101 ms with the upper bound of 400 ms [6].

We introduce a new algorithm called MRAI with Flexible Load Dispersing (FLD-MRAI) that limits MRAI intervals based on advertisement events that occur in the network. We also propose modifications to reusable timers and changes to MRAI durations based on BGP advertisement events. An original BGP router always prefers the local shortest path as the degree of preference (DoP). We do not consider routing policies and we assume that each Autonomous System (AS) contains only one BGP router [3]. We propose the FLD-MRAI algorithm for peer-to-peer networking in heterogeneous and large networks. FLD-MRAI also performs well in networks where the traffic load is unspecified.

BGP relies on a variety of routing policies that affect its performance. Persistent route fluctuations may be triggered with or without routing policies [7]. Route Flap Damping (RFD) may reduce processing load caused by network instability without increasing route convergence time. It may also prevent BGP route oscillations caused by network instabilities such as router configuration errors, transient data link failures, incorrect implementations of a router policy, and software defects. A change in route attributes reflects a flap [8]. RFD may be configured based on a number of parameters and assigns a penalty to each route. Whenever the route flap occurs, the penalty increases and if its value exceeds the *threshold suppress limit*, the route advertisement is suppressed. The parameter *half-life* specifies the time interval during which the penalty is to be reduced by half. A route may be reused and may be advertised again when route penalty is lower than the *reuse limit* [8].

BGP allows a user to configure peer-to-peer policies. There is no centralized control over the policy configurations chosen by different organizations and no global parameters defined to configure these routing policies. This may cause persistent route oscillations that affect the BGP stability and convergence time [7].

In this paper, we evaluate the effect of routing policies on BGP convergence time, number of updates, and number of flaps. We implement in ns-2.34 a routing policy module

called BGP-RP with two routing policies: AS path list and community list.

This paper is organized as follows. In Section II, we provide literature survey. We describe the FLD-MRAI algorithm in Section III. The implementation of FLD-MRAI and BGP-RP is described in Section IV. Their performance evaluation is described in Section V. We conclude with Section VI.

II. BACKGROUND ON BGP CONVERGENCE

A single reusable MRAI timer for all route advertisements sent during a short time interval has been proposed in the past [2], [3]. This interval defines the granularity of an MRAI round and the total number of reusable MRAI timers. The update messages may be divided into higher-priority and lower-priority classes [9]. The updates in the higher priority class advertise the routes faster than the lower priority class. A global timer is used to reduce overhead. The receiver classifies the update messages based on the per-destination forwarding-path tree. According to the priority class, on-tree updates are processed faster from the receiver's perspective. Consequently, the sender has to infer the priority class of updates (higher or lower) and may experience additional overhead.

Networks with routers that have heterogeneous MRAI timers may experience significantly worse routing convergence and exponential increase in the number of BGP update messages [10]. The adaptive MRAI timers based on the announced paths have been recommended. These improved MRAI timers decrease the BGP convergence time and guarantees network stability [11]. Experiments have shown that setting keepalive timer to 10 s and hold timer to 15 s reduces BGP convergence time. The path exploration damping (PED) algorithm proposes timer of 35 s, which may reduce the number of update messages and convergence time and may be a viable alternative to default MRAI timers [12]. Several artifacts in BGP message handling procedures that may cause superfluous invocations of the MRAI timer during the route selection process have been identified [13]. These changes to route establishment process do not adversely affect the processing of the MRAI timer and result in faster BGP convergence. The delay due to convergence limits of BGP may also be examined based on the power laws of the Internet topology and the BGP protocol standards [14], [15]. These reports also show that processing efficiency of router's CPU and the value of MRAI timer significantly affect the BGP convergence time.

The BGP model that considers convergence properties, number of updates, and effects of routing policy scenarios has been reported [16], [17]. It was illustrated via SSFNET simulations that the sender side loop detection (SSLD) and optimal values for MRAI reduce the BGP convergence time [3]. For each specific network topology, there is an optimal MRAI value that reduces the BGP convergence time [18]. Furthermore, the routing instability increases BGP convergence time, number of updates, and packet loss. Moreover, higher levels of instability sometimes cause loss of the internal large area connectivity. An earlier study [19] developed a model that provided theoretical upper and lower

bounds of the convergence time in case of both path and BGP router failures. Measurements demonstrated that latency due to router or link failure might reach tens of minutes while delay due to failure of multi-homed networks may last as long as fifteen minutes after a network fault. SSLD detects loops in the path and after their removal only paths without loops are announced. Simulation results also show that modified MRAI that performs SSLD converges within 30 s.

When the shortest path to destination becomes available, the network converges more quickly than when path or BGP router fails [19], [20]. This is because Internet Protocol version 4 (IPv4) BGP has a large topology diameter that consists of thousands of ASes. The MRAI delay of 30 s increases the update transmission time and it is constant for every advertisement event in the exterior Border Gateway Protocol (eBGP).

The load-balancing algorithm is employed in the homogenous client-server architecture where data relocation takes place when local node has no available CPU to execute processes [21]. The effect of BGP processes on active routers was analyzed in the Sprint Internet Protocol (IP) network [22]. It was shown that BGP processes utilize 60% of a router's CPU time during active CPU cycles. BGP processes consume the maximum CPU utilization during short intervals (5 s). The large number of messages during a CPU cycle may increase CPU load, which increases BGP convergence time and affects router stability. A router's CPU load depends on the number of BGP messages received during a specific MRAI round. A router receives large number of update messages due to the Internet size. Large number of update messages increases the size of BGP routing table, which may require large memory and CPU utilization [23]. Hence, memory and CPU utilization are the essential requirements for a BGP router to successfully send information to all other BGP peers in the network. A high CPU utilization of a BGP router also causes additional queuing delays within BGP convergence period [24].

The selective RFD mechanism may suppress the withdrawn routes for up to one hour [25]. Network topologies affect the BGP convergence time. The increase in the maximum *suppresses threshold time*, decreases the number of flaps and updates [26]. The RFD+RG algorithm performs route flap damping without losing reachability of routes and may be employed with any RFD algorithm [27]. Furthermore, persistent route oscillation may exist with some policy configurations that may cause BGP instability [7]. The BGP route oscillations also occur with no configuration of BGP routing policies [28]. The mismatch in policy configurations between two routers may also cause network instabilities [29].

