DeepBGP: A Machine Learning Solution to reduce BGP Routing Convergence Time by Fine-Tuning MRAI

UFAM

Ricardo Bennesby

Instituto de Computação
Universidade Federal do Amazonas

Manaus-AM November 2019

DeepBGP: A Machine Learning Solution to reduce BGP Routing Convergence Time by Fine-Tuning MRAI

UFAM

Ricardo Bennesby

Supervisor: Prof Edjard Souza Mota, Ph.D.

Instituto de Computação
Universidade Federal do Amazonas

A Thesis presented to PPGI/UFAM as a fullfiment of the requirements for the degree of
Doctor of Informatics (D. Sc.)

Manaus-AM November 2019

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

Silva, Ricardo Bennesby da
S586d DeepBGP: A Machine Learning Solution to reduce BGP Routing Convergence Time by Fine-Tuning MRAI / Ricardo Bennesby da Silva. 2019
141 f.: il. color; 31 cm.

Orientador: Edjard Souza Mota
Tese (Doutorado em Informática) - Universidade Federal do Amazonas.

1. bgp. 2. convergence time. 3. lstm. 4. network management. 5. interdomain routing. I. Mota, Edjard Souza II. Universidade Federal do Amazonas III. Título

FOLHA DE APROVAÇÃO

" A Machine-Learning Solution to reduce BGP Routing Convergence Time in a Hybrid SDN-Interdomain environment by Fine-Tuning MRAI"

RICARDO BENNESBY DA SILVA

Tese de Doutorado defendida e aprovada pela banca examinadora constituída pelos Professores:

Prof. Edjard de Souza Mota - PRESIDENTE

Prof. Eduardo Luzeiro Feitosa - MEMBRO INTERNO

Prof.ª Eulanda Miranda dos Santos - MEMBRO INTERNO

Prof. José Neuman de Souza - MEMBRO EXTERNO

Prof. Ítalo Fernando Scotá Cunha - MEMBRO EXTERNO

Manaus, 18 de Novembro de 2019

EPISTLES XXXIII., XXXIV.

Never be discovered if we rest contented with discoveries already made. Besides, he who follows another not only discovers nothing but is not even investigating. What then? Shall I not follow in the footsteps of my predecessors? I shall indeed use the old road, but if I find one that makes a shorter cut and is smoother to travel, I shall open the new road. Men who have made these discoveries before us are not our masters, but our guides. Truth lies open for all; it has not yet been monopolized. And there is plenty of it left even for posterity to discover. Farewell.

Seneca's Epistles Volume I, 1976

Acknowledgement

Tomo a liberdade de exibir, de modo especial, essa parte da tese na língua portuguesa.

Agradeço primeiramente a Deus pelo dom da vida e por todas as oportunidades que me foram dadas para crescer pessoal e profissionalmente.

À minha família, Madalena, Mara, Marilene, Gabi, Bruna, Lucas, Juliana, Jozimar e Sanches, por todo o apoio e orientações dados em cada decisão tomada em minha vida. Em especial à minha mãe, Madalena Cardoso, pelo amor e doação dedicados a mim. Devo muito a ela pela educação e amor recebidos. Sempre disposta a me ajudar a superar os momentos difíceis. É minha fortaleza; sem ela eu não teria chegado onde estou. Agradeimento também à minhas madrinhas Gilvete e Jandui por todo o carinho desde minha infância. Recordo também minha avó, dona Clarita, que partiu para junto de Deus.

À Késsia Cris, noiva e melhor amiga que levarei para a vida toda. Acompanha meus passos desde antes de iniciar a graduação. Sempre compreensiva e paciente nos momentos difíceis, me deu muita força nessa caminhada. Apoiou-me, aconselhou-me, incentivou-me. É um exemplo para mim também pela sua dedicação ao trabalho e extrema responsabilidade e amor nas coisas que faz. Agradeço também à sua família, por todo apoio, em especial a d. Nazaré, Sanara e sr. Jorge.

Ao professor Edjard Mota, meu orientador, por ter me acolhido no grupo de pesquisa a que faço parte, por ter me orientado em projetos de pesquisa ainda na graduação, no mestrado e no doutorado. Posso dizer que é um grande orientador, que me ouviu quando tive dificuldades, deu ideias quando preciso, mostrou os caminhos a seguir e esteve sempre presente para ajudar no trabalho. Mostrou a importância de trabalhar em grupo, ressaltando que é em equipe que somos mais fortes e temos conquistas mais significativas. Hoje é mais do que orientador, é um amigo o qual serei eternamente agradecido. Foram longos 11 anos de muito aprendizado, momentos desafiadores e muitas conquistas. Agradeço também meus companheiros do nosso grupo de pesquisa, o LIA, que passaram na minha trajetória em algum momento desses anos, em especial a Alexandre Passito e a Paulo César, grande amizade que cultivo desde o início da graduação e que também levarei para a vida. Passamos por muitas situações e desafios, e a experiência adquirida com eles foi incrível.

Aos meus amigos de UFAM, que estiveram comigo todos estes anos nesse processo de formação. Agradeço em especial ao Paulo Cesar, Flavio Montenegro, Bruno Dias, Jonathan Byron, Crysthian Carvalho, Emory Raphael e Hugo Cunha. A todos os professores do ICOMP/UFAM, pois tiveram papel importante na minha formação em Ciência da Computação. À FAPEAM, pela bolsa de estudo que me auxiliou financeiramente durante o desenvolvimento deste trabalho. Aos professores presentes em minha qualificação e defesa de doutorado, prof. Dr. Italo Cunha (UFMG), profa. Dra. Eulanda Santos, prof. Dr. Neuman (UFC) e prof. Dr. Eduardo Feitosa pelas contribuições valiosas para este documento e para a defesa da tese. Um agradecimento especial ao prof. Italo Cunha e à equipe responsável pelo projeto PEERING, que tiveram paciência de me deixarem utilizar os recursos do projeto por um longo tempo para realizar meus experimentos.

Agradeço também aos meus amigos da Paróquia São Pedro, em Petrópolis. Estão comigo desde a minha infância, e inegavelmente são parte importante do que hoje sou.

Abstract

The organization of the Internet is composed of administrative domains, known as Autonomous Systems (ASes), that exchange reachability information by means of the Border Gateway Protocol (BGP). Since a high convergence delay leads to packet losses and service unavailability, such a protocol has to converge as fast as possible. As this can happen due to BGP's own mechanism of UPDATE messages, that produces a humongous amount of messages, BGP reduces the number of UPDATEs exchanged between two BGP routers by holding consecutive announcements from a router to a neighbor for a given amount of time. The BGP timer responsible for this task is called Minimum Route Advertisement Interval (MRAI), which has an important impact in routing convergence. The Software-Defined Networking (SDN) paradigm can be used to leverage interdomain routing services performance via the logically centralized controlling benefits of intradomain settings. SDN principles has been successfully deployed in data centers, LANs, and in several other studies, where each each AS is modeled with a logically centralized routing control, offering new opportunities and bringing BGP routing convergence improvements. In this work, an extensive survey is presented on the state-of-the-art about research efforts to achieve better BGP routing convergence time. Furthermore, I pinpoint the open issues in this research field and propose *DeepBGP*, to the best of my knowledge, the first hybrid framework endowed with a learning mechanism, that integrates the SDN paradigm within interdomain routing domains, to improve the interdomain routing convergence time. This is achieved by employing the LSTM learning technique that allows the tuning of MRAI value aiming to reduce the convergence time according to learned patterns from collected BGP UPDATE features. The PEERING platform was used to provide a real scenario that allows the sending of announcements to the Internet. With the benefits of having such an actual testbed I carried out experiments with protocol characteristics that can impact the routing convergence. The experimental results show that the adaptive MRAI in the *DeepBGP* framework is able to reduce the BGP routing convergence time when compared to the use of static MRAIs.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Motivation

The current Internet is partitioned into network groups under administrative domains called Autonomous Systems (ASes). Communication and reachability of information across different ASes is performed by the Border Gateway Protocol (BGP) [13], which has been widely deployed for many years and serves as the *de facto* Interdomain routing protocol of the Internet. However, the structure imposed on the Internet by its adoption yields an architecture which lacks the flexibility to allow and foster innovation [14], [15], [16]. Routing inconsistencies and anomalies, policy conflicts, delayed convergence, and security inefficiency against DDoS attacks are, among others, some issues that the current architecture does not address.

One issue that has a considerable impact on interdomain routing performance is the BGP slow convergence. Whenever there is a topology change, which may occur due to policy changes, broken links or connections or the entrance of a new node, the convergence process is triggered, and while a stable state is not reached, packet losses and inconsistencies might take place. During this period, the rapid fluctuation in network reachability characterizes the routing instability [17]. BGP may take tens of minutes to converge when such changes happen in the network, and while a stable state is not reached [18].

The high convergence delay presented on the Internet incurs in service unavailability with packet loss and poor quality for applications [19]. With the rise of many real-time applications on the Internet such as Skype, network banking, and video conferences, the demand on traffic increased. The slow convergence of BGP, for example, heavily impacts on VOIP service, accounting for almost 90% of the dropped calls [20]. Besides that, performing traffic engineering is hard in current BGP implementation. This condition generates more traffic in the network, increasing the load and incurring in overhead at routers processing

the messages [21, 22]. In the last few years, a number of efforts have been made to improve interdomain routing convergence, proposing BGP modifications, changes on interdomain architecture or new routing protocols [23].

To reduce the rate in the huge number of update messages that arrive at BGP routers to be processed, generating a high overhead, BGP presents a timer that limits the number of messages propagated by BGP speakers, called Minimum Route Advertisement Interval (MRAI) [24]. This timer is very important in BGP convergence process. The standard value for MRAI between BGP routers from different ASes is 30 seconds, as recommended by the BGP RFCs [13, 25]. Griffin et al. [26], in contrast, argue that the default recommended value of 30 seconds used globally is somewhat arbitrary and heavily impacts on interdomain convergence. The MRAI values that incur in reduced convergence delay vary according to a number of factors, e.g., the network topology and traffic. However, to determine which MRAI values should be applied by the ASes to reduce convergence delay – thus improving interdomain routing performance – remains an open issue. Several solutions have been proposed to solve such problem, but the deployment of those solutions is a difficult task, since many of them require the adoption from a large number of ASes. This complexity makes the BGP behavior unpredictable and prone to errors. Such undesirable situation is mainly due to the fact that the Internet's architecture and infrastructure are tightly coupled.

An architecture that enables innovation must be extensible and abstract [16]. An architecture is extensible when it permits that new functionalities or applications be easily added or replaced, evolving according to new needs. An architecture is abstract when low-level details are not considered, making the creation of applications faster and trustful. Any significant change on network architecture involves considerable costs to network vendors and operators. With a logically centralized architecture, with the routing decisions separated from routers, many of these problems could be solved. A logically centralized Routing Control Platform (RCP) placed on each AS, with a full view of the whole domain and with the routers only performing the forwarding task, could be more easily configurable, manageable, and less prone to errors [15].

An approach that provides the separation between the network infrastructure and the architecture, creating new possibilities, is the Software Defined Networking approach (SDN) [27]. SDN effectively separates the network on data plane and control plane and is currently used on several datacenters. Although SDN and other recent approaches have potential to enable the separation between infrastructure and architecture, enabling architectural innovation, few solutions have been proposed to use this approach to address current interdomain issues.

With the separation between data and control planes provided by the SDN paradigm, the control application that runs on the SDN controller may have access to information

and statistics about the traffic and the network state inside its domain, due to the logically centralized fashion on which it is deployed. Although also possible, it is much harder to be done at networks with a fully distributed model, with the complexity of information and statistics collection increasing proportionally to the domain size. The information collected by the routing control application can then be used to detect meaningful patterns and create models that determine the MRAI value the AS should use to impact in reduced convergence delay in the interdomain routing scenario.

One promising way to make the routing application learn to recognize patterns from the observed data is through machine learning. The machine learning approach, with its several models and algorithms, has been extensively used to extract patterns from data [28]. Inspired in some features of the biological neural networks, the artificial neural networks are one of the most studied and deployed models of machine learning. They have been used by several companies and institutes to achieve outstanding performance on important and complex problems. However, as explained by [2] in their work to detect BGP anomaly, most solutions apply classic learning mechanisms to make real-time prediction based on features obtained without considering how traffic characteristics change in time. Due to this aspect found in BGP traffic, short-term features are probably not the best choice.

Considering this, we chose to create models based on a connectionist machine learning approach such as the *Long Short-Term memory* (LSTM), a type of Recurrent Neural Network (RNN), that is suitable to model problems involving sequential data, where the sequence of events is required to predict the desired output. A machine learning model based on LSTM networks can be used by the control application in an AS that deploys SDN to observe and learn from the data collected from switches when convergence events happen in the network, and then choose MRAI values according to presented values in implemented features. The control application can adapt to the network state and improve itself based on the experience obtained from the actions taken; in this case, the action of choosing a MRAI value that will impact in a reduced convergence time of the whole network of which the AS is part of.

1.2 Thesis Hypothesis

According to the presented motivation, I formulate the hypothesis of this thesis as follows:

The SDN paradigm endowed with AI learning techniques is a feasible approach to improve BGP routing convergence time among Autonomous Systems by learning patterns that make it possible to tune MRAI to a proper value.

1.3 Research Questions

The research questions of investigation in this thesis are as follows:

1. Which network features impact most on convergence time behavior?
2. How well can a LSTM-based Routing Application predict the convergence time behavior after a prefix announcement to the Internet?
3. How the MRAI forecasting solution performs when compared to BGP with standard MRAI? (Where the prefix announcement must be sent through the AS with the MRAI forecasting solution to reach the Internet).
4. How the MRAI prediction solution performs when compared to BGP with MRAI turned off? (Where the prefix announcement must be sent through the AS with the MRAI forecasting solution to reach the Internet).

1.4 Objectives

The main objective of this work is to provide a mechanism that improves interdomain routing by exploring the benefits of the SDN paradigm to estimate a MRAI value in a threshold that implies on reduced convergence time and minimized number of UPDATE messages exchanged, being able to adapt according to information collected from the network.

Additionally, this work has the following specific objectives:

1. Provide a LSTM-based Routing Application able to forecast the routing convergence time after a prefix announcement to the Internet.
2. Identify the network features that impact most on convergence time.
3. Provide a MRAI forecasting solution that improves the routing convergence when compared to BGP with standard MRAI value and to BGP with turned off MRAI.
4. Provide a solution that enables extensibility and abstraction, without constraints to ASes policy expression.

1.5 Thesis Contributions

The contributions of this thesis are as follows:

1. **A survey on approaches to reduce BGP convergence time:** This work provides a discussion on the efforts made to address the slow interdomain routing convergence problem. This discussion led to a survey on the state-of-the-art proposed solutions to address this issue [23], contributing to better understanding on BGP behavior.

2. **D-Forecaster: A LSTM-based BGP convergence time predictor:** A machine learning model based on LSTMs networks that predicts the BGP routing convergence by detecting and learning routing patterns is presented. The data used by the learning component is obtained from updates related to Beacon prefixes collected from public BGP update sources distributed through different Internet locations.

3. **The DeepBGP Framework:** With the knowledge obtained from the survey [23], analyzing the most promising paths to address this problem, a framework is proposed in this work by extending the SDN paradigm with the development of an interdomain routing application. This application enables the communication between ASes deploying SDN and legacy ASes, the collection of data and statistics on traffic from inside and outside of its domain, without the need to modify existing BGP.

4. **How to conduct real-time experiments with the DeepBGP framework:** This work describes in details how all the elements that comprise the DeepBGP framework are configured to perform the experiments presented in this thesis. Among the MRAI-pred elements are ExaBGP router, the SDN paradigm, VPN tunels and the Keras machine-learning tool. Besides, the datasets, the third-party tools, and the developed code used in this work is made available in open-source repository [29, 30] so that one can replicate the experiments and extend to other research purposes that can help, for example: (1) in the study of the reduction of path exploration by observing the order the updates arrive before an announcement, (2) to provide a better understanding on the effects in the use of adaptive MRAI timers in the interdomain routing convergence, (3) in the analysis of the applied policies impact in the BGP routing convergence, (4) in the BGP routing innovation without the need of modification and replacement of the current deployed protocol.

1.6 Thesis Outline

Chapter 2 presents a background on important topics related to this work, to give the reader necessary understanding on the content from the other chapters of this thesis. In addition, this chapter presents the main issues on interdomain routing convergence we observed in our literature review.

Chapter 3 provides a review on the efforts to reduce the BGP interdomain routing convergence time. This chapter content is a literature review in which the problem and the techniques that have been addressed by researchers on this topic are classified into five kinds of approaches: speeding up updates, limiting path exploration, efficient policy configuration, multipath and centralized control of the network. The characteristic of each type of approach is described in this chapter.

Chapter 4 presents the design of a "cognitive" component to detect and learn routing patterns in order to predict the BGP routing convergence time. To achieve that, we first filter BGP announcements related to Beacon prefixes, from collected updates obtained from public BGP update sources distributed through different Internet locations. Then, we extracted features from the collected data to use as input to machine learning models known as Long Short Term Memory (LSTM), suited to detect and learn patterns from time series. In addition, we describe how the LSTM-based convergence time prediction application can help network operators to make better decisions to investigate what can affect the interdomain routing performance.

Chapter 5 presents the main contribution of this thesis, i.e. a hybrid framework endowed with a learning mechanism, that integrates the SDN paradigm within Autonomous Systems, to improve the interdomain routing convergence time. This is achieved by employing the LSTM learning technique that allows the tuning of MRAI value aiming to reduce the convergence time according to learned patterns from collected BGP UPDATE features. Our framework uses VPN tunnels to connect to the PEERING MUX routers and to collect in real-time features from the BGP messages that its BGP routers receive in the SDN-AS to fine tune the MRAI value.

Chapter 6 provides a discussion on how our framework can be deployed and extended to new solutions. It also summarizes the thesis by presenting the limitations and future directions for this work. To conclude, a list the publications during the doctorate period is presented.

Chapter 2

Background and Issues on BGP Routing Convergence

2.1 Background

2.1.1 Border Gateway Protocol

The current Internet is divided into network groups under administrative domains called Autonomous Systems (ASes) [31]. To enable reachability information exchange between ASes, policy expression and other features necessary to Internet operation, the Border Gateway Protocol (BGP) is widely used. BGP [13] is a path vector protocol executed on some routers in an AS domain, generally in the border routers. The routers that run BGP are called BGP speakers. In this thesis, we refer to interdomain routing in the Internet and BGP interchangeably, since BGP is the only routing protocol that currently makes interdomain routing possible on Internet.

As BGP is a path vector protocol, one of its most important attributes is the AS_PATH, which carries the list of ASes that the information must traverse to reach its destination. Each AS is uniquely identified on the Internet by an attribute named AS_Number. Another important attribute is the Network Layer Reachability Information (NLRI), by which BGP supports Classless Interdomain Routing (CIDR). The NLRI carries the list of destinations that BGP is trying to inform its BGP neighbors about. Its format is comprised of one or more 2-tuples [length,prefix]. For example, the NLRI [24,192.168.1.0] has the prefix 192.168.1.0 and the 24-bit mask length, which may also be represented as 192.168.1.0/24. Figure 2.1 depicts the usage of the AS_PATH and NLRI attributes to exchange reachability information among ASes. In the figure, AS1 announces its NLRI 10.10.1.0/24 with AS_PATH [1] to

AS2, which announces it to AS3 with AS_PATH [2,1], so that a traffic flow from AS3 may reach AS1.

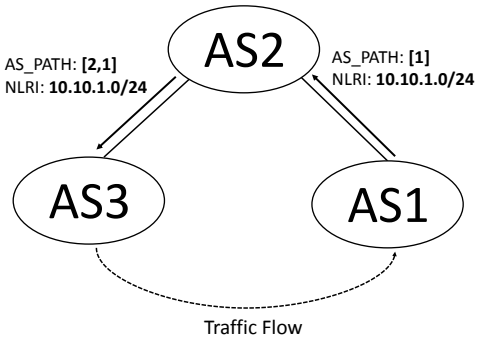


Figure 2.1: Example of Update attributes AS_PATH and NLRI in a simple topology

The BGP speakers synchronize with each other in order to keep a consistent state. The BGP information collected from any border router should reflect the routing behavior of the AS, depending on local policies. BGP speakers are categorized into two types according to the level of connection they establish with other speakers: BGP speakers that peer with others inside the same AS are called interior BGP (iBGP) speakers and if the speaker connects with peers in other ASes, those outside speakers are called external BGP (eBGP) peers [1]. Figure 2.2 depicts a system topology with three ASes and the relationship between iBGP and eBGP routers.

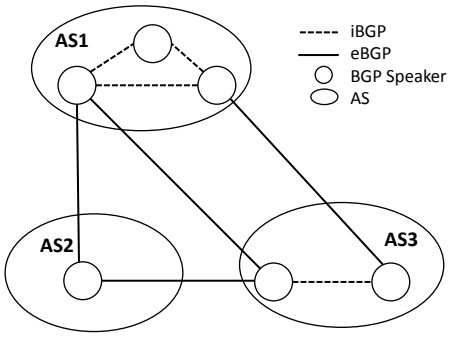


Figure 2.2: Example of BGP system topology [1]

Four types of messages may be exchanged in a BGP session:

- OPEN: Opens a session between BGP speakers.
- UPDATE: Advertises a new feasible route or withdraw an unfeasible route.

- NOTIFICATION: Notifies when an error is detected in the session, and when a peer is notified, the session is closed.

- KEEPALIVE: Periodically verifies if the connection between BGP speakers persists in the session.

The first message exchanged between the BGP speakers is the OPEN message. If no error as an authentication failure or a malformed attribute list is detected in the message, the session is initialized and important information about each speaker is learned. In the next step, an UPDATE message is sent by each speaker. The very first time this message is received, the speaker will download the entire routing table sent by its neighbors [13, 32]. This message may carry a withdrawal to a path that no longer exists or the announcement of a new best path [31]. If any error is detected in a BGP message or during the establishment of a session between BGP peers, a NOTIFICATION message is sent by the peer that detected the error, resulting in the end of the session. The KEEPALIVE messages are exchanged in a pre-defined interval (in seconds) to ensure that the session is alive.

Current BGP implementation is based on the concepts of Simple Path Vector Protocol (SPVP), which was defined by Pei et al. in 2002 [5]. SPVP is the path vector protocol in which each node must select and use only one of its available paths to reach each destination, and advertise only its selected and preferred routes to its neighbors. The latest path received from some neighbor replaces the previous path sent by the same neighbor, and becomes a candidate route for path selection. Another important aspect of the BGP protocol comprises the import and export policies, that specify which neighbors should receive routes during export, and from which neighbors the routes may be accepted during import [24]. Besides that, BGP has a timer that constrains the announcement of events, the *Mininum Routing Advertising Interval* (MRAI), which will be detailed further.

Besides the described representation, a model of topology and relationship between ASes is important to be considered in order to simulate and understand BGP dynamics. The term *BGP dynamics* refers to the aspects related to BGP behavior, e.g. how routing policies influence the announced routes, the routing information table growth and consistency, its convergence time, its stability, resiliency to failures, the path selection process, how BGP reacts to a route change, among others [33, 34]. A given research may choose to focus on only one or more aspects of BGP dynamics depending on the problem trying to be addressed. However, a simulation of BGP dynamics without relationship information may result in misleading inferences [35]. Relationship between ASes might be described as:

1- Transit relationship: One AS pays to another for Internet access.

2- Peer relationship: Two ASes exchange traffic without cost, for mutual benefit.

These aspects of relationships incur in more realistic experiments of Internet routing and might be implemented on most known network simulators that permit interdomain routing simulation.

An interesting illustration of BGP behavior is described by Li et al. [36] as a wave model. When an Update is triggered – due to a network change – reachability information is propagated through the topology from the source node, traveling along all the paths containing BGP speakers, similar to a wave propagated in a lake when something is dropped in it. The wave keeps being propagated through BGP speakers until it reaches its destination. If the new information is less preferred than the existing path information, then the wave dies out that BGP speaker, until a new wave arrives as a preferred path than the existing one.

2.1.2 BGP Router Model

In general, a BGP router or BGP speaker is a router that runs the BGP protocol software, besides other routing protocols. The BGP router maintains the reachability information exchanged between ASes in a table named *Routing Information Base* (RIB). The RIB is one of the most important elements of BGP protocol, organizing all information and considering not only routing but also the policy relations while storing and sending routes.

To better organize its model in accordance to its functions, a RIB is subdivided into three parts [37]:

- *Adj-RIBs-In*: This is the part of RIB responsible for storing the routing information received from peers through UPDATEs. For each neighbor of a BGP speaker, there exists an associated Adj-RIBs-In instance. When the BGP selection process is triggered, the deployed input policies filter the routes at Adj-RIBs-In that must be available to be selected and installed in Loc-RIB.

- *Loc_RIB*: The Loc_RIB contains the selected best paths chosen after the selection process takes place and input policies filter the routes received by Adj-RIBs-In. The routes in RIB are then installed in *Forwarding Information Base* (FIB), which is a table used by routers to forward the packets to the indicated outputs. Loc_RIB is also referred to as the main RIB.

- *Adj-RIBs-Out*: For each neighbor of a BGP speaker, there also exists an associated Adj-RIBs-Out instance. After the routes are installed in Loc_RIB, output policies are applied to select the routes that will be advertised to each peer. The Adj-RIBs-Out contains the information of the permitted routes after the output policies are filtered.

This model is an abstract concept of a BGP router. The real implementation depends on vendors and manufacturers; some implementations keep only one instance of RIB, for example, to save storage and memory.

The BGP Router model explained in this section, with its elements and the relation between them, is illustrated in Figure 2.3.

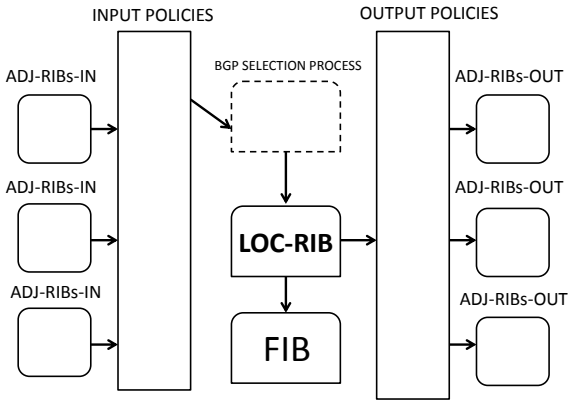


Figure 2.3: BGP Router model

2.1.3 Convergence Delay

We may say that when the main routing table (the *Loc_RIB*) of a BGP speaker remains without changes, the whole system is stable. When there is a change in its topology, due to a failure or a new node entrance, the main BGP speakers will advertise each other, exchanging updates, until the stability of the system is achieved again. This process is known as *convergence* [38].

There are four events, known as *T events* triggered by topological or policy changes [39, 40] that may lead to a convergence process:

Tup - A route that was unavailable is announced as available by some router. It may occur when a link is recovered from a previous failure and new routes to a destination are learned.

Tdown (faildown) - A previously available route becomes unavailable. A link failure incurs in a network partition, and the nodes which lose communication with some destination advertise withdrawals to its neighbors informing the unavailability of the routes learned from the disconnected destination.

Tshort - A new *AS_Path* replaces an existing one, when the new AS_Path is shorter since a longer AS_Path is considered less preferable. This happens when a node detects a recovered link, and learns a new and more preferred route.

Tlong (failover) - The current *AS_Path* is replaced by a worse (longer) one. This occurs when a node detects a failure in a link that it uses to reach a destination. This node will select a less preferred route and advertise the newly chosen route to its neighbors.

Pei et al. [39] presented a definition of converged state, which is reached when the convergence process finishes:

Definition 1*: Converged State* - A node *v* reaches a converged state if and only if its *Routing Information Base* (RIB) does not change until next triggering of the convergence event.

When a failure happens, BGP might withdraw affected routes and announce new routes repeatedly until the state becomes stable, generating a huge number of messages, while new paths are explored [31]. A failure always results in invalid routes, which are defined as routes that do not reach the destination [5]. The recovery time from failures may be dependent on BGP parameters, such as MRAI timer, topologies adopted, as well as the overhead of routers at the recovery process. When an used route fails, BGP withdraws the current used route and selects a backup path. The new route is then advertised to its neighbors. But this new route may also be invalid, and will be withdrawn only when the neighbor that advertised it sends a withdraw message. This behavior may lead BGP to use a number of backup routes until a valid one is selected, in a process known as *convergence delay* or *recovery time* [41]. A definition of network convergence delay is done by Pei et al. [39], as follows:

Definition 2*: Network Convergence Delay* - a network convergence delay begins when a *T* event happens, and finishes when convergence state is reached again by the network.

The convergence delay can be considered as the period of routing instability in the network, or the amount of time in which a network change is propagated over Internet topology [35], and therefore from the perspective of a routing protocol like BGP, the goal should be to minimize this delay. The *Routing instability* has a number of origins besides the update events mentioned, such as route configuration errors, transient physical and data link problems, and software bugs [17]. This instability is also called *route flaps* and contributes to poor end-to-end network performance and infrastructure efficiency. Labovitz et al. [18] showed that the BGP convergence delay for isolated route withdrawals can be greater than 3 minutes in 30% of the cases and could be as high as 15 minutes. They also found that packet loss rate can increase by 30 times and packet delay by four times during recovery.

Using the concept of convergence delay we can define *routing delay* as the time taken by a router *v*, to receive a routing update from a neighboring router *u*, including the delays generated by queuing, transmission and routing information processing, and to update its RIB. The upper bound of this delay is represented by *d*.

Improving convergence may help in faster re-routing of the packets, reducing packet loss and delay in cases of failure [42].

2.1.4 MRAI Timer

The huge number of update messages that arrive at BGP routers to be processed generates a high overhead. While some messages are still on a queue waiting to be processed by the router, new messages may be generated by other routers. This indicates that it is possible that when a message is processed, the announced route is no longer valid and false information is propagated, until the whole system becomes stable after a considerable amount of time.

BGP presents a timer that limits the number of messages propagated by BGP speakers, called Minimum Route Advertisement Interval (MRAI) [24]. This timer is very important in BGP convergence process.

Before a BGP speaker starts sending updates to any peer/update group, it checks if the MRAI is not expired for that peer. A BGP speaker starts this timer on per-peer basis every time it completes sending the full batch of updates to the peer. If the subsequent batch is prepared to be sent and the timer is still running, the update will be delayed until the timer expires. This is a damping mechanism to prevent unstable peers from flooding the network with updates. After several experiments a value of 30 seconds was agreed for the default [26] and recommended by the BGP RFCs [13].

2.1.5 Software-Defined Networking

The SDN approach has the potential to bring innovation to the Internet because it enables architecture diversity and network modularity. It divides a network in Control and Data planes, each one with a well defined function. With SDN a network is managed through a logically centralized controller that rules a group of switches using a standard interface. These controlled switches may be from different vendors [43].

The Data Plane comprises the network physical elements responsible for traffic forwarding like switches, hubs, and routers. The Control Plane is composed by the Network Operating System (NOS), the NOS Interface, and the applications that run on top of it. NOS is responsible for managing the network resources and enforcing the actions decided by the applications onto the Data Plane elements. The communication between the Data and Control planes is done by the standard interface which must be deployed on network equipment. The most well known and used interface for the Data and Control planes' communication is the OpenFlow (OF) protocol [27] which provides APIs to install packet-forwarding rules, store and query traffic statistics, and learn topology changes [43]. With SDN, developers

create applications without worring about low-level details. The API provided enables fast prototyping, implementation, and deployment of new applications for managing the networks in a fast and less prone-to-error manner.

Figure 2.4 depicts the SDN basic architecture. The physical network equipment, including hosts and OF switches, resides on the Data Plane. The Control Plane is comprised by the Controller (NOS) and a set of applications that manage the network.

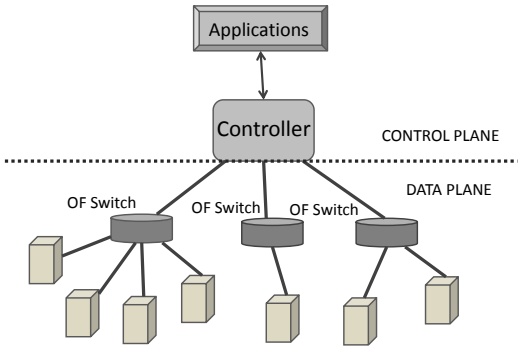


Figure 2.4: SDN architecture

2.1.6 Machine Learning Models

Machine learning can be defined as computational models that improve its performance or its accurate predictions based on learning/experience. The term *learning* is used to refer to a knowledge obtained from past experiences available to the model, typically in form of collected data. Therefore, the machine learning algorithm can search for regularities in the data, using the found patterns to take actions such as classifying the data according to a certain number of clusters [44]. The success of the predictions of a machine learning algorithm depends on the quality and the size of the collected data.

There is also a number of machine learning applications to address issues related to BGP: identification of anomalies based on deviation from the normal BGP-update dynamics for a given destination prefix [45]; analysis of BGP convergence properties by using a probabilistic model checker [46]; the use of Support Vector Machines (SVMs) and Hidden markov Models (HMMs) to detect and classify BGP anomalies [47]; training of a model to predict the edge types in a given AS graph, which is very useful to business relationship inference between ASes [48]; the use of neural networks to prevent poison message failure propagation and avoid some large scale failures and instabilities [49]; the detection of anomalous events resulting from prefixes with unintended limited visibility by separating them from the prefixes

with intended limited visibility that come from legitimate routing operations [50]; anomaly detection on global routing structure using a framework based on data mining algorithms [51]; the detection of BGP instability from deterministic, recurrence and non-linear properties of speakers by applying Recurrence Quantification Analysis (RQA) algorithm [52]; the leverage of data mining techniques to train an Internet Routing Forensics framework to discover patterns in BGP events and detect the occurrence of abnormalities [53].

A machine learning model can be built by using different types of learning: in *supervised learning* a set of labeled entries is used for training and makes predictions for unseen entries, being the most common scenario used for classification, regression, and ranking; in *unsupervised learning* a set of unlabeled entries is used to make prediction on unseen entries, being difficult to evaluate its performance due to the absence of labels; in *semi-supervised learning* both labeled and unlabeled are presented to the learning algorithm; in *reinforcement learning*, the algorithm actively interacts with the environment, receiving rewards for its actions, and learning from these rewards.

