On iBGP Multicasting in Software Defined Networks

by

Ukemeobong Okon Bassey

Thesis submitted in partial fulfillment of the requirements
For the Master of Applied Science degree in
Electrical and Computer Engineering

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Abstract

In the Internet today, learnt prefixes are forwarded within autonomous systems (ASs) over internal Border Gateway Protocol (iBGP) sessions. Existing schemes for iBGP routing include the full-mesh (FM) solution, route reflection (RR) solution and confederation. Optimal prefix routing and route diversity are the main strength of the FM solution. However, it is rarely employed in a large networks due to its deficiency in aspects including scalability and large Routing Information Base (RIB) size requirement of routers. This is due to the fact that routers in this topology are required to peer with every other router within the AS. To combat these challenges, the RR scheme provides solution for scalability by decreasing the iBGP sessions requirement. Notwithstanding, the RR solution has its own challenges which includes reduced route diversity, introduction of divergence and forwarding anomalies. Also, the FM optimality may be lost since the Route Reflectors are responsible for reflecting the learnt prefixes to their corresponding clients based on its partial view of the network. The concept of Software Defined Networking (SDN) entails decoupling of the control plane from the forwarding plane such that the control plane is logically centralized benefiting from an overall knowledge of the network for decision making. In this work, we propose a solution based on multicasting which employs relay nodes in the iBGP message dissemination. Our solution brings session management scalability and minimization of duplicate prefix announcement through elimination of peer sessions deemed unnecessary. SDN controller is employed to configure and coordinate the multicast tree.

Acknowledgements

First, I would like to express my sincere gratitude to Prof. Amiya Nayak for his continuous support, motivation, advice and immense knowledge throughout my thesis work. I could not have imagined having a better mentor for my Masters study.

Also, I would like to thank my family and friends for providing me with unfailing support and encouragement throughout the process of researching and writing my thesis. This accomplishment would not have been possible without them. Thank you.

Contents

List of Tables

List of Figures

List of Abbreviations

BGP	Border Gateway Protocol
iBGP	Internal Border Gateway Protocol
eBGP	External Border Gateway Protocol
IGP	Interior Gateway Protocol
TCP	Transport Control Protocol
UDP	User Datagram Protocol
CPU	Central Processing Unit
RIB	Routing Information Base
IGRP	Interior Gateway Routing Protocol
IS-IS	Intermediate System-to-Intermediate System Protocol
SDN	Software Defined Networking
ASBR	Autonomous System Border Router
RN	Relay Node
RR	Route Reflection
CIDR	Classless Inter-Domain Routing
FM	Full-Mesh
MED	Multi-Exit Discriminator
API	Application Program Interface
XORP	eXtensible Open Router Platform (Open source routing protocol)
GPL	General Public License

Chapter 1

Introduction

1.1 Motivation

Connectivity and message dissemination in the network is handled by routing protocols. Routing protocols in a network such as the Internet enable network devices, e.g routers to connect with each other, send messages across the network and make good decisions based on messages learnt from other routers. In a nutshell, routing among routers involves first the discovery stage. Here, the routers who are active discover and learn about the existence of other routers. After discovery, messages are exchanged and stored in the routers based on the algorithm specified by the network administrators. Following this, routers can make decisions based on the locally stored information in collaboration with their inbuilt algorithms. Various routing protocols exist, and routers can be seen as multiprotocol elements since they are able to support quite a number of protocols. Some of the protocols include IGRP (Internet Gateway Routing Protocol), Enhanced IGRP, IS-IS (Intermediate System to Intermediate System protocol), BGP (Border Gateway Protocol), etc. Our focus is on BGP, particularly the iBGP.

As per BGP specification [24], internal routers are only allowed to reflect routes learnt

over external BGP sessions to its iBGP peers and vice versa. BGP deployment revolves around three schemes, namely, BGP full-mesh topology, BGP confederation and BGP route reflectors. These three have their strong and weak points, each trying to improve the weakness of the others. To ensure complete message dissemination to each node and to enable all routers to make optimal routing decisions, the full-mesh scheme requires that all nodes peer with each other. The implication of this is that there is a high number of sessions at each router in the real network. For this type of network, the need to configure each router for every network change becomes burdensome. Hence, the full-mesh scheme is rarely employed in a large network. Coupled with the scalability issues, is the great number of duplicates which are mostly unnecessary as stated by the authors in [6]. These duplicates require large RIB size and large CPU capacity. An improvement on the scalability issue was brought about by the route reflector (RR) scheme. In this scheme, peers are no longer required to establish BGP sessions with all other peers.

The RR scheme tackles these challenges by selecting specific nodes to serve as route reflection points. As a result, only the route reflectors are required to have a fully meshed iBGP sessions with each other. The RR nodes have client and non-clients. It reflects route advertised by its clients to both clients and non-client. In this way, the need for full mesh peering is eliminated. The Route Reflection concepts elects multiple RR nodes to avoid cases of single point failure. Studies in [35], [19] have shown that the RR while solving full-mesh scheme challenges can also lead to unnecessary filtering of required prefixes by the route reflectors. One of the reasons for this is due to the fact that the route reflectors make their route selection decisions based on their partial knowledge of the network topology. Lack of predefined configuration guidelines for networks using route reflection scheme is another persisting issue. Although problem detection and checking for correctness in route reflector iBGP graphs have been discussed in [4], [16],

[11], it is still a point of concern. Investigation carried out by [6], [20], [14] and others indicate the impact of churn in the network.

The case of suboptimal egress point occurs when route reflectors advertise a route that is not optimal from the perspective of the receiving client in terms of IGP cost. As a result, the depending peer is unable to learn its best exit point. The route reflection network can get into an unpredictable state when best egress point selected by routers differ due to the other in which the BGP prefixes arrive. A recently proposed scheme in [7] employs multicasting for redistribution of BGP prefixes. Although this work brings an improvement to BGP message dissemination, it faces the challenge of unreliability in BGP prefix distribution and relatively high number of established BGP sessions. Also, the dissemination of unnecessary prefix duplicates may still occur. We target to limit the BGP sessions and BGP prefix duplication which may occur. [6] concludes that duplicate announcement is a significant contributor to network churn. They show that these duplicates exert a lot of burden on the CPU through the generated churn. Another study in [1] shows that BGP consumes about or over 60 percent of core routers CPU cycles and high CPU utilization can disrupt other network protocol task. Following the mentioned cases, we focus our work on the objectives stated below.

1.2 Objectives

Given the impact of churn in the network, we aim to minimize prefix duplicates by eliminating unnecessary peering sessions in the multicast tree coupled by filtering performed at selected nodes termed relay nodes. This will lead to reduction in the RIB size of BGP speakers, therefore reducing the burden on the processing unit.

In order to allow for the deployment of this scheme in large networks, our objective includes improvement of the scalability property through minimizing internal BGP peer-

ing sessions. Even so, proper message dissemination of prefixes is necessary if nodes are to make optimal routing decisions, as a result we will target the preservation of full-mesh optimality and correctness in the network.

Finally, we will aim at establishing only feasible iBGP sessions between peers at all times by ensuring that the BGP sessions follow the IGP network links. Following this model will limit the convergence time since our algorithm strictly follows the network IGP links which is known for its fast reconvergence property.

1.3 Contribution

- We propose a multicast relay-based algorithm for BGP prefix dissemination coordinated by the SDN controller. Our algorithm establishes sessions only between peers directly linked through IGP network links. This ensures that all iBGP sessions are feasible. Also, with our algorithm, we ensure through multicasting that routers are aware of at least one path to each external destination while we store the backup routes at the relay nodes.

- We present an alternate scheme which aims at distributing the relay node filtering workload across multiple selected relay nodes to minimize single point failure and attacks. We employ the concept of SDN to ensure that routing decisions are based on the knowledge of full network topology. This approach is scalable thus making the deployment feasible in a larger and more realistic network.

1.4 Thesis Organization

This thesis is organized as follows:

- *Chapter 2* describes the concept of SDN and the working principle of BGP. We give an overview of the existing and ongoing related works in BGP schemes including FM scheme, RR scheme and confederation. We point out the contributions and caveats of these schemes.

- *Chapter 3* describes the BGP prefix duplicate problem in legacy multicasting. Our proposed algorithm is explained in detail through architectural and mathematical instances and examples. Comparison is made with some existing schemes like full-mesh and route reflectors.

- *Chapter 4* elaborates on the implementation tools and simulation of the proposed algorithm. We present the tools and acquired results based on the algorithm evaluation.

- *Chapter 5* presents an alternative way of selection of relay nodes for the iBGP network in order to eliminate single point failure, attacks and overloading. We further elaborate on the strength and weakness of the selection process.

- *Chapter 6* gives a summary of the work done and discusses improvements of the proposed work including reliability issues, coincidental of autonomous system border router and relay node leading to overload point in a node, and the concept of cluster based relay node selection. We conclude by pointing to the direction of future works.

Chapter 2

Background and Related Work

In this chapter, we give an overview of the concept of SDN, BGP process and the multi-casting message dissemination. We further present existing work towards enhancing BGP performance. In particular, sectors including BGP routing mechanism, newly proposed architecture and selection of route reflectors and optimal egress point are considered.

2.1 Software Defined Networking

The concept of SDN has attracted lots of attention leading to its vast application in the network. Simply, SDN can be seen as the abstraction of network device intelligence into a central logic hub to collectively serve those devices e.g switches in the network. One of the key drivers of SDN is simplicity. Legacy switches support a variety of protocols. This leads to a "soup of protocols" [11] and sometimes these protocols compete for resources. The idea of SDN takes the logic out of the switch boxes into a central intelligent software-based hub [17]. This concept results in several advantages. One of such is that network administrators are now able to program their routers in a more customized form to fit their respective application needs. Additionally, the controller which is the

central point for switch logic has a complete view of the network leading to optimal routing and forwarding decisions. This beats the legacy switch concept where each switch had to make a decision based on its partial view of the network. Another area of interest of SDN is reduction in cost given that switches arrive as dumb boxes waiting to be programmed. Cost in this sense includes, but not limited to, switch design cost, purchasing cost, building and operation cost. With the legacy switches, operation cost rose due to the fact that each vendor had to write its own codes for their switches. With SDN frameworks, codes establishing common functionalities and protocols can be reused therefore decreasing the operation/development cost and time. Interestingly, SDN also presents automation of some network task. Automation makes the network operations agile enabling appropriate responses to network changes. In comparison, the concept of legacy switches also leads to inflexibility because network operators' choices are limited to what the networking devices vendors provide. As a network operator, you have to work around the provided device operation and capability to accomplish your unique application needs. However, with SDN model, network devices are programmed to work around each unique network application needs. The concept of SDN is being applied widely in BGP works including [29], [37] and [30].

2.1.1 SDN Architecture

The architecture of SDN comprises three basic components. They are the application layer, the control layer and the forwarding or data plane layer [11]. These layers all have a part to play in the networking life cycle of switches. The application layer lies above the control layer and it is where the networking applications run. Needs of each networking applications may differ, these needs are sent to the controller layer as policies to be implemented in the network. The controller layer lies between the application layer

and the forwarding layer. This layer coordinates and organizes network forwarding plane elements. It is responsible for taking the policies sent by the network application and turning them into rules. These rules are further translated into the switches flow table for decision making. The bottom layer is called the data plane, where the actual forwarding occurs [13]. Each switch has one or more flow tables where each incoming packet is matched and the corresponding action is executed. SDN architecture is depicted in Figure 2.1.

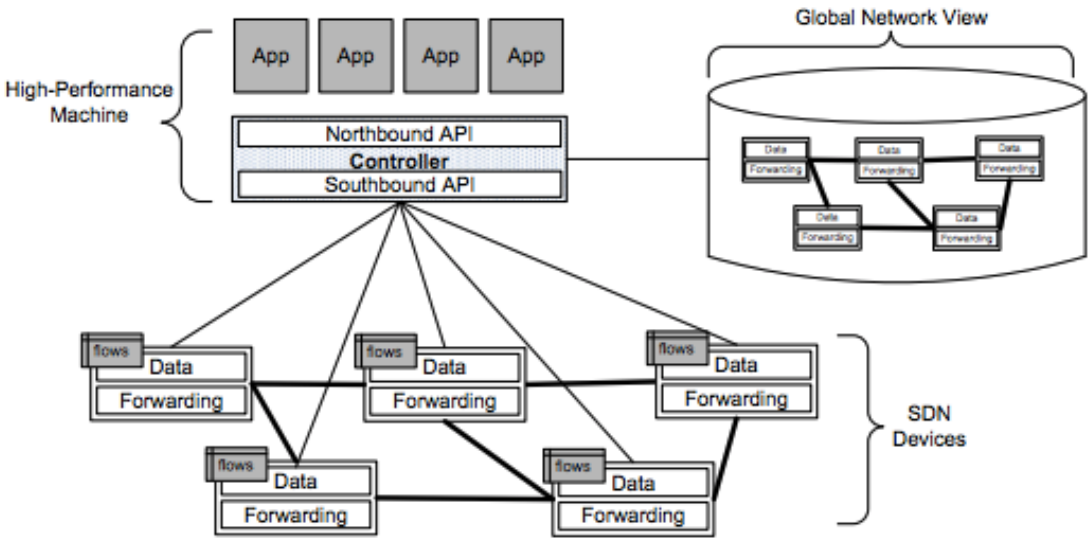

Figure 2.1: SDN Architecture from [5]

2.1.2 SDN Strength and Weakness

SDN proposed solution even though promising has its own pitfalls. One of such is in the area of latency. From the SDN architecture as presented in Figure 2.1, some packets may have to undergo round trip travel from the forwarding element to the SDN controller and back. Packets for which the switch has no policy on how to handle falls into

this category. As the number of switches increase without controller capability increase, the delay of servicing and processing such packet by the controller to each switch may increase. A proposed solution may be to use multiple synchronized controllers or to limit the number of no-policy packets sent to the controller [36]. The idea of a centralized SDN controller also comes with the issue of single-point failure. This implies that once the controller is down the networking co-ordination is lost since all forwarding elements are tied to the controller. To battle this, redundancy schemes such as hot plugging have been proposed. However, they may pose a cost issue since it involves having standby controllers who have to mirror or take snap shots of data sets to ensure synchrony and consistency during recovery. And of course, there is the scalability barrier since the load and pressure of responsibility on the controller increases with the increase in number of dependent forwarding element. This may be solved with implementing clusters of controllers where load balancing is performed dynamically among the controllers. The area of security in the SDN controller has also attracted lots of research work including [34], [9] and [5]. Its importance cannot be undermined because a single logic point means that attackers can easily direct all attacks to the centralized controller and so the robustness of the SDN controller should withstand these anticipated attacks.

2.2 Multicasting Concept

Multicasting presents the ability to send messages from a source to multiple destinations [8]. The multicast concept is depicted in Figure 2.2, where message from the server is sent to multiple hosts. This idea is to shift the burden of message dissemination away from a particular router by having respective routers duplicate the messages when next hop destination is greater than one. In multicasting, there is a group head and a

corresponding group address. For every message dissemination, the sender appends the group address to the message to be sent [15]. Group members who are listening receive these messages and send it down the multicast tree.