III. PROPOSED FLD-MRAI ALGORITHM

A. CPU Utilization and Processing Delay

In the proposed FLD-MRAI algorithm, we use an empirical value of 200 ms as the processing delay. The FLD-MRAI algorithm processes update messages within 200 ms rounds and it operates in the case of both normal and high

network loads. When DoP prefers the shortest path, then FLD-MRAI considers this scenario as a normal load. When DoP prefers a longer path in the presence of the shortest path based on certain conditions, then FLD-MRAI considers this scenario as a high load. We assume that all received updates are processed within one processing round. The source BGP router sends an advertisement of destination address D to neighboring BGP routers at time t_o . After advertising a destination, the source BGP router begins receiving path-updates from neighboring BGP routers. MRAI consists of two states: idle and processing. The source BGP router prioritizes the received updates and assigns the highest priority to the update with the shortest path. A critical factor in processing delay is to estimate CPU time needed for sending update messages. If one task demands higher CPU utilization, then the router dedicates fewer CPU cycles to other tasks. When a router's CPU utilization is high, then the router responds slowly to subsequent requests in the queue. The BGP router calculates available CPU of neighboring routers based on the priority of updates. Percentage of available CPU of the neighboring router is calculated as:

$$CPU_{available} = 100 - CPU_{active} \quad (1)$$

$$CPU_{active} = 100 * (CPU_{current}/CPU_{max}), \quad (2)$$

where $CPU_{available}$ is the percentage of available CPU of the neighboring router, CPU_{active} is the percentage of active CPU utilization of the neighboring router, $CPU_{current}$ is the current CPU utilization, and CPU_{max} is the maximum CPU utilization. The maximum CPU elapsed time is always equal or greater than the current CPU utilization time. If the router's queue is empty, then the maximum CPU elapsed time is equal to the current CPU utilization time.

At the beginning of the BGP decision process, a router calculates the DoP of each new, replaced, and/or withdrawn route [1]. The default DoP depends on:

- The local routes that originate from the local AS and have 100% value of LOCAL_PREF variable are given the highest priority. The route having the shortest path is called $Route_{info}$.
- The default DoP of a route is subject to the routing policies among the ASes. The ASes having the same routing policy are given the highest priority.

In the proposed FLD-MRAI algorithm, the BGP router calculates DoP based on the available percentage of CPU utilization. The modified DoP_{mod} is a function of $Route_{info}$ and $CPU_{available}$.

The implemented FLD-MRAI algorithm does not consider routing policies for calculating DoP. The routers with higher $CPU_{available}$ are given the highest priority. $CPU_{available}$ is calculated every time a router receives the updates of a new or withdrawn route. The default DoP changes every time a router receives the new or withdrawn route update [1]. When a BGP router sets priorities based on DoP, it always considers available CPU. A path with the highest value of DoP_{mod} is given the highest priority. After calculating available CPU, BGP router compares available

CPU of the neighboring BGP routers according to priorities based on the shortest path.

Suppose that R_1 and R_2 are neighboring BGP routers based on the first and the second priority paths, respectively. Assume that C_1 and C_2 are available CPU of BGP routers R_1 and R_2 , respectively. A default BGP router follows the DoP rule to always prefer the local shortest path to send data and, hence, it follows a path that includes R_1 , which belongs to the first priority path.

If C_1 is larger than C_2 , then FLD-MRAI detects this scenario as a normal load and follows the shortest path. In the normal load scenario, DoP remains unchanged and, hence, CPU utilization does not affect the computation of DoP.

If C_1 is smaller than C_2 , then FLD-MRAI calculates waiting time in the queue of R_1 and transmission time to R_2 from the BGP source router. If the waiting time is larger than the transmission time, then BGP source router checks DoP of path including R_2 is larger than the path including R_1 , then BGP router chooses a second priority path. Otherwise it switches back to the first priority path. If the load disperses to the longer path based on specified conditions, then the FLD-MRAI algorithm detects this scenario as a high load. FLD-MRAI may be also implemented in networks where the volume of traffic is unspecified. The two load cases and four advertisement events in the case of normal load are shown in Figure 1.

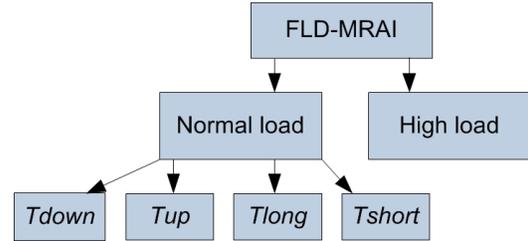


Figure 1. Flowchart of the FLD-MRAI algorithm.

To illustrate the difference between the FLD-MRAI and the default MRAI algorithms [1], we consider a simple network with five routers shown in Figure 2. Suppose that R_0 is the source router and that it advertises to destination R_2 . There are two possible paths: R_0 - R_1 - R_2 and R_0 - R_4 - R_3 - R_2 . The default BGP router chooses the preferred path R_0 - R_1 - R_2 without considering available CPU.

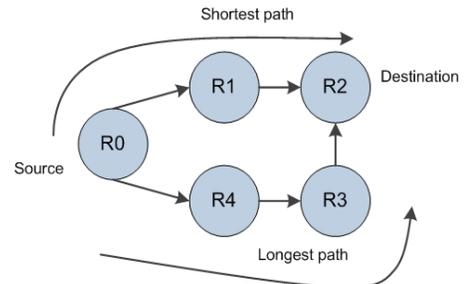


Figure 2. Example of the network with five routers.

If available CPU of R_1 is smaller than R_4 , then requests from R_0 will wait in the queue of R_1 . According to FLD-

MRAI, R_0 calculates available CPU of R_1 and R_4 . If available CPU of R_4 is smaller than R_1 , then FLD-MRAI detects this scenario as a normal load. If the available CPU of R_4 is larger than R_1 , then the algorithm calculates waiting time in the queue of R_1 and data transmission time to R_4 . If the waiting time is larger than the transmission time, then R_0 calculates the DoP of both paths. If DoP of path R_0 - R_4 - R_3 - R_2 is larger, then R_0 prefers to select R_4 and FLD-MRAI detects this scenario as a high load. The total processing time of one update message remains less than or equal to 200 ms.

B. Modified Reusable Timers

MRAI permits a BGP router to announce routes of a destination to its peers after one MRAI round. The optimal MRAI value is difficult to calculate. It depends on network topology and the active time of each MRAI interval, which depends on advertisement events. Instead of using default MRAI [1], we propose modifications of MRAI values based on advertisement events T_{down} , T_{up} , T_{short} , and T_{long} . MRAI timer depends on network conditions and advertisement events [19].

T_{down} event: The destination becomes unreachable after link failure and BGP router withdraws the previously existing path. All peers receive withdrawal announcements immediately without delay and source BGP router looks for a new route to the destination.

