definition

Dept. of Information Engineering and Computer Science

Master's Degree in
Computer Science

Final Dissertation

Quieting the noise in the Internet Inter Domain Protocol

A simulation base study of BGPs noise reduction mechanisms and their interaction

Supervisors
Prof. Renato Lo Cigno
Prof. Timothy G. Griffin (Univ. of Cambridge)

graduating student
Milani Mattia

Accademic Year 2019/2020

Thanks

...thanks to...

Contents

Summary **3**

1 Introduction **4**

1.1 Internet Today . . . 4

1.2 Interaction between variables and convergence . . . 5

1.3 The goal of this thesis . . . 5

2 BGP state of the art **6**

2.1 BGP . . . 6

2.2 BGP inherent noise vs external noise . . . 8

2.3 BGP MRAI, designed to reduce inherent noise . . . 9

2.4 BGP RFD, designed to reduce external noise . . . 10

2.5 Topologies . . . 11

3 Discrete Event Simulator **14**

3.1 DES Environments . . . 15

3.1.1 Clique environment . . . 15

3.1.2 Fabrikant environment . . . 16

3.1.3 Internet-like environment . . . 16

4 Abstract Finite State Model of inherent noise **17**

4.1 BGP generalization . . . 17

4.2 BGP FSM experiments . . . 17

4.2.1 MRAI and BGP FSM . . . 21

4.3 BGP FSM explosion . . . 22

5 BGP MRAI dependency **25**

5.1 Different MRAI strategies . . . 25

5.1.1 Centrality based strategy . . . 25

5.1.2 Strategy comparison . . . 26

5.2 Clique Experiments . . . 26

5.3 Internet like experiments . . . 27

5.4 Strategy dependence . . . 28

5.5 Pareto Efficiency Front . . . 31

5.6 Signal dependence . . . 32

5.7 Position dependence . . . 33

5.7.1 Different signal sources . . . 34

5.7.2 Hierarchical influence . . . 35

6 BGP RFD **39**

6.1 RFD on toy topologies . . . 39

6.2 RFC 2439 VS RFC 7196 . . . 39

6.3 Mice VS Elephants . . . 39

6.3.1 Mice . . . 39

6.3.2 Elephants 39

7 RFD and MRAI Interaction 40

7.1 RFD on toy topologies 40

7.2 RFC 2439 VS RFC 7196 42

7.3 Mice VS Elephants 44

7.3.1 Mice 45

7.3.2 Elephants 47

7.3.3 MRAI influence on Mice and Elephants 48

8 Conclusion 50

8.1 Future Works 50

References 50

A Appendix 53

Abbreviations 61

Summary

...summary....

1 Introduction

With the name “The Internet” we define a network composed by more than 60 000 entities that share their knowledge in order to permits us to reach every website whenever we want. We are used to think about Internet as something far away from us, something that we do not have to care about, we can use it leaving all the complexity out. But, by a more physical point of view, what is the Internet? It is nothing more than a big network where interconnected entities keeps the prefixes reachable. Those entities are in reality called Autonomous Systems (ASes) and their function is to hold and control some Internet Protocol (IP) prefixes used by one or more operators.

Every AS is responsible for the connectivity to the prefixes that it shares. We know that the networks are able to react to changes thanks to routing protocols, and Internet is not different in that. Every AS has to keep active its own instance of an Internet routing protocol in order react to changes. This routing protocol is the glue of the Internet, its the thing that permits us to always be able to reach the other side of the world without knowing the actual route that our packets take.

The path that we use could change because of different factors that could be technical, economical or even political. Thats because the AS relationships are controlled by contracts and different contracts can have different fees applied for the transmission of the knowledge. Some paths may be used only as backups if the primary one fail, or even used only for certain type of traffic flows. This policies must be implementable in the Internet routing protocol that has to discern on which path, among all the known alternatives, is the best one considering the AS convenience.

The protocol that has been created to handle this situations is the Border Gateway Protocol (BGP). It has been released in the 1989 and is in use on the Internet since the 1994. It reached its last version, the fourth one, in 2006 [1]. Is easy to imagine that in almost 30 year BGP is changed a lot from the beginning and also the needs of the different ASes are changed a lot because of the technology improvements. Up to now, BGP can be expanded with tens of Request For Comments (RFCs) that improve the range of possibilities, those optional parameters are actually very important for the ASes because the complexity of the relationships is growth a lot in the last years.

BGP is an instance of the *Bellman-Ford* distance vector routing protocol that shares other than the prefixes known by the speakers also the path used to reach the destination. Other than that, to control all the possible policies applied by the ASes, BGP implements also different parameters and attributes that can be personalized. In BGP multiple parameters play a central role, and the correct setting of them could influence the performances of, not only the single AS but the entire network. For this reason, the research is still active to find new technologies and a trade-off between, convergence time and messages transmitted that could be sustainable by the current hardware.

In this thesis, more precisely in Chapter 2, I’m going to introduce two of those important BGP mechanisms, Minimum Route Advertisement Interval (MRAI) and Route Flap Damping (RFD). The first one is used to compress multiple input messages into one output message in order to reduce the network load provoked by a change. The second one, RFD, is used to penalize unstable paths, suppressing the route and blocking the spreading of it to further nodes to circumscribe the zone of instability.

The existence of those two mechanisms has been studied separately many different times [2–5]. But, there are almost no studies on the interaction of them. Even if the effects of one interact with the other.

1.1 Internet Today

Internet, as a network, is constantly growing, in terms of ASes, prefixes and messages transmitted. This continuous growth increase also the load on the BGP nodes that receives more messages and have to manage the effects in terms of memory and processing power. By consequence, this increases also the load on the network, because of the nature of BGP to act as echo-chamber.

Thanks to the annual report from Asia-Pacific Network Information Centre (APNIC) we can have a snapshot of the situation of Internet and the evolution of it. The data collected by APNIC from 2007 concern the knowledge of the AS 131072 that has two links with other ASes, one in Japan and the other one in Australia[1]. Prefixes of smaller and smaller sizes are continuously shared, the number of /24 networks distributed in the last year has been growing constantly. Fomenting the problem described above, generating a vicious circle of messages.

This redundancy of the BGP nodes provoke the *Path Exploration* problem. This particular issue occurs when a node enter in a transitory state where it continuously shares non optimal paths while it doesn't reach a stable state. Provoking the propagation of non ideal routes to other nodes causing a vicious circle. The growing of Internet is not negligible because of this problem, a continuous growth in terms of nodes and edges cause the growth of favorable conditions for the *Path Exploration* problem.

1.2 Interaction between variables and convergence

The two parameters that will be studied in this thesis are MRAI and RFD and how the interaction between them works. Our first hypothesis is that, indeed, there is an interaction. This hypothesis is sustained by the fact that both parameters operate to reduce the noise of BGP, with different parameters and different behaviour, but, if a node wants to transmit a message it must respect MRAI and the input could be caused by RFD that suppress/reintroduce a route. On the opposite case, a too small MRAI value could permit different message storms that would trigger the RFD suppression systems creating vicious circle.

One of the goals of reducing the value of MRAI is to reach a faster convergence paying the cost of more messages. Unfortunately, looking only to MRAI is not possible to get reliable results for general purposes, in-fact, like showed in Chapter 7, is possible to obtain the opposite result due to the fact that to solve RFD suppression is required a longer time.

On the opposite case if we tune in the wrong way RFD we would end up to be too much permissive, leaving to handle all the noise to MRAI that could be not effective if the storms are sufficiently delayed in time.

For those reasons is important to study these two parameters together, because there could be a strong co-dependence.

1.3 The goal of this thesis

The goal of this thesis is to prove that the noise reduction mechanisms of BGP interact one another, studying this interaction through simulative experiments and from those give useful hints on how those two parameters interact with one another. In order to build the basis for future experiments that can study more deeply and maybe in a formal way the phenomena. Is also mandatory for this thesis to develop the platform where those experiments would be executed and make that platform public available[2].

In Chapter 2 are going to be presented the protocol and more deeply the two parameters studied, while in Chapter 3 I'm going to present the structure of the platform that will be used in Chapters 4 to 7 to perform the experiments about the *Path Exploration* problem and then how MRAI and RFD can impact the performances, conclusions to follow in Chapter 8.

The BGP community has not yet reached a common agreement on what values to use for this reason I will evaluate different possible techniques that can be applied to both the parameters and comment on the network performances obtained.

[1] source APNIC data

[2] GitHub repository

2 BGP state of the art

BGP is the protocol used to control the information spreading on the Internet. It is at the version 4 published in 2006 with the RFC 4271 [1]. BGP is a Path vector (PV) protocol, it distinguishes itself from the Distance Vector (DV) and Link State (LS) protocols with the major difference that it shares other than the knowledge of a path also the path itself to reach the destination.

BGP has two major sub-categories with a difference in the flow of information direction:

- ***Exterior BGP (eBGP)***, we talk about eBGP when the flow of information goes from the inside of the AS to other ASes;
- ***Interior BGP (iBGP)***, we talk about iBGP when the flow of information goes from the outside of the AS to the inside of it, to make aware of the new route also the internal routing protocol.

My main interest in this thesis is for the eBGP part of the protocol, for now on I will refer to it talking in general of BGP without further distinguishing from the internal protocol.

When there is an interconnection between two ASes that creates a BGP link, I will talk about peering referring to the connection, and those BGP speakers interconnected will be the peers. Each BGP link is based on a direct Transmission Control Protocol (TCP) connection. On these links, every AS can configure its own policies that would be used to evaluate routes at the reception or in the moment there is something new to share. There are three different possible type of relations that can be created by two BGP speaker, accordingly to the Center for Applied Internet Data Analysis (CAIDA):

- ***customer-to-provider (c2p)*** This relationship highlights the fact that lower AS pays a higher level AS to get connectivity and access to the Internet;
- ***peer-to-peer (p2p)*** This relationship is used to share the knowledge between two ASes of their customer providers without paying a higher level AS;
- ***sibling-to-sibling (s2s)*** This relationship defines the connection between ASes under the same Internet Service Providers (ISP).

During this thesis, I will only consider the first two relationships c2p and p2p, the schema in Figure 2.1 shows how the traffic is affected based on the type of the link crossed.

As shown by the flows with a different colour in Figure 2.1 a single AS will share information considering the receiving link. If something comes from one of its customers it will share the knowledge with every other link that it has, even other customers (always respecting the output policies). If a route has come from its provider or a peer then it will be only shared with its own customers. Those policies are commonly called "valley-free" and are dictated by convenience, an AS has all the advantages when other ASes decide to use it to reach a specific destination. For this reason is convinient for an AS that everyone knows about its clients networks, and, in the opposite side, that its clients knowes only the route through it to reach other networks.

This behaviour can be modelled with a variant of the Stratified Shortest Path (SSP) algebra described in [3]. This is the same algebra that will be used in Chapter 3 to describe the links relationships.

2.1 BGP

Once a BGP node has established a connection with another peer it will start to exchange routes with that neighbour, always respecting the policies. Its important to underline that, a BGP speaker

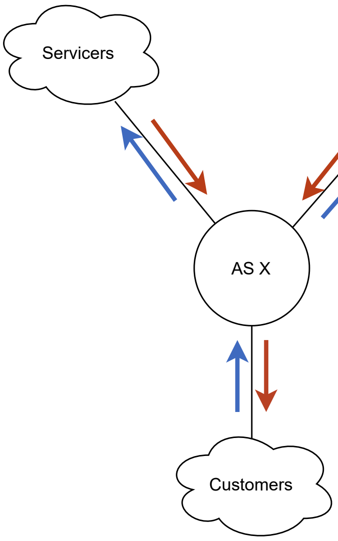


Figure 2.1: Distribution schema for the ASes, the row colours distinguish different flows, *AS_X* is a customer of the servicer set, a peer with the peers set and servicer for the customers set

only shares its best routes and in this protocol the best decision is dictated by the policies, and then other possibilities would be evaluated (number of hops, bandwidth etc). For this reason the best path decided by a node could differ from the actual best path from a topological point of view.

Every BGP node has a Routing Information Base (RIB) as data structure to keep the information about the received routes, the alternative routes and what should be exported. The RIB is divided into 3 sections.

- ***ADJ-RIB_in*** This RIB contains all the routes that have been received by other AS in order to be evaluated;
- ***LOC-RIB*** This RIB contains all the best routes that have been chosen from the *ADJ-RIB_in* from the node;
- ***ADJ-RIB_out*** There is an output RIB for every neighbour of the node, it contains the route that should be advertised to the specific node.

One of the most important parts of BGP is its decision process, that would be applied to discern between the routes in the *ADJ-RIB_in* in order to update the *LOC-RIB* and, if necessary also the *ADJ-RIB_out*. The decision process is composed of three parts:

1 Calculation of the preference: This function is called every time there are new reachability information that needs to be evaluated, it will assign/update the preference value at every route in the *ADJ-RIB_in* using policy filters pre-configured. If a route doesn't respect the policy filters it will be then marked as ineligible, otherwise, a *PREF_VALUE* will be calculated and assigned to the route.

2 Route selection: This function is called at the end of the first phase, it collects all the eligible routes and evaluates them removing routes that would create loops or that creates conflicting situations. The evaluated routes are then ordered by the *PREF_VALUE* and then the best route will be then installed in the *LOC-RIB*. In case of ties, there is an algorithm that can be used to break them.

3 Route dissemination: This function can be called in different situations, it will use the information in the *LOC-RIB* to populate every *ADJ-RIB_out*, according to configuration policy.

The decision process is also responsible for the route aggregation and information reduction. At the end of the third phase, the BGP speaker will execute the *Update-send* process, that is responsible for the effective dissemination.

There are multiple types of packets that can be sent by a BGP speaker, but we will focus only on the advertisement (ADV) packets. The ADV packets are responsible for the dissemination of the information to other nodes that will analyze and use the attribute inside the message to assign a preference value to the route. In particular, there are two sections of the ADV messages that will contain additive information and subtractive information. We can distinguish ADV messages using the type of information that are transmitting:

- ***UPDATE***, this type of messages represent the distribution of new reachability information, a new best route to a destination will be shared through an update message;

- ***WITHDRAW***, this type of messages are distributed when a node want to share that it doesn't know how to reach a destination anymore.

Inside those packets, there are different attributes that permit to transfer information about the route (advertised or withdrawn). There is an attribute that describes the address that the route represents, another one that contains the path that will be used to reach the destination, the next-hop used, etc. During the years multiple new RFC have introduced, modified, updated and removed attributes that can be found inside an advertisement message. Not all the attributes are mandatory for BGP nodes, in fact, for a node is possible to receive a route with an attribute that it is not able to interpret but (if configured to do so) it will share the route with also the unknown attributes.

Is important to remember that all those information are useful for the policy filters that every node can have, for example, some nodes would automatically discard any route that contains a specific AS in the received path.

The *Update-send* process is responsible for the distribution of the messages that are stored in the *ADJ-RIB_out*. It execute again some checks on the RIB, removing unfeasible routes and removing routes that have already been advertised to the pear. It also has to respect a temporal constraint, introduced in [1], a BGP speaker can't send to the same neighbour routes for the same destination more often than the MRAI value. MRAI act as a timer whose goal is to avoid continuous update storm caused by decision changes in some peers in the network.

Another property that can be found in BGP nodes that affects the messages transmitted is the Implicit Withdraw (IW). This property permits to reduce the number of messages that are distributed. Without this option when a BGP node discovers a new best path to reach a destination should send a withdraw followed by an update to its neighbourhood. Thanks to this option is sufficient to send just the update, the other nodes will learn that the best path is changed simply looking to the previous alternative and comparing them.

2.2 BGP inherent noise vs external noise

With the term *BGP Noise* I would like to define a particular behaviour of BGP. It underline a situation where the nodes are distributing routes that would be retrieved few moments later. Producing an intense distribution of messages that are not optimal. This behaviour is defined as noise because the tendency of BGP to act as echo-chamber and amplify this distribution of incorrect information. In fact, a BGP node that receive an update for a neighbour will probably redistribute the route to multiple neighbours, peers and servicer. Given the hierarchical topology of the internet this behaviour grows exponentially while the information reach the center of the network.

There are basically two sources of noises in BGP, the inherent noise of the protocol and the noise caused by external sources. Those two type of noises are triggered by different causes and are not discernible one another.

The first one is caused by the protocol itself when it tries to converge on new knowledge. The sharing of new routes can cause new ADV that can then trigger the *Path Exploration* problem. This is a noise caused by the protocol itself that acts as echo-chamber for new best paths that change until the best possible path is taken in consideration.

One BGP parameter tries to limit this noise acting as a message cache with a compression system. This parameter is MRAI and it permits to avoid sending a message for every new one received using a timer. Only the best decision after that time will be shared.

The second type of noise is caused by a source outside the protocol. A miss-configuration or a faulty interface can cause the send of not necessary messages. For example, the withdraw and the advertisement of a route at a continuous interval. This type of behaviour will cause continuous storms of messages and the triggering of the first noise type. Also, the transmission of a withdraw followed by an announcement is considered a “flap”.

BGP introduce a parameter with the RFC 2439 [6] that is called RFD to overcome this behaviour. This parameter increases a value every time a flap or a route change are detected, when this value passes a predefined threshold the route will be suppressed. This value will always decay using an exponential decay function, even if it doesn't overpass the threshold. The decay function is calculated defining the time that the function should take to half the value, by default 15 min. Once the route has been suppressed the BGP speaker must wait enough time that the value goes below another threshold before sharing it again.

Those two parameters are clearly connected one another from the fact that one triggers the other and vice versa, is possible to create a particular topology that has different performances based on the values assigned to those parameters. Think about two clique networks connected one another by only one node, that node will act as a bottleneck, probably its RFD threshold would be easily overpassed and then it depends on its decay function before it can send again the network to the second clique triggering MRAI on those nodes that will experience the path exploration problem.

2.3 BGP MRAI, designed to reduce inherent noise

MRAI is one of the parameters that mostly affect the convergence of BGP. A high value of it could unnecessarily delay the transmission of messages, but, in the opposite case, a value too small can provoke a lot of messages, one for every decision change of the node. The main function of it is to reduce the intrinsic noise of BGP compressing the messages received. There are a lot of studies about it, and it has already been shown that the number of messages and the convergence time can depend on it [7]. Also, it has been already proven by Fabrikant, Rexford et al. [2] that an incorrect configuration of it could lead to tremendous consequences.

MRAI has been introduced in the 4Th version of BGP [1] and it is nowadays a mandatory mechanism for every BGP node, otherwise the load in terms of messages to process and decision changes would be incalculable. Its main purpose, as anticipated in Section 2.2 is to prevent, or at least mitigate the noise created by BGP itself.

At the base of MRAI, there is a timer that controls how much time must pass between one ADV and the following one. The timer is peer-based, for each interconnection an AS could have a different MRAI, but it acts for every destination in parallel, this means that there is a different timer for each destination that a node would share for every BGP relations that it has. The idea behind it is that, in the period of time between one ADV and the following one the BGP node will be able to receive other possible routes. Being able to evaluate them and transmit a decision considering multiple alternatives. It has the property to compress the input messages sequence in order to have an output message sequence with a smaller number of ADV.

The behaviour of a BGP node with MRAI is defined as follow for every change in the *ADJ-RIB_out* of a neighbour caused by the *Update-send* process:

- If there isn't an active MRAI timer for the destination changed send the ADV and set an MRAI timer.
- If there is an active MRAI timer for the destination then don't send anything.
- When the active MRAI timer ends if there is still the necessity to send an ADV then send it and set another MRAI timer.

The second passage permits the route selection process to be executed multiple times before the actual transmission of the decision. That because MRAI limits only the transmission and not the decision process. The condition to the last passage is due to the fact that the compression some times could lead to the not necessity to actually send a message.

The default value defined in the RFC 4271 for MRAI is equal to 30 s. But, MRAI is a really controversial parameter, it has received multiple revisions and studies. In 2008 there has been a proposal to reset its default value to 5 s [8] thanks to different studies that take into consideration the dimension of the topology and the latency [4]. In 2010 a proposal RFC of the Internet Engineering Task Force (IETF) [9] says that the default would be left to the arbitrary choice of the operators and that withdraw message could completely avoid it. But this freedom would damage the convergence and the number of messages distributed as showed by Fabrikant et al. [2].

Is clear that MRAI affect the network performances, but what affects MRAI and, by consequences, the performances? Obviously, the choice itself of a different MRAI strategy as showed for example in [10], where the centrality has been used to obtain better results in the case of network faults. But, giving the fact that, the main function of MRAI is to compress the input messages also the sequence of messages receipt could be meaningful. Giving that MRAI is a link-based parameter also the number of links that a node has could influence it and by consequences the position in the topology. A well connected node will be more likely to receive multiple paths and messages than one with only one link.

2.4 BGP RFD, designed to reduce external noise

RFD is a parameter introduced in BGP to overcome the problems caused by the exterior sources of noise. Its main function is to avoid fluctuating routes to overload BGP nodes with continuous message storms. It has been introduced with the RFC 2439 in 1998 [6]. Also, RFD is a controversial parameter, it has been studied and evaluated again different times, but, recent studies showed that the majority of the operators still use deprecated values from 2001 [5]. Furthermore, other studies show that the majority of the ADV that travels through the Internet are generated by a restricted set of ASes but RFD seems to be too much restrictive and affect the majority of the ASes traffic [11].

RFD will use a single value, called *figure of merit*, to evaluate the actual situation of a route, while this value evolve the RFD algorithm will make a decision on what to do. The evolution of the *figure of merit* is dictated by the messages received, with fluctuations, it will grow, while, over time, it will use a quadratic decay function to decrease. Fluctuations, or flaps, are represented by the reception of the withdraw and the announcement of a route, a path change is also considered a flap, even if thanks to IW is limited to just one ADV.

There are other parameters that are part of this BGP component, the more important are presented in Table 2.1.

Parameter	Cisco default values
withdrawal penalty	1.0
re-advertisement penalty	0.0
attribute change penalty	1.0
suppress threshold	2.0
half-life (min)	15 (900s)
Reuse Threshold	0.75
Max Suppress Time (min.)	60 (3600s)

Table 2.1: RFD parameters

Other than the name of the parameters, in Table 2.1 is showed also the default value decided by Cisco. The RFC 2439 [6] gives some guidelines on how to set those values but the actual choice is left to the discretion of implementors. The first three parameters, *Withdraw, re-advertisement, attribute change* represent the penalty applied to the *figure of merit* when the homonym event happen. The *suppression threshold* represents the level at which the BGP node will suppress the route and don't advertise it until the figure of merit goes below the *reuse threshold*. The decay of the *figure of merit* follows a quadratic decay function which rate is calculated using the *Half-life* parameter. the *Max Suppress Time* will override all this parameters because, as defined in the original RFC, a route cannot be suppressed for more than that time, it doesn't matter the figure of merit accumulated by this route.

An example of the figure of merit evolution could be seen in Figure 2.2, This image has been taken from [5].

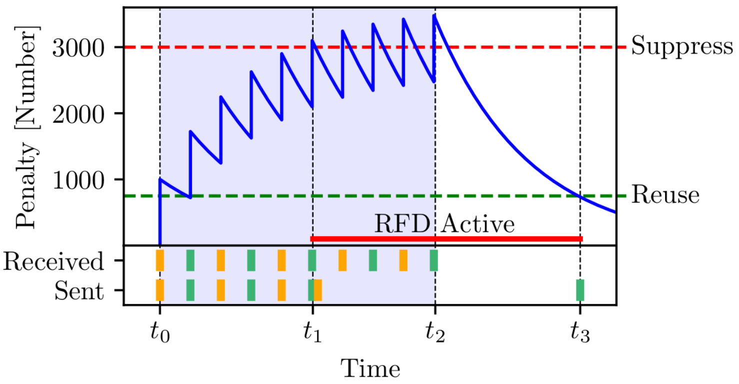


Figure 2.2: Example of evolution of the RFD *figure of merit* taken from [5], yellow messages represent withdraws and green ones are advertisement, the dashed lines are the suppression and reuse threshold

Figure 2.2 shows a hypothetic evolution of the RFD filter, it doesn't rely on the default value of the Cisco implementation. Is possible to see in the lower part of the plot the messages received by the BGP speaker, yellow one represent withdraws while the green are announcements. Is possible to see that the penalty value grows at each flap and as soon it reaches the suppression threshold the route will not be advertised to any neighbour. While after the decay has reached the reuse level is possible to advertise the route again.

Is possible to see that RFD doesn't make any difference on its own on what is causing the flaps, it simply reacts to the actual situation of the network. Is not even possible to determine where is located the flap, if the source is flapping heavily for some reasons or there is an AS in the middle of the path that is malfunctioning.

RFD has a troubled history, maybe even more than MRAI. In 2006, thanks to the publication of [12], the Réseaux IP Européens Network Coordination Centre (RIPE) recommends to disable it [13]. A few years later the publication of the article from Pelsser et al. [11] RIPE and IETF shares that now RFD should be used with the updated parameters [14, 15]. Unfortunately, the study from Gray et al. [5] in 2020 shows that the majority of the AS uses RFD with outdated parameters from the RFC 2439.

RFD can influence the network convergence time, and this behaviour is mostly impacted by the threshold and the decay function applied to the figure of metit. A more permissive threshold could be helpful in situations where the flaps are caused by transitional behaviour. I will study and analyze the impact of the threshold in Chapter 6.

