Dept. of Information Engineering and Computer Science

Master's Degree in
Computer Science

Final Dissertation

Title

Subtitle (optionl)

Supervisors
......

graduating student
Milani Mattia

Accademic Year 2019/2020

Ringraziamenti

...thanks to...

Contents

Summary

...summary....

Minimum Route Advertisement Interval (MRAI)

1 Introduction

- How is internet built
- the protocol that controls internet

1.1 Internet nowadays

- Use today studies to show how internet is today

1.2 Correlation between variables and convergence

- Expose the hypothesis of the correlation

Forget about the possibility to converge in seconds or even sub-seconds when we talk about internet routing convergence there are a lot of factors that influence it. The convergence time is mostly affected by some timers that rules the Internet. It could require up to different minutes to achieve a complete convergence, spread a new routing information to all the nodes.

One of the most effective timers is MRAI and it has been already proven FiXme: Insert citation that whith

1.3 Goal of this thesis

- Why is important understand this correlation?

2 BGP state of the art

- BGP de facto standard on the internet
- What is an AS
- interconnection between ASes

2.1 BGP

- High level of BGP
- BGP messages
- BGP Update messages
- BGP policies

2.2 BGP Wedgies

- What are wedgies?
- why are them important?
- which situations them occur?

2.3 BGP MRAI

- What is MRAI?
- Previous works on MRAI
- Suppositions on the MRAI influence

2.4 BGP RFD

- What is RFD?
- Why is used RFD?
- Evolution of RFD?
- RFD Today

3 Discrete Event Simulator

Experiments on Border Gateway Protocol (BGP) are not applicable on the Internet, for this reason different studies shows their results using a simulate environment [1] FiXme: Insert other citations. The majority of the studies uses small graphs, and each node of the graph simulate the behaviour of a BGP speaker. Each node represent also a single Autonomous System (AS) and the BGP speaker is it's own exterior router, for simplicity reduced to one speaker that handles all the connections.

For this reason I decided to use and expand a Discrete Event Simulator (DES) that permits to have different grades of freedom, respecting on the other side all the properties required for a reliable simulator environment. I decided to use the *Simpy*[1] package to make the environment evolve. I decided for this package for the extensive documentation and because it has been already used for different studies, demonstrating its adaptability [2, 3].

I developed the DES as a highly modular environment. In Figure 3.1 is possible to see the basic

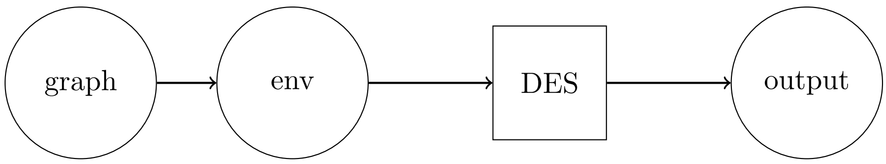


Figure 3.1: Discrete event simulator structure

idea of the simulator. The first component needed is a graph, represented by a *graphml* file, this file is descriptor of the network. it defines also all topological information and all the properties of each single node. FiXme: Look for a Cref implementation of this In Code 3.1 is possible to see an example of a *graphml* file, it describes that node 0 contains a single destination and that the edge between nodes 2 and 5 is controlled by the policy `2, 2, 2` that defines a servicer-provider policy. Policies are encoded using the convention described in [4].

```
<node id="0">
     <data key="d0">10.0.0.0/24</data>
</node>
<edge source="2" target="5">
    <data key="d2">2, 2, 2</data>
</edge>
```

Code 3.1: Graph example

The graph is then embedded in the environment file, this file is in *json* format and it describes how the environment is characterized, it gives the initial values for the Random Number Generator (RNG) so that each experiment is replicable and other properties, like where the output should be saved, and, most importantly how the experiment should be conducted. There are two possible evolution of the environment:

- ***Continuous evolution***: In this category all the nodes that contains at least a destination will continuously share and retrieve the destination accordingly with the distributions defined in the environment;

- ***Signaling evolution***: Is possible to define a precise signal that should be executed by the nodes that contains a destination, for example, the signal "AWA" defines that there will be an announce followed by a withdraw and the an other announce.

