Effect of Minimal Route Advertisement Interval Timers on Border Gateway Protocol Convergence

Rajvir Gill
rajvirg@sfu.ca

Communication Networks Laboratory
http://www.ensc.sfu.ca/research/cnl

School of Engineering Science
Simon Fraser University
Roadmap

- Introduction
- BGP convergence
- Flexible Load Dispersing (FLD)-MRAI algorithm:
  - CPU utilization and modified DoP
  - modified reusable timers
  - duration of MRAI
  - space and time complexity
- Implementation of FLD-MRAI
- Performance evaluation
- Conclusions, future work, references
**Introduction**

- BGP is de-facto inter AS routing protocol
- Operates successfully in a network of the Internet's size
- Supports CIDR
- BGP peer routers exchange four types of messages:
  - open
  - update
  - notification
  - keepalive
- BGP utilizes a path vector algorithm called the best path selection algorithm to select the best path

BGP: Border Gateway Protocol
AS: Autonomous System
CIDR: Classless Inter-Domain Routing
Minimal Route Advertisement Interval: MRAI

- MRAI is the interval limitation that affects BGP convergence
- Default value: 30 s
- MRAI timers control the MRAI value:
  - per-destination
  - per-peer
- Optimal MRAI value depends on:
  - network size
  - topology
  - traffic volume
  - network conditions

MRAI timers

- **Per-destination:**
  - associated with each network destination
  - may not be used because of the Internet size

- **Per-peer:**
  - associated with each peer in the network
  - starts ticking when the source router sends a route advertisement to peers
  - adversely affect advertisements to each destination
Processing delay

- Total time of an update waiting in the queue and the time required for a BGP router to process
- **Uniform processing delay:**
  - BGP router processes update messages sequentially
  - delay in processing updates affects the processing time of update messages that follow
- **Measurements:**
  - update messages are processed within 200 ms
  - average processing time is 101 ms with the upper bound of 400 ms

Motivation

- One of the **major problems** of BGP: a longer convergence time and a large number of update messages due to unreachable destinations
  - path to the destination failure
  - router failure
- Possible solution:
  - an algorithm that decreases the BGP convergence time and the number of update messages exchanged in the network
Contributions

- MRAI with Flexible Load Dispersing (FLD-MRAI) algorithm:
  - modifies reusable timers
  - employs MRAI durations based on BGP advertisement events
  - applies to heterogeneous and large networks
  - performs well in networks with unspecified traffic load
Roadmap

- Introduction
- BGP convergence
- Flexible Load Dispersing (FLD)-MRAI algorithm:
  - CPU utilization and modified DoP
  - modified reusable timers
  - duration of MRAI
  - space and time complexity
- Implementation of FLD-MRAI
- Performance evaluation
- Conclusions, future work, references
BGP convergence

- **BGP convergence time:** the time interval between first update message sent, until all update messages that are a consequence of the original update are received.
- **Adaptive MRAI timers** decrease the BGP convergence time and guarantee network stability.
- **PED algorithm proposed timer:** 35 s.
- **Processing efficiency of router’s CPU affects the BGP convergence time.**

**PED:** Path Exploration Damping
BGP convergence

- SSLD and optimal values for MRAI reduce the BGP convergence time.
- Delay due to router or link failure increases BGP convergence time.
- Shortest path to destination decreases BGP convergence time.
- Router’s CPU load depends on the number of BGP messages.
- High CPU utilization of a BGP router causes delay.
- Mismatch in policy configurations between two ASes may also cause network instabilities.


SSLD: Sender Side Loop Detection
CPU: Central Processing Unit
Roadmap

- Introduction
- BGP convergence
- Flexible Load Dispersing (FLD)-MRAI algorithm:
  - CPU utilization and modified DoP
  - modified reusable timers
  - duration of MRAI
  - space and time complexity
- Implementation of FLD-MRAI
- Performance evaluation
- Conclusions, future work, references
FLD-MRAI algorithm: scenarios

- Processing of updates:
  - normal load: DoP prefers the shortest path
  - high load: DoP prefers a longer path in the presence of the shortest path
  - empirical value for processing delay: 200 ms
  - MRAI consists of two states: idle and processing