Neural networks

Artificial Neural networks, or simply *neural networks*, are a powerful machine learning method inspired in the biological brain, i.e., it is designed to model the behavior of the brain neurons when performing a task [54].

Neural networks bring several benefits to the problems they are applied to. Some of them are shared with other machine learning methods. Among those benefits are:

- A strong capacity of representing hypothesis, being able to capture underlying regularities inside datasets and therefore building internal representations from the data [55].

- Through its parallel structure, neural networks are capable of strong generalization, learning from inputs not present in the training dataset. This property makes it possible to find good solutions for complex and intractable problems.

- They have potential to be fault tolerant due to its implementation design, i.e., its performance degrades very slowly when facing failure events.

- Neural networks deal very well with knowledge representation due to its structure, where every neuron is potentially impacted by the activation of all other neurons of the network. Therefore, contextual information can be naturally represented in a neural network [54].

Long Short-Term Memory (LSTM)

By observing human thinking we can conclude that it does not emerges from scratch, but happens in a context of past experiences and observations. Traditional neural networks are not suited for problems that demand a reasoning on previous events in order to accurately predict next events. Recurrent Neural Networks (RNNs) are designed to address this issue. They augment the traditional neural net capability by including a chain architecture that stores and passes the output of one step of the network to the next. RNNs are often used to model problems involving sequential data, where the sequence of events is required to predict the desired output. However, vanilla RNNs suffer from the *vanish gradient* problem, i.e., for long sequences of training the neural network gradients becomes more close to zero and the learning process becomes harder at each training iteration, making the RNN forgets what it has learned from previous long sequences [56]. To address this problem, the *Long Short-Term memory* (LSTM) was proposed in 1997, being able to handle long sequences of data [57], being applied to solve speech recognition [58], stock price forecasting [59], and other problems where remembering past observations is important to an accurate prediction [60].

A LSTM network has a chain structure of repeating modules similar to a RNN, however with a different architecture inside of them. In LSTM, each repeating module has four layers (RNNs have only one). Figure 2.5 shows a representation of essential elements in a LSTM network. The bigger rectangle in the figure represents one repeating module. Inside this module we can observe four smaller rectangle. Those rectangle represent the four layers of a LSTM; the output of each layer has an activation function (sigmoid or tanh). The input Xt' represents an input vector of our collected features at time t'. The output from each layer is another vector and the circles with *X* and + are used for operations between those vectors. Two of the most important concepts in LSTM models are the cell states and the gates. The cell states (represented in figure 2.5 by letter *C*) forwards or throw away information according to the decision made by the *forget gate (ft)*. The *input gate (it)* decides which information will be stored in the cell state. The third gate is the *output gate (ot)*, which defines what parts of the cell state will be sent as output of the module.

2.2 Issues on Interdomain Routing Convergence

The Internet has some crucial issues [38], which brings the question of architecture redesign into consideration [61]. Despite all researchs and publications on interdomain field, there still exists some difficult and unanswered questions [35] like: How do route changes and other factors like policies, topology properties, and iBGP configurations affect end-to-end

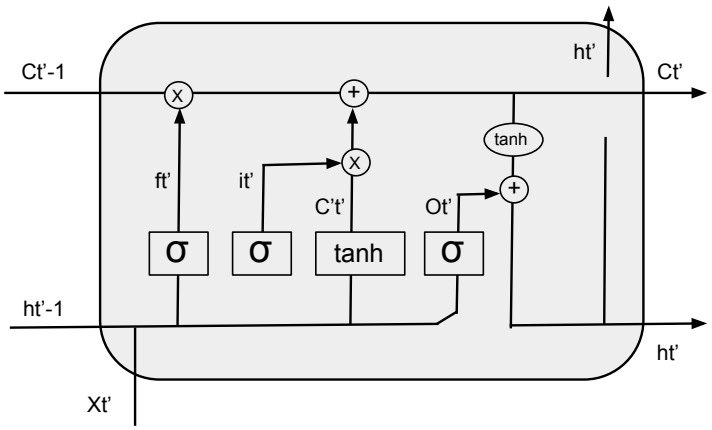


Figure 2.5: Anatomy of a LSTM network [2]

path performance on the Internet [62]? BGP presents many problems on what concerns to policy enforcement capabilities, security, scalability, and complexity [63].

In this section, the most evident issues detailed in literature that impact interdomain routing convergence are described.

2.2.1 Dynamic behavior poorly understood

Although BGP is the “glue" of the Internet and its specification is very well known, its dynamic behavior is rather poorly understood. According to John et al. [64], the required understanding of the protocol is a hard task because BGP is complex and unpredictable and this is due to the fact that the routers operate in a distributed state, which may lead to inconsistencies. The authors believe that those factors are the root cause for Internet instabilities. A system with unpredictable behavior is vulnerable to misconfiguration, as it becomes difficult to differentiate between good and bad behaviors. Besides that, protocol unpredictability makes the network administrators and operators resistant to accept changes in the current protocols, and intended deployments in their networks.

BGP’s fully distributed and asynchronous model makes a deterministic prediction of its precise behavior very difficult, since it depends on the sequence the BGP advertisements arrive from different speakers [65]. This aspect was observed by Mahajan et al. [66] in 2002, which presented a quantitative study on the frequency of Internet disconnectivities caused by BGP misconfiguration. The analysis of data collected from 23 monitoring points on the Internet during three weeks revealed that between 200 and 1200 prefixes (equivalent to 0.2% to 1% of the global RIB size) suffer from misconfigurations daily. They are focused on two types of misconfigurations: origin misconfiguration, which is accidental insertion of routes in BGP RIBs; and export misconfiguration, which is accidental announcement of routes

that should have been blocked by filters. It is evident that BGP behavior should be simpler and easier to manage, which is a strong requirement to avoid many of inconsistencies that negatively impact on interdomain convergence.

2.2.2 Service Unavailability due to convergence delays

The high convergence delay is one of the reasons why the Internet is not always available. Several applications require more and more network resources in an interdomain structure which is not yet prepared to handle increasingly more exigent demands [67]. When the topology changes due to a link failure, all routers need to be notified, and new information is updated in its *Forwarding Information Bases* (FIBs), leading to a slow convergence process [61]. Labovitz et al. [18] found in 2000 that BGP may take several minutes to converge when changes affecting several paths take place. During the period of convergence forwarding loops may occur. With such long convergence delay, many interactive applications are disrupted with packet losses, stuck in loops or delivered out of order [20].

In other words, maintaining a consistent routing table is critical for the proper functioning of the Internet. Also, the high convergence delay it often presents incurs in service unavailability with packet losses and poor quality for applications [19]. The inconsistencies caused by the effects of BGP dynamics directly impact on the traffic inside an AS [68]. Many different factors may render the interdomain service unavailable, such as TCP disconnection, security issues, failure events and so on. Also, the convergence of a system depends on the availability of the interdomain routing service provided by BGP. Service unavailability may also stems from difficulties in configuration or understanding on BGP dynamics, as a consequence of the issue previously described. It turns out to be a NP-complete problem to determine if an AS or group of ASes will converge, as detailed by Cittadini et al. [69].

2.2.3 Conflicts caused by individual policies

The Internet was initially created and used as in a research environment. When it arrived in the commercial world, new challenges appeared. The interdomain routing became necessary, but unlike other existing routing protocols it did not have a clear and globally defined objective, with each domain presenting individual goals that in many cases conflicted with the commercial interests of other ASes [70].

These individual goals are defined through flexible policies which the BGP enables the administrators to express. The main drawback of this flexibility is that it is not possible to guarantee convergence to a stable state, like in most routing protocols. Even with simple policy configurations BGP may cause permanent oscillations in routing [69].

The types of relationship between ASes also have impacts on the types of policies applied. Observing only the group of policies of a restricted number of ASes is not enough to tackle most issues caused by the policies with conflicting interests. Chang et al. [71] detected that at least 45% of routing changes occur during transit peering, which means that the analysis based solely on the end-to-end relationship is not enough to detect and guarantee routing stability and fast convergence.

2.2.4 Transient failures and oscillations

Failures may be caused by faildown or failover events. When the failures disconnect the destination from network, we have a *faildown event*. When nodes still having a physical connection that would incur in a best path, but are forced to use alternative paths to reach destination, we have a *failover event* [38]. A faildown event may induce into BGP a latency in the range of 3 to 15 minutes [40].

Failover events may lead to two types of failures: Transient routing and transient forwarding failures. A *transient routing failure* occurs when a router looses the routes to the destination while still having a physical reachable path, after a failover, with its *loc_rib* becoming empty [72]. A *transient forwarding failure* – also called blackhole problem – occurs when a forwarding path stops at a router during the transient routing failure. From this latter type of failure, when the forwarding path creates a loop, we may say that a transient forwarding loop occurred. Some of the existing works, such as EPIC [73] and Ghost Flushing [40], accelerate BGP convergence, eliminating the transient forwarding loops, but exaggerate transient routing failures [38].

John et al. [64] observed that routing protocols used nowadays give priority to responsiveness over consistency. An update received by a router must be processed as fast as possible because the network should quickly react to changes. The consistency ensures that packets will follow the route specified in the path attributes to be delivered. With demands of responsiveness, comes black holes and routing loops. The Internet tries to enforce responsiveness, which is a living property, over consistency, which is a safety property, but fails to achieve effectiveness on both the aspects.

To address or to minimize those type of failures, Holterbach et al. [74] proposed SWIFT, a framework that predicts the extent of a remote failure out of some BGP messages to gain significant speed at the price of some accuracy. It enables existing routers to quickly restore connectivity in cases of outages related to BGP. Another solution is LIFEGUARD, a system proposed by Katz-Basset et al. [75] that seeks the automatic failure localization and mitigation for short-term outages. LIFEGUARD techniques perform reroutes around failures with low impact to existing routes.

2.2.5 Ghost information

The false information that keeps circulating within the network causing complexities is referred to as Ghost Information. Afek et al. [40] claim that ghost information affects convergence because it remains in the network for some time before disappearing completely, thereby inducing a delay in convergence. The problem here is that one failure in a node, as in a faildown event, propagates recursively through the network, for a long period of time. This is a part of the information of paths to a destination that is disconnected from the network.

One of the effects of the delayed convergence is that during the convergence process some ghost routes are installed in RIBs, so the routers rely on wrong information to generate more wrong information, propagating this in the network until the valid routes are founded. The MRAI timer also delays the convergence while preventing the routes to be propagated before the predefined time expires. Ghost-flushing and EPIC processes may accelerate convergence triggered by faildown events from *O(D)* to *O(d)*, where *D* represents the network diameter, and *d* represents the delay generated by queuing, transmission, processing of the BGP messages, and RIB update. However, they perform poorly with failover events, because they speed up the withdrawal of routes but do not perform anything for the alternative routes.

2.2.6 Increase of table size and path exploration

The growth in number of ASes connected to the Internet, with more connections per AS, and the increased number of different applications – especially those with real-time requirements – impose a direct impact on interdomain routing convergence. Most of the proposed solutions to address the growing need for traffic engineering lacks the ability to convey specific and useful information and are based on periodic message flooding technique. This condition generates more traffic in the network, increasing the load and incurring in overhead at routers processing the messages [21, 22].

Some years ago the Route Flap Damping (RFD) and MRAI were considered good solutions for the convergence delay problem, but with the increase of demand and rise of many real-time applications on Internet such as Skype, network bank and video conferences, they became less attractive [76]. It was demonstrated by Kushman et al. [20], which experimented that BGP slow convergence may have profound impact on VOIP service. In their experiments, they stated that as many as 50% of dropped VOIP calls are caused by BGP updates circulating in the process of network convergence. These are the most critical issues and account for almost 90% of the dropped calls. This is a very serious problem since the industry expects that VOIP will replace most of the current land-line telephone connections. Contradicting the belief that congestion was the only significant cause of VOIP issues, almost half of the

performance issues are influenced by the inability of BGP to quickly re-converge after some event. The best-path selection time is proportional to the table size as well as time required for update batching, and the size of these tables continue growing with the increase of new ASes and connections on the Internet [40].

Zhao et al. [3] studied the convergence delay of some networks in Latin America region by taking some snapshots of the BGP routing tables of monitored routers. They noticed that more close the networks are to the larger Internet service providers, the faster is the convergence time. From the observation of a single router, they showed an example of delayed convergence. Figure 2.6 depicts a registration from AS 4777, a prefix registered on the Latin American and Caribbean Internet Addresses Registry (LACNIC) [1] on January 25, 2003. This example shows that AS 4777 switches between several paths trying to reach AS 10715, until discovering some minutes later that the destination was unreachable and sends a withdrawal to its neighbors, thereby taking a long time to converge. A route to AS 10715 was announced again later, indicating the occurrence of a failover event. With experiments, Zhao et al. verified that networks connected to small providers take longer to converge than those connected to large providers.

TIME	AS_PATH announced by AS 4777
03:07:45	4777 2516 3561 1916 10715
05:32:32	4777 2516 1239 3561 1916 10715
05:34:56	4777 2497 701 4230 8167 10715
05:36:20	4777 2497 2914 701 4230 8167 10715
05:36:47	4777 2497 1 701 4230 8167 10715
05:37:15	4777 2516 209 701 4230 8167 10715
05:37:42	Withdrawal
06:03:45	4777 2516 3561 1916 10715

Figure 2.6: Delayed convergence example from AS 4777 [3]

Besides the increase in demand of network applications, the Internet architecture has changed from a hierarchical topology where regional providers connect to large National Backbones via Network Access Points (NAPs) to a new emerging Internet architecture that composes a much denser connection mesh. With the growth in number of nodes and connections, and also due to the impact of traffic engineering and commercial policy, new changes on Internet traffic take place, contributing to the increase of path exploration at

[1] LACNIC [77] is one of the organizations responsible for assigning and managing IP and AS Number addressing.

routers [78]. Path exploration also delays the whole system convergence, which being one of those factors that many researchers are trying to address to achieve a better convergence time.

Chapter 3

Efforts on Reducing Convergence Delay

Reduction of convergence delay and message complexity of the Internet brings many benefits, for example, it provides QoS and high availability to Internet services [10]. This chapter presents the main efforts from the literature on the reduction of convergence delay observed in the Internet during last years. The content of this chapter is part of a survey published in the IEEE Communications Surveys & Tutorials [23]. In the end of each section we provide a summary, describing the advantages, disadvantages, unique features, and our overall insights. In the next section, we first describe some researches that do not propose any new mechanism but contribute to the implementation of most methods and techniques that are discussed in the later subsections. BGP churn investigations, convergence behavior monitoring and analysis, and policy impact are some of the research investigations that lead to better understanding on BGP dynamics, therefore improving the efforts to issue the interdomain convergence problem. They are presented in Tables 3.1 and 3.2.

Based on our research, we classified the methods and techniques in five different approaches. These approaches are: speeding up, limiting path exploration, efficient policy configuration, multipath and multi-path forwarding, and centralized control. They are summarized in table 3.3. The subsections *limiting path exploration* (VI.C) and *multipath and multi-path forwarding* (VI.E) are subdivided into two parts. For each method or mechanism presented in this section we provide a systematic assessment of them in Tables 3.4–3.12 according to the following criteria:

- *Incremental deployment*: the mechanism modifies the BGP implementation and does not require that all ASes with standard BGP need to be replaced so that it achieve it's goal. The new mechanism may be gradually deployed in a partial number of existing ASes and coexists with ASes with "legacy" standard BGP implementation, being able to enhance convergence time of the whole system.

- *Computational overhead*: the proposed mechanism incurs in extra overhead to routers that may increase convergence delay.
- *Changes BGP behavior*: the mechanism changes standard BGP behavior in some aspect (e.g. its decision process, the transmission of only its best route to a destination, among others), which could lead to further inconsistencies if not deployed carefully.
- *Constraints to BGP expressiveness*: the suggested mechanism requires constraints to BGP policy expressiveness to issue routing instabilities and improve routing convergence.
- *Coordination between ASes*: the proposed mechanism requires a major level of coordination between ASes than current BGP does.

3.1 Descriptive and Analytical Approaches

In this section, we describe the efforts in understanding some important factors of the interdomain routing that impact on convergence. These efforts are of descriptive, monitoring or analytical nature, and although they do not introduce new mechanisms, they contribute to the development and advances in interdomain convergence research. In the end of this subsection, we present some papers on BGP churn investigation.

Zhang et al. [79] focused their work on the evaluation of the packet delivery performance of the BGP protocol, in cases of a failure followed by a recovery, i.e, a *Tdown* followed by a *Tup*. Their results showed that, in general, reducing the convergence time may indeed improve packet delivery. As a long MRAI timer may add considerable delay to the convergence of the system, re-examines its values might be an important step to improve packet delivery. However, it is not an easy task since a shorter *Tdown* convergence time may incur in a reduced packet delivery. The authors also observed that connectedness is different from reachability: *connectedness* occurs when a physical connection exists between a source and a destination at time t, while *reachability* at time t is given by the forwarding state of all the paths, hop by hop. In a previous work [80], the authors used a different approach, measuring the destination reachability by counting the percentage of packets sent by a source, which were received by a destination.

Shahoo et al. [81],[82] noticed the impact that the processing overhead of routers has on convergence delay and the importance of the MRAI timer value. For example, they consider that node A sends updates to node B at time t and assume that MRAI is high enough so that all update that arrives at B have been already processed at time t + MRAI. Changing MRAI will have the following consequences:

1- If MRAI increases, the nodes will have to wait longer before sending new updates, which will increase the convergence delay. In this phase, the number of updates sent remains constant, while the delay increases linearly.

2- If MRAI value decreases, the updates are generated at a faster rate, but the processing load of the routers increases. So, a node could send out an update to a neighbor before it has processed all queued update messages. If one of the remaining updates at the queue changes the routes it has just advertised, another update needs to be sent. It will increase the workload on nodes, which will in its turn increase the convergence delay. Fabrikant et al. [24] observed that although MRAI has a default value of 30 seconds, router vendors are lowering or eliminating the timer, hoping that with a small value, the convergence time will be reduced. The authors show that decrementing MRAI does not improve convergence time, and the network behavior in convergence process might actually get even worse. So, the deployment of MRAI changes may be done carefully. To evaluate BGP convergence behavior with MRAI some metrics are defined as:

- Disparity: The ratio Γ between the highest and lowest MRAI values in use:

  $\Gamma = t^*/t_*$

- Diversity: The number υ of different MRAI settings in use:

  $\upsilon = |U_i\{t_i\}|$

  Where t^* denotes the slowest MRAI in the system:

  $t^* = \max_i t_i$

  And where t_* denotes the fastest MRAI in the system:

  $t_* = \min_i t_i$

- S_i is the time that a router takes to process a route, since when a new route is received by router i, the router preferences are changed and the new route gets ready to be exported according to export policies. Although it has a non-zero value, S_i is much lower than any t_i value, since modern routers may process updates far faster than usual with MRAI values of 5 seconds for iBGP and 30 seconds for eBGP sessions, which are frequently used nowadays.

The MRAI *disparity* and *diversity* definitions are useful metrics to understand the network convergence. When we have a system where the fastest MRAIs are much faster than the slowest ones, we may detect a high disparity Γ. They also showed that the number of different MRAIs in use, the υ, may help increase exponentially the number of updates exchanged during convergence process.

Fabrikant et al. also proposed a corollary where they determined that a system may converge in $T' = O(nt^*)$ time. Since $\Gamma = t^*/t_*$, it is also true that $T' = O(n\Gamma t_*)$ time. The authors showed that homogeneous values of MRAI have linear convergence time, while several different MRAI timers may exponentially increase the number of update messages and the convergence delay. They suggested that if each node could be aware of the timer settings of the other nodes on the path, in a dynamic scheme, it could avoid updates quicker than those nodes, and therefore reduce the convergence delay in a global instance.

Griffin et al. [26] further observed that every topology has a certain MRAI that results in favorable convergence time. The authors evaluated the influence of MRAI, and other techniques, namely *Sender Side Loop Detection* (SSLD) and *Withdrawal Rate Limiting* (WRATE), on the convergence time. In their experiments, they considered two types of situations for each topology: The first is the *UP* scenario, which happens when a node announces a single destination and the system is allowed to converge. The second is the *DOWN* scenario, which takes place when a withdrawal is announced and the system is allowed to converge. They made some observations from the experiments, as follows:

For each kind of topology and situation there exists a value for MRAI for which the average number of updates necessary for system convergence is stable. For some networks, such as simple chain of ASes with one router topology, the MRAI may be 0, and the number of updates required for convergence becomes independent of MRAI. However, in more complex networks, with several alternate paths, MRAI is necessary to reduce the number of updates. Another observation is that for each topology and situation there exists an optimal value for MRAI, for which the convergence time is the minimum possible. Increasing this value, increases the convergence time linearly.

Griffin et al. noticed from the two observations that the MRAI optimal values for the number of updates and for the convergence time are very close. They also observed that: optimal value of MRAI increases when workload at routers also increases; the WRATE timer, which constrains the sending of withdrawals, may impact positively or negatively on convergence time, depending on the topology and situation (UP or DOWN); and SSLD never increases convergence time, decreasing it at a low rate, in some cases. The authors concluded from the experiments that SSLD and WRATE values are negligible when MRAI is used, because its value is more determinant on convergence time, but MRAI optimal value changes from network to network, making it impossible to determine a general value for it in practice. However, they considered the possibility of customized values for MRAI on different locations of the Internet, according to the number of alternate paths it present, and stated that the default recommended value of 30 seconds used globally is somewhat arbitrary and heavily impacts on interdomain convergence. The IETF draft about MRAI

values [83] proposes a revision to MRAI default, stating that the assigned value in RFC 4271 is actually deprecated, and that the implementations must let network operators choose this value according to their networks.

Qiu et al. [84] made experiments to estimate an optimal MRAI for the Internet, based on the study made by Griffin et al. [26] that for every topology exists an optimal MRAI value, called *Mo*, which is the best trade-off between the stability and convergence time. Their analysis also revealed the existence of a value for MRAI – the safe MRAI – which adequately reduces the number of propagated updates, by reducing the number of candidate paths, without causing inconsistencies. There is a minimum safe MRAI that might be inferred from the network topology, which enables the estimation of *Mo* with precision and becomes the upper bound of it. The experiments were made on the *Planet Lab* [85] platform, that provides a real Internet environment for experiments. The authors concluded from the experiments that the current MRAI value used on the Internet is about 5–10 times the optimal value, i.e., the optimal value is far less than the default 30 seconds.

Li et al. [34] revisited Labovitz et al. [17] research on BGP dynamics, published more than two decades ago, analyzing how it has changed over the years. Their analysis was made from BGP Updates collected in a six-month period. From the observed BGP dynamic behavior, Li et al. divided the BGP events into three groups: forwarding dynamics, policy fluctuations, and pathological duplicates. The *forwarding dynamics* comprises of topological changes and impact the forwarding paths; *policy fluctuations* are due to policy changes, without affecting forwarding paths; *pathological duplicates* occur when the redundant BGP updates are present. The authors observed that BGP is healthier than when the study of Labovitz et al. was published, with reduced rate policy fluctuations and pathological updates, although it still occurs. However, as in Labovitz experiments, they also observed predominant inter-arrival times around 30 seconds, affected by the MRAI timer. It shows that in a period of almost a decade MRAI timer continued having a considerable impact on BGP dynamics.

Mao et al. [86] worked with simple topologies and showed the case where a single announcement or withdrawal might incur in route penalties to accumulate beyond suppression threshold. They experimented with different parameters settings, such as a set of MRAI values, the presence or absence of SSLD, WRATE, and policies, and the implementation or not of RFD in the experiments.

Among other conclusions, Mao et al. observed that in some topologies, as in clique, disabling damping incurs in better convergence than while keeping it enabled, because stable routes might suffer penalties and be suppressed. The RFD mechanism actually increases the convergence time of stable routes, sometimes by more than one hour. RFD strongly penalizes well-connected ASes because with a high number of connections the number of exchanged

messages increases. Reachability issues and packet loss were detected when no physical link failures or congestion occured in the network. It means that valid routes might be wrongly withdrawn due to the RFD mechanism [87].

Griffin et al. [88] presented a study on the stable paths. They explain that it is possible to guarantee that BGP will always converge by using three complementary approaches. The first is by using operational guidelines, which is a collection of rules that all ASes must follow to ensure policy safety and correctness. The second approach is based on static analysis, making programs analyze routing policies and detect if any policy conflict or other inconsistency exists, which may cause protocol divergence. The third approach is a dynamic detection, which prevents and suppresses routing oscillation as soon as it is detected. RFD uses this third approach, but the authors know its issues and do not recommend its use. To detect potential oscillation on network, Cittadini et al. [89] proposed a heuristic algorithm that performs static detection in an AS.

Wang et al. [90] presented a model which formally characterizes transient routing failures. The goal of this approach is to provide a better understanding of transient routing failures by identifying the sufficient conditions for this type of failure. The authors created the concept of a path availability graph, which helps to discover when a router is affected by a routing failure.

Caesar et al. [33] observed that to achieve success in understanding BGP dynamic behavior it is necessary to discover what causes a route change and what does this change originate. The authors observed route updates from multiple vantage points to discover the location where a convergence event takes place. From the observation, a BGP health inferencing system was developed, classifying the updates in groups according to the correlation between them, and determining the cause and location of a given observed convergence event. An interesting observation from analysis of the experiments with data from Routeviews and RIPE [91] for some months divulged that the convergence process happens quickly when it occurs at the Internet core, while the convergence delay is usually considerably higher when the convergence event takes place at the edge networks.

Rexford et al. [92] observed that there is a small group of popular prefixes responsible for most part of Internet traffic that has considerably stable routes. They also observed that most part of BGP routing instability occurs with a small group of unpopular prefixes. This is important for the operators to observe that routes corresponding to popular prefixes probably present stable paths, and to avoid make changes to it, which may introduce inconsistencies and increase convergence delay.

In a study on data collected during three years, Oliveira et al. [93] showed that the majority of the prefixes are highly active for only one or few days, with a small number of

them being stable for a longer time. The performance of CAGG in reducing convergence delay is better than of RFD, according to authors' experiments.

ASes currently enable policies being expressed by a language such as RPSL or by ranking and filters. Rankings establish how each router inside an AS should order different routes to a destination, and filters which routes should be advertised or hidden from each neighbor. Feamster et al. [94, 95] studied BGP stability under policies expressed by ranking and filters, showing that commonly used rankings do not ensure routing stability, and proved that the routing system is guaranteed to converge to a stable state with the requirement that routes are ranked by ASes based on AS-path lengths.

Feldmann et al. [96] proposed a methodology for studying the dependency of BGP pass-through times as factors related to routers running BGP process. Among the factors that may affect pass-through times are the number of peers, the routing table size, the BGP update rate, and the CPU load, besides other additional parameters, as input/output queue length and BGP policy settings. In general, there are three approaches for measuring convergence delay in the Internet: passively observing regular traffic, actively injecting traffic at end points, and passively measuring actively injected traffic. Feldmann et al. results show that large BGP convergence times may not just stem from protocol related parameters, such as MRAI and RFD, but also the delay imposed by the router, the pass-through-delay, may play a big role as well.

Wang et al. [97] proved by experimental results that improving MRAI not only guarantee network stability and robustness but also accelerate the routing convergence time. They divided the problem of convergence into two categories: One is the routing convergence due to the strategy conflict between ASes; this problem generally does not involve AS internal structure. Another category is routing oscillation because of some faultiness of the BGP system, which is related to the AS internal structure. Wang et al. showed by experiments in SSFNet simulator that there is a relationship between convergence time and MRAI configuration, and that exists an optimal time interval to make the convergence time shortest for each simple simulated network topology.

Wenhua et al. [98] analyze the relationship between topology parameters and route convergence delay. Authors assume that the four basic topologies (focus, clique, tree, and ring) are the most widely used network inter-connection methods, and after some steps derived the upper bound convergence delay for any network topology. The two main parameters of the network topology are the network degree and the network diameter.

With average node degree increasing, BGP convergence delay in Down Phase goes up but the delay in Up Phase does not strictly change. With network diameter increasing, BGP convergence delay in Up Phase goes up but delay in Down Phase does not have distinct

changes. Choosing topology structure with trade-off network parameters could lead to a small satisfying convergence delay in both Up and Down phases, once the two parameters may not get the minimum value at the same time.

3.1.1 BGP churn investigation

We refer to *BGP churn* as the rate of routing updates that BGP routers must process.

Schrieck et al. [99] identified some factors that may cause BGP churn issues. First, some interdomain links are unstable and frequently fail, transmitting this information to neighboring ASes through withdrawals. Second, as a path vector protocol, BGP suffers from path exploration when routes are unavailable. Besides the two detailed factors, MRAI and RFD may also be source to BGP churn and further delay convergence.

Elmokashfi et al. [100] presented a study on the evolution of BGP churn in four networks at the Internet backbone during a period of seven years and eight months via the RouteViews project. By taking an exploratory data analysis, the authors discovered that duplicate update announcements are the major part of detected churn. On its turn, the main cause of long-term periods of churn are misconfigurations of other rare events at the monitored ASes.

In another paper, Elmokashfi et al. [101] observed that the churn of BGP updates are caused by an interplay of three factors:

- The routing protocol, which has the BGP mechanisms such as RFD and MRAI, and routing policies, etc.
- Events like link failures, BGP updates, traffic engineering operations, etc.
- The characteristics of the Internet topology.

The authors focused on the study of influence of the third factor, the topology, on the existence of BGP churn. They created a topology generator to examine what determines churn at different locations of the generated topologies, and analyzed how the number of updates increases with the size of each topology. Their topology generator also considered business relationship aspects.

Elmokashfi et al. [102] also developed a model to try to predict how BGP churn will evolve in the future, considering interdomain topology, routing policies, and traffic characteristics that frequently changes. The developed model showed that the number of updates normalized by the size of the topology is constant, and qualitatively similar to IPv4 and IPv6.

Huston [103] presented a report on BGP churn in the period from 2008 to 2014. He concluded from his observations that the number of routing updates has remaining stable

for some years, despite a continued growth in the number of prefixes being announced in the routing table, as more ASes are connected to the Internet. This is explained by the fact that as the Internet continues to grow, the pattern of new connected ASes is repeated through the network topology. It means that the Internet is growing in density, rather than in size, resulting in relatively constant AS path length.

One effort to address BGP churn is the CAGG (Churn Aggregation) [76], which tries to reduce path exploration by aggregating multiple AS_PATHs in one route to be announced, without harming convergence. To aggregate those paths, CAGG finds the Path Locality of the small set of AS_PATHs explored by a highly active prefix and normalize the transient paths, reducing the total number of exchanged updates.

Tables 3.1 and 3.2 present a summary of the descriptive and analytical approaches detailed in this section and a short description of them.

Solution	**Authors**	**Short Description**
MRAI Reconsideration	Zhang et al. [79]	Re-examines MRAI value by experimenting BGP convergence with a failure followed by a recovery.
MRAI Trade-off	Shahoo et al. [81, 82]	Analyzes the impact of change of MRAI value by observing the trade-off between holding updates and reduced load at routers.
MRAI Impact analysis	Fabrikant et al. [24]	Proposes some metrics to evaluate the impact different MRAIs have on BGP convergence behavior and recommend homogeneous values.
Revisions to MRAI	IETF draft [83]	Proposes a revision to MRAI default value, stating that its value is deprecated.
Optimal MRAI	Griffin et al. [26]	Observes that depending on topology there exists a value for MRAI for which the total number of exchanged messages become stable and convergence time is minimum.
Safe MRAI	Qiu et al. [84]	Experiments the existence of a safe MRAI, which reduces the number of propagated updates by reducing the number of candidate paths, without incurring on inconsistencies.
BGP Routing Dynamics Revisited	Li et al. [34]	Analyzes how BGP dynamics has changed since a decade, and states the importance of MRAI in BGP dynamics.
RFD Exacerbation	Mao et al. [86]	Shows that RFD might wrongly penalize valid routes and makes them to be withdrawn .
Stable paths	Griffin et al. [88]	Presents three complementary approaches that ensures BGP convergence.
Static Detection	Cittadini et al. [89]	Performs static detection on ASes by using a heuristic algorithm.
Failure Characterization	Wang et al. [72]	Presents a model that formally characterizes transit routing failures.
BGP Health Inferencing	Caesar et al. [33]	Discovers the cause of route changes by classifying updates in groups according to correlations between them.
Popular prefixes	Rexford et al. [92]	Observed that a small group of unpopular prefixes originates routing instability.

Table 3.1: Descriptive and Analytical Approaches (part 1)

Solution	Authors	Short Description
Ranking and Filters	Feamster et al. [94, 95]	Proves that under policies expressed by rankings and filters and ranked based on AS_PATH length, BGP is guaranteed to converge.
BGP Pass-Through Times	Feldmann et al. [96]	Proposed a methodology for studying the dependency of BGP pass-through times as factors related to routers running BGP process.
Research of BGP Convergence Time	Wang et al. [97]	Proved by experimental results that improving MRAI guarantee network stability and robustness, and also accelerate the routing convergence time.
Convergence Delay and network topology	Wenhua et al. [98]	Analyzed the relationship between topology parameters and route convergence delay.
Churn issues	Schrieck et al. [99]	Identified some factors that may cause BGP churn issues.
BGP churn evolution	Elmokashfi et al. [100]	Presented a study on the evolution of BGP churn in four networks at the Internet backbone.
BGP churn factors	Elmokashfi et al. [101]	Observed that the churn of BGP updates are caused by an interplay of some factors.
BGP churn prediction	Elmokashfi et al. [102]	Presented a model that tries to predict how BGP churn will evolve in the future, considering interdomain topology, routing policies, and traffic characteristics that frequently changes.
BGP churn report	Geoff Huston [103]	Presented a report on BGP churn in the period from 2008 to 2014.
CAGG (Churn Aggregation)	Wang et al. [76]	Tries to reduce path exploration by aggregating multiple AS_PATHs in a route to be announced, without harming convergence.

Table 3.2: Descriptive and Analytical Approaches (part 2)

Table 3.3 presents a summary of the main approaches detailed in this chapter and a short description of them.