Generally, multicast applications require UDP transport protocol, high bandwidth with low jitter during the transfer of messages. The issue of unreliability of multicasting given its UDP transport protocol has been handled in research works including [18] [25] and [22]. Multicast technology is mostly employed in broadcasting of video, media streaming and video conferencing. We introduce in this thesis a solution based on multicasting which employs an elected relay node to redistribute all necessary prefixes to internal nodes while holding back the duplicates. The duplicates are only employed as back-up route when the optimal route fails.

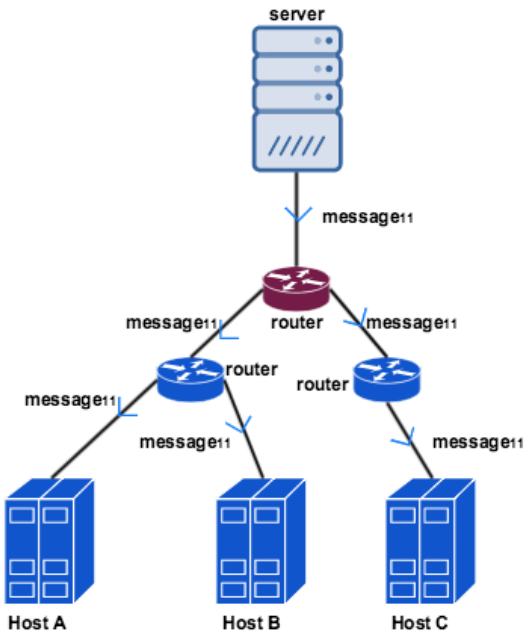

Figure 2.2: Multicasting Example

2.3 Border Gateway Protocol

2.3.1 BGP Working Principle

iBGP sessions are established between nodes of the same AS. Nodes participating in the BGP process are called BGP peers or BGP speakers. To create BGP sessions between BGP peers a couple of messages are exchanged. This includes the OPEN message, KEEP-Alive message, UPDATE message and the notification message [24]. The life cycle of each of these messages are triggered when events occur. For instance, the open message is sent once a TCP connection has been established, the update message is sent when BGP peers want to advertise routes to other peers, the keepalive message is sent to prevent the connection from closing at the expiration of timer, and the notification message is used once an erroneous situation is detected. Once the BGP session is established among peers, reachability information can be advertised to peers.

A BGP speaker has Loc-RIB, Adj-RIB-in table as well as Adj-RIB-out table [24]. The Loc-RIB consist of routes that this BGP speaker has selected for its own local use. It is preeminent that these selected routes be valid and reachable. The Adj-RIB-in table consist of all advertised routes that were received by the peer before filtering, while the Adj-RIB-out has the routes that have been selected for advertisement to other BGP peers. The BGP speaker has the freedom to choose which route to advertise, however, only one advertisement per destination prefix is allowed. BGP-4 being the latest version comes with the support for Classless Inter-Domain Routing(CIDR) and aggregation of routes [24].

2.3.2 BGP Decision Process

The Internet is made up of ASs interconnected with each other. These ASs exchange reachability information using the BGP [24]. Inside each AS, the learnt prefix is propagated using the internal BGP called iBGP. BGP speakers advertise selected reachability information over sessions with other BGP peers. An iBGP session is strictly made up of two BGP speakers within the same AS [24]. The ASBRs learn prefixes external to their local AS and are responsible for redistributing these prefixes to other internal routers. All routers have a routing table and are able to keep multiple routes for a given prefix. An iBGP peer is only permitted to advertise one route per prefix destination. Prefixes advertised are those that have been filtered by the BGP selection process.

Given a received route set, S, towards a given prefix, BGP selection process is depicted in Figure 2.3. The `local_preference` [24] attribute indicates the internal AS preference of an advertised prefix. If there is more than one route, the shortest `AS_path` is selected from the remaining set. `AS_path` attribute also serves to detect loops in the network. Lowest Multi-Exit Discriminator [24] attribute is employed when ASs are multi-connected, that is two or more links connecting a pair of ASs. The selection process goes further to select routes learnt through eBGP over that of iBGP. Lastly, IGP cost metric are utilized for selection, and in case of multiple routes still remaining a tie-break mechanism such as lower ip address can be used to pick a route.

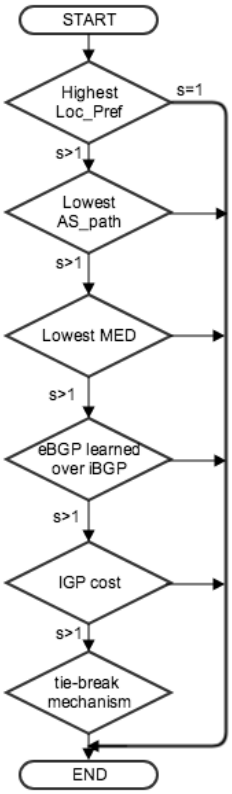

Figure 2.3: BGP Decision Process

Some of the research works performed with regards to improving the efficiency of advertising destination reachability information are discussed later. Each scheme aims at improving already existing mesh, route reflection and confederation schemes. Some of these works are discussed below.

2.4 BGP Prefix Redistribution Schemes

2.4.1 Full-Mesh Scheme

The FM BGP scheme requires that all BGP speakers in the network peer with all other BGP speakers [31]. This scheme ensures that peers are one-hop from each other, thereby reducing latency and reconvergence time. Yet, in large networks its ability to scale is

very poor. As result, schemes including confederation and RR have been proposed to fulfill these needs.

2.4.2 Confederation Scheme

The confederation scheme proposes the division of an AS into children ASs, while the AS is still being advertised as a single system to peers external to the AS [31]. Each unit of the child AS is called a confederation and is fully meshed. Its main purpose was to reduce the FM size of the network. However, this scheme results in unnecessary routes duplication and complexity in the AS. Also, there is the issue of maintenance overhead in this type of network [31].

2.4.3 Route Reflection Schemes

The RR scheme allows a selected set of BGP peers termed "Route Reflectors" to re-advertise prefixes to iBGP peers [2]. The route reflectors have clients and non-clients [2]. Prefixes received from clients are advertised to all other clients and non-clients, however prefixes received from non-clients are only advertised to clients. The route reflectors are only required to fully mesh with other route reflectors [2]. This reduces the number of sessions in the network and increases simplicity. However, in certain network topology the best next hop from the perspective of the route reflector is not necessarily the best for its clients who are depending on this RR decision. This could lead to route reflector anomalies including non-optimal exit point, route oscillation and routing loops [26].

Route oscillation in route reflector scheme takes place when nodes choose next hops which do not belong in the same cluster and so the network is unable to get into a stable state. Here, the node keeps selecting different next hop which causes a change in next hop of other nodes continuously creating a cycle of best path change. According to [26],

this is mostly due to misconfiguration. Routing loops on the other hand occurs when generated packets passes through its source more than once. Below, we give a summary of some of the related works performed with this scheme.

2.4.3.1 Route Diversity in RR Networks

According to [21], the main aim of this work was to improve route diversity by making available an alternative route per destination at the cost of adding more BGP sessions. The algorithm aims at providing each router with the knowledge of at least two ways to reach a given destination if and only if prefixes are received into the network at different ASBRs. While meeting this goal, they keep the number of sessions established well below that of the BGP full mesh topology scheme. Specifically, the proposed algorithm increases diversity by adding more iBGP sessions to the already existing route reflector topology. This algorithm relies on the IGP topology of the network, eBGP route received at the border routers, iBGP route reflection network and the concept of external best route [10]. This scheme calculates a number of iBGP sessions to be added to the route reflector network by selecting and adding an iBGP session between an ASBR and a BGP peer if the ASBR adds a redundant path to the routes already learnt by the BGP peer.

The authors in [10] conclude that route diversity can be established using this algorithm, however for some cases where no next hop router is found to act as a redundant path towards learnt prefixes, it is impossible to achieve this without creating additional external BGP sessions. Some caveat in this algorithm are the effect of ordering of router addition which can cause future consideration of this router as next hop diversity to be void. Also, the network operator has to implement tie-break rules to select a next hop from a pool of available next hop routers. Convergence issue is another factor to look out for while using the algorithm.

2.4.3.2 Routing Anomalies in RR Networks

Skeleton [26] idea thrives to overcome the routing anomalies of route reflector BGP scheme. This work as presented in [26] implements a subgraph employing the underlay physical graph. This way, the algorithm has the same number of nodes as the physical network and iBGP sessions are limited to the links between these nodes. As a result, validation of sessions between peers is accomplished. The case of non-optimal exit point is due to the partial network information used by the route reflector in deciding which path to advertise to its client.

The referred work in [26] presents conditions by which these RR anomalies might exist and they further prove that their proposed algorithm violates these conditions. The main idea of this work is first to calculate the IGP path between each node and ASBR within an autonomous system and then the optimal next hop is picked from this output. Therefore, internal nodes are able to receive advertised prefixes from their optimal point. Skeleton faces challenges with respect to robustness to failure and Multi-exit discriminator oscillation. The evaluation of the algorithm leads to reduction in iBGP sessions in comparison to the network physical links. However, the topology is prone to failure due to lack of redundancy scheme in its architecture.

2.4.3.3 Selection of BGP Prefixes

In legacy BGP network, peers are only allowed to advertise only one prefix per destination. However, in some networks, peers can advertise more than one prefix per destination through the BGP Add-Paths. These additional paths may be selected following different criterias. The authors in [27] show that the selection criteria employed affects network characteristics differently. As explained in [27], lack of the BGP path diversity can be traced to two factors. The first is the fact that in RR networks, a route reflector only

advertises one path to its client or non-clients. Secondly, AS routers who have preferred an iBGP route over eBGP route will only advertise the preferred route.

The proposed path selection modes in [27] are the Add-All-Paths and the Add-Group-Best Path selection. The Add-All-Paths mode proposes that a peer advertises all received routes to its iBGP peers. This way, all routers are aware of all available routes and loss of connectivity time is reduced in cases of next hop failure. A network applying this mode has no multi-exit discriminator oscillations and each router is able to make its own IGP cost metric routing decision based on full knowledge of the network. Adversely, the employment of this scheme means that each router has to store all available routes, leading to large memory consumption.

Another selection method analyzed is the Add-N-Paths [27], which suggest that a subsection of N paths should be advertised to iBGP peers. The problems with this scheme is its lack of optimal routing guarantee and avoidance of multi-exit discriminators.

2.4.3.4 Optimal Egress Points in iBGP Routing

In RR network where route reflectors are in charge of redistributing learnt routes to both clients and non-clients, it is important, yet difficult to check for correctness and route optimal selection at the RR point. This work proposes a model to check for optimal routing in a RR graph. The input of this model [4] is the IGP topology and the iBGP topology of the AS with its corresponding next hops. The algorithm checks for possible deflection that could occur between any router and each of the ASBR next hops.

The proposed model finds the optimal BGP route by checking for the following con-

ditions. Validity condition is seen as the ability of a BGP peer to learn atleast one route to its closest next hop. Optimality condition fulfilment in [4] is when a BGP router knows a path to its closest egress point. Determinism, no forwarding loop and absence of deflection are other conditions to be fulfilled if the BGP network is optimal [4], [6].

2.4.4 Other Prefix Dissemination Schemes

2.4.4.1 Shortest Path Routing

The proposed work in [3] presents a routing mechanism where BGP prefixes in the AS follow the IGP physical links in the network. The rule here is such that a BGP speaker "a" only advertises an egress point to "b", if and only if "b" can be found along the shortest path from "a" to the referred egress point. This approach [3] is based on the theory that if each router is able to learn its best egress point, then all the caveats of the route reflector scheme can be avoided.

The authors in [3], separates nodes into black and white nodes. A black node is seen as a node that blocks other nodes from getting to their optimal egress point, while white nodes are always going to learn prefixes advertised by their optimal egress point regardless of quasi-equivalent conditions. Authors in [3] prove that any node belonging to the shortest IGP path is a white node and by extension a non-blocking node. This means that nodes will always learn prefixes from their optimal egress point provided there is a white node between them.

It is recognized by the researchers in [3] that duplicate prefixes learning can still occur in a network adopting this scheme. Given that they aim at simplicity of BGP configuration, their approach seems to be backward compatible with legacy routers. Caveats of this scheme includes the computation of path cost from each router to prospective egress

point candidate. Also, this algorithms' need for filters may prove it rather expensive but for new filtering technologies like Route Target Constraint [3].

2.4.4.2 Cluster-Based Routing

Authors in [38] incorporate parallel processing to BGP since the traditional router control plane struggles to meet the growing requirements of networks. [38] proposes a scheme leveraging a cluster of routers architecture for processing and storage of routes to increase scalability. This scheme employs the balanced neighbor assignment [38] to distribute routers into clusters. Routes are synchronized within the BGP clusters over reliable BGP multicast to maintain the legacy BGP behavior. Nodes in the cluster are elected periodically to act as a scheduler node, this feature handles node failures.

The architecture in Figure 2.4 from [38] has two parts, the SDBGP agents and the internal BGP communication protocol called the iCBGP. An SDBGP agent is implemented on every cluster router. The iCBGP peering represents another form of peering session for SDBGP agents to communicate. They are able to synchronize routes through this iCBGP protocol and still retain their normal BGP behavior with other BGP peers. Multiple SDBGP agents are used to increase BGP processing speed [38].

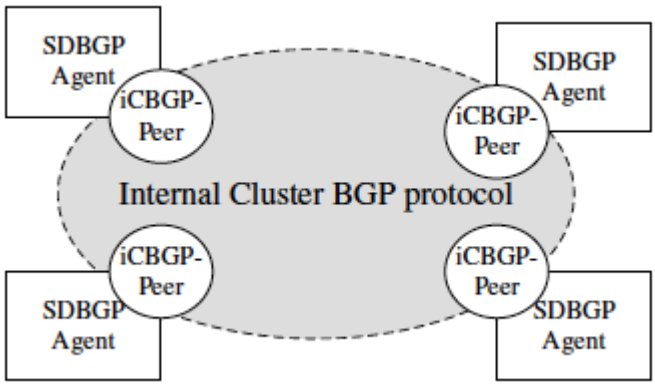

Figure 2.4: SDBGP Architecture [38]

Even though, this scheme eliminates single point failure, increases the reliability and processing speed, the potential complexity of implementing the additional peering protocols is a point of concern given that there is no standard protocol for this implementation yet.

2.4.4.3 Multicast-Based Routing

BGP multicast [7] presents an idea of BGP prefix dissemination through multicasting technology. It follows that BGP routing aims at redistributing prefixes from the ASBR, a single source to the iBGP peers within the AS, multiple destinations. This operation coincides with the multicasting operation of distributing messages to members belonging to the multicast group. Hence [7] proposes that ASBRs should be registered as group heads, each with its own unique group address. The internal routers have the responsibility of joining any group of preference.