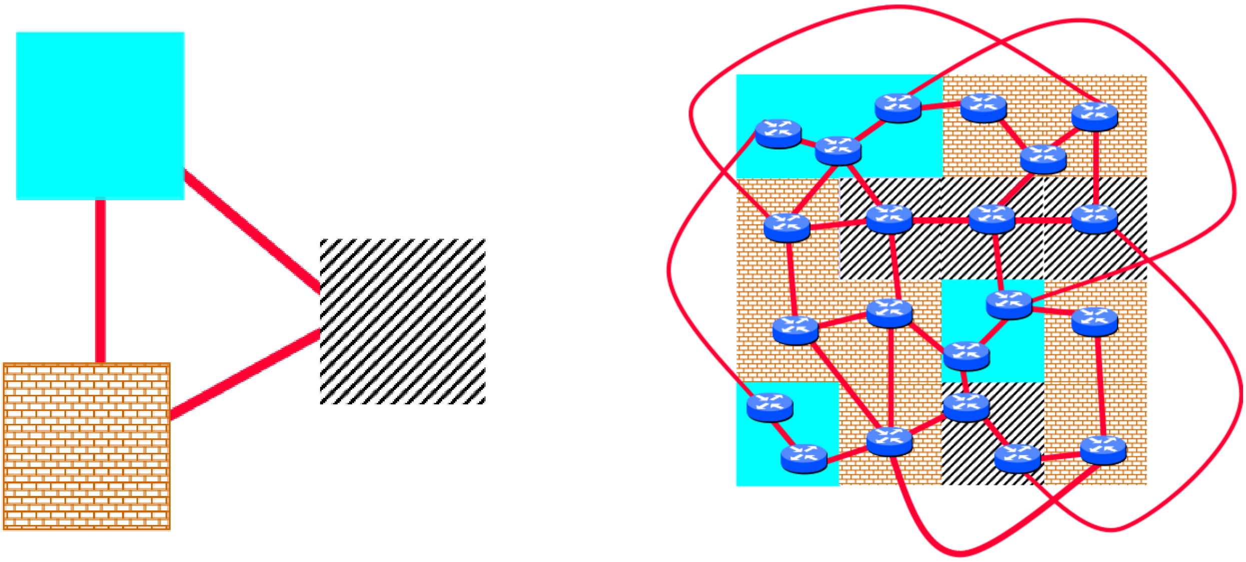
2.5 Topologies

BGP has been designed to throw away information. Is important to remember that the actual topology of the Internet depends on the level of abstraction required. If we consider the graph of the relationships within the ASes is not possible to assume that this also represents the geographical graph. In fact, one AS can have multiple connections with another AS that are distributed along different geographical points. Is also possible to have a connection through a tunnel that will permit to have a connection that physically passes over other devices. Is possible to see an example of this distinction in Figure 2.3

In this thesis I will consider only the first case, all my topologies are composed by the connections between the ASes without considering the actual physical layer where this interconnection resides.

I will then use three different topologies to show particular situations or behaviours of the network with certain events.

- ***Clique topology***, This topology represent the higher level of the internet, in fact, all the Tier one nodes are interconnected in a clique network;



(a) AS Graph

(b) Actual geographical topology

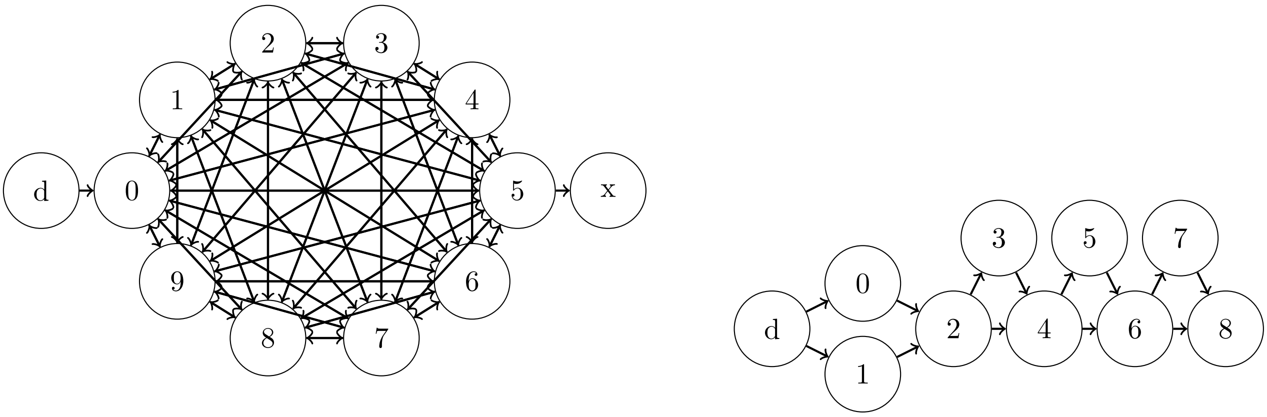
Figure 2.3: Differences between two levels of detail on the Internet graph

- ***Fabrikant topology***, This is a particular topology because it represent a special case that can be present in the internet, and also is possible to show that special configurations of this topology can lead to difficult situations.

- ***Internet-like topology***, This kind of topologies have the goal to resemble the real internet using statistical information about it.

I choose the clique topology because it is one of the possibly worst case scenario for a BGP network, accordingly with [16]. A clique graph could be composed by an n number of nodes, and every node has $n-1$ edges towards every other node. An example of clique topology can be sawed in Fig. 2.4a, I added two nodes to the network, outside the clique, in order to be able to have an input and an output point.

The Fabrikant topology is inspired by [2] where is used a similar chain network to show, in a theoretical way, how an uncontrolled MRAI setting could lead to explosive situations. This topology will be used for the same purpose in this thesis, an example of it is showed in Figure 2.4b. The explosion is caused by the *Path exploration* problem, at a certain point the node 2 would change idea on which path to follow to reach d and all nodes will go through a transition state where they will share their backup path.

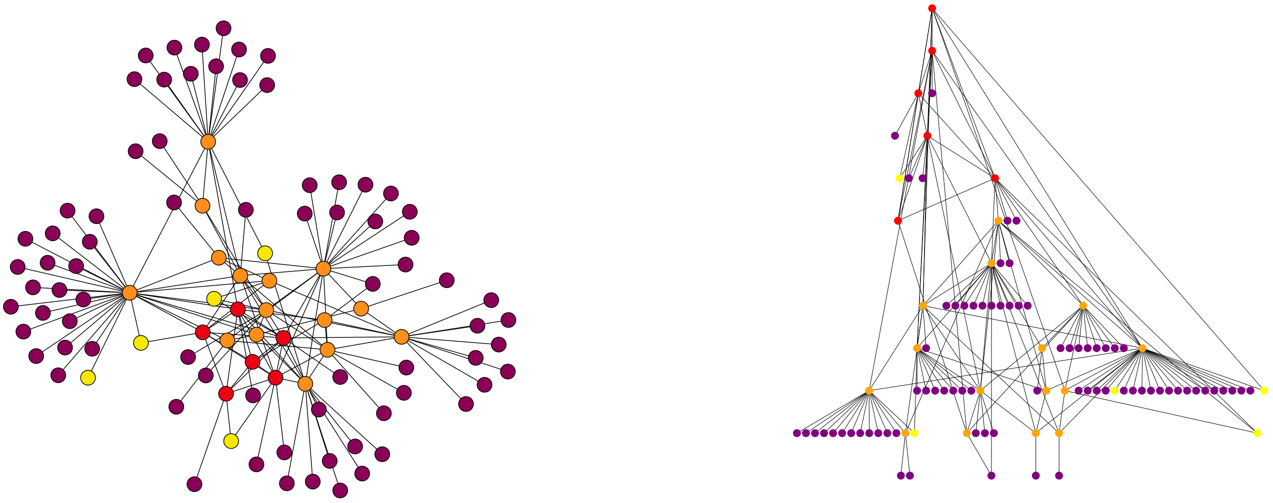


(a) Clique graph example

(b) Fabrikant chain graph example

Figure 2.4: Simple topologies used in the thesis

The last type of topology used points to resemble the properties of the actual Internet graph. It uses the property studied and described by Elmokashfi et al. in [17]. An example with few nodes is available in Fig. 2.5. The internet graph is actually a hierarchical graph clearly separated in different levels by the type and properties of the nodes.



(a) Internet like graph with an “explosive”

(b) “Hierarchical” Internet like graph

Figure 2.5: Internet like graph coloured to show the hierarchical structure, 4 types of nodes, T (tier 1 mesh), M, CP, C (Customers, the purple one)

3 Discrete Event Simulator

Experiments on BGP are not applicable on the actual Internet, for this reason different studies show their results using a simulate environment [7] or experiments executed on a testbed [18]. In the simulations the majority of the studies use small graphs and each node of the graph simulate the behaviour of a BGP speaker. Each node represents a single AS and the BGP speaker used as 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 simulative environment. I decided to use the *Simpy*[1] package to create the environment and make it evolve. This package present an extensive documentation and it has already been used for different studies, demonstrating its adaptability [19, 20], other than an active community.

I developed the DES as a highly modular environment.

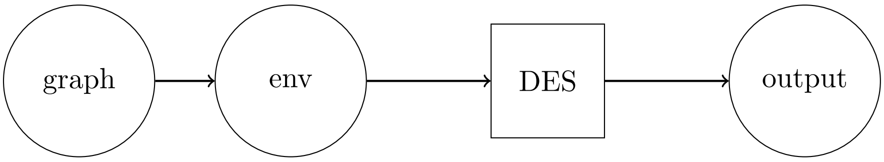


Figure 3.1: Discrete event simulator structure

In Figure 3.1 is possible to see the basic idea behind the simulator. The first component needed is a graph, represented by a *graphml* file, this file is the descriptor of the network. It defines also all the topological information and properties of every single node. 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. The policies are encoded using the convention described in [3]. In the graph file Is possible to define multiple topological properties, for example the MRAI value in an edge or the set of parameters that controls RFD.

```
<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 is a *JSON* file that describes how the environment is characterized. Inside of it is possible to define the initial values for the Random Number Generator (RNG) so that each experiment is replicable. In the environment file is also possible to define different properties that would be used to make the experiments evolve, for example the distribution that will be used for the network delay. Is also possible to define multiple values for single properties and decided which combination of them to use before the experiment starts. There are two possible evolution of the environment that can be used:

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

[1] Simpy website

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

The DES take as input this *JSON* that describe all the information, it creates an object for each node in the network, respecting the topological characteristics defined in the *graphml* file. After the initialization, all the nodes that contain a destination will schedule the first advertisement to their neighbours. And then, all the nodes can evolve using the distributions and the properties defined in the environment. 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. For each event the DES will record the exact moment when it happened and the associated values, for example for the reception of a message would be registered the node that received the message and the message itself in plain text.

3.1 DES Environments

Thanks to the environment codification in a *JSON* file is possible to define experiments with a high grade of freedom. Some elements can be defined as vectors of properties, those vectors can contain n objects that describe that property, but in each run just one of them can be used. Each run is composed by a combination of one object from each vector. For example, if we have 5 different possible seeds and 3 different delays, the total number of runs combinations is 15, as shown in Code 3.2. If a property has only one object, then, this object will be used in every combination. If all the properties has only one object then the environment describes only a single run. A unique identifier is associated to every possible run, through it is possible to execute specific experiments.

```
"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, no messages are lost during transmissions. 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 in the future works.

During the thesis I used different standard environments to study different properties and behaviours of BGP network, the main ones are described in Sections 3.1.1 to 3.1.3.

3.1.1 Clique environment

One of the special environment that I defined in my experiments uses clique graphs of different dimensions, an example of the clique graph is given in Figure 2.4a. This environment points to produce a high load of messages that each node has to process in order to take the best decision on how to reach the single destination distributed.

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, it also gives the possibility to study how BGP parameters can influence the messages distribution in stressful cases like the clique one. I'm going also to study how the variation of those parameters from the standard can impact the performances in this environment of high load.

3.1.2 Fabrikant environment

Another interesting case to test the path exploration problem is the one presented in [2]. In that study, Fabrikant et al. presents how particular MRAI setting could make the network converge with an exponential behaviour. The continuous decision change in the nodes RIB is the cause of the problem. I used the basic example of their study to investigate how the choice of MRAI is fundamental for the network convergence. An example of the network used is presented in Figure 2.4b.

The path exploration problem is caused by the delay on the 0-2 edge. The node 2 will receive the destination through node 1, after a small amount 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 reconfiguration of all the other nodes that will use the backup links for a while, announcing their new paths. A wrong configuration of MRAI can provoke the entire exploration of the possibility set.

This environment is also used to show how a BGP Finite State Machine (FSM) would explode in terms of possible states and edges producing an enormous amount of output signals from the same input. This behaviour is presented in Section 4.3.

3.1.3 Internet-like environment

The last noteworthy environment is the one whose purpose is to simulate the Internet behaviour. This has been possible thanks to the study by Elmokashfi et al. [17] 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 2.5.

The different nodes are coloured accordingly with the node type represented. The tier one nodes that generate the central clique are coloured in red and is possible to notice in Figure 2.5b that they are in the highest levels of the networks. This environment has been used to study the behaviour of the network with topologies resembling the real internet.

In the next studies, I generally refer to this kind of topology with the terms "Internet-like", and the dimension is never less than 1000 nodes, otherwise is not possible to ensure all the Internet topological properties.

The goal of the experiments that use that kind of environments is to study the general network performances in terms of average convergence time and the number of messages transmitted. This would help to see and study behaviours that are more typical in the actual Internet.

[2] Networkx internet as graph generator

4 Abstract Finite State Model of inherent noise

The incremental nature of BGP suggests that is possible to study the evolution of the protocol looking to the events that have been triggered and the causes of them. A model that can give this sort of information would be helpful to debug the protocol, analyze faulty situations or prevent them. The main idea behind this model of the protocol has been taken and expanded from [21]. I'm going to show that inferring information from BGP events and actions is more difficult than expected. In fact, due to the *Path Exploration* problem nodes will experience an explosion in the number of possible evolutions, presented in Section 4.3. In addition to that, multiple inputs can lead to the same output of the node, for this reason, it is not enough to know it to infer a precise input signal, showed in Section 4.2. Furthermore, MRAI can make the situation even more complicated, increasing the number of transitions because of all the new possible combinations of inputs.

4.1 BGP generalization

The main idea behind the BGP FSM is to represent the knowledge as states and a set of messages as transitions. The knowledge is represented by the actual routes that the node knows on how to reach a single destination, we will have a different FSM for every destination. Transitions encode the messages that a node has received to trigger the state change, on the edges are also inserted the response messages that the node will transmit. We can see an example of these transitions in Figure 4.1

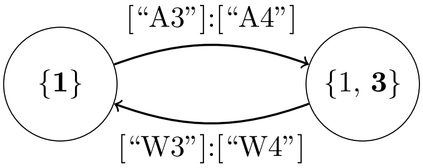


Figure 4.1: Example of the BGP FSM state transition

In Figure 4.1 there are two states, both represent the knowledge of the node, the first one represents the RIB with just the route 1, in the second state the RIB will contain both the routes 1 and 3. This transition is caused by the reception of the advertisement of the route 3 and will cause the transmission of another advertisement. The opposite transition is caused by the reception of the withdraw of the route 3 with the consequent withdraw of the route 4. The route which id is bold represent the actual best path chosen by the node, in this case the reception of "A3" provoke the change of the best path, noticeable in the second state where the route 3 is bold.

Thanks to MRAI the evaluation of multiple messages could be delayed and provoke then the compression of them. For this reason on the edges is possible to see multiple messages, for example "[A1W1A1]", that will be compressed in "[A1]" and then evaluated. The output part of the transition will be influenced by MRAI, the output message strongly depend on the number of messages that the node was able to compressed.

4.2 BGP FSM experiments

The first experiments that I executed are about the translation of a single node evolution in a FSM. The goal is to reproduce what has been shown in [21]. The graph used for the study is presented in Fig. 4.2.

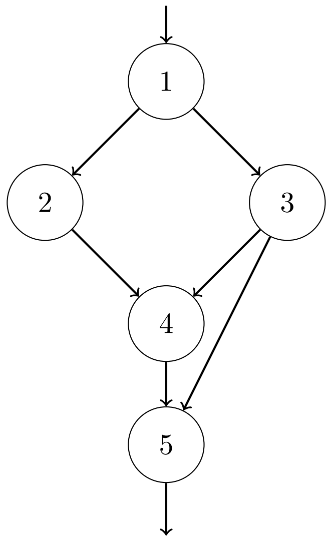


Figure 4.2: Graph from fig 4 of [21] used to study the FSM of the nodes

This topology, Figure 4.2, present a Stable Paths Problem (SPP) with five node [22]. The SPP model is used to eliminate much of the complexity of BGP. The arrows in the graph represent the flow of information, node 1 is the one that will receive a new route to reach a hypothetical destination and it will spread this information using an ADV towards all its neighbours. After a while node 1 will also distribute the withdraw of that route. The translation to the Communicating Finite-State Machine (CFSM) will use an enumeration to encode all the paths that a single node will encounter, for example, the path "5 3 1" will be converted in *a3*, each path has its own identifier. In case of withdrawing the route will be encoded as *w3*.

The properties of the environment for this experiment are listed in Table 4.1.

Property	Value
Seeds	$[1, 50]$
Signaling	"AW"
Withdraws delay	Uniform distribution between 20 s and 30 s
Announcement delay	Uniform distribution between 20 s and 30 s
MRAI	0 s for every link
Link delay	Uniform distribution between 0.001 s adn 1 s, uniform distribution between 0.012 s and 3 s
IW	Active

Table 4.1: FSM example environment properties

The total number of runs generated by this environment is 100 (50 seeds and 2 possible delay distributions). MRAI has been setted to 0 s in order to be ininfluent during the simulation.

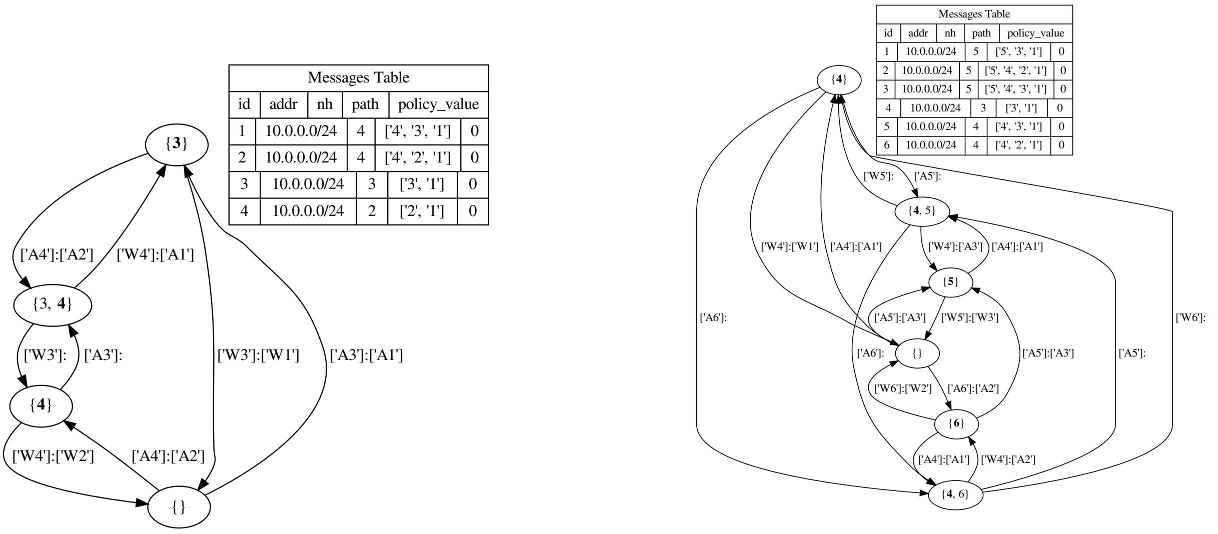
The two nodes we are most interested are node 4 and node 5, because of their position after a fork. The first one is going to receive the messages of node 2 and 3, for sure there will be two announcements and two withdraws because of the signal imposed on node 1. Node 2 will share an announcement followed by a withdraw and the same node 3. Those messages can be reordered in different way, always respecting the local order. For each sequence the node 4 can take different decision, for example use the path though node 3 and then change its decision using the one received from node 2. Is possible to see all the possible combinations of inputs and outputs of node 4 in Table 4.2. The messages from node 2 and 3 would have a single unique id, because both the nodes only know one path to the destination. While, node 4 has the possibility to propagate two different paths, each one with a unique id. The output signals of node 4 would be part of the set of all the possible inputs of node 5.

Input signal	Output signal
$a2a3w2w3$	$a4a5w4$
$a2a3w3w2$	$a4w4$
$a3a2w2w3$	$a5a4a5w5$
$a3a2w3w2$	$a5a4w4$
$a2w2a3w3$	$a4w4a5w5$
$a3w3a2w2$	$a5w5a4w4$

Table 4.2: Node 4 different possible inputs and output

The node 5 is going to receive all the possible outputs from node 3 and 4. The number of input signals of node 5 is composed by all the possible combinations of messages in the output set of node 4 and 3. The total number of possible inputs for node 4 is 6, like showed in Table 4.2, for node 5 is going to be 71. In Table 4.2 is possible to notice that there are no inputs that produce the same output, thats not true also for the node 5. In fact, node 5 has in total 52 unique possible output sequence, given by all the 71 different inputs.

From the 100 total runs, we can generate the CFSM of node 4 and node 5, in order to be able to study how the nodes react to different input signals. The two CFSM are presented in Figure 4.3.



Messages Table

id	addr	nh	path	policy_value
1	10.0.0.0/24	4	['4', '3', '1']	0
2	10.0.0.0/24	4	['4', '2', '1']	0
3	10.0.0.0/24	3	['3', '1']	0
4	10.0.0.0/24	2	['2', '1']	0

(a) Node 4 CFSM from the environment of Table 4.1

Messages Table

id	addr	nh	path	policy_value
1	10.0.0.0/24	5	['5', '3', '1']	0
2	10.0.0.0/24	5	['5', '4', '2', '1']	0
3	10.0.0.0/24	5	['5', '4', '3', '1']	0
4	10.0.0.0/24	3	['3', '1']	0
5	10.0.0.0/24	4	['4', '3', '1']	0
6	10.0.0.0/24	4	['4', '2', '1']	0

(b) Node 5 CFSM from the environment of Table 4.1

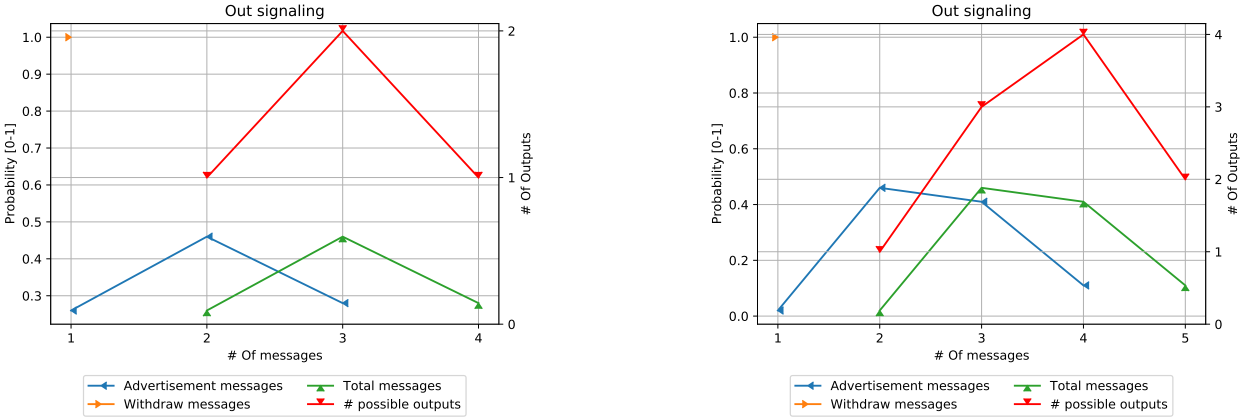
Figure 4.3: CFSM of nodes 4 and 5 of the graph Figure 4.2 with an input signal of "AW", 100 total different runs, MRAI ininfluent. FiXme: Remove message tables?

The states of the CFSM in Figure 4.3 are represented by the knowledge of the nodes, composed by the routes that are in the RIB of the node. The bold value is the actual best route to the destination chosen by the node. If in the state transition to a new state the best path is not affected then the node will not transmit the new route to its neighbours, for an example take a look to Figure 4.3a from the state $\{4\}$ to the state $\{3, 4\}$ where the node 4 will learn a new route that is not the best one.

The effects of the implicit withdraw can be seen in Figure 4.3b the transition from $\{4, 6\}$ to $\{4, 5\}$ thanks to the reception of the announcement $a5$ from the node 4.

As written in [21], I would like to underline the fact that, given the 52 unique possible outputs of the node 5 it would be very difficult to infer the initial signal that provokes all the transitions.

We can also analyze those output signals, knowing all the events for each single run we can infer which were the most common output signals experienced by a single node. Is sufficient to take all the transmitted messages of a node and look the sequence of advertisement and withdraws.



(a) Node 4 output signals study

(b) Node 5 output signals study

Figure 4.4: Output signal study of nodes 4 and 5 of the graph Figure 4.2 with an input signal of “AW” at node 1, 100 of total runs, MRAI ininfluent

The plots in Figure 4.4 represents the probability of an output signal of a certain length to be detected and the number of unique output signals that have been found for each possible length. The x axis represents the number of messages in the output signal, a message is a single announcement or withdraw. The first y axis represents the probability to see a certain number of messages taking a random output signal from the set. Three lines refers to this axis, the blue one represents the number of advertisement messages in the output signal correlated with the respective probability. For example, in Figure 4.4a there is a probability around 0.45 to have exactly two advertisement messages per output signal. And respectively a probability slightly larger than 0.25 to have only one advertisement or three. We can also notice that we didn’t see more than three advertisements or less than one. The green line instead represents the total number of messages in the signal, without distinguishing between advertisement and withdraws. By the fact that we will always experience one withdraw (the orange line) the green line is simply shifted by one unit in respect of the advertisement line. There is a probability around the 0.45 to have exactly 3 messages in the output signal. The second y axis refers to the number of unique output signals encountered and their length. For example, in Figure 4.4b we will have 1 unique output signal of length 2, 3 signals of length 3, 4 of length 4 and 2 of length 5. In total we saw 10 different unique signals on 100 different runs.

Those plots do not give a complete perspective of all the possible outputs that can be generated but only the one encountered during the runs. In fact, during the 100 runs, we encountered only the output signals listed in Table 4.3.