3.2 Speeding up approaches

A modification presented in [40] created the ghost flushing rule, which makes the router to send for withdrawal of a prefix to its neighbors at the moment when failures occur, without impeding of a timer such as MRAI. It reduces the convergence delay from *n*30* to *d*h*, where *d* is the length of the longer *AS_Path* and *h* is the average delay between two BGP neighbor routers. Another rule, the ghost busting, ensures that the announcements are delayed by the MRAI timer. However, ghost flushing modifies the BGP semantics, by transforming one implicit withdrawal in one explicit withdrawal and one new route announcement, which makes BGP unsafe because some routers may be without a route to destination for a certain time while a stable path to destination exists.

Solution	Short Description
Speeding up Approaches: [40], [31], [104], [105], [106], [107], [108], [109].	Makes convergence faster by accelerating the propagation of updates, changing timer settings or by modifying BGP path selection process.
Limiting Path Exploration: [5], [110], [111], [87], [112], [39], [73], [19], [76], [113], [70], [114], [64], [6], [4], [10], [115], [116].	Makes less updates circulate through network, trying to eliminate inconsistencies, and reducing convergence delay.
Efficient Policy Configuration: [117], [118], [119], [120], [121], [122], [123], [124].	Conflicts due to policies are solved and inconsistencies that introduce delay on network are avoided.
Multipath and Multi-path Forwarding: [125], [38], [80], [20], [126], [7], [127], [128], [129], [75], [130].	Packets carry alternative routes beside the best ones, and when convergence events happens, they quickly switch to the backup routes, or may be forwarded to multiple next-hops.
Centralized Control: [61], [70], [131], [132–135], [136], [137], [136], [138], [63], [139] , [140], [61], [141], [142], [27],[8], [143], [15], [144, 145], [146], [61], [147], [148], [149], [65].	Enables a high level and powerful management of the network, with the potential to enable the creation of new solutions to reduce routing convergence of Internet.

Table 3.3: Classification of efforts to reduce Convergence Delay in the Internet

The flight-or-fight response mechanism is presented as an alternative approach to ghost-flushing solution [31]. The *flight-or-fight* term refers to a dangerous or emergency situation where humans have to take a quick answer to survive. The flight-or-fight mechanism recognizes that a route failure is a short-term survival situation, which requires agility in path exploration, ignoring the analysis of several parameters that are analyzed on the RIB when a new path needs to be chosen. The only analyzed cases are if the route is the older and the selected ID is the lowest. So, a failure is treated by the mechanism as an emergency situation. An announcement message, however, is not dealt as urgent, keeping the routing selection process as it is being done currently by BGP. The flight-or-fight mechanism may be incrementally deployed, since it still working with standard BGP, but changes BGP behavior by modifying its natural selection process.

Nykvist et al. [104] tested the effect of different settings of MRAI timers on BGP's convergence properties. They used clique topologies on their experiments and tested the *Split Horizon* (SH) and sender-side loop detection (SSLD) technique on the BGP simulator they built. In the SH method, the routers do not announce a route to the same peer from which they learned the path; in the SSLD method, the router makes a loop detection on its routes before sending it to peers. It differs from BGP, where the receiver is responsible for loop detection. With SSLD the risk of generating unnecessary path exploration is reduced [4]. For routing protocols based solely on the path, the computation will finish

in finite time and it might be proven. BGP, however, is also driven by policies, rendering the proof of its convergence impractical, unless that the constrains defined by models such as Gao-rexford [1] are applied, following the determined economic conditions. Besides that, keeping the topology unmodified but changing policy settings may lead to a different behavior of the network. Their experiments showed that SSLD improves convergence time. They also showed that increasing MRAI timer from 0 to 8 seconds leads to a decrease in the number of announcements and in convergence time, considering the settings chosen for their experiments.

Deshpande et al. [105] characterized the impact that the topology and message processing have on routing convergence time. They evaluated the convergence time of random generated graphs in terms of MRAI *rounds* by obtaining analytic expressions from them. With the analysis, it was possible to identify classes of scenarios where MRAI timers became redundant. Thereafter, some proposals to reduce convergence time was presented. From the simulation, they observed that when a preferred route (a shorter one, for example) is held by MRAI, if new less-preferred routes arrive, when no link issues or policies are detected, it indicates that invalid MRAI instances are installed on the peers that sent the new routes. To avoid the redundant MRAI occurrences, the authors apply the following procedure:

- When an advertisement arrives, check if MRAI is installed for the sender.
- Check if the received route length is longer than the one currently used for the destination.
- If the two previous conditions occur the MRAI is canceled for that destination and the announcement is sent.

They also proposed a dynamic MRAI, where a variable called MRAI_THRESHOLD is used for each BGP peer, and updated according to the frequency of route changes. Besides this variable, a MRAI_COUNTER entry is used to count the number of received routes for a destination. If the network presents several oscillating routes, the MRAI_THRESHOLD might decrease to eliminate unnecessary updates. Lasković et al. [106] also proposed an adaptive MRAI algorithm to dynamically adjust MRAI values. They called this implementation *BGP with adaptive MRAI* (BGP-AM). Their results with ns-2 simulator showed a reduction in convergence time and number of updates when compared to traditional BGP.

The work of Elmokashf et al. [107] explored some guidelines for configuration of rate-limiting timers such as MRAI in order to achieve desired convergence properties. They analyzed how the interaction of different timers and configurations impact on the churn present in a BGP session. Besides MRAI, which is a timer implemented typically on Cisco's

routers, there is *OutDelay*, a timer implemented on Juniper's routers that differs from MRAI by holding each route during the value specified as OutDelay value, instead of adding a delay for each prefix or peer individually, as MRAI does. They observed that the use of standard MRAI and OutDelay values reduce the daily rate of updates of Internet in two-thirds, and the increase in the standard value logarithmically reduces churn. They also presented a model that quantifies the impact of timers on convergence reduction. Pei et al. [150] observed that transient forwarding loops may occur during the convergence process and that the duration of the transient loop is close to the convergence time. Besides that, they showed that the loop duration is linearly proportional to the MRAI value used. Pei et al. then experimented some mechanisms that tries to reduce convergence delay, but they stated that in their experiments only two of them avoided forwarding loops and speed up the convergence process – the Ghost Flushing and the Assertion [5] mechanisms.

Sun et al. [108] proposed a mechanism called DUP (Differentiated Update Processing), that reduces convergence time by 80% and might be incrementally deployed on the Internet. On the DUP approach, BGP updates are clustered in different classes representing different priorities. Updates with high priority are processed and propagated quickly, while updates with low-priority might be delayed. The DUP algorithm observes if, while sending an update to a neighbor, this neighbor has sent updates before for the same prefixes to the sender. If this happens, the update receives low priority. This scheme reduces the number of low-priority updates, the convergence time and the router overhead.

Another work about trying to discover an optimal value for MRAI timer, reducing convergence without increasing the number of advertisements, was proposed by Alabdulkreem et al. [109]. They experimented with several different topologies identifying the value of MRAI that brings the best convergence time on each topology. To determine the best MRAI value for a scenario based on Internet topologies, they used the *BRITE* and the *RouteViews* tools to generate topologies, configured test scenarios with 60 different values of MRAI using the *OPNET* simulator, trained a Neural Fuzzy system with collected results, and used the Particle Swarm Optimisation (PSO) algorithm to find the best value of each module. The experiment was done once and offline.

The constructed model performed an input-output mapping of N instances of a set, considering four inputs and one output for each instance. The first input is the number of ASes (nodes) in the network (represented as a graph). The second input is the average degree of the network graph. The third input is the diameter of the network, and the fourth input is the MRAI value (ranging from 0 to 30 seconds). The output is composed of the convergence time and the number of exchanged updates in the network. This set was used to train the neuro fuzzy system. For each experiment, a node was disconnected from network, causing

routing changes and triggering the sent of update messages to neighbors until the system reach a stable state. Then, the number of exchanged message and convergence time was recorded, and the PSO algorithm [151] was used to determine the best value for MRAI. The authors concluded this value as 3 seconds, reducing convergence time and the number of exchange updates by 45%.

The main issue with this approach is that it is not scalable, requiring a global network view to collect all the required data to Neuro Fuzzy, which is impractical due to the fully-distributed nature of Internet architecture.

To disperse the load and reduce the routers' overhead Gill et al. [152] proposed the MRAI with Flexible Load Dispersing (FLD-MRAI). A router CPU load depends on the number of messages received during a specific MRAI round. Using the empirical value of 200ms, FLD-MRAI considers a scenario of normal load when the Degree of Preference (DoP) of a route is always the one with a shortest path. When DoP prefers longer paths, FLD-MRAI considers this scenario as a high load.

Instead of using default MRAI, FLD-MRAI modifies MRAI values based on Tup, Tdown, Tshort, and Tlong events. Besides that, two categories are considered when choosing MRAI reusable timers by FLD-MRAI: the active and the idle times. A long idle interval during a previous MRAI round may indicate that the active interval is small and the update was advertised in shorter time than the default value. A short idle interval, on its turn, may indicate that the active interval is longer than expected and that the previous round should have lasted longer.

The MRAI value setting that leads to faster convergence time represents a trade-off, since the higher the deployed MRAI value, the more NLRIs are batched into fewer update messages. However, it creates a delay in the propagation of those updates. In contrast, a considerable short MRAI value can propagate updates faster, but at the cost of less batching of the updates and a higher amount of BGP messages, which can lead to convergence time increasing [153]. The main cause of the updates batching is the delay that each router presents while processing updates during CPU overload.

Summary: As advantages, we may highlight that some improvements brought by speeding up approaches are currently used in BGP, e.g., the ghost flushing rule, where the withdrawal is sent to neighbors without restriction of MRAI. A reduction in convergence delay was achieved when it became widely deployed [40]. Also, this subsection characterizes the factors that most impact on interdomain routing convergence, like topology and message processing time, and other different timers and configurations. This give researchers a better understanding of protocol behavior, for example, the relation between transient forwarding loop and convergence delay, during convergence process.

However, the presented approaches have some disadvantages. Reduced convergence time by itself does not indeed improve packet delivery. Actually, seeks to speed up convergence time by modifying BGP parameters without trying to discover and tackle the root causes of delayed convergence does not eliminate inconsistencies. Decrementing the MRAI timer might create the illusion of faster convergence as a result, but it was shown in this subsection that it may even worsen convergence time when a wide range of MRAI values are applied in different ASes.

The use of timers such as MRAI is important in high load scenarios where routers become overloaded and the number of propagated updates is higher than in normal load scenarios. However, changing timers to speed-up convergence might be done very carefully, with a consistent planning. The Griffin [88] and the Gao-rexford [**?**] models can be extended to a path propagation and processing model to consider the MRAI, the network load, the available router CPU, and the router's queue where updates wait to be processed. Most of the approaches described in this subsection may be incrementally deployed and coexist with standard BGP implementation since they generally focus on modifying BGP timers without changing BGP behavior or requiring global coordination. On the other hand, to have a significant impact on interdomain convergence the majority of them require the large acceptance and deployment of ASes and ISPs. Those are very interesting approaches, but they should be carefully combined with other approaches to achieve better results.

Table 3.4 presents a summary of the main approaches detailed in this section, where: **(1)** = Incremental Deployment; **(2)** = Computational Overhead; **(3)** = Changes BGP behavior; **(4)** = Constraints to BGP expressiveness; **(5)** = Coordination between ASes.

3.3 Limiting path exploration

Some methods try to improve convergence only by speeding up the sending of the messages or adjusting MRAI value, as detailed before, but do not handle the root of the problem while trying to improve path exploration [67]. Figure 3.1 illustrates an example of path exploration induced by a link failure. When the link between AS1 and AS2 fails, AS2 sends a withdrawal W to ASes 3 and 5. AS5 explores a new path, $5 \Rightarrow 3 \Rightarrow 2 \Rightarrow 1$, and announces it to its neighbors. When AS3 sends a withdrawal to ASes 4 and 5, AS5 explores a new path, $5 \Rightarrow 4 \Rightarrow 3 \Rightarrow 2 \Rightarrow 1$, until receives AS4 withdrawal and discovers that AS1 is unreachable. Until this, successive path exploration occurs, as well as packet loss, and the convergence process suffers a high delay. This subsection presents approaches that limit path exploration, reduce the number of propagated updates, eliminate inconsistencies, and decrease convergence delay.

Solution	Short Description	(1)	(2)	(3)	(4)	(5)
Ghost-Flushing [40]	Reduces convergence delay from *n*30* to *d*h* by sending withdrawals without impediment of hold timers.	✓		✓		
Flight-or-fight [31]	Ignores several parameters of path selection when update is a withdrawal, speeding path exploration.	✓		✓		
MRAI + SSLD [104]	Tests different settings of MRAI timers and show that the use of SSLD with MRAI decreases the number of announcements and convergence time of the routing system.	✓				
Redundant MRAI [105]	Identifies classes of scenarios where MRAI values become redundant and proposes techniques to avoid these occurrences.	✓	✓			
BGP-AM [106]	Presents an adaptive version of MRAI that adjusts its value in order to reduce convergence time.	✓				
OutDelay / MRAI [107]	Gives some guidelines for configuration of MRAI(Cisco) and OutDelay(Juniper) timers and quantify the impact on convergence reduction.	✓				
Differentiated Update Processing (DUP) [108]	Updates are clustered on different classes according to priorities, and the ones with higher priorities are propagated.	✓	✓			
MRAI Optimization [109]	Experiments several different topologies identifying the value of MRAI that brings best convergence time without increasing the number of Updates.					✓
Flexible Load Dispersing (FLD-MRAI) [152]	Disperse the traffic load and reduce the routers' overhead.		✓	✓		

Table 3.4: Speeding up approaches

We subdivided this subsection into methods that first seek to detect instability to then apply countermeasures and others that limit path exploration without trying to detect instability first.

3.3.1 Limiting path exploration by first detecting instability

In the presence of network changes some approaches try to first detect the instabilities in the routing updates to then apply the actions that will limit path exploration and reduce convergence time.

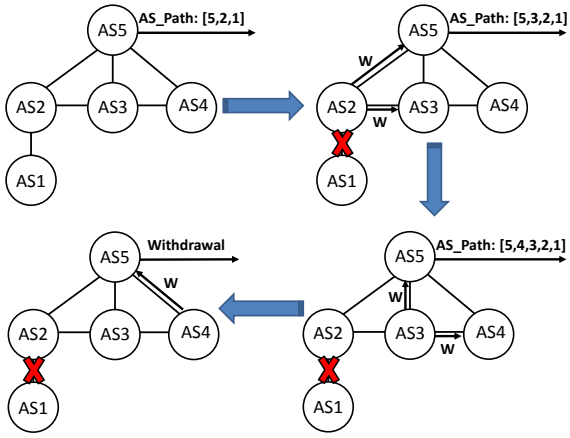


Figure 3.1: Example of path exploration triggered by a withdrawal [4]

One method that improves convergence by limiting path exploration was proposed by Pei et al. through Consistency Assertions [5]. This mechanism uses path information to create two consistency assertions for path vector routing algorithms, that are used to compare similar routes and identify infeasible routes. The developed method looks for relationships between path vector routes to detect invalid routes. This approach reduces both convergence time and the total number of route update messages. The authors used the definition of Simple Path Vector Protocol (SPVP), detailed in chapter 2 to create a Consistency theorem for SPVP, as follows:

Consistency Theorem for SPVP: Consider a Node R that learned two paths from neighbor nodes N1 and N2 to reach destination node D. Neighbor N1 advertised *(N1,A,B,C,D)* and neighbor N2 advertised *(N2,X,B,Y,Z,D)*. While comparing both paths we might say that they are not consistent. If one considers that N1 announced path is correct, then B should reach D through *(B,C,D)*. Otherwise, if one considers N2 announced path to be correct, B will reach D via path *(B,Y,Z,D)*. As B might advertise only one path to D, one of the neighbors N1 or N2, or both, are advertising an invalid path to reach D. From this theorem the concept of a valid path is defined as:

Valid Path: Suppose *(N1,P1,P2,...,Pn,D)* was the last path to D announced by N1. This path might be represented as *path(N1,D)*. According to the definition of SPVP, each node Pi (where $1 \leq i \leq n$) might use only one path, denoted by *path(Pi,D)*. For each i, *path(N1,D)* might be expressed by *path(N1,Pi)* + *path(Pi,D)*, where $path_{N1}(Pi,D)$ is the last path from Pi to D reported by N1. Considering this, a path is valid if, and only if, $path_{N1}(Pi,D)$ = *path(Pi,D)*, i.e., *path(N1,D)* is only valid if each path from each Pi to D is correct. Otherwise, the route is invalid. And from this definition comes the Path Consistency definition:

Path Consistency: Two paths are consistent if, and only if, they are both valid. The method does not indicate which one, or whether both of the paths are invalid. If a path is not consistent it is marked as an infeasible path. Figure 3.2 is used by the authors to explain what happens when a path becomes infeasible:

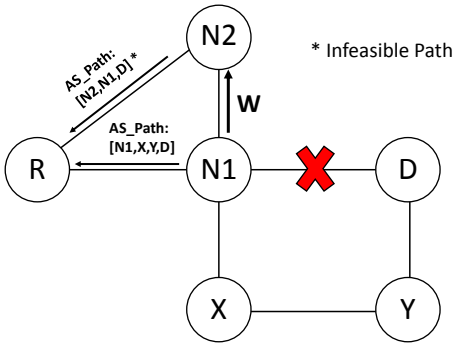


Figure 3.2: Example of topology with infeasible path after link failure[5]

Suppose a failure occurs between nodes N1 and D, in the link that connects them directly. N1 will advertise to R the new path *(N1,X,Y,D)* to reach D. But before N1's withdrawal arrives at N2, it will advertise to R the path *(N2,N1,D)* to reach D. The paths are inconsistent because they intersect at N1 and $(N1,X,Y,D) \neq (N1,D)$, so $path_{N1}(Pi,D) \neq path_{N2}(Pi,D)$. The path learned directly from N1 has a preference over the path learned indirectly from N2, so the path (N2,N1,D) is marked as infeasible and is not used by R to reach D. If R receives an announcement of path (N2,N1,X,Y,D) later from N2, it will be considered by R, since it is consistent with *path(N1,D)*, and the infeasible mark is removed from the preferred path coming from N2.

To deploy the Consistency Assertion mechanism, the BGP update message need to be modified by adding the consistency feature to the community attribute, which is a 32-bit value, capable of conveying routing policy information. Besides that, the processing and route selection algorithm also need to be changed.

The RFD is another technique to reduce path exploration. However, the authors of RFD detail some problems encountered while deploying the solution at some of the Internet Exchange Points (IXPs). One of the related issues was that flap damping should not be used with IBGP, because suppressing IBGP routes resulted in routing loops. The deployment must be done carefully and the parameters properly chosen to avoid the incurrence of critical inconsistencies.

To address the side effects of the use of RFD, which might wrongly classify a route as unstable and mistakenly increment the penalty for it, Lijun et al. [111, 154] presented a modified version of the RFD mechanism which still continues damping persistent oscillating

routes but making the relatively stable ones converge quickly. The routes carry a suppression mark which informs the destination if the route is likely to oscillate. The task of eliminating the persistent oscillating routes is delegated to the ASes closer to the originating routes, i.e. the neighboring ASes. The further ASes are responsible for the elimination of invalid routes through a new mechanism, the Invalid Routes Damping. The mechanism reduces convergence delay and communication overhead. Pelsser et al. [87] proposed a more simple enhancement to RFD, exploring the possibility to promote minimum number of modifications to it, while improving its performance on damping persistent oscillating routes and avoiding to penalize well-behaved routes. They adjusted just the RFD parameters, without any impact on normal converging prefixes. Zhang et al. [155] detected that due to route reusing at routers, false damping also might be triggered. Thus, they recommend the use of BGP-RCN or RCN to improve the performance of RFD, eliminating false route suppression.

The BGP with Root Cause Notification (BGP-RCN) [39], is a method that tries to discover the root cause of a failure or policy change. BGP-RCN proposes that each AS node listed in AS_PATH attribute of the routes presents a sequence number *ts(v)*, which will indicate if a route is valid or not. When a failure occurs in the network, the node *v* will notice the problem, selects an alternative route, increments *ts(v)* from *i* to *i+1* and announce to its neighbors about the new path using rcn = *(v,i+1)*. Without this approach, a node could select a backup route which is also invalid, as the previous route, and the error is propagated as a ghost route. With RCN, a node will learn that every route with *ts(v)* smaller than the last announced *ts* associated to node *v* is invalid. For instance, if the *ts* of node *v* has value 0 in the routes installed on RIB's router, and an announcement arrives indicating that node *v* has incremented *ts* to 1, all the routes used by the router to a particular destination containing node *v* are invalid. It prevents invalid backup routes from being chosen as new preferred routes and propagated further.

BGP-RCN requires BGP modification, by adding a new community attribute of 32 bits to BGP. The experiments were built on SSFNET simulator [156] and compared to traditional BGP, BGP-Assertion and Ghost Flushing. The authors generated some topologies with 110 nodes each, organized as Clique, Backup-Clique and Internet-derived, for different tests. The four convergence events, *Tup, Tdown, Tlong, Tshort* were analyzed in the tests for the four methods. BGP-RCN presented good performance on tests, reducing convergence time to *O(d)*, where *d* is the network diameter. With BGP-RCN the convergence is faster and much less announcements are sent than in the other three methods, for almost all convergence events and topologies tested. As per the authors' demonstration, this method greatly improves current convergence time after routing changes. However, its main issue is that it needs wide adoption by providers and acceptance of BGP modifications in order to cause significant

impact on convergence – the same challenge of the Consistency Assertions we described before.

A different approach was proposed by Sahoo et al. [112] to improve the convergence time by invalidating the routes which may be affected by large-scale failures. Those routes may generate the ghost information we described earlier. The goal of the authors was to collect statistics from the BGP speakers and identify the ASes that are probably unstable. After identifying the routes that contain those unstable ASes, the algorithm marks those routes as invalid and they are not propagated further.

The scheme to identify and classify invalid routes requires that statistics were collected at BGP routers, focusing on withdrawn and replaced routes. The solution runs independently at each BGP speaker; i.e. coordination between ASes is not required. Several updates should be collected to identify the failed/affected ASes, because those ASes will normally present a large number of the generated updates.

The used metrics were:

1- *FailCount*: A value incremented for each AS at each BGP router when a route containing that AS is withdrawn or replaced. It is the likelihood that an AS has failed. The ASes that have the largest failCounts are used to determine that the routes that contain those ASes are invalid.

2- *AvgFailedPathLength*: Indicates the average path length of the routes replaced/withdrawn during the last history time slots.

As it is difficult to detect failed ASes with a small amount of data, the scheme is only executed if the number of destinations is bigger than a parameter called *largeFailureThresh*. They tested different values for this parameter and observed that:

1- Independent of the value chosen for *largeFailureThresh*, the convergence delay was reduced when the invalidation scheme was used in relation to BGP.

2- The performance improves when *largeFailureThresh* is decreased. A lower value means that the algorithm will be executed more often and more ASes will be marked as suspect. This leads to a quicker convergence.

With higher threshold, the amount of data analyzed across the network goes down, leading to increased error rate. They observed that the convergence delay decreased by up to 75% when the scheme ran at just 20% of the nodes and up to 50% while running at just 10% of nodes, meaning that the number of nodes that run the scheme is very important to obtain good results.

Chandrashekar et al. developed an enhanced path vector protocol called EPIC [73] which limits path exploration to achieve a faster convergence. The authors demonstrate the

proposed solution correctness and gain in convergence time when compared to BGP and other solutions.

Luo et al. [19] presented the Routing Change Origin (RCO), an approach which tries to reduce convergence time in a similar way as that of BGP-RCN. RCO adds a new attribute to BGP updates to indicate a source and origin of a convergence event. The goal is to eliminate all invalid alternative routes with the same source of prefix being withdrawn. With this approach, all routes to that prefix are invalidated. The experiments done with RCO focused on Tdown events caused by withdrawals. The BGP version with RCO presented a better performance on authors' experiments in SSFNET simulator, with reduced number of updates and convergence time than the current version of BGP.

Another consistency work is the STAMP protocol, where Liao et al. [113] developed a mechanism to use multiple BGP processes to compute paths that are not affected by the same sets of events. Those paths are called complementary, and each AS has two different sets of them. The main property of complementary routes is that they need to be disjoint, i.e., besides source and destination ASes, they must not share any nodes. Two processes may be distinguished by identifying the TCP ports. The STAMP protocol ensures consistency between the two running BGP processes. The biggest advantage of STAMP protocol is that it does not change BGP behavior or adds complexities to the standard interdomain routing protocol, although it inputs some overhead with the multiple BGP created processes.

Feldmann et al. [114] proposed a method to detect origins of routing instabilities. They used observation points to correlate information obtained from updates. It may be noticed that a routing instability occurs when a withdrawal or new best path is announced, and the first AS which announced a new path is the cause of the instability. The authors presented heuristics that take the union of the paths to derive the candidate ASes for the instability. ASes presented in two compared paths may probably be removed from the candidate set, but caution is necessary. Another source of information is the lack of BGP updates. It might mean that the system is stable or one of the two communicating ASes might have gone down and thus, is the source of instability. They observed that some of the updates are local, mainly when the only attribute that changes in a route is the next hop. Global changes must bring modifications in the AS Path attribute.

John et al. [64] stated that a consistent state in a distributed system must be a priority to obtain a more predictable and secure behavior. With this in mind, they presented the Consensus routing, which separates *safety* and *liveness* properties, and uses two logical modes of packet delivery – the *stable* and the *transient* modes. They achieve high availability while using the consensus approach with transient heuristics such as deflections, backup path, etc. This availability leads to a better convergence time in the experiments done.

Separating consistency safety and liveness, Consensus routing improves system availability, and makes it less prone to errors, by making its behavior simpler to understand. Consistency safety ensures that a router will forward a packet according to the path the packet has been following from previous routers. The liveness property makes the system reacts quickly to changes, caused by failures or policy changes. In the stable mode, the consensus protocol ensures that a route is only adopted after all routers interested in using it agree about its consistent state. The transient mode maintains the network with high availability, detecting when a router does not have a stable route or when the consensus protocol is still computing a route, and uses heuristics combined with local information to forward the packet to destination, as fast as possible.

Consensus routing does not require changes in BGP, running on top of current implementations. The Consensus router takes the output of a BGP policy engine locally and uses the global Consensus algorithm to determine the most recent consistent state. A consensus router should maintain a RIB, similar to BGP, storing for each prefix the most recent updates received from each neighbor, the locally selected best route, and the route it advertised to each neighbor. Besides that, a consensus router must keep a history for each announced prefix, ordered by time, and a *Stable Forwarding Table* (STF).

Consensus protocol uses an identifier called *trigger number*, very similar to the sequence numbers used in the RCN approach. It is a globally unique identifier that ensures that when a router accepts a new update all routers that had received the old route from it has already processed the update which informs the change. The protocol counts on a structure called *consolidator* to compute the snapshots of the ASes to maintain consistency. If an AS fails to send its snapshot to one of the consolidators or loose messages, due to slow operation or misbehavior, this AS is considered inconsistent and is not used to forward traffic, therefore being excluded by Consensus. To ensure that consolidators will operate in the same information and protect the system from consolidator failures, the Paxos algorithm is used [157].

Lakshminarayanan et al. [6] proposed a new routing paradigm which aims not only to reduce the convergence time but also completely eliminate the convergence process. The proposed paradigm is named Failure-Carrying Packets (FCP), and enables packets to discover alternative paths by themselves when failures happen, without requiring routers to present complete consistency state. After a failure is discovered locally, the packets are routed to destination while a path to it exists in the network. To achieve this, all routers in the system need to keep a network map, which as assumed by authors, does not change. Besides that, the routers need to know the list of failed links in the network. When a packet arrives at a router, the router knows about any failure of this packet along the path it traversed. It avoids

the requirement of send withdrawals to all other routers and prevents transient forwarding loops.

Figure 3.3 shows an example of how FCP works. Suppose node N1 sends packets to node Nd, using the path N1 ⇒ N2 ⇒ N3 ⇒ Nd. When packets arrive at N3, it knows the link N3 ⇒ Nd has a failure, so it searches on network map for an alternative path to Nd, and insert the failure link N3 ⇒ Nd information on packets' header. When packets arrive at N5, it puts the information of failed link N5 ⇒ N7 and forward packets through link N5 ⇒ N6, until it reaches Nd.

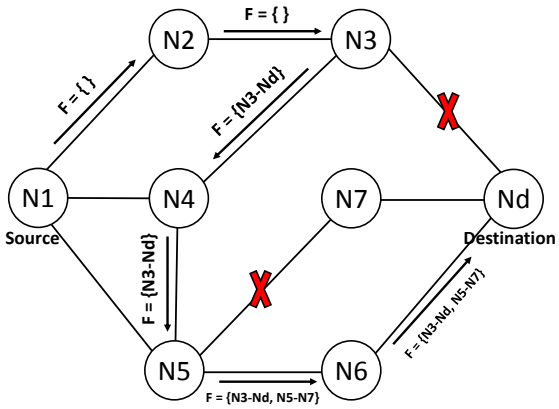


Figure 3.3: FCP routing example [6]

To allow the routers to have different network maps maintaining consistency, authors presented FCP with source routing, the SR-FCP. With this extension, packets have the entire routes to destination inserted into their headers by routers. The packets are then routed successfully to destination. The main drawback of SR-FCP is that it might increment substantially the packet overhead, adding source routes to it. Although FCP was designed originally for intradomain environment, as OSPF, the authors extended it to improve interdomain routing and demonstrated how it might contribute to AS stability and reduce convergence delay. The deployment of this solution is a challenge, since it requires the insertion of failed links listing the AS source routes into the packet headers.

Huston et al. [4] proposed a new approach to limit path exploration, the Path Exploration Damping (PED). Authors argue that PED is a more efficient mechanism than MRAI, RFD, and WRATE, in the task of correctly suppressing some routes and reduce path exploration. PED suppresses routes with AS_PATH equal to or longer than the previously announced route to the same destination. They collected traffic from two ASes and experimented the performance of a BGP speaker with PED, obtaining a reduction of 32% on total number of announcements and 77% on path exploration, compared to MRAI. Two convergence states

were identified in [36], being *reachability* and *optimality*. The article stated reachability as the most important state. Optimality ensures that every BGP speaker knows the best route to reach a destination. Reachability ensures that a route to reach destination always exists. Huston et al. analyzed RFD, MRAI, WRATE and PED on both stages, and detected that PED always achieves reachability at least as fast as MRAI.

Stabilizing BGP route selection is also an alternative to reduce path exploration, as detailed by Godfrey et al. [116]. They observed three possibilities to work in BGP's selection process:

- Reduce the number of exchanged routing information due to an external failure.
- Identify and select the most stable routes, avoiding convergence events.
- Filter all routes between a source and a destination during an instability period.

The first technique may be limited by its potential impact. The second technique incurs in a trade-off between deviation and stability. Network operators define their preferred routes, which may introduce instabilities depending on the deployed policies. When the chosen routes differ from the routes preferred by operators, authors state that a *deviation* occurs. By improving stability one may impact on deviation from preferred routes, which is undesired since deviation is strongly related to routing policies, traffic engineering and other operator interests. The third technique incurs in a trade-off between stability and route availability.

To provide a safe trade-off between the three techniques described above, Godfrey et al. proposed the *Stable Route Selection* (SRS). This approach is flexible, preferring more stable paths in route selection process, with minimum deviation from ASes policies, without reducing availability. The experiments were made in small-scale and large-scale scenarios and showed better stability improvement than RFD, which is deployed and used in most routers in the Internet.

Liun et al. [158] designed a mechanism named *time window* based on the penalty value of RFD to improve BGP routing convergence. The mechanism judges the route stability in the network by observing multiple correlative routes received from different peers jointly. Then, the BGP speaker may find unstable routes earlier and ignore them in routing decision process. Time window mechanism makes stable routes get the chance to be selected earlier, thus curtail path exploration. No additional information is required to be added to the BGP updates, making time window mechanism a more practical method to be incrementally deployed on the Internet.

Tables 3.5 and 3.6 present a summary of the main approaches detailed in this section, where: **(1)** = Incremental Deployment; **(2)** = Computational Overhead; **(3)** = Changes BGP behavior; **(4)** = Constraints to BGP expressiveness; **(5)** = Coordination between ASes.

Solution	Short Description	(1)	(2)	(3)	(4)	(5)
Consistency Assertions [5]	Uses path information to create consistency assertions for path vector algorithms and compare routes to identify inconsistent paths.			✓		✓
Route Flap Damping (RFD) [110]	Identifies unstable routes and suppresses them by establishing penalties until the routing scenario becomes stable.			✓		
Invalid Routes Damping [111]	Modifies RFD to make relatively stable routes, converges quickly by adding to them a suppression mark, indicating the likelihood of oscillation.					✓
Adjusted RFD [87]	Proposes minimum changes to RFD to improve its performance on damping persistent oscillations.			✓		
Speculative Route Invalidation [112]	Collects statistics from BGP speakers to detect probably unstable ASes and mark the routes as invalid, which contain instabilities.	✓		✓		
Root-Cause Notification (RCN) [39]	Proposes that each AS listed in an AS_PATH holds a sequence number indicating if a route is valid or not. When a node notices a failure, it updates its sequence number, invalidating backup invalid routes.			✓		✓
EPIC [73]	Describes an enhanced path vector protocol to limit path exploration after failures.		✓			

Table 3.5: Limiting Path Exploration by first detecting instability (part 1)

3.3.2 Limiting path exploration without previous instability detection

There are approaches that deploy actions that will limit path exploration without the requirement of first detect the occurrence of instabilities in the network. Some of them create estimations of the routes with paths likely to present instabilities, then deploy countermeasures to those selected routes, and finally evaluate the percentage of instability events detected.