Even though [7] reduces number of sessions compared to the FM scheme, the implementation complexity of maintaining the multicast tree is noted and assigned to the SDN controller [7]. The first challenge faced by this proposed scheme is that of unreliable message transfer performed by multicast UDP. Secondly, the network administrator has

a burden of deciding which algorithm to use in building the multicast tree as per network needs.

2.5 Conclusion

In this chapter, we gave an overview of SDN concepts, its architecture and comparison to legacy networking control architecture. SDN strength and weakness is discussed. We explained the working principle of BGP. We further gave instances of research works that has been done so far regarding the area of BGP. In each instance, we provided an overview of the proposed ideas and difficulty encountered by each idea. In the next chapter, we present our idea of BGP message dissemination.

Chapter 3

Relay-Based Multicast iBGP

In this chapter, we describe an instance of the prefix duplication issue and scalability problems with the existing schemes as pointed out previously. We present our proposed BGP message dissemination method and its working principle. Our main aim is to improve the scalability in the iBGP network and to minimize duplicates in prefixes advertised, thereby decreasing the CPU workload in processing the RIB.

3.1 Problem Statement

3.1.1 Definitions

- *Quasi-equivalence*: Routes are said to be quasi-equivalent if they cannot be distinguished using the local-pref, AS-path, origin type or MED attributes [4]. In other words, these routes are equally acceptable based on the imposed internal network policies. The routers in this AS therefore select any of the quasi-equivalent route based on their IGP cost metrics towards the corresponding BGP next-hop.

- *Routing Information Base (RIB)*: This is the routing table of a router that contains a list of routes towards prefixes.

- *Loc-RIB*: Routes that have been selected to be employed by this BGP speaker are placed here [24].

- *Adj-RIB-in*: Routes advertised to a BGP speaker by its peers are first placed in the RIB-in table. They have to be filtered before being transferred to the Loc-RIB or RIB-out table.

- *Adj-RIB-out*: Routers are free to decide whether to advertise a route or not. Routes that are to be advertised are placed in the RIB-out table. Routes are advertised in UPDATE messages [24].

- *Origin Group*: For this thesis, a node's origin group is the multicast group of which the node has the minimum IGP cost to the group head. Hence, this is the very first group the node joins on becoming active in the network.

- *Target Group*: Any other multicast group a node joins through the relay node after first joining its origin group.

- *Relay Node*: A node belonging to more than one multicast group which relays information from one group to another, typically from target group to its origin group.

For the purpose of illustration, we will consider the network shown in Figure 3.1. In the network, we have six nodes and each link represents the IGP link with its assigned cost metric.

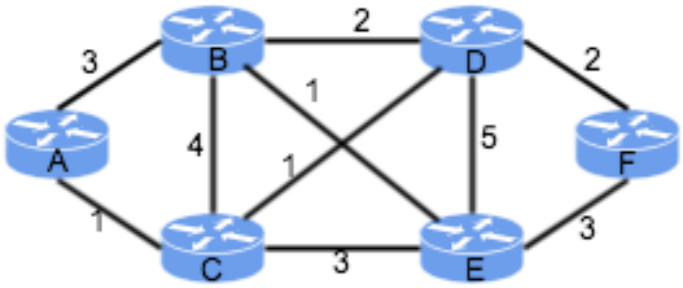

Figure 3.1: Base Topology

3.1.2 Multicast iBGP Duplicate Issue

Applying multicast routing to iBGP as in [7] decreases the number of messages and memory consumption significantly compared to Full-Mesh BGP topology. In [7], all nodes are free to join multiple groups as per interest without restrictions, in order to achieve fm-routing optimality. However, the case of duplication becomes a point of concern given the degree of freedom of the network nodes. Take the case where roots of multiple groups advertise the same prefixes assuming that there has been an occurrence of a quasi-equivalent condition. Each multicast root will advertise its own copy of this prefix to its group members. A node belonging to multiple groups in this network will receive this duplicate prefix as many times as the number of multicast tree roots who are advertising the prefix. Particularly, this node receives the same prefix from both its optimal egress point and other sub-optimal egress point. Afterwards, the node only picks the duplicate prefix advertised by its optimal egress point and discards the rest. This leads to unnecessary memory consumption and processing. Our aim is to minimize the duplicate prefix advertisements from these sub-optimal nodes.

With respect to this work, we say that there is duplication if a node, belonging to multiple groups, receives same prefix advertisement from more than one root of its

membership groups given a quasi-equivalence condition. That is to say that if a node "N" belongs to multicast groups G_1, G_2, G_3, ...G_n, and it receives same prefix advertisement that are equally acceptable from these group roots, then duplication has occurred. Our focus is on the message dissemination within the AS, so we assume that policies between autonomous systems have been fulfilled and prefixes advertised by ASBRs have already been vetted and filtered by the local network policies and processes.

3.1.3 Example

The Figures 3.2, 3.3 and 3.4 are derived from the network topology shown previously and all the network nodes are BGP speakers. We assume that BGP speakers A, B, F are ASBRs and therefore roots of multicast trees while nodes D, E, C are internal nodes free to join any multicast group of interest. Again, assuming the three internal nodes join the three multicast groups and that the ASBRs are advertising the prefixes which include W, X, Y, Z. Each prefix notation "n $\rangle\rangle$ Wn, Yn, Xn, Zn" implies that n is the root of the multicast group and "Wn...Zn" represents copy of n's advertised prefix to a particular node. Internal routers join groups in any order but for simplicity, we depict it in stages.

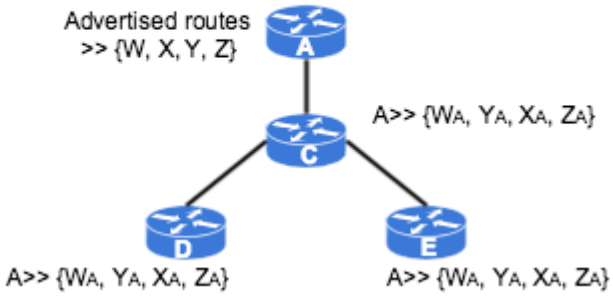

Figure 3.2: iBGP Multicast Duplicate - Stage 1

From the iBGP multicast stages as in Figures 3.2, 3.3 and 3.4, all the nodes C, D, and E join all multicast groups A, B, and F. In Stage 1, as depicted in Figure 3.2, the first ASBR, A advertises its prefixes to C, D, E.

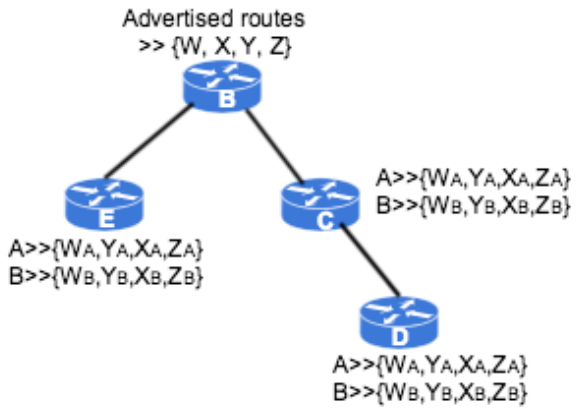

Figure 3.3: iBGP Multicast Duplicate - Stage 2

In Stage 2, as depicted in Figure 3.3, ASBR B advertises similar prefixes to its group members and F also does same towards its group members as in Figure 3.4. At the end, nodes C, D, E all receive the duplicated message with different next-hops.

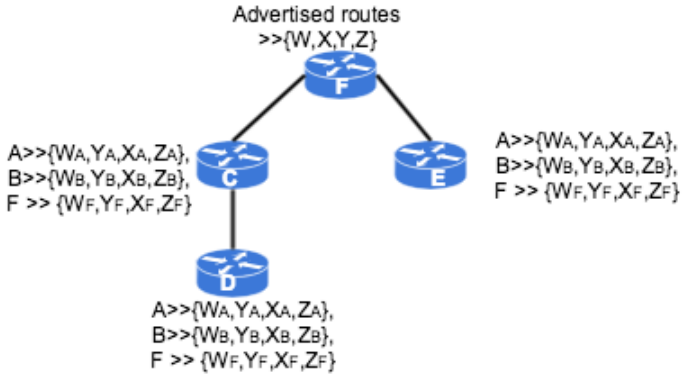

Figure 3.4: iBGP Multicast Duplicate - Stage 3

Drawing from this, it can be seen that the case of duplicate messages increases as roots advertise similar acceptable prefixes. Even though internal nodes C, E, D, receive same prefixes with different next hops, each of these nodes will only select one of this advertised route, that is the one from its optimal egress point (multicast root). For instance, node C, E, D, will select the route from next hop A, B, A, respectively according to the IGP metric shortest part given in Figure 3.1 since there are all equally acceptable.

3.2 Proposed Algorithm

Given the illustration above, we proceed to describe our proposed algorithm.

3.2.1 Relay-Based Multicast Model

To limit the duplication, a basic idea is to establish BGP sessions between roots (ASBR) of all multicast groups in mesh so that each root is able to eliminate already advertised routes from getting to its members. This is inefficient given that there might be a more cost-effective path from other tree members to the adjacent multicast group. There is

also the issue of increased sessions. In our scheme, all nodes are permitted to join at most one group at initialization. For our purpose, we define an "origin group" as the multicast group a node joins at the initialization stage, while "target groups" refer to other groups apart from the origin group that a node has interest in becoming a member.

Our objectives are achieved by electing a group member of the origin group closest to the target group to act as a relay router. This way, prefixes are advertised by relay nodes from target groups to an origin group. The prefixes advertised by the relay nodes are filtered and existing tree sessions may be utilized. The proposed idea aims to minimize duplicates by eliminating peer sessions deemed unnecessary while replacing some sessions with more efficient but lesser or equal number of peering sessions.

3.2.2 Relay-Based Multicast Algorithm

We assume that all network nodes in this AS are BGP speakers can reflect prefixes learnt over iBGP sessions.

Table 3.1: Symbols and Meaning used in algorithms and equations

Table of Notation	
Symbol	Meaning
$A_{routers}$	Set of all ASBRs
N_{nodes}	Set of all network nodes excluding ASBRs
$sp(A_j, N_i)$	Shortest path between ASBR, A_j, and node, N_i
M_{A_j}	Multicast group of ASBR A_j
R_{A_j}	Relay node of multicast group of A_j
`cum_sp`	Cumulative shortest path
Σ_{iBGP}	Total iBGP sessions for proposed algorithm
Σ_{RN}	Total relay node sessions
Σ_{FM}	Total FM sessions

Phase 1 - Initialization: For each internal node construct the shortest path to all ASBRs, they will act as multicast roots. Nodes become group members of one and only one egress point (ASBR) group which is the minimum of the shortest path in the

constructed graph. A multicast tree is formed by ASBR as root and nodes of which it qualified as the minimum shortest path cost. Peer sessions are established between all directly connected nodes in the multicast tree. The link weights in our graph G(V,E), where "V" is the set of vertices and "E" is the set of edges, can be set to value inversely proportional to link capacity (bandwidth) to discourage traffic from using low bandwidth links. At the end of this phase, all nodes belong to only one multicast group. This group is the node's "origin group", while all other groups except the origin belong to the "target group".

Phase 2 - Relay-node Selection: Relay-node Selection processing is performed per group since there is only one relay node per multicast group. For each group, say G_1, a node is elected to act as relay node between the origin and the target groups. In this case, G_1 is the origin group, while $G_2 ... G_n$ refers to the target groups.

- Elect a group member of G_1 which has the minimum cumulative shortest path to the roots of the target groups by iterating through all the group members of G_1. In our algorithm, the cumulative shortest path `cum_sp` is the sum of the cost from each node to all ASBRs which belong to the set of the target group. In other words,

$$\texttt{cum\_sp} = \sum_{i=1}^{K} sp(n, G_i)$$

  This elected node joins the new groups on behalf of origin group members. The election of relay node is performed per link metric change or occurrence of relay-node failure.

- The elected node acts as a relay router between the group pairs, transmitting BGP prefixes from target groups into its original group G_1 and filtering duplicate prefixes already advertised by its own ASBR (optimal egress point) or from another border

router with a preferred cost. The redistribution can be performed over already existing tree.

The inputs of this algorithm include the IGP topology, the eBGP routes learnt at the ASBR and the multicast trees with ASBR as root with their preferred shortest path nodes as group members. The algorithm is depicted in Algorithm 1.

Algorithm 1 SDN Controller Multicast Computation and Relay Node Selection

Input: $A_{routers} \cup N_{nodes}$: Set of all network nodes
Output: Relay node of each multicast group

```
1: Procedure_initialization():
2: Init multicast group for all A_routers
3: for N_i in N_nodes do:
4:     for A_j in A_routers do:
5:         Calculate sp(A_j, N_i)
6:         //Selection of minimum shortest path
7:         if sp(A_j, N_i) < sp(A_{j-1}, N_i) then
8:             add N_i to group M_{A_j}
9:         end if
10:    end for
11: end for
12: Procedure_RelayNode_selection():
13: A'_j refers to set of all ASBR excluding A_j
14: Set cum_sp(A_j, A'_j) = ∞
15: for node in M_{A_j} do
16:    Calculate cum_sp = sp(node, A'_j)
17:    if cum_sp < cum_sp(A_j, A'_j) then
18:        cum_sp(A_j, A'_j) = cum_sp
19:    end if
20:    Select node as relay node
21: end for
```

Consider two scenarios in the network shown in Figures 3.5 and 3.6. In the first scenario, A, F are ASBR (root of the multicast) while B, C, D, E are internal nodes and hence potential group members.