Signal	Frequency
$a1a2a1w1$	28
$a2a1w1$	23
$a2w2$	26
$a1a2w2$	23

(a) Node 4 output signals encountered

Signal	Frequency
$a1a2a3w3$	15
$a1a3w3$	16
$a2a1a2w2$	19
$a1a2w2$	28
$a1w1$	2
$a2a1a3w3$	6
$a2a1a2a3w3$	8
$a3a1a2a3w3$	3
$a2a1w1$	2
$a3a1a3w3$	1

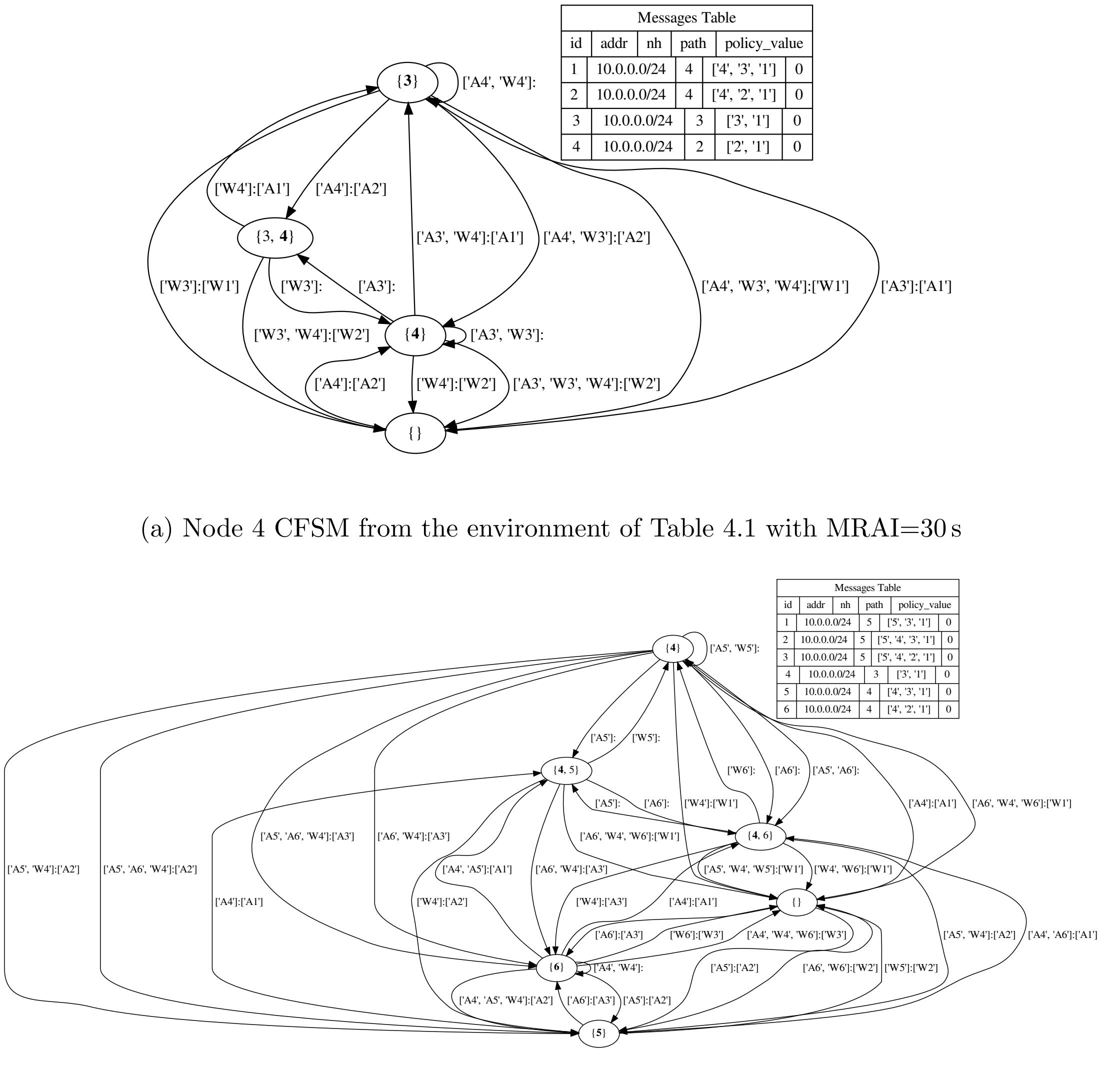
(b) Node 5 output signals encountered

Table 4.3: Node 4 and 5 different output signals encountered during the 100 runs, using the environment in Table 4.1, MRAI is ininfluent during the simulations, the signal used “AW”

4.2.1 MRAI and BGP FSM

How would MRAI affect the study of the signals produced by Figure 4.2? The answer is that the number of states will be the same but the number of possible transitions will explode because there will be a lot more possible input signals that will be compressed and evaluated by the nodes. Multiple transitions between the same nodes can exists because multiple input messages could be compressed in the same set by MRAI. For example the sequence "A" could produce the same effects of the sequence "AWA" once its compressed.

We can see the effects of MRAI on the CFSMs in Figure 4.5.



Messages Table

id	addr	nh	path	policy_value
1	10.0.0.0/24	4	['4', '3', '1']	0
2	10.0.0.0/24	4	['4', '2', '1']	0
3	10.0.0.0/24	3	['3', '1']	0
4	10.0.0.0/24	2	['2', '1']	0

(a) Node 4 CFSM from the environment of Table 4.1 with MRAI=30 s

Messages Table

id	addr	nh	path	policy_value
1	10.0.0.0/24	5	['5', '3', '1']	0
2	10.0.0.0/24	5	['5', '4', '3', '1']	0
3	10.0.0.0/24	5	['5', '4', '2', '1']	0
4	10.0.0.0/24	3	['3', '1']	0
5	10.0.0.0/24	4	['4', '3', '1']	0
6	10.0.0.0/24	4	['4', '2', '1']	0

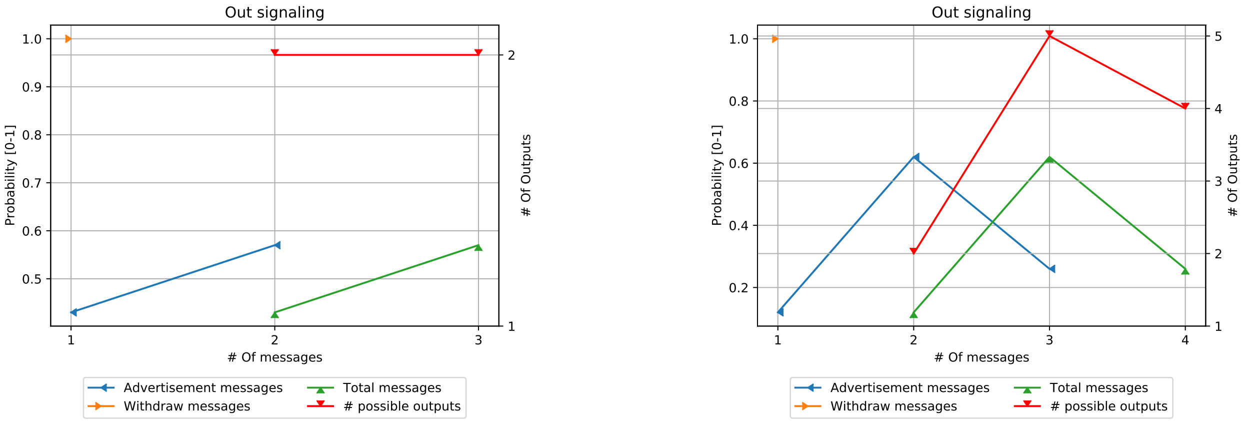
(b) Node 5 CFSM from the environment of Table 4.1 with MRAI=30 s

Figure 4.5: CFSM of nodes 4 and 5 of the graph in Figure 4.2 with an input signal of "AW" and with MRAI=30 s, in total has been executed 100 runs

The results in Figure 4.5 have been generated with the same environment of the one in Figure 4.3 but with an MRAI value equal to 30 s in each link, as defined by the RFC 4271 [1]. Figure 4.3a and Figure 4.5a permits us to compare the two CFSMs of node 4, is possible to notice a big difference in terms of edges between one figure and the other, the first one has 8 transitions, the second one 15. For the node 5, we pass from 16 transitions in Figure 4.3b to 36 thanks to MRAI in Figure 4.5b.

But the positive effects of MRAI can be found in the output signals, showed in Figure 4.6.

Comparing Figures 4.4b and 4.6b is possible to notice that there is a different distribution in the output signals. The x axis never reaches the value of 5, this means that the output signals of the node 5 never used more than 4 messages. While, without MRAI there were 2 unique output states with 5



(a) Node 4 output signals study with MRAI=30 s

(b) Node 5 output signals study with MRAI=30 s

Figure 4.6: Output signal study of nodes 4 and 5 of the graph Figure 4.2 with an input signal of "AW" at node 1 with MRAI=30 s for every link

messages, with a total frequency of 11. And we can also notice that the majority of the signals this time have a length of 3 messages, instead of the previous 4. This is a hint that MRAI can have positive effects on the number of output messages produced by single nodes, having, however, more possible transitions to consider.

This makes inferring the input signal even more difficult because we have a lower number of output state with a broader number of possible transitions.

4.3 BGP FSM explosion

We know that MRAI is not an easy parameter, the incorrect setting of it can lead to an explosion of messages and an exponential convergence time. This problem has been studied by Fabrikant et al. [2] and the origin of it has been attributed to the *path exploration* issue. This is a well-known problem in the BGP community and it is experienced by a node when it enters in a transitory phase where it accepts and publishes not optimal paths towards the destination before reaching a stable state. *Path exploration* can lead to an enormous amount of messages even with a small set of nodes [23].

As we saw in Section 4.2.1, MRAI can influence the CFSMs of the nodes and their output signals, which impact could it have if it is not set correctly?

I have then created an environment that resembles the study conducted in [2] using a topology like the one described in Section 3.1.2 with 3 rings and I compared different MRAI settings. The environment properties are presented in Table 4.4.

Property	Value
Seeds	$[1, 30]$
Signaling	"A", "AW", "AWA", "AWAW"
Withdraws delay	Uniform distribution between 5 s and 10 s, Uniform distribution between 10 s and 15 s
Announcement delay	Uniform distribution between 5 s and 10 s, Uniform distribution between 10 s and 15 s
Link delay	Uniform distribution between 0.5 s adn 3 s, uniform distribution between 2 s and 4 s

Table 4.4: Fabrikant experiments environment

In total, for each signalling experiment this environment produces 240 runs. I have then introduced 4 different MRAI strategies for each different signal. The different MRAI strategies are the following one:

- ***Fixed* 30 s**: MRAI is fixed for each link to 30 s;
- ***No MRAI***: MRAI is fixed for each link to 0.0 s;
- ***Ascendant***: MRAI will be doubled at each leach ($1 - 2 - 4 - 8 - ...$);
- ***Descendent***: Reverse of the ascendant case, MRAI will be divided by two at each leach.

Another important factor to consider during those experiments is the IW capability of BGP. This parameter will influence the number of messages that will be transmitted.

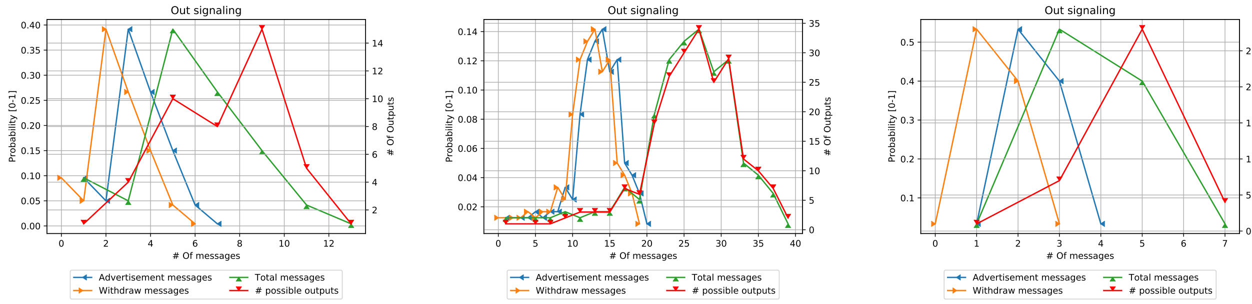
The results of all those different experiments, in terms of CFSM states and transitions are exposed in Table 4.5

Signaling	IW	No MRAI		Fixed 30s		Ascendent		Descendent	
		$\|S\|$	$\|T\|$	$\|S\|$	$\|T\|$	$\|S\|$	$\|T\|$	$\|S\|$	$\|T\|$
"A"	Yes	12	19	15	26	7	12	16	24
	No	30	100	30	125	9	21	30	132
"AW"	Yes	52	181	37	103	24	71	40	80
	No	51	221	57	263	22	90	58	274
"AWA"	Yes	51	170	25	50	33	148	50	137
	No	69	364	37	180	30	203	66	419
"AWAW"	Yes	77	461	38	132	54	300	53	148
	No	78	500	62	429	48	350	66	441

Table 4.5: Fabrikant CFSMs results, $|S|$ is the dimension of the states set $|T|$ is the dimension of the transitions set, The worst results for each category are colored in gray, the topology contains 3 rings, as Figure 2.4b, the environment is described in in Table 4.4

As is possible to see from the grey squares in Table 4.5 the more complex CFSMs are the ones without MRAI and with a descendent MRAI timing. In fact, the descendent strategy is the example described in [2] that provoke the extremely high number of transitions triggering the *Path Exploration* problem. Is also noticeable that the IW has a huge effect on both the number of states and the number of transitions. This because there are less possible combinations of input signals for the nodes. The opposite case in respect of the *Descendent* strategy obtains great results, even better than the actual standard of 30 s for each link. This performance improvement is caused by the fact that each leach will wait enough time to have more information from its predecessor in order to have more information to make the best decision.

The *Path Exploration* problem is also noticeable evaluating the output signals of the last node of the chain. Results about the output signal of the node 8 (the last node of the gadget) are presented in Figure 4.7.



(a) Node 8 output signals study with ***Fixed* 30 s** strategy

(b) Node 9 output signals study with ***Descendent*** strategy

(c) Node 9 output signals study with ***Ascendant*** strategy

Figure 4.7: Output signal study of nodes 8 of the graph Figure 2.4b with an input signal of "AWA" from node d with the ***Fixed* 30 s**, ***Descendent*** and ***Ascendant*** strategies, without the help of the IW

For simplicity I decided to study only the output signals caused by the sequence "AWA" and without considering the *No MRAI* case, results presented in Figure 4.7. The first signal study, Figure 4.7a, is the one that represents the actual standard value of the protocol [1]. We can notice in that particular output study that the maximum detected length of a signal is 13 and it's the last likely output, while the most probable output length is 5. While, we can infer the *Path Exploration* problem by the spike of unique output signals with a length of 9, this represent that the node experienced multiple changes in its decisions. The worst-case scenario is the one represented by Fig. 4.7b where the maximum length of the output signal reaches almost 40 messages, but the most probable output signal has a length between 20 and 30. This is the marker of a lot of decision changes in the best path for the destination. Opposite to that case, we found the *Ascendant* strategy in Figure 4.7c where the output signals never used more than 7 messages. In this last scenario we don't see a clear appearance of the *Path Exploration* problem, due to the fact that the node 8 is the last of the chain and the MRAI values permits to the other nodes to shares almost their best possible path.

In conclusion of this chapter, we can say without doubts that MRAI influences the performances, confirming what has been presented in [2], that an incorrect setting of it can lead to an explosion in terms of messages. And by consequence, to an explosion on the number of states and transitions. Making difficult infer the initial sequence knowing only the output signal. It is also noticeable that a different set of MRAI values can also lead to a better scenario than the standard one. An alternative to the standard MRAI has been already presented in [18] with the use of centrality metrics. Is possible to study, other than specific nodes in the network also the performances of all the sets of BGP speaker, in order to study the impact of MRAI in general and what can affect it.

5 BGP MRAI dependency

MRAI is one of the parameters that mostly has caused divergences in the scientific community. And, after the introduction in the protocol since version 4 [1], is one of the parameter more studied for it multiple effects, the correct setting can improve the protocol performances, while, an incorrect one can generate exponential convergence behaviours even in small network [2, 7].

The protocol strictly depends on this parameter, because as we saw in Chapter 4, the incorrect use of it can lead to tremendous consequences, even worst of not having it at all. In other cases, with a particular setting of it is possible to improve the network performances. Recent studies about centrality metrics on routing protocols introduce, through the distributed computation of the metric, a timer trade-off improvement [24, 25]. This kind of approach has been also applied on BGP with positive results on network failures [10, 18].

All those study points out how we can set MRAI to improve network performances, but what about how MRAI reacts to different problems? Is it possible that MRAI reacts differently based on where the signal occurs? In fact, our hypothesis is that is not enough just look to the MRAI setting because other factors can be relevant too. For example, a change near the central clique of T nodes could provoke a large storm of messages because MRAI doesn't affect in time the spreading of information. While, a change in the periphery could be cushioned without it reaching the center of the network multiple times.

5.1 Different MRAI strategies

An MRAI strategy define a particular way to set all the timers accordingly to some rules or properties. For example, the default strategy of MRAI, defined in [1], is to set all the timers to 30 s. In Section 4.2 I used 4 different MRAI strategies to show that an incorrect setting of it can lead to dangerous consequences. Another strategy that can be used during some experiments is the complete avoidance of MRAI, this is the *NoRFD* technique.

The hypothesis behind all the different strategies is that is possible to adjust the trade-off between the network performances adapting this timer. Recent studies try to use the centrality to adapt MRAI using the position of the node as property [18].

5.1.1 Centrality based strategy

The centrality based strategy is inspired from the *PopRouting* idea that has been successfully implemented in other protocols [24, 25]. The idea behind those studies is to exploit the fact that the more central nodes in the network should be more reactive to changes while the nodes that resides in the border of the network don't have a major influence. Load centrality and betweenness centrality are examples of centrality metrics that can be calculated on a graph [26], the load centrality is defined as follow.

Definition 5.1.1 (Load Centrality) *Consider a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ and an algorithm to identify the (potentially multiple) minimum weight path(s) between any pair of vertices s, d. Let $\theta_{s,d}$ be a quantity of a generic commodity that is sent from vertex s to vertex d. We assume the commodity is always passed to the next hop following the minimum weight paths. In case of multiple next hops, the commodity is divided equally among them. We call $\theta_{s,d}(v)$ the amount of commodity forwarded by vertex v. The* load centrality *of v is then given by:*

$$LC(v) = \sum_{s,d \in \mathcal{V}} \theta_{s,d}(v) \tag{5.1}$$

This mechanism is compatible with network protocols where there is an exchange of messages where is possible to introduce the commodity. In BGP is possible to include optional attributes inside the ADV messages. For this reason I have created the Destination Partial Centrality (DPC) and demonstrated that is possible to calculate it in a distributed way [10] with small changes to the protocol.

The DPC better adapt to BGP because it doesn't relay on the assumption that every node must distribute a load. The number of network prefixes distributed represent the load introduced on a network by a node. Not all BGP speakers distribute an IP prefix, but every node that forward traffic have a non-zero DPC centrality value.

We call $\mathcal{C} \subseteq \mathcal{V}$ the set of nodes that can be source and/or destination of traffic (they export at least one network) and N_s, N_d the number of networks that are exported by node s and d, respectively, then $\theta_{s,d} = \frac{N_s+N_d}{2}$. DPC $\Delta(v)$ of any vertex $v \in \mathcal{V}$ is defined as

$$\Delta(v) = \sum_{s,d \in \mathcal{C}} \theta_{s,d}(v) \tag{5.2}$$

Is then possible to configure MRAI accordingly with the centrality metric described, as has been done in [18]. I assume that the information in the ADV messages propagates in the network in three phases, identifying three propagation graphs:

- ***Ascending phase graph*** $\mathcal{G}_{\mathcal{A}}(\mathcal{V}^{\mathcal{G}_{\mathcal{A}}}, \mathcal{E}^{\mathcal{G}_{\mathcal{A}}})$*:* made by the nodes updated without reaching tier one nodes;

- ***Tier one graph*** $\mathcal{G}_{\mathcal{T}}(\mathcal{V}^{\mathcal{G}_{\mathcal{T}}}, \mathcal{E}^{\mathcal{G}_{\mathcal{T}}})$*:* made by tier-1 nodes;

- ***Descending graph phase*** $\mathcal{G}_{\mathcal{D}}(\mathcal{V}^{\mathcal{G}_{\mathcal{D}}}, \mathcal{E}^{\mathcal{G}_{\mathcal{D}}}) = \mathcal{G}(\mathcal{V}, \mathcal{E}) - \mathcal{G}_{\mathcal{A}}(\mathcal{V}^{\mathcal{G}_{\mathcal{A}}}, \mathcal{E}^{\mathcal{G}_{\mathcal{A}}}) - \mathcal{G}_{\mathcal{T}}(\mathcal{V}^{\mathcal{G}_{\mathcal{T}}}, \mathcal{E}^{\mathcal{G}_{\mathcal{T}}})$*:* the rest of the graph.

Considering a graph-wide maximum timer $T = 30\,\mathrm{s}$ and DPC $\Delta(i) \in [0, 1]$ for node i, DPC-based MRAI T_{ij} used by node i with neighbor j is set as follows:

$$T_{ij} = \begin{cases} \frac{T}{2}\Delta(i) & \forall i \in \mathcal{V}^{\mathcal{G}_{\mathcal{A}}} \\ \frac{T}{2} & \forall i \in \mathcal{V}^{\mathcal{G}_{\mathcal{T}}} \\ \frac{T(1-\Delta(i))}{2} + \frac{T}{2} & \forall i \in \mathcal{V}^{\mathcal{G}_{\mathcal{D}}} \end{cases} \tag{5.3}$$

5.1.2 Strategy comparison

In order to make every strategy comparable is possible to adapt the timers to respect a given $MRAI_{mean}$ value. Given M as the set of all the timers used in the network defined as follow:

$$M = \{T_{ij} | \forall (i, j) \in \mathcal{E}\}$$

The average value of MRAI is represented by $M_{avg} = EM$. Given M_{avg} and the required $MRAI_{mean}$ is possible to define $k = \frac{MRAI_{mean}}{M_{avg}}$. Giving us the possibility to adapt M in order to respect $MRAI_{mean}$ multiplying each element of M by k.

$$M_k = \{T_{ij}k | \forall (i, j) \in \mathcal{E}\}$$

5.2 Clique Experiments

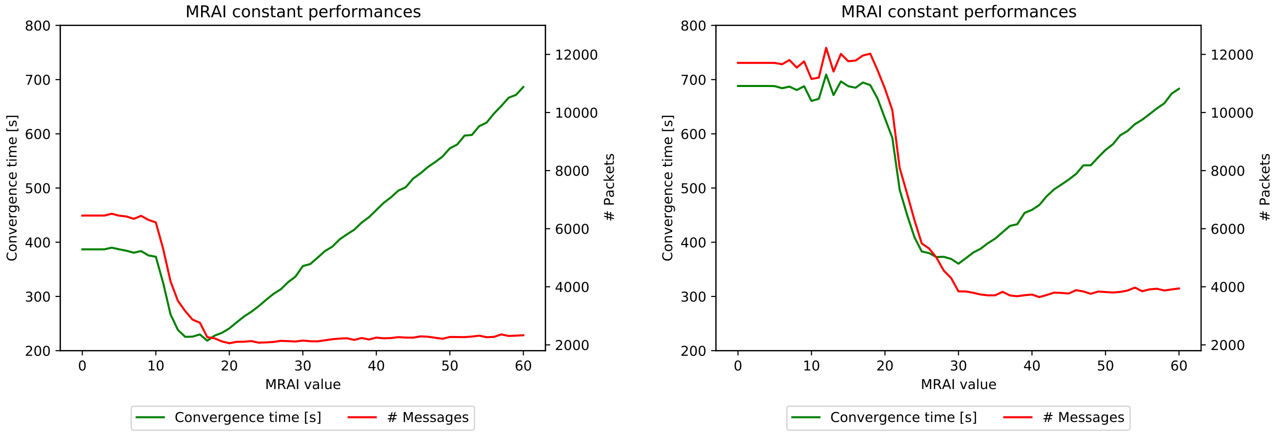
The clique topology is one of the worst-case scenarios as specified in Labovitz et al. [16]. For this reason I decided to start my study from it. I used two approaches in this Environment, the first one keeps the IW active the second one doesn't use this property. To emphasize the effects of this parameter with the effects also of different MRAI configurations.

The Environment properties are listed in Table 5.1

As described in Table 5.1, for each MRAI value has been executed 10 different runs of the environment. The clique graph used in these experiments is composed by 15 nodes. The MRAI strategy used is the *fixed* one, so every link will have the same MRAI value. The results are presented in Figure 5.1

Property	Value
Seeds	$[1, 10]$
Signaling	“AW”
Withdraws delay	Uniform distribution between 1 s and 5 s
Announcement delay	constant distribution of 5 s
MRAI	$[0, 60]$
Link delay	Uniform distribution between 0.0001 s and 0.5 s

Table 5.1: Clique environment properties, 10 possible different runs



(a) Network performances **with IW**

(b) Network performances **without IW**

Figure 5.1: Evolution of the network performances on the clique graph of 15 nodes using a fixed MRAI from 0 to 60 seconds. Each point is the average of the 10 different runs provided by the environment in Table 5.1

Is possible to notice in Figure 5.1 both the effect of MRAI and IW. Those plots represent the network performances in terms of convergence time and number of messages transmitted to reach the convergence after the transmission of the signal “AW”. The convergence time is described by the average value from all the nodes in the network. Each point in the plots is the average of the 10 runs executed with the *Fixed* MRAI value on the x axis. The left y axis should be used with the convergence time, the green line, while the second y axis represents the number of messages transmitted, the red line.