T_{up} event: If network recovers from a link failure during a T_{down} event, then the previously preferred path becomes available. This is called a T_{up} advertisement event. BGP router sends T_{up} advertisements to destination without further delay.

T_{long} event: After a BGP router failure, previously announced preferred path to destination becomes unavailable. After the failure of the shortest path, a longer path to the destination is advertised and this event is called T_{long} advertisement event.

T_{short} event: When a network receives an advertisement of a new shorter path compared to the current path, then the available shorter path changes the current path. If network recovers from the BGP router failure, then the shortest path replaces the longer path.

Percentage of occurrence of advertisement events is given in Table 1 [20].

TABLE 1. LIST OF EVENTS DURING BGP CONVERGENCE.

| Events | Number of events occurring during BGP convergence period |
|--------------|--|
| T_{down} | 43.4% |
| T_{up} | 39.9% |
| T_{long} | 7.3% |
| T_{short} | 7.4% |
| Unidentified | 2.0% |

Differentiation of advertisement events permits us to identify two categories and proposes two values of MRAI timers. FLD-MRAI computes the duration of MRAI rounds individually for each destination. After the processing time, three events may occur:

- new update message is received
- no new update is received

- MRAI reusable timer has expired.

The instance of the MRAI timer when BGP router actually processes the received update messages is active time. The remaining period is idle time. During each advertisement event, FLD-MRAI calculates idle time and enters the processing state. FLD-MRAI calculates idle time during initiation of a new round. FLD-MRAI chooses duration of MRAI round according to the computation of idle time. A long idle interval during previous MRAI round may indicate that the active interval is small and the update is advertised in shorter time than the default value. Similarly, a short idle interval may indicate that the active interval is longer than expected and the previous MRAI round should have lasted longer. The granularity of the MRAI timers is assigned as the threshold for defining the idle interval. The idle time is calculated as:

$$T_{idle}(D) = MRAI_{total} - M_{last}, \quad (3)$$

where $T_{idle}(D)$ is the idle time of the destination, $MRAI_{total}$ is the total MRAI interval, and M_{last} is the time instance of the last message received. We implement changes in reusable MRAI timers that independently limit advertisements of many destinations. The main advantage of a reusable timer is that only one reusable timer is required for all paths advertised during a short time interval. We propose specific durations of a reusable MRAI timer for different advertisement events. We also use a smaller number of reusable MRAI timers where each timer belongs to a definite time interval. The number of rounds per reusable MRAI timer controls the duration of MRAI round, which is calculated as:

$$MRAI_{duration} = R_n * (t_n * g), \quad (4)$$

where $MRAI_{duration}$ is the duration of MRAI round, R_n is the number of rounds per reusable MRAI timer, t_n is the number of reusable MRAI timers, and g is granularity.

The reusable MRAI timer is associated with each route advertisement sent. Timers need to store pointers and these pointers are required only for non-converged routes. Hence, overhead of storing pointers is minimal. When an MRAI timer expires, reusable timers keep a list of paths that need to be advertised.

C. Duration of MRAI

It is essential to analyze the duration of MRAI timers for BGP advertisement events [19], [20]. Since the BGP convergence time for T_{down} or T_{long} event is larger than for T_{up} or T_{short} event, we propose longer duration of MRAI timers for T_{down} and T_{long} events and smaller duration for T_{up} and T_{short} events. We propose the same MRAI value for T_{down} and T_{long} events and for T_{up} and T_{short} events. In case of FLD-MRAI with granularity of 1s, the proposed minimum duration is 15 s. The threshold of an idle period is 1 s.

If the BGP router detects idle time longer than 1 s, then it processes the received update message during the time interval well before the expiration of a timer. Two types of advertisement events may occur: T_{short} or T_{up} . In both cases, the previously announced shortest preferred route to the destination becomes available. T_{up} occurs after link

failure recovery while T_{short} occurs on either arrival of a new shortest path update or after recovery of BGP router failure. The BGP router sends updates to the destination in case of both advertisement events without further delay. Therefore, in case of these two events, we propose duration of MRAI round equal to 15 s. Pseudocode of the proposed FLD-MRAI algorithm is shown in Figure 3.

```

when sending advertisement of destination D to peers at  $t_o$ 
  set ( $S_i$ ) // priority numbers on received updates
  according to the shortest path
  if ( $C_1 < C_2$ ) // calculate and compare the
  available CPU utilization of 1st and 2nd
  priority neighboring router
  if  $W(t) < T(t)$  // calculate and compare waiting time
  and transmission time
  else (wait in queue of the 1st priority path)
  if  $dop_2 < dop_1$  // calculate and compare the degree of
  preference
  choose 2nd priority path
  MRAI = 30 s
  goto processing state
  else (wait in queue of the 1st priority path)
  else if ( $C_1 > C_2$ )
  wait in queue of the 1st priority path
  // duration of MRAI is based on the idle
  time
  goto processing state

when initiation of new round
  if ( $Idle(D) > 1$  s) //  $T_{short}$  or  $T_{up}$  may occur
  set modified_reusable timer = 15 s
  else if ( $e \in$  network failure) // event changes due to
  the network failure
  choose 2nd priority path // after expiration of the
  timer
  set modified_reusable timer = 30 s
  goto processing state
  else if ( $e \notin$  network failure)
  goto processing state
  else if ( $Idle(D) < 1$  s) //  $T_{long}$  or  $T_{down}$  may occur
  set modified_reusable timer = 30 s
  else if ( $P_t \in P_s$ ) // if the shortest path becomes
  available
  choose shortest path // after expiration of the timer
  set modified_reusable timer = 15 s
  goto processing state
  else ( $P_t \notin P_s$ ) // if the shortest path is not available
  goto processing state

```

Figure 3. Pseudocode of the proposed FLD-MRAI algorithm.

If the BGP router detects idle time shorter than 1 s, then BGP router processes the received update message during the time interval very close to the timer expiration. Two types of advertisement events may occur: T_{long} or T_{down} . In both cases, the previously announced shorter preferred route to the destination becomes unavailable. Hence, the BGP router withdraws the updates in both events and announces longer paths compared to the previously preferred paths. Thus, the duration of MRAI timer should be larger. We propose duration of MRAI round of 30 s. Hence, the BGP

router doubles the value of MRAI round for these events. The reusable timer automatically adjusts its duration to 30 s by using two MRAI rounds of 15 s without expiration of the reusable timer after the first 15 s round. After the expiration of the second round, FLD-MRAI assigns one round for the reusable timer (15 s). The maximum period of MRAI round is equal to the default MRAI value (30 s). If the shortest path becomes available during T_{long} or T_{down} event, then BGP router withdraws the current update. After the expiration of two rounds of reusable timer, BGP router chooses the shortest path with duration of one round (15 s).