[1] Simpy website

The DES take as input this *json* file where all the information are described, it creates an object for each node in the graph file, with each own characteristics. After the initialization all the nodes that contains a destination will schedule the first advertisement of it to their neighbour. The simulation run will terminate only if there are no more events scheduled or if the maximum simulation time is reached.

The DES will then produce a *CSV* output, with all the events that can be analyzed to see the evolution of a specific node or to evaluate the whole network.

3.1 DES Environments

- Example environment
- Explanation of more complex environments
- Internet like graphs
- Fabrikant Graphs

Thanks to the environment codification in a *json* file is possible to define experiments with a high grade of freedom. Is possible to define multiple delays as probability functions vectors that will provide multiple runs possibility. For example, if we have 5 different possible seeds and 3 different delays, the total number of runs combinations is 15, as showed in Code 3.2. is possible to run one of the possible combination of parameters through the identifier of the single run.

```
"simulation" : {
    // seed(s) to initialize RNG
    "seed" : [0, 1, 2, 3, 4],
    ....
    // Multiple withdraw distributions
    "withdraw_dist": [{"distribution": "unif", "min": 5, "max": 10, "int": \
   0.1},
                      {"distribution": "unif", "min": 8, "max": 10, "int": \
   0.1},
                      {"distribution": "unif", "min": 2, "max": 3, "int": \
   0.1}],
    ....
}
```

Code 3.2: Environment example

In the environment is possible to define also the processing time, this time is used inside each BGP node to emulate the processing of information or the evaluation of a packet. Though the *delay* parameter is possible to define the default delay on the edges, is important to remember that the links are FIFO so there is no reordering of messages in the same link, there is also no messages lost. That because it was out of the scope of this thesis to study the evolution of the protocol with packet loss, but it could be a future work.

3.1.1 Clique environment

One of the special environment that I used it's composed by a clique graph graph of different dimensions, an example of clique graph is given in Figure 3.2.

The only node that shares a destination is the node "d", the node 0 will then spread the knowledge to the whole network, and the node "x" will act as a black hole for all the possible paths that the node 5 will share. This topology is used to enforce the path exploration problem.

3.1.2 Fabrikant environment

Another interesting chase to test the path exploration problem is the one presented in [5]. In that study Fabrikant et al. presents how particular MRAI setting could make the network converge with an exponential bechaviour because of the path exploration problem. I used the basic example of their study to investigate how the choose of MRAI is foundamental for the network convergence. An example of the network used is presented in Figure 3.3.