DoP: Degree of Preference
FLD-MRAI algorithm: CPU utilization

- Calculation of available CPU:
  - CPU utilization is high, then the router responds slowly to subsequent requests in the queue
  - based on the priority of updates:
    \[
    \begin{align*}
    CPU_{\text{available}} &= 100 - CPU_{\text{active}} \\
    CPU_{\text{active}} &= 100 \times \left(\frac{CPU_{\text{current}}}{CPU_{\text{max}}}\right)
    \end{align*}
    \]
    - \(CPU_{\text{available}}\): percentage of available CPU of the neighboring router
    - \(CPU_{\text{active}}\): percentage of active CPU utilization of the neighboring router
    - \(CPU_{\text{current}}\): current CPU utilization
    - \(CPU_{\text{max}}\): maximum CPU utilization
  - calculated every time a router receives the updates of a new or withdrawn route
FLD-MRAI algorithm: modified DoP

- Modified DoP ($\text{DoP}_{\text{mod}}$):
  - function of $\text{Route}_{\text{info}}$ and $\text{CPU}_{\text{available}}$,
    where $\text{Route}_{\text{info}}$ is the route having the shortest path
  - new, replaced, and/or withdrawn route
  - path with the highest value of $\text{DoP}_{\text{mod}}$ is given the highest priority
FLD-MRAI algorithm: example with five routers

- $R_0$ is the source router and it advertises to destination $R_2$
- Two possible paths: $R_0$-$R_1$-$R_2$ and $R_0$-$R_4$-$R_3$-$R_2$
- Original BGP router: $R_0$-$R_1$-$R_2$
- FLD-MRAI: $R_0$-$R_1$-$R_2$ (normal load) and $R_0$-$R_4$-$R_3$-$R_2$ (high load)
Roadmap

- Introduction
- BGP convergence
- Flexible Load Dispersing (FLD)-MRAI algorithm:
  - CPU utilization and modified DoP
  - modified reusable timers
  - duration of MRAI
  - space and time complexity
- Implementation of FLD-MRAI
- Performance evaluation
- Conclusions, future work, references
FLD-MRAI algorithm: advertisement events

- MRAI timer depends on network conditions and advertisement events:
  - Tdown: link failure
  - Tup: link failure recovery
  - Tlong: router failure
  - Tshort: router failure recovery

- Occurrence of advertisement events:

<table>
<thead>
<tr>
<th>Events</th>
<th>Number of events occurring during BGP convergence period</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tdown</td>
<td>43.4%</td>
</tr>
<tr>
<td>Tup</td>
<td>39.9%</td>
</tr>
<tr>
<td>Tlong</td>
<td>7.3%</td>
</tr>
<tr>
<td>Tshort</td>
<td>7.4%</td>
</tr>
<tr>
<td>Unidentified</td>
<td>2.0%</td>
</tr>
</tbody>
</table>

FLD-MRAI algorithm: modified reusable timers

- After the processing time, three events may occur:
  - no new update received
  - new update message received
  - MRAI reusable timer expired

- Calculation of idle time:
  \[ T_{idle}(D) = MRAI_{total} - M_{last} \]

  \( T_{idle}(D) \): is the idle time of the destination

  \( MRAI_{total} \): total MRAI

  \( M_{last} \): time instance of the last message received
FLD-MRAI algorithm: modified reusable timers

- **One reusable timer** is required for all paths advertised during a short time interval.
- Number of rounds per reusable MRAI timer controls the duration of MRAI round:

\[
\text{MRAI}_{\text{duration}} = R_n \times (t_n \times g)
\]

- \(\text{MRAI}_{\text{duration}}\): duration of MRAI round
- \(R_n\): number of rounds per reusable MRAI timer
- \(t_n\): number of reusable MRAI timers
- \(g\): granularity
Roadmap

- Introduction
- BGP convergence
- Flexible Load Dispersing (FLD)-MRAI algorithm:
  - CPU utilization and modified DoP
  - modified reusable timers
  - duration of MRAI
  - space and time complexity
- Implementation of FLD-MRAI
- Performance evaluation
- Conclusions, future work, references
FLD-MRAI algorithm: duration of MRAI