If network conditions change due to path or BGP router failure during T_{up} or T_{short} , then the event changes to T_{down} or T_{long} . After the expiration of the reusable timer, the source BGP router chooses a second priority path and uses reusable timer twice. Hence, the proposed algorithm processes T_{long} and T_{down} updates with default duration of MRAI (30 s), while T_{up} and T_{short} use half of the default MRAI duration (15 s). The advertisements during T_{up} and T_{short} experience shorter delay and, hence, decrease the BGP convergence time.

The duration of MRAI round may be 15 s or 30 s, based on computation of the idle time. After the expiration of each reusable timer, the timer may be used again with a variable duration. We use 15 timers with granularity 1s and change value of the number of rounds (one or two). FLD-MRAI assigns two rounds for a reusable timer with the 30 s duration (T_{long} or T_{down}). If the BGP router does not receive update messages of the current destination within previously described MRAI period, then we assume that the routes have converged.

An illustrative example is shown in Figure 4 where each of the 15 reusable MRAI timers with granularity 1s takes one round during the time interval of 15 s. $Timer_0$ lasts one round of 15 s. After expiration, it is reused as $Timer_{15}$. If $Timer_2$ updates occur during T_{down} or T_{long} , then duration of MRAI round is set to 30 s. After expiration, $Timer_2$ is reused as $Timer_{32}$ whose duration depends on the idle time.

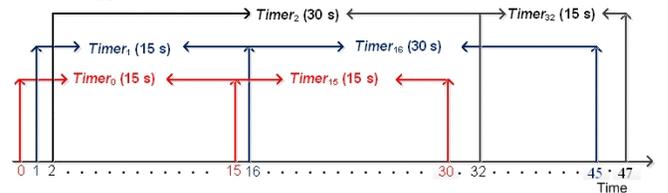


Figure 4. 15 reusable timers with MRAI rounds equal to 15 s or 30 s.

D. Space and Time Complexity of the FLD-MRAI Algorithm

The FLD-MRAI algorithm changes the routing path only when the best route to the destination is found. Hence, the algorithm depends on the number of non-converged routes n over a period of the MRAI interval. The implementation of the FLD-MRAI algorithm requires that the router keeps three variables of each non-converged route: $CPU_{current}$, CPU_{max} , and M_{last} . These variables are integer counters, which a router may easily store. The space complexity of the FLD-MRAI algorithm is $O(n)$.

The time complexity of the FLD-MRAI algorithm is the number of processes executed at the beginning of the MRAI round as shown in Figure 3. At the initiation of a new MRAI round, the algorithm recalculates the idle time for each new advertisement received.

The FLD-MRAI algorithm requires divisions, multiplications, and subtractions. The division and multiplication operations are used in calculation of $CPU_{available}$, T_{idle} , and $MRAI_{duration}$. The time complexity of the division and multiplication depends on n while the time complexity of the subtraction is constant. The maximum value of $MRAI_{duration}$ is 30 s. To simplify estimation of the time complexity, we approximate these variables with constants equal to their maximum values. This establishes the upper bound of time complexity and we may assume that divisions and multiplications used in calculation of variables do not depend on n . The computation of $CPU_{available}$ requires one subtraction, one multiplication, and one division (1). The computation of T_{idle} requires one subtraction while $MRAI_{duration}$ requires two multiplications. Hence, the time complexity of the computation of these variables is $O(n)$.

The BGP router may send only one advertisement and one withdrawal during a single MRAI round. The number of neighbors and non-converged routes during one MRAI round affects the maximum number of update messages. The time complexity of the computation of the idle time is $O(n)$ if the number of neighbors is constant.

IV. IMPLEMENTATION OF FLD-MRAI AND BGP-RP

We implement FLD-MRAI algorithm and routing policies using the ns-2.34 network simulator and the ns-BGP 2.0 [5] developed module, which was ported from the SSFNET simulator [3].

A. NS-2 Implementation

The BGP modifications are implemented in the existing ns-2.34 and ns-BGP 2.0 C++ class hierarchies. We realize

various network topologies using the OTcl class hierarchy. The routing structure of a modified ns-2 node consists of the forwarding plane and control plane, as shown in Figure 5. Forwarding plane consists of the address classifier (`classifier_`) that categorizes received packets whether to be processed or forwarded to neighboring nodes and the port classifier (`demux_`) that forwards packets to their destinations based on their port numbers. Control plane controls computation, maintenance, and implementation of routes in routing tables [5]. In an ns-2 node, the route object (`rtobject`) synchronizes several dynamic routing protocols. C++ class `PeerEntry` stores information about each peer connection and contains two routing tables: `AdjRIBIn` (Adjacent Routing Information Base, Incoming) and `AdjRIBOut` (Adjacent Routing Information Base, Outgoing). `AdjRIBIn` stores the Network Layer Reachability Information (NLRI) exchanged between BGP routers learned from a neighboring BGP. `AdjRIBOut` stores the NLRI exchanged between BGP routers, which are to be announced to a neighboring BGP. To implement the FLD-MRAI algorithm, we modify reusable timers, `DoP`, `rtProto/BGP`, and address classifier (`classifier_`). We also use an empirical value of processing delay, which is implemented in the input buffer (`inbuf_`). FLD-MRAI computes percentage of CPU utilization and MRAI rounds for each destination in `rtProto/BGP`. Furthermore, it computes the value of `DoP` for paths in the address classifier (`classifier_`). FLD-MRAI stores the update messages and forwards them to port classifier (`demux_`) after computing the CPU utilization of neighboring BGP routers. The expired reusable timer is used again in the BGP decision process that is associated with `rtProto/BGP`. All decisions in the BGP-RP implementation are made in the `rtproto/BGP` class. Policy decisions are made based on information obtained from `AdjRIBIn` and `AdjRIBOut`.