The effects of MRAI are present in both the plots but in two different moments. In Figure 5.1a MRAI affects both the convergence time and the number of messages around 10 s up to 20 s. After the threshold of 20 s, the effects of MRAI are counterproductive, the convergence time is negatively affected because the nodes start to wait more time without obtaining more useful information. This can be seen also in the number of messages that reaches a constant value. Other messages are not necessary to converge and MRAI doesn’t compress the input anymore, so the nodes are not able to aggregate more information.

In Figure 5.1b we can see the same effects but with a higher MRAI value. The number of transmitted messages reaches the constant value with an MRAI value around 30 s. The effects of IW can be saw also in the number of messages and the convergence time with a low MRAI. In Figure 5.1b is possible to reach even 12 000 messages with an MRAI equal to 0 s, while, with IW active, there are no more than 6500 messages.

5.3 Internet like experiments

The internet like environment is more complex than the clique one, but it permits to have a more close vision of what can really happen on the Internet. During my studies, I used different topologies with 1000 nodes resembling the Elmokashfi properties [17] already described in Section 3.1.3.

Using this graphs I will look for a possible interaction between MRAI and other factors that can influence the network. First of all, MRAI has a dependence on how it is set, I'm going to compare different MRAI strategies that can be used on an Internet-like graph. Another influencing factor could be the signal used as an input or even the position of the node that provoke the change.

5.4 Strategy dependence

Like showed in Section 4.3, the network performances depend on the MRAI strategy chose. For this reason, the first goal of my study is to point out this differences. In order to do that, the first investigation that I would like to present is the one that takes in consideration how the standard protocol evolves on an Internet environment with different possible MRAI strategy.

The property of the environment chosen are described in Table 5.2

Property	Value
Seeds	$[1, 10]$
Signaling	"AW"
Withdraws delay	Uniform distribution between 1 s and 60 s
Implicit withdraw	Active
MRAI	$[0, 60]$
Link delay	Uniform distribution between 0.012 s and 3 s

Table 5.2: Internet like environment properties, 10 different possible runs for each experiment, 61 experiments in total, one for each MRAI value

The graph is an *Internet-like* graph with 1000 nodes. The node that will execute the signal has been chosen randomly between all the nodes of type "C". This graph will be the same for all the experiments in this section.

For each MRAI strategy, that I'm going to present, has been executed 61 experiments, one for each possible value of MRAI. For each experiment, thanks to the environment variables, has been executed 10 runs. In total for each MRAI strategy has been executed 610 different runs.

As MRAI strategies I decided to use the following two:

- ***Fixed*:** Every link will have the same MRAI value;
- ***DPC*:** This strategy assign a different MRAI value to each link depending on the centrality of the node [18] like described in Section 5.1.1.

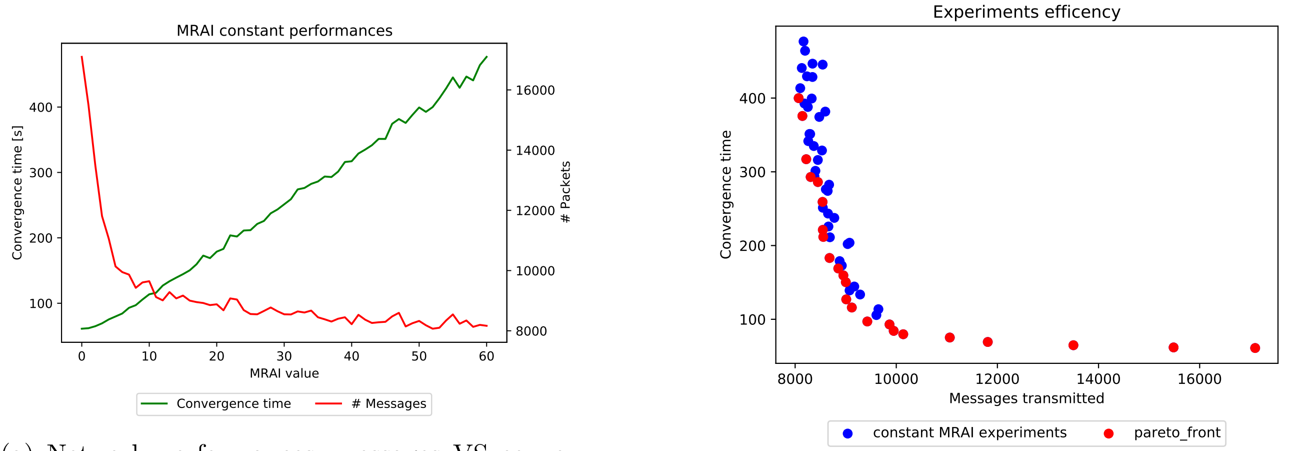
Thanks to the fact that has been already demonstrated that is possible to calculate the DPC strategy in a distributed way [10] I assume that is calculated in advance and that every node knows it's own centrality to set the timers.

To permit a comparison between those two different strategies I used the constraint introduced in Section 5.1.2. The MRAI value required by the experiment is the mean that must be respected by the strategies. For the fixed strategy an MRAI value equal to 10 s implies that every link would have 10 s as timer. The DPC strategy, on the other hand, has to adapt its values to respect the MRAI required, like has been showed in Section 5.1.2.

The results of the *Fixed* strategy are showed in Figure 5.2.

As is possible to see in Figure 5.2a without MRAI we would have a low convergence time, dictated mostly by network delays and processing time. With, on the other hand, an enormous amount of messages. Slightly increasing the MRAI value, the number of messages fell down reaching a constant value around 8000, while the convergence time continuously grows linearly, as it happened for the clique graph in Figure 5.1. This continuous linear grow is dictated by the fact that nodes keep meaningful information for more time before sharing them with their neighbourhood.

Figure 5.2b represent the Pareto front of those experiments. The Pareto frontier is the set of values that are Pareto efficient, this concept has been already used in engineering to define the set of best outcomes from the trade-off of two different parameters [27]. We can clearly see that the majority of the points is concentrated on the left of the chart, this means that few MRAI values would give as a



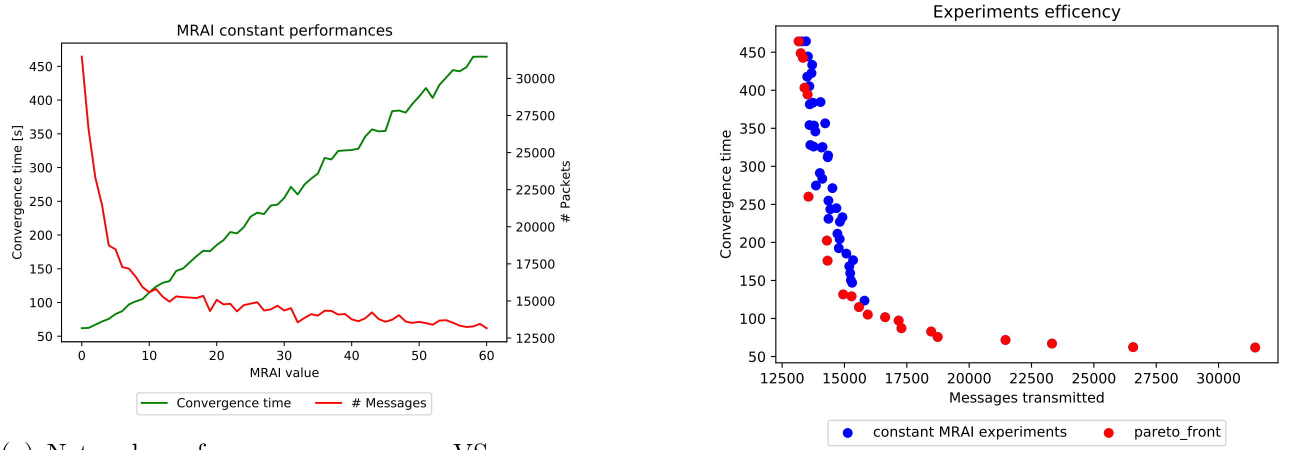
(a) Network performances, messages VS convergence time with different MRAI values

(b) Pareto front of Messages VS Convergence time

Figure 5.2: Evolution of the network performances on the **Internet Like** graph of 1000 nodes using a fixed MRAI from 0 to 60 seconds. Each point is the average of 10 runs, in total has been executed 610 runs. FiXme: Adjust figure dimensions

result a high number of messages and a small convergence time. While, multiple MRAI values permits to have results concentrate around the same amount of messages transmitted. This can confirm the fact that MRAI would not influence messages after a certain threshold but only the convergence time.

The results of the same environment without IW are showed in Figure 5.3.



(a) Network performances, messages VS convergence time with different MRAI values

(b) Pareto front of Messages VS Convergence time

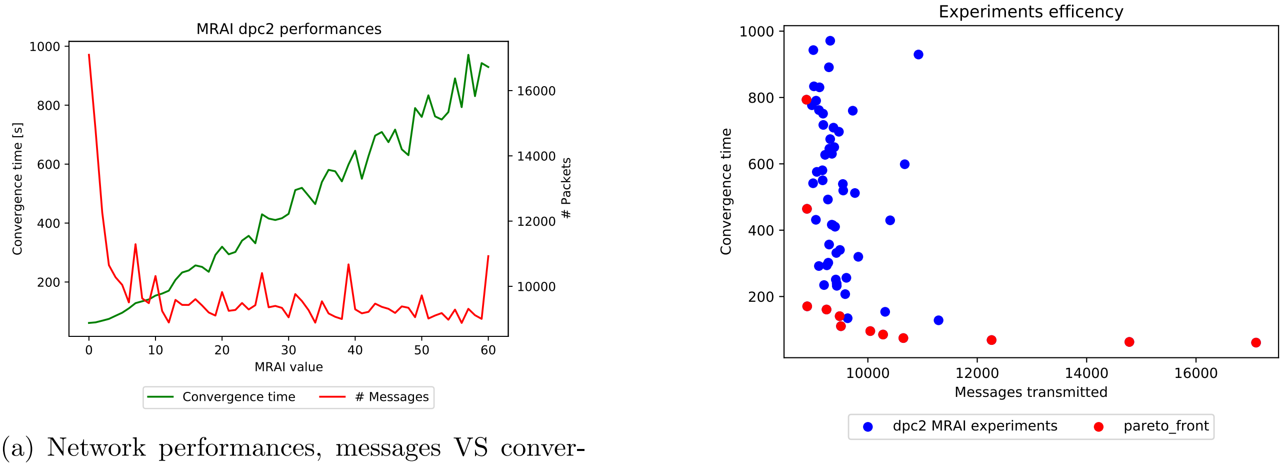
Figure 5.3: Evolution of the network performances on the **Internet Like** graph of 1000 nodes **without IW**, MRAI strategy *Fixed*from 0 to 60 seconds. Total number of runs 610, 10 for each MRAI value used. FiXme: Adjust figure dimensions

Also in this case, comparing Figures 5.2 and 5.3, is possible to notice that IW helps to reduce the number of messages and the convergence time without impacting the network performances trend.

The second strategy, the one dependant on the DPC, produce the results in Figure 5.4 As mentioned before, all the timers are adjusted to respect the same mean as in the *fixed* MRAI experiments. For this reason points with the same MRAI value are comparable one another.

This second strategy leads to the performances showed in Figure 5.4, where is possible to notice that the number of messages transmitted fell down very quickly and it reaches the convergence value with an MRAI value of 10. But, it is also noticeable that there are a lot more spikes in this trend, that deviate more from the constant value around 9000 messages. And by consequence, the convergence time is affected by this behaviour.

Comparing Figures 5.2 and 5.4 is possible to notice that the two strategies lead to a different trend. Both are equal at the beginning with MRAI equal 0, but, after a while, both the number of transmitted



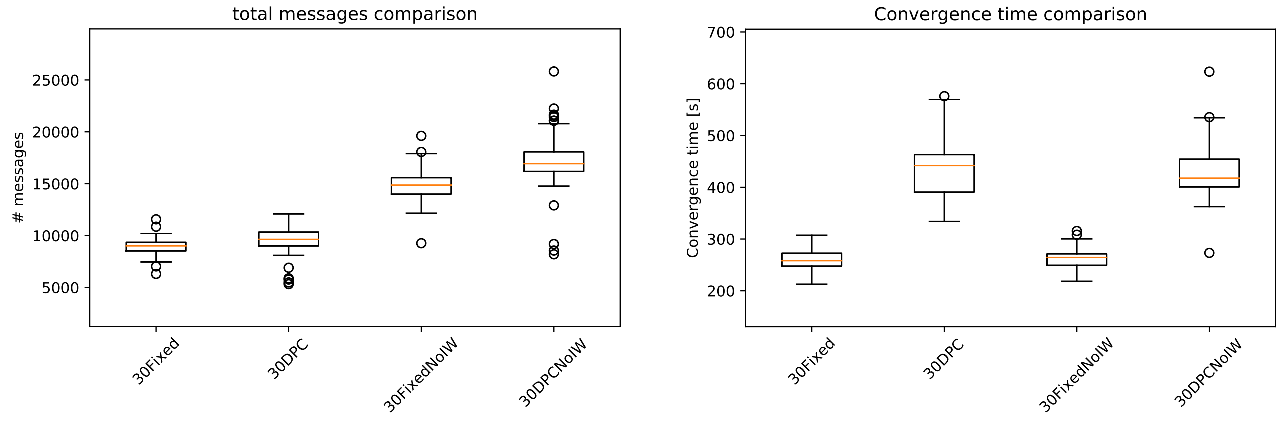
(a) Network performances, messages VS convergence time with different MRAI values

(b) Pareto front of Messages VS Convergence time

Figure 5.4: Evolution of the network performances on the **Internet Like** graph of 1000 nodes using a *DPC* MRAI strategy with an $MRAI_{mean}$ from 0 to 60 seconds.

messages and the convergence time diverge. The number of messages with the DPC strategy variate more and it converges around 9000 messages, while the *fixed* strategy reaches 8000 messages as stable point. And the convergence time with the second strategy grows more quickly. This is caused by the central clique of tier-one nodes that have a high MRAI value. The high MRAI value is caused by the fact that all the leaves have 0.0 as centrality producing an MRAI value of 0 s and to respect the $MRAI_{mean}$ value the central nodes need a massive value. For example, with an $MRAI_{mean}$ of 30 s the node 1 (that is one of the central clique nodes) has an MRAI value of 79.35 s for all its neighbours.

The standard value of MRAI is 30 s as described in [1] so I compared those strategies performances with that amount with a box-plot in Figure 5.5. I decided to ru100 different runs for each strategy with the $MRAI_{mean}$ fixed to 30 s.



(a) Network performances, messages necessary to reach convergence with different MRAI strategies

(b) Network performances, time required to reach convergence with different MRAI strategies

Figure 5.5: Network performances comparison with different MRAI strategies, Graph internet like with 1000 nodes, MRAI value 30 s, number of runs for each strategy 100

In Figure 5.5 we can compare those two strategies, the first figure, Figure 5.5a represent the number of messages transmitted by the 100 runs, we can see that the two strategies, with the use of IW, are really close one another. While, in the time required to convergence, Figure 5.5b, there are some huge difference between the two strategies. The fact that the DPC strategy requires almost the double of the *Fixed* one is not negligible.

In conclusion, I can confirm that the MRAI strategy is one of the factors that can influence the Network performances.

5.5 Pareto Efficiency Front

The strategies exposed in Section 5.4 are just few of the available alternatives. For this reason, I would like to explore the set of possibilities looking for MRAI configuration randomly generated.

I'm going to study the space of possibilities that are generated through the Pareto efficiency plot and compare the results with the Pareto efficiency graphs already presented in Section 5.4. To permit this comparison I would set MRAI randomly but, like for the DPC strategy, respecting the average required.

The environment used for those experiments is shown in Table 5.3

Property	Value
Seeds	$[1, 10]$
Signaling	"AW"
Withdraws delay	Uniform distribution between 1 s and 60 s
Implicit withdraw	Active
MRAI mean	$[0, 60]$
MRAI values	Uniform distribution between 1 s and 120 s
Experiments per MRAI mean	10
Link delay	Uniform distribution between 0.012 s and 3 s

Table 5.3: Random MRAI environment properties

Thanks to this environment I'm going to run in total more than 600 complete experiments. For each MRAI mean value, I will generate 10 different graphs with a random MRAI value assigned to each link. I will then execute 10 different runs for each random graph that will produce the average result of 1 experiment. A single point is defined by the average convergence time and number of messages of 10 runs.

In Figure 5.6 is possible to see all the 610 points generated.

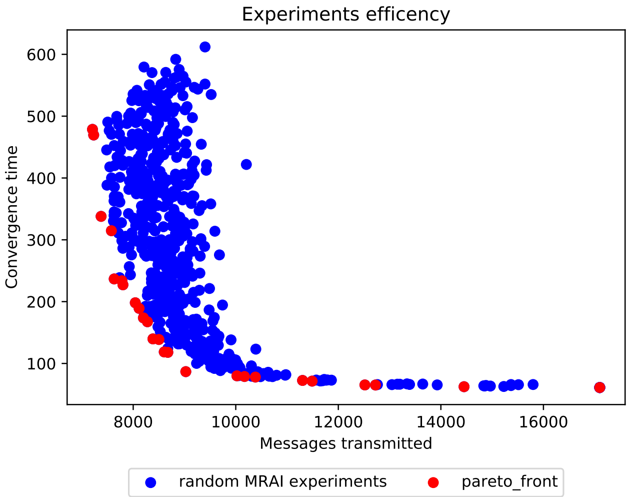


Figure 5.6: Pareto front generated by 610 experiments on an internet-like topology with 1000 nodes, MRAI generated randomly and adapted to the mean, 610 total experiments, each point is the average of 10 points. FiXme: add $[s]$ to the y axis

As we can see the trend in Figure 5.6 is similar to the one that we saw for the same signal in Section 5.4. For the majority of configurations, the number of messages transmitted is never over 10 000 but the time required to converge can grow over 600 s.

In Figure 5.7 is present a comparison between the random experiments, the fixed MRAI strategy and the DPC strategy from Figures 5.2b and 5.4b

As we can see in Figure 5.7 all the strategies have the same behaviour, but is also possible to see that the random strategy is the only one with experiments that produce less than 8000 messages. This

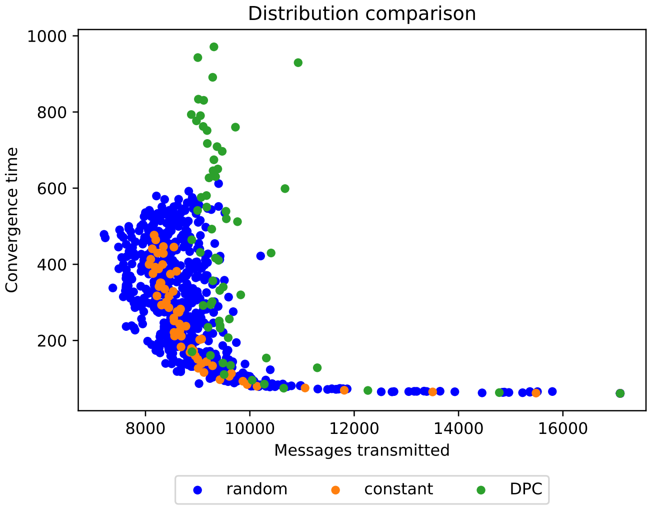


Figure 5.7: Pareto front generated by 601 experiments on an internet-like topology with 1000 nodes, MRAI generated randomly and adapted to the mean, vs fixed MRAI strategy and DPC MRAI strategy. FiXme: add [s] to the y axis

is important because it is the proof that better possibilities exists rather that the classical one.

For this reason, MRAI can be tuned to have a better trade-off between the number of messages transmitted and convergence time.

5.6 Signal dependence

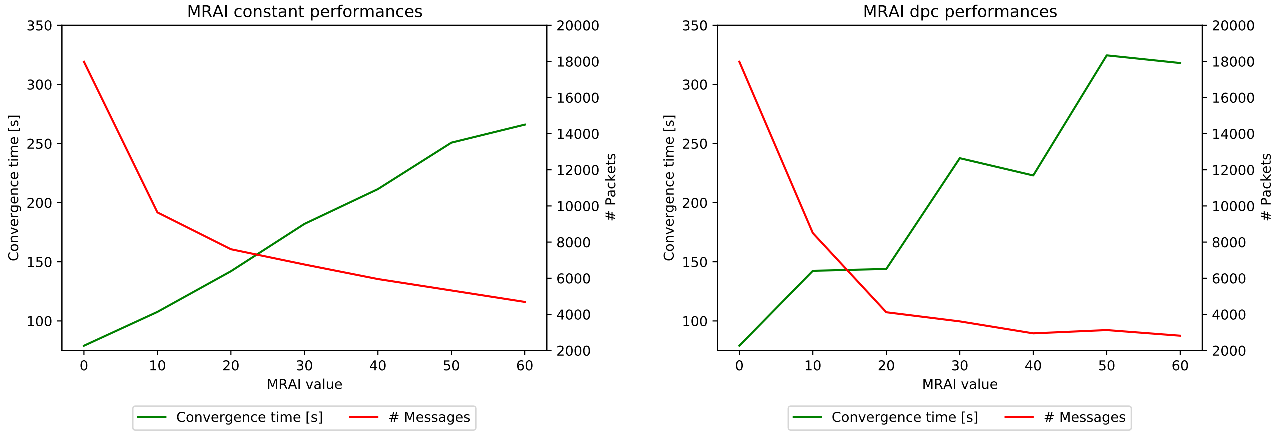
I would like to analyze how much the signal can impact the convergence performances with the two different strategies of Section 5.4.

For this reason, I use the same environment described before and execute the experiments with different input signals from the same node, "AWA", "AWAW" and "AWAWA".

In those experiments plays a role also the "*re-advertisement distribution*" for the second and third "A", it has been set to a uniform distribution between 1 s and 60 s, like the "*withdraw distribution*".

For those experiments, I didn't evaluate the case with IW deactivated. Because, like we saw in Figures 5.2 and 5.3 there is not a difference in terms of the trend. Is straightforward to imagine that the performance of the following experiments wouldn't have big differences in terms of tendency.

In Figure 5.8 is possible to see the evolution for the signal "AWA".



(a) Network performances, *fixed* MRAI strategy

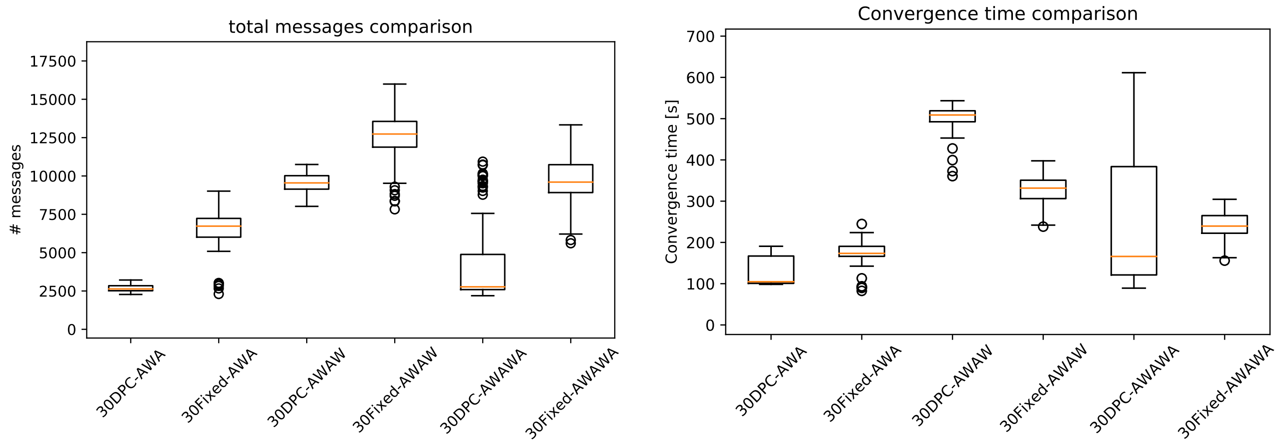
(b) Network performances, DPC MRAI strategy

Figure 5.8: Network performances comparison with different MRAI strategies, Graph internet like with 1000 nodes, signal "AWA". Each point represent the average between 10 runs.

Is possible to notice in Figure 5.8 a big difference compared to the plots in Figures 5.2 and 5.4. The DPC strategy was able to outcome the standard *fixed* strategy over multiple prospective. Analyzing Figure 5.8b is possible to see that the red curve, the one that refers to the number of messages transmitted, has a very fast fell. With an average MRAI timer *of* 30 s the number of messages is less than 1/4 in respect of an MRAI *mean* of 0 s.

The convergence time curve has a completely different trend in respect of the previous experiments. There are three steps in the trend, caused by the fact that now the timer is able to effectively compress the signal. MRAI doesn't affect the first message, in this case the first "A" , but it can affect the next two messages. In fact, some nodes are able to cache both the "WA" part of the signal and completely avoid sending anything at all, because they have already transmitted the first "A". The complete compression of the signal "AWA" is "A". The other evolution, for the "AWAW" and "AWAWA" signals, are showed in Figures A.1 and A.2 Those valley are provoked by the gain in performances that the compression give that overcomes the downside of having to make nodes wait longer.

Like before, comparing the standard 30 s fixed MRAI, I executed 100 different runs for each strategy and each different signal, the results are showed in Figure 5.9b.



(a) Network performances, messages necessary to reach convergence with different MRAI strategies

(b) Network performances, time required to reach convergence with different MRAI strategies

Figure 5.9: Network performances comparison with different MRAI strategies, Graph internet like with 1000 nodes, MRAI value 30 s, number of runs for each strategy 100, signal "allSignals"

Is possible to notice in Figure 5.9 that both the strategies have different performances in respect of the signal produced by the single node source. In particular, performances are better when the signal ends up with an "A". That's because, after the first "A", giving a MRAI timer long enough, a node is able to compress a sequence that ends with another "A" to the empty set and don't send anything more. While if the sequence ends up with a "W" it has to, at least, send another message to notify the withdraw.