- Duration of MRAI timers:
  - longer duration: $T_{\text{down}}$ and $T_{\text{long}}$
  - shorter duration: $T_{\text{up}}$ and $T_{\text{short}}$
- MRAI value:
  - same MRAI value: $T_{\text{down}}$ and $T_{\text{long}}$
  - same MRAI value: $T_{\text{up}}$ and $T_{\text{short}}$
- Minimum duration: 15 s
FLD-MRAI algorithm: duration of MRAI

- Idle time is longer than 1 s:
  - $T_{short}$ or $T_{up}$
  - the shortest path becomes available
  - duration of MRAI round: 15 s
  - use one MRAI round: 15 s

- Idle time is shorter than 1 s:
  - $T_{long}$ or $T_{down}$
  - router or link failure
  - duration of MRAI round: 30 s
  - use two MRAI rounds: 15 s
FLD-MRAI algorithm: example of reusable timers

- 15 reusable MRAI timers with granularity 1 s
- Timer\(_0\) lasts one round of 15 s (T\text{short} or T\text{up})
- Timer\(_2\) lasts one round of 30 s (T\text{down} or T\text{long})
- After expiration, the duration of timers depends on the idle time
FLD-MRAI algorithm: pseudocode

```
when sending advertisement of the destination D to peers at t₀
  set (S₁) // priority numbers on received updates according to the shortest path
  if (C₁ < C₂) // calculate and compare the available CPU of the first and second priority neighboring router
    if W(t) < T(t) // calculate and compare the waiting and transmission times
      else (wait in queue of the first priority path)
      if dop₂ < dop₁ // calculate and compare the degree of preference
        choose the second priority path
        MRAI = 30 s
        goto processing state
      else (wait in queue of the first priority path)
      else if (C₁ > C₂)
        wait in queue of the first priority path // duration of MRAI is based on the idle time
        goto processing state
```
FLD-MRAI algorithm: pseudocode

```
when initiation of the new round
    if (Idle(D) > 1 s)  // Tshort or Tup may occur
        set modified_reusable timer = 15 s
    else if (e ∈ network failure)
        // events change due to the network failure
        choose the second priority path  // after expiration of the timer
        set modified_reusable timer = 30 s
        goto processing state
    else if (e ∈ network failure)
        goto processing state
    else if (Idle(D) < 1 s)  // Tlong or Tdown may occur
        set modified_reusable timer = 30 s
    else if (P_t ∈ P_s)  // if the shortest path becomes available
        choose the shortest path // after expiration of the timer
        set modified_reusable timer = 15 s
        goto processing state
    else (P_t ∉ P_s)  // if the shortest path is not available
        goto processing state
```
Roadmap

- Introduction
- BGP convergence
- Flexible Load Dispersing (FLD)-MRAI algorithm:
  - CPU utilization and modified DoP
  - modified reusable timers
  - duration of MRAI
  - space and time complexity
- Implementation of FLD-MRAI
- Performance evaluation
- Conclusions, future work, references
Space complexity:
- depends on the number of non-converged routes (n)
- router keeps three variables of each non-converged route: $CPU_{current}$, $CPU_{max}$, and $M_{last}$
- variables are integer counters that a router may easily store
- space complexity is $O(n)$
FLD-MRAI algorithm: time complexity

- Time complexity:
  - division, multiplication, and subtraction operations: $\text{CPU}_{\text{available}}$, $T_{\text{idle}}$, and $\text{MRAI}_{\text{duration}}$
  - division and multiplication depend on input size $n$
  - subtraction is constant
  - approximate these variables with constants equal to their maximum values
  - calculation of variables do not depend on input size $n$
  - FLD-MRAI requires two subtractions, three multiplications, and one division (equations: pages 14, 19, and 20)
FLD-MRAI algorithm: time complexity

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

- Introduction
- BGP convergence
- Flexible Load Dispersing (FLD)-MRAI algorithm:
  - CPU utilization and modified DoP
  - modified reusable timers
  - duration of MRAI
  - space and time complexity
- Implementation of FLD-MRAI
- Performance evaluation
- Conclusions, future work, references
Implementation: FLD-MRAI

- ns-2.34 network simulator
- ns-BGP 2.0 developed module (ported from the SSFNET simulator)
- Routing structure of a modified ns-2 node:
  - **forwarding plane**: categorizes the received packets whether to be processed or forwarded to neighboring nodes
  - **control plane**: controls computation, maintenance, and implementation of routes in routing tables
Implementation: FLD-MRAI