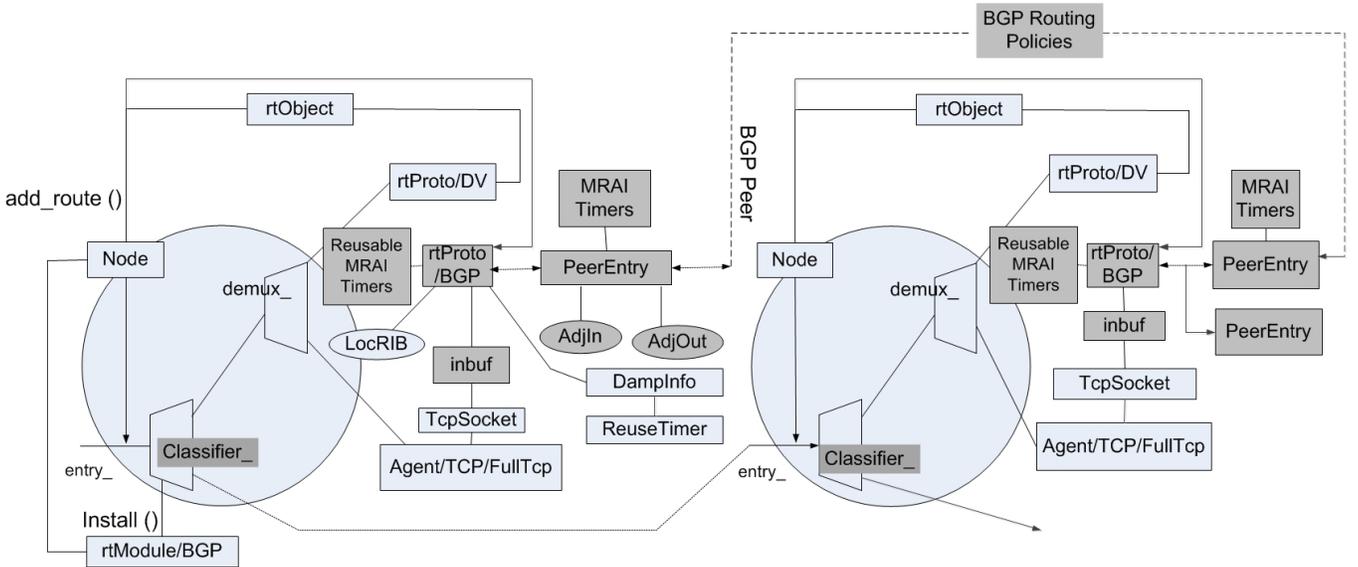


Figure 5. Implementation of the FLD-MRAI algorithm and BGP routing policies in the ns-BGP node.

B. Implementation of BGP-RP

BGP routing policies are very commonly used in deployed networks. BGP routers process paths according to the policies set by various Internet service providers (ISPs). These administrative routing policy decisions are driven by market conditions rather than network properties. Networks that have routing policies may permit or deny specific paths. Routing policies in networks increase convergence time, number of updates, and number of flaps. We examine the effect of RFD on the BGP convergence time by implementing the BGP routing policies in a module named BGP-RP. In addition to routing policy module, we also implement the AS path list and community list routing policies [30]. We configure these two routing policies by using regular expressions. The `tre-0.8.0` [31] library for regular expressions is implemented in `ns-2.34`.

Networks use the AS path list policy to filter the BGP AS path attributes, which define the entire set of AS numbers that have policy approval to reach any particular network. AS path attributes also help BGP to find the best path. We use a regular expression string to identify the attribute pattern to deny or permit the list. There are two types of community list policies: numbered or named. Both identify and filter the routes according to common attributes between two networks. If a network has multiple routing policies, then for inbound updates BGP-RP first processes the filter list that may deny access to any network. It then processes the route map and the type of policy list. For outbound, BGP-RP processes the type of policy list first, followed by filter list and route map. Pseudo code of a routing policy module is shown in Figure 6.

```

when receiving a route r with prefix d from neighbor
if (rouPolicy = 1)
    // routing policy of source matches with neighbor
update RIB
    // update the Routing Information Base
else if (rouPolicy = 0)
    // routing policy of source does not match with
neighbor
update RIB
    // update the Routing Information Base

```

Figure 6. Pseudocode of the routing policy module.

C. Simulation Scenarios

Performance evaluation of FLD-MRAI and BGP-RP is based on various network topologies. We compare performance of the FLD-MRAI algorithm when the duration of the MRAI timers is 30 s (FLD-MRAI-30) or 15 s (FLD-MRAI-15) for all advertisement events. We also compare the FLD-MRAI algorithm with default MRAI timers with duration of 30 s (default-MRAI-30) and 15 s (default-MRAI-15) and the adaptive MRAI [2]. We also analyze the results of AS path list and community list policies implemented in BGP-RP. Four parameters were considered when designing simulation scenarios: network size, network traffic, BGP events, and total simulation time. The simulation time depends on simulation parameters.

D. Simulated Topologies

We evaluate the proposed FLD-MRAI and BGP-RP using topologies derived from the BCNET BGP traffic collection [32], the Georgia Tech Internetwork Topology Models (GT-ITM) generator [33], and the Boston university Topology Representative Internet Topology generator (BRITE) [34]. Five network topologies are listed in Table 2

TABLE 2. NETWORK TOPOLOGIES USED IN SIMULATIONS.

| Topology | Number of nodes | Topology generator |
|------------|-----------------|---------------------------------|
| Topology 1 | 67 | Manually from BCNET BGP traffic |
| Topology 2 | 100 | GT-ITM |
| Topology 3 | 200 | GT-ITM |
| Topology 4 | 300 | BRITE |
| Topology 5 | 500 | BRITE |

Network Topology 1 consists of 67 nodes built manually from collection of BCNET BGP traffic [35], [36]. We examine routing tables of BCNET BGP traffic and analyze AS numbers and connections between them. The behavior of links was generated from the BCNET BGP traffic.

Network Topology 2 and Topology 3 were generated using the GT-ITM generator. GT-ITM generates topologies based on three models: flat random, N-level hierarchy, and transit-stub hierarchical. Topologies consisting of 100 and 200 nodes were generated using transit-stub hierarchy due to two reasons: the transit-stub model matches better today's Internet topology and it has a precise hierarchical configuration comparable to the Internet tiers that allow a provider to divide traffic into separate levels [37]. The ASes in a stub network have no information about ASes in other stub networks. The ASes exchange traffic between stub networks through transit ASes. They have smaller degree of connectivity compared to a transit AS. The GT-ITM generator initially creates a connected random graph for creating a transit-stub model topology where each node signifies a transit AS. Every node in the transit AS has a link with the stub AS. Network may be linked with any of the six graph connection methods: Doar-Leslie, Exponential, Waxman1, Waxman2, Locality, or PureRandom [33]. We generate topologies using the PureRandom method.

Network Topology 4 and Topology 5 were generated using the topology generator BRITE [34], which generates different types of Internet topologies from models that are intended to capture the Internet topology on AS, router, or LAN levels [2]. We generate AS-level topologies from the Generalized Linear Preference (GLP) model because it captures the power laws and the clustering behavior of the Internet [5].