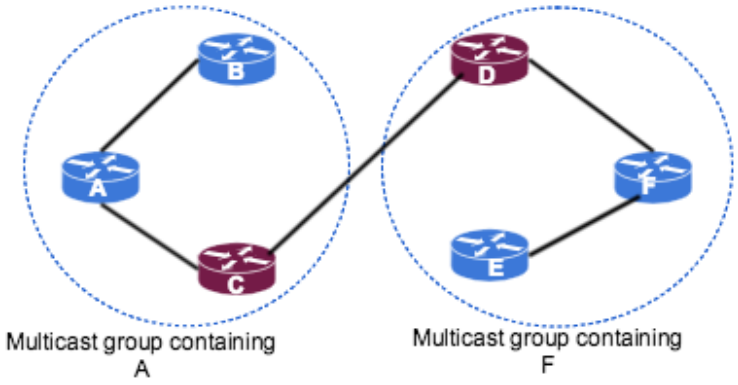

Figure 3.5: iBGP sessions in relay-based Multicast scheme

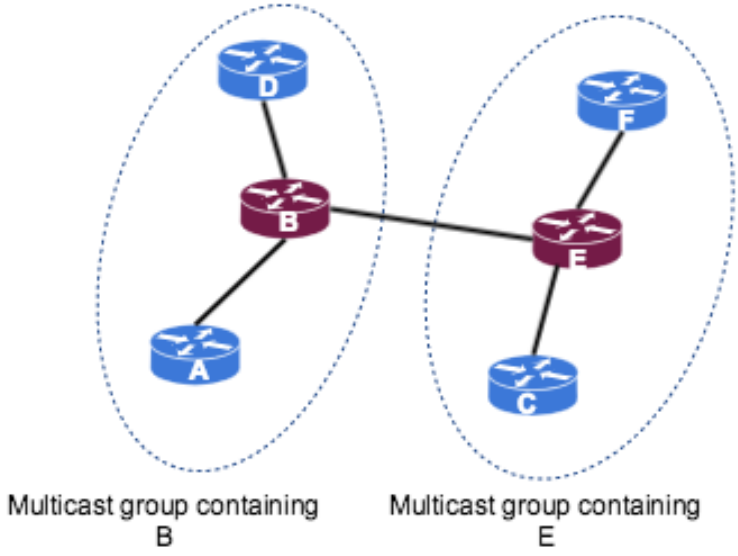

Figure 3.6: iBGP sessions in relay-based Multicast scheme

In the first phase, i.e initialization, internal nodes B, C join group containing A while D, E join group containing F respectively since that is their closest egress point with respect to IGP metric. Following this is the relay node selection phase in order to enable node E and F to receive prefixes from multicast group containing A, the iteration of Phase 2 follows and D is elected as relay node since it is the closest to the target

group root A. Similarly, C is elected as relay node for group containing A towards target group F so that B and C can receive advertisement from multicast group containing F as depicted in Figure 3.5. With the second scenario, B and E are considered multicast group root and the similar process takes place. It therefore follows that our iBGP sessions are established between peers as in Figures 3.5 and 3.6. We describe the number of peer sessions implication below.

Definition 1 *Number of iBGP session in Full-mesh scheme is defined by the number of nodes, N in the AS. Mathematically, Σ_{FM} can be given as*

$$\Sigma_{FM} = \frac{N \times (N-1)}{2}$$

Definition 2 *Number of iBGP session in proposed relay multicast-iBGP scheme is a function of ASBR which will act as the multicast group root and the number of nodes in each group. From above algorithm, the number of relay nodes is equal to multicast roots since each multicast group must have one relay node. There is a mesh session formation among the relay nodes, so number of sessions of the relay nodes can be given by the Equation below, where K is the number of multicast groups.*

$$\Sigma_{RN} = \frac{K \times (K-1)}{2}$$

Additionally, every other node i, in the network has a session M_i in the multicast tree towards the multicast root. We note that, every member of the multicast group has at most one session towards the multicast group root unlike the route reflector where a node may have sessions with multiple route reflectors. The total number of sessions in our proposed algorithm is given by

$$\Sigma_{iBGP} = \frac{K \times (K-1)}{2} + \sum_{i=1}^{K} M_i$$

Following our algorithm, the network sessions scenario is depicted in Figures 3.5 and 3.6. The full-mesh sessions for the same network is depicted in Figure 3.7, the RR scheme sessions in Figure 3.8, and legacy multicast scheme is depicted in Figure 3.9.

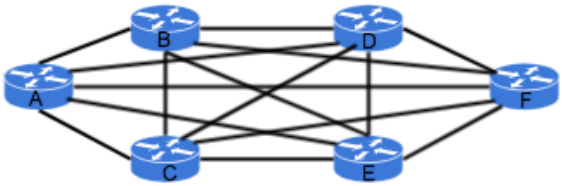

Figure 3.7: iBGP Sessions in Full-Mesh Scheme

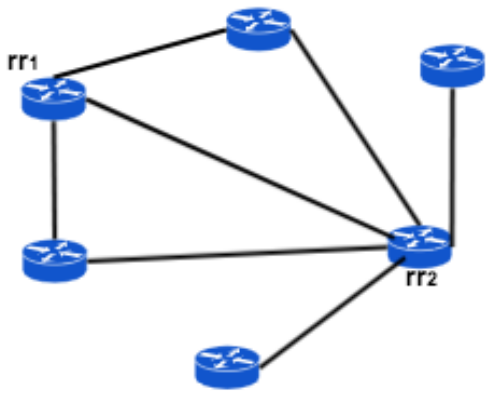

Figure 3.8: iBGP Sessions in Route Reflector Scheme

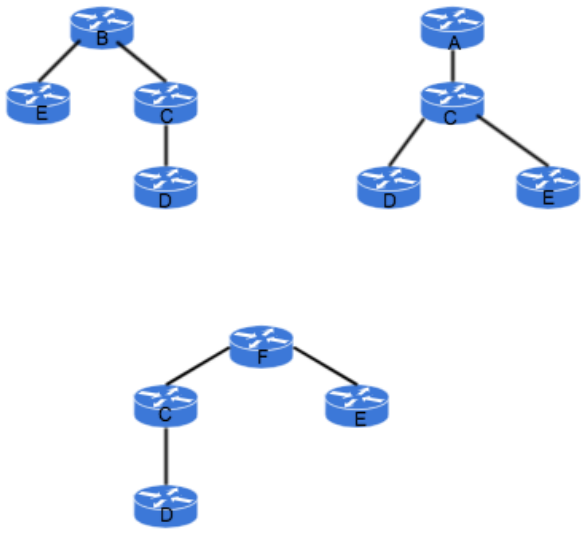

Figure 3.9: iBGP Sessions in Legacy Multicast Scheme

3.3 Joining a Multicast Group

When a node appears in a network, it receives the existing multicast group information from the SDN controller, it then finds the shortest path to all available multicast root. The minimum shortest path is selected and it joins that group as its origin group. It is now eligible for relay node election. The procedure of joining the network is shown in Figure 3.10. The following joining algorithm describes the process.

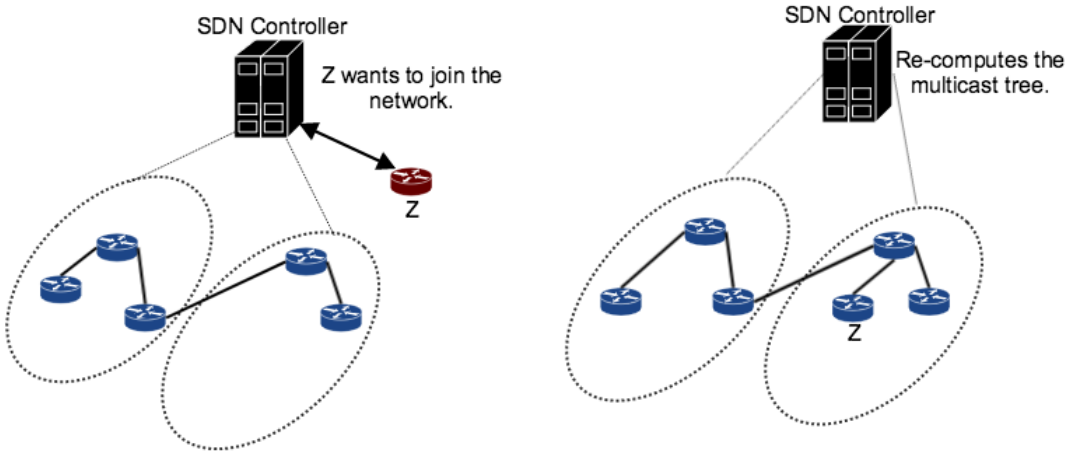

Figure 3.10: Node joining network handled by the SDN controller

```
1: Procedure_join_group(node, borderRouters)
2: Init minCostOfNode = ∞
3: Init selectedMulticastGroup = nil
4: for br in borderRouters do
5:     cost = sp(node,br)
6:     if cost < minCostOfNode then
7:         minCostOfNode = cost
8:         selectedMulticastGroup = br
9:     end if
10: end for
11: join group of selectedMulticastGroup
```

3.4 Leaving a Multicast Group

When a node leaves, all routes advertised by it are withdrawn. If the node was a relay node, then a new relay node is elected for that group, whereas if it was an ASBR (multicast root) then the multicast group is destroyed. The SDN controller keeps track of this event, since all leave and join request is sent to the SDN controller, while all other involuntary failures such as node failure are detected by the SDN controller. The process of node leaving the network is depicted in Figure 3.11. The algorithm is given below.

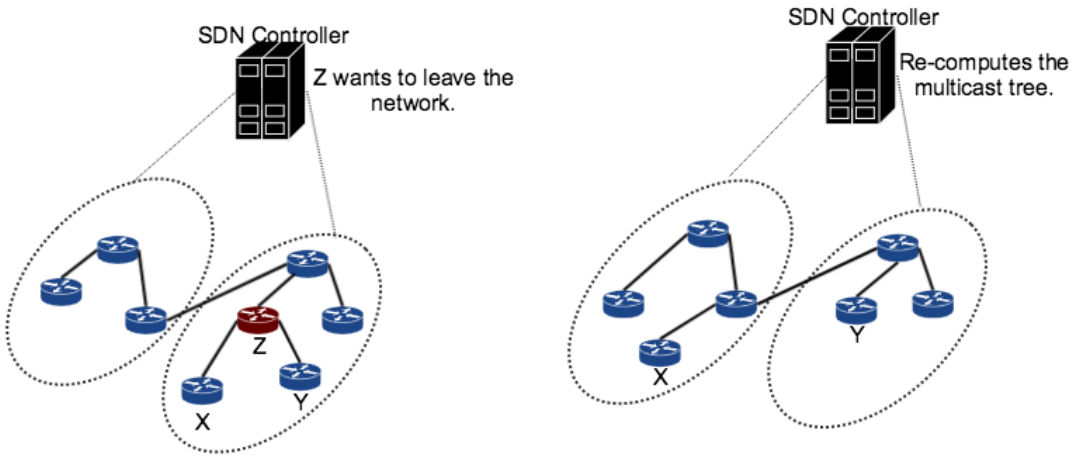

Figure 3.11: Node leaving the network handled by the controller

```
1: Procedure_leave_group(node, multicastGroup)
2: // nodeType refers to node category relayNode, borderNode, regularNode
3: if nodeType(ndoe) == "relayNode" then
4:     withdrawAdvertisedRoutes(node)
5:     relayNodeSelection(multicastGroup)
6: else if nodeType(ndoe) == "borderNode" then
7:     withdrawAdvertisedRoutes(node)
8:     deleteMulticastGroup(node)
9: else nodeType(ndoe) == "regularrNode"
10:    withdrawAdvertisedRoutes(node)
11: end if
```

3.5 Comparing Route Reflectors and Relay Nodes

As seen previously, relay nodes are allowed to join more than one group per need, including its optimal group. Their purpose is to relay prefixes from target groups to origin group members without repetition of prefixes already advertised by its origin multicast root. This operation is similar to that of a route reflector in some ways, so we proceed to compare the relay nodes in this context to route reflector nodes in Table 3.2.

Table 3.2: Comparing Route Reflectors and Multicast Relay Nodes

Route Reflector	**Relay Node**
Has clients (cluster members) and non-clients.	Group members may be considered as clients to receive prefixes from target group.
Reflect prefixes learnt from both client and non-clients.	Relay Node reflects prefixes learnt from target multicast group into its origin group.
Route Reflectors must be fully meshed with each other.	Relay Node for each origin group has a session with all other multicast groups. They are considered as fully meshed.
Appends originator ID and cluster list to prefix before advertising to prevent loops.	Appends the origin multicast group address of group members.
Selected per cluster basis.	Selected per multicast group basis, equal to the number of multicast groups.

3.6 Discussion and Conclusion

In this chapter, the relay-based iBGP multicast scheme is introduced. The coordination of the multicast tree and phase computation is handled by the SDN controller. Relay nodes are selected based on the least cumulative shortest path cost. These relay nodes act as filters and all BGP sessions follow valid IGP links.

In this scheme, the peer sessions follow the multicast tree which purely follows the shortest path IGP graph. As a result, the sessions are limited by the IGP links thereby improving scalability and eliminating the case of infeasible paths where peer sessions are created over unreachable nodes. Additionally, the IGP network converges very fast in practice meaning that the iBGP network is able to converge accordingly at this speed.

The filtering performed by the elected relay routers decreases the transmission of duplicate messages. Joining a new group through the closest member allows to utilize already existing peer sessions, eliminating the need to create new peer sessions between the joining node and every ASBR of interest. The authors in [3] proved that any router

belonging to the shortest IGP path between a node “n”, and a source “s”, is always white for the pair(n,s) and hence the transmission path is non-blocking. With reference to this work, black nodes are nodes that prevents other nodes from getting advertised prefixes from their optimal egress point while the presence of white nodes along a path ensures that prefixes from optimal egress point will be received. Accordingly, elected node is connected to the root of target groups through white, non-blocking nodes since it employs the shortest path to get to respective group heads. Following this, the listeners in all multicast groups will always be able to learn routes announced by the ASBR of both origin and target groups excluding the duplicate routes received from the internal group’s best egress point. Backup routes can be provided by relay routers to its group members if the optimal point prefix is no longer available. Having the routers act as relay nodes of the different multicast groups brings about diversity such that if prefix learnt over one session to a multicast root fails the next best route over another multicast root can be advertised into the origin multicast group.

In the following chapters, we will give our result comparing number of sessions and BGP messages in these schemes. Then we will consider other methods of selecting the relay nodes. Selection of these nodes can be implemented depending on unique network needs and characteristics.

Chapter 4

Evaluation of Relay-Based Multicast iBGP

In this chapter, we provide overview of tools and technologies used in running the simulations. We provide the parameters and configurations used for the experiment. The improvement induced by our proposed model with respect to metrics including number of iBGP sessions, RIB Size and memory consumption of BGP peers are shown in the results of the simulations. We analyze the derived graphs and discuss the implications.

4.1 Overview of Tools Used

4.1.1 Pyretic

Pyretic is an SDN programming language that runs on POX controller. Its main target is to simplify programming of OpenFlow switches by network administrators by providing a high-level abstraction of specifying network policies. Using pyretic allows for the switches policies to be specified collectively, as compared to the legacy way of specifying policies

individually for each switch in the network. Another advantage of pyretic is its program modularity which ensure isolation of policies per switch, hence eliminating the worries of switch policy interference. In pyretic, policy combination is performed using one of the two concepts, namely, parallel composition or sequential composition [23]. Monitoring network conditions is an important need of the network administrators, so it is integrated into the pyretic policy function. Given the change in network conditions, pyretic supports both static and dynamic policy assignment [23]. Pyretic policies specification is viewed as functions. These functions when invoked accepts packets and returns modified packets at specified output locations. A policy to drop packets is a function that returns nothing, return of a packet implies unicast packet forwarding, while return of more than one packet implies multicast packet dissemination.

In our experiment, we use the parallel composition operator + and sequential composition operator >> to specify the policies. As per pyretic standard syntax in [23], we use the syntax in Table 4.1 to write our policies. The pyretic runtime supports operation with different OpenFlow controllers, connecting the OpenFlow client over a socket.