Other than that is possible to notice that the DPC techniques have better results in terms of messages transmitted, while it could have a higher convergence time. This is caused, like before, by the high MRAI values used by the most central nodes.

In conclusion, there is an interaction between MRAI and the sequence of messages transmitted by the source node. In particular more the timer is able to compress the sequence more the performances can be improved. Is not possible to increase the MRAI timer indefinitely in order to catch everything, because of the side effect on the convergence time.

5.7 Position dependence

The last factor of influence for MRAI that I would like to study is how much the position of the signal source can influence the convergence. The main hypothesis is that a node closer to the central clique, that generates a signal would provoke a message storm bigger in respect of a node on the perimeter of the network. Because of the possibility from the nodes to catch the signal and compress it before reaching the central clique. A node near the central clique would provoke a case of *Path exploration* directly on the central clique. This hypothesis could be true only if MRAI is large enough to block the storm near the source of the change, compressing the signal and exporting only the correct information at the end of it.

5.7.1 Different signal sources

As first try, I have decided to analyse 10 different destinations chosen randomly on the same graph, this graph is an Internet like topology with 1000 nodes. I have then used the same environment with all the different destinations. I also configured different MRAI strategies, repeating the experiments for all of them. With this results is possible to analyze how different signal sources provoke different network performances and also study how different MRAI strategies adapt to different nodes that provoke messages storms.

The environment used by those experiments is the one described in Table 5.4.

Property	Value
Seeds	$[1, 10]$
Signaling	"AWAWA"
Withdraws delay	Uniform distribution between 0.1 s and 60 s
Announcement delay	Uniform distribution between 0.1 s and 60 s
Link delay	Uniform distribution between 0.0001 s and 0.5 s
MRAI	$[0, 60]$ with steps of 10
Random nodes	10
MRAI strategies	Fixed MRAI, DPC, reverse DPC

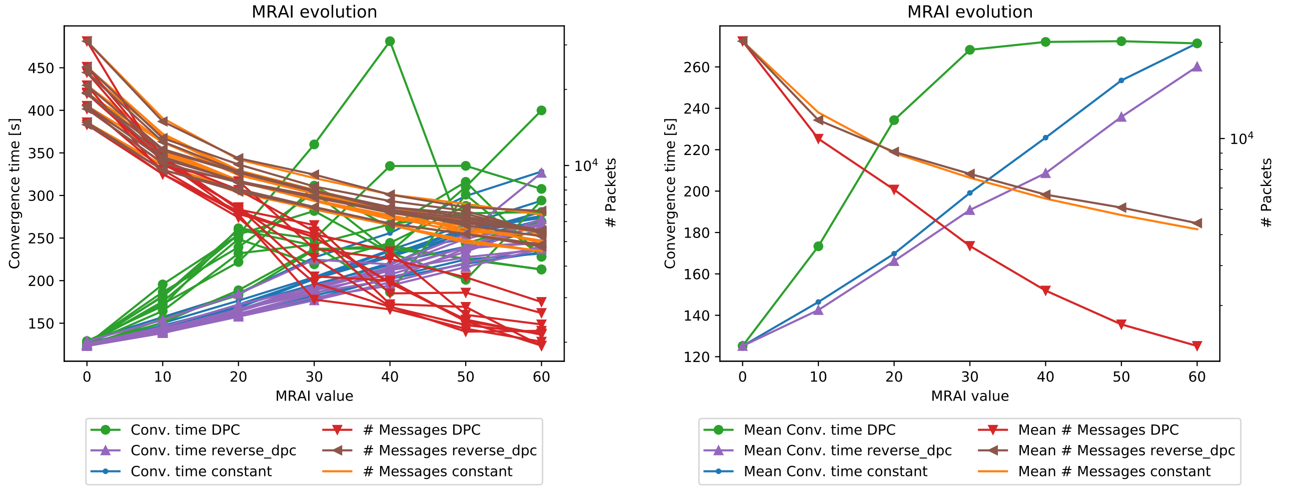
Table 5.4: Different signal sources environment properties

Like mentioned in Table 5.4 I decided to use another MRAI strategy and look forward to its performances. That strategy is the reverse of the one base on the DPC described in Section 5.1.1. I will use a higher MRAI for the first part of the graph and a smaller one in the descending phase.

In total, I have executed 700 runs for each MRAI strategy with the 10 different signal sources. The resulting network evolution with different MRAI values is shown in Figure 5.10. Each point of Figure 5.10a represents the average of the 10 runs executed for the specific destination with that MRAI configuration. Figure 5.10b shows the average result obtained from all the 10 different destinations for a specific MRAI strategy.

Figure 5.10 shows how, changing the source of the signal, also changes the network performances with multiple MRAI values. Notice that the second y-axis, the one that represents the number of packets transmitted, is in log scale. We can use the plot in Figure 5.10a to see the differences between one destination and the others, while Figure 5.10b expose the differences between the different strategies. We can see in Figure 5.10a that all the 10 destinations cause a similar behaviour with the fixed strategy and the reverse DPC, both in terms of messages transmitted and also convergence time. In both techniques, the difference between the 10 destinations in terms of convergence time is a few tens of seconds, with linear growth. Thanks to Figure 5.10b is possible to see that the number of messages reaches a convergence value between 5000 and 6000. The DPC strategy seems to bee more volatile with the growth of MRAI, both in terms of messages and also convergence time. In Figure 5.10a, from the multiple green spikes is possible to see that the position of the source is highly affective using this strategy. In terms of messages transmitted the DPC strategy always goes better than the other two strategies, respecting the trend in Section 5.6.

We can conclude that the position of the source can influence the behaviour of MRAI, it is more influent when the strategy used relies on topological information, like the DPC strategy. But, the chose of the strategy is more influent, the number of messages transmitted by the DPC strategy are always less than the one transmitted by the other two strategies, for every destination.



(a) Network performance evolution of all the 10 different signal sources with the different MRAI strategies

(b) Average of the network performances of 10 different signal sources are chosen with different MRAI strategies

Figure 5.10: Network performances given 10 different signal sources chosen randomly on an Internet like graph of 1000 nodes, with different MRAI strategies used, fixed, DPC, Reverse DPC, and different MRAI mean values. Each point represent the average of 10 different runs.

5.7.2 Hierarchical influence

What about the position in the hierarchy? Internet is very strong hierarchical graph, Figure 2.5b is an example with a small set of nodes but is possible to distinguish different levels in the graph. If we take the central clique as the root of the graph then all the nodes will be at a certain distance (in terms of hops) from it.

The question is: Nodes that are on the same hierarchical level reacts in the same way?

To analyze this possibility I decided to take randomly 3 nodes from each hierarchical level of an Internet-like topology of 1000 nodes, the number of levels in this graph is 4. The total number of destinations is 12 and for each one of them I executed an experiment with multiple MRAI strategies and different $MRAI_{mean}$ values.

The properties of this environment are summarized in Table 5.5.

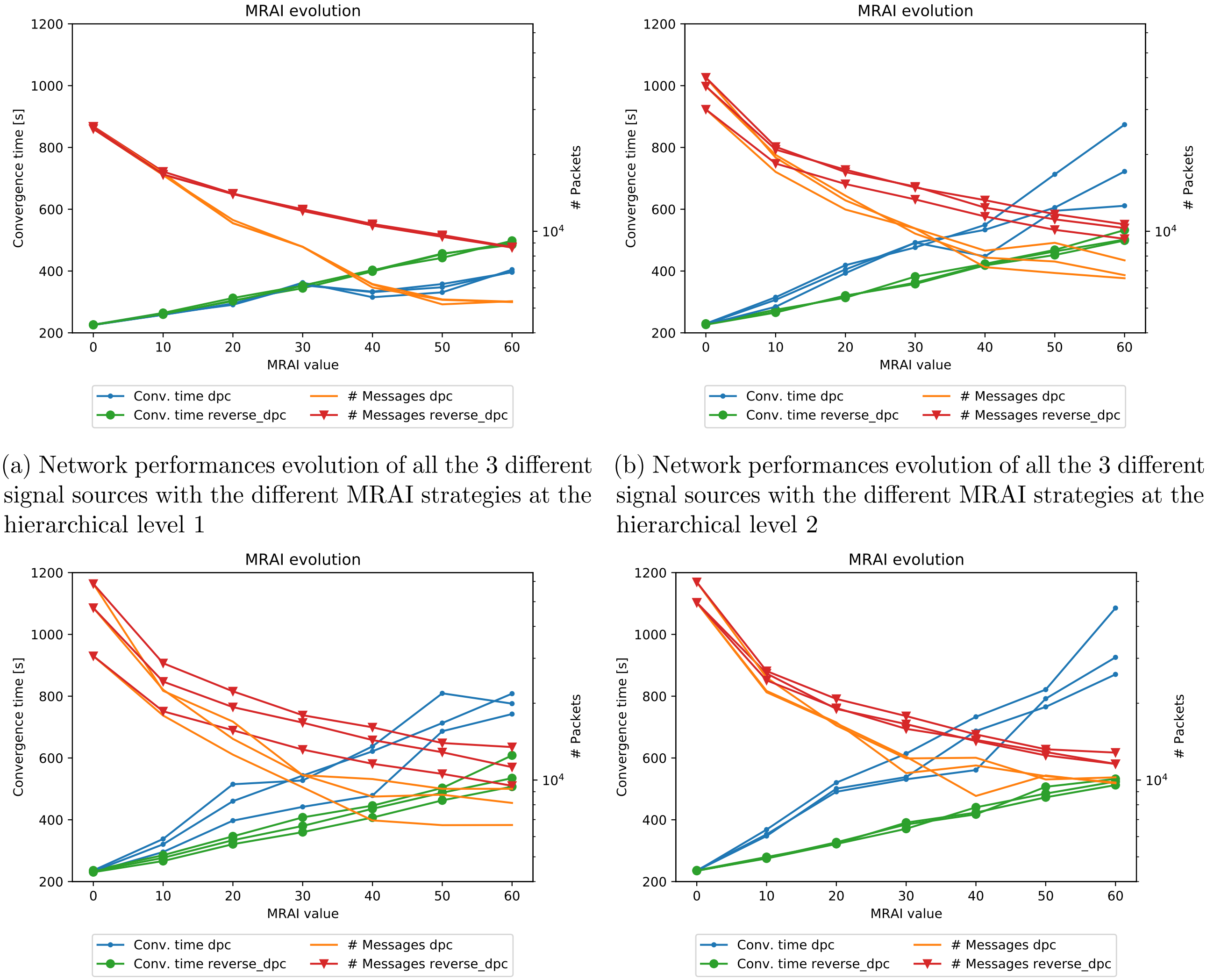
Property	Value
Seeds	$[1, 10]$
Signaling	“AWAWAWAW”
Withdraws delay	Uniform distribution between 0.1 s and 60 s
Announcement delay	Uniform distribution between 0.1 s and 60 s
Link delay	Uniform distribution between 0.0001 s and 0.5 s
MRAI	$[0, 60]$ with steps of 10
Number of levels	4
Random dst per level	3
MRAI strategies	DPC, reverse DPC

Table 5.5: Hierarchical experiments environment properties

Given that, I am evaluating the impact of nodes by their distance from the center of the network, it is a good way also to test strategies which goal is to enforce those points. For this reason, I chose those two strategies that relies on the centrality of the nodes.

The results in Figure 5.11 shows the different evolution of the random sources for each different level, from the first to the fourth.

In Figure 5.11 is possible to see the evolution of the network with different source nodes from different levels and how MRAI influence the network performances. Starting from the first level in



(a) Network performances evolution of all the 3 different signal sources with the different MRAI strategies at the hierarchical level 1

(b) Network performances evolution of all the 3 different signal sources with the different MRAI strategies at the hierarchical level 2

(c) Network performances evolution of all the 3 different signal sources with the different MRAI strategies at the hierarchical level 3

(d) Network performances evolution of all the 3 different signal sources with the different MRAI strategies at the hierarchical level 4

Figure 5.11: Network performances given 3 different signal sources chosen randomly for each level of the Internet like graph, number of nodes 1000, number of levels 4, different MRAI strategies DPC and reverse DPC FiXme: Adjust messages ticks

Figure 5.11a, where the source node is directly connected with a node of the central clique we can see different important things. First of all, the variance between the three different sources is very small, both in terms of messages transmitted and also convergence time. Both the techniques, DPC and the reverse of it, in terms of messages transmitted starts from a value around 25 000 with MRAI equal to 0 s and both reaches a value under 10 000 units. In terms of convergence time, the reverse strategy grows linearly as expected while the DPC technique is able to gain a strong reduction thanks to the efficient messages compression.

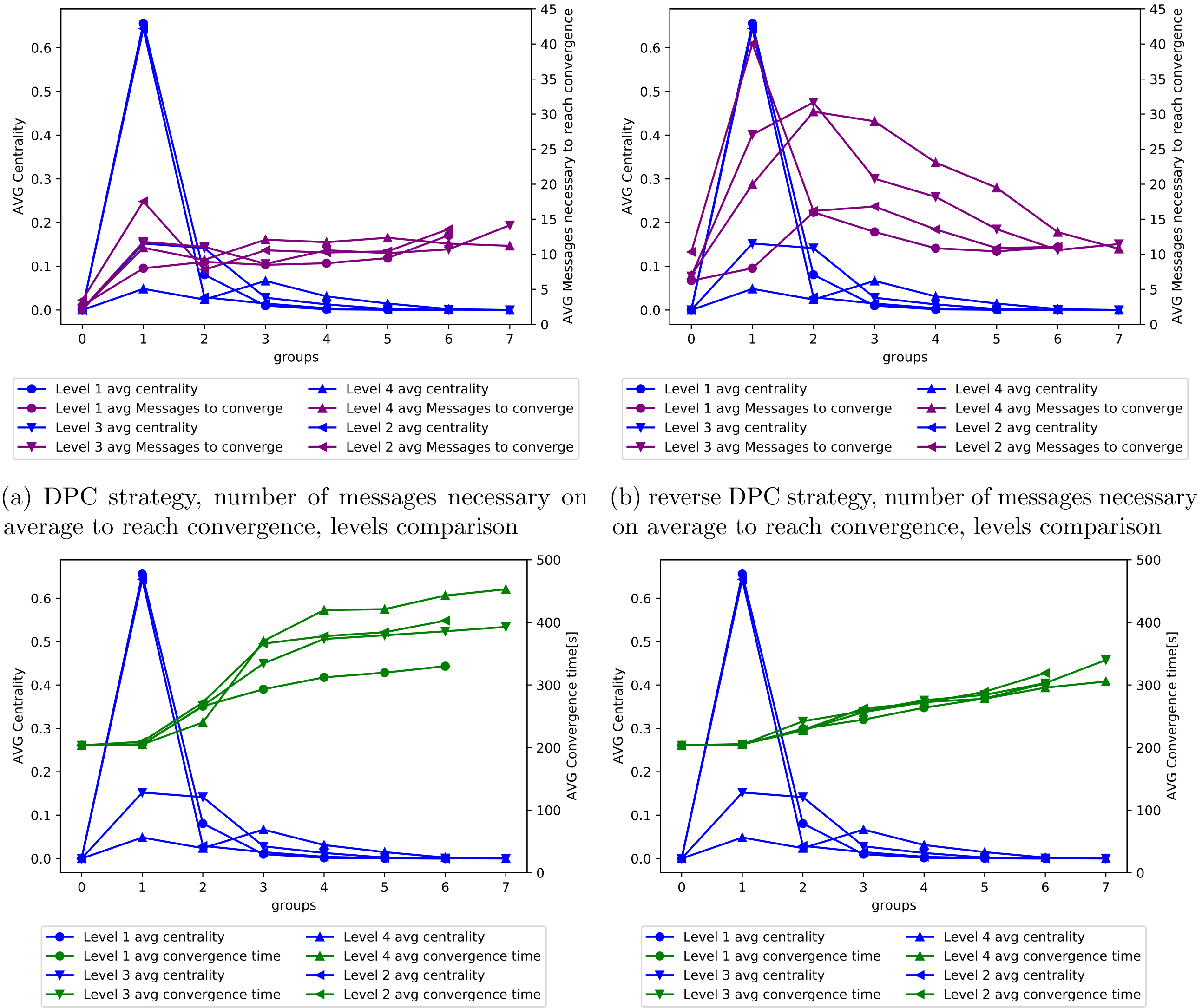
Going to the second and third level, respectively in Figures 5.11b and 5.11c, we can see that there is an increase of the variance between the single source trends, both in terms of messages and also convergence time. Also, the number of messages transmitted slowly increases, at the beginning the number of messages reaches 60 000 units converging around 10 000.

In Figure 5.11d we can see the worst-case scenario, not in terms of variance between the sources, but in this case, we have the worst performances. The convergence time grows even over 1000 s for the DPC strategy. While the number of messages transmitted with MRAI equal to 0 s touches 60 000 reaching a convergence value slightly over 10 000.

The influence that the position of the source nodes has to a hierarchical graph like the Internet is clear. The performances with MRAI at 0 s are only attributable to the strategic position of the source node. But, while MRAI grows, the position plays a central role on the performances. Nodes closer to the central clique have less space to provoke fluctuations in terms of possible paths, causing a less active effect of the *Path exploration* problem. Our initial hypothesis is proven to be wrong, nodes that are farthest are more incline to provoke a more intense path fluctuation and by consequence there are inferior performances.

Another hypothesis is that there is a difference in the single nodes based on the position in the topology. Nodes that are more central, if the signal comes from closer nodes could react with a more explosive *Path exploration* behaviour. If the signal comes from the periphery of the network, the other side of it could require more time to converge in respect of a signal that starts near the center.

To prove that hypothesis I decided to use the data from this set of experiments with MRAI value equal to 30 s. I calculated for each node the average performances, based on the level, convergence time, number of messages necessary to reach the convergence state and the DPC centrality value. I have then grouped nodes that are at the same distance from the source, to calculate the distance I used the distance in terms of hops in the *Best Path*. I then calculated the average performances of each group. The results are showed in Figure 5.12.



(a) DPC strategy, number of messages necessary on average to reach convergence, levels comparison

(b) reverse DPC strategy, number of messages necessary on average to reach convergence, levels comparison

(c) MRAI strategy DPC, convergence time levels comparison

(d) MRAI strategy reverse DPC, convergence time levels comparison

Figure 5.12: Internet like topology of 1000 nodes grouped by the distance from the signal source, different level comparison with MRAI strategy DPC and reverse DPC, MRAI value equal to 30 s FiXme: Maybe could be more interesting to see how DPC vs *Fixed* goes

In Figure 5.12 is possible to see an analysis of the average nodes performances grouped by the distance from the source of the signal. Those performances are taken from the case where $MRAI_{mean}$ is equal to 30 s. On the left side, in Figures 5.12a and 5.12c are showed the performances of the nodes that uses the DPC strategy. While, on the other side, Figures 5.12b and 5.12d, expose the performances in case of the reverse DPC strategy.

In all the plots in Figure 5.12 the blue line represents the average normalized centrality of the different nodes groups at different levels. Is possible to see that in the case that the nodes are in the first or the second level the nodes with the highest average centrality are the nearest ones. While if the node is far away from the central clique the nodes with the higher average centrality are in the second or third group. The values for those groups are low because of the high number of nodes in those groups, most of them having a small centrality value. The centrality refers to the first y-axis on the left.

The two plots in Figures 5.12a and 5.12b, respectively the DPC case and the reverse DPC, show the average number of messages required by each group of nodes to reach the convergence state. The main difference between them is in the number of messages experienced by the nodes near the source. In the DPC case, the value remains stable around 10 messages, while, with the reverse strategy, this number explodes up to 40 messages because of the *Path exploration* problem provoked by the ascending part of the graph.

Is possible to notice that some of the lines end up at the 6^{th} hop instead of the 7^{th}, that's because for the level 1 and 2 there are no best paths to the destination that goes over the 6 hops.

The convergence time is presented in Figures 5.12c and 5.12d. Is easily noticeable that with the DPC strategy after the more central groups there is a separation between the lines, but the trends are the same. The convergence time slowly increases in farthest nodes, thanks to the particular MRAI strategy that permits to wait for enough time to receive all the information necessary. This is also the reason for the small variance in terms of messages transmitted. With the reverse DPC strategy, on the other hand, is possible to have a more stable trend in the convergence time that linearly grows. This linear trend is caused by the fact that nodes could end up to send more ADV in order to correct a non best route. This is one of the consequences of the *Path Exploration* problem in the central nodes.

Is then possible to conclude that the position of the node can highly impact the MRAI behaviour, signals from the periphery of the network can more easily cause ADV storms if MRAI is not large enough to catch them. Other than that, the position can influence the performances by itself, but MRAI can help to reduce the number of messages paying a higher convergence time. The single nodes load can be affected by both the position and the strategy used, a strategy that better reacts to the *Path Exploration* problem can prevent huge loads on the central nodes.

6 BGP RFD

6.1 RFD on toy topologies

6.2 RFC 2439 VS RFC 7196

6.3 Mice VS Elephants

6.3.1 Mice

6.3.2 Elephants

7 RFD and MRAI Interaction

RFD is another parameter of BGP used to prevent messages storms. It is used to avoid flapping routes to continuously make the network unstable. When a network flaps a certain value is increased and when it overpass a threshold then the route is suppressed and not advertised anymore until it goes back below the threshold (or after a certain time).

RFD, other than MRAI, is one of the most studied parameters of BGP because of its influence in the convergence time [11, 12]. RFD received different updates from its first implementation, but recent studies showed that most of the providers still use outdated parameters [5].

The use of deprecated values can lead to a heavy restrictive suppression of some routes, delaying the correct spreading of information. Some cases of suppression are caused by faulty interfaces that heavily flaps hundreds of times, while other times is just an update of the node configuration that cause the route to flaps a couple of times.

In the following chapters, I am going to show how legacy RFD can affect small flaps and how would the new version of RFD react to them. Finally, I would look forward to understanding the correlation between RFD and MRAI. When a suppressed route is shared again it could provoke messages storms that triggers different MRAI sessions, or the opposite case, a low MRAI that cause the growth of the figure of merit that suppresses a route.

7.1 RFD on toy topologies

I firstly studied RFD on toy topologies, to see the effects of it in small networks, like I did in Section 5.2. As a graph, I used a clique of dimension 10, the source of the signalling is connected to the node 0 while the node 5 act as unique servicer for the node x. The node 5 won't be able to share information to node x because of RFD. Node x would have to wait until the RFD value of 5 fell below the reuse threshold in order to be able to converge.

The parameters used for RFD are the default *CISCO* parameters, showed in table Table 7.1 and are going to be used by all the nodes.

Parameter	Value
withdrawal penalty	1.0
re-advertisement penalty	0.0
attribute change penalty	1.0
suppress threshold	2.0
half-life (min)	15 (900s)
Reuse Threshold	0.75
Max Suppress Time (min.)	60 (3600s)

Table 7.1: Cisco default RFD parameters

The parameters of the environment are in Table 7.2.
Messages in the signal are delayed by 300 s for two reasons:

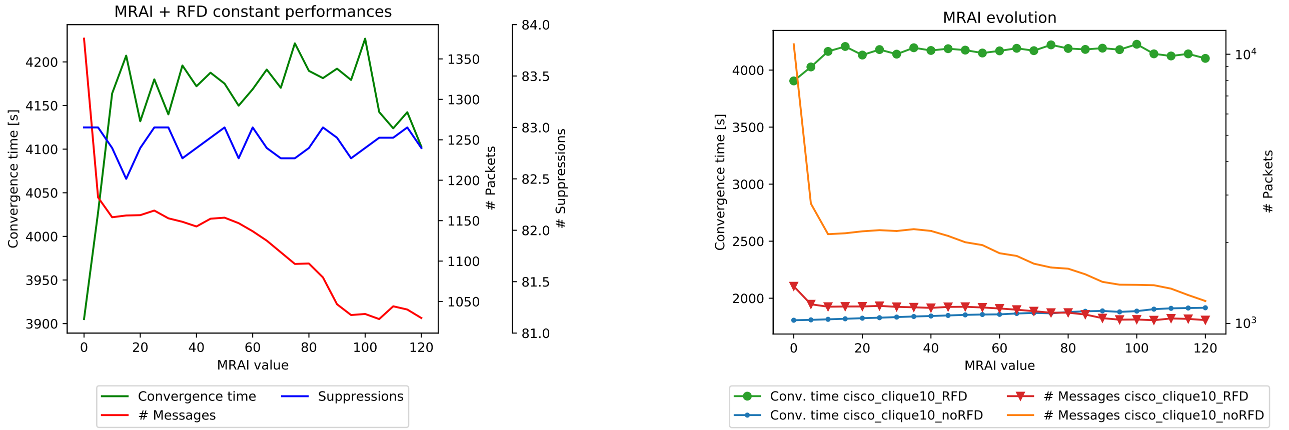
- We don't want that MRAI compress parts of the signal;
- I'm trying to simulate one of the possible behaviours tat triggers a RFD suppression, the human faulty reconfiguration of the node.

The signal contain 3 flaps, the first one is hypothetically attributed to a configuration that doesn't work properly, the second one is caused by a buggy correction of the configuration and the last one by the introduction of a correct configuration.

Property	Value
Seeds	$[1, 10]$
Signaling	“AWAWAWA”
Withdraws delay	Constant distribution of 300 s
Announcement delay	constant distribution of 300 s
MRAI	$[0, 120]$
Link delay	Uniform distribution between 0.012 s and 3 s

Table 7.2: Environment parameters used for the experiments on RFD with the clique graph

The MRAI strategy used in all the experiments is the *fixed* one.