E. Assumptions

We adopt several assumptions when analyzing the impact of FLD-MRAI on BGP convergence time and the overall number of update messages. The long-term instabilities such as route flaps are slower in suppressing a path, which may cause longer BGP convergence time. This

suppression time is much higher than the MRAI value. Hence, we do not consider route flap damping when evaluating the performance of FLD-MRAI. The impact of the interior Border Gateway Protocol (iBGP) does not affect the BGP convergence time because we assume that each AS consists of a single BGP router. We also assume that the BGP convergence procedure is complete if BGP router receives no update message from other BGP routers within 60 s.

V. PERFORMANCE EVALUATION

A. Network Topology 1

We consider two simulation scenarios: normal load and high load.

1) *FLD-MRAI with Normal Load Scenario*: In this scenario, we consider four cases: *Tshort*, *Tlong*, *Tup*, and *Tdown*. The source node begins sending traffic at 30 s and at 130 s. Simulation results are shown in Table 3 and Table 4.

Case 1: In case of *Tshort* event, majority of BGP routers with default-MRAI-30 require more than four MRAI rounds to find the best route. However, majority of BGP routers with FLD-MRAI require less than three MRAI rounds, resulting in an average BGP convergence time of approximately 67 s.

Case 2: In case of *Tlong* event, when the shorter path becomes unavailable, the current path is replaced with the longer path. Most BGP routers with default-MRAI-30 need approximately five MRAI rounds to find the best route. Conversely, most BGP routers with FLD-MRAI need approximately three MRAI rounds to find the best route, resulting in an average BGP convergence time of approximately 77 s.

Case 3: In case of *Tup* event, after some time an unreachable destination becomes available and some of the BGP routers first send update messages to non-optimal paths, which affects the BGP convergence time. Most BGP routers with default-MRAI-30 need more than four MRAI rounds to obtain the best route. However, majority of BGP routers with FLD-MRAI need approximately two MRAI rounds to obtain the best route, resulting in an average BGP convergence time of approximately 65 s.

Case 4: In case of *Tdown* event, reachable destination becomes unreachable and after the expiration of the current MRAI round, BGP router chooses another path. Most BGP

routers with default-MRAI-30 require approximately five MRAI rounds to obtain the best route. However, majority of BGP routers with FLD-MRAI require more than two MRAI rounds to get the best route, resulting in an average BGP convergence time of approximately 75 s.

The overall number of FLD-MRAI update messages for *Tshort* (374), *Tlong* (445), *Tup* (391), and *Tdown* (386) is smaller than for all other BGP options. On average, in all four cases, the proposed FLD-MRAI modifications help reduce the average convergence time by approximately 43% and number of update messages by approximately 40%.

2) *FLD-MRAI with High Load Scenario*: If the load disperses to a longer path due to DoP, then FLD-MRAI considers this scenario as a high load. According to default DoP, the source router will follow the shortest path even in the case of high load and the request will wait in the queue of the neighboring router. However, in case of DoP of FLD-MRAI, the source router follows the path having large available CPU. Simulation results for an average BGP convergence time and number of update messages received are listed in Table 3 and Table 4, respectively. BGP convergence time depends on the length of routes from the origin to other BGP routers. In the case of high load scenario, we repeat simulations using different nodes as the origin. In this scenario, the source node sends traffic at 200 s and at 730 s.

The majority of BGP routers with default-MRAI-30 require 39 MRAI rounds to obtain the best route, resulting in an average BGP convergence time of approximately 1,192 s. However, majority of BGP routers with FLD-MRAI requires 25 MRAI rounds to obtain the best route, resulting in an average BGP convergence time of approximately 765 s, which is smaller than adaptive MRAI and default-MRAI-15 times. Furthermore, FLD-MRAI performs better than FLD-MRAI-30 and FLD-MRAI-15 and reduces average convergence time by 36%. The network has to wait for many MRAI rounds to converge due to high load of update messages. However, BGP router based on FLD-MRAI changes its path according to available CPU and BGP converges within few MRAI rounds. The FLD-MRAI reduces the number of overall update messages by 70%, from 14,911 to 4,526. Hence for high load scenario, the FLD-MRAI algorithm performs better than other BGP options.

TABLE 3. AVERAGE CONVERGENCE TIME (S) FOR NETWORK TOPOLOGY 1 FOR VARIOUS BGP OPTIONS.

| Scenarios | default-MRAI-30 | default-MRAI-15 | adaptive MRAI | FLD-MRAI-15 | FLD-MRAI-30 | FLD-MRAI |
|---------------|-----------------|-----------------|---------------|-------------|-------------|----------|
| <i>Tshort</i> | 126.66 | 145.49 | 131.90 | 68.73 | 69.40 | 66.93 |
| <i>Tlong</i> | 138.47 | 143.81 | 142.62 | 78.45 | 79.60 | 77.07 |
| <i>Tup</i> | 126.39 | 145.50 | 132.02 | 66.03 | 67.80 | 65.33 |
| <i>Tdown</i> | 138.52 | 145.62 | 141.40 | 74.73 | 75.86 | 74.73 |
| High load | 1,192.07 | 1,196.21 | 1,047.42 | 767.62 | 782.62 | 764.63 |

TABLE 4. OVERALL NUMBER OF UPDATE MESSAGES FOR NETWORK TOPOLOGY 1 FOR VARIOUS BGP OPTIONS.

| Scenarios | default-MRAI-30 | default-MRAI-15 | adaptive MRAI | FLD-MRAI-15 | FLD-MRAI-30 | FLD-MRAI |
|---------------|-----------------|-----------------|---------------|-------------|-------------|----------|
| <i>Tshort</i> | 726 | 1304 | 870 | 375 | 373 | 374 |
| <i>Tlong</i> | 608 | 1073 | 1142 | 452 | 456 | 445 |
| <i>Tup</i> | 681 | 1262 | 763 | 399 | 420 | 391 |
| <i>Tdown</i> | 673 | 1251 | 751 | 394 | 415 | 386 |
| High load | 14,911 | 27,566 | 12,362 | 10,094 | 10,549 | 4,526 |

3) *BGP-RP*: RFD considers change in each route withdrawal or route attribute as a flap. In network Topology 1, we repeat simulations using different nodes as the origin. The number of flaps varies with variation in route length between the origin and other BGP routers. The ns-BGP module does not have routing policy module. Hence, BGP routers without policies always follow the shortest path to the destination.