Table 4.1: Pyretic syntax used in writing policies in our POX controller

Pyretic Syntax	
Syntax	Summary
identity	returns original packet
none	returns empty set
match(f=v)	identify if field f matches v, none otherwise
modify(f=v)	returns packet with field f set to v
fwd(a)	modify(port=a)
flood()	returns one packet for each local port on the network spanning tree

4.1.2 POX

POX controller is an open source, python based SDN controller developed by Nicira. Its target compatible operating systems include, linux, windows and mac os. Since it is built as NOX in python, it supports the same graphic user interface and virtualization tools as NOX. It has support for OpenFlow version 1.0.

We implement our network using pyretic SDN programming language with POX. Computation of multicast tree and assignment of dynamic multicast addresses are handled by our POX controller. The POX controller writes policies to our network nodes ensuring that prefixes are disseminated following the computed multicast tree. In the case of network changes resulting from node leaving or joining the network, the POX controller notifies other network nodes and appropriate actions follow, whether it is deleting of multicast group, withdrawing of advertised routes or re-computation of multicast trees.

4.1.3 OpenFlow

OpenFlow protocol is a standard southbound protocol employed by SDN controllers to exchange messages with switches. Switches which support the OpenFlow protocol are called "OpenFlow switches" while others which support both legacy routing and OpenFlow are referred to as "hybrid switches". It provides a means to program the flow tables in switches. As such, network administrators are able to classify and compel their network traffic to follow certain routes. A flow table, secure channel and Openflow protocol are three necessary components of the OpenFlow switches [32]. Each flow entry in the flow table has a corresponding action carrying information regarding how packets matching this flow entry should be treated. The secure channel is needed for a secure bidirectional exchange of packets between the OpenFlow switches and the SDN controller. OpenFlow

switches employ Ternary Content Addressable Memory (TCAM) for wildcard lookups. This concept suffices the fast-forwarding characteristic of the OpenFlow switches.

In our simulation, we deploy OpenFlow-supported network devices in mininet virtual environment. The deployed network nodes are able to communicate with the controller with OpenFlow messages.

4.1.4 Mininet and MiniNExT

Mininet employs the concept of virtualization to deploy network hosts. To enable the switches in the virtualized environment support OpenFlow, it employs the Open vSwitch. Virtual Ethernet is used to create links between virtualized switches, this virtual Ethernet is provided by the linux kernel [33]. TCP connections can then be set up between the created nodes depending on the desired topology of the user. The network of nodes created in mininet virtual environment can be connected to the SDN controller for management and coordination of the network. Mininet can be used to simulate the behavior of SDN controllers. It uses python for configuration, can operate in emulation mode, and is relatively scalable.

Mininet, however, does not give any assurance of its virtualized switches sending packets at the same rate. The unpredictability of nodes sending packets is an outstanding limitation of Mininet given that this factor depends on operating system properties including CPU speed, bandwidth and the size of network nodes [33]. Results gotten from experiment performed on mininet are largely unrepeatable. MiniNExT stands for Mininet Extended. It is an extended form of mininet which can create PID namespace for each running process. Its support for filesystem and runtime makes it appropriate for use with the Quagga daemon process which requires filesystem for running its routing suites.

In our simulation, we use MiniNExT to integrate Quagga into the virtual environment of mininet. We use two scripts for MiniNExT: the first python script creates our topology while the second script starts and customizes the routers running in our virtual environment.

4.1.5 Quagga

Quagga is a software routing suite which implements various routing protocols [12]. It carries the GPL license. Other related routing suite includes the XORP and BIRD [12]. Quagga implementation includes its core daemon called "zebra" which provides an interaction interface. Each running Quagga instance is a separate process which brings about modularity. Also, the occurrence of failure in a Quagga instance does not affect the other, hence increasing reliability. The supported routing protocols are handled by individual protocol daemon, our BGP protocol is handled by "bgpd", the Border Gateway daemon of Quagga.

As shown in the Quagga architecture, the routing modules communicate with the central Quagga daemon, making use of the library functions in Quagga [12]. In Quagga, all the routing daemons listen to the Virtual Terminal Interface, called VTY, at their respective ports. The daemons "zebra daemon", "ripd", "ripng", "ospfd", "bgpd", and "ospf6d" have associated ports 2601, 2602, 2603, 2104, 2605 and 2606, respectively. Following this, we use port 2605 to communicate with our Quagga BGP instance.

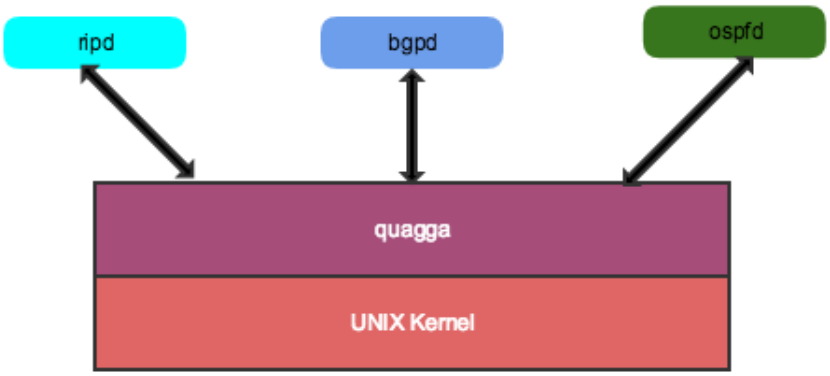

Figure 4.1: Quagga Architecture

4.1.6 EXABGP

This is a software tool used in injecting BGP routes into the network. It can also be used for gathering network information. ExaBGP speaks BGP with our network routers through scripts. Using this tool, we control the route announced by BGP peers and the overall BGP announcement via ExaBGP.

4.2 Performance Evaluation

4.2.1 Assumptions

We assume that every BGP speaker is able to re-advertise BGP prefixes. Also, the decision to advertise a prefix or not solely lies with the BGP speaker. For the RR network, we assume that a client can peer with more than one route reflector, particularly two route reflectors in our case. The topology implemented by our algorithm is depicted in Figure 4.2 below.

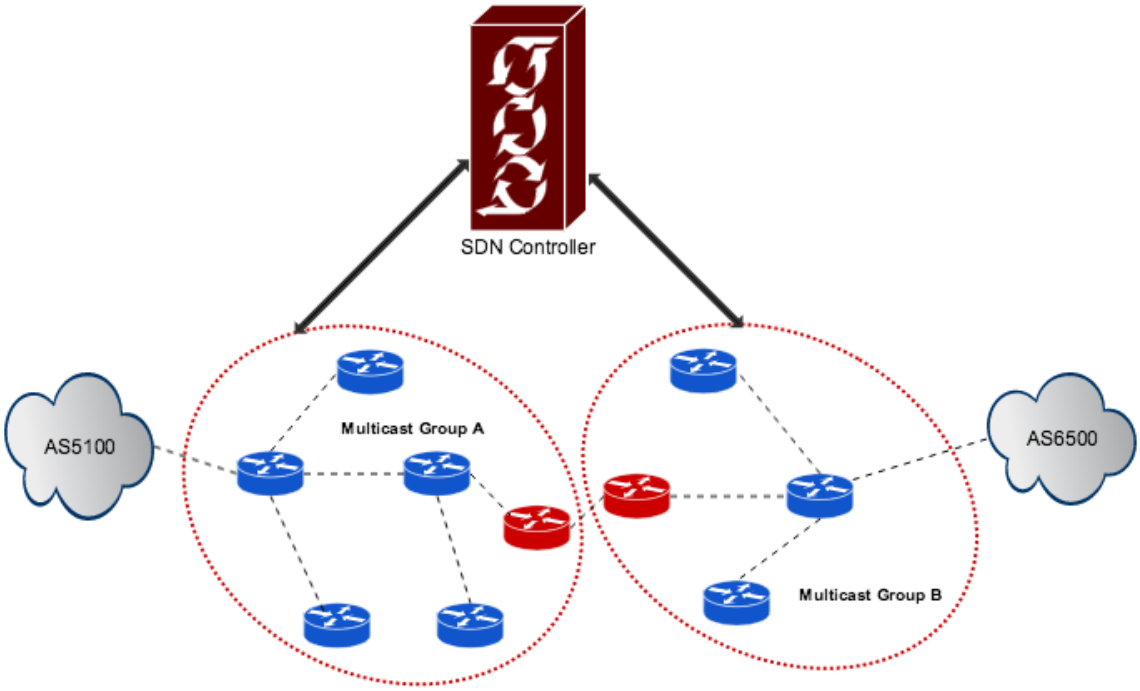

Figure 4.2: Implementation Network Topology

4.2.2 Performance Metrics

Our simulation measures the following performance metrics for each scheme:

- *Total BGP sessions* depicts the number of iBGP sessions established by BGP peers in the network.
- *Total BGP messages* represents the total amount of BGP messages received and sent by the BGP peer.
- *RIB size* is the size of routes entered into the RIB of the BGP peer.
- *Memory Consumption* is the total memory usage of the BGP peer.

- *Convergence time* represents the time taken by a BGP scheme to react and re-adjust to network changes.

4.2.3 Simulation Environment

We present the evaluation of our proposed algorithm in this section. First, we study the number of BGP sessions generated by the full-mesh and route reflector scheme for BGP message dissemination. We compare it to the sessions generated by our algorithm in the same network topologies. In this work, to study the growth of BGP sessions, a small and medium network are considered. For the small network, we consider the range of 5 to 20 nodes, while in the medium network has 30 to 80 nodes. To build our full-mesh topology, we established peer sessions between all running BGP Quagga instances in the network.

Each Quagga instance has a file for bgd daemon configuration. For the route reflection scheme, we establish iBGP between route reflectors and add clients to each. In our network, a client can establish sessions with more than one route reflector for reliability. With regards to the simulation of our algorithm, each ASBR is a multicast group head, and relay node is selected per group from active multicast members. Mesh BGP sessions are established between all selected relay nodes. Additional BGP sessions are established between group peers following the multicast tree towards the ASBR which is considered the optimal egress point for this group. In Table 4.2, we depict the network details used in simulating the number of BGP sessions for the relatively small network.

Table 4.2: Number of route reflectors, border routers and relay nodes deployed per network size in a relatively small network.

Network Size Parameters in Deployed Schemes			
AS nodes (Network size)	No. of Route reflectors in RR topology	No. of Border routers in proposed topology	No. of Relay routers in proposed topology
5	2	2	2
8	3	4	4
11	4	5	5
14	6	6	6
17	8	9	9
20	8	9	9

In Table 4.3, we depict the parameters used in getting the number of sessions for the medium sized network of 30 to 80 nodes.

Table 4.3: Number of route reflectors, border routers and relay nodes deployed per network size in medium-sized network.

Network Size Parameters in Deployed Schemes			
AS nodes (Network size)	No. of Route reflectors in RR topology	No. of Border routers in proposed topology	No. of Relay routers in proposed topology
30	10	10	20
40	15	18	36
50	18	20	40
60	20	22	44
70	23	25	50
80	27	28	56

4.2.4 Analysis

We observe in Figure 4.3 that both the route reflector and the proposed relay-based multicast schemes have lesser number of sessions established in comparison to the full-mesh topology. This is due to the fact that in full mesh topology all BGP peers created

a session with all other nodes in the network. The advantage is that every node is one hop from each other. The cost of this idea is the high increase in the number of BGP sessions established. However, our topology shows an improvement in the number of sessions established between all network peers compared to both full-mesh and route reflector scheme. Even though, the cost of this is that nodes are no longer one hop from each other, the benefit which includes reduction in memory consumption supersedes the cost. Similarly, in the medium sized network, there is also improvement in the number of iBGP sessions established in the network in comparison to FM and RR scheme. It was also observed that, as the network topology size grows, the need for more route reflectors also increases. Moreover, our topology maintains the same elected relay nodes per group provided no new groups are created.

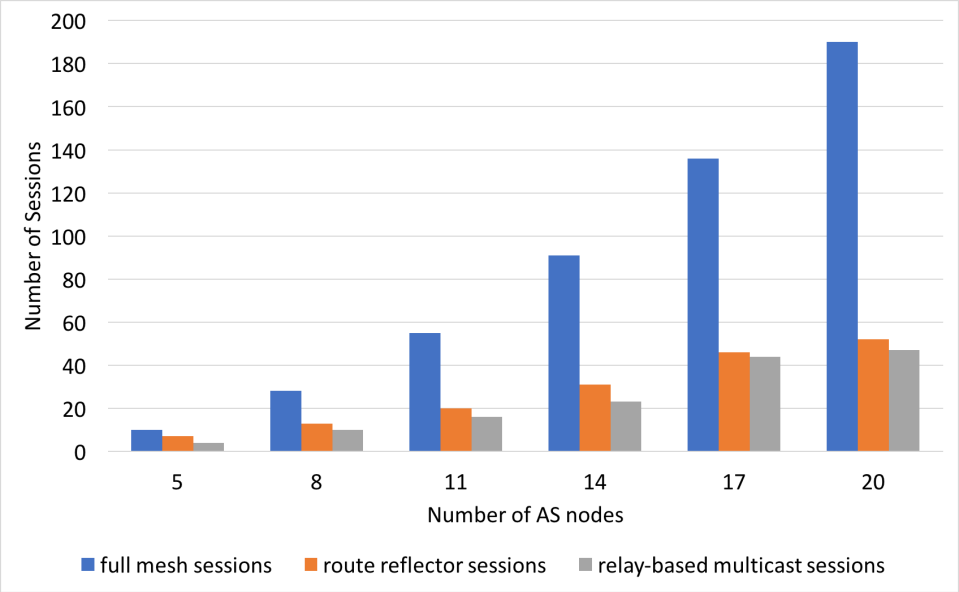

Figure 4.3: Small Topology BGP Sessions

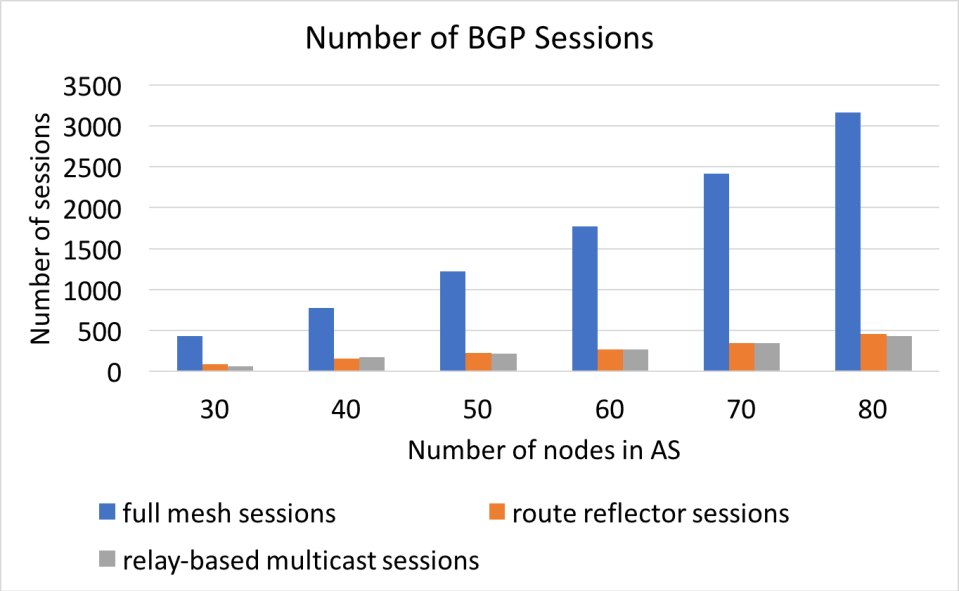

Figure 4.4: Number of BGP sessions

In the memory and routing information analysis, we considered a base network of 10 BGP peers as shown in Figure 4.2. The base network has 2 ASBRs, one peering externally with AS6500 and the other, AS5100. Since we had two ASBRs, it follows that we had 2 multicast groups and hence 2 relay nodes in total. Each multicast group has a relay router who acts as a filter to its origin group. We carry out our experiment in Ubuntu Virtual Box environment. In our test network, we coordinate the forwarding policies of router using the pyretic SDN controller. Each BGP peer is a running Quagga and "bgpd" daemon instance. When a router comes alive in the network, if it is an ASBR then the controller assigns to it a multicast address, else the peer joins one multicast group of which it is closest to the ASBR given the IGP cost. Every multicast group has one relay router. Routers are elected as relay router if the previous relay router fails or when a new group is formed as explained in our algorithm in the previous chapter. Once a peer fails or leaves the network, all advertised routes are withdrawn and if the peer was an ASBR, the multicast group is removed. We employ ExaBGP software suite to

inject the BGP prefixes into the network. The simulation was run for 30 times and data retrieved from a BGP peer of the multicast group, we term this peer "peer-cara3". The number of BGP prefixes advertised range from 22 to 65 BGP routes.