Diagram showing the implementation of FLD-MRAI with various components such as `rtObject`, `rtProto/DV`, `MRAI Timers`, `PeerEntry`, `LocRIB`, `inbuf`, `TcpSocket`, `Agent/TCP/FullTcp`, `AdjIn`, `AdjOut`, `DampInfo`, `ReuseTimer`, and functions like `addRoute()`, `node()`, `Classifier_`, and `install()`.
Simulation scenarios: performance comparisons

- **FLD-MRAI algorithm:**
  - FLD-MRAI-30
  - FLD-MRAI-15
  - default-MRAI-30
  - default-MRAI-15
  - adaptive MRAI
## Simulated topologies

<table>
<thead>
<tr>
<th>Topology</th>
<th>Number of nodes</th>
<th>Topology generator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topology 1</td>
<td>67</td>
<td>Manually from BCNET BGP traffic</td>
</tr>
<tr>
<td>Topology 2</td>
<td>100</td>
<td>GT-ITM</td>
</tr>
<tr>
<td>Topology 3</td>
<td>200</td>
<td>GT-ITM</td>
</tr>
<tr>
<td>Topology 4</td>
<td>300</td>
<td>BRITE</td>
</tr>
<tr>
<td>Topology 5</td>
<td>500</td>
<td>BRITE</td>
</tr>
</tbody>
</table>

**GT-ITM**: Georgia Tech Internetwork Topology Models  
**BRITE**: Boston university Representative Internet Topology gEnerator
## BGP routing table (RIB)

<table>
<thead>
<tr>
<th>Time</th>
<th>Peer’s IP</th>
<th>Peer’s AS</th>
<th>Source IP</th>
<th>AS path</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011-10-24, 05:18</td>
<td>216.6.50.9</td>
<td>6327</td>
<td>207.23.253.2</td>
<td>6327-7575-56203-1221</td>
</tr>
<tr>
<td>2011-10-24, 05:18</td>
<td>207.23.253.34</td>
<td>6453</td>
<td>207.23.253.2</td>
<td>6453-2914-2519-1221</td>
</tr>
<tr>
<td>2011-10-24, 05:18</td>
<td>216.6.50.9</td>
<td>6327</td>
<td>207.23.253.2</td>
<td>6327-2516-2519-1221</td>
</tr>
<tr>
<td>2011-10-24, 05:18</td>
<td>207.23.253.34</td>
<td>6453</td>
<td>207.23.253.2</td>
<td>6453-4725-7670-18144-1221</td>
</tr>
<tr>
<td>2011-10-24, 05:18</td>
<td>216.6.50.9</td>
<td>6327</td>
<td>207.23.253.2</td>
<td>6327-2516-7670-18144-1221</td>
</tr>
<tr>
<td>2011-10-24, 05:18</td>
<td>207.23.253.34</td>
<td>6453</td>
<td>207.23.253.2</td>
<td>6453-4725-1221</td>
</tr>
<tr>
<td>2011-10-24, 05:18</td>
<td>216.6.50.9</td>
<td>6327</td>
<td>207.23.253.2</td>
<td>6327-4725-1221</td>
</tr>
</tbody>
</table>

**RIB**: Routing Information Base
Number of nodes in a generated topology is calculated as:

\[ N = T \times N_t \times [1 + (K \times N_s)] \]

- \( N \): number of nodes
- \( T \): fully connected transit domain
- \( N_t \): average number of nodes per transit AS
- \( K \): average number of stub ASes per transit AS
- \( N_s \): average number of nodes per stub AS

<table>
<thead>
<tr>
<th>Symbols</th>
<th>100-node topology</th>
<th>200-node topology</th>
</tr>
</thead>
<tbody>
<tr>
<td>( T )</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>( N_t )</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>( K )</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>( N_s )</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>( N )</td>
<td>100</td>
<td>200</td>
</tr>
</tbody>
</table>

GT-ITM: Georgia Tech Internetwork Topology Models
BRITE topologies

- BRITE generates different types of Internet topologies from different models
- Generate AS-level topologies