We compare simulation results of BGP module without policies to BGP-RP with implemented AS path list and community list policies. Simulation results show that permitting or denying incoming or outgoing traffic from other networks lead to long convergence time. BGP-RP with AS path list policy shows similar behavior to BGP without routing policies as shown in Table 5. Hence, permitting or denying a community permits searching for another path to reach the destination and may result in prolonged convergence time. BGP-RP with AS path list policy works better than BGP-RP with community list policy. It is difficult to infer the internal routing structure in the Internet because the Internet Service Providers follow policy decisions based on financial agreements rather than BGP convergence.

TABLE 5. COMPARISON OF BGP MODULE WITH AND WITHOUT POLICIES: CONVERGENCE TIME, NUMBER OF UPDATES, AND NUMBER OF FLAPS.

| Scenario | Convergence time (s) | Number of updates | Number of flaps |
|----------------------------|----------------------|-------------------|-----------------|
| BGP without policies | 129.29 | 745 | 302 |
| BGP-RP with AS path list | 130.65 | 1,102 | 437 |
| BGP-RP with community path | 147.13 | 970 | 430 |

B. Network Topology 5

1) *FLD-MRAI*: Networks with large diameters require faster BGP convergence when the BGP convergence procedure takes many MRAI rounds. The router's CPU utilization depends on the number of BGP update messages received during MRAI rounds. Networks with large diameters, having CPU utilization available helps to lower the BGP convergence time and avoids network congestion. Simulation results for average BGP convergence time and the number of update messages received during the period

of convergence are shown in Table 6 and Table 7, respectively. As the number of network nodes increases, the volume of traffic increases. The FLD-MRAI algorithm performs even better as the number of network nodes increase.

On average, the proposed modifications help reduce the average convergence time by approximately 24% (57%) and number of update messages by approximately 70% (80%) for events in normal (high) network load events.

The average percentage of improved performance of FLD-MRAI over default MRAI (30 s) based on different network topologies is shown in Table 8. On average, FLD-MRAI has approximately 24% (46%) smaller BGP convergence time and approximately 47% (57%) smaller number of exchanged update messages than default BGP settings when the algorithm detects normal (high) network load.

TABLE 8. PERFORMANCE OF FLD-MRAI.

| Events | Convergence time (s) | Overall number of updates |
|---------------|----------------------|---------------------------|
| <i>Tshort</i> | 24% | 47% |
| <i>Tlong</i> | 26% | 51% |
| <i>Tup</i> | 27% | 47% |
| <i>Tdown</i> | 25% | 46% |

2) *BGP-RP*: The convergence time of BGP-RP with AS path list policy is similar to BGP without routing policies, as shown in Table 9.

TABLE 9. COMPARISON OF BGP MODULE WITH AND WITHOUT POLICIES: CONVERGENCE TIME, NUMBER OF UPDATES, AND NUMBER OF FLAPS.

| Scenario | Convergence time (s) | Number of updates | Number of flaps |
|----------------------------|----------------------|-------------------|-----------------|
| BGP without policies | 498.92 | 24,822 | 15,425 |
| BGP-RP with AS path list | 522.19 | 24,366 | 15,879 |
| BGP-RP with community list | 562.55 | 26,791 | 17,792 |

VI. CONCLUSIONS

In this paper, we propose BGP modifications to reduce the convergence time and number of update messages exchanged during normal and high traffic loads.

TABLE 6. AVERAGE CONVERGENCE TIME (S) FOR NETWORK TOPOLOGY 5 FOR VARIOUS BGP OPTIONS.

| Scenarios | default-MRAI-30 | default-MRAI-15 | adaptive MRAI | FLD-MRAI-15 | FLD-MRAI-30 | FLD-MRAI |
|---------------|-----------------|-----------------|---------------|-------------|-------------|----------|
| <i>Tshort</i> | 772.91 | 775.71 | 782.42 | 659.90 | 792.67 | 601.50 |
| <i>Tlong</i> | 795.61 | 778.60 | 783.26 | 660.66 | 794.67 | 608.66 |
| <i>Tup</i> | 773.03 | 775.65 | 782.34 | 659.66 | 793.05 | 602.51 |
| <i>Tdown</i> | 796.13 | 779.60 | 784.71 | 661.09 | 794.46 | 609.33 |
| High load | 918.02 | 909.48 | 906.42 | 930.95 | 951.70 | 530.39 |

TABLE 7. OVERALL NUMBER OF UPDATE MESSAGES FOR NETWORK TOPOLOGY 5 FOR VARIOUS BGP OPTIONS.

| Scenarios | default-MRAI-30 | default-MRAI-15 | adaptive MRAI | FLD-MRAI-15 | FLD-MRAI-30 | FLD-MRAI |
|---------------|-----------------|-----------------|---------------|-------------|-------------|----------|
| <i>Tshort</i> | 8,330 | 12,298 | 6,526 | 6,342 | 7,755 | 2,579 |
| <i>Tlong</i> | 8,349 | 12,286 | 6,514 | 6,315 | 7,721 | 2,564 |
| <i>Tup</i> | 8,323 | 12,292 | 6,520 | 6,331 | 7,734 | 2,523 |
| <i>Tdown</i> | 8,326 | 12,286 | 6,526 | 6,315 | 7,721 | 2,565 |
| High load | 13,353 | 13,422 | 10,466 | 6,141 | 6,257 | 2,672 |

We propose modified DoP, which depends on the calculation of available CPU. We also propose separate durations of MRAI intervals for different events that occur during BGP advertisements. The proposed FLD-MRAI algorithm employs modified reusable MRAI timers. To evaluate its performance, we generate various simulation topologies and advertisement events. Manually created topology, transit-stub hierarchical topology, and topology based on the GLP model have been used.

Simulation results show that FLD-MRAI performs better than other BGP options at the cost of computing available CPU of neighboring routers. Furthermore, the CPU processing capability and duration of MRAI timers greatly affect the BGP convergence time.

In today's Internet, most networks exchange routing information at low cost with the help of routing policies. Simulation results show that AS path list policy performs similarly to BGP without any policy setup and performs better than the community list policy. FLD-MRAI and BGP-RP with AS path list policy exhibit the best performance in networks with large diameter and may help improve performance of today's Internet.