(a) Network performances with the standard cisco RFD

(b) Network performances standard RFD vs no RFD

Figure 7.1: Evolution of the performances changing MRAI in the links standard RFD vs no RFD, graph clique of 10 nodes, MRAI strategy fixed, signal “AWAWAWA”

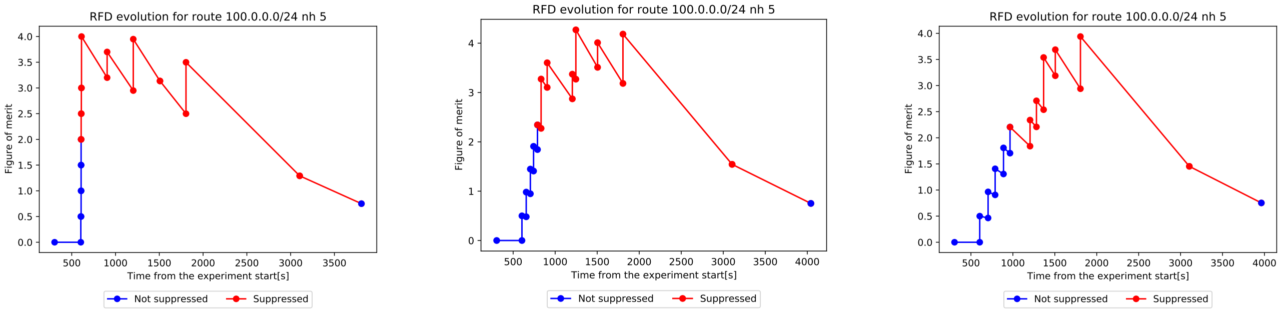
The plot in Figure 7.1a contains a third line that represent the average total number of suppression detected on the experiment, for each experiment has been executed 10 different runs. The blue line that represents the number of suppression refers to the third y-axis on the right.

In Fig. 7.1a is possible to see that small changes to MRAI can lead to some small differences in the number of suppressions. Also, the number of messages decreases rapidly and reaches a constant value around 1000, as expected by the passage from an MRAI of 0 s to a few seconds. The convergence time stays stable around 4000 s due to the fact that there are almost no variations in the number of suppression, and those suppressions always take the same time to be solved. A node wouldn’t be considered converged until it has also solved the suppressed routes.

In Figure 7.1b is possible to see the gap between the use of RFD and without it. Notice that the packet axis is in log scale. The difference in the convergence time is due to the fact that with RFD some nodes block the best path that takes a lot of time to become available again. While MRAI grows the set of nodes that suppress routes decreases but the convergence time is highly affected by a restrict subset of them. In our case, for example, the suppressions on nodes 0 and 5 play an important role. The first one for the spreading in the whole network, the second one for the transmission of information to node x.

For this reason, we can look more deeply on what happened to the figure of merit of node x and five in Figures 7.2 and 7.3. The blue points represent that the route has not been suppressed yet. Red points represent that the figure of merit has overpassed the suppression thresholds and the route has been blocked.

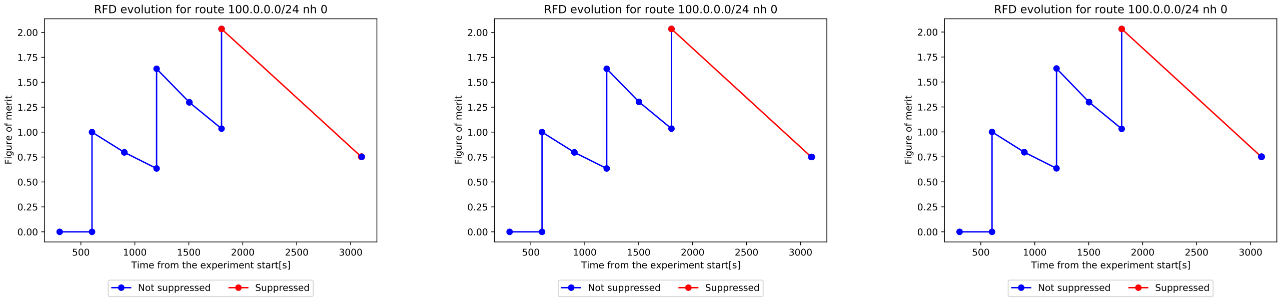
The node x is a leaf of the network that will absorb everything the node 5 sends to it. In Figure 7.2 is possible to see the evolution of the figure of merit with different MRAI values. In the first case, with an MRAI equal to 0 s, we will see a huge spike caused by a lot of messages and route changes that the



(a) MRAI = 0s (b) MRAI = 50s (c) MRAI = 100s

Figure 7.2: Evolution of the figure of merit in the node X with different MRAIs

node 5 sends to it. While in the other two cases Figures 7.2b and 7.2c the MRAI seems to not be much effective on the route through node 5. The messages are more delayed with high MRAI but the growth of the figure of merit has the same trend. We can see that the route has been suppressed around 1000 s and it is going to become available again around 4000 s. In this period of time, from 1000 s to 4000 s, node x still receives some updates from node 5 that affects its best path, and this makes the figure of merit evolve. The evolution of the figure of merit stops around 2000 s that's because also, the node 5 has suppressed the route, Figure 7.3, and doesn't send any more advertisements. The point around 3000 s represent the moment when the route becomes available again for node 5 that communicates the change to x.



(a) MRAI = 0s (b) MRAI = 50s (c) MRAI = 100s

Figure 7.3: Evolution of the figure of merit in the node X with different MRAIs

The evolution of the figure of merit of the best path of node 5 is different from the one of node x. In fact, it is not influenced by MRAI as we can see in Figure 7.3. That because the node 5 is directly connected to the node 0 that every 300 s forward the message of d. 300 s are a delay large enough to be not affected by the compression effect of MRAI. Around 2000 s node 5 suppress the route (as any other node in the clique) and stops to forward it to node x until 3000 s when it becomes available again.

Node x took almost 4000 s to converge because of the big fluctuations of node 5 that suffers from the *Path Exploration* problem, path changes are considered bad behaviour in RFD.

In conclusion, we can say that RFD can be affected by MRAI and that RFD can prevent a lot of messages at the cost of very high convergence times.

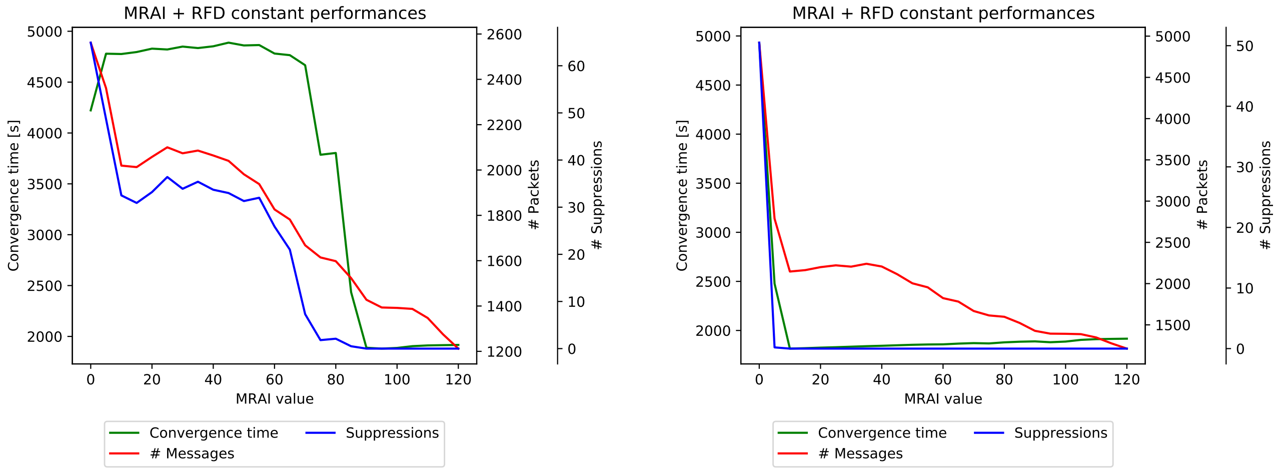
7.2 RFC 2439 VS RFC 7196

The difference in the two RFC that defines RFD [6, 15] is in the parameters used. In fact, the RFC 7196 modify the figure of merit threshold that is increased up to at least 6.0, introducing two new set of possible RFD filters that can be used:

- ***Aggressive***, Suppression threshold no less than 6.0;
- ***Conservative***, Suppression threshold no less than 12.0;

Respectively 3 and 6 times the actual standard.

I have then repeated the same experiments of Section 7.1 with the same clique graph, but with the two new RFD strategies, the results are showed in Figure 7.4.



(a) RFD 7196 Aggressive on the clique topology

(b) RFD 7196 Conservative on the clique topology

Figure 7.4: MRAI influence with different RFD strategies from [15]

We can see two completely different evolutions of the performances in Figure 7.4. On the left plot, we can see the evolution with the *Aggressive* strategy. MRAI is more effective to this strategy in respect to the standard one. The number of suppressions fell down to almost 0 with an MRAI near 90 s. The message trend is similar to the one of the case without RFD but with an important difference in the case of MRAI equal 0 s, the number of average messages is around 2600 in respect of the 10 000 without RFD. While with a high MRAI the message trends are similar and equal when the number of suppressions reach 0.

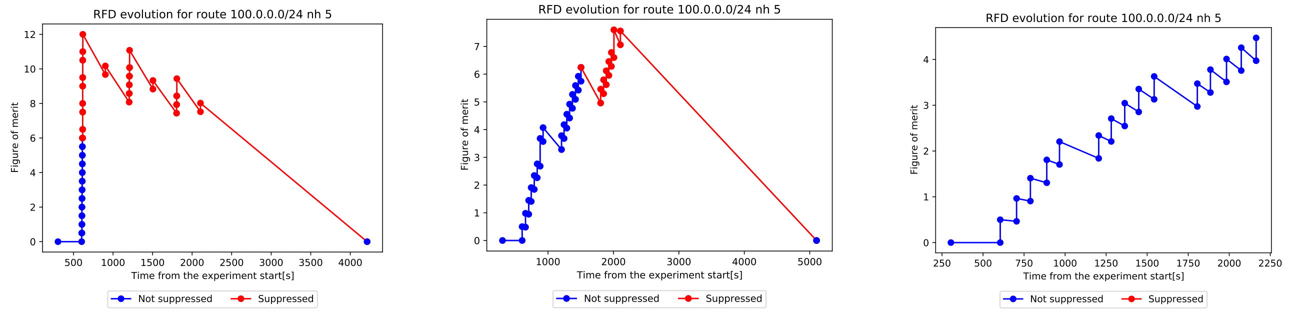
The convergence time, on the other hand, has a different trend in respect of the one that we saw in Figure 7.1a. Here we see a descending trend caused by the fact that MRAI is able to avoid some messages and, as a consequence, avoid the growth of the figure of merit in some nodes permitting to the convergence time to decrease. Once the number of suppressions reaches 0 obviously the network performances are equal as in the *NoRFD* case.

In Figure 7.4b we can see the evolution of the network with the *Conservative* strategy, the threshold of this strategy is the double of the *Aggressive* strategy. The effects of this difference are huge, is sufficient an MRAI of 10 s to avoid at all any suppression, causing the trend, in terms of messages and convergence time, to be equal to the no RFD case. Also with an MRAI of 0 s is possible to see a difference in terms of messages and convergence time in respect of the other two strategies. This is the strategy that more likely resembles the *NoRFD* one, having a convergence time incredibly more low, at the cost of few hundreds of messages.

We can now take a look more closely to what happens to the figure of merit for the only route that node x receives with an MRAI of 30 s. Results are showed in Figure 7.5, reporting the evolution of the figure of merit in the *Aggressive* strategy. The *Conservative* case is not presented because it is never going to suppress the route.

We can see in Figures 7.5a and 7.5b that MRAI plays an important role in the figure of merit of node x. In the first case, the route would be delayed up to 4000 s with a suppression value that touches 12 few seconds after the first flap. In the second one, the growth is slower but it passes the threshold around 1800 s reaching a value of 7. For this reason, it requires a higher time to become available again. We can also notice that the route become available with a figure of merit of 0, that's because it has been triggered the max suppression threshold, with the default value of cisco after 1 h a route would become available again, no matter the evolution of the figure of merit. With a higher MRAI node 5 is able to compress more routes, the effects are visible in Figure 7.5c, where the figure of merit never goes over the threshold.

In conclusion, if MRAI, with the standard RFD was playing a more marginal role because of the



(a) MRAI = 0s, RFD 7196 Aggressive, clique topology

(b) MRAI = 50s, RFD 7196 Aggressive, clique topology

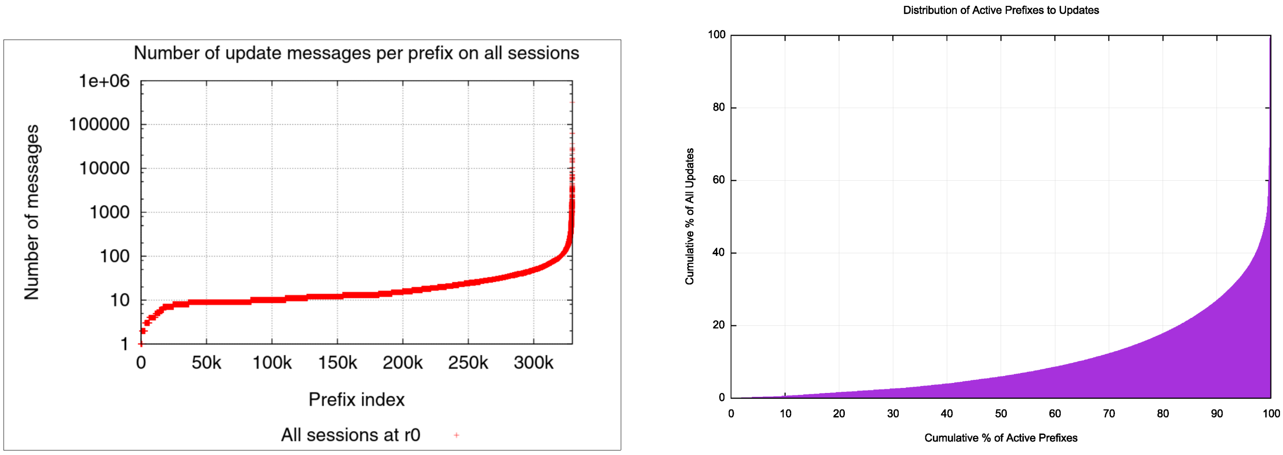
(c) MRAI = 100s, RFD 7196 Aggressive, clique topology

Figure 7.5: Evolution of the figure of merit in the node X with different MRAIs, with RFD 7196 aggressive in a clique topology

restrictive threshold, now, with those strategies, it plays a more relevant position and acts as a key factor between the suppression or not of a route.

7.3 Mice VS Elephants

From the work of R. Bush et al., [11] we know that the majority of the ADV that are transmitted on the Internet are from a small set of ASes. Those ASes with their flaps causes update storms almost continuously. I report a figure form [11] for simplicity in Figure 7.6a Thanks to the studies of APNIC[1] we also know that this behaviour is still present nowadays, the Figure 7.6b is taken from one of their annual reports and shows that 10% of all the active prefixes produce more or less the 70% of the total messages received.



(a) Prefixes and number of updates associated, figure from [11]

(b) Prefixes and number of updates associated, [apnic 2019]

Figure 7.6: Prefixes influence on updates

We can then divide those prefixes in two sets:

- ***Mice***, this set represent the majority of the prefixes, all the prefixes that does not generate more than 100 updates in Figure 7.6a
- ***Elephants***, this set represent the remaining part of the prefixes, those that produces the majority of the messages.

Thanks to a annual review of BGP by APNIC, presented at RIPE 52 [28], we can also have an example of those elephants prefixes. This example is shown in Figure 7.7, it takes in consideration the

[1] APNIC BGP 2019 report

prefix "202.64.49.0/24" showing that in a relatively small period of time it has produced thousands of ADV per day. In this case, this particular prefix has produced 198.370 ADV producing in total 96.330 flaps.

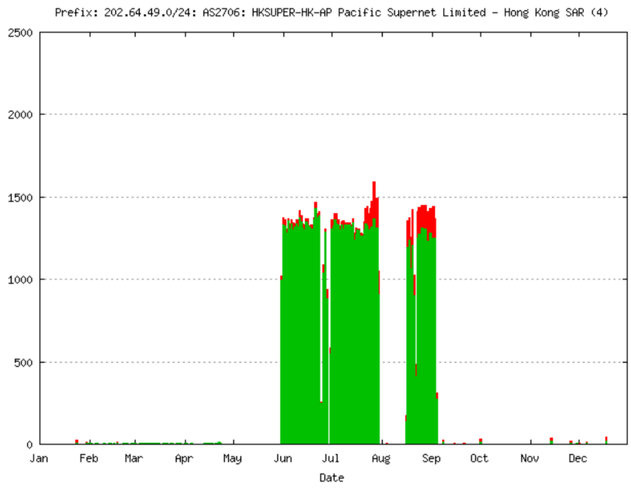


Figure 7.7: 202.64.49.0/24 flaps plot from [28]

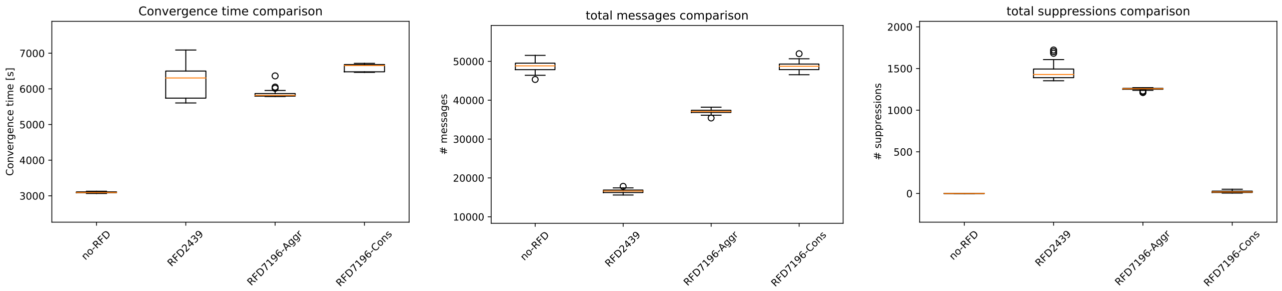
I have then used this data to configure two new environments for the simulations. The first one points to reproduce the *Mice* behaviour, the second one the *Elephants*.

In both these environments, I have then compared the four different strategies of RFD, *NoRFD*, standard RFD from the RFC 2439 and the two from [15].

The topology used for those experiments is an *Internet like* topology with 1000 nodes and MRAI is fixed to 30 s for all the links. The source of the signal has been chosen randomly on the graph. For each experiment has been executed 50 runs.

7.3.1 Mice

The particularity of the *Mice* experiments is in the signal, we have a low number of flaps interleaved by a long timer. I have then used a signal with 5 flaps, "AWAWAWAWAWA" with a delay of 300 s (5 min) between each message. The results are presented in Figure 7.8. I have executed 50 runs for each RFD strategy.



(a) Convergence time respect to the RFD strategy

(b) Number of messages respect to the RFD strategy

(c) Number of suppressions respect to the RFD strategy

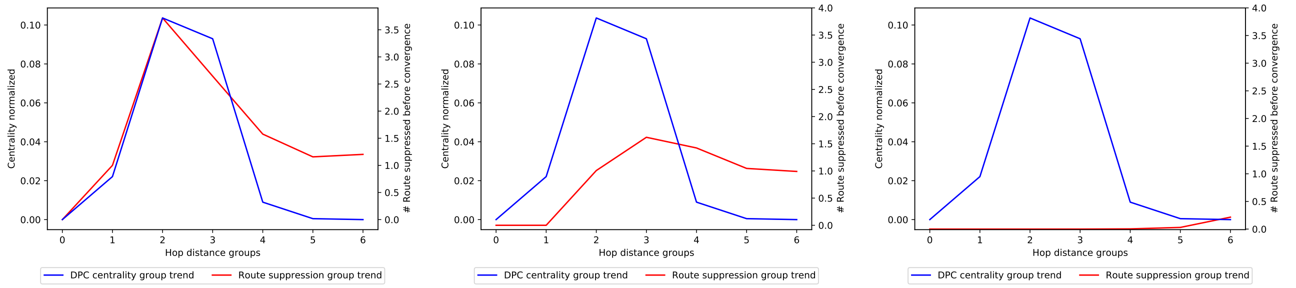
Figure 7.8: Internet like topology 1000 nodes, MRAI=30s, random destination, 5 flaps, 300 s message delay, Network performances, 50 runs per strategy.

From Figure 7.8c we can see that there is a big difference in the number of suppression. The standard strategy produces on average almost 1500 suppressions and the effects of those suppressions can be seen in Figures 7.8a and 7.8b. On average, it presents a convergence time higher than 6000 s but with a number of total messages transmitted around 16 000 with a very low variance. A different case is presented by the *Conservative* strategy from RFC 7196 [15]. The threshold in this last case is so permissive that we have a really small number of suppression. For this reason, the number of messages transmitted, on average, is similar to the *NoRFD* case, around 50 000. While, the convergence time is around 6500 s, like the standard RFD strategy. This proves that few suppression can heavily influence the network performances, in particular the convergence time. Also because the recover from a suppression with a higher threshold would require more time.

In the middle we find the *Aggressive* strategy, we can see from the suppression boxplot that it produces a smaller number of suppression in respect of the legacy strategy with a smaller variance. Also, the convergence time respect this trend, in fact, the average time is below 6000 s. While The number of messages transmitted is more than double in respect of the strategy described by the RFC 2439.

We can then conclude that a small number of suppression can affect both the performances, like the few suppressions in the *Conservative* strategy for the convergence time. Also, the few missing suppression in the *Aggressive* strategy will enormously impact the number of messages transmitted.

Is also possible to study which are the nodes that produce the suppression and how far are them from the signal source. We can see the results of this study, for each suppression technique in Figure 7.9.



(a) RFD 2439 Strategy

(b) RFD 7196 Aggressive Strategy

(c) RFD 7196 Conservative Strategy

Figure 7.9: Internet like topology 1000 nodes, MRAI = 30 s, random destination, 5 flaps, 300 s between messages, Suppression trend VS avg hop centrality

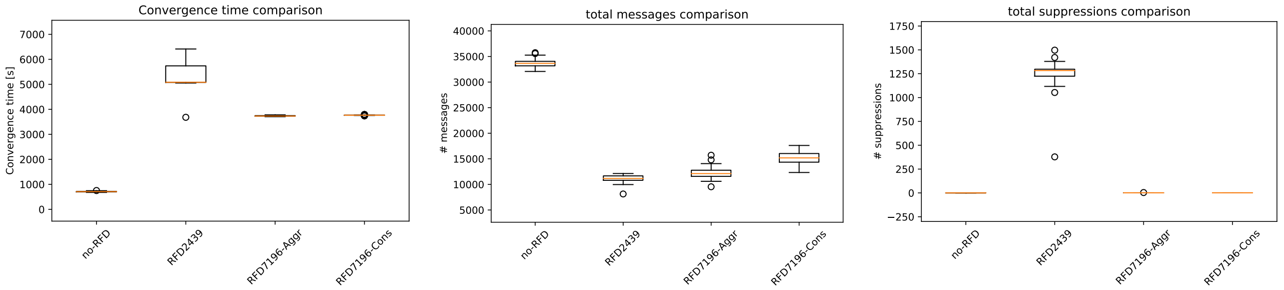
For the plots in Figure 7.9 the x axis represent the distance from the source node in terms of hops and all the other nodes are grouped by this distance. The blue line represents the average centrality of the groups, for each node of the graph I calculated the centrality using the DPC metric then grouped them and calculated the average value. As expected the central nodes have a higher centrality and them are a few hops of distance from the source node. The centrality trend is equal for each plot in Figure 7.9 because the graph and the source node are the same for each experiment.

The red line represents the average number of suppressions per group. As we can see with the standard strategy, Figure 7.9a, on average, the route has been blocked 1 time by the nearest nodes and then, this value increase reaching the center clique up to 3.5 times and then slowly decreases in the following groups. In the farthest group, we will still see on average 1 suppression. The *Aggressive* strategy, Figure 7.9b present a similar behaviour, the nearest nodes don't block the route, while the central nodes start blocking it with a maximum average of 1.6 times. After those central nodes, the farthest nodes, that have a low centrality will block it on average 1 time, like the legacy strategy. The *Conservative* strategy, presented in Figure 7.9c, has a different trend. We can see that the central nodes do not block the route, while only the farthest ones block it a few times, with an average value of 0.2 times. This can give us some hints, a very high threshold can promote the path exploration problem that will cause multiple update storms in farthest nodes.

From those experiments we can see that having a higher threshold could help to spread the knowledge near the source of the flaps, but once the *Path exploration* problem takes over, the nodes are going to suppress the destination. This is a good behaviour because it circumscribes an area in which the information can spread instead of blocking it almost everywhere. Those few suppression can highly impact in general the average convergence rate of the network. Is important to consider that a higher threshold means also a higher time to make the destination available again, maybe a new decay function should be considered.

7.3.2 Elephants

The elephants prefixes, as I mentioned in Section 7.3, are the ones that produce the majority of the ADV. And we also know, thanks to [28], that is possible to see over thousands of messages per day. For this reason, the *elephants* environment signal is composed of 100 flaps, with a delay between the messages of 3 s. All the other properties of the environment are unchanged. The results are presented in Figures 7.10 and 7.11.