Gupta et al. [70] described an approach called *Secure Multi-Party Computation* (SMPC) to address some BGP issues like lack of flexibility and autonomy. They noticed that most of the interdomain routing proposals to solve known BGP problems do not consider that currently only way to achieve success was keeping BGP's distributed model largely intact. The authors proposed that a number of k clusters may do the interdomain computation, instead of doing this locally at routers, as it is done nowadays. This increases or at least keep current degrees of flexibility and autonomy. The convergence may be achieved faster by this approach, because the routing processing for all domains is done by the clusters, instead of trusting the long path exploration process of each domain. With this approach there is no further need to set parameters such as route-flap damping threshold, MRAI timers, etc. The

Solution	Short Description	(1)	(2)	(3)	(4)	(5)
Routing Change Origin (RCO) [19]	Adds a new attribute to BGP updates to indicate source and origin of a convergence event.		✓			
Churn Aggregation (CAGG) [76]	Aggregates multiple AS_PATHs in one route to be announced without harming convergence.			✓		✓
STAMP protocol [113]	Uses multiple BGP processes to compute paths that are not affected by the same set of events.	✓	✓			
Locating Instabilities [114]	Presents heuristics to detect origins of routing instabilities.	✓				
Consensus Routing [64]	Separates consistency safety and liveness, and makes it less prone to errors and simple to understand.					✓
Failure-Carrying [6]	Proposes a new routing paradigm that enables packets to discover alternative paths by themselves.		✓	✓		✓
Path Exploration Damping (PED) [4]	Suppresses routes with higher accuracy than RFD, reducing path exploration to 77%.	✓				
Stable Route Selection (SRS) [116]	Stabilizes BGP route selection process, providing a safe trade-off between stability, deviation, and availability.	✓	✓		✓	
Time Window [158]	Designed a mechanism based on the penalty value of RFD to improve BGP routing convergence.	✓			✓	

Table 3.6: Limiting Path Exploration by first detecting instability (part 2)

convergence time becomes a matter of delay between the network and the clusters, which might be resolved with improvements in machines and algorithms. The clusters are able to pre-compute paths after a failure to identify inconsistencies that could cause disruptions in the Internet. With this approach future improvements on the protocol would require changes only in the cluster, preserving the interdomain protocol from suffering significant modifications.

Bremler-Barr et al. [10] observed that most research on routing convergence focus on fail down events, but high complexity is also observed after up events, such as the re-establishment of a link. They observed race conditions during path explorations, i.e. with link delay variance routers may choose less preferred routes before receiving the updates with the most preferred ones. The authors suggested minor modification to BGP so that updates are ordered, redundant updates eliminated, and convergence latency reduced. The message complexity is also reduced from O(DE) to O(E), where D is the Internet diameter and E the number of Internet connections.

A pseudo ordering algorithm was introduced by Bremler-Barr et al. making little changes on the waiting rule applied by MRAI in the BGP speakers, therefore waiting long enough

to make sure it received a less preferred route. Each router should wait *Delta* seconds after its preferred route has changed to make an announcement. The delay Delta might be given by D*h in a basic version, where D is the network diameter and h is the maximum link delay between two BGP speakers; it might be given by *min(l*h,Dh)* in an adaptive version, considering the AS_PATH length in the route. The authors showed that their solution reduces wrongly triggered events of RFD, and also reduces the load on routers, since in the experiments every node sent announcements to neighbors only once. The decrease in message complexity makes it easier for the researchers to do the root cause identification of inconsistencies.

Another approach to reduce path exploration was presented by Lambert et al. [115]. They enforced update messages that arrive on routers to be ordered, eliminating the need to share additional information between ASes. This is achieved by a mechanism called MRPC, for metrics and routing policies compliance. They showed that if timers are carefully designed and correctly scheduled, the order updates are propagated through the network, convergence time as well as the exchange of updates between routes might be reduced. MRPC relies only on path metrics and its routing policies, and presents considerable gain when compared to MRAI timer.

Table 3.7 presents a summary of the main approaches detailed in this section, where: **(1)** = Incremental Deployment; **(2)** = Computational Overhead; **(3)** = Changes BGP behavior; **(4)** = Constraints to BGP expressiveness; **(5)** = Coordination between ASes.

Solution	Short Description	(1)	(2)	(3)	(4)	(5)
Secure Multi-Party Computation [70]	Proposes that the interdomain computation might be done by a number of k clusters, freeing each domain from the long path exploration process.	✓				
Message Ordering [10]	Suggests minor modifications to BGPs that updates are ordered and convergence delay is reduced.	✓	✓			
MRPC timers [115]	Order updates that arrive on routers and establish guidelines to carefully design timers to correctly schedule the order updates are propagated through network.	✓	✓			

Table 3.7: Limiting Path Exploration without previous instability detection

Summary: The efforts on limiting path exploration have the advantage of addressing the root cause of problems due to protocol's complex behavior, thereby eliminating inconsistencies. The faster routing convergence comes naturally as a consequence, in the case of successful deployment of the proposed mechanisms.

Although very promising in solving the slow interdomain convergence problem, most of the mechanisms described in this subsection were not effectively and widely deployed, mainly because they require changes in BGP attributes or in BGP selection process algorithm, or require coordination between ASes, thereby encountering resistance in its adoption. However, some of them are capable of being incrementally deployed, coexisting with standard BGP implementation. The few mechanisms that achieved a high level of deployment, like RFD, which was found later to actually increase convergence delay, incorrectly penalizing well connected ASes. Besides that, some of the described approaches heavily increments the overhead at routers while trying to detect inconsistencies in paths, negatively impacting the convergence time.

The presented methods are capable of solving several issues on the convergence topic with the recent emergence of new routing paradigms that coexist with BGP. These methods are very promising considering BGP's distributed model, scalability, and also requiring only minor changes in architecture and infrastructure.

3.4 Efficient Policy Configuration

Griffin et al. [159] demonstrated that there exist sets of policies that cause ASes to exchange updates indefinitely, without converging to a stable state. When policy conflicts are absent, BGP will certainly converge to a stable state, like other routing protocols. Labovitz et al. [160] showed that the policies applied by the ASes and the deployed topologies have a great impact on interdomain routing convergence. Complex routing policies in BGP increases the number of messages in the Internet backbone [10]. The existence of consistent policies, free from conflicts, is fundamental for convergence stability. We present here some efforts that try to address the problem of inconsistencies generated by inefficient policy configuration.

An approach that tries to address this problem was proposed by Ahronovitz et al. [117], which created a dynamic method that detects and solves BGP routing oscillations. Each BGP router should keep only local path state information to detect instabilities. The decision on forwarding or withdrawal of a message is based on local data and local policy. The AS routing policies are respected and few constraints are required to ensure convergence to a stable state.

Varadhan et al. [118] suggested a combination of policy analysis, suppression of unstable routes, and the advantages of the widely deployed commercial Internet infrastructure, to handle route oscillations.

Obradovic et al. [119] analyzed the impact of preference policies on convergence time. Besides, they also proposed a real-time model to obtain a theorem to determine the upper bound of convergence time.

Zhang et al. [120] presented the Grapevine Border Gateway Protocol (G-BGP), that was designed to remove fault-agnostic instability on networks under different routing policy scenarios. The G-BGP modified original BGP by inserting into the update messages, fine-grained information about which nodes are affected by failures. They also demonstrated analytically that, by removing fault-agnostic instability, G-BGP achieves optimal convergence time in different test scenarios with up to 115 ASes, while BGP demonstrates poor convergence performance. In turn, Sami et al. [161] showed that if two stable routing outcomes exist, it is possible that BGP persistent oscillations occur. The authors proved that to achieve BGP safety it is necessary to restrict ASes policy expressiveness.

Observing that ASes are motivated by business relationships, usually involving competition, Levin et al. [121] developed a game-theoretic model, where agents are network nodes aiming to send traffic to a destination. The results show that in realistic settings, and with its competitive policy characteristics, BGP is incentive-compatible, making it possible to apply game theory. The authors defined two games: The Convergence Game and the One-Round Game. In the first game, each AS is instructed to receive updates from neighbors that announce their destinations, choose a most-preferred node from the route which is the best to send traffic to and announce new route to all neighbors. The One-Round Game is the full-information and non-sequential game underlying Convergence Game, and applies Pure Nash equilibria to incur in stable solutions. The theory executes best-reply dynamics as the best-interest of all AS players.

Wang et al. [122] ensured global stability by applying a wider range of policies than the best-known sufficient conditions of BGP, by presenting neighbor-specific BGP (NS-BGP), which allows for extension of the best route selection of routers. Each router might select a route based on its length, security, or even select a less expensive route. They showed that NS-BGP provides more flexibility to ASes while guaranteeing routing stability. By modifying BGP route selection process, NS-BGP changes BGP standard behavior.

Guo et al. [123] proposed a model to deeply analyze the impact routing policies have on interdomain convergence, taking into account the wide use of Multi-Exit Discriminator MED, one of the BGP attributes used on its path selection process. They were motivated by the observation that many works on interdomain convergence ignore internal AS complexity and MED attribute while modeling their scenarios, with each AS being represented by only one node in a graph, and a single link to each neighbor. MED is important because it is used by many ASes to model cold potato routing of IBGP sessions. One of the first depth

analysis on the MED oscillation problem was made by Griffin et al. [162], creating the formal definition of *MED Induced Routing Anomalies* (MIRA), and showing that the problem may spread across many ASes. The main contribution of Gao et al. is to provide a model for interdomain routing with iBGP sessions and general policies considering MED attribute, analyzing inconsistencies in route preferences during convergence. Then, they represent all general routing policies as relationships of three routing policies and prove that the absence of dispute cycles is a sufficient condition to ensure BGP convergence. A dispute cycle is defined by Griffin et al. [159] as a circular set of relationships involving routing policies of different ASes. The model proposed by Guo et al. may be incrementally deployed since it does not modify BGP behavior or attributes, but improves the use of MED.

Li et al. [124] proposed an integrated solution called *stableBGP* that aims to solve a general class of BGP instability problems, namely path explorations and route oscillations. Path explorations are known as transient instability, as nodes will oscillate but finally the routes with failed links are withdrawn until stable routes are chosen. Route oscillations are known as persistent instability since nodes keep changing its chosen routes due to routing policy conflicts. They enhanced BGP by adding a new attribute to UPDATE message to indicate the root cause of any route change, represented by an AS_PATH segment. To make the identification in cases of possible policy conflicts, a necessary condition to route oscillations due to policy conflicts was defined as follows:

Necessary condition to route oscillations: It occurs when routes with higher preference are replaced by routes with shorter or equal AS_PATH length. To avoid this condition, all nodes must not select again longer paths that have just been replaced.

In the case of link failure, stableBGP relies on root cause to remove invalid routes, inserting in the UPDATE message its path segment. When a neighbor receives the UPDATE with the root cause attribute, it directly filters any routes containing the indicated path segment. StableBGP generates a root cause when two events occur: *Good News* or *Bad News*. *Good news* event occurs when route selection is triggered by route updates with no failures, but in presence of policy conflicts. *Bad news* occurs when route selection is triggered by failures in network links, making path explorations to take place. To indicate this, they included in the UPDATE a Cause Type (CT) field, besides the Root Cause (RC). The authors conducted several experiments with stableBGP, with different scenarios, considering the following metrics: number of route changes, convergence time, and the number of BGP updates exchanged. StableBGP was compared with ordinary BGP, and other approaches like Root Cause and Ghost Flushing, having better performance than them on the three metrics used for evaluation.

Cittadini et al. [163] contributed with a methodology for systematically check BGP policies for convergence. They provide a description of a heuristic algorithm that statistically checks BGP configurations for guaranteed routing convergence without false positives. Besides that, they propose a modular tool that processes native router configurations and reports the presence of potential oscillations. Authors verify whether a given BGP network is stable by simply analyzing static properties of its configurations, without considering protocol dynamics. The proposed heuristic, the Greedy+ algorithm, extends the Greedy algorithm, guaranteeing that only stable configurations are reported as to converge to a stable state.

Summary: The efforts on efficient policy configuration have the advantage of addressing inconsistencies due to policy conflicts. The proposed methods deal with routing oscillations by providing a better understanding and modeling of the policies applied to ASes. We observed that many approaches on BGP convergence delay consider simple models of BGP in their experiments, such as represent each AS by one single router or by extremely simplifying its policies by considering only the shortest path criteria of the selection process.

But by trying to modify policy configurations or reduce the likelihood of instabilities, some of the approaches of this subsection might constrain policy expressiveness, which is one remarkable characteristic of BGP, and change BGP behavior. Other approaches require coordination between ASes to work, while some others are lightweight mechanisms that could be incrementally deployed.

The described approaches represent an important step in the understanding of the complex and dynamic behavior of BGP. With new routing paradigms that provide more management of network operator, most of those approaches might be adapted and successfully deployed.

Table 3.8 presents a summary of the main approaches detailed in this section, where: **(1)** = Incremental Deployment; **(2)** = Computational Overhead; **(3)** = Changes BGP behavior; **(4)** = Constraints to BGP expressiveness; **(5)** = Coordination between ASes.

3.5 Multipath and Multi-path forwarding

In our research, we found two different concepts of multipath routing. To avoid misinterpretations we will categorize as *multipath* the approaches that offer the possibility of alternative routes to be used quickly when a convergence event is detected, without the need to spend time exploring the RIBs looking for a new best path. In this case, more than one path is announced to the same destination in an update message, but the update still being forwarded through one next hop at each routing step.

Solution	Short Description	(1)	(2)	(3)	(4)	(5)
Dynamic Oscillation Detection [117]	Presents a dynamic method that detects and solve BGP oscillations based on local policy.	✓				
Policy Combination [118]	Suggests the combination of policy analysis with suppression of unstable routes to handle oscillations.	✓				
Policy Analysis [119]	Analyzes the impact of preference policies on convergence time.				✓	
Grapevine BGP (G-BGP) [120]	Inserts fine-grained information on BGP updates and removes fault-agnostic instabilities under different policy scenarios.				✓	✓
Game-Theoretic Model [121]	Applies game theory to BGP, which has policy competitive characteristics and is incentive-compatible.					✓
Neighbor-Specific BGP (NS-BGP) [122]	Ensures global stability by applying a range of policies and by extending route selection of routers with NS-BGP.			✓		
General Policies Model for iBGP [123]	Provides a model with MED attribute for analyzing interdomain routing with iBGP sessions, detecting inconsistencies in route preferences during convergence.	✓				
StableBGP [124]	Proposes an integrated solution for a general class of BGP instability problems: path explorations and route oscillations.			✓		✓
Greedy+ [163]	Contributed with a methodology for systematically check BGP policies for convergence.	✓				

Table 3.8: Efficient Policy Configuration

In contrast, the *multi-path forwarding* categorizes the approaches that send packets to the same destination through multiple next-hops, and which one of the goals include increasing the network bandwidth by utilizing the unused capacity of multiple paths to the destination.

BGP uses the SPVP model explained before, where each source node might use only one best path to reach a destination. The approaches detailed in this subsection use multipath routing and multi-path forwarding to achieve a faster convergence time.

3.5.1 Multipath

One of the proposals that aims to reduce convergence delay by offering multi-path is R-BGP [125], which ensures that the ASes on Internet remains connected, as long as the underlying network is still connected. The authors noticed that even when there is a path which is compliant with established policies, BGP may loose connectivity between the sender and the receiver. Instead of trying to reduce the convergence delay directly, they focused on protecting the data plane from the issues which may occur when the convergence process takes place. They ensure that the forwarding is never made by the data plane in the inconsistency state. It means that forwarding loops and situations like an AS mistakenly believing that a path to a destination does not exist are avoided.

R-BGP has the following properties:

- If a policy compliant path exists between two nodes in the underlying network, they will certainly never be disconnected by the BGP dynamics.
- R-BGP advertises only one path to each neighbor, like BGP does.

R-BGP uses the idea of failover paths. An AS might advertise a path, labeled as a failover path, to its neighbors. This route is pre-computed before a link failure occurs and is kept as a backup. When a transient disconnection event takes place, searching for a new route to a destination is no longer necessary, because the traffic may be quickly diverted to the failover path. Advertising failover paths without a restricted control would cause an explosion of routing updates and router overhead. The authors apply some strategies to choose the paths that make the BGP connectivity remain during convergence process. They recommend that at most one failover path should be advertised by the destination and exclusively to its next-hop domain belonging to the used paths. The failover path chosen must be as much consistent as possible against failures, i.e. the failover path must not also contain a failed link. To avoid that, the adopted strategy is to advertise, among the available paths, the most disjoint one from the primary.

A simpler and more lightweight solution is the indicative re-routing scheme, proposed by Qiu et al. [38], in which a simple indicator of alternative routes is carried with each route. It indicates if the sender has alternative routes. An alternative route is always inferior to an ordinary route. It means that an alternative route is only used when no ordinary route is available. The indicators are not influenced by MRAI timer, in order to accelerate the propagation of the indicative routes. The indicative route scheme replaces a part of the role of withdrawal in the standard BGP.

This scheme actually does not eliminate transient forwarding loops. The solution provided by authors is to combine the solutions: EPIC or Ghost-Flushing to solve transient forwarding loops and to use re-routing indicative scheme to solve the transient routing failure. This combination was demonstrated by Qiu et al. to be a good solution. The multi-path feature was used in this work, which means that when a route is exchanged, alternative routes are sent, besides the ordinary routes. With this scheme, during the failover convergence process, some optional routes are used as possible alternative routes and the relevant neighbors receive the sent packets. The experiments showed that the transient routing failures may be completely eliminated, but the transient forwarding loops may occur even more frequently.

It is necessary to notify which of the multi-path routes announced is the best. If the router that receives the announcement chooses a route among the alternative routes as the best path, a forwarding loop failure might occur. A label must be used to mark the alternative routes as less relevant than the ones which are considered as the best by the sender router.

Pei et al. [80] studied the influence of network connectivity on packet delivery during routing convergence. Increasing the connectivity on the networks improves packet delivery. An alternative path to a destination is required to ensure packet delivery when best path fails. For example, Kushman et al. [20] proposed a mechanism to improve VOIP performance by ensuring that the destinations will remain connected during convergence process, rather than directly trying to reduce convergence time.

Yang et al. [126] presented the *New Internet Routing Architecture* (NIRA) to give to the users of ISPs, control over the routes that their packets will take following a desired path to destination. This work uses source routing principles, since the source router or AS decides the path which will be taken to reach destination. This approach might impact positively on convergence once the source chooses those routes whose paths acquire good performance on packet delivery, with high throughput and low latency.

Caesar et al. [127] proposed the division of the network on the deployed topology and the status of the links. They argue that the currently used model, which recomputes paths after topological changes, generates overhead and delays convergence time. The authors advocate the separation of the failure recovery task from path computation, which will make

networks more efficient, more scalable and easier to manage. The main idea is that the router computes paths based only on the deployed topology, which is comprised of the physically or virtually deployed links, without requiring extra processing whenever the current state of the links change, either to up or to down. Two key elements of this work are multi-path and application-layer replication, because when a failure happens, changing the network state, the source node ignores this network state and simply forward the packets to the substitute endpoints. Multi-path routing simplifies traffic engineering problem and enables more effective dynamic load-balancing.

Source routing is another important approach which explores the multi-path approach to improve the exchange communication between ASes by providing diverse paths to a destination, thus reducing the dependence on the best path. Source routing is still used nowadays, but rarely, mainly because it permits users to specify desired paths, and this task is done by ISPs while creating their topologies based solely on the destination addresses. Yang et al. [128] extended source routing and eliminated some issues related to its adoption, by providing a small set of diverse paths, instead of the exponentially possible routes. They proposed a tag-based architecture that provides path diversity through routing deflections. The diverse set of paths is constructed by the routers, which might deflect packets according to the developed routing deflection rules. The selected available paths must be tagged with a hint that let end-systems know which path routers were used for a given packet that arrived. The authors argue that the solution is scalable and works well with current Internet model and may be incrementally deployed by the ISPs.

Szekeres et al. [129] analyzed the major part of multi-path efforts and their impacts in BGP behavior, scalability, resilience, and stability. Their results showed that R-BGP presented better trade-off, maintaining connectivity during convergence, at the cost of small number of extra BGP messages. There are other approaches that address network failures using alternate policy-compliant paths. For example, Katz-Basset et al. [75] presented LIFEGUARD, a system that locate failures and in a few seconds repairs data plane outages, eliminating the need to wait for the network operators to identify and mitigate the failure, which may take several minutes, depending on the failure. Another work, SafeGuard, proposed by Li et al. [130], enables packets to carry an estimate of the remaining path cost associated to a destination prefix, which makes the detection of path inconsistencies and the planning of a working path for anticipated failure and recovery, easier. It effectively reduces packet loss without increasing the complexity of the network convergence.

Tables 3.9 and 3.10 present a summary of the main approaches detailed in this section, where: **(1)** = Incremental Deployment; **(2)** = Computational Overhead; **(3)** = Changes BGP behavior; **(4)** = Constraints to BGP expressiveness; **(5)** = Coordination between ASes.

Solution	Short Description	(1)	(2)	(3)	(4)	(5)
R-BGP [125]	Ensures that ASes on Internet remain connected, while the underlying network is still connected.		✓			✓
Indicative Re-Routing Scheme [38]	Indicates the existence of alternative routes that may carry a simple indicator.		✓	✓		✓
Packet delivery [80]	Studied the influence of network connectivity on packet delivery during routing convergence.	✓				
VOIP improvement [20]	Proposes a mechanism to improve VOIP performance by ensuring that the destinations remain connected during convergence.	✓				
New Interdomain Routing Architecture (NIRA) [126]	Presents NIRA, that gives ISP users control over the path that their packets follow to reach the destination.		✓			✓

Table 3.9: Multipath (part 1)

3.5.2 Multi-path forwarding

To make multi-path forwarding possible are necessary algorithms able to compute multiple paths. A router with the multi-path forwarding implemented and active will distribute packets with the same Forwarding Equivalent Class (FEC) to multiple outgoing next-hops. So, packets or flows with the same FEC are delivered to one of the multiple next-hops using the round-robin or random criteria [164].

He et al. [165] describe the benefits of the flexible division of traffic over multiple paths, which include improving end-to-end reliability and avoiding congested paths. They provide an end host with access to multiple paths through Internet, and direct control over how can each path be used by which traffic.

Xu et al. [7] noticed that although source routing gives end hosts or edge hosts the power to select the end-to-end paths used by a route, the transit ASes remain without control over the routes, still having security issues. The main reason of why it is hard to convince ISPs to adopt source routing is due to the absence of control over the traffic that traverses through it.

Figure 3.4 presents a situation that happens in current BGP version, where an AS A wants to send traffic to F through B and C, but both B and D only send traffic to E so as to reach F, represented by the black arrows. To address this issue Xu et al. presented MIRO, a multi-path routing protocol which gives transit domains control over traffic, achieving high degree of flexibility. AS B, for example, might want to send traffic through E by financial incentives,

Solution	Short Description	(1)	(2)	(3)	(4)	(5)
Dynamic Route Re-computation [127]	Advocates the separation of the failure recovery task from path computation.	✓				
Source Selectable Path Diversity [128]	Extends source routing, eliminating some issues related to its adoption by providing a small set of diverse paths.	✓				
LIFEGUARD [75]	Locates failures and repairs data plane outages, eliminating the need of operators to identify and mitigate failures.		✓			
SafeGuard [130]	Enables packets to carry an estimative remaining path cost associated to a destination prefix.					✓

Table 3.10: Multipath (part 2)

but also might want to send traffic from A to F through C. This is not possible in the current version of BGP, because B must use only one path to F. With MIRO, each AS still learns one route from each AS, but may negotiate alternative routes. The convergence time might be accelerated with MIRO when an AS notices that a link has failed, became congested or has poor throughput, then switching to another path with high performance, and delivering BGP updates faster. But, as MIRO deals with business relationships and policies, special care must be taken to avoid violation of those relationships, which may lead to problems with the protocol convergence.

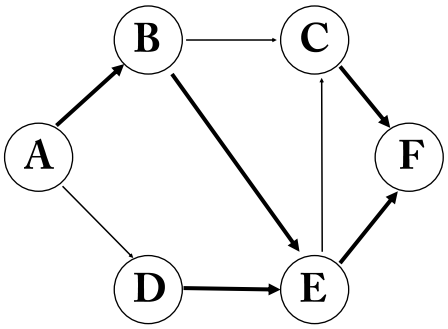


Figure 3.4: Example of routing from AS A to AS F [7]

Bless et al. [166] proposed the Fast Scope Rerouting (FaSRo) approach that tries to improve interdomain stability and reduce convergence time by limiting the notification scope of updates and switch to alternative paths. It achieves a quick reaction to a short-time problem

for the trade-off of temporarily installing a non-optimal route from a global point of view. The FaSRo process running on BGP routers detects the failure and selects an alternative path, based on its Adj-RIBs-In.

During the period T, when a link failure happens, FaSRo takes control of the error processing, keeping BGP Finite-State-Machine (FSM) unaware in order to avoid BGP issuing BGP messages. After all routes in FaSRo path have been notified, the control is given back to BGP. If link outage exceeds the FaSRo timer T it is assumed that the problem is not of temporary nature, and the behavior is switched back to normal BGP operations. In contrast to RFD, the FaSRo approach does not consume much CPU and network load. Besides that, the FaSRo mechanism isolates changes only to the peers that are necessarily affected by the change.

Table 3.11 presents a summary of the main approaches detailed in this section, where: **(1)** = Incremental Deployment; **(2)** = Computational Overhead; **(3)** = Changes BGP behavior; **(4)** = Constraints to BGP expressiveness; **(5)** = Coordination between ASes.

Solution	Short Description	(1)	(2)	(3)	(4)	(5)
Forwarding Equivalent Class (FEC) Lee et al. [164]	Delivers packets with the same FEC to one of the multiple next-hops using round-robin.	✓	✓			✓
Flexible Path Division He et al. [165]	Describes the benefits of the flexible division of traffic over multiple paths			✓		✓
Multi-path Interdomain Routing (MIRO) [7]	Gives transit domains control over traffic, achieving high degree of flexibility.			✓		✓
Fast Scope Rerouting (FaSRo) [166]	Tries to improve interdomain stability and reduce convergence time by limiting the notification scope of updates and switch to alternative paths.				✓	

Table 3.11: Multi-path forwarding

Summary: Multipath routing makes traffic engineering simpler and enables dynamic load-balance efficiently. It also accelerates the convergence process, when routers are not overloaded, by quickly selecting alternative paths, without the path exploration in RIB.

However, multipath routing presents some problems. First, most approaches spend too much memory and bandwidth to compute the alternative routes. Second, the multipath re-routing does not completely eliminate transient routing failure because, depending on the network topology and policies, it is possible that all paths are not allowed to be exchanged.

The same happens to multi-path forwarding, where the extra overhead generated at routers also led to scalability problems. To avoid this issue, routers would be allowed to announce all paths in their *rib_in*s. The main issue with this approach is that it would increase system overhead. Besides that, transient forwarding loops may occur, and it is hard to convince ISPs about its adoption because it does not offer control over the packets that transit through them.

Multipath routing and multi-path forwarding mechanisms have potential to evolve inter-domain routing to quickly react to failures and converge faster, but as speed-up approaches, they do not tackle the root cause of failures. Applying them to BGP would require changing its concept and model, achieving more success if deployed on hybrid architectures that communicate with BGP without modifying it.

3.6 Centralized Control

The service of IP assignment on Internet is done by centralized servers. The routing, on the other side, is entirely distributed, done on individual routers running inside each AS. In a centralized control scheme, the routing selection process is much more simplified since the management and control of the network are done by a single decision plane, instead of being distributed among several routers. With this architecture, the router's design also becomes more simple [61]. Besides that, the current interdomain model makes it difficult for the network operators to plan network changes because the impact of a topology or policy change is in many cases unknown [70]. As shown by Feamster el at. [131], failures are more likely to occur inside ASes than at its boundaries, which gives a large importance to centralized control.

For scalability reasons, centralized control is generally applied to *intradomain routing*. We describe some of those mechanisms but our emphasis is on the benefits that they bring to interdomain convergence. So, despite the fact that some of the solutions presented here were created to be deployed in the intra-AS domain, our focus in this section is on the impact of convergence that they might bring to the whole distributed system.

To maintain BGP speakers inside one AS consistent, all border routers might maintain a full-mesh connection, in a setup that is not scalable when the number of them increases in the AS. The most used alternative is a logically centralized router, called Route Reflector [167] (RR), that receives and gathers routes from all BGP routers in the AS, and then redistribute the best route for each destination. To improve scalability more RRs may be used and connected to a central RR, according to the size of the network. As showed by Musunuri et al. [168], RRs might solve instabilities in many cases. However, route oscillations might happen when RR is used, as detailed in some works [169–171].

A very innovative approach that uses a centralized control scheme and changes the way networks are managed today is Software Defined Networking (SDN) [132–135]. With SDN the network is divided into two parts: the *control plane*, where the logical decisions on what to do with the traffic takes place, and the *data plane*, where the traffic is forwarded by the network equipment (as switches, routers, middleboxes, etc.), according to the decisions made on the control plane.

The migration of current networks to SDN could make network management easier and impact positively on interdomain routing convergence. But SDN, in principle, was constrained to campus sized and smaller networks, besides being increasingly applied to data center networks. The existing tools that enable the emulation of SDN networks inside today's ISPs did not support until about a short time ago the necessary evaluation of the impact that the transition to an SDN-based architecture would bring [136]. Tools such as Mininet [137] are able to emulate complex SDN network topologies, but they do not provide a natural support to enable interaction with today's networks, which use legacy protocols.

An approach that tries to address this issue is the MiniNext [136] environment, which supports traditional SDN emulator, existing IGPs and BGP protocols. It was built as an extension of Mininet and aims at helping network operators to adopt SDN by enabling the evaluation of the impact on SDN deployment in networks running legacy routing protocols. Before deploying it on real networks, the desired scenarios may be replicated into the emulated environment.

One important SDN principle that may be used to improve interdomain routing convergence is the separation of routing from routers, and the creation of a logically centralized control [138]. With this approach, the management of the ASes is heavily simplified, making it easier to modify parts of the routing protocols to improve performance, convergence, and other properties, solely by dealing with software.

In [63], the control is outsourced from a group of ASes to an external trusted provider called service contractor. The contractor and the outsourcer work together on a policy plan that best attends the requirements of the client, optimizing routing inside the network. As each contractor manages several ASes, it enhances the interdomain routing by taking advantage of the logically centralizing nature of SDN to take efficient routing decisions, address policy conflicts, and troubleshooting errors that suddenly appear, while remaining compatible with current BGP, preserving its policy, privacy, and business models. This scheme improves routing stability, resulting in shorter and more stable paths, being advantageous even to ASes that are not part of a cluster, and hence, resulting in technical and economic benefits.

Gämperli et al. [139] also built an emulation framework that enables experimentation with hybrid BGP and SDN networks. They evaluated the impact that a proof-of-concept

Inter-Domain Routing (IDR) controller has on convergence time by trying to take advantage of the benefits of logical centralization provided by the SDN architecture. Similar to the MiniNext framework, this work also supports Quagga [172], a routing software suite, that runs the most famous protocols used on Internet, such as OSPF, RIP, and BGP, which makes it possible for SDN to communicate with legacy networks. An IDR controller was built over a BGP speaker inside the SDN network, using the POX controller [173], to evaluate the convergence time under network centralization.

The experiments made to test the impact of centralization showed that the convergence time may be reduced in a linear manner. The main experiment was done with withdrawal messages, varying the number of SDN nodes in scenario in a clique of 16 nodes. The announcement and failover tests did not acquire linear reduced convergence time, but small reductions were detected in it [140].

Fu et al. [61] described the steps that the IP networks go through during convergence process: failure detection time, LSA generation time, LSA flooding and reporting time, SPT computation time, and FIB update time. These steps are similar in both decentralized link-state routing protocols, such as OSPF, and networks with centralized control. Although Fu et al. solution aims to solve intradomain routing convergence, those principles might be applied to centralized control in interdomain context, like the other solutions in this section.

Yan et al. presented Tesseract [141], a system based on the 4D architecture, developed to control ASes. In the 4D architecture, the control plane of the network is decomposed in decision, dissemination, discovery and data planes. The *data plane* operates in network switches. The actions of the data plane are controlled by the decision plane. In the *discovery plane* each switch is responsible for the discovery of its capabilities, such as how many FIB entries it supports, and its connectivities to other switches in the network. When a connection information is discovered, it is reported to the decision plane via the dissemination plane logical connections. The *dissemination plane* is responsible for maintain a reliable connection between the network switches and the decision plane. The last part of the 4D architecture is the *decision plane*; all the network decisions are made by a logically centralized element that gathers information from the discovery plane. The 4D architecture is described in [142].

The Tesseract system enables direct control of the network providing two services: the dissemination service and the node configuration service. The dissemination service was explained previously, and in Tesseract it enables the plug-and-play boot-strapping of the system. The node configuration service is responsible for abstracting the nodes' hardware and software details from decision element, generating a packet lookup table interface, which enables easier knowledge of the network. A table contains a match rule for each arriving

packet and an action provided by the decision plane associated to that rule, in a very similar scheme as the one proposed by the Openflow Protocol [27].

The routing convergence of Tesseract is explored in terms of single link failures, switch failures and regional failures, besides link flapping. The concerns with convergence delay in the system are because of how much time will be expended in a re-convergence from an out-of-date to an updated network controlled by a decision element. The authors measure cases of failures for intradomain convergence in Tesseract, and for OSPF protocol. In tests with single link failures, Tesseract and Fast OSPF had similar performance, for both enterprise networks and backbone networks. In the case of switch and regional failures, in enterprise networks Tesseract and OSPF have same performance. Sometimes Tesseract has faster convergence than OSPF in regional failures than in single switch failure. It happens because in a regional failure several switches are affected, reducing the amount of updates Tesseract must send. In the case of link flapping, the experiment shows that Tesseract or another solution using centralized control may handle networks that change frequently, even because damping algorithms may be applied easily on it than in fully distributed networks.

Jain et al. [8] presented a private WAN called B4, which connects the data centers of Google distributed around the world. To make a traffic engineering that maximizes infrastructure utilization, B4 uses SDN with OpenFlow, managing its traffic routing and balancing load at switches according to the application demands. They argue that with traditional WAN architectures they did not achieve the demanded level of control, fault tolerance, efficiency and scale. They stated that traffic engineering with traditionally distributed routing protocols is sub-optimal, but with a centralized routing scheme, like the one provided by SDN, a faster and deterministic global convergence might be achieved in the case of failures. They found, for example, that most failures incur from software, rather than hardware; separating the logic part from switches makes the error detection and correction tasks easier. To achieve that, they created their own hardware to support SDN and gained efficiency and flexibility with the innovation of new services, as centralized traffic engineering.