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node placement</td>
<td>Random</td>
</tr>
<tr>
<td>Growth type (how nodes join in topology)</td>
<td>Incremental</td>
</tr>
<tr>
<td>Preferential connectivity</td>
<td>On</td>
</tr>
<tr>
<td>Bandwidth distribution</td>
<td>Constant</td>
</tr>
<tr>
<td>Alpha (GLP-specific exponent)</td>
<td>0.45</td>
</tr>
<tr>
<td>Beta (GLP-specific exponent)</td>
<td>0.65</td>
</tr>
<tr>
<td>M (number of links per new node)</td>
<td>1</td>
</tr>
<tr>
<td>N (number of nodes)</td>
<td>300 or 500</td>
</tr>
</tbody>
</table>

BRITE: Boston university Representative Internet Topology Generator
Assumptions

- Do not consider route flap damping when evaluating the performance of FLD-MRAI
- Each AS consists of a single BGP router
- BGP convergence procedure is complete if BGP router receives no update message from other BGP routers within 60 s
Roadmap

- Introduction
- BGP convergence
- Flexible Load Dispersing (FLD)-MRAI algorithm:
  - CPU utilization and modified DoP
  - modified reusable timers
  - duration of MRAI
  - space and time complexity
- Implementation of FLD-MRAI
- Performance evaluation
- Conclusions, future work, references
Validation test: network with 5 routers

- Topology with the **minimum number of nodes** is used to analyze the BGP convergence time
- FLD-MRAI algorithm is validated by using a **simple network of five routers**
- We tested the employed modifications in ns-BGP for both normal and high loads
- BGP convergence time of both scenarios with FLD-MRAI is compared to **default-MRAI-30**
- Node 0: source node
- Node 2: destination node
Validation test: network with 5 routers

- **Normal load scenario events:**
  - Tlong : n1 fails
  - Tshort : n1 recovers
  - Tdown: link between n0 and n1 fails
  - Tup : link between n0 and n1 recovers

- **High load scenario:** high traffic load to n1

- Simulation results indicate that FLD-MRAI performs as expected

<table>
<thead>
<tr>
<th>Scenarios</th>
<th>default-MRAI-30 (s)</th>
<th>FLD-MRAI (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tshort</td>
<td>88.70</td>
<td>52.70</td>
</tr>
<tr>
<td>Tlong</td>
<td>93.10</td>
<td>71.59</td>
</tr>
<tr>
<td>Tup</td>
<td>88.60</td>
<td>55.50</td>
</tr>
<tr>
<td>Tdown</td>
<td>93.05</td>
<td>60.90</td>
</tr>
<tr>
<td>High load</td>
<td>102.91</td>
<td>56.81</td>
</tr>
</tbody>
</table>
Validation test: completely connected graph

- Validate performance of the FLD-MRAI algorithm
- Choose the completely connected network with fifteen nodes
- We match simulation results of the convergence time and the number of update messages with results of the previous studies
- Simulate only Tdown event
- FLD-MRAI decreases the number of update messages from 3,200 to 1,500

Validation test: completely connected graph

- Minor changes may exist due to the different simulation setups
- Results of the BGP convergence time and the number of update messages for default-MRAI-30 are similar to the results reported in the previous studies.

\[ M_0 : \text{Optimal MRAI value for FLD-MRAI} \]
\[ M_1 : \text{Optimal MRAI value for default-MRAI-30} \]
Performance evaluation: network Topology 1

- FLD-MRAI algorithm: convergence time (s) of the $T_{short}$ event for network Topology 1 for various BGP options
Performance evaluation: network Topology 1

- FLD-MRAI algorithm: convergence time (s) of the $T_{long}$, $T_{up}$, and $T_{down}$ events for network Topology 1 for various BGP options
Performance evaluation: network Topology 1

- FLD-MRAI algorithm: overall number of update messages for network Topology 1 for various BGP options
Performance evaluation: network Topology 1

- FLD-MRAI algorithm: convergence time (s) of the high load scenario for network Topology 1 for various BGP options
Performance evaluation: network Topology 1

- FLD-MRAI algorithm: overall number of update messages for the high load scenario
Performance evaluation: network Topology 5

- FLD-MRAI algorithm: average convergence time (s) for network Topology 5 for various BGP options