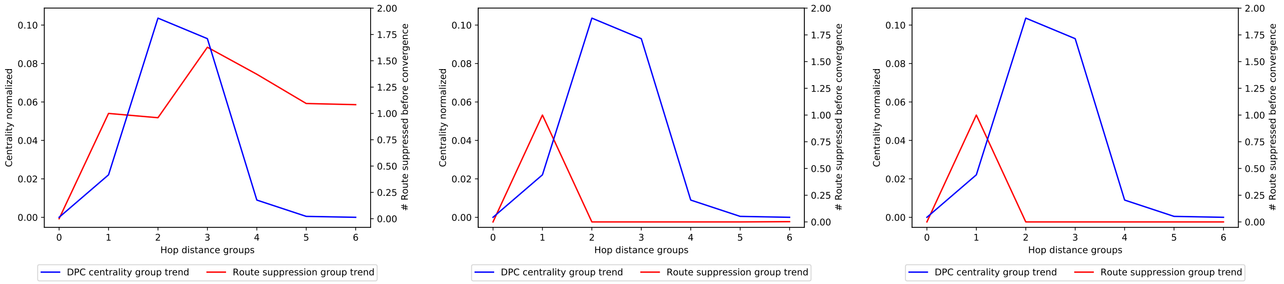
(a) Convergence time respect to the RFD strategy

(b) Number of messages respect to the RFD strategy

(c) Number of suppressions respect to the RFD strategy

Figure 7.10: Internet like topology 1000 nodes, MRAI = 30 s, random destination, 100 flaps, 3 s delay, Network performances

Is possible to see in Figure 7.10 that this time we have a different behaviour from all the 3 RFD strategies. In Figure 7.10c we can see that the standard strategy, on average, does more than 1250 suppression, producing the lowest number of messages, around 11 000, but the highest convergence time with more than 5000 s. All the suppression are trigger by the *Path Exploration* problem that causes ADV storms that trigger the suppression on the majority of the nodes. The two new strategies would produce on average just a few suppression in respect of the legacy one, but the number of messages doesn't differ too much. While there is a huge improvement on the convergence time, on average, both the new strategy permits the network to converge in less than 4000 s. All three strategy produce 1/3 of the messages produced by the *NoRFD* strategy.



(a) RFD 2439 Strategy

(b) RFD 7196 Aggressive Strategy

(c) RFD 7196 Conservative Strategy

Figure 7.11: Internet like topology 1000 nodes, MRAI = 30 s, random destination, 100 flaps, 3 s delay, suppressions by distance from the source

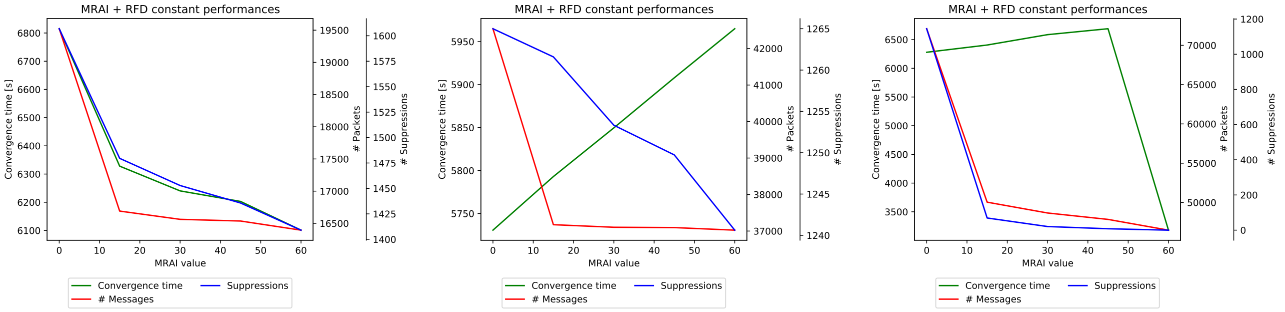
We can see in Figure 7.11 the comparison between the average number of suppressions per node group of the different strategies. In Figures 7.11b and 7.11c we can notice that both strategies reacts in the exact same way at the elephant environment. The only nodes that suppress the route are the nodes that are closer to the source. All the other nodes of the network don't experience enough messages to block the route. In the first figure, Figure 7.11a, we can see that, on average, every node suppress at least one time the source of the signal. The hypothesis behind this trend is that the intervention of the closer nodes is not timely enough and all the other nodes have the time to experience the *Path exploration* problem. With a lower threshold is sufficient a small number of ADV storms to trigger the RFD suppression.

We can say that all the strategies protect the network from a huge load of messages. In Figure 7.10b we can see that the use of RFD reduces to 1/3 the number of messages necessary to reach convergence.

The difference is the convergence time, more nodes experience suppression then more time is necessary to converge because there will be more ADV when the figure of merit becomes lower enough to activate again the route. For this reason there is a difference of more than 1000 s between the different techniques. This experiments reinforce the hypothesis that a small number of suppressions is more significant in respect of thousands of them.

7.3.3 MRAI influence on Mice and Elephants

We can now study the influence of MRAI on those two cases. The environments are equal to the previous section. The results of the *Mice* case are exposed in Figure 7.12, while the results of the elephant case are in Figure 7.13.



(a) RFD 2439 Strategy (b) RFD 7196 Aggressive Strategy (c) RFD 7196 Conservative Strategy

Figure 7.12: Internet like topology 1000 nodes, random destination, 5 flaps, 300 s delay, Network performances, MRAI strategy fixed

FiXme: Redo this graphs with more MRAI values, update the figures with the same y-range

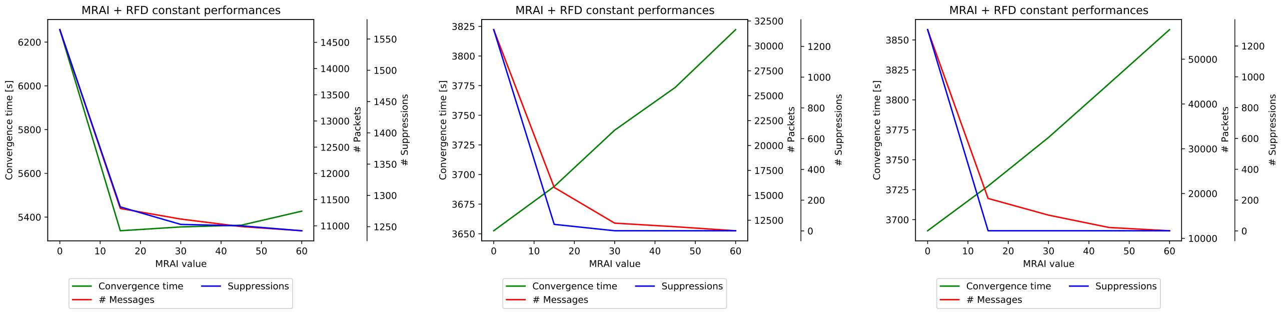
We can see in Figure 7.12 how the different RFD strategies react, on the same topology, with different MRAI settings. The network performances with the legacy RFD strategy from the RFC 2439 [6] are presented in Figure 7.12a. First of all, we can see the influence of MRAI on the number of suppressions that decrease from 1600 with an MRAI equal to 0 s to almost 1400 with MRAI = 60 s. We can notice that the messaging trend reacts as expected with the increasing of MRAI, but it is noticeable that with an MRAI of 0 s there are less than 20 000 messages thanks to RFD. The convergence time doesn't have the same trend as other MRAI experiments, it has a decreasing trend, while we were expecting an increasing one. This is caused by the routes that don't suppress anymore the route. It is greater the gain obtained by the suppression reduction than the disadvantage caused by MRAI that requires the node to wait more time.

The second case we can analyze is the *Aggressive* strategy presented in Figure 7.12b. We can easily notice that this time the variation in terms of suppression is smaller, going from a value of 1265 to 1240 suppressions. The number of messages has a similar trend to the standard strategy, but with a different range. At the beginning, with MRAI = 0 s there are more than 42 000 messages, after a while it converges around a value of 37 000 messages. The convergence time has the expected trend by the growth of MRAI. The number of messages higher is caused by the fact that RFD requires more time to activate itself, and in the meanwhile, a lot of messages storm will pass the network. The convergence time, in this case, is not affected by the decrease in the number of suppressions. The gain obtained by the RFD suppression trend doesn't compensate the effects of MRAI that makes the convergence time grow.

In the last strategy, the *Conservative* one, we can see another, different behaviour. In terms of suppressions MRAI makes a huge difference, we go from 1200 suppressions to 0 and just an MRAI of 15 s reduce it around 50. Also in this case the number of messages is higher than the other two strategies. Like before for the *Aggressive* strategy this is caused by the not restrictive thresholds. An interesting behaviour can be seen for the convergence time, in fact like for the *Aggressive* strategy it starts growing also with more than 1000 suppressions of difference, but as soon the number of suppression touches 0 it goes back to the *NoRFD* behaviour.

All this comparison can be also seen in Figures A.8 and A.9.

In Figure 7.13 are presented the results obtained with the elephant environment and multiple MRAI values.



(a) RFD 2439 Strategy (b) RFD 7196 Aggressive Strategy (c) RFD 7196 Conservative Strategy

Figure 7.13: Internet like topology 1000 nodes, random destination, 100 flaps, 3 s delay, Network performances

The trends in the elephant case are completely different in respect to the mice environment. Starting from the standard strategy in Figure 7.12a we can see that the number of suppression decreases of a few hundred units thanks to a higher MRAI. Also, the number of messages decrease from around 14 500 reaching a stable state around 11 000. While the convergence time benefits of the suppression rate decrease, reaching a valley around 5400 s. But, after that point, the effect of avoiding the next suppression set is not enough to keep a descending trend, while MRAI acquire a more predominant position making the convergence time slightly increasing.

A different behaviour can be saw in Figures 7.13b and 7.13c, where the number off suppression, thanks to MRAI, reaches a number slightly higher than 0. The number of messages reaches the same convergence point around 12 500 but with a completely different starting point. With MRAI at 0 s the aggressive strategy presents a number of messages around 32 500 while the conservative strategy is around 60 000, almost the double of the *Aggressive* one. This huge difference is caused by the fact that the *conservative* strategy requires more flaps to overcome the suppression threshold, and all those messages can cause more and more updates storms due to the *Path exploration* problem in the other parts of the network. In both, *Aggressive* and *Conservative* strategy the convergence time is not affected by the variation on the number of suppressions but it's only affected by the growth of MRAI.

The fact that in the last two strategies the time is not affected by the huge number of suppression could be saw as an error, but it is not. In fact, the suppressions with an MRAI of 0 s happens few seconds after the beginning of the experiment and the majority of the nodes will suppress the route, but we know that after 3600 s a route can't be suppressed anymore. For this reason after that time, there will be just a last update storm to propagate the reintroduction of the route (this will cause some suppressions too). And on average all the nodes will converge a few seconds after 1 h.

We can then conclude that MRAI influences both the *Mice* and *Elephants* cases. The major effects can be saw on the two modern strategies of RFD. For the *Mice* environment, those two strategies will tend to have a behaviour similar to the *NoRFD* strategy. And MRAI would influence the number of suppressions and indirectly the convergence time and the number of messages transmitted. We can see from Figure A.9 that also the set of nodes affect by suppressions changes. We can see even more effects in the *Elephants* environment, where MRAI would affect the number of suppressions. Both the *Aggressive* and *Conservative* strategies would present just few suppression in comparison of the thousands of suppression triggered with the legacy 2439 strategies. And the new strategies would have a high impact on the convergence time, at the cost of a few hundred messages on average. In Figure A.11 is possible to see the effects on the set of nodes that effectively suppress the route, in the legacy case even the more distance nodes would suppress it, while with the new strategies is sufficient a suppression near the source, and MRAI would help to prevent suppressions due to the *Path exploration* problem.

8 Conclusion

In this thesis, I exposed different noise problems that BGP contains and studied the parameters that are used to curb the problem. I have then analyzed the results from thousands of experiments in order to provide a solid baseline that shows how MRAI and RFD are related one another. I have also shown that, due to the *Path exploration* problem, even in small networks is really difficult to infer the causes behind a transmitted signal.

The instruments developed during this thesis are publicly available in the hope that other scientist could use them to study properties of BGP that is an extremely vast protocol.

I have then analyzed how new techniques differ from the standard or legacy one, like in the case of RFD where the legacy values are still present on the internet and can have a huge impact on the convergence. I also studied the impact that can have, in terms of performances, MRAI on RFD, how after a certain threshold the gain obtained from the lower number of suppression is ininfluent in terms of convergence time and messages transmitted.

This thesis creates the basis for studies on the correlation of BGP parameters and would also be a warning for those studies that point to improve only one aspect of BGP. Remember to always look from a different perspective because what looks like an improvement, on the one hand, could bring the overall performance of the network to collapse.

8.1 Future Works

The development of the platform to increment the number of features is one of the major points in the future works, but during the experiments we have also made some assumptions/restrictions to the environment. Those restrictions could be relaxed to study more heterogeneous environments.

Policies

One of the first assumption is that every node accept everything comes from its neighbours and redistribute it. This is not always true, ASes on the internet can have any sort of policy, checking any possible attribute of the message received. A possible future work could be to study the bibliography behind those policies in order to be able to implement them and study again the performances of the network with those restrictions.

Multiple destinations and path aggregation

During the experiments I never introduced more than one destination subjected to a signal, even if the DES permits to have multiple of them. Obviously more destinations could produce more messages and more ADV storms, but a possible interesting point could be to see the reactions of the nodes with the path aggregation activated and how it can impact the performances.

Bibliography

[1] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," RFC 4271, Internet Engineering Task Force, Tech. Rep. 4271, Jan. 2006, updated by RFCs 6286, 6608, 6793, 7606, 7607, 7705.

[2] 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.

[3] 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.

[4] J. Qiu, R. Hao, and X. Li, "The optimal rate-limiting timer of bgp for routing convergence," *IEICE Transactions on Communications*, vol. 88, no. 4, pp. 1338–1346, 2005.

[5] C. Gray, C. Mosig, R. Bush, C. Pelsser, M. Roughan, T. C. Schmidt, and M. Wahlisch, "Bgp beacons, network tomography, and bayesian computation to locate route flap damping," in *Proceedings of the ACM Internet Measurement Conference*, 2020, pp. 492–505.

[6] C. Villamizar, R. Chandra, and R. Govindan, "Bgp route flap damping," RFC 2439, Tech. Rep., 1998.

[7] 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.

[8] P. Jakma, "Revised default values for the bgp'minimum route advertisement interval'," *draft-jakma-mrai-02. txt (Internet Draft)*, 2008.

[9] ——, "Revisions to the bgp'minimum route advertisement interval'," *Internet Draft draft-ietf-idr-mrai-dep-02*, 2010.

[10] M. Milani, "BGP e Load Centrality: Implementazione del calcolo della centralità nel protocollo BGP." [Online]. Available: http://dit.unitn.it/locigno/preprints/Milani_Mattia_laurea_2017_2018.pdf

[11] C. Pelsser, O. Maennel, P. Mohapatra, R. Bush, and K. Patel, "Route flap damping made usable," in *International Conference on Passive and Active Network Measurement.* Springer, 2011, pp. 143–152.

[12] Z. M. Mao, R. Govindan, G. Varghese, and R. H. Katz, "Route flap damping exacerbates internet routing convergence," in *Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*, 2002, pp. 221–233.

[13] P. Smith and C. Panigl, "Ripe routing working group recommendations on route-flap damping," *ripe-378, May*, 2006.

[14] R. Bush, C. Pelsser, M. Kuhne, O. Maennel, K. Mohapatra, P.and Patel, and R. Evans, "Ripe routing working group recommendations on route-flap damping," *ripe-580, January*, 2013.

[15] C. Pelsser, R. Bush, K. Patel, P. Mohapatra, and O. Maennel, “Making route flap damping usable,” RFC 7196, Tech. Rep., 2014.

[16] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, “Delayed internet routing convergence,” *ACM SIGCOMM Computer Communication Review*, vol. 30, no. 4, pp. 175–187, 2000.

[17] 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.

[18] M. Milani, M. Nesler, M. Segata, L. Baldesi, L. Maccari, and R. L. Cign o, “Improving bgp convergence with fed4fire+ experiments,” in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2020, pp. 816–823.

[19] 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.

[20] 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.

[21] T. G. Griffin, “A Finite State Model Update Propagation for Hard-State Path-Vector Protocols,” accessed: 19-01-2021. [Online]. Available: https://github.com/tiamilani/BGPFSM/blob/master/Biblio/FSM_model.pdf

[22] T. G. Griffin, F. B. Shepherd, and G. Wilfong, “The stable paths problem and interdomain routing,” *IEEE/ACM Transactions On Networking*, vol. 10, no. 2, pp. 232–243, 2002.

[23] S. Deshpande and B. Sikdar, “On the impact of route processing and mrai timers on bgp convergence times,” in *IEEE Global Telecommunications Conference, 2004. GLOBECOM’04.*, vol. 2. IEEE, 2004, pp. 1147–1151.

[24] L. Maccari and R. Lo Cigno, “Improving Routing Convergence With Centrality: Theory and Implementation of Pop-Routing,” *IEEE/ACM Trans. on Networking*, vol. 26, no. 5, pp. 2216–2229, Oct. 2018.

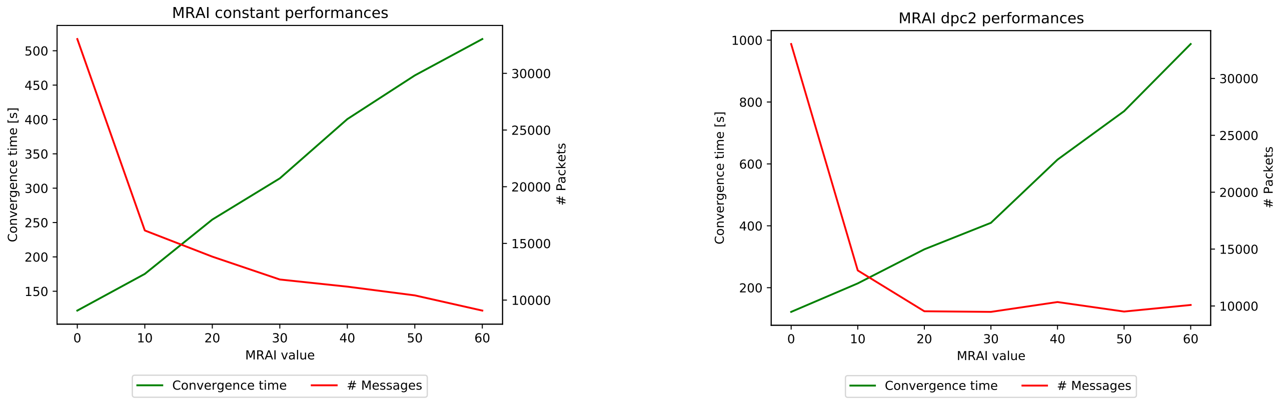
[25] L. Maccari, L. Ghiro, A. Guerrieri, A. Montresor, and R. Lo Cigno, “On the Distributed Computation of Load Centrality and Its Application to DV Routing,” in *37th IEEE Int. Conf. on Computer Communications (INFOCOM)*, Honolulu, HI, USA, Apr. 2018, pp. 2582–2590.

[26] U. Brandes, “On Variants of Shortest-Path Betweenness Centrality and their Generic Computation,” *Social Networks*, vol. 30, no. 2, pp. 136–145, May 2008.

[27] E. Goodarzi, M. Ziaei, and E. Z. Hosseinipour, *Introduction to optimization analysis in hydrosystem Engineering*. Springer, 2014.

[28] G. Huston, “a bgp year in review,” 2006. [Online]. Available: https://meetings.ripe.net/ripe-52/presentations/ripe52-plenary-bgp-review.pdf

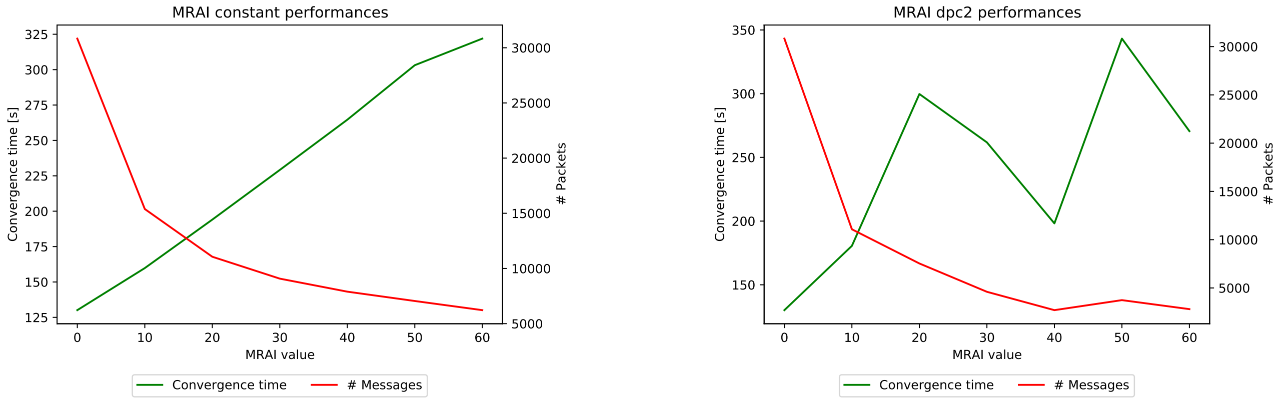
Appendix A Appendix



(a) Network perforcances, *fixed* MRAI strategy

(b) Network perforcances, DPC MRAI strategy

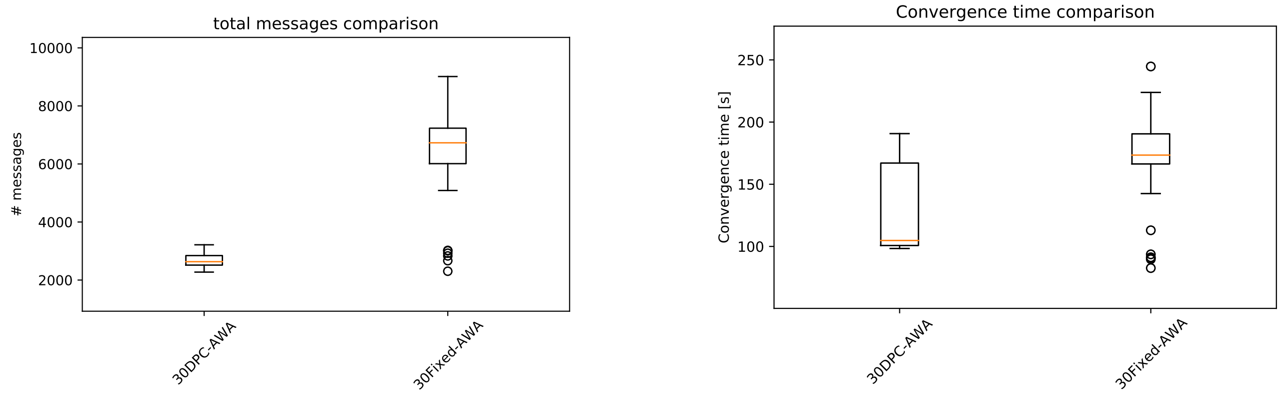
Figure A.1: Network perfomances comparison with different MRAI strategies, Graph internet like with 1000 nodes, signal “AWAW”



(a) Network perforcances, *fixed* MRAI strategy

(b) Network perforcances, DPC MRAI strategy

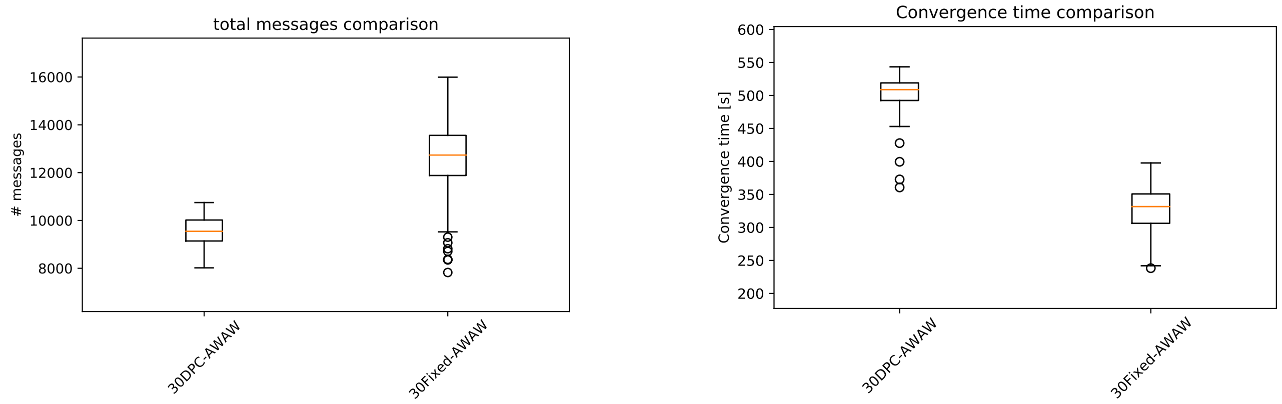
Figure A.2: Network perfomances comparison with different MRAI strategies, Graph internet like with 1000 nodes, signal “AWAWA”



(a) Network perforcances, messages necessary to reach convergence with different MRAI strategies

(b) Network perforcances, time required to reach convergence with different MRAI strategies

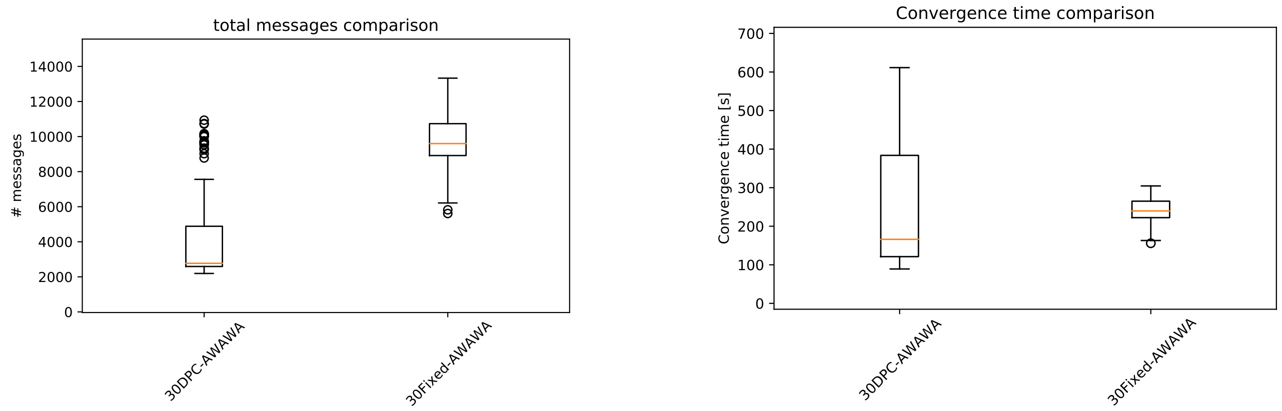
Figure A.3: Network perfomances comparison with different MRAI strategies, Graph internet like with 1000 nodes, MRAI value 30 s, number of runs for each strategy 100, signal "AWA"