At the *switch hardware* layer, which is equivalent to data plane, they created the OpenFlow Agent (OFA), a modified version of OpenFlow to support the hardware characteristics of their switches. The OFAs connect to remote OpenFlow Controllers (OFC) to forward certain packets and events to it and receive from OFCs the forwarding actions to take. OFCs are hosted by Network Control Servers (NCS) at the *site controllers layer*. For fault-tolerance, Paxos is used on this layer, detecting application-level failures, and electing one of available replicas as the primary instance in the case of failures. The used controller is a modified version of Onix [174]. Quagga is used to enable communication with BGP on NCS. At the *global* layer there are the network control applications (NCA), such as SDN gateways and

Central TE server. An overview of this architecture might be seen in Figure 3.5. This work gives very interesting and useful insights into the benefits of a logically centralized control scheme, like SDN, not only to convergence routing time, but also for important interdomain routing aspects.

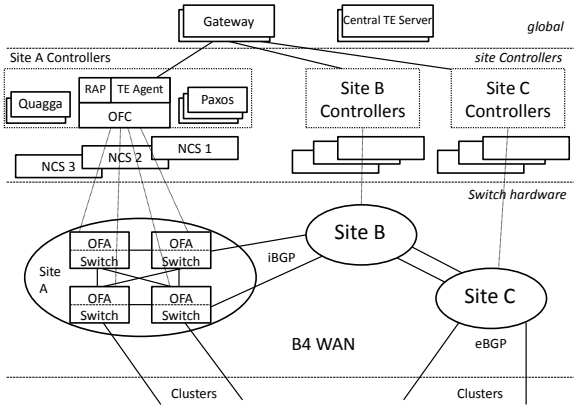


Figure 3.5: B4 Architecture Overview [8]

Vissicchio et al. [143] argued that the flexibility of SDN might bring advantages over distributed route computation. They proposed a mechanism called Fibbing, which allows the controller to create a fake topology vision for each router, according to the desired FIB. This flexibility permits the controller to reduce router load, besides improving scalability and resilience to routing failures. Most of the many approaches that uses SDN to innovate on interdomain routing are flexible to improve policy expression, and are easier to incrementally deploy, without requiring changes to standard BGP.

Other works use the centralized control approach to improve interdomain routing, and consequently obtain the benefits of this approach on convergence as we described before. Feamster et al. [15] and Caesar et al. [175] discussed how the separation of routing from routers to a control plane, creating a Routing Control Platform may help to decrease current interdomain complexity, ease management, and reduce inconsistencies. Gupta et al. [176] present the Software Defined IXP (SDX), an approach based on SDN, which promises to provide applications that improve interdomain routing, without the need to remove BGP, enabling incremental deployment in ASes. RouteFlow [177] is a project that implements an RCP and one of the first solutions that enabled the communication of SDN with legacy networks using BGP. To achieve that, RouteFlow uses a dedicated server, which runs legacy protocols, and creates one Virtual Router for each switch in the network. This solution may lead to overhead and scalability issues. Bennesby et al. [144, 145] proposed an application to provide interdomain routing capabilities to SDN, running on top of the SDN controller

called NOX [178]. This solution, however, does not enable communication between SDN-based networks and networks using BGP. Berde et al. [179] presented ONOS, a distributed SDN control platform built to achieve performance, scalability, fault-tolerance, and high availability in enterprise and interdomain scenarios. ONOS has an application which enables the interdomain communication with legacy networks – the SDN-IP [180]. SDN-IP uses Quagga [172] to deal with BGP messages and express routing policies with existing IP networks.

Another routing model based on interdomain centralization that also uses SDN and Quagga is presented by Kotronis et al. [146]. They created an emulated Python-based emulation framework called SIREN for enabling hybrid BGP-SDN experiments. SIREN enables communication between Mininet and Quagga, runs a POX controller, but uses ExaBGP [181] to implement internal BGP speaker, instead of Quagga. SIREN applies the idea of outsourcing the routing control logic to a contractor [63], as explained before, creating clusters of ASes managed by an SDN controller. Like SDN-IP, their work enables gradual penetration of SDN in interdomain environment. Kotronis et al. advocate that interdomain centralization might bring benefits to many aspects of current interdomain routing, as in policy expressiveness, network management, and network convergence time. They demonstrated one of those benefits by making a proof-of-concept experiment where they evaluated the impact the gradual penetration of cluster SDN ASes in a legacy interdomain routing system has on convergence time and stability. Their failover experiment indicates that the adopted scheme might linearly reduce churn rates and convergence times even for little SDN penetration.

Chang et al. [149] presented another important step in the incremental deployment of new routing solutions into current architecture by introducing *Supercharge Me*. The key idea is to make SDN controller to pre-compute backup forwarding entries and immediately enable them after an occurrence of failure, making failure recovery faster and improving convergence.

The centralized architecture model opens possibilities to apply new or some of the existent ideas that are not viable on current Internet model, as we saw before. One interesting example is the model proposed by Viswanathan et al. [65] that permits to analyze if a given BGP configuration is probabilistically safe, i.e. if its probability of convergence is 1. The presented procedure computes the probability of convergence of any BGP configuration, given the initial network state. They also provided a method that computes the expected convergence time for it. The BGP execution is modeled as an automaton, where the mapping between each node and permitted paths compose the *states*, and the set of announcement events compose the *transition labels*. The network states then lead to the creation of a

probabilistic transition matrix, used to compute the probability of convergence and expected convergence time of a BGP configuration. The model properties defined over time intervals ensure that the probability distribution of possible best paths, or network state, is a Markov process, enabling BGP behavior to be computed as a linear system. The BGP execution model formalism is based on Griffin et al. [26]. With the fully distributed architecture of current ASes it is an impractical task to compute large BGP configurations to compose the transition matrices. With a centralized control scheme, network information might be easily collected and used potentially to make solutions like the probabilistically safe BGP model possible in real scenarios.

Table 3.12 presents a summary of the main approaches detailed in this section, where: **(1)** = Incremental Deployment; **(2)** = Computational Overhead; **(3)** = Changes BGP behavior; **(4)** = Constraints to BGP expressiveness; **(5)** = Coordination between ASes.

Solution	Short Description	(1)	(2)	(3)	(4)	(5)
Route Reflector [167]	Receives and gathers routes from all BGP routes in the AS.	✓				
Software-Defined Networks (SDN) [135]	Divides the network into two parts: control plane, where logical decisions take place, and data plane, where the traffic is forwarded by network equipment.	✓				
Intradomain Routing Convergence [61]	Describes the steps that the IP networks go through during convergence process over the intradomain perspective.		✓	✓		
Tesseract [141]	Enables direct control of the network providing two services: dissemination and node configuration.	✓		✓		

Table 3.12: Centralized Control of Network

Summary: Centralized control has the advantage of giving operators the chance to manage its network with much ease, avoiding several misconfigurations due to management complexity. The described approaches might work with BGP in a hybrid environment, creating new solutions for the existing issues, without modifying BGP or requiring its complete replacement, working well with gradual or partial deployments.

Centralized control introduces some challenges that must be addressed to avoid new problems. Since it is logically centralized, extra techniques must be applied to ensure high availability and security. By designing a solution using this paradigm it is also necessary to replicate its control servers to guarantee scalability of the solution, else the traffic demand will not be supported.

Centralized control approach is one of the new routing paradigms that might bring innovation and a new way to solve issues that are not addressed yet. It was initially conceived to be applied to intradomain environment, but several works are also extending it to interdomain environment. This is why intradomain mechanisms (like OSPF) are also discussed in this subsection. Centralized control, however, will not solve interdomain routing convergence problem alone, but working together with other approaches described in this section seems very promising.

Chapter 4

DeepBGP Framework and BGP Routing Convergence Time with D-Forecaster

4.1 The DeepBGP framework

To address the issues described in the previous chapter we propose the *D-Forecaster* (Deep Forecaster): a LSTM-based BGP convergence time forecasting model that provides a lightweight routing application that dynamically chooses a MRAI value which leads to a reduced BGP routing convergence time when compared to the absence of MRAI and to the default MRAI value. This model is part of a framework called *DeepBGP* (Deep-Learning BGP) that besides the MRAI forecasting through D-Forecaster should be able to be extended to support other BGP-related applications. The main components of the DeepBGP are illustrated in the figure 4.1. In this chapter we describe three important elements of the DeepBGP framework: the ExaBGP router (inside the SDN-AS environment), the PEERING platform, and the D-Forecaster. The D-Forecaster is also part of the SDN-AS in the DeepBGP framework, but in this chapter we give more emphasis in this component, given its importance in the BGP routing convergence time forecasting.

4.1.1 The SDN-AS and the ExaBGP tool

There are some BGP daemon tools, such as Quagga and BIRD, that are used in several interdomain research projects. However, those tools control by default all the BGP message handling after the session is established with a BGP peer, but we need to give the control of the sending of the update messages to our routing application that should hold the announcements according to the MRAI value chosen. They can be configured to offer control to external applications but it is harder than other tools, such as ExaBGP.

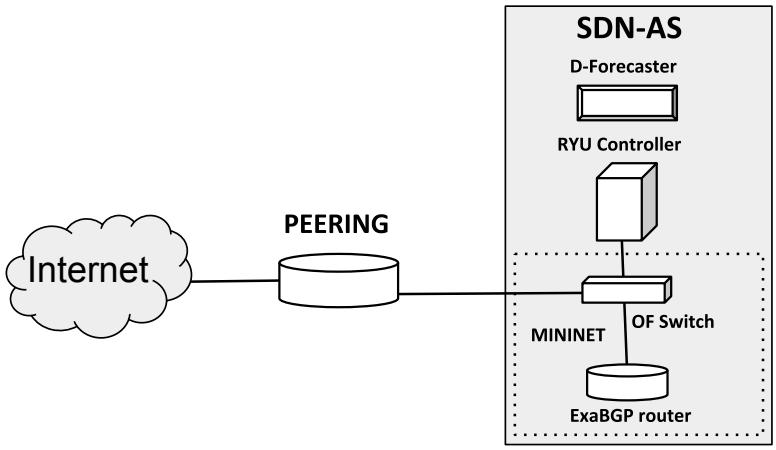


Figure 4.1: The DeepBGP Framework

The ExaBGP tool [181] suits our requirement by providing BGP UPDATE message sending control by external scripts. ExaBGP is a python-based tool that handles all the TCP handshake and the exchange of OPEN, KEEPALIVE and NOTIFICATION messages between BGP peers, giving the user application the control of BGP UPDATEs (announcements and withdrawals). Another advantage of ExaBGP is that it can be easily integrated with SDN applications, by converting BGP messages in plain text or JSON format which can be handled by the applications of administrators and engineers that manage their networks.

In this work, we used Mininet [137], an environment developed to support large scale network emulation in a lightweight manner. Mininet is (a) highly flexible - allowing a wide range of types of topologies and new functionalities; (b) scalable - it is possible to have an environment with hundreds or thousands of switches in one laptop; and (c) realistic - it achieves high confidence results when compared to real networks. In the DeepBGP framework, we use an ExaBGP router as a host in the Mininet environment connected to an OF Switch. The Routing Application running on top of the Controller manages the network and controls when ExaBGP router will send BGP updates to a peer. The ExaBGP router has three interfaces: the first interface is connected to the OF Switch and the seconds is connected to the BGP peer from an external AS. Those interfaces belong to the data plane and are responsible for receiving and sending BGP messages from and to a peer. The third ExaBGP interface is in the control plane because it receives the commands to send UPDATEs directly from the Routing Application in the control plane. We call this AS with ExaBGP modeled with the SDN paradigm as SDN-AS. This is illustrated in figure 4.2.

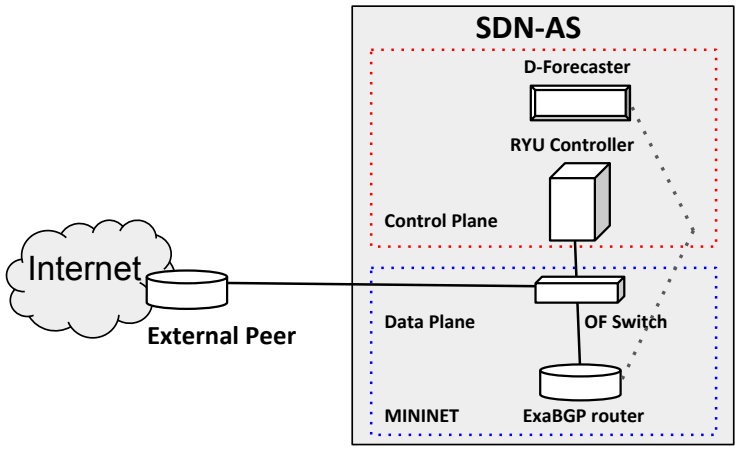


Figure 4.2: The SDN-AS setup

4.1.2 The PEERING platform

Despite the advances in networking innovation and in the way they are used, some known problems in the BGP protocol still open and little changes have been presented in recent years. Since the wide extent of the BGP protocol, most part of interdomain researches are based on passive measurements of past events or are performed within simulated environment, which incurs on limitations in the observed results and in the deployment of new approaches.

To enable researchers to actively conduct experiments where they can directly measure how Internet responds to changes in their experiments, the PEERING (Pairing Emulated Experiments with Real Interdomain Network Gateways) project [182] provides a testbed where researchers can peer their emulated experiments with the real Internet BGP routers. The PEERING platform manages an AS with AS_Number 47065 and owns a set of IP addresses that are allocated to different experiments according to the approved researchers proposals.

To maintain experiments isolated and safe the PEERING testbed require that their routers can only be connected to research experiments through a VPN tunel. To this end, we installed the OpenVPN tool [183] to connect directly to the PEERING router MUXes through the established VPN tunnels. To complete the setup of the PEERING VPN tunel we need to use the certs and keys files provided by the PEERING team after the experiment proposal approval.

The PEERING platform currently provides connection to the Internet through MUXes distributed through eight different universities across the world. Figure 4.3 shows the available PEERING MUXes which can be connected by the researcher experiments through a VPN tunel.

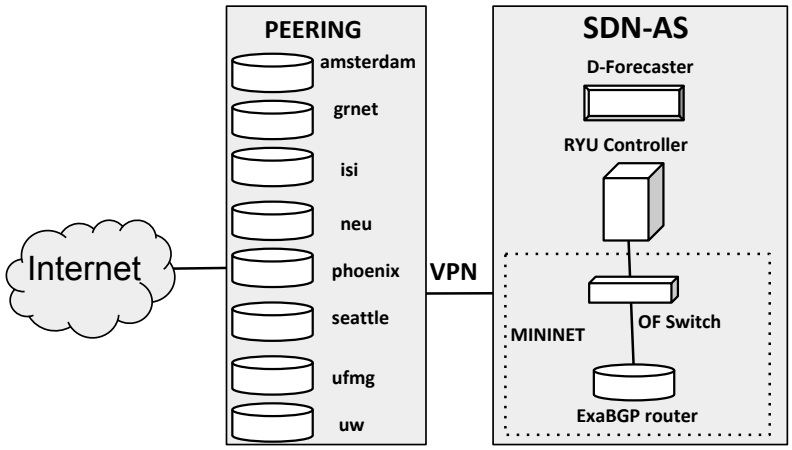


Figure 4.3: Connection between SDN-AS VM and PEERING MUXes

4.1.3 The D-Forecaster

The D-Forecaster plays an important role in the DeepBGP framework. It is deployed as an application in the context of an SDN-AS, as showed in figure 4.2. Before seeking a MRAI value that leads to reduced routing convergence time, the DeepBGP should be able to forecast the convergence time given the collected features.

To evaluate if D-Forecaster is able to forecast BGP routing convergence time, we decided to isolate it from the DeepBGP framework and test it as a standalone routing application. To do that, we use data collected from public BGP data sources and the announcements of Beacon prefixes to measure the routing convergence time.

BGP data collection from public sources

Monitoring BGP is important to understand its dynamics. In order to ease the different research purposes related to BGP events, public projects such as *RIPE RIS* [184] and Oregon's *RouteViews* [185] have been collecting BGP updates and routing tables from several routers around the world for more than a decade. The RIS project has routers that collect and store BGP updates every five minutes and the RouteViews project stores them every fifteen minutes. The update files are stored by both projects in the MRT format. The data collected from these projects has been used by researchers to study, analyze and learn about BGP aspects such as the Internet-growth and churn evolution [100], Internet topology and relationship [186], BGP anomaly detection [187], BGP stability [188], identification of the causes of Internet outages [189] and others.

Although researchers have obtained interesting insights and better understanding on BGP dynamics from passive monitoring of the data from those projects, active measurements

are important to better understanding on the routing events that happens on the Internet. RIPE RIS project provides the RIS Routing Beacons [190], where the routing speakers send announcements and withdrawals at predefined time scheduling. The term *Beacon* refers to a public prefix with global visibility which is announced and withdrawn periodically everyday through the Internet. The RIS Beacon functionality is processed by the RIS collectors, where each one has a specific address scheme. All RIS Beacons present the following schedule of announcement and withdrawals:

- Announcements at: 00:00, 04:00, 08:00, 12:00, 16:00, 20:00 (UTC).
- Withdrawals at: 02:00, 06:00, 10:00, 14:00, 18:00, 22:00 (UTC).

Table 4.1 lists the Beacon Addresses and their prefixes announced and withdrawn [190].

Table 4.1: Routing Beacon Address Scheme

Collector	**Address**
RRC00	84.205.64.0/24
RRC01	84.205.65.0/24
RRC03	84.205.67.0/24
RRC04	84.205.68.0/24
RRC05	84.205.69.0/24
RRC06	84.205.70.0/24
RRC07	84.205.71.0/24
RRC10	84.205.74.0/24
RRC11	84.205.75.0/24
RRC13	84.205.77.0/24
RRC14	84.205.78.0/24
RRC16	84.205.73.0/24

The analysis of the BGP dynamics aspects by only observing data from past events is hard when what triggered the updates is not well known. With a previous knowledge about the source of updates and the time of announcements and withdrawals, it is possible to obtain better insights and more accurate measures of their effects on route propagation and convergence. The particular study of BGP convergence, which we are interested in, can be performed by observing the updates sent by the Beacon routers of the RIS Beacons from the perspective of the routing collectors of both RIS and RouteViews projects.

To measure the convergence time after one announcement or withdrawal of a Beacon prefix daily sent according to the schedule, we choose some routing collectors from RIS and RouteViews projects to obtain the updates of a specific day and observe the propagation of

the updates. The choice was based on the number of peering connections of those routing collectors from which they receive the updates propagated through the Internet.

The files in the updates from both RIS and RouteViews sources are stored in the MRT (*Multi-threaded Routing Toolkit*) format [191]. The format can be used to export messages from routing protocols such as BGP. The first field from the MRT common header is the timestamp, which express in seconds the time the packets are stored by the router collectors. The timestamps in MRT formats are expressed in the Coordinated Universal Time (UTC), which standardize all timestamps.

In this work we get the update records from the RIPE RIS RRC00 collector, that currently receives updates from more than 50 peers, and group the collected updates for the beacon prefix according to the update event. For example, after downloading the updates from a given day, consider that we want to measure the convergence time for the announcement of beacon prefix 84.205.64.024 sent at 00:00:00 (UTC). We developed scripts that search for the beacon prefix in the updates of that day and sort the updates that match the search by their timestamps.

Each beacon announcement and withdrawal leads to a convergence event: six *Tup* events, where a previously unavailable route to the beacon prefix is announced, and six *Tdown* event, where a previously available route is withdrawn and becomes unavailable. Each RIS Beacon *Tup* event is always followed by a *Tdown* event of the same prefix two hours later.

We use a terminology for BGP update propagation similar to the one described by Mao et al. [9], where *input signals* refer to an update generated at a Beacon, and the *output signals* refer to the updates generated at different locations and generated by an input signal. Considering this terminology, the input signals are the Beacon announcements and withdrawals in the predetermined schedule and the output signals are the updates received from several nodes to the same prefix triggered by the input signal.

Mao et al. also defines terminologies for two types of convergence. The *relative convergence time* is the time between the first update that arrives at a route collector from any peer and the last update from a specific peer. For example, the relative convergence for collector RCC00 related to peer 45.61.0.85 is the time between the first update received at RCC00 by any of each peers and the last update received by peer 45.61.0.85. Another concept is the *end-to-end* convergence, where they consider the time since the input signal sending based on beacon timestamp and last update received for a specific peer. For example,the end-to-end convergence for collector RCC00 related to peer 45.61.0.85 is the time between the beacon prefix announcement or withdrawal timestamp and and the last update received by peer 45.61.0.85. A simple method to determine when a converged state is reached is to define a timeout such that when the arrival time difference between two subsequent updates from

some peer is higher then the timeout, the converged state is reached. It is important because sometimes noise is generated for situations such as BGP session reset between routers.

We extend Mao et al. definitions of convergence time and define the *source convergence time*, where the convergence time for a node is given by the elapsed time between the beacon announcement/withdrawal timestamp and the last update timestamp related to the beacon prefix received by any of source node peers. The *source convergence time* can be described as an instance of the definition of *converged state*, presented in section 2, but instead of considering a node reaches its convergence state only when its RIB does not change for any prefix, we consider that its converged state is reached when its RIB remains unchanged for the beacon prefix until the next beacon event is triggered. Figure 4.4 illustrates the relative, the end-to-end, and the source convergence time, which we are interested in.

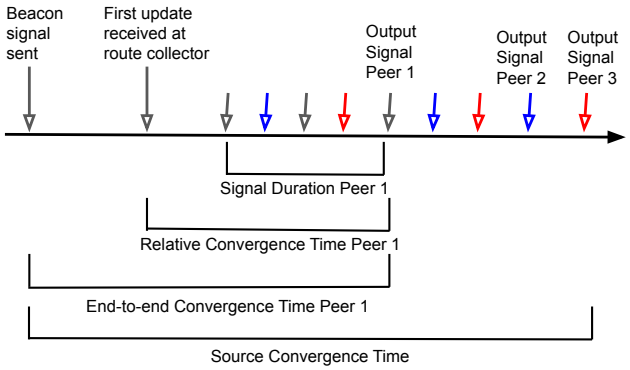
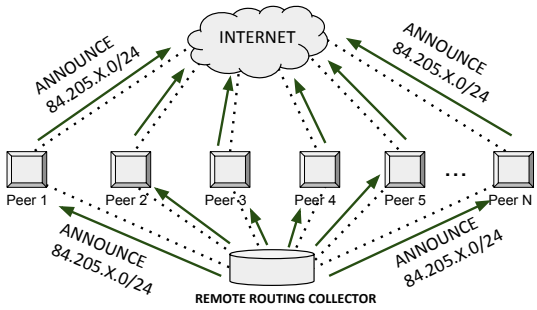


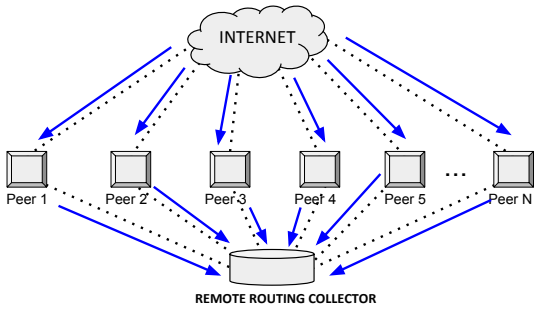
Figure 4.4: Instances of convergence time [Adapted from [9]]

Figure 4.5 illustrate the process we used to calculate the source convergence time. We divide this process in two parts. First, the beacon prefix (84.205.X.0/24, where the value of X changes for each RRC) is announced according to the Beacon project prefix schedule. Each RRC has BGP sessions with a varying number of peers from which UPDATEs are received and sent. Then, those peers send the announcements to other peers, until all the BGP routers in the Internet that receive the announce achieve a stable state, where no more UPDATEs triggered by the first beacon prefix announcement are received by those routers. This is illustrated in Figure 4.5(a). Second, we download the UPDATEs related to the beacon prefix from the RRC from which the announcement was sent. To calculate the convergence time from the collected UPDATEs, we take the time difference between the timestamp of the Beacon input signal and the timestamp of the last update in the output signal related to that prefix. Figure 4.5(b) shows the updates output signals related to the beacon prefix that arrive at the remote routing collector.

This process can help in the evaluation of proposed tools or mechanisms related to the BGP routing convergence and stability, and they are open sourced at [29].



(a) Beacon prefix announcement



(b) UPDATEs received by collector from peers

Figure 4.5: Beacon prefix announcement and source convergence time

A model to learn BGP convergence patterns

With the growth in computing power, enormous amounts of data becoming available [192], and the progress in machine learning research , creative solutions have been found to a lot of problems that we could not address or did not have good solutions for.

In the research of BGP dynamics we can obtain a large volume of data from the updates stored by RIS and RouteViews project. The Internet routing table currently has more than 700,000 prefixes. The volume of available data provides us the possibility to use machine learning algorithms to find patterns in them and propose new solutions to existing problems on the Internet. For example, learning-based techniques have been applied to those data to detect and classify BGP anomalies [187], [45], [47], [193].

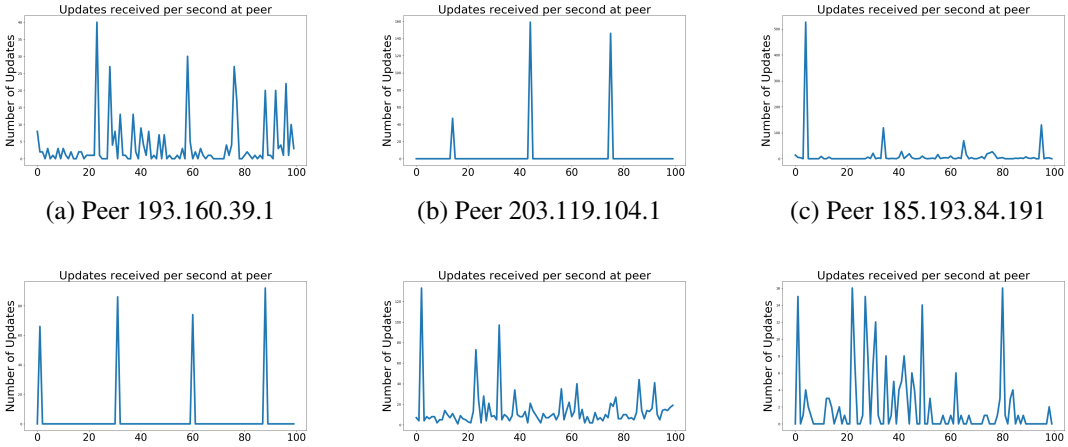
However, as explained by [2] in their work to detect BGP anomaly, most solutions apply classic learning mechanisms to make real-time prediction based on features obtained from traffic only of the present moment. Due to the occurrence the UPDATE burst happens and the noise found in the BGP dynamic traffic, short-term features are probably not the best choice. Using an sliding time-window, we can create a model to learn patterns from historical data sources by considering the Internet traffic as a multi-dimensional time sequence.

Motivated by the advances in computing power, in learning algorithms, and in the volume of available data from BGP public sources, our goal is to use those historical data from BGP updates to learn patterns in traffic and to predict the convergence time in a near future. This is an opportunity to improve the convergence of the BGP protocol.

By predicting the convergence time using a LSTM model we are solving a problem of time series prediction. For the model to achieve a reasonable prediction it is necessary to find patterns in the time series data. We can collect and extract information from data of routing collectors in a time period just before the announcement or withdrawal for a beacon prefix was sent.

We take the collector RRC00 as our source of observation and analysis. We start by collecting the total number of updates received by each of RRC00 peers during the last 300 seconds before beacon prefix announcement event. We tested other values, such as 60 and 100, but we do not choose a shorter time series value because it could loose longer dependencies and ignore important patterns.

We observed that each peer of RC00 presents a different behavior based on the total number of updates to whatever destination received from its neighbors. To illustrate that, figure 4.6 shows the plots of the number of updates received in the last 100 seconds from 6 peers of RC00.



(a) Peer 193.160.39.1 (b) Peer 203.119.104.1 (c) Peer 185.193.84.191

(d) Peer 208.51.134.248 (e) Peer 212.25.27.44 (f) Peer 213.200.87.254

Figure 4.6: Number of received updates (Y axis) in the last seconds (X axis) before beacon announcement from some of RRC00 peers

As shown in figure 4.6, some peers present a very well-defined behavior while others are more dynamic and harder to predict. For example, peers 203.119.104.1 and 208.51.134.248 (figure 4.6 (b) and 4.6 (d), respectively) are deploying MRAI with default value 30, since

during a intervals of 30 seconds no updates are received, and after the 30 seconds-interval several updates are detected.

The behavior of those peers can make pattern detection easier and the use of features such as *total number of received announcements* and total number of received withdrawals is enough to learn and predict the time the updates related to a beacon prefix will arrive at the collector after an announcement event is triggered. However, only the features *total number of announcements* and *total number of withdrawals* are not enough to learn the relative convergence time for all RC00 peers. For example, peers 193.160.39.1 and 213.200.87.254 (figure 4.6 (a) and 4.6 (f), respectively) do not present patterns that can be easier learned by the described two features, since they probably do not use a mechanism to make updates time arrival more predictable, such as the MRAI timer.

We made an extensive literature review on the research of BGP convergence time [23], and based on this study we defined the following features to be collected from updates shown in Table 4.2,in which Num_Announce stands for "Number of Announcements".

1. Total Num_Announce (from all dest.)	2. Number of Withdrawals
3. Number of duplicate Announcements	4. Longest AS_PATH length
5. Shortest AS_PATH length	6. Average AS_PATH length
7. Number of Prepended ASes per AS_PATH	8. Number of *TShort* events
9. Number of *TLong* events	10. Num_Announce from neighbor AS 1
11. Num_Announce from neighbor AS 2	12. Num_Announce from neighbor AS 3
13. Average Edit Distance	14. Maximum Edit Distance
15. Number of Edited paths	

Table 4.2: Learning Features for BGP convergence predictor

All features are extracted from BGP updates. (1) and (2) are volume features that are important to learn the relative convergence time patterns for peers that are MRAI-constrained. Additional features are implemented to make our model to learn more complex behavior. Feature (3) collects the number of duplicate announcements that happens when two or more routes have the same AS_PATH but different attributes, which can indicate some policy change. Prepended ASes, a technique where one AS can append its own AS number multiple times in the AS_PATH of an announced route, is used to make the route less preferred than other, also due to policy preferences, and it is collected by feature (7). Features (4), (5) and (6) capture patterns in the AS_PATH lengths to a given peer. Features (8) and (9) are important to learn a pattern in *TShort* and *TLong* events because a deviation on the learned behavior can indicate the occurrence of sessions resets or session re-establishment, which can on its turn impact on the relative convergence after beacon event. We observed by analysing the AS_PATHs from routes received by each peer that all of them received

those updates from at least three other neighbor ASes. It is valid only for multihomed ASes; otherwise, in case of stub ASes, those features should be zeroed or unused by the model. Then, we established for each peer a ranking on the three neighbor ASes which send the higher number of announcements, and implemented it as features (10), (11), (12). The edit distance, collected in features (13), (14), and (15), is a metric applied to AS_PATH attributes between two routes, where it computes the minimum number of insertions, deletions, and substitutions needed to modify one AS_PATH to match the other AS_PATH. For example, the edit distance between AS_PATH [3549 3356] and AS_PATH [3549 8697 8376] is two because the operations one insertion and one substitution are required to match the two AS_PATHs [47].

The characterization of the experiments for beacon prefix convergence time forecasting, the methodology description, the data and the results from experiments are described in more details as follows.

Datasets generation and Experiments Methodology

We collected statistics from the described features in intervals of 1 second. We decided to keep this granularity of 1 second interval for feature collection to ensure our predictor mechanism can detect patterns in peers that deploy MRAI with default value of 30 seconds and also in peers that do not.

After downloading data from RIS and RouteViews from five months of 2018, we developed some scripts available at [29] to extract the features from the raw data, following the steps described in Section 4.1.3. Then, we used the timestamps of each collected update to identify the ones related to the beacon prefix that happens in the time series of 300 seconds before the beacon announcement event with each time step lasting 1 second.

Considering X as an array of all the 15 features listed in the previous section and t as the time where the beacon event is triggered, the array of features X is collected at each time step. For each beacon event we collect an array Y [ty 1, ty 2, ty 3] by each peer representing 3 timestamps for updates that arrived after the beacon prefix announcement. We chose 3 timestamps because we observed that most of the collector peers converge with at most 3 timestamps. However, this brings an important limitation to our model, since when updates arrive with more than 3 timestamps the convergence time forecasting will be affected. The collection of the 300 X arrays for each time series compose one input instance and the correspondent array Y compose one target instance for the model. This process is illustrated in figure 4.7.

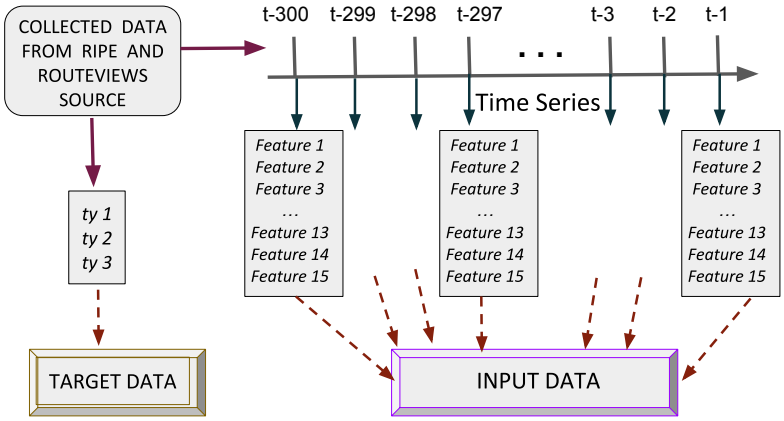


Figure 4.7: Data collection methodology

This is repeated for all beacon events from every day from June 01, 2018 to October 31, 2018, creating the datasets used to train and evaluate the BGP convergence predictor model. The general dataset structure is represented as a matrix in Table 4.3.