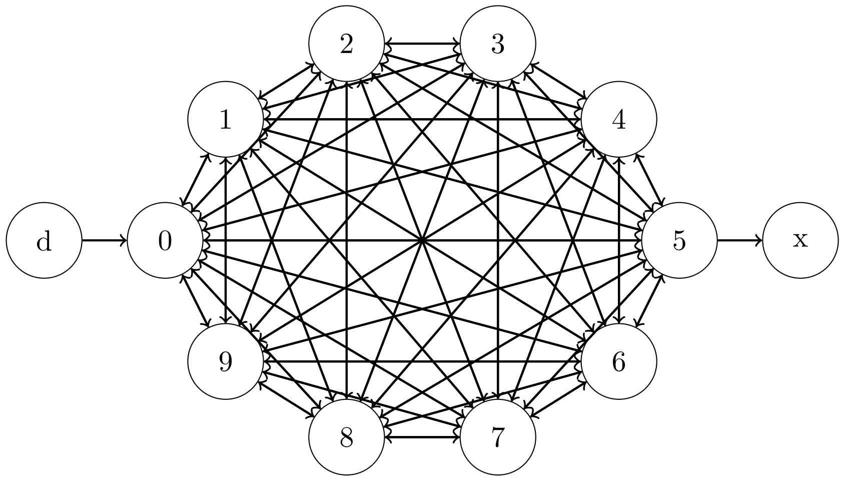


Figure 3.2: Clique graph example

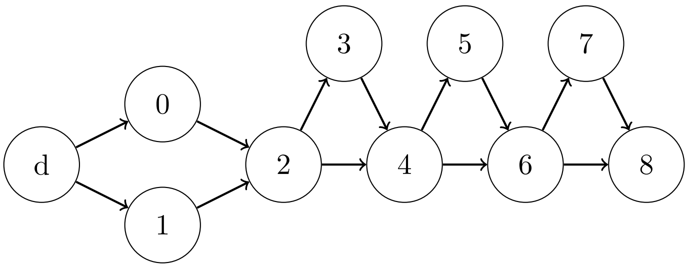


Figure 3.3: Fabrikant chain graph example

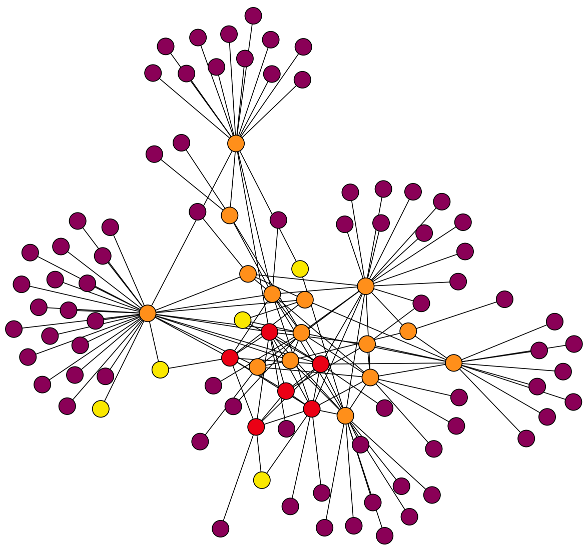
The path exploration problem is caused by the delay on the node 0-2 edge. The node 2 will receive the destination through node 1, after a small ammount of time the network will converge to the best path (without using the backup links). But, after a while, node 2 will receive the network also through node 0 and it will prefer this new path, provoking then the riconfiguration of all the other nodes that will use the backup links for a while, announcing their new path. A wrong configuration of MRAI can provoke the entire exploration of the possibility set.

3.1.3 Internet-like environment

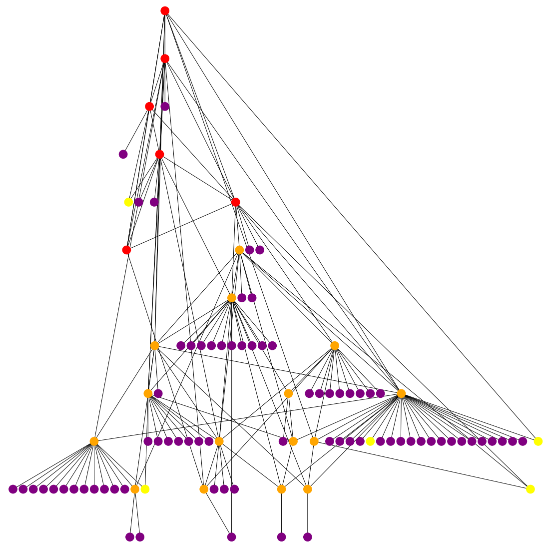
The last noteworthy environment is the one whose purpose is to simulate Internet behaviour. This has been possible thanks to the study by Elmokashfi et al. [6] and the internet like graph generator present in Networkx[2] (a python library famous for graph and network studies). An example with a small set of nodes is presented in Figure 3.4

The different nodes are colored accordingly with the node type represented. The tier one nodes that generate the centrali clique are colored in red, and is possible to notice in Figure 3.4b that them are in the highest levels of the networks. This environment has been used to study the behaviour of the network with topologies resempbling the real internet.