REFERENCES

- [1] Y. Rekhter and T. Li, "A border gateway protocol 4 (BGP-4)," *IETF RFC 1771*, Mar. 1995.
- [2] N. Laskovic and Lj. Trajković, "BGP with an adaptive minimal route advertisement interval," in *Proc. IPCCC*, Phoenix, AZ, USA, Apr. 2006, pp. 142–151.
- [3] T. G. Griffin and B. J. Premore, "An experimental analysis of BGP convergence time," in *Proc. ICNP*, Riverside, CA, USA, Nov. 2001, pp. 53–61.
- [4] B. Premore, "An analysis of convergence properties of the border gateway protocol using discrete event simulation," Ph. D. Thesis, Dartmouth College, 2003.
- [5] T. D. Feng, R. Ballantyne, and Lj. Trajković, "Implementation of BGP in a network simulator," in *Proc. ATS*, Arlington, VA, USA, Apr. 2004, pp. 149–154.
- [6] A. Feldmann, H. Kong, O. Maennel, and A. Tudor, "Measuring BGP pass-through times," in *Proc. PAM*, Antibes Juan-les-Pins, France, Apr. 2004, pp. 267–277.
- [7] K. Varadhan, R. Govindan, and D. Estrin, "Persistent route oscillations in inter-domain routing," *Computer Networks Journal*, vol. 32, no. 1, pp. 1–16, Mar. 2000.
- [8] C. Villamizar, R. Chandra, and R. Govindan, "BGP route flap damping," *IETF RFC 2439*, Nov. 1998.
- [9] W. Sun, Z. M. Mao, and K. G. Shin, "Differentiated BGP update processing for improved routing convergence," in *Proc. ICNP*, Santa Barbara, CA, USA, Nov. 2006, pp. 280–289.
- [10] A. Fabrikant, U. Syed, and J. Rexford, "There's something about MRAI: timing diversity can exponentially worsen BGP convergence," in *Proc. INFOCOMM*, Shanghai, China, Apr. 2011, pp. 2975–2983.
- [11] B. Wang, "The research of BGP convergence time," in *Proc. ITAIC*, Chongqing, China, Aug. 2011, vol. 2, pp. 354–357.
- [12] G. Huston, M. Rossi, and G. Armitage, "A technique for reducing BGP update announcements through path exploration damping," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 8, pp. 1271–1286, Oct. 2010.
- [13] S. Deshpande and B. Sikdar, "On the impact of route processing and MRAI timers on BGP convergence times," in *Proc. GLOBECOM*, Dallas, TX, USA, Nov. 2004, vol. 2, pp. 1147–1151.
- [14] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the Internet topology," in *Proc. SIGCOMM*, Cambridge, MA, USA, Sept. 1999, pp. 251–262.
- [15] G. Siganos, M. Faloutsos, P. Faloutsos, and C. Faloutsos, "Power-laws and the AS-level Internet topology," *IEEE/ACM Trans. Networking*, vol. 11, no. 4, pp. 514–524, Aug. 2003.
- [16] R. Teixeira, A. Shaikh, T. Griffin, and J. Rexford, "Dynamics of hot-potato routing in IP networks," in *Proc. ACM SIGMETRICS*, New York, NY, USA, June 2004, pp. 307–319.
- [17] T. Griffin and G. Wilfong, "An analysis of BGP convergence properties," in *Proc. SIGCOMM*, Cambridge, MA, USA, Aug. 1999, pp. 277–288.
- [18] C. Labovitz, G. Malan, and F. Jahanian, "Origins of Internet routing instability," in *Proc. INFOCOMM*, New York, NY, USA, Mar. 1999, pp. 218–226.
- [19] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed Internet routing convergence," in *Proc. ACM SIGCOMM*, Stockholm, Sweden, Aug. 2000, pp. 175–187.
- [20] D. Pei and J. V. Merwe, "BGP convergence in virtual private networks," in *Proc. IMC*, Rio de Janeiro, Brazil, Oct. 2006, pp. 283–288.
- [21] S. Aggarwal and M. Aggarwal, "Dynamic load balancing based on CPU utilization and data locality in distributed database using priority policy," in *Proc. ICSTE*, Phuket, Thailand, Oct. 2010, vol. 2, pp. 388–391.
- [22] S. Agarwal, C. Chuah, S. Bhattacharyya, and C. Diot, "Impact of BGP dynamics on router CPU utilization," in *Proc. PAM*, Antibes Juan-les-Pins, France, Apr. 2004, pp. 278–288.
- [23] Operational Experience with the BGP-4 protocol [Online]. Available: <http://tools.ietf.org/html/draft-ietf-idr-bgp4-op-experience-01>.
- [24] Troubleshooting BGP [Online]. Available: http://www.nanog.org/meetings/nanog42/presentations/PSmith_BGP.pdf.
- [25] Z. Mao, R. Govindan, G. Varghese, and R. Katz, "Route flap damping exacerbates Internet routing convergence," in *Computer Communication Review Journal*, vol. 32, no. 4, pp. 221–233, Aug. 2002.
- [26] Route flap damping considered useable [Online]. Available: <http://www.iepg.org/2010-11-ietf79/101107.iepg-rfd.pdf>.
- [27] Flap damping with assured reachability [Online]. Available: <http://irl.cs.ucla.edu/~j13park/rfdrg.pdf>.
- [28] L. Gao and J. Rexford, "Stable Internet routing without global coordination," in *Proc. ACM SIGCOMM*, San Diego, CA, USA, Dec. 2001, pp. 681–692.
- [29] C. Labovitz, R. Wattenhofer, S. Venkatachary, and A. Ahuja, "The impact of Internet policy and topology on delayed routing convergence," in *Proc. INFOCOMM*, Anchorage, AL, USA, Apr. 2001, pp. 537–546.
- [30] BGP++ [Online]. Available: <http://www.ece.gatech.edu/research/labs/MANIACS/BGP++/>.
- [31] TRE [Online]. Available: <http://laurikari.net/tre/>.
- [32] BCNET [Online]. Available: <https://wiki.bc.net/atl-conf/display/Content/Home>.
- [33] GT-ITM [Online]. Available: <http://www.cc.gatech.edu/projects/gtitm/>.
- [34] BRITE [Online]. Available: <http://www.cs.bu.edu/brite>.
- [35] T. Farah, S. Lally, R. Gill, N. Al-Rousan, R. Paul, D. Xu, and Lj. Trajkovic, "Collection of BCNET BGP traffic," in *Proc. 23rd ITC*, San Francisco, CA, USA, Sept. 2011, pp. 322–323.
- [36] S. Lally, T. Farah, R. Gill, R. Paul, N. Al-Rousan, and Lj. Trajkovic, "Collection and characterization of BCNET BGP traffic," in *Proc. 2011 IEEE PACRIM*, Victoria, BC, Canada, Aug. 2011, pp. 830–835.
- [37] Inet: Internet topology generator [Online]. Available: <http://topology.eecs.umich.edu/inet/inet-2.0.pdf>.