Each row of the matrix represents one beacon event i, where each time series x is the input to the learning mechanism and the values of y represent the array output of length 3 of the timestamps the updates related to beacon prefix arrive after the triggered beacon event.

Table 4.3: Dataset matrix representation

event	**t-300**	**t-299**	**...**	**t-1**	**t+y_1**	**t+y_2**	**t+y_3**
i_1	$x_{1,1}$	$x_{1,2}$	...	$x_{1,300}$	$y_{1,1}$	$y_{1,2}$	$y_{1,3}$
i_2	$x_{2,1}$	$x_{2,2}$	...	$x_{2,300}$	$y_{2,1}$	$y_{2,2}$	$y_{2,3}$
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
i_m	$x_{m,1}$	$x_{m,2}$	...	$x_{m,300}$	$y_{m,1}$	$y_{m,2}$	$y_{m,3}$

The LSTM learning model is the core element to the convergence predictor mechanism, since it receives the data statistics in the time series format seconds before the beacon prefix announcement and uses it to learn the relative convergence patterns from all collector peers. The model should be adapted to be used to calculate the convergence time for prefixes other than the beacons.

But there are a number of hyperparameter settings that should be considered when implementing the learning mechanism. The Keras framework [194] requires the input data of the model to be reshaped to a 3D vector of the shape (S x t x f), where S = *Number of Samples*, t = *Number of Time Steps* and f = *Number of Features*. Two important parameters

to choose when implementing a LSTM network is the number of layers and the number of cells. We tested our model with different values for each parameter and chose the ones that led the model to perform better. This is shown when we detail the experiment results. The LSTM model was tested with 1 and 2 hidden layers, and with 16, 32, and 64 cells in each layer.

Besides LSTM hidden layer(s), we tested adding 1 and 2 dense layers to the model. The added dense layer is a fully connected layer that follows the LSTM hidden layer(s) and has 3 outputs (each output represents the timestamps related to the updates related to the beacon event that arrive from peer after prefix announcement).

We added a dropout layer to reduce overfitting. Dropout is a technique that randomly sets some cells or neurons weights to 0. We tested dropout values 0.2 and 0.3 and deployed a different number of epochs for training, ranging from 200 to 600 epochs.For each layer we should also choose an activation function, which is important to make the hypothesis space of each layer richer than just with linear operations, taking the benefits from deep and non-linear representations. We experimented two popular activation functions: *sigmoid* and *relu*. Finally, we should define a loss function and an optmizer for our model. Since we are dealing with a regression problem, our model uses the *mse* (mean squared error) as its loss function [56]. We used the optimizer Adam because of its high performance when compared to other optimizers [195]. Since each collector peer presents its own dynamic behavior, each peer has its own LSTM model. Then, the train and test dataset are splitted to be redirected to the appropriate model. This is depicted in figure 4.8.

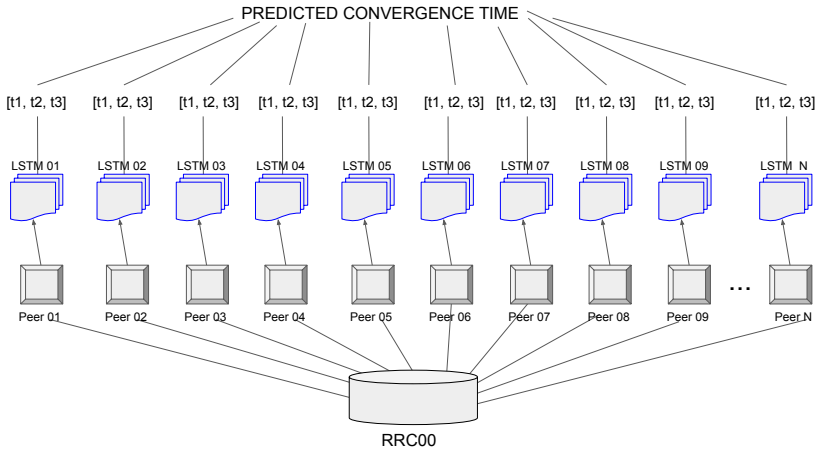


Figure 4.8: The BGP Convergence Predictor framework

We implemented our solution in Python using the open-source Keras deep learning library [194] running on top of [196], a library used for numerical computation using data flow graphs. Our models were trained using two GPUs GTX1070TI with 96GB RAM.

Results and Analysis

After processing the collected data used to compose the data sets we detected the occurrence of noise, which leads to *outliers* for our learning models. The noise can be characterized as an event that presents a very long output signal, i.e., a source convergence time much higher than for other events for a given beacon prefix. It can be caused by the *Route Flap Damping (RFD)*, a mechanism that can be deployed at BGP routers to suppress unstable routes. A single router reboot can cause routes to be suppressed by a long period of time and make source convergence time much harder to predict.

We also observed that some RRC00 peers were inactive at some periods of the data set and re-established BGP sessions with the collector, causing a high volume of updates that can also generate noise. To filter those noises we used a simple heuristic described by [9]: if the difference between two consecutive updates received from the same peer is above an empirically determined threshold, the event is considered an *outlier* to the model and is removed from the data set. We observed that a threshold value of 90 seconds is enough to identify and filter noisy data. One example of noise in source convergence target data is illustrated in figure 4.9. In this figure we observe a series of predictions (in red) and target values for source convergence time (in blue) and one unique event has a convergence above 7000 seconds, which is much higher than a normal source convergence time from RCC00.

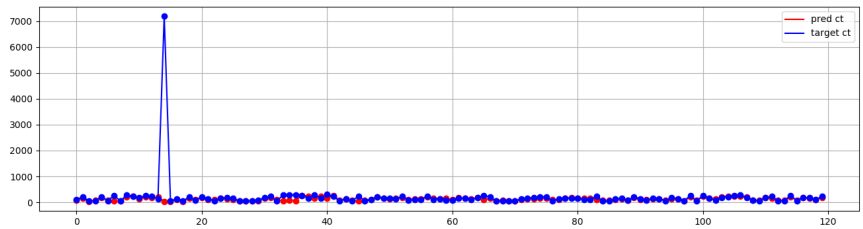


Figure 4.9: BGP Convergence prediction with noise

After collecting and preparing our data we separated it in train and test data set. With the updates collected from 5 months we created 3 pairs of train-test sets:

1. **Train Set 01**: Months 06, 07, 09, 10 / **Test Set**: Month 08

2. **Train Set 02**: Months 06, 07, 08, 10 / **Test Set**: Month 09

3. **Train Set 03**: Months 06, 07, 08, 09 / **Test Set**: Month 10

After cleaning data by filtering noise we needed to choose a metric to evaluate our model. A regularly employed metric for regression problems is *Root Mean Squared error* (RMSE). RMSE is used to calculate the model error based on the standard deviation between predicted and target source convergence times, and it can be calculated as follows:

$$\sqrt{\left(\frac{\Sigma(y-\hat{y})^2}{n}\right)}$$

Where y is the actual target value, $\hat{y}$ is the predicted target value, and n is the number of samples.

The computation of RMSE applied to all output signals from RC00 neighbors is interesting to observe how the models associated to each neighbor contribute to the mean error between predicted and target values. However, since the goal of our predictor mechanism is to predict the BGP source convergence time, we decided to apply the metric only to the computed elapsed time between the last output signal from all peers and the beacon timestamp. Besides that, RMSE alone is not enough to analyze how good our model prediction is when compared to the target. We look for a metric that provides a standardized measure of error by giving us the percentage of the error between the predicted and the target value, which can help better explain the model accuracy than the RMSE metric does. A metric that meets these requirements is the *Mean Absolute Percentage Error* (MAPE), which presents a suitability for forecasting applications, becoming frequently used as a quality measure for regression models [197]. Then, we use MAPE to evaluate our models and obtain their accuracy from this metric. MAPE can be calculated as follows:

$$\left(\frac{1}{n}\Sigma\frac{|y-\hat{y}|}{|y|}\right)100$$

Where y is the actual target value, $\hat{y}$ is the predicted target value, and n is the number of samples. The model accuracy is calculated by subtracting the MAPE value from 100. MAPE must not be used when one of the target values is 0, since it would make MAPE undefined. This is not the case for our experiment, where all target values are always positive.

To evaluate the features relevance for the model, we grouped the features in 5 sets and trained the main model with all features and 5 additional models with the selected sets of features removed. The sets of features are selected and grouped as follows:

- Set 1: Number of announcements, Number of withdrawals, Number of duplicate announcements.
- Set 2: Longest AS_PATH length, Shortest AS_PATH length, Average AS_PATH length, Number of prepended ASes.
- Set 3: Number of *TShort* events, Number of *TLong* events.
- Set 4: Number of announcements from neighbor AS 1, Number of announcements from neighbor AS 2, Number of announcements from neighbor AS 3.
- Set 5: Average edit distance, Maximum edit distance, Number of edit paths.

Table 4.4 presents the performance of the models with different sets of features using Train Set 01 and Month 08 Test Set. We observe in Table 4.4 that the model achieves its higher accuracy for month 08 test set when all features are used to train it. MAPE metric becomes higher when features set 1 is removed from the model, showing that the number of announcements, the number of withdrawals and the number of duplicate announcements are the most important set of features for the train-test data sets.

Table 4.4: Performance of Model with Train Set 01 and Test Set 08-2018

Model	**Feature Set**	**MAPE**	**Accuracy**
01	Set 1	28.24%	71.76%
02	Set 2	23.07%	76.93%
03	Set 3	25.31%	74.69%
04	Set 4	23.55%	76.45%
05	Set 5	23.20%	76.80%
06	All features	22.34%	77.66%

Figure 4.10 shows the source convergence times for test set 08-2018 and train set 01 with all features, where the target values are plotted in blue and the prediction values are plotted in red.

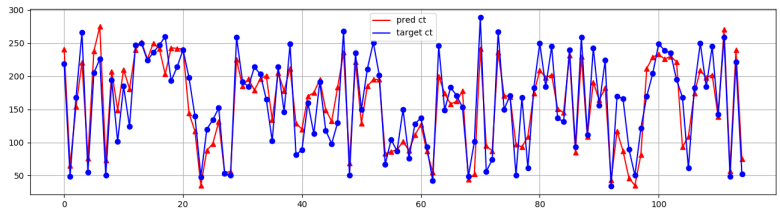


Figure 4.10: BGP Convergence prediction- Month 08

Table 4.5 presents the performance of the models with different sets of features using Train Set 02 and Month 09 Test Set. We observe from the results presented in the table that model trained after removing features set 2 achieves highest accuracy of above 80%, in contrast to the accuracy of 78.19% when all features are used to train and test the model. A very similar accuracy is achieved when features sets 1 and 3 are removed from models. It shows that sets 1, 2, and 3 presents a high correlation for the training set 02, indicating that when used together they can make the model accuracy worse. In contrast, features set 5 presents the highest MAPE, highlighting the importance of the edit distance features for the trained model.

Table 4.5: Performance of Model with training set 02 and Test Set 09-2018

Test Set	**Feature Set**	**MAPE**	**Accuracy**
09-2018	Set 1	20.52%	79.48%
09-2018	Set 2	19.87%	80.13%
09-2018	Set 3	20.56%	79.44%
09-2018	Set 4	21.36%	78.64%
09-2018	Set 5	24.76%	75.24%
09-2018	All features	21.81%	78.19%

Figure 4.11 shows the source convergence times for test set 09-2018 and train set 02 with all features, where the target values are plotted in blue and the prediction values are plotted in red.

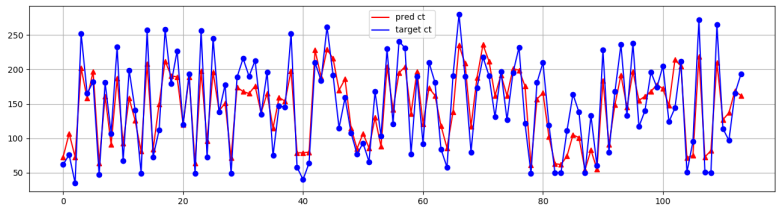


Figure 4.11: BGP Convergence prediction- Month 09

In Table 4.6 the worst accuracy is presented for model trained and tested with set 2, where features related to AS_PATH length and prepended ASes are removed from the model, achieving 72.40% accuracy. The removal of other sets of features presented a similar accuracy (76%), which still worse than when all features are used to train and test the model (77.5%).

Figure 4.12 shows the source convergence times for test set 10-2018 and train set 03 with all features, where the target values are plotted in blue and the prediction values are plotted in red.

Table 4.6: Performance of Model with training set 03 and Test Set 10-2018

Test Set	Feature Set	MAPE	Accuracy
10-2018	Set 1	23.08%	76.92%
10-2018	Set 2	27.60%	72.40%
10-2018	Set 3	23.71%	76.29%
10-2018	Set 4	23.92%	76.08%
10-2018	Set 5	24.01%	75.99%
10-2018	All features	22.50%	77.5%

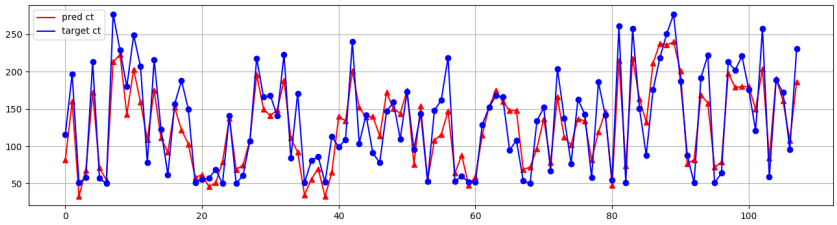


Figure 4.12: BGP Convergence prediction- Month 10

From the plots of figures 4.10, 4.11, and 4.12, we can observe that the implemented models are able to predict the BGP source convergence time for announcements and can offer researchers and network operators useful insights on the BGP dynamic behavior. However, we also observe that for some events there is still significant difference between the predicted and the target data points, which is quantified in MAPE values described in the tables for each test. One important factor that made the accuracy variation on each test is that some peers were inactive during some months used in the training set in some experiments and in test set in others, i.e., no sessions were established and hence no updates were received during those periods of time.

4.2 Chapter Summary

In this chapter, the main elements of the DeepBGP framework and its elements were presented. The D-Forecaster application is a key element, since it forecasts the convergence time for a given prefix and is used by the other elements of the DeepBGP to choose the MRAI value accordingly.

The experiments with D-Forecaster were made separated from the SDN-AS context. Despite the good obtained results, some limitations were observed and modifications are necessary to use it in the DeepBGP adaptive MRAI case.

1. Three Output labels for each peer: It was initially used to learn the next timestamps the updates would arrive based on the time the updates arrived previously. When more than 3 updates arrive after an announcement, the D-Forecaster will make wrong forecasting. It should be modified to compute a single output, which should compute the convergence time.

2. Another limitation is that the convergence time was calculated based on the updates that arrived solely on the source node originating the announcements. It is changed in D-Forecaster used in DeepBGP to MRAI forecasting.

3. The accuracy is influenced by the patterns in updates received by peers from the beacon prefix announcement originating collector. It is not possible in many situations, for example, with the PEERING experiments, where the only peer is the router MUX that connects our experiment to the Internet.

Given the D-Forecaster standalone implementation observed limitations, some modifications were made to it in order to use it to give the DeepBGP framework information on routing convergence time and enable it to forecast the MRAI value that should lead to reduced BGP routing convergence time. Those modifications and the experiments with the adaptive MRAI solution performed by DeepBGP are detailed in the next chapter.

Chapter 5

Improving BGP Routing Convergence Time with Adaptive MRAI

In this chapter, the DeepBGP framework is described in more details and the D-Forecaster is modified to perform real-time experiments with the PEERING platform. The framework is tested using static MRAIs against adaptive MRAI values. To test the adaptive MRAI with learning models other than LSTM, we also implemented the D-Forecaster as Decision Trees and SVR, two classical machine learning models for regression problems.

The machine learning methods SVR and DT were implemented and tested using the python Sklearn library. We first give some details on the implementation decision; then, the experiment design and methodology are described. Finally, the experiment results are presented.

5.1 Implementation details

In this section we describe details of the D-Forecaster implementation. We divide its implementation into three parts: the first part deals with the installation of rules at OF Switches to establish a BGP session with the MUXes from the PEERING platform and exchange BGP messages with them and a parser; the second part handles the communication with the ExaBGP node; and the third part implements the time-series learning model using LSTM networks.

5.1.1 Rules installation and parser

The D-Forecaster and all the scripts that communicate to it are implemented in the Python programming language and run on top of the Ryu Controller.

The D-Forecaster running with the Ryu controller [198] in the control plane is able to install rules at the OF switches in the data plane to forward all packets with IP source and destination equal to the PEERING MUX IP. The first two rules installed at the OF Switch inside the SDN-AS are responsible to send the whole packet content from and for the TCP port used by BGP (179) and which match the connected MUX IP to the Ryu Controller and to the D-Forecaster. The other rule installed at the OF switches make all the packets other than the packets with BGP content to be forwarded directly without sending to the Controller.

After connecting to a MUX, the D-Forecaster needs to extract information from received packets with the BGP Updates. To perform this a packet sniffer method was implemented in the D-Forecaster to parse all packets with BGP related content, considering the protocols that carry the BGP data: first parsing the MAC header, then the IP header and finally the TCP header and its data. The BGP content is actually the TCP data and is the bigger part of our parser. When receiving the full table from MUX several packets arrive at the SDN-AS containing many UPDATEs. The last UPDATEs inside each packet usually are truncated since the packet limit was reached. Besides that, despite BGP comes on top of TCP, some packets arrive out of order and our parser use the TCP sequence to handle it. The D-Forecaster parser deals with those situations and ensures all information received from BGP UPDATEs arriving from peering MUXes are extracted and stored correctly.

5.1.2 ExaBGP configuration script

The announcement control must be delivered to the D-Forecaster, which is the control, the intelligence. The D-Forecaster decides when and how the announcement should be. The ExaBGP serves the application and sends the announcement according to required by the Routing App. However, the Routing and the BGP speaker node are in different planes and have different addresses. In other terms, they are in different namespaces. This can be done by making a call to a controlling application from the exabgp configuration file, which on its turn is called from the Mininet script:

The controllingApp.py tries to establish a socket connection with an application to receive a message and when the print command is applied the message is interpreted and executed by the ExaBGP node. The controllingApp.py is the server side, while the client side is in the D-Forecaster, which can send the BGP announcements. To achieve this goal the BGP speaker running ExaBGP needs to be able to communicate to the Routing App. A solution to this was to add a new interface to the BGP speaker and connect it to a node in the root namespace, where the controller and applications are placed. It makes the BGP speaker to be present in both the control and data planes, where its data plane presence is possible by its interface connected to the OpenFlow switch and through where it receive the BGP packets

from PEERING muxes. Its control plane presence is possible by its second interface which is connected to a root node that access the D-Forecaster and enable it to receive commands to announce routes

5.1.3 The learning model implementation

Since the model described in this chapter is based on the model used in BGP convergence predictor, most of the features from Table 5.1 are already described on Table 4.2 as relevant for learning purposes and shall be the ones to be extracted from BGP updates.

1. Total Num_Announce	2. Number of Withdrawals
3. Number of duplicate Announcements	4. Longest AS_PATH length
5. Shortest AS_PATH length	6. Average AS_PATH length
7. Number of Prepended ASes per AS_PATH	8. Number of *TShort* events
9. Number of *TLong* events	10. Num_Announce from neighbor AS 1
11. Num_Announce from neighbor AS 2	12. Num_Announce from neighbor AS 3
13. Average Edit Distance	14. Maximum Edit Distance
15. Number of Edited paths	16. MRAI value

Table 5.1: Learning Features for BGP convergence predictor

The new feature to the model is the feature 16, which is the MRAI value that is applied on each event. The set of values are (1,5,15,30) seconds. We chose to use those values to train and test the models for the following reasons:

- MRAI=1: Absence of MRAI. Used to measure the convergence time when MRAI is not deployed. We tried use MRAI=0, but ExaBGP ignores the first update and sends only the second
- MRAI=5: This value of MRAI is usually deployed for BGP sessions between peers inside the same AS
- MRAI=15: Half of the standard MRAI, this value is used some related work, such as the FLD-MRAI [152]
- MRAI=30: The standard MRAI value deployed on most BGP router implementations

5.2 Experiment Design and Methodology

In this section, the design and the methodology for the experiments are presented. There is a number of papers that show the impact of faildown events in the BGP convergence time.

Faildown events cause *Tdown* events, where a previously available route becomes unavailable. When *Tdown* happens, most part of updates sent to neighbors are withdrawals, which are currently not constrained by MRAI timer. Since this work focus on the impact of MRAI in the BGP routing convergence time, *Tdown* events are not considered in the experiments.

5.2.1 Experiment Design

Bremler-Barr et al. [10] observed that the distributed nature of BGP leads to the problem of path exploration due to race conditions. The race condition situation happens when a router receive an announcement in the inverse preference order, i.e., longer AS_PATHs arrive first than shorter AS_PATHs). Then, the less preferred paths are propagated before the more preferred updates. This is caused by links with variable delays between the source and destination. The race condition is an interesting problem to evaluate how MRAI impacts the BGP convergence time. To generate race conditions, we create a scenario with variable link delays (ranging from 100ms to 400ms) and connects this topology to the SDN_AS through the OF Switch. Figure 5.1 shows one of the scenarios used to generate the race conditions in our experiments. We consider each router represents one AS. The triangle topology illustrated in figure 5.1 is the base topology to other topologies derived from this one and used in our experiments.

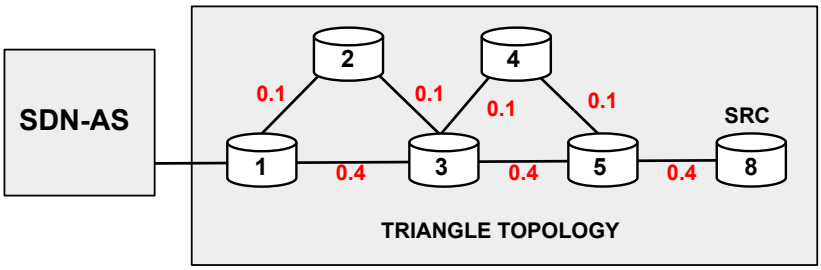


Figure 5.1: Custom scenario used to generate the *Tup-Tlong* events due to race condition. Based on the topology used by Bremler-Barr et al. [10]

The last BGP router from this topology is connected to an OF Switch that is linked to the SDN-AS. In figure 5.1, the router 1 is the router that receives UPDATEs form other routers in the triangle topology and deliver them to the BGP router in the SDN-AS, which on its turn sends the UPDATEs to the PEERING MUX, accordingly to the architecture illustrated in

figure 4.3. We call the triangle topology and the derived ones as the custom topologies from the DeepBGP framework. In this example, considering that the BGP preferred-path policy is the shortest AS_PATH, and that the source node is AS 8, the *AS_PATH1* [8 5 3 1] should be preferred over *AS_PATH2* [8 5 4 3 2 1], since *AS_PATH1* is shorter. However, the total link delay for *AS_PATH1* is higher than for *AS_PATH2*. It makes the AS 1 announce the less-preferred *AS_PATH2* before the most-preferred *AS_PATH1*.

Some PEERING MUXes present limited connectivity. Since our experiment measures the effects of MRAI on the BGP convergence, we chose to send announcements through MUXes which receives the full RIB from the Internet, which currently has more than 750,000 prefixes. Attending to this requirement, we chose the MUXes *neu* and *seattle* to establish peering sessions with the BGP router at the SDN-AS. We consider that MRAI is implemented per-peer, which is the best option, since MRAI per-destination can generate extra overhead at routers due to the large number of prefixes in the Internet, with currently more than 750,000 prefixes.

Since the UPDATE announcements from the custom topologies that arrive at the BGP router in the SDN-AS use private AS_Numbers, we need to convert those AS_Numbers to real ones. Once the announcements send to the Internet through the MUXes need to use the PEERING testbed AS_Number, we use a technique called *prepending* to exchange each private AS_Number in the AS_PATH from the custom topology by the PEERING AS_Number 47065. For example, the *AS_PATH* [8 5 3 1] is modified to *AS_PATH* [47065 47065 47065 47065].

The experiments for the DeepBGP framework are designed with the elements presented in figure 5.2, where different custom topologies generate announcements at each experiment that are sent to the SDN-AS, which establishes peering connection with a PEERING MUX, that sends the announcement through the Internet.

The process to generate the dataset and to test the adaptive MRAI application is described as follows:

5.2.2 Dataset generation methodology

Since our mechanism is, to the best of our knowledge, the first to deploy a learning mechanism such as LSTM to predict BGP routing convergence time, there is not a dataset available to train and test our model. It means that we need to generate the dataset. This is not a fast process, because when the connection between when the BGP session is established, the ExaBGP router receives the full RIB from the PEERING MUX router, which can last from 10 to 15 minutes. During this process, a large number of packets and UPDATEs arrive at the ExaBGP router. We are not interested in those UPDATEs, so we do not collect the features

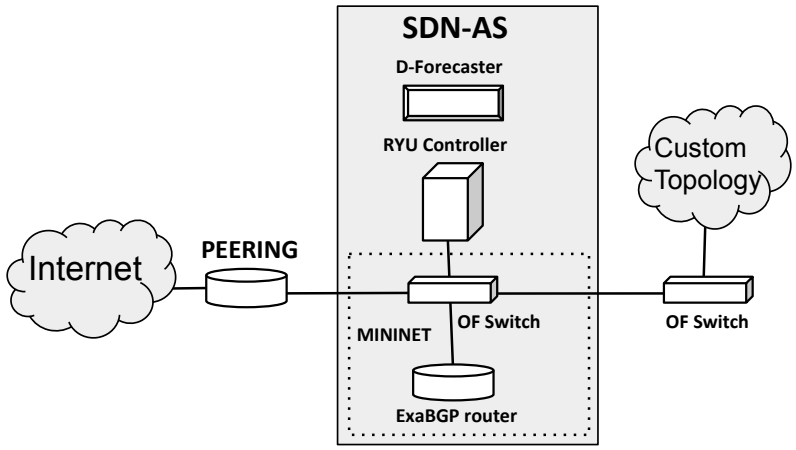


Figure 5.2: The DeepBGP framework setup

during this time period. After receiving an End-of-RIB message, the feature collector module at the D-Forecaster starts to collect the features after this amount of processed prefixes. After 60 timesteps of feature collection the announcement is sent from the custom topology source node. Two UPDATEs arrive from the custom topology: the first is sent imediately; the second UPDATE is holded by the MRAI value set for that experiment. Before sending a withdrawal to restart the process with a different MRAI value, we need to wait a time enough to the network converge. We empirically observe that 30 minutes is a safe time to ensure the network converges to the PEERING announcement.

To calculate the convergence time, the UPDATEs with the PEERING prefixes announced are downloaded from different RIPE RIS collectors. A script was developed to sort the UPDATEs from the RIS collectors and to calculate the convergence time in seconds. The calculated convergence time is used to label the dataset with the collected features. An instance of the output from the calculated convergence time is illustrated in figure 5.3.

The steps followed to generate the dataset for training the learning model are described next, followed by figure 5.4 that illustrates the methodology and the components involved in the generation of the dataset instances. Each numbered box in the figure represents a step in the dataset generation process.

1. A connection is established between the PEERING MUX and the ExaBGP router from the VM where the D-Forecaster runs

2. ExaBGP receives the full RIB from its PEERING MUX, until an End-of-RIB arrives

3. After receiving the full RIB, periodic updates are received from the PEERING MUX and the D-Forecaster starts to collect the features in time series format

```
15:13:34 58057 174 156 47065 47065 47065 47065
15:13:34 64271 3214 3356 156 47065 47065 47065 47065
15:13:35 35619 58057 174 156 47065 47065 47065 47065
15:13:35 3549 3356 156 47065 47065 47065 47065
15:13:37 49432 48362 1764 3356 156 47065 47065 47065 47065
15:13:37 50763 8943 2914 3356 156 47065 47065 47065 47065
15:13:41 6894 174 156 47065 47065 47065 47065
15:13:48 8607 3356 156 47065 47065 47065 47065
15:14:05 47422 58057 174 156 47065 47065 47065 47065
15:14:12 55720 2914 3356 156 47065 47065 47065 47065
15:14:40 60474 58057 174 156 47065 47065 47065 47065
15:15:05 50300 3356 156 47065 47065 47065 47065
15:15:30 328474 328474 328474 328474 328474 328474 37271 3356 156 47065 47065 47065 47065
15:19:11 51088 6453 3356 156 47065 47065 47065 47065
15:19:31 328474 328474 328474 328474 328112 37271 3356 156 47065 47065 47065 47065
*** tdelta convergence time: 419 ***
.............................
```

Figure 5.3: Example of sorted updates from convergence event and the calculated convergence time (in seconds)

4. The custom topology is started and the first BGP announcement arrives at the OF Switch and is sent to the D-Forecaster. Since the first announcement is not constrained by the MRAI parameter, the D-Forecaster instructs ExaBGP router to send the announcement to the Internet through the PEERING MUX. The second PEERING prefix announcement is constrained by MRAI seconds, i.e., it is sent only after MRAI seconds elapsed since the sending of the first announcement. Then, the D-Forecaster stores the feature arrays as a sample in the dataset

5. Raw data collected and available from RIPE RIS public remote routing collectors (rrc) are downloaded. We get updates with the timestamp starting close to the time the PEERING prefix announcement was sent

6. The downloaded updates are filtered by the PEERING prefix from the rrc's and the convergence time is calculated. The output convergence time is given by y

5.2.3 Testing methodology

Similar to the dataset generation process, the testing process also involves two steps. The first step is performed on real-time. The main difference is that to test the adaptive MRAI prediction application, the trained model is loaded using the Keras library and after the first UPDATE is sent, the collected features are given as input to the prediction function, that tests the four MRAI values and deploys the MRAI with lower predicted convergence time.

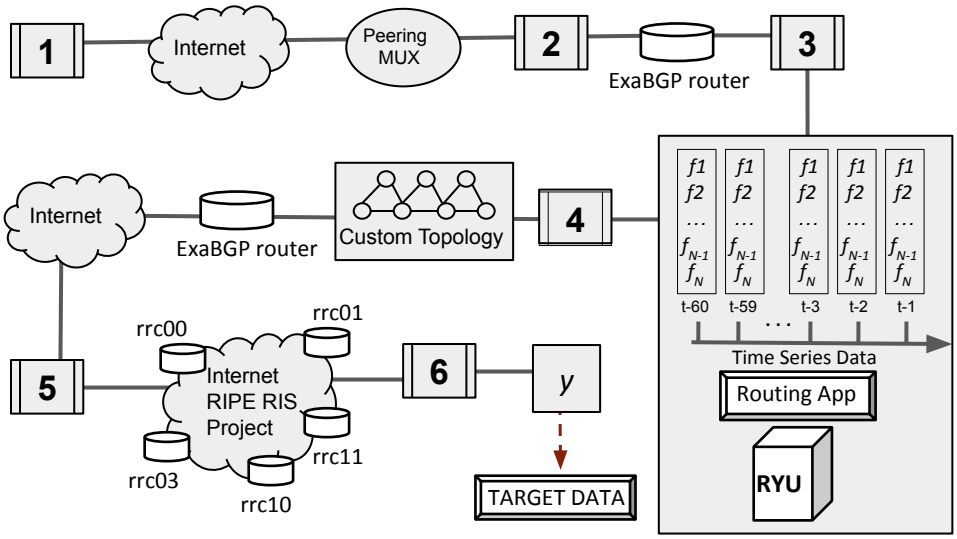


Figure 5.4: Illustration of BGP data collection and dataset generation process

Figure 5.5 illustrates the methodology and the components involved in the experiments to evaluate the MRAI predictor application.

1. A connection is established between the PEERING MUX and the ExaBGP router from the VM where the D-Forecaster runs

2. ExaBGP receives the full RIB from its PEERING MUX, with more than 750,000 prefixes

3. After receiving the full RIB, periodic updates are received from the PEERING MUX and the D-Forecaster starts to collect the features in time series format

4. The weights of the trained LSTM model are loaded by the MRAI prediction module that communicates with the D-Forecaster. The custom topology is started and the first BGP announcement arrives at the OF Switch and is sent to the D-Forecaster. Since the first announcement is not constrained by the MRAi parameter, the D-Forecaster instructs ExaBGP router to send the announcement to the Internet through the PEERING MUX

5. The second announcement is sent by neighbor in custom topology to the ExaBGP router

6. When the second update arrives, the 15 features described in table 5.1 are given as input to the LSTM model plus different MRAI values (1,5,15,30).

7. The D-Forecaster chooses the MRAI value with the smallest predicted convergence time y' and the second update is hold for the chosen MRAI value. After some

observations we determine that the experiment should wait 30 minutes to make sure the BGP converges to the triggered event before a withdrawal is sent. The process is repeated using MRAI values 1 and 30 seconds to be compared with the MRAI after prediction

8. Data is downloaded from RIPE RIS public rrc's and the convergence time is calculated to the predicted MRAI and MRAI = 1 and MRAI = 30

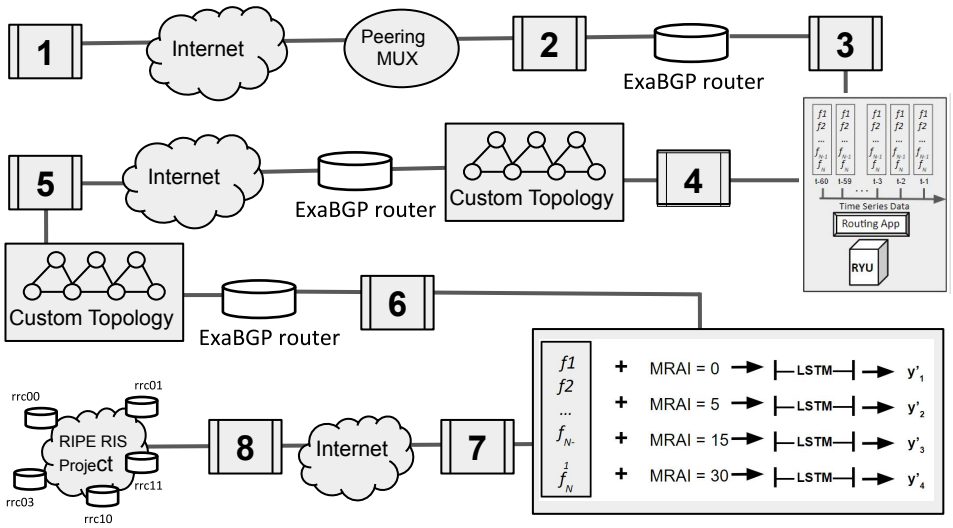


Figure 5.5: Illustration of BGP MRAIs testing process

5.3 Results and Analysis

In this section we present the results of the experiments with *Tup* and *TLong* events and the analysis. Following the testing process illustrated in figure 5.5 we tried to evaluate the D-Forecaster with adaptive MRAI by comparing its performance with the D-Forecaster with static MRAIs (1, 5, 15 and 30) when two announcements are sent at the same time from ExaBGP routers from two SDN-ASes connected to the same PEERING MUX (*neu*). However, due to the BGP dynamics, such as the MRAIs adopted at routers through the Internet where the announcements arrive, impact on the experiments result. For example, observe figure 5.6.