[2]Networkx internet as graph generator



(a) Internet like graph with an "explosive" layout



(b) Internet like graph with a "hierarchical" layout

Figure 3.4: Internet like graph colored to show the hierarchicaly structure, 4 tipes of nodes, T (tier 1 mesh), M, CP, C (Customers, purple one)

4 BGP as a Finite State Machine

- Why an FSM could be useful

4.1 BGP generalization

- BGP as an FSM main idea
- signaling transmutation

4.2 BGP FSM experiments

- First experiments
- Signaling interpretation

4.3 BGP FSM explosion

- Fabrikant experiments
- Path exploration
- States explosion
- signaling explosion

5 BGP MRAI dependance

- Why BGP depends on MRAI?
- Why MRAI prevents messages explosions
- Previous works

5.1 Clique graph

- clique experiments
- messages vs convergence time

5.2 Internet like graph

- Environment state
- what is an internet like graph
- cite elmokashfi
- messages vs time

5.3 Pareto Efficiency Front

- There is space for improvements?

5.4 Strategy dependence

- Describe the convergence time dependance on MRAI
- Describe previous works on this track
- show the differences between different strategies

5.5 Signal dependance

- There is a dependance on the signal and on the effects of MRAI?
- Show the difference between different signals

5.6 Position dependance

- And how much is influent the position?
- Hierarchically?

6 RFD and MRAI correlation

- Expose more deeply what is RFD
- Expose previous studies about RFD
- Today RFD? Outdated

6.1 RFD on toy topologies

- What is the impact of RFD?
- In which occasion is present RFD?
- Clique
- Variations thanks to MRAI

6.2 RFC 2439 VS RFC 7196

- Time comparison between both of them
- how them react differently?
- why?

6.3 Mice VS Elephants

- What is Mice VS Elephants?
- How has been studied in the past?
- Introduce how MRAI affects mice VS elephants

7 Conclusion

- Wrap up
- Path exploration explosion of the FSM
- MRAI convergence dependency
- RFD and MRAI co-dependency

7.1 Future Works

:)

Bibliography

[1] T. G. Griffin and B. J. Premore, "An experimental analysis of bgp convergence time," in *Proceedings Ninth International Conference on Network Protocols. ICNP 2001.* IEEE, 2001, pp. 53–61.

[2] N. Matloff, "Introduction to discrete-event simulation and the simpy language," *Davis, CA. Dept of Computer Science. University of California at Davis. Retrieved on August*, vol. 2, no. 2009, pp. 1–33, 2008.

[3] G. Dagkakis, C. Heavey, S. Robin, and J. Perrin, "Manpy: An open-source layer of des manufacturing objects implemented in simpy," in *2013 8th EUROSIM Congress on Modelling and Simulation.* IEEE, 2013, pp. 357–363.

[4] M. L. Daggitt and T. G. Griffin, "Rate of convergence of increasing path-vector routing protocols," in *2018 IEEE 26th International Conference on Network Protocols (ICNP).* IEEE, 2018, pp. 335–345.

[5] A. Fabrikant, U. Syed, and J. Rexford, "There's something about mrai: Timing diversity can exponentially worsen bgp convergence," in *2011 Proceedings IEEE INFOCOM.* IEEE, 2011, pp. 2975–2983.

[6] A. Elmokashfi, A. Kvalbein, and C. Dovrolis, "On the scalability of bgp: The role of topology growth," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 8, pp. 1250–1261, 2010.

Appendix A First Appendix

Abbreviations

AS Autonomous System

BGP Border Gateway Protocol

DES Discrete Event Simulator

MRAI Minimum Route Advertisement Interval

RNG Random Number Generator