(a) Network perforcances, messages necessary to reach convergence with different MRAI strategies

(b) Network perforcances, time required to reach convergence with different MRAI strategies

Figure A.4: Network perfomances comparison with different MRAI strategies, Graph internet like with 1000 nodes, MRAI value 30 s, number of runs for each strategy 100, signal "AWAW"



(a) Network perforcances, messages necessary to reach convergence with different MRAI strategies

(b) Network perforcances, time required to reach convergence with different MRAI strategies

Figure A.5: Network perfomances comparison with different MRAI strategies, Graph internet like with 1000 nodes, MRAI value 30 s, number of runs for each strategy 100, signal "AWAWA"

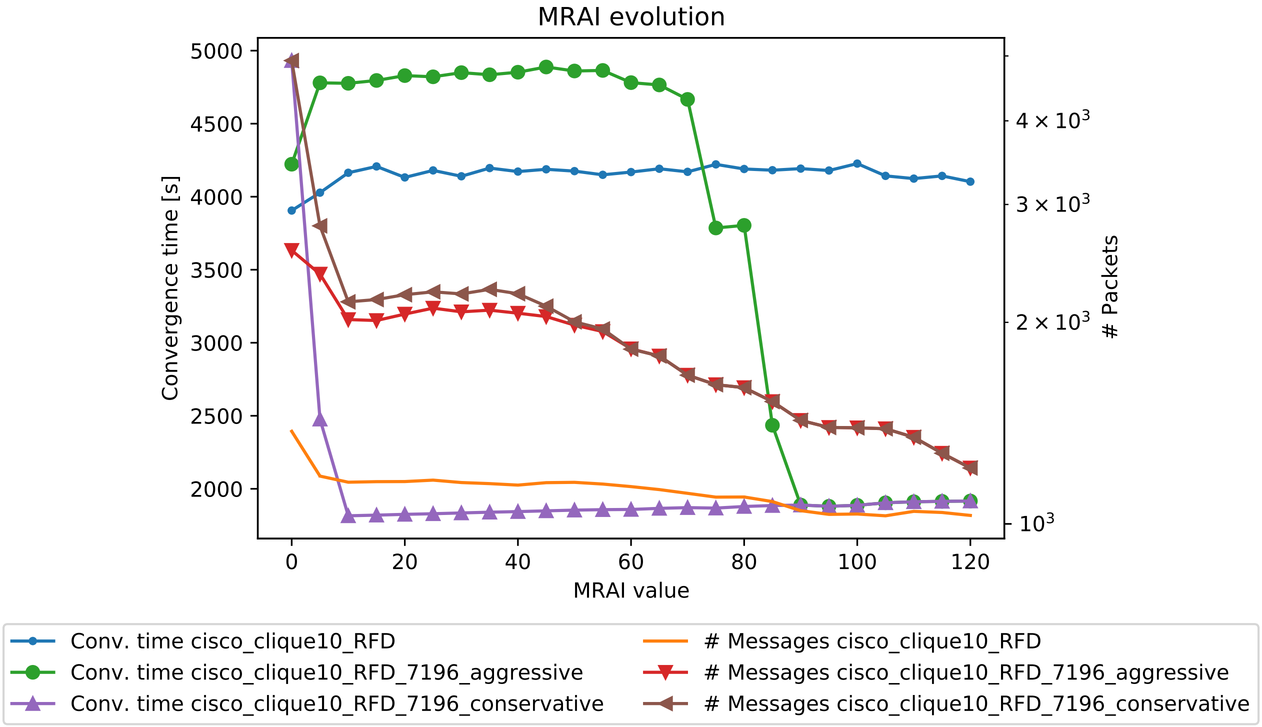
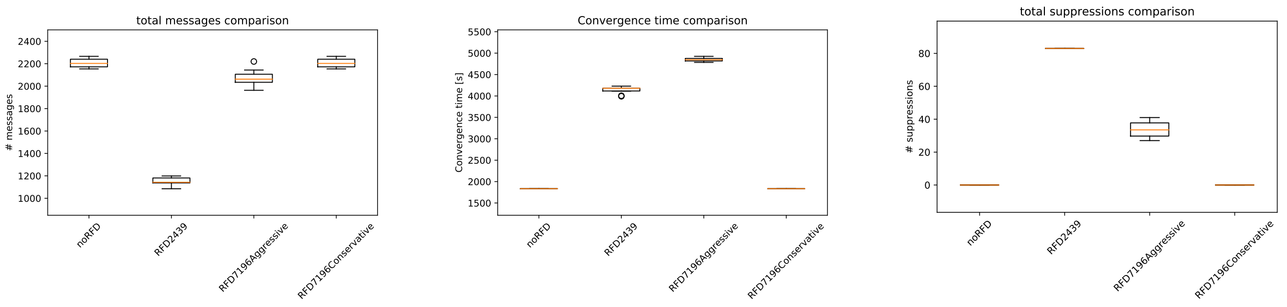


Figure A.6: Comparison of the *clique* topology with RFD 2439 and the with RFD 7196 strategies

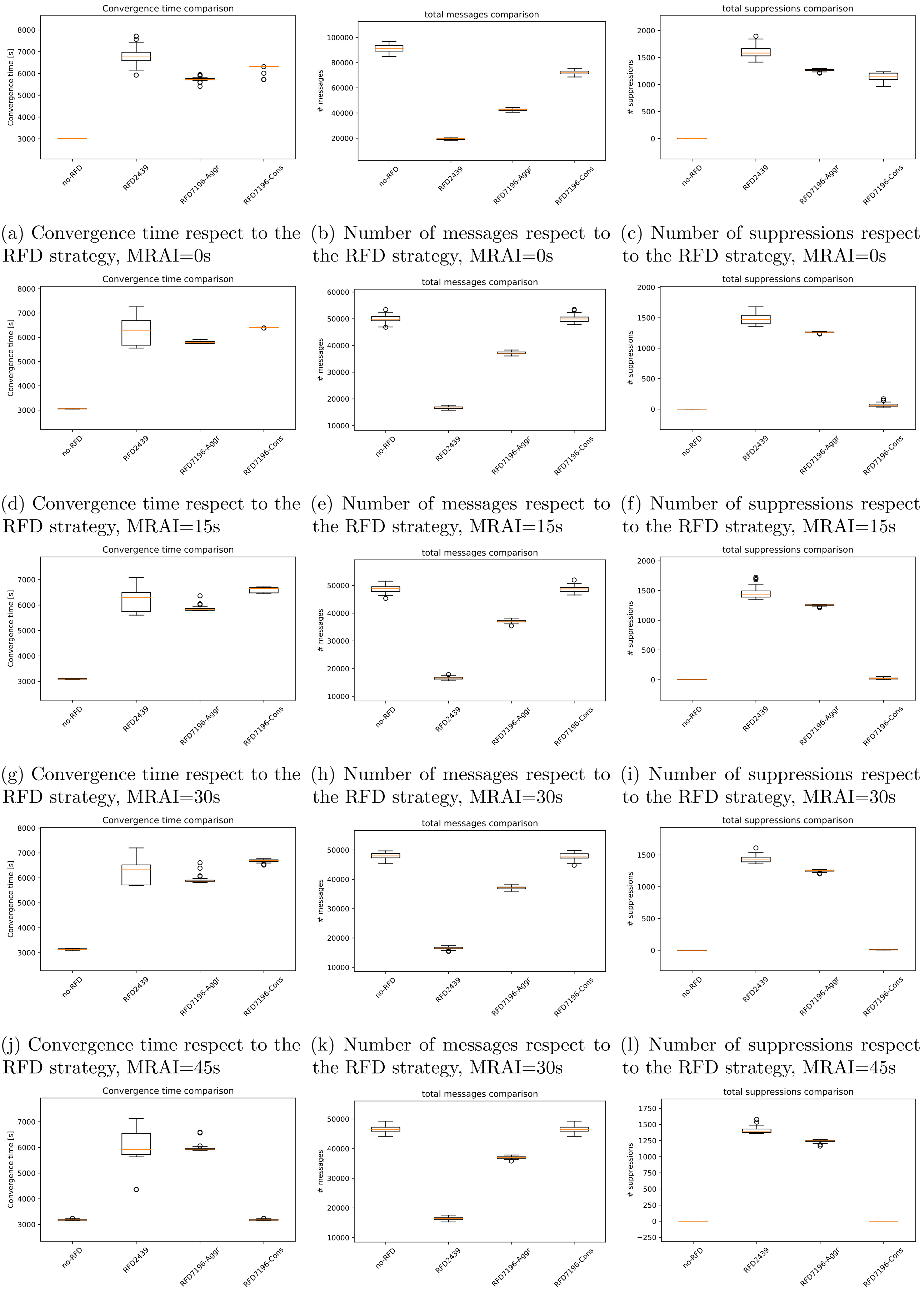


(a) clique topology, MRAI=30s, 10 runs, Messages comparison

(b) clique topology, MRAI=30s, 10 runs, Convergence time

(c) clique topology, MRAI=30s, 10 runs, Number of suppressions

Figure A.7: Clique topology, MRAI=30s, 10 runs, comparison of the network performances



(a) Convergence time respect to the RFD strategy, MRAI=0s

(b) Number of messages respect to the RFD strategy, MRAI=0s

(c) Number of suppressions respect to the RFD strategy, MRAI=0s

(d) Convergence time respect to the RFD strategy, MRAI=15s

(e) Number of messages respect to the RFD strategy, MRAI=15s

(f) Number of suppressions respect to the RFD strategy, MRAI=15s

(g) Convergence time respect to the RFD strategy, MRAI=30s

(h) Number of messages respect to the RFD strategy, MRAI=30s

(i) Number of suppressions respect to the RFD strategy, MRAI=30s

(j) Convergence time respect to the RFD strategy, MRAI=45s

(k) Number of messages respect to the RFD strategy, MRAI=30s

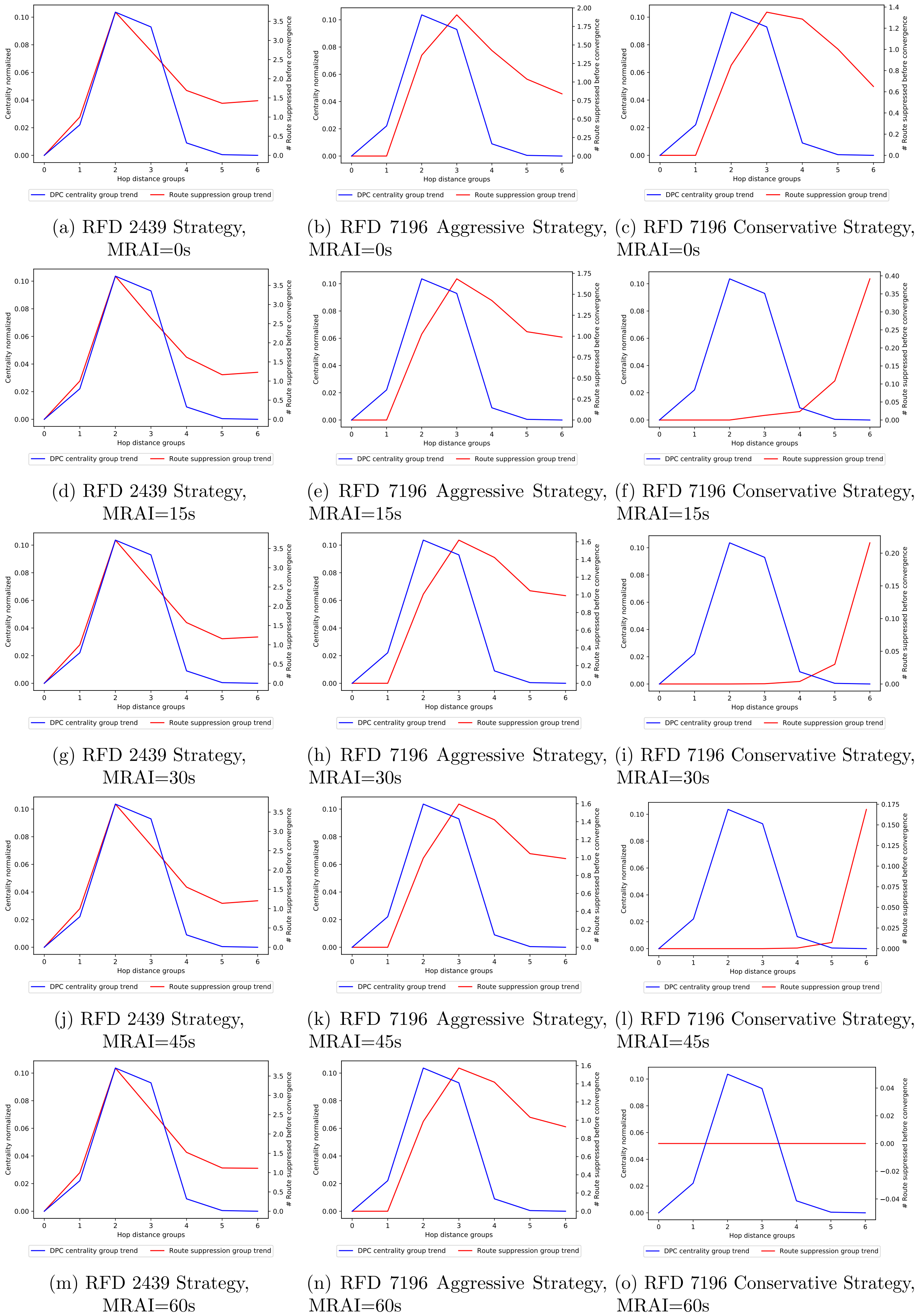
(l) Number of suppressions respect to the RFD strategy, MRAI=45s

(m) Convergence time respect to the RFD strategy, MRAI=60s

(n) Number of messages respect to the RFD strategy, MRAI=60s

(o) Number of suppressions respect to the RFD strategy, MRAI=60s

Figure A.8: Internet like topology 1000 nodes, random destination, 5 flaps, 300s delay, Network performances



(a) RFD 2439 Strategy, MRAI=0s

(b) RFD 7196 Aggressive Strategy, MRAI=0s

(c) RFD 7196 Conservative Strategy, MRAI=0s

(d) RFD 2439 Strategy, MRAI=15s

(e) RFD 7196 Aggressive Strategy, MRAI=15s

(f) RFD 7196 Conservative Strategy, MRAI=15s

(g) RFD 2439 Strategy, MRAI=30s

(h) RFD 7196 Aggressive Strategy, MRAI=30s

(i) RFD 7196 Conservative Strategy, MRAI=30s

(j) RFD 2439 Strategy, MRAI=45s

(k) RFD 7196 Aggressive Strategy, MRAI=45s

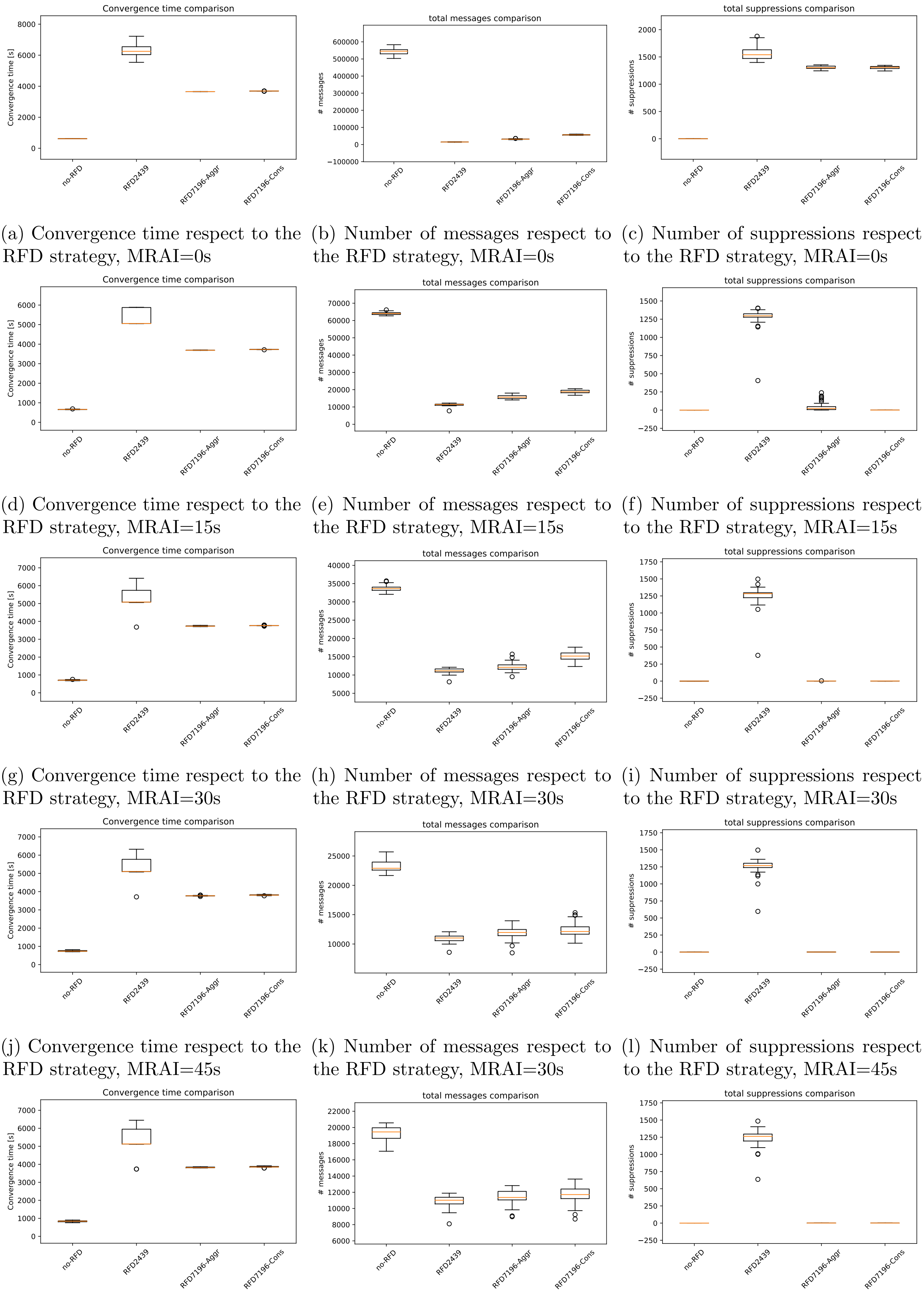
(l) RFD 7196 Conservative Strategy, MRAI=45s

(m) RFD 2439 Strategy, MRAI=60s

(n) RFD 7196 Aggressive Strategy, MRAI=60s

(o) RFD 7196 Conservative Strategy, MRAI=60s

Figure A.9: Internet like topology 1000 nodes, random destination, 5 flaps, 300s delay, Suppression trend VS avg hop centrality



(a) Convergence time respect to the RFD strategy, MRAI=0s

(b) Number of messages respect to the RFD strategy, MRAI=0s

(c) Number of suppressions respect to the RFD strategy, MRAI=0s

(d) Convergence time respect to the RFD strategy, MRAI=15s

(e) Number of messages respect to the RFD strategy, MRAI=15s

(f) Number of suppressions respect to the RFD strategy, MRAI=15s

(g) Convergence time respect to the RFD strategy, MRAI=30s

(h) Number of messages respect to the RFD strategy, MRAI=30s

(i) Number of suppressions respect to the RFD strategy, MRAI=30s

(j) Convergence time respect to the RFD strategy, MRAI=45s

(k) Number of messages respect to the RFD strategy, MRAI=30s

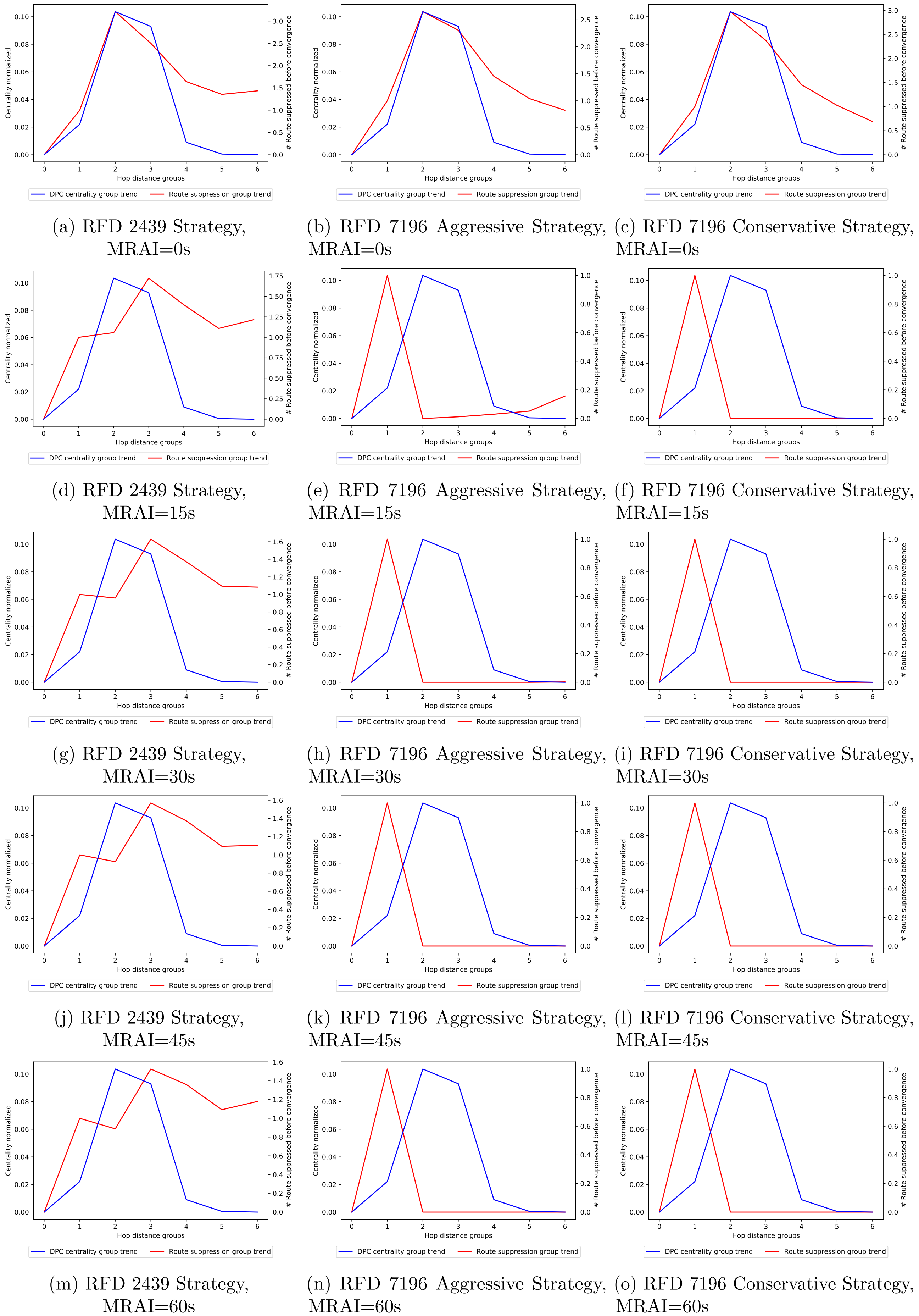
(l) Number of suppressions respect to the RFD strategy, MRAI=45s

(m) Convergence time respect to the RFD strategy, MRAI=60s

(n) Number of messages respect to the RFD strategy, MRAI=60s

(o) Number of suppressions respect to the RFD strategy, MRAI=60s

Figure A.10: Internet like topology 1000 nodes, random destination, 100 flaps, 3s delay, Network performances



(a) RFD 2439 Strategy, MRAI=0s

(b) RFD 7196 Aggressive Strategy, MRAI=0s

(c) RFD 7196 Conservative Strategy, MRAI=0s

(d) RFD 2439 Strategy, MRAI=15s

(e) RFD 7196 Aggressive Strategy, MRAI=15s

(f) RFD 7196 Conservative Strategy, MRAI=15s

(g) RFD 2439 Strategy, MRAI=30s

(h) RFD 7196 Aggressive Strategy, MRAI=30s

(i) RFD 7196 Conservative Strategy, MRAI=30s

(j) RFD 2439 Strategy, MRAI=45s

(k) RFD 7196 Aggressive Strategy, MRAI=45s

(l) RFD 7196 Conservative Strategy, MRAI=45s

(m) RFD 2439 Strategy, MRAI=60s

(n) RFD 7196 Aggressive Strategy, MRAI=60s

(o) RFD 7196 Conservative Strategy, MRAI=60s

Figure A.11: Internet like topology 1000 nodes, random destination, 100 flaps, 3s delay, Suppression trend VS avg hop centrality

Abbreviations

ADV advertisement

APNIC Asia-Pacific Network Information Centre

AS Autonomous System

BGP Border Gateway Protocol

c2p customer-to-provider

CAIDA Center for Applied Internet Data Analysis

CFSM Communicating Finite-State Machine

DES Discrete Event Simulator

DPC Destination Partial Centrality

DV Distance Vector

eBGP Exterior BGP

FSM Finite State Machine

iBGP Interior BGP

IETF Internet Engineering Task Force

IP Internet Protocol

ISP Internet Service Providers

IW Implicit Withdraw

LS Link State

MRAI Minimum Route Advertisement Interval

p2p peer-to-peer

PV Path vector

RFC Request For Comment

RFD Route Flap Damping

RIB Routing Information Base

RIPE Réseaux IP Européens Network Coordination Centre

RNG Random Number Generator

s2s sibling-to-sibling

SPP Stable Paths Problem

SSP Stratified Shortest Path

TCP Transmission Control Protocol