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Tshort</td>
<td>772.91</td>
<td>775.71</td>
<td>782.42</td>
<td>659.90</td>
<td>792.67</td>
<td>601.50</td>
</tr>
<tr>
<td>Tlong</td>
<td>795.61</td>
<td>778.60</td>
<td>783.26</td>
<td>794.67</td>
<td>660.66</td>
<td>608.66</td>
</tr>
<tr>
<td>Tup</td>
<td>773.03</td>
<td>775.65</td>
<td>782.34</td>
<td>659.66</td>
<td>793.05</td>
<td>602.51</td>
</tr>
<tr>
<td>Tdown</td>
<td>796.13</td>
<td>779.60</td>
<td>784.71</td>
<td>794.46</td>
<td>661.09</td>
<td>609.33</td>
</tr>
<tr>
<td>High load</td>
<td>918.02</td>
<td>909.48</td>
<td>906.42</td>
<td>930.95</td>
<td>951.70</td>
<td>530.39</td>
</tr>
</tbody>
</table>
Performance evaluation: network Topology 5

- FLD-MRAI algorithm: overall number of update messages for network Topology 5 for various BGP options

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Tshort</td>
<td>8,330</td>
<td>12,298</td>
<td>6,526</td>
<td>6,342</td>
<td>7,755</td>
<td>2,579</td>
</tr>
<tr>
<td>Tlong</td>
<td>8,349</td>
<td>12,286</td>
<td>6,514</td>
<td>6,315</td>
<td>7,721</td>
<td>2,564</td>
</tr>
<tr>
<td>Tup</td>
<td>8,323</td>
<td>12,292</td>
<td>6,520</td>
<td>6,331</td>
<td>7,734</td>
<td>2,523</td>
</tr>
<tr>
<td>Tdown</td>
<td>8,326</td>
<td>12,286</td>
<td>6,526</td>
<td>6,315</td>
<td>7,721</td>
<td>2,565</td>
</tr>
<tr>
<td>High load</td>
<td>13,353</td>
<td>13,422</td>
<td>10,466</td>
<td>6,141</td>
<td>6,256</td>
<td>2,672</td>
</tr>
</tbody>
</table>
Performance evaluation

- The average percentage of improvement of FLD-MRAI over default MRAI (30 s) based on different network topologies

<table>
<thead>
<tr>
<th>Events</th>
<th>Convergence time (s)</th>
<th>Overall number of updates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tshort</td>
<td>24%</td>
<td>47%</td>
</tr>
<tr>
<td>Tlong</td>
<td>26%</td>
<td>51%</td>
</tr>
<tr>
<td>Tup</td>
<td>27%</td>
<td>47%</td>
</tr>
<tr>
<td>Tdown</td>
<td>25%</td>
<td>46%</td>
</tr>
</tbody>
</table>
Roadmap

- Introduction
- BGP convergence
- Flexible Load Dispersing (FLD)-MRAI algorithm:
  - CPU utilization and modified DoP
  - modified reusable timers
  - duration of MRAI
  - space and time complexity
- Implementation of FLD-MRAI
- Performance evaluation
- Conclusions, future work, references
Conclusions

- Proposed FLD-MRAI algorithm employs:
  - BGP modifications to reduce the convergence time and number of update messages exchanged during normal and high traffic loads
  - modified DoP that depends on the calculation of available CPU
  - separate durations of MRAI for different events that occur during BGP advertisements
  - modified reusable MRAI timers
Conclusions

- Simulation results show that FLD-MRAI performs better than other BGP options at the cost of computing available CPU of neighboring routers.
- The CPU processing capability and duration of MRAI timers greatly affect the BGP convergence time.
- FLD-MRAI exhibit the best performance in networks with large diameter and may help improve performance of today’s Internet.
Future work

- Analyse the effect of iBGP on the convergence time and the number of update messages
- Explore the effect of routing policies on the BGP convergence time along with the MRAI
- Implement various simulation scenarios and settings
- Test the FLD-MRAI algorithm in a real test-bed

iBGP : Interior Border Gateway Protocol
References:

http://www.sfu.ca/~ljilja/cnl

References: BGP

References: MRAI

References: topology generator

Acknowledgements

Examiner:
- Prof. Emeritus R. H. Stephen Hardy

Supervisor:
- Prof. Emeritus William A. Gruver

Chair:
- Prof. John D. Jones

Senior supervisor:
- Prof. Ljiljana Trajković
Thank you!