In figure 5.6, the announcement from D-Forecaster with Static MRAI and the announcement with Adaptive MRAI are sent at 19:23:00 (in blue) and at 19:23:05 (in red) to the PEERING MUX, which does not use MRAI. We consider that both Static and Adaptive

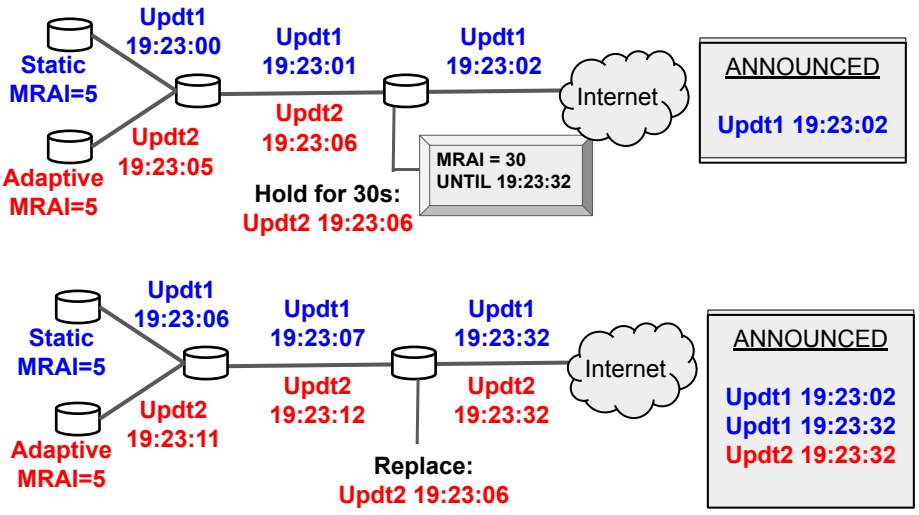


Figure 5.6: The impact of MRAI in concurrent announcements

MRAIs are 5. At 19:23:01, the UPDATE1 from static MRAI arrives at a BGP router that employs MRAI. The UPDATE1 is sent to the Internet and the MRAI is activated. When UPDATE2 arrives at BGP router at 19:23:06, the announcement is hold by MRAI. When the second UPDATEs from static and adaptive MRAIs arrive at 19:23:07 and 19:23:12, respectively, the new route from UPDATE2 from 19:23:12 replaces the old UPDATE2 from 19:23:06. When MRAI expires, UPDATE1 and UPDATE2 are sent to the Internet both at 19:23:32. In practice, only one UPDATE from Adaptive MRAI was sent because it suffered interference from the Updates from router with Static MRAI that activate MRAI from the router after the PEERING MUX. This is not a desired situation because this case was not present when the data was generated to train the model and can impact on the measured convergence time results.

To avoid the described issue, we must isolate each instance of the experiment. Hence, the announcements are generated by only one convergence event at a time for static and adaptive MRAI values. The experiment was repeated 50 times for each MRAI value and the average convergence time was calculated. The results from those experiments are shown in figure 5.10 and in table 5.4:

5.3.1 Decision Tree (DT)

Decision tree (DT) is a type of supervised machine learning with a step by step process that goes through to decide a category something belongs (classification) to or a continuous quantity output (regression).

A decision tree processes provided data and split it into two separate branches, which might be split again. When the data reaches a node which can not be split anymore, we say it reached a lead node. One interesting observation about a decision tree is how the threshold for each decision is generated. They are important to maximize the information gain at each step. That means that whatever criteria is chosen, it will split at the best location given the data in that branch [11]. A decision tree structure is illustrated in figure 5.7.

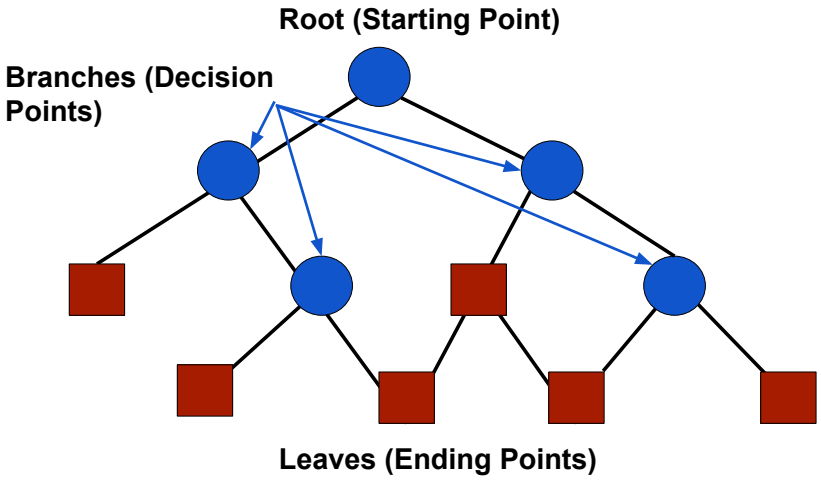


Figure 5.7: A decision tree basic structure [11]

The most commonly used criteria for regression trees is "RMSE", which was explained in the previous chapter.

Table 5.2 shows a slight improvement in the average routing convergence time when the adaptive MRAI is used training the D-Forecaster with the Decision Tree algorithm. In our experiments, the adaptive MRAI improves the average BGP routing convergence time in 5.33% for MRAI=1, 8.46% for MRAI=5, 1.12% for MRAI=15, and 11.30% for MRAI=30.

MRAI	Average Convergence Time	Improve
1	441.555	5.33%
5.	456.652	8.46%
15	422.75	1.12%
30	471.269	11.30%
Adaptive	418.5	-

Table 5.2: Average Convergence Time for Fixed and Adaptive MRAIs with Decision Tree during *Tup-TLong* events

5.3.2 Support Vector Machine Regression (SVR)

SVM is a machine learning system that uses a high dimensional feature space. It deploys prediction functions that are expanded on a subset of support vectors. With only some few support vectors, SVM is able to generalize complex structures. The Support Vector Machine Regression (SVR) is an adaptation of the classical machine learning algorithm SVM to regression problems [199].

With SVR, we try to make the error fit inside a given threshold. The parameter ε indicates the distance the line drawn by SVR is from the hyper plane. The training seeks to find a decision boundary so that the data points closer to the hyper plane are within the boundary line. This is illustrated in figure 5.8.

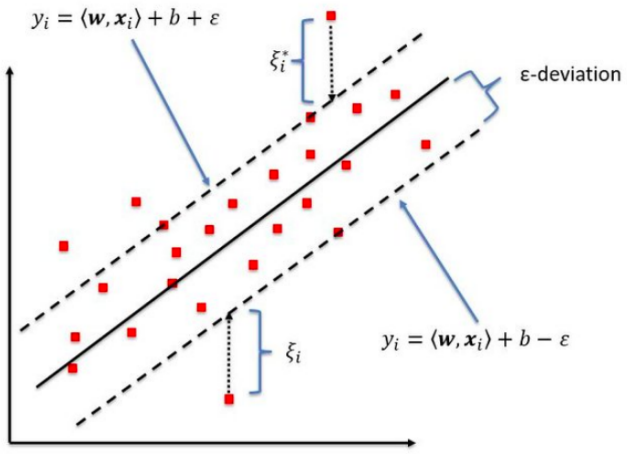


Figure 5.8: One-dimension SVR model [12]

Table 5.3 also shows a slight improvement in the average routing convergence time. In the experiments, the adaptive MRAI with SVR improves the average BGP routing convergence time in 8.05% for MRAI=1, 11.09% for MRAI=5, 3.96% for MRAI=15, and 13.84% for MRAI=30.

MRAI	Average Convergence Time	Improve
1	441.555	8.05%
5.	456.652	11.09%
15	422.75	3.96%
30	471.269	13.84%
Adaptive	406	-

Table 5.3: Average Convergence Time for Fixed and Adaptive MRAIs with SVR during *Tup-TLong* events

5.3.3 LSTM

The parameters of the LSTM learning model from the D-Forecaster were determined empirically by sending multiple announcements and measuring the convergence time. When the average convergence time is higher than one of the previous set of employed parameters, new modifications are made.

The LSTM layer used in the learning model from the D-Forecaster has 38 neurons or cells, followed by a Dropout layer with value 0.1, to reduce overfitting. After Dropout, a Dense layer with 6 neurons is added, using *sigmoid* as the activation function. Lastly, the output layer has 1 neuron with a *relu* activation function. The output of the model is the predicted convergence time for the announcement sent to PEERING. The model is trained for 100 epochs and uses *mse* (mean squared error) as its loss function, with the Adam optimizer. The collected time series data has 60 timesteps, where the features from each timestep comprise the period of 10 seconds of collection. Figure 5.9 shows the LSTM-based model architecture.

Figure 5.9: Convergence Prediction Model

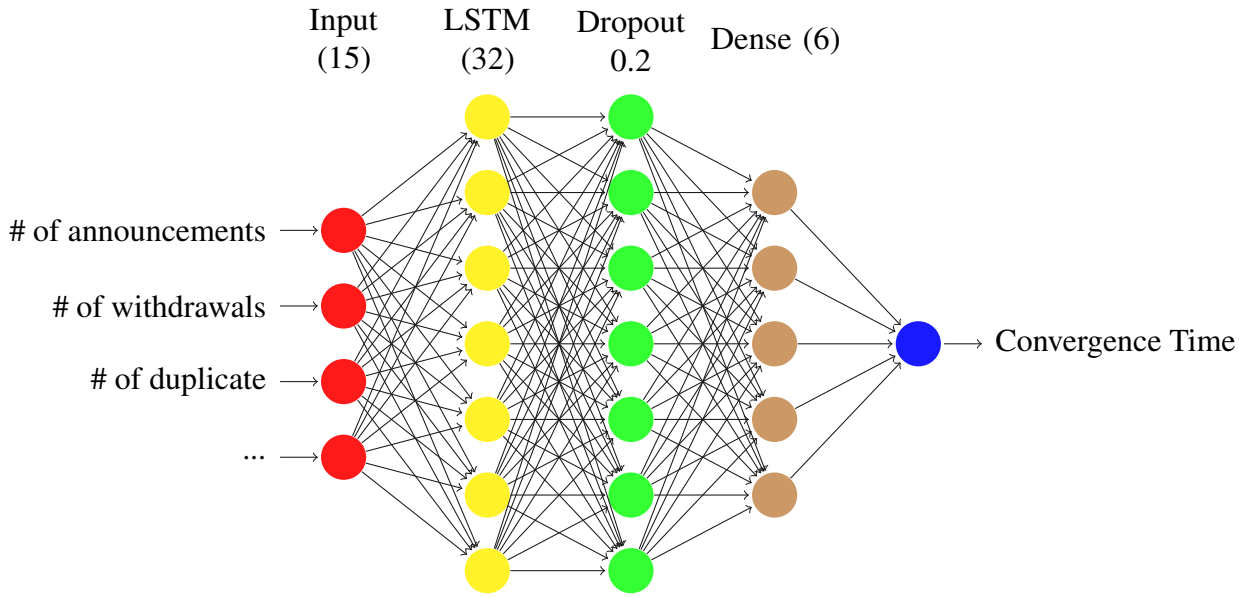


Table 5.4 show that the use of the adaptive MRAI with LSTM reduces the convergence time on the average. The adaptive MRAI outperforms the static MRAI by improving the average BGP routing convergence time in 10.72% for MRAI=1, 13.67% for MRAI=5, 6.75% for MRAI=15, and 16.35% for MRAI=30.

MRAI	Average Convergence Time	Improve
1	441.555	10.72%
5.	456.652	13.67%
15	422.75	6.75%
30	471.269	16.35%
Adaptive	394.2	-

Table 5.4: Average Convergence Time for Fixed and Adaptive MRAIs with LSTM during *Tup-TLong* events

5.3.4 Average and Median Convergence Times

Figure 5.10 show the average convergence time of DeepBGP with static MRAIs and with adaptive MRAIs from D-Forecaster built using the machine learning models Decision tree, SVR, and LSTM, respectively.

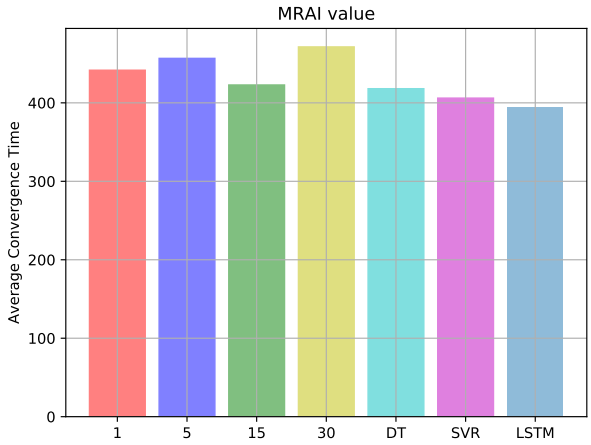


Figure 5.10: Average Convergence time for static and adaptive MRAIs

The *average* metric can be skewed by values much higher or much lower than others. To ensure this is not the case in our experiments, the *median* is also used to further evaluate the adaptive MRAI. The *median* is calculated by sorting the measured convergence times for all the experiment samples from each MRAI instance and then taking the element in the middle. If the number of the samples is even, the two middle elements are summed and the result is divided by two.

Figure 5.11 and table 5.5 show that for *median* the adaptive MRAI (DT, SVR, and LSTM) timer also presents smaller convergence time than static MRAI, but the difference is slightly small in median than observed with the average between static and adaptive MRAIs.

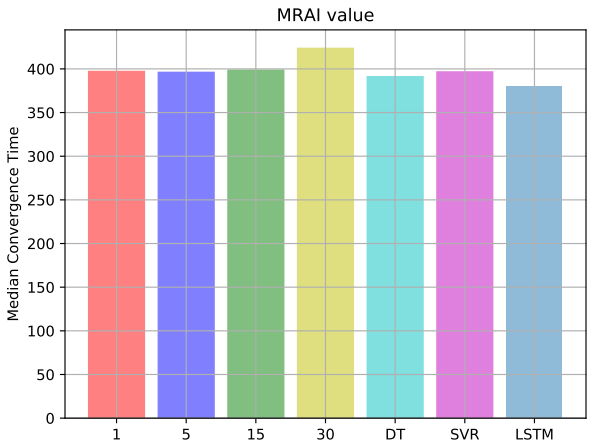


Figure 5.11: Median Convergence time for static and adaptive MRAIs

MRAI	Median Convergence Time
1	397
5.	396
15	398.5
30	423.5
DT	391
SVR	396.5
LSTM	379.5

Table 5.5: Median Convergence Time for Fixed and Adaptive MRAIs during *Tup-TLong* events

The advantage of the Adaptive MRAI is that it detects relevant patterns in the features collected on real-time and fine-tune MRAI according to patterns learned by the LSTM model. For example, the number of received updates at each time-step from the time series can indicate if the MRAIs from close neighbors at some extent are active or approximately determine in which phase the MRAIs are. The increase in number of withdrawals and a certain variation in the edit distance feature can indicate some network state that can delay the UPDATEs delivery or the increase in the number of AS_Numbers that appears with low frequency can indicate that more UPDATEs are going to be generated and possibly the convergence time will increase.

It is interesting to observe that the median convergence time for static MRAIs 0, 5 and 15 are very close, and for MRAI 30 the median increases by 25 seconds. With a number of experiment replications sufficiently large, one possible outcome behavior would be the

median difference be equal to the difference between the MRAI values, i.e., we could be able to observe a difference in the order of 15 seconds between static MRAIs 15 and 30.

The observed median convergence times for the adaptive MRAI for Decision Tree, SVR and LSTM also present a similar value, but LSTM presented a slightly better convergence time. One interesting result from the experiments with the learning models is that for Decision Tree and SVR the convergence time forecasting returned in almost 90% of the cases the same output convergence time for all the four MRAI values, i.e., no matter the MRAI value added to the other 15 features the forecast convergence was the same for all of them and the value 1 was chosen by default. Although the MRAI 1 was also the most chosen value for LSTM it was more balanced and the convergence time was always different for each of the MRAI values added to the features set when forecasting. It showed that sequential and temporal data observed by the D-Forecaster with LSTM was important to differentiate between MRAI values, since DT and SVR used only data from the last timestep before the announcement, ignoring historical and sequential features information.

Another observed limitation that could affect the result is that the DeepBGP is deployed at only one point and has limited impact over the Internet. Although it is important that the prefix is controlled and generated by the DeepBGP, it could have a more significant impact if deployed at different ASes with distinct levels of connectivity. For example, the impact of the DeepBGP in multihomed ASes or at IXPs can be much more significant than in stub ASes. It can give us an intuition on how can MRAI values such as 5 and 15 leads to similar or quite better results than when MRAI 1 or 0 is used. For example, MRAI 15 can be better than MRAI 1 in cases where one or more routers in the path to reach the route collectors are with a high number of update messages to be processed, i.e., more updates in small time interval increase the router load. In those cases waiting a certain time before sending a new update can be beneficial to reduce the number of updates to be processed, since it can send the update a little later but triggering less update announcements from other routers and impacting in the final calculated BGP convergence time. This was the case observed in our results. However, we expect that, in a large number of observations, the average and median convergence times are achieved faster when lower MRAI values are deployed. Besides that, the difference between the average and median convergence times for static MRAIs should be close to the MRAI value. For example, the difference between the average convergence times for MRAI 0 and MRAI 15 should be 15 seconds. It was not observed in the average convergence time in our experiments, where average convergence time for MRAI 15 was much better than for MRAI 1, but this difference is reduced for the median convergence time. It can indicates that it would probably also be observed for average convergence time with a considerable larger number of observations.

Chapter 6

Conclusions

This chapter concludes the thesis and present future work. The contributions are reinforced in section 6.1, future directions are discussed in section 6.2, and the list of publications during this research is presented in section 6.3.

6.1 Conclusion

To the best of our knowledge, this work is the first to apply connectionist Machine Learning models to address interdomain routing convergence issues testing the performance of the proposed application in a real scenario, since the UPDATEs are sent to the Internet. It presents the characteristics we learnt from the survey [23] that are relevant in any proposed solution that aims to address the issue of providing faster BGP routing convergence on real-world scenarios.

We first presented an extensive literature review on the main efforts on the reduction of interdomain routing convergence time observed from the Internet during last years. Advantages, disadvantages, unique characteristics, and overall insights were presented for the approaches described in the survey, providing a better understanding on the BGP convergence behavior.

The survey gave me insights to conduct a research on how I could use the available BGP data by giving them as input to machine learning models such as Long Short-Term Memory (LSTM) after those data is formatted as time series. The data was obtained from the public BGP data sources RIPE RIS and OREGON's RouteViews, and the UPDATEs used in the datasets are originated from the Beacon project. The developed models are able to predict the source convergence time with an accuracy of up to 80%.

From the knowledge obtained from the developed model to predict convergence time, I created the DeepBGP framework, where a proposed mechanism is able to fine tune the BGP

MRAI parameter aiming to reduce the routing convergence time for new routes announced to the Internet in *Tup* followed by *Tlong* events induced by race condition situations. I show that a routing model with adaptive MRAI parameter is able to learn relevant characteristics from traffic generated in a timely sequence that arrive from its BGP neighbors and that adapts its value according to observed patterns, presenting a reduced convergence time when compared to keeping a single MRAI value constant.

From the described approaches in chapter 3, most of the related work present only one or two of those characteristics that our framework presents:

- **Incremental Deployment:** The DeepBGP framework with its D-Forecaster can be deployed and coexist with current BGP routers without the need to replace the classic protocol implementation. Obviously, the impact is more significant in a scenario with the extent of the Internet when more instances of the proposed solution are deployed. We believe that innovation will happen every time more on Internet architecture, protocols and other applications due to promising paradigms such as SDN.

- **Computational Overhead:** The D-Forecaster in the DeepBGP framework is a lightweight solution, where little overhead was added in our experiments during the BGP UPDATE processing and the MRAI value prediction. Some approaches described in chapter 3 require modifications to BGP implementation that add extra overhead at routers by analysing a large number of AS_PATHs from UPDATEs that try to identify root causes of delay. In the DeepBGP framework, BGP router UPDATEs arriving from data plane are processed by the D-Forecaster that can run in a separate server at the control plane. The D-Forecaster performance can be improved, for example, with the improvement of the server where it runs.

- **BGP behavior modification:** There are proposed solutions that try to accelerate UPDATE message delivery by modifying BGP decision process. Those modifications on BGP behavior can create inconsistencies with existing classic BGP deployment on routers in the Internet. The D-Forecaster preserves the BGP main behavior by only tuning the MRAI value for each new convergence event, i.e., it only changes the time in seconds the new UPDATEs will be hold before send it to a BGP peer.

- **BGP expressiveness:** BGP expressiveness is given by the power the protocol give network managers and administrators to model BGP accepted routes and the routes that are advertised to neighbors according to the policies of the Autonomous System. Part of the BGP behavior hard to predict comes from this aspect, which is an important characteristic of the protocol. Some approaches try to bring more predictability by

constraining BGP expressiveness, which can limit one important feature of the protocol. The DeepBGP framework preserves the BGP expressiveness.

- **Coordination between ASes:** Some of the proposed solutions to address the interdomain routing convergence and stability issues require the mechanism to be deployed on the most number of ASes it is possible to work. It compromises other important characteristic, which is the incremental deployment, since it does not work with current BGP implementation without the proposed modifications. The DeepBGP framework, on its turn, can work well with existent BGP implementation withou requiring any new coordination between ASes, besides the one already provided by current BGP protocol.

- **Impact on Internet real scenario:** Many of the proposed solutions in interdomain routing are performed in simulators constrained to environments without the scalability on the Internet. The PEERING platform and RIPE RIS and RouteViews enables the DeepBGP framework to send UPDATEs to routers across the Internet and obtain measurements closer to a real-world scenario.

The last contribution of this work is that the developed code and datasets used in this thesis are available in an open repository [29**?**] and can be extended to address new problems and the experiments can be replicated. A tutorial on how to configure and use the framework is available on the repository.

The DeepBGP framework currently presents some limitations. It is deployed at one single point that constrains the sending of UPDATEs to PEERING with MRAI. We would like to experiment the performance of the proposed solution by increasing the number of DeepBGP frameworks instances in the scenario to evaluate how the increasing deployment of the adaptive MRAI can impact on the BGP routing convergence. However, this is a hard task to achieve the scalability of the number of current existing ASes in the Internet. Besides, we abstract the aspects of intradomain routing that can impact on the routing convergence, since in the model each BGP router represents an AS. This is a common abstraction in interdomain routing models.

Some improvements can be made to the DeepBGP framework and to the D-Forecaster presented in this thesis:

- New features can be implemented or modified in the model to improve its prediction accuracy besides the ones presented in this paper.

- New datasets can be generated using available scripts to train and test the model further.

- We collected the UPDATEs to measure the convergence time from the collectors of RIS project in the Adaptive MRAI experiments. More experiments can be done with collectors from other sources, such as from RouteViews, and compare to the results presented in this thesis.

6.2 Future Work

The presented DeepBGP framework opens up new directions to be explored by addressing other issues related to BGP protocol.

One interesting research direction is how to address inconsistencies caused by conflicts in the policies applied between ASes with different commercial interests. BGP offers great flexibility and expressiveness to network administrators, but the main drawback of this feature is that it makes the protocol behavior harder to predict and to detect inconsistencies when they occur. Anwar et al. [200] presented a methodology that account for the potential causes of routing violations in control plane data by surveying a sample of network operators about their deployed policies. Among the identified causes are the complex AS relationships. A promising direction for this work would be the extension of MRAI-pref framework to support agent-based functionalities such as AgNOS [201].

Queiroz *et al.* [201] built AgNOS as an integration between Autonomous Networks and SDN. In the AgNOS multi-agent environment, the agents cooperate to exchange some level of information that helps to detect, for example, DDoS attacks. Since DeepBGP framework also deploys the SDN paradigm, it could be extended to support a multi-agent application, where multiple SDN-ASes can use some machine learning technique to learn to what extent policies can cause routing inconsistencies and use the agents to cooperate to find a policy configuration that mitigate or prevent the policy inconsistencies. The challenge is to incentive network operators to share some level of policy information with other ASes without constrains its commercial and competitive interests. This is an interesting problem to be investigated.

The *Route Flap Damping* (RFD) is a mechanism used in many BGP routers today to reduce the number of oscillations caused by unstable routes. However, Mao et al. [86] show that RFD can actually increases the convergence time of stable routes, sometimes by more than one hour. RFD strongly penalizes well-connected ASes because with a high number of connections the number of exchanged messages increases. Reachability issues and packet loss were detected when no physical link failures or congestion occurred in the network. It means that valid routes might be wrongly withdrawn due to the RFD mechanism. We believe that DeepBGP framework can be extended to use new models that try to learn patterns

of possible sources of instabilities to establish a penalty to routes that oscillate frequently, without penalizing well-behaved stable routes.

Two future works were described, but other BGP-related issues can be addressed by extending or modifying the DeepBGP framework.

6.3 Publications

The following list presents the papers published during this research and papers under peer-review by the community:

Published papers:

- R. Bennesby, E. Mota, P. Fonseca, and A. Passito, “Innovating on interdomain routing with an inter-SDN component,” in IEEE 28th International Conference on Advanced Information Networking and Applications (AINA), pp. 131–138, May 2014.
- R. Bennesby and E. Mota, “A survey on approaches to reduce bgp interdomain routing convergence delay on the internet,” IEEE Communications Surveys Tutorials, vol. PP, no. 99, pp. 1–1, 2017.
- P. Fonseca, E. Mota, R. Bennesby, and A. Passito, "BGP Dataset Generation and Feature Extraction for Anomaly Detection", IEEE Symposium on Computers and Communications (ISCC), 2019 (*to appear in*).

Submitted papers under evaluation:

- R. Bennesby, E. Mota, P. Fonseca, and A. Passito, “Analysis and prediction of BGP convergence time using LSTM networks,” in Elsevier Computer Networks (COMNET), 2019.
- R. Bennesby, E. Mota, P. Fonseca, and A. Passito, “A Machine-Learning Approach to MRAI Fine Tuning to reduce BGP Routing Convergence Time,” in Elsevier Journal of Network and Computer Applications (JNCA), 2019.

Bibliography

[1] L. Gao and J. Rexford, "Stable Internet routing without global coordination," in *Proceedings of the 2000 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '00, (New York, NY, USA), pp. 307–317, ACM, June 2000.

[2] M. Cheng, Q. Xu, J. Lv, W. Liu, Q. Li, and J. Wang, "Ms-lstm: A multi-scale lstm model for bgp anomaly detection," in *2016 IEEE 24th International Conference on Network Protocols (ICNP)*, pp. 1–6, Nov 2016.

[3] X. Zhao, D. Massey, D. Pei, and L. Zhang, "A study on the routing convergence of latin american networks," in *Proceedings of the 2003 IFIP/ACM Latin America Conference on Towards a Latin American Agenda for Network Research*, LANC '03, (New York, NY, USA), pp. 35–43, ACM, 2003.

[4] G. Huston, M. Rossi, and G. Armitage, "A technique for reducing BGP update announcements through path exploration damping," in *IEEE Journal on Selected Areas in Communications*, vol. 28, pp. 1271–1286, Oct. 2010.

[5] D. Pei, X. Zhao, L. Wang, D. Massey, A. Mankin, S. Su, and L. Zhang, "Improving BGP convergence through consistency assertions," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, pp. 902–911, 2002.

[6] K. Lakshminarayanan, M. Caesar, M. Rangan, T. Anderson, S. Shenker, and I. Stoica, "Achieving convergence-free routing using failure-carrying packets," in *Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '07, (New York, NY, USA), pp. 241–252, ACM, 2007.

[7] W. Xu and J. Rexford, "MIRO: Multi-path interdomain routing," in *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '06, (New York, NY, USA), pp. 171–182, ACM, 2006.

[8] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, and A. Vahdat, "B4: Experience with a globally-deployed software defined wan," in *Proceedings of the ACM Conference on SIGCOMM*, SIGCOMM '13, (New York, NY, USA), pp. 3–14, ACM, 2013.

[9] M. Mao, R. Bush, T. Griffin, and M. Roughan, "BGP beacons," in *Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement*, IMC '03, (New York, NY, USA), pp. 1–14, ACM, 2003.

[10] A. Bremler-Barr, N. Chen, J. Kangasharju, O. Mokryn, and Y. Shavitt, "Bringing order to BGP: Decreasing time and message complexity," in *Proceedings of the Twenty-sixth Annual ACM Symposium on Principles of Distributed Computing*, PODC '07, (New York, NY, USA), pp. 368–369, ACM, 2007.

[11] S. Hartshorn, *Machine Learning With Random Forests And Decision Trees: A Visual Guide For Beginners*. ebook kindle, 2014.

[12] T. Kleynhans, M. Montanaro, A. Gerace, and C. Kanan, "Predicting top-of-atmosphere thermal radiance using merra-2 atmospheric data with deep learning," *Remote Sensing*, vol. 9, p. 1133, 11 2017.

[13] Y. Rekhter, T. Li, and S. Hares, "RFC 4271: A Border Gateway Protocol 4 (BGP-4)," tech. rep., IETF, 2006.

[14] B. Raghavan, M. Casado, T. Koponen, S. Ratnasamy, A. Ghodsi, and S. Shenker, "Software-defined internet architecture: decoupling architecture from infrastructure," in *Proceedings of the 11th ACM Workshop on Hot Topics in Networks*, HotNets-XI, pp. 43–48, ACM, 2012.

[15] N. Feamster, H. Balakrishnan, J. Rexford, A. Shaikh, and J. van der Merwe, "The case for separating routing from routers," in *Proc. of the ACM SIGCOMM Workshop on Future Directions in Network Architecture*, FDNA'04, (New York, NY, USA), pp. 5–12, ACM, 2004.

[16] T. Koponen, S. Shenker, H. Balakrishnan, N. Feamster, I. Ganichev, A. Ghodsi, P. Godfrey, N. McKeown, G. Parulkar, B. Raghavan, J. Rexford, S. Arianfar, and D. Kuptsov, "Architecting for innovation," *SIGCOMM Comput. Commun. Rev.*, vol. 41, pp. 24–36, July 2011.

[17] C. Labovitz, G. R. Malan, and F. Jahanian, "Internet routing instability," *IEEE/ACM Transactions on Networking*, vol. 6, pp. 515–528, Oct. 1998.

[18] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed Internet routing convergence," in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, vol. 30 of *SIGCOMM '00*, (New York, NY, USA), pp. 175–187, ACM, Oct. 2000.

[19] J. Luo, J. Xie, R. Hao, and X. Li, "An approach to accelerate convergence for path vector protocol," in *Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE*, vol. 3, pp. 2390–2394 vol.3, Nov. 2002.

[20] N. Kushman, S. Kandula, and D. Katabi, "Can you hear me now?!: It must be BGP," *SIGCOMM Comput. Commun. Rev.*, vol. 37, pp. 75–84, Mar. 2007.

[21] A. Pontes, A. Drummond, N. Fonseca, and A. Jukan, "PCE-based inter-domain lightpath provisioning," in *IEEE International Conference on Communications (ICC)*, pp. 3073–3078, June 2012.

[22] A. Pontes, N. Fonseca, and A. Drummond, "Schemes for inter-domain lightpath establishment based on PCE architecture," *Optical Switching and Networking*, vol. 19, Part 1, pp. 10 – 21, 2016.

[23] R. Bennesby and E. Mota, "A survey on approaches to reduce bgp interdomain routing convergence delay on the internet," *IEEE Communications Surveys Tutorials*, vol. PP, no. 99, pp. 1–1, 2017.

[24] A. Fabrikant, U. Syed, and J. Rexford, "There's something about MRAI: Timing diversity can exponentially worsen BGP convergence," in *INFOCOM'11*, pp. 2975–2983, Apr. 2011.

[25] P. A. Lapukhov, P. and E. J. Mitchell, "Use of bgp for routing in large-scale data centers," RFC 7938, Aug. 2016.

[26] T. Griffin and B. Premore, "An experimental analysis of BGP convergence time," in *Ninth International Conference on Network Protocols*, pp. 53–61, Nov. 2001.

[27] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: enabling innovation in campus networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, pp. 69–74, Mar. 2008.

[28] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learnin- From Theory to Algorithms*. 32 Avenue of Americas, New York, USA: Cambridge University Press, 2014.

[29] BGP-Convergence-Predictor. https://github.com/ricardo-bennesby/bgp-convergence, 2019.

[30] MRAI-Pred-Framework. https://github.com/ricardo-bennesby/mrai-pred, 2019.

[31] E. Alabdulkreem, H. Al-Raweshidy, and M. Abbod, "Using a fight-or-flight mechanism to reduce BGP convergence time," in *Communications and Networking (ComNet), 2014 International Conference on*, pp. 1–4, Mar. 2014.

[32] D. Medhi and K. Ramasamy, *Network Routing: Algorithms, Protocols, and Architectures*. Morgan Kaufmann, 2007.

[33] M. Caesar, L. Subramanian, and R. H. Katz, "Towards root cause analysis of Internet routing dynamics," in *Berkeley EECS Annual Research Symposium*, 2004.