We show the result of BGP received messages for the full-mesh scheme, RR scheme and our algorithm in Figure 4.5 and the RIB size in Figure 4.6 of "peer-cara3". We observe that there is a decrease in the RIB size and total BGP message received by "peer-cara3" during the run with our algorithm compared to when we ran the full-mesh and RR AS. We achieve about 60% improvement with respect to the total number of messages sent within the AS compared to the FM scheme. The decrease in total BGP messages is due to appointment of relay nodes and the reduced number of sessions at the BGP peer, which in turn implies lesser update messages, notifications and other messages.

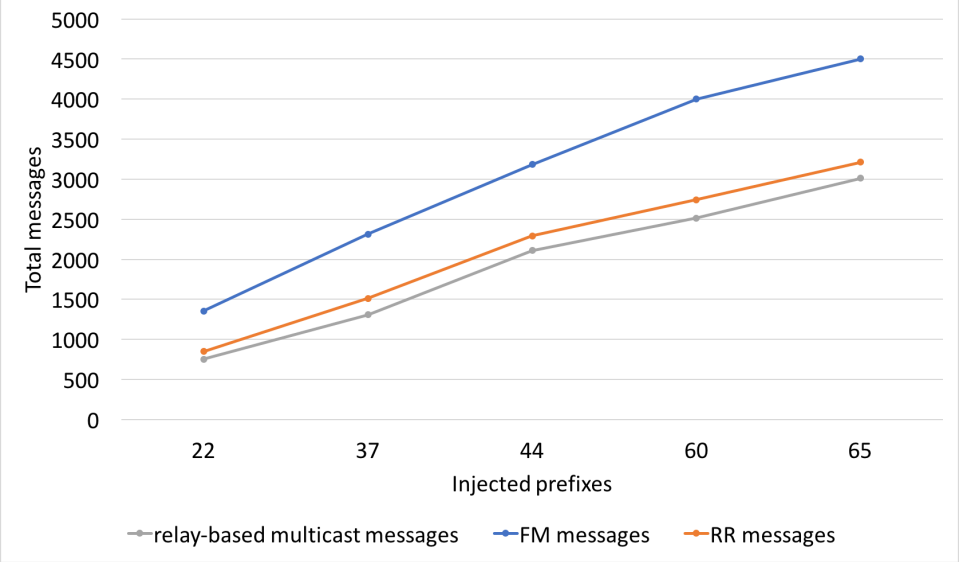

Figure 4.5: BGP received messages

Moreover, with respect to the RIB size of "peer-cara3", we achieved a 10% decrease compared to FM scheme coupled with a relatively smaller improvement with respect to the RR scheme. This decrease is significant when viewed cumulatively from the

perspective of the entire network. The decrease of RIB size in "peer-cara3" can be traced to the filtering performed by its elected relay node.

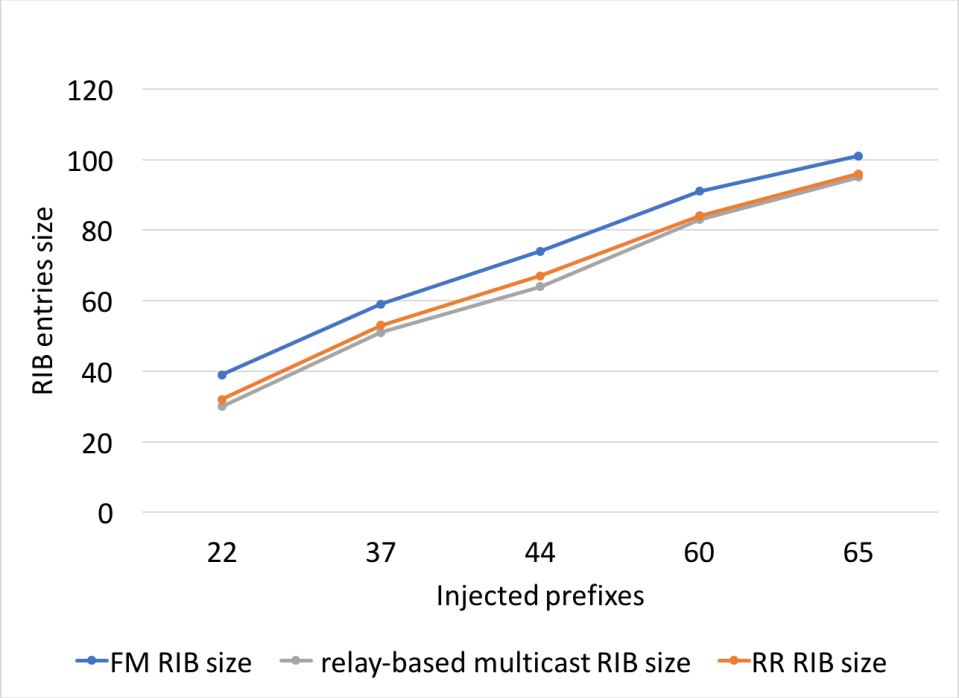

Figure 4.6: Routing Infomation Base Size

We also observed that there is a reduction in memory consumption in our relay-based multicast scheme compared to the full-mesh topology since peering sessions assumes memory. Most importantly, this is due to the fact that our scheme restricts unnecessary duplicate prefix dissemination to network nodes and rather employs the elected relay-routers to store the duplicate route collectively in case of optimal route failure. While in full-mesh scheme, a router is responsible for individually storing all unique and duplicate routes received which amplifies the memory needs as the network grows. The memory consumption in RR scheme is relatively similar to our scheme. This is shown in Figure 4.7.

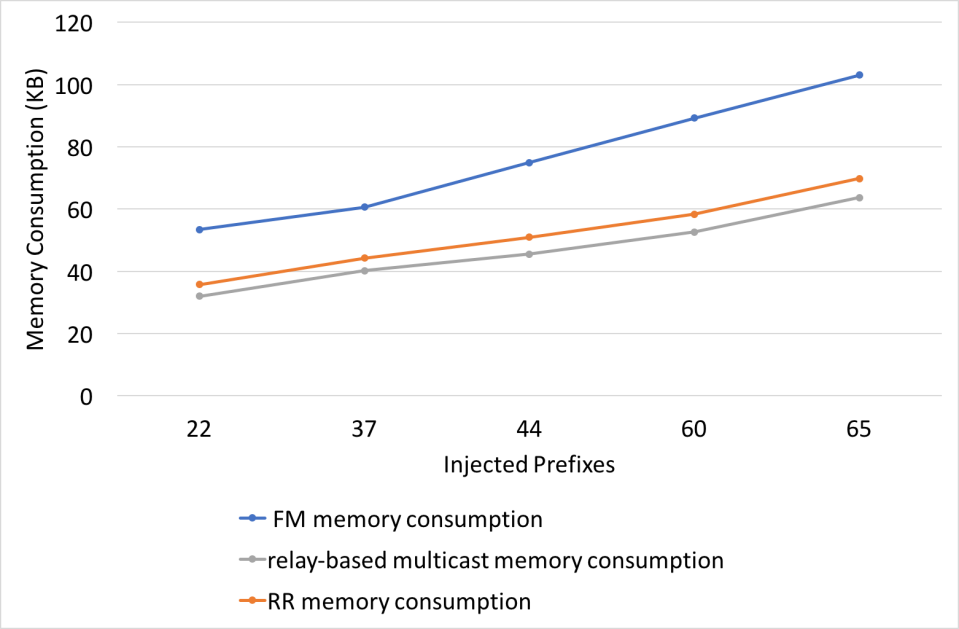

Figure 4.7: BGP Peer Memory Consumption

Finally, we note that one of the cost associated with our approach is the convergence time and computation. Our scheme requires that a multicast tree is built and that advertised routes travel through the multicast tree to group members; this has an impact in the convergence time as we depict in Figure 4.8. Network changes have to travel down the multicast path, resulting in about 30% convergence time increase compared to FM scheme and 17% increase compared to the RR scheme.

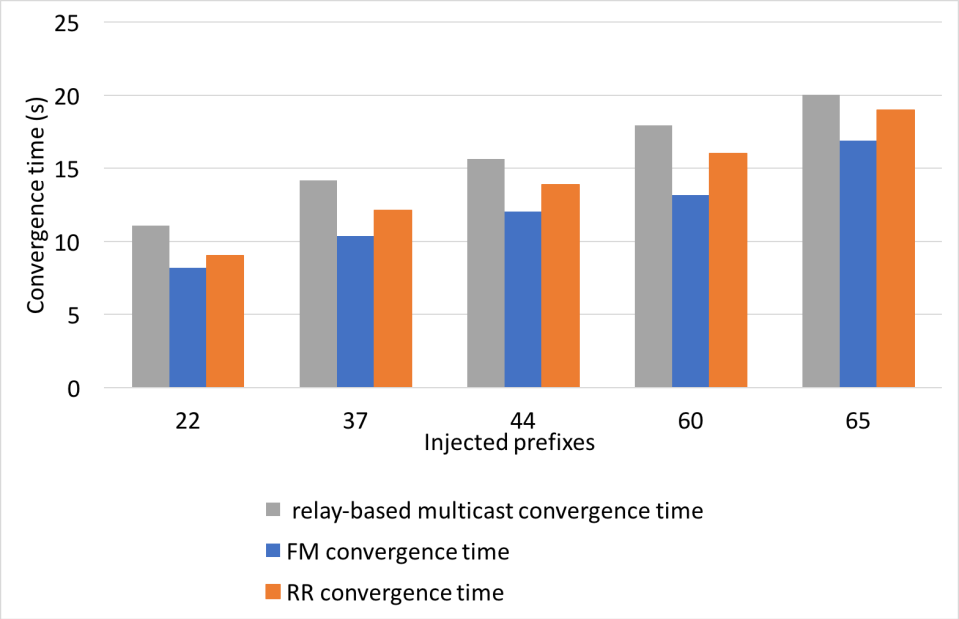

Figure 4.8: BGP convergence time

4.3 Conclusion

In this section, we showed that using our message dissemination scheme in BGP route advertisement leads to benefits including, decrease in the number of iBGP sessions, memory consumption and RIB size of BGP peers. The employment of SDN controller enables routing policies to be made based on the full network topology knowledge. The reduction in the circulation of duplicates by collectively storing them at elected relay node in this algorithm leads to reduction in CPU load of internal peers.

The higher cost of convergence time can be traced to the fact that network changes have to travel through the multicast tree, which has a relatively longer path than the FM and RR scheme. Notwithstanding, the improvements realized by our algorithm supersedes the small cost of required additional time. In real networks, the tradeoff between convergence time and minimization of unnecessary duplicate which in turn reduces the

CPU work load seems to be a reasonable decision.

Through our simulations outcome, we show that our algorithm can be employed in realistic networks since the results bring reasonable improvement with regards to FM scheme and is mostly similar to RR scheme. However, we note that since duplicate routes are stored at the relay routers, the cases of overloading at the relay nodes may occur. Some techniques to handle this case are discussed in the next chapter.

Chapter 5

On Multi-Node Relay-Based Multicast iBGP

The relay-based multicast iBGP algorithm presented in the previous chapter achieves reduction in iBGP sessions and memory consumption. However, pitfalls including single-point failure and overloading at the relay routers become critical. In this chapter, we present an improved model, where multiple relay routers are elected per multicast group to increase reliability. The cost and benefit of this method are analyzed.

5.1 Problem Statement

For this chapter, we say that there is increase in route distribution reliability and diversity if at least two relay nodes are available to disseminate BGP prefixes from target multicast groups into an origin multicast group. In the Figure 5.1, we have three multicast groups, Group A, Group B, and Group C. We consider C, as the origin group and `{B, A}` to be the set containing target multicast groups.

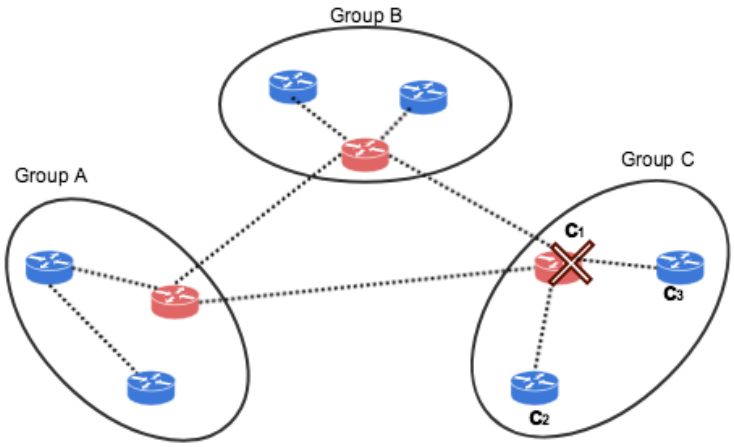

Figure 5.1: Event of relay node failure

In the first instance in Figure 5.1, the node C_1 has been elected as the relay node for multicast group C. Node C_1, filters out duplicate prefixes already advertised by its ASBR, multicast group head. In the event of node C_1 failure or attack, the network suffers critically before a new relay node is elected and recovery takes place. Also, in the event of rapid increase in BGP prefixes advertised by the target groups, without any increase in the bandwidth and memory of relay node C_1, overloading and ultimately crashing may occur as shown in Figure 5.2. These events are undesirable and should be avoided.

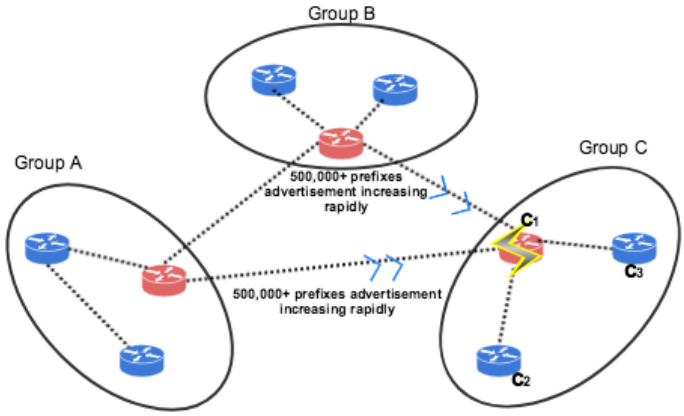

Figure 5.2: Node crash due to rapid increase in advertise prefixes

Consider the second scenario, in Figure 5.3, where Group C is again the origin group and nodes C_1 and C_2 have been elected as relay node for the multicast group. In this example, if there is a case of rapid increase in BGP prefixes advertised by the ASBRs of the target groups, the burden will be shared between the relay nodes rather than being handled by individual effort. Accordingly, if there is an attack or failure at a relay node point, the second relay can undertake the task of re-advertisement until a second relay node is elected. This concept acts as a form of "hot plugging" reliability model [28].

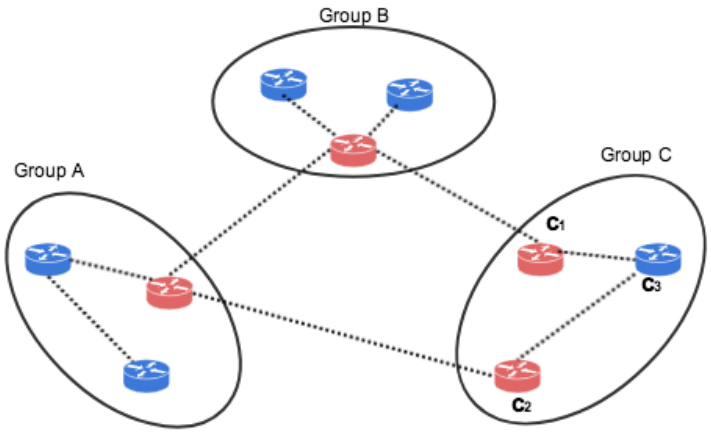

Figure 5.3: Load sharing between relay nodes

5.2 Proposed Algorithm

Our algorithm proposes the use of a set of relay routers per multicast group. We give the conditions for nodes eligibility for relay node election and selection. This algorithm introduces a minimal number of iBGP sessions to be added to the AS. We achieve route diversity and evade single-point failure, decrease the risk of vulnerability to single-point attack, and increase reliability. The algorithm takes as input the IGP link metrics and outputs the set of relay nodes per multicast group. The set of selected relay node is set to 2 for simplicity, so each multicast group must elect 2 relay nodes. The working principle of the algorithm is divided into two parts as in the previous algorithm, the initialization phase and the relay node selection phase. The initialization phase remains same, where every node joining the network calculates its cost to all available ASBR and then joins the multicast group of which the cost is minimum. In the relay node selection phase, the first round involves the selection of a node with minimum cumulative cost to target groups added to the relay node set. The second step of selection compares the

cumulative cost of other group nodes to that of the already selected node. A node is selected to join the relay node set, if and only if, it minimizes some of the highest cost of the already selected relay node.

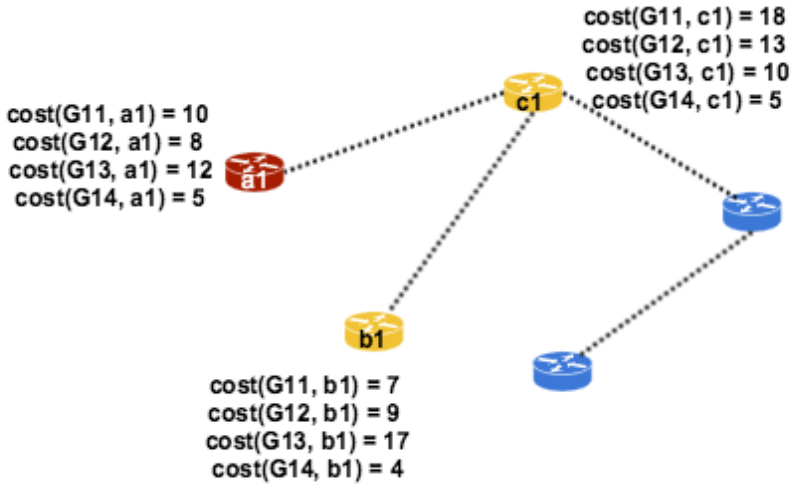

Figure 5.4: Applying the relay node algorithm, node set = 2

Take the case of Figure 5.4 for example, where the cumulative cost of a_1 is 35, b_1 is 37 and c_1 is 46. In the first round of relay node election, node a_1 was selected because it had the least cumulative cost, 35. In the second round, both nodes b_1 and c_1 are eligible for relay node positions. However, node b_1 is selected since it has the next minimum cost to some of the target groups. After the selection, target group re-assignment occurs, and a_1 will be assigned target group G12 and G13 while b_1 is assigned G11 and G14 since it minimizes the cost for these groups.

5.2.1 Algorithm for Initialization Phase

For the following algorithm, we use the symbols given in Table 5.1 below.

The initialization stage depicted here involves assigning multicast addresses to all

Table 5.1: Symbols and Meaning Used in Algorithm

Table of Notations	
Symbol	Meaning
$A_{routers}$	Set of all ASBRs
r_{relay}	Selected relay node
$r_{potential}$	Nodes eligible to become relay node
$sp(A_j, N_i)$	Shortest path between ASBR, A_j and node, N_i
M_{A_j}	Multicast group of ASBR A_j
N_{nodes}	All network nodes excluding ASBRs
`cum_sp`	Cumulative shortest path

ASBRs in the network by the SDN controller, and then joining of multicast groups by nodes based on their closest ASBR.

Algorithm 2 Initialization Algorithm - SDN Controller

```
1: Procedure_initialization():
2: Init M_{A_j} for all A_routers;
3: for N_i in N_nodes do:
4:     for A_j in A_routers do:
5:         Calculate sp(A_j, N_i);
6:         // Selection of minimum shortest path
7:         if sp(A_j, N_i) < sp(A_{j-1}, N_i) then
8:             add N_i to group M_{A_j}
9:         end if
10:    end for
11: end for
```

5.2.2 Algorithm for Relay Node Selection Phase, set =2

Here, we give the algorithm for the second phase which involves the relay node selection. Each multicast group undergoes an election given its target groups in the first round, and the node with the cumulative minimum cost to the target group is selected. The second election adds another node to the relay node set which now has one node already. The selection of the second relay node depends on the minimization effect of this node

compared to the first selected relay node. All nodes in the multicast group are eligible to undergo this election, except the ASBR and the already selected node.

Algorithm 3 Algorithm for Selection of 2 Relay-Nodes per Multicast Group - SDN Controller

```
// A'_j refers to all target multicast group
Compare_Cost_Minimization(r_potential, r_relay):
for A_j* in A'_j do:
    if sp(r_potential, A_j*) < sp(r_relay, A_j*) then
        calculate minimization cost
    end if
end for
Procedure_RelayNode_Selection():
Init relay node, R_{A_j} for all A_routers
Set cum_sp(A_j, A'_j) = ∞
for node in M_{A_j} do
    //cumulative cost between each node and target group
    cum_sp = sp(node, A'_j)
    if cum_sp < cum_sp(A_j, A'_j) then
        cum_sp(A_j, A'_j) = cum_sp
        select node as relay node
    end if
end for
// r_relay is relay node selected at first round
R_{A_j} = [r_relay] //relay node set
//minimization cost is mc
mc = 0
for r_n in r_potential do:
    ccm = Compare_Cost_Minimization(r_n, r_relay)
    if ccm>mc of all other r_potential then
        mc = ccm
        add r_n to R_{A_j}
    end if
end for
```

5.3 Leaving and Joining Multicast Group

We describe below the process of joining and leaving multicast groups by nodes. When a node appears in the network, the SDN controller informs the node about the existing multicast groups. The node undergoes the join process and becomes the member of the selected multicast group.

5.3.1 Algorithm for Joining Multicast Group

Joining of multicast groups by network nodes is coordinated by the SDN controller. The controller informs arriving nodes of the existing multicast groups.

Algorithm 4 Algorithm for Joining Multicast Group

```
1: Procedure_join_group(node, borderRouters)
2: Init minCostOfNode = ∞
3: Init selectedMulticastGroup = nil
4: for br in borderRouters do
5:     cost = sp(node,br)
6:     if cost < minCostOfNode then
7:         minCostOfNode = cost
8:         selectedMulticastGroup = br
9:     end if
10: end for
11: join group of selectedMulticastGroup
```

5.3.2 Algorithm for Leaving Multicast group

We now present the algorithm for leaving multicast group when a node voluntarily or involuntarily leaves the network. The node type is checked and an appropriate action is taken by the SDN controller accordingly.

Algorithm 5 Algorithm for Leaving Multicast Group

```
1: Procedure_leave_group(node, multicastGroup)
2: // nodeType refers to node category relayNode, borderNode, regularNode
3: if nodeType(node) == "relayNode" then
4:     withdrawAdvertisedRoutes(node)
5:     relayNodeSelection(multicastGroup)
6: else if nodeType(node) == "borderNode" then
7:     withdrawAdvertisedRoutes(node)
8:     deleteMulticastGroup(node)
9: else nodeType(node) == "regularNode"
10:    withdrawAdvertisedRoutes(node)
11: end if
```

5.4 Implementation and Evaluation

5.4.1 Assumptions

We assume that every BGP speaker is able to re-advertise BGP prefixes. Also, the decision to advertise a prefix or not solely lies with the BGP speaker. For the RR network, we assume that a client can peer with more than one route reflector, particularly two route reflectors in our case.

5.4.2 Simulation Environment

In this section, we describe the implementation of this algorithm. We employ the same tools as in Chapter 4. However, this time, for every multicast group, two relay node instances are created, and we establish BGP sessions between the relay nodes and their corresponding target group. For this work, we divide the workload, and therefore the target groups, in half among the selected relay nodes Quagga instance. Following this, our testbed has three groups as in Figure 5.5 with each relay node attending to a single target group. We show the RIB size and the memory consumption as compared to the

previous algorithm. The simulation is repeated 35 times, and the results have been averaged.

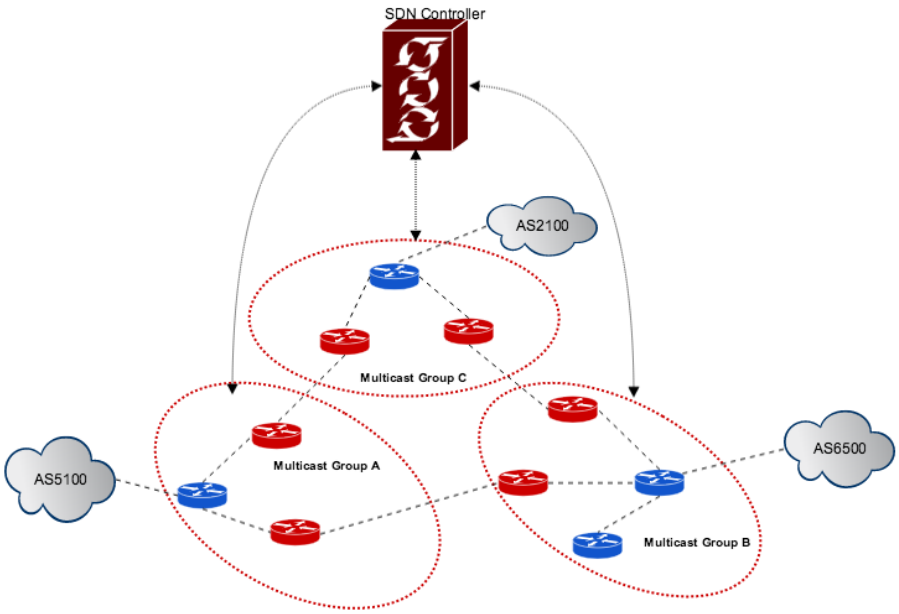

Figure 5.5: Implemented network topology

5.4.3 Analysis

To study the impact of RIB processing load, we simulated an instance where a single relay node receives all advertised routes from target groups and compared it to another instance where two relay nodes are selected and each only receives advertised routes from its assigned target group. We observe in Figure 5.6 that there is about 63% decrease in the RIB processing load on a particular relay node instance since the work load is shared among selected relay nodes. It should be noted that the decrease in RIB processing load depends on factors including the number of relay nodes used, the number of assigned target group per relay node, prefixes advertised by target groups, etc.

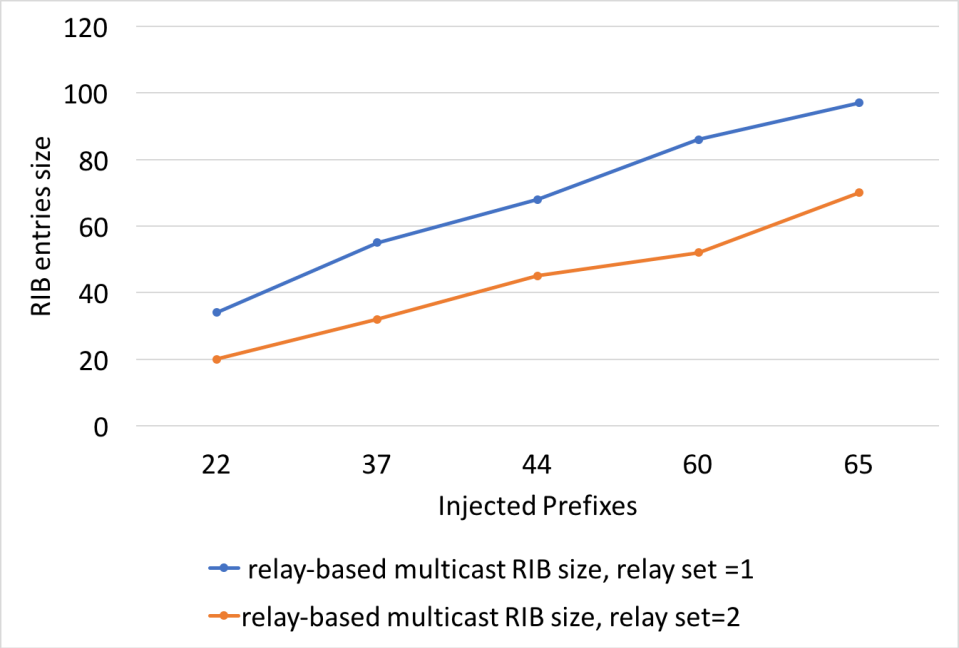

Figure 5.6: RIB size showing RIB load when all advertised routes are received by (first scenario) one relay node and (second scenario) two relay nodes.