[34] J. Li, M. Guidero, Z. Wu, E. Purpus, and T. Ehrenkranz, "BGP routing dynamics revisited," *SIGCOMM Comput. Commun. Rev.*, vol. 37, pp. 5–16, Mar. 2007.

[35] H. Shaza, *Impact of Topology on BGP Convergence*. PhD thesis, Vrije University of Amsterdam, 2010.

[36] T. Li and G. Huston, "BGP stability improvements' draft-li-bgp-stability-01."

[37] S. Halabi, *Internet Routing Architectures*. Cisco, second ed., 2007.

[38] J. Qiu, F. Wang, and L. Gao, "BGP rerouting solutions for transient routing failures and loops," in *Military Communications Conference, 2006. MILCOM 2006. IEEE*, pp. 1–7, 2006.

[39] D. Pei, M. Azuma, D. Massey, and L. Zhang, "BGP-RCN: Improving BGP convergence through root cause notification," *Comput. Netw.*, vol. 48, pp. 175–194, June 2005.

[40] Y. Afek, A. Bremler-Barr, and S. Schwarz, "Improved BGP convergence via ghost flushing," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 10, pp. 1933–1948, 2004.

[41] A. Sahoo, K. Kant, and P. Mohapatra, "BGP convergence delay under large-scale failures: Characterization and solutions," *ICC*, vol. 39, pp. 111–122, Aug. 2009.

[42] P. Kirsur and O. Alani, "Reducing BGP convergence time by fine tuning the MRAI timer on different topologies." http://www.cms.livjm.ac.uk/pgnet2012/Proceedings/Papers/1569603251.pdf, 2012.

[43] D. Drutskoy, E. Keller, and J. Rexford, "Scalable Network Virtualization in Software-Defined Networks," *Internet Computing, IEEE*, vol. 17, no. 2, pp. 20–27, 2013.

[44] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[45] J. Zhang, J. Rexford, and J. Feigenbaum, "Learning-based anomaly detection in bgp updates," in *Proceedings of the 2005 ACM SIGCOMM Workshop on Mining Network Data*, MineNet '05, (New York, NY, USA), pp. 219–220, ACM, 2005.

[46] S. Haeri, D. Krešsić, and L. Trajković, "Probabilistic verification of bgp convergence," in *2011 19th IEEE International Conference on Network Protocols*, pp. 127–128, Oct 2011.

[47] N. M. Al-Rousan and L. Trajković, "Machine learning models for classification of bgp anomalies," in *2012 IEEE 13th International Conference on High Performance Switching and Routing*, pp. 103–108, June 2012.

[48] J. Susan and L. Ruan, "A machine learning approach to edge type prediction in internet as graphs," Tech. Rep. TR15-09, Iowa State University, Ames, Iowa, United States, 2015.

[49] X. Du, M. A. Shayman, and R. Skoog, "Using neural networks to identify control and management plane poison messages," in *IFIP/IEEE Eighth International Symposium on Integrated Network Management, 2003.*, pp. 621–634, March 2003.

[50] A. Lutu, M. Bagnulo, J. Cid-Sueiro, and O. Maennel, "Separating wheat from chaff: Winnowing unintended prefixes using machine learning," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pp. 943–951, April 2014.

[51] I. O. de Urbina Cazenave, E. Köşlük, and M. C. Ganiz, "An anomaly detection framework for bgp," in *2011 International Symposium on Innovations in Intelligent Systems and Applications*, pp. 107–111, June 2011.

[52] B. Al-Musawi, P. Branch, and G. Armitage, "Detecting bgp instability using recurrence quantification analysis (rqa)," in *2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC)*, pp. 1–8, Dec 2015.

[53] J. Li, D. Dou, Z. Wu, S. Kim, and V. Agarwal, "An internet routing forensics framework for discovering rules of abnormal bgp events," *SIGCOMM Comput. Commun. Rev.*, vol. 35, pp. 55–66, Oct. 2005.

[54] S. Haykin, *Neural Networks and Learning Machines, third edition*. Prentice Hall, 2014.

[55] "Feature Weighting Using Neural Networks." http://axon.cs.byu.edu/papers/zeng_martinez_ijcnn04.pdf.

[56] F. Chollet, *Deep Learning with Python*. Greenwich, CT, USA: Manning Publications Co., 1st ed., 2017.

[57] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, pp. 1735–1780, Nov. 1997.

[58] J. Li, A. Mohamed, G. Zweig, and Y. Gong, "Lstm time and frequency recurrence for automatic speech recognition," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 187–191, Dec 2015.

[59] W. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," *PLoS ONE*, vol. 12, 07 2017.

[60] K. Krishna, D. Jain, S. V. Mehta, and S. Choudhary, "An lstm based system for prediction of human activities with durations," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, pp. 147:1–147:31, Jan. 2018.

[61] J. Fu, P. Sjödin, and G. Karlsson, "Intra-domain routing convergence with centralized control," *Computer Networks*, vol. 53, no. 18, pp. 2985 – 2996, 2009.

[62] F. Wang, Z. M. Mao, J. Wang, L. Gao, and R. Bush, "A measurement study on the impact of routing events on end-to-end Internet path performance," in *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '06, (New York, NY, USA), pp. 375–386, ACM, 2006.

[63] V. Kotronis, X. Dimitropoulos, and B. Ager, "Outsourcing the routing control logic: Better Internet routing based on SDN principles," in *Proceedings of the 11th ACM Workshop on Hot Topics in Networks*, HotNets-XI, (New York, NY, USA), pp. 55–60, ACM, 2012.

[64] J. P. John, E. Katz-Bassett, A. Krishnamurthy, T. Anderson, and A. Venkataramani, "Consensus routing: The Internet as a distributed system," in *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, NSDI'08, (Berkeley, CA, USA), pp. 351–364, USENIX Association, 2008.

[65] R. Viswanathan, K. Sabnani, R. Holt, and A. Netravali, "Expected convergence properties of BGP," in *Network Protocols, 2005. ICNP 2005. 13th IEEE International Conference on*, pp. 13 pp.–15, Nov. 2005.

[66] R. Mahajan, D. Wetherall, and T. Anderson, "Understanding BGP misconfiguration," in *Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '02, (New York, NY, USA), pp. 3–16, ACM, 2002.

[67] M. Yannuzzi, X. Masip-Bruin, and O. Bonaventure, "Open issues in interdomain routing: a survey," *IEEE Network*, vol. 19, pp. 49–56, Nov. 2005.

[68] S. Agarwal, C. nee Chuah, S. Bhattacharyya, and C. Diot, "The impact of BGP dynamics on intra-domain traffic," in *In Proceedings of ACM SIGMETRICS*, vol. 32, (New York, NY, USA), ACM, June 2004.

[69] L. Cittadini, G. D. Battista, and M. Rimondini, "On the stability of Interdomain routing," *ACM Comput. Surv.*, vol. 44, pp. 26:1–26:40, Sept. 2012.

[70] D. Gupta, A. Segal, A. Panda, G. Segev, M. Schapira, J. Feigenbaum, J. Rexford, and S. Shenker, "A new approach to interdomain routing based on secure multi-party computation," in *Proceedings of the 11th ACM Workshop on Hot Topics in Networks*, HotNets-XI, (New York, NY, USA), pp. 37–42, ACM, 2012.

[71] D.-F. Chang, R. Govindan, and J. Heidemann, "The temporal and topological characteristics of BGP path changes," in *proceedings of IEEE ICNP*, 2003.

[72] F. Wang, J. Qiu, L. Gao, and J. Wang, "On understanding transient interdomain routing failures," *IEEE/ACM Transactions on Networking*, vol. 17, pp. 740–751, June 2009.

[73] J. Chandrashekar, Z. Duan, Z. Zhang, and J. Krasky, "Limiting path exploration in BGP," in *Proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societies- INFOCOM*, vol. 4, pp. 2337–2348, Mar. 2005.

[74] T. Holterbach, S. Vissicchio, A. Dainotti, and L. Vanbever, "Swift: Predictive fast reroute," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, SIGCOMM '17, (New York, NY, USA), pp. 460–473, ACM, 2017.

[75] E. Katz-Bassett, C. Scott, D. R. Choffnes, I. Cunha, V. Valancius, N. Feamster, H. V. Madhyastha, T. Anderson, and A. Krishnamurthy, "LIFEGUARD: Practical repair of persistent route failures," in *Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, SIGCOMM '12, (New York, NY, USA), pp. 395–406, ACM, 2012.

[76] X. Wang, O. Bonaventure, and P. Zhu, "Stabilizing BGP routing without harming convergence," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 840–845, Apr. 2011.

[77] "Latin American and Caribbean Internet Addresses Registry." http://www.lacnic.net/.

[78] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian, "Internet inter-domain traffic," in *Proceedings of the ACM SIGCOMM 2010 Conference*, SIGCOMM '10, (New York, NY, USA), pp. 75–86, ACM, 2010.

[79] B. Zhang, D. Massey, and L. Zhang, "Destination reachability and BGP convergence time [border gateway routing protocol]," in *IEEE Global Telecommunications Conference- GLOBECOM '04*, vol. 3, pp. 1383–1389, Nov. 2004.

[80] D. Pei, L. Wang, D. Massey, S. F. Wu, and L. Zhang, "A study of packet delivery performance during routing convergence," in *Proceedings of IEEE International Conference on Dependable Systems and Networks (DSN)*, 2003.

[81] A. Sahoo, K. Kant, and P. Mohapatra, "Improving BGP convergence delay for large-scale failures," in *International Conference on Dependable Systems and Networks-DSN*, pp. 323–332, June 2006.

[82] A. Sahoo, K. Kant, and P. Mohapatra, "BGP convergence delay after multiple simultaneous router failures: Characterization and solutions," *Computer Communications*, vol. 32, pp. 1207–1218, May 2009.

[83] "Revisions to the BGP 'Minimum Route Advertisement Interval' draft-ietf-idr-mrai-dep-02." https://tools.ietf.org/id/draft-ietf-idr-mrai-dep-02.html.

[84] J. Qiu, R. Hao, and X. Li, "The optimal rate-limiting timer of BGP for routing convergence," in *IEICE Transactions on Communications*, vol. 88-B, pp. 1338–1346, 2005.

[85] "Planet Lab- A Global Research Network Platform." https://www.planet-lab.org/.

[86] 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*, SIGCOMM '02, (New York, NY, USA), pp. 221–233, ACM, 2002.

[87] C. Pelsser, O. Maennel, P. Mohapatra, R. Bush, and K. Patel, "Route flap damping made usable," in *Proceedings of the 12th International Conference on Passive and Active Measurement*, PAM'11, (Berlin, Heidelberg), pp. 143–152, Springer-Verlag, 2011.

[88] T. Griffin, B. Shepherd, and G. Wilfong, "The stable paths problem and interdomain routing," *IEEE/ACM Transactions on Networking*, vol. 10, pp. 232–243, Apr. 2002.

[89] L. Cittadini, M. Rimondini, M. Corea, and G. Di Battista, "On the feasibility of static analysis for BGP convergence," in *IFIP/IEEE International Symposium on Integrated Network Management- IM '09*, pp. 521–528, June 2009.

[90] F. Wang, J. Qiu, L. Gao, and J. Wang, "On understanding transient interdomain routing failures," *IEEE/ACM Transactions on Networking*, vol. 17, pp. 740–751, June 2009.

[91] "Réseaux IP Européens (RIPE)." https://www.ripe.net/.

[92] J. Rexford, J. Wang, Z. Xiao, and Y. Zhang, "BGP routing stability of popular destinations," in *Proceedings of the 2Nd ACM SIGCOMM Workshop on Internet Measurment*, IMW '02, (New York, NY, USA), pp. 197–202, ACM, 2002.

[93] R. Oliveira, R. Izhak-Ratzin, B. Zhang, and L. Zhang, "Measurement of highly active prefixes in BGP," in *IEEE Global Telecommunications Conference- GLOBECOM '05*, vol. 2, pp. 5 pp.–, Nov 2005.

[94] N. Feamster, R. Johari, and H. Balakrishnan, "Implications of autonomy for the expressiveness of policy routing," *IEEE/ACM Transactions on Networking*, vol. 15, pp. 1266–1279, Dec. 2007.

[95] N. Feamster, R. Johari, and H. Balakrishnan, "Stable policy routing with provider independence," tech. rep., 2005.

[96] A. Feldmann, H. Kong, O. Maennel, and A. Tudor, *Measuring BGP Pass-Through Times*, pp. 267–277. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004.

[97] B. Wang, "The research of BGP convergence time," in *6th IEEE Joint International Information Technology and Artificial Intelligence Conference*, vol. 2, pp. 354–357, Aug 2011.

[98] W. Wenhua, S. Qingguo, and Z. Qin, "On the relationship between BGP convergence delay and network topology," in *11th IEEE International Conference on Communication Technology*, pp. 546–549, Nov 2008.

[99] V. Schrieck, P. Francois, C. Pelsser, and O. Bonaventure, "Preventing the unnecessary propagation of BGP withdraws," in *Proceedings of IFIP Networking* (Y. T. Luigi Fratta, Henning Schulzrinne and O. Spaniol, eds.), pp. 495–508, Springer Verlag, May 2009.

[100] A. Elmokashfi, A. Kvalbein, and C. Dovrolis, "BGP churn evolution: A perspective from the core," *Networking, IEEE/ACM Transactions on*, vol. 20, no. 2, pp. 571–584, 2012.

[101] A. Elmokashfi, A. Kvalbein, and C. Dovrolis, "On the scalability of BGP: The roles of topology growth and update rate-limiting," in *Proceedings of the 2008 ACM CoNEXT Conference*, (New York, NY, USA), pp. 8:1–8:12, ACM, 2008.

[102] A. Elmokashfi and A. Dhamdhere, "Revisiting BGP churn growth," *SIGCOMM Comput. Commun. Rev.*, vol. 44, pp. 5–12, Dec. 2013.

[103] G. Huston, "The Churn Report." https://labs.apnic.net/?p=457, 2014.

[104] J. Nykvist and L. Carr-Motykova, "Simulating convergence properties of BGP," in *Proceedings of Eleventh International Conference on Computer Communications and Networks*, pp. 124–129, Oct. 2002.

[105] S. Deshpande and B. Sikdar, "On the impact of route processing and MRAI timers on BGP convergence times," in *IEEE Global Telecommunications Conference- GLOBECOM '04*, vol. 2, pp. 1147–1151, Nov. 2004.

[106] N. Lasković, "BGP with an adaptive minimal route advertisement interval," Master's thesis, Simon Fraser University, 2006.

[107] A. Elmokashfi, A. Kvalbein, and T. Cicic, "On update rate-limiting in BGP," in *IEEE International Conference on Communications (ICC)*, pp. 1–6, June 2011.

[108] W. Sun, Z. Mao, and K. Shin, "Differentiated BGP update processing for improved routing convergence," in *Proceedings of IEEE International Conference on Network Protocols*, ICNP '06, (Washington, DC, USA), pp. 280–289, IEEE Computer Society, 2006.

[109] E. Alabdulkreem, H. Al-Raweshidy, and M. Abbod, "MRAI optimization for BGP convergence time reduction without increasing the number of advertisement messages," in *Proceedings of the International Conference on Soft Computing and Software Engineering, SCSE'15*, vol. 62, pp. 419 – 426, 2015.

[110] C. Villamizar, R. Chandra, and R. Govindan, "BGP route flap damping," RFC 2439, Nov. 1998.

[111] W. Lijun, W. Jianping, and X. Ke, "A variation of route flap damping to improve BGP routing convergence," in *14th IEEE International Workshop on Quality of Service-IWQoS*, pp. 297–301, June 2006.

[112] A. Sahoo, K. Kant, and P. Mohapatra, "Speculative route invalidation to improve BGP convergence delay under large-scale failures," in *Proceedings of the 15th International Conference on Computer Communications and Networks- ICCCN*, pp. 461–466, Oct. 2006.

[113] Y. Liao, L. Gao, R. Guerin, and Z.-L. Zhang, "Reliable interdomain routing through multiple complementary routing processes," in *Proceedings of the 2008 ACM CoNEXT Conference*, (New York, NY, USA), pp. 68:1–68:6, ACM, 2008.

[114] A. Feldmann, O. Maennel, Z. M. Mao, A. Berger, and B. Maggs, "Locating Internet routing instabilities," in *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '04, (New York, NY, USA), pp. 205–218, ACM, 2004.

[115] A. Lambert, M. Buob, and S. Uhlig, "Improving Internet-wide routing protocols convergence with MRPC timers.," in *Proceedings of the 2009 ACM CoNEXT Conference*, pp. 325–336, ACM, 2009.

[116] B. Godfrey, M. Caesar, I. Haken, Y. Singer, S. Shenker, and I. Stoica, "Stabilizing route selection in BGP," *IEEE/ACM Transactions on Networking*, vol. 23, pp. 282–299, Feb. 2015.

[117] E. Ahronovitz, J. Konig, and C. Saad, "A distributed method for dynamic resolution of BGP oscillations," in *20th International Parallel and Distributed Processing Symposium- IPDPS*, pp. 10 pp.–, Apr. 2006.

[118] K. Varadhan, R. Govindan, and D. Estrin, "Persistent route oscillations in inter-domain routing," *Computer Networks*, vol. 32, no. 1, pp. 1 – 16, 2000.

[119] D. Obradovic, "Real-time model and convergence time of BGP," in *Proceedings of IEEE Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies- INFOCOM*, vol. 2, pp. 893–901, 2002.

[120] H. Zhang, A. Arora, and Z. Liu, "A stability-oriented approach to improving BGP convergence," in *Proceedings of the 23rd IEEE International Symposium on Reliable Distributed Systems*, pp. 90–99, Oct. 2004.

[121] H. Levin, M. Schapira, and A. Zohar, "Interdomain routing and games," in *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, (New York, NY, USA), pp. 57–66, ACM, 2008.

[122] Y. Wang, M. Schapira, and J. Rexford, "Neighbor-specific BGP: More flexible routing policies while improving global stability," in *Proceedings of the Eleventh International Joint Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '09, (New York, NY, USA), pp. 217–228, ACM, 2009.

[123] H. Guo, W. Su, H. Zhang, and S.-Y. Kuo, "On the convergence condition and convergence time of BGP," *Computer Communications*, vol. 34, no. 2, pp. 192 – 199, 2011. Special Issue: Open network service technologies and applications.

[124] Q. Li, M. Xu, J. Wu, P. Lee, and D. Chiu, "Toward a practical approach for BGP stability with root cause check," *J. Parallel Distrib. Comput.*, vol. 71, pp. 1098–1110, Aug. 2011.

[125] N. Kushman, S. Kandula, D. Katabi, and B. M. Maggs, "R-BGP: Staying connected in a connected world," in *Proceedings of the 4th USENIX Conference on Networked Systems Design & Implementation*, NSDI'07, (Berkeley, CA, USA), pp. 25–25, USENIX Association, 2007.

[126] X. Yang, D. Clark, and A. W. Berger, "NIRA: A new inter-domain routing architecture," *IEEE/ACM Transactions on Networking*, vol. 15, pp. 775–788, Aug. 2007.

[127] M. Caesar, M. Casado, T. Koponen, J. Rexford, and S. Shenker, "Dynamic route recomputation considered harmful," *SIGCOMM Comput. Commun. Rev.*, vol. 40, pp. 66–71, Apr. 2010.

[128] X. Yang and D. Wetherall, "Source selectable path diversity via routing deflections," in *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '06, (New York, NY, USA), pp. 159–170, ACM, 2006.

[129] A. Szekeres, "Multi-path inter-domain routing: The impact on BGP's scalability, stability and resilience to link failures," Master's thesis, Vrije University of Amsterdam, 2011.

[130] A. Li, X. Yang, and D. Wetherall, "Safeguard: Safe forwarding during route changes," in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, CoNEXT '09, (New York, NY, USA), pp. 301–312, ACM, 2009.

[131] N. Feamster, D. Andersen, H. Balakrishnan, and F. Kaashoek, "Measuring the effects of Internet path faults on reactive routing," in *Proceedings of the 2003 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '03, (New York, NY, USA), pp. 126–137, ACM, 2003.

[132] N. Feamster, J. Rexford, and E. Zegura, "The road to SDN: An intellectual history of programmable networks," *SIGCOMM Comput. Commun. Rev.*, vol. 44, pp. 87–98, Apr. 2014.

[133] F. Hu, Q. Hao, and K. Bao, "A survey on software-defined network and openFlow: From concept to implementation," *IEEE Communications Surveys and Tutorials*, vol. 16, pp. 2181–2206, Fourthquarter 2014.

[134] M. Jammal, T. Singh, A. Shami, R. Asal, and Y. Li, "Software-Defined Networking: State of the Art and Research Challenges," *ArXiv e-prints*, May 2014.

[135] D. Kreutz, F. Ramos, P. Verissimo, C. Esteve, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," *ArXiv e-prints*, June 2014.

[136] B. Schlinker, K. Zarifis, I. Cunha, N. Feamster, E. Katz-Bassett, and M. Yu, "Try before you buy: SDN emulation with (real) interdomain routing," in *Presented as part of the Open Networking Summit 2014 (ONS 2014)*, (Santa Clara, CA), USENIX, 2014.

[137] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: rapid prototyping for software-defined networks," in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, Hotnets-IX, pp. 19:1–19:6, ACM, 2010.

[138] N. Feamster, H. Balakrishnan, J. Rexford, A. Shaikh, and K. Merwe, "The Case for Separating Routing from Routers," in *ACM SIGCOMM Workshop on Future Directions in Network Architecture (FDNA)*, (Portland, OR), Sept. 2004.

[139] A. Gämperli, V. Kotronis, and X. Dimitropoulos, "Evaluating the effect of centralization on routing convergence on a hybrid BGP-SDN emulation framework," in *Proceedings of the 2014 ACM Conference on SIGCOMM*, SIGCOMM '14, (New York, NY, USA), pp. 369–370, ACM, 2014.

[140] A. Gamperli, "Evaluating the effect of SDN centralization on Internet routing convergence," Master's thesis, ETH Zurich, 2014.

[141] H. Yan, D. A. Maltz, E. Ng, H. Gogineni, H. Zhang, and Z. Cai, "Tesseract: A 4d network control plane," in *Proceedings of the 4th USENIX Conference on Networked Systems Design & Implementation*, NSDI'07, (Berkeley, CA, USA), pp. 27–27, USENIX Association, 2007.

[142] A. Greenberg, G. Hjalmtysson, D. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan, and H. Zhang, "A clean slate 4d approach to network control and management," *SIGCOMM Comput. Commun. Rev.*, vol. 35, pp. 41–54, Oct. 2005.

[143] S. Vissicchio, L. Vanbever, and J. Rexford, "Sweet little lies: Fake topologies for flexible routing," in *Proceedings of the 13th ACM Workshop on Hot Topics in Networks*, HotNets-XIII, (New York, NY, USA), pp. 3:1–3:7, ACM, 2014.

[144] R. Bennesby, P. Fonseca, E. Mota, and A. Passito, "An Inter-AS Routing Component for Software-Defined Networks," in *NOMS, IEEE*, pp. 138–145, 2012.

[145] R. Bennesby, E. Mota, P. Fonseca, and A. Passito, "Innovating on interdomain routing with an inter-SDN component," in *IEEE 28th International Conference on Advanced Information Networking and Applications (AINA)*, pp. 131–138, May 2014.

[146] V. Kotronis, A. Gämperli, and X. Dimitropoulos, "Routing centralization across domains via SDN: A model and emulation framework for BGP evolution," *Computer Networks*, pp. –, 2015.

[147] P. Francois, C. Filsfils, J. Evans, and O. Bonaventure, "Achieving sub-second IGP convergence in large IP networks," *SIGCOMM Comput. Commun. Rev.*, vol. 35, pp. 35–44, July 2005.

[148] D. Levin, A. Wundsam, B. Heller, N. Handigol, and A. Feldmann, "Logically centralized?: State distribution trade-offs in software defined networks," in *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, HotSDN '12, (New York, NY, USA), pp. 1–6, ACM, 2012.

[149] M. Alan, T. Holterbach, M. Happe, and L. Vanbever, "Supercharge me: Boost router convergence with SDN," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM '15, (New York, NY, USA), pp. 341–342, ACM, 2015.

[150] D. Pei, X. Zhao, D. Massey, and L. Zhang, "A study of BGP path vector route looping behavior," in *Proceedings of the 24th International Conference on Distributed Computing Systems*, pp. 720–729, 2004.

[151] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, 1995.

[152] R. Gill, R. Paul, and L. Trajkovic, "Effect of MRAI timers and routing policies on BGP convergence times," in *IEEE 31st International Performance Computing and Communications Conference (IPCCC)*, pp. 314–323, Dec 2012.

[153] Alcatel-Lucent and C. Bookham, *Versatile Routing and Services with BGP: Understanding and Implementing BGP in SR-OS*. Wiley; 1 edition, 2014.

[154] W. Lijun, W. Jianping, and X. Ke, "Modified flap damping mechanism to improve inter-domain routing convergence," *Comput. Commun.*, vol. 30, pp. 1588–1599, May 2007.

[155] B. Zhang, D. Pei, D. Massey, and L. Zhang, "Timer interaction in route flap damping," in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, ICDCS '05, (Washington, DC, USA), pp. 393–403, IEEE Computer Society, 2005.

[156] "Scalable Simulation Framework." http://www.ssfnet.org/.

[157] T. Chandra, R. Griesemer, and J. Redstone, "Paxos made live: An engineering perspective," in *Proceedings of the Twenty-sixth Annual ACM Symposium on Principles of Distributed Computing*, PODC '07, (New York, NY, USA), pp. 398–407, ACM, 2007.

[158] W. Lijun, W. Jianping, and X. Ke, "Utilizing route correlation to improve BGP routing convergence," in *9th International Conference on Telecommunications*, pp. 211–218, June 2007.

[159] T. Griffin, B. Shepherd, and G. Wilfong, "Policy disputes in path-vector protocols," in *Proceedings of the Seventh Annual International Conference on Network Protocols*, ICNP '99, (Washington, DC, USA), pp. 21–, IEEE Computer Society, 1999.

[160] C. Labovitz, A. Ahuja, R. Wattenhofer, and S. Venkatachary, "The impact of Internet policy and topology on delayed routing convergence," in *Proceedings of IEEE Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies-INFOCOM*, vol. 1, pp. 537–546, 2001.

[161] R. Sami, M. Schapira, and A. Zohar, "Searching for stability in interdomain routing," in *IEEE INFOCOM 2009*, pp. 549–557, Apr. 2009.

[162] T. Griffin and G. Wilfong, "Analysis of the MED oscillation problem in BGP," *Proceedings of the 10th IEEE International Conference on Network Protocols*, pp. 90–99, 2002.

[163] L. Cittadini, M. Rimondini, S. Vissicchio, M. Corea, and G. Di Battista, "From theory to practice: Efficiently checking BGP configurations for guaranteed convergence," *IEEE Transactions on Network and Service Management*, vol. 8, pp. 387–400, Dec 2011.

[164] Y. Lee, I. Park, and Y. Choi, "Improving TCP performance in multipath packet forwarding networks," *Journal of Communications and Networks*, vol. 4, pp. 148–157, June 2002.

[165] J. He and J. Rexford, "Toward internet-wide multipath routing," *IEEE Network*, vol. 22, pp. 16–21, Mar 2008.

[166] R. Bless, G. Lichtwald, M. Schmidt, and M. Zitterbart, "Fast scoped rerouting for BGP," in *11th IEEE International Conference on Networks (ICON)*, pp. 25–30, Sept. 2003.

[167] T. Bates, R. Chandra, and E. Chen, "BGP route reflection: An alternative to full mesh internal BGP (IBGP)," RFC 4456, April 2006.

[168] R. Musunuri and J. Cobb, "A complete solution for iBGP stability," in *Communications, 2004 IEEE International Conference on*, vol. 2, pp. 1177–1181 Vol.2, June 2004.

[169] A. Basu, C. Ong, A. Rasala, B. Shepherd, and G. Wilfong, "Route oscillations in i-BGP with route reflection," in *Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '02, (New York, NY, USA), pp. 235–247, ACM, 2002.

[170] T. Klockar and L. Carr-Motyckova, "Preventing oscillations in route reflector-based i-BGP," in *Proceedings of 13th International Conference on Computer Communications and Networks- ICCCN*, pp. 53–58, Oct. 2004.

[171] A. Rawat and M. Shayman, "Preventing persistent oscillations and loops in IBGP configuration with route reflection," *Comput. Netw.*, vol. 50, pp. 3642–3665, Dec. 2006.

[172] "Quagga." http://www.nongnu.org/quagga/.

[173] "POX." http://http://www.noxrepo.org/pox/about-pox//.

[174] T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama, and S. Shenker, "Onix: A distributed control platform for large-scale production networks," in *Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation*, OSDI'10, (Berkeley, CA, USA), pp. 1–6, USENIX Association, 2010.

[175] M. Caesar, D. Caldwell, N. Feamster, J. Rexford, A. Shaikh, and J. van der Merwe, "Design and implementation of a routing control platform," in *Proc. of the 2nd Conference on Symposium on Networked Systems Design & Implementation - Volume 2*, NSDI'05, pp. 15–28, USENIX Association, 2005.

[176] A. Gupta, L. Vanbever, M. Shahbaz, S. Donovan, B. Schlinker, N. Feamster, J. Rexford, S. Shenker, R. Clark, and E. Katz-Bassett, "SDX: A software defined Internet exchange," in *Proceedings of the 2014 ACM Conference on SIGCOMM*, SIGCOMM '14, (New York, NY, USA), pp. 551–562, ACM, 2014.

[177] C. Rothenberg, M. Nascimento, M. Salvador, C. Corrêa, S. Cunha de Lucena, and R. Raszuk, "Revisiting routing control platforms with the eyes and muscles of software-defined networking," in *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, HotSDN '12, (New York, NY, USA), pp. 13–18, ACM, 2012.

[178] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker, "NOX: towards an operating system for networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, pp. 105–110, July 2008.

[179] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O'Connor, P. Radoslavov, W. Snow, and G. Parulkar, "ONOS: Towards an open, distributed sdn os," in *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*, HotSDN '14, (New York, NY, USA), pp. 1–6, ACM, 2014.

[180] P. Lin, J. Hart, U. Krishnaswamy, T. Murakami, M. Kobayashi, A. Al-Shabibi, K. Wang, and J. Bi, "Seamless interworking of SDN and IP," *SIGCOMM Comput. Commun. Rev.*, vol. 43, pp. 475–476, Aug. 2013.

[181] "ExaBGP." https://github.com/Exa-Networks/exabgp.

[182] B. Schlinker, K. Zarifis, I. Cunha, N. Feamster, and E. Katz-Bassett, "Peering: An as for us," in *Proceedings of the 13th ACM Workshop on Hot Topics in Networks*, HotNets-XIII, (New York, NY, USA), pp. 18:1–18:7, ACM, 2014.

[183] M. Feilner, *OpenVPN: Building and Integrating Virtual Private Networks: Learn How to Build Secure VPNs Using This Powerful Open Source Application*. Packt Publishing, 2006.

[184] RIPE, "Routing Information Service." https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris, 2018.

[185] Oregon, "Route Views." http://www.routeviews.org/routeviews/, 2018.

[186] L. Subramanian, S. Agarwal, J. Rexford, and R. Katz, "Characterizing the Internet hierarchy from multiple vantage points," in *Proceedings of IEEE Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies- INFOCOM*, vol. 2, pp. 618–627, 2002.

[187] B. Al-Musawi, P. Branch, and G. Armitage, "Bgp anomaly detection techniques: A survey," *IEEE Communications Surveys Tutorials*, vol. 19, pp. 377–396, Firstquarter 2017.

[188] R. V. Oliveira, R. Izhak-Ratzin, B. Zhang, and L. Zhang, "Measurement of highly active prefixes in bgp," in *GLOBECOM '05. IEEE Global Telecommunications Conference, 2005.*, vol. 2, pp. 5 pp.–, Nov 2005.

[189] G. Aceto, A. Botta, P. Marchetta, V. Persico, and A. Pescapé, "A comprehensive survey on internet outages," *Journal of Network and Computer Applications*, vol. 113, pp. 36 – 63, 2018.

[190] R. Beacons, "RIS Routing Beacons." https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris/ris-routing-beacons, 2018.

[191] L. Blunk, M. . Karir, and C. Labovitz, "Multi-threaded routing toolkit (mrt) routing information export format," RFC 6396, Oct. 2011.

[192] N. Elgendy and A. Elragal, "Big data analytics: A literature review paper," in *Advances in Data Mining. Applications and Theoretical Aspects* (P. Perner, ed.), (Cham), pp. 214–227, Springer International Publishing, 2014.

[193] J. Obstfeld, X. Chen, O. Frebourg, and P. Sudheendra, "Towards near real-time BGP deep analysis: A big-data approach," *CoRR*, vol. abs/1705.08666, 2017.

[194] Keras, "Keras." https://keras.io/, 2015.

[195] N. Reimers and I. Gurevych, "Optimal hyperparameters for deep lstm-networks for sequence labeling tasks," *CoRR*, vol. abs/1707.06799, 2017.

[196] TensorFlow, "TensorFlow." https://github.com/tensorflow/tensorflow, 2015.

[197] B. L. G. Myttenaere, Boris Golden and F. Rossi, "Mean absolute percentage error for regression models," *Neurocomputing*, vol. 192, p. 38, 2016.

[198] "RYU Controller." https://github.com/osrg/ryu.

[199] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," in *Proceedings of the 9th International Conference on Neural Information Processing Systems*, NIPS'96, (Cambridge, MA, USA), p. 155–161, MIT Press, 1996.

[200] R. Anwar, H. Niaz, D. Choffnes, I. Cunha, P. Gill, and E. Katz-Bassett, "Investigating interdomain routing policies in the wild," in *Proceedings of the 2015 Internet Measurement Conference*, IMC '15, (New York, NY, USA), pp. 71–77, ACM, 2015.

[201] A. Passito, E. Mota, R. Bennesby, and P. Fonseca, "Agnos: A framework for autonomous control of software-defined networks," in *2014 IEEE 28th International Conference on Advanced Information Networking and Applications*, pp. 405–412, May 2014.