More interestingly, we observed the reduction in memory consumption of the relay nodes in this topology as compared to the relay node memory using the previous concept where we had a single relay node. We see that the load is shared between the elected relay nodes and each relay node establishes BGP sessions with its assigned target group which result in a reasonable decrease in memory consumption of the relay node as depicted in Figure 5.7. Additional benefits include decreased occurrence of overloading and correspondingly increase case of reliability and fault tolerance. It was also observed that the number of sessions is not affected since we maintained a single BGP session between groups, regardless of which relay node in the group actually established the session.

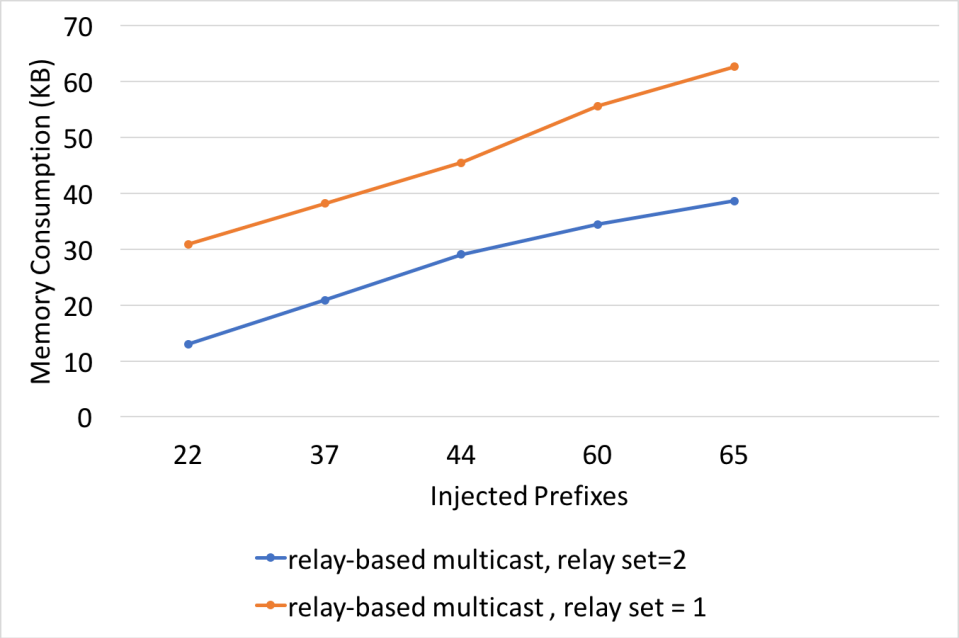

Figure 5.7: Memory consumption at relay nodes

5.5 Conclusion

In this chapter, we provided an improved method of selecting relay nodes to improve reliability, share load and minimize single point error. We have shown that there is reduction in workload at the relay node point if we increase the set of relay node from 1 to 2 nodes. Again, we note that even though this algorithm brings these benefits, that comes with a cost of more computations to be performed by the SDN controller. While we employed 2 relay nodes for simulation, we believe this solution can be generalized to accommodate more relay nodes. Finally, depending on the unique network needs and structure more relay nodes could be selected as per need.

Chapter 6

Conclusion and Future Work

In this chapter, we will summarize the findings and discus the future directions of the proposed work in the previous chapters yet to be explored. Particularly, we will discuss the reliablity issues involved since our algorithm employs unreliable UDP in the dissemination of prefixes among iBGP speakers. We also point to a new way of selection of relay nodes based on clusters to improve load sharing and fault tolerance.

6.1 Summary of Work Done

In this thesis, we started by pointing out the duplicate problem in multicast iBGP prefix advertisement in the Internet. First, we proposed a relay-based multicast BGP message dissemination involving a two phase process. The first phase describes the initialization process and the second phase describes the relay node selection process. In this scheme, we employ one relay node per multicast group. SDN controller is employed to coordinate the multicast tree and policies of the network.

Furthermore, we show by simulation that compared to FM and RR schemes, our algorithm is capable of achieving a decrease in the total number of iBGP sessions established

in the AS. We also achieve a 60% decrease in the total amount of messages compared to FM scheme, and a reasonable improvement in memory consumption and RIB size of the internal BGP peer. Generally, our performance is very close to that of RR scheme. We note that the cost of our algorithm is the convergence time. Since messages take longer path in our algorithm compared to the FM scheme, additional time is required for the AS to react to changes and return to a stable state.

Due to the possibility of overloading, single point attack and unreliablity involved in using a single relay node per group, we proposed another algorithm which employs multiple relay nodes per multicast group for prefix advertisement. Particularly, we employ two relays nodes per group, yet we note that more number of relay nodes may be employed following unique network needs. We show by simulation that using multiple relay nodes per multicast group can achieve a 63% decrease in RIB entries to be processed and a reasonable decrease in memory consumption at each relay node in comparison to the single relay node scheme. We point out that the achieved improvement in RIB size and memory consumption is highly dependent on factors including the number of relay nodes used, the number of assigned target groups per relay node, prefixes advertised by target groups, etc.

Through our scheme, we minimized prefix duplicates by eliminating unnecessary peering sessions in the network and filtering at elected relay nodes. This way, we achieved a significant reduction in unnecessary burden at BGP peers and its RIB size. The reduction in peer sessions also leads to scalability which allows for the deployment of this approach in large networks. The proposed scheme established only feasible iBGP sessions between peers since the BGP sessions follow the IGP network links. Following this model will limit the reconvergence time since our algorithm strictly follows the network IGP links which is known for its fast reconvergence property.

Employing SDN for decision making ensured that routing decisions are made based on the knowledge of full network topology. Hence, we minimized suboptimal route decision. We ensured that routers are aware of atleast one path to each external destination while we stored the backup routes at the relay nodes. Additionally, we established a kind of priority scheme since the proposed scheme enforces that network nodes first and foremost establish peering session with their minimum IGP shortest path egress point at initialization.

6.2 Future Work

6.2.1 Multicast Reliability & Cluster-Based Relay Selection

The proposed algorithm in this work employs multicasting for the message dissemination. The implication of this is that instead of the reliable TCP protocol, multicast employs UDP protocol which is unreliable.

- The unreliability of the message dissemination presents a point of concern in our scheme. In order to increase the reliability and stability in the dissemination of BGP prefixes, packet loss detection scheme can be performed. Also, to increase message distribution reliability, we can categorize multicast peers into an hierarchical structure, where peers are ranked according to peer stability.

- Finally in order to further explore the message dissemination and duplicate concept in iBGP multicast mechanism, it will be interesting to establish elected relay routers on cluster basis. Fault tolerance can be improved since the load of a failed relay node in the cluster can be immediately shifted to be handled by another cluster members.

References

[1] S. Agarwal, C. Chuah, and S. Bhattacharyya. Impact of BGP Dynamics on Router CPU Utilization. In *5th International Workshop of Passive and Active Network Measurement*, pages 278–288, 2004.

[2] T. Bates, R. Chandra, and E. Chen. BGP Route Reflection - An Alternative to Full Mesh IBGP. In *RFC 2796*, 2000.

[3] M. Buob, A. Lambert, and S. Uhlig. iBGP2: A Scalable iBGP Redistribution Mechanism Leading to Optimal Routing. In *IEEE INFOCOM*, pages 1–9, 2016.

[4] M. Buob, M. Meulle, and S. Uhlig. Checking for Optimal Egress Points in iBGP Routing. In *6th International Workshop on Design and Reliable Communication Networks*, pages 1–8, 2007.

[5] T. Dargahi, A. Caponi, M. Ambrosin, G. Bianchi, and M. Conti. A Survey on the Security of Stateful SDN Data Planes. *IEEE Communications Surveys and Tutorials*, pages 1–1, 2017.

[6] A. Elmokashfi, A. Kvalbein, and C. Dovrolis. BGP Churn Evolution: A Perspective from the Core. In *IEEE INFOCOM*, pages 1–9, 2010.

[7] F. Godn, S. Colman, and E. Grampn. Multicast BGP with SDN Control Plane. In *7th International Conference on the Network of the Future (NOF)s*, pages 1–5, 2016.

[8] A. Golechha, S. Karanje, and J. Abraham. Comparative Study of Multicasting Protocols Based on Average End-to-End Delay. In *International Conference on Computing, Analytics and Security Trends (CAST)*, 2016.

[9] N. Gray, T. Zinner, and P. Tran-Gia. Enhancing SDN Security by Device Fingerprinting. In *IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 2017.

[10] T. G. Griffin and G. Wilfong. On the Correctness of iBGP Configuration. In *ACM SIGCOMM*, 2002.

[11] P. Gransson and C. Black. Software Defined Networks: A Comprehensive Approach. In *Elsevier*, 2014.

[12] K. Holter. Wireless Extensions to OSPF: Implementation of the Overlapping Relays Proposal. In *Master Thesis, University of Oslo*, 2006.

[13] T. Huang, F. R. Yu, C. Zhang, J. Liu, J. Zhang, and Y. Liu. A Survey on Large-Scale Software Defined Networking (SDN) Testbeds: Approaches and Challenges. In *IEEE Communications Surveys and Tutorials, Vol. 19, No. 2*, 2017.

[14] G. Huston and G. Armitage. Projecting future IPv4 Router Requirements from Trends in Dynamic BGP Behaviour. In *Australian Telecommunication Networks and Applications Conference (ATNAC)*, 2006.

[15] M. Ji and R. Chen. Caching and Coded Multicasting in Slow Fading Environment. In *IEEE Wireless Communications and Networking Conference (WCNC)*, 2017.

[16] Inc. Juniper Networks. Configuring BGP routing Advertising Routes: BGP advertise-best-external-to-internal. In *http://www.juniper.net/techpubs/software/erx/junose71/swconfig-bgp-mpls/html/bgp-config10.html*, 2007.

[17] P. Lin, J. Bi, and H. Hu. BTSDN: BGP-based Transition for the Existing Networks to SDN. In *6th International Conference on Ubiquitous and Future Networks (ICUFN)*, 2014.

[18] K. Mahajan, D. Sharma, and V. Mann. ATHENA: Reliable Multicast for Group Communication in SDN-based Data Centers. In *IEEE 9th International Conference on Communication Systems and Networks (COMSNETS)*, 2017.

[19] J. Park, R. Oliveira, A. Shane, M. Danny, and Z. Lixia. BGP Route Reflection Revisited. *IEEE Communications Magazine, Vol. 50, No. 7*, pages 70–75, 2012.

[20] J. H. Park, D. Jen, M. Lad, S. Amante, D. McPherson, and L. Zhang. Investigating Occurrence of Duplicate Updates in BGP Announcement. In *Krishnamurthy A., Plattner B. (eds) Passive and Active Measurement*, volume 6032, 2009.

[21] C. Pelser, T. Takeda, E. Oki, and K. Shiomoto. Improving Route Diversity through the Design of iBGP Topologies. In *IEEE International Conference on Communications*, pages 5732–5738, 2008.

[22] T. Pfeiffenberger, J. L. Du, P. B. Arruda, and A. Anzaloni. Reliable and Flexible Communications for Power Systems: Fault-tolerant Multicast with SDN/OpenFlow. In *7th International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–6, 2015.

[23] J. Reich, C. Monsanto, N. Foster, J. Rexford, and David Walker. Modular SDN Programming with Pyretic Proposal. In *Usenix, The Advanced Computing Systems Association*, volume 38, 2013.

[24] Y. Rekhter, T. Li, and S. Hares. IETF RFC 4271 a Border Gateway Protocol (BGP-4). In *Reston: Internet Society*, 2006.

[25] B. Rong, I. Khalil, and Z. Tari. Making Application Layer Multicast Reliable is Feasible. In *31st IEEE Conference on Local Computer Networks*, pages 483–490, 2006.

[26] B. Sarakbi and S. Maag. BGP skeleton An Alternative to iBGP Route Reflection. *International Journal of Distributed Sensor Networks*, 12:1–5, 2010.

[27] V. Schrieck, P. Francois, and O. Bonaventure. BGP Add-Paths: The Scaling/Performance Tradeoffs. *IEEE Journal on Selected Areas in Communications, Vol. 28, No. 8*, pages 1299–1307, 2010.

[28] L. Shen, T. Fang, W. He, and X. Ruan. A Novel Strategy to Achieve Distributed Control and Redundancy for Input-series-output-parallel Inverter System. In *IEEE Energy Conversion Congress and Exposition (ECCE)*, 2015.

[29] W. J. A. Silva and D. F. H. Sadok. Control Inbound Traffic: Evolving the Control Plane Routing System with Software Defined Networking. In *IEEE 18th International Conference on High Performance Switching and Routing (HPSR)*, 2017.

[30] A. I. Swapna, Md. R. Huda, and M. K. Aion. Comparative Security Analysis of Software Defined Wireless Networking (SDWN)- BGP and NETCONF Protocols. In *19th International Conference on Computer and Information Technology*, 2016.

[31] P. Traina, D. McPherson, and J. Scudder. Autonomous System Confederations for BGP. In *RFC 5065*, 2007.

[32] S. Wang, H. Chiu, and C. Chou. Comparisons of SDN OpenFlow Controllers over EstiNet: Ryu vs. NOX. In *International Symposium on Advances in Software Defined Networks*, 2015.

[33] S. Wang, C. Chou, and C. Yang. EstiNet OpenFlow Network Simulator and Emulator. *IEEE Communications Magazine, Vol. 51, No. 9*, pages 110–117, 2013.

[34] K. Wrona, S. Szwaczyk, M. Amanowicz, and K. Gierlowski. SDN Testbed for Validation of Cross-layer Data-centric Security Policies. In *International Conference on Military Communications and Information Systems (ICMCIS)*, 2017.

[35] L. Xiao, J. Wang, and K. Nahrstedt. Optimizing IBGP Route Reflection Network. In *7th ACM Conference on Embedded Networked Sensor Systems*, pages 1765–1769, 2003.

[36] K. Xie, X. Huang, S. Hao, M. Ma, P. Zhang, and D. Hu. E3MC: Improving Energy Efficiency via Elastic Multi-Controller SDN in Data Center Networks. *IEEE Access, Vol. 4*, pages 6780–6791, 2016.

[37] T-W. Yang, Y-L. Hsieh, M-H. Chen, and C-F. Chou. Exploiting Path Diversity in Content Delivery Network with the Collaboration of SDN. In *2017 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW)*, 2017.

[38] X. Zhang, X. Lu, J. Su, B. Wang, and Z. Lu. A Scalable,Distributed BGP Routing Protocol Implementation. In *IEEE 12th International Conference on High Performance Switching and Routing*, pages 191–196, 2011.