Performance Evaluation of the Border Gateway Protocol

by

Negar Navai

B.Sc., Azad University of Tehran, Iran, 1994

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

Master of Applied Science

in

THE FACULTY OF GRADUATE STUDIES

(Department of Electrical and Computer Engineering)

We accept this thesis as conforming
to the required standard

The University of British Columbia

October 2000

Department of Electrical & Computer Engineering

The University of British Columbia
Vancouver, Canada

Date Oct. 12, 2000

Abstract

The Border Gateway Protocol (BGP) is the *de facto* inter-domain routing protocol used to exchange reachability information between autonomous systems in the global Internet. The BGP is a path vector routing protocol. Distance vector routing protocols can take a long time to converge after a topological change. It is believed that the adoption of the path vector solves this problem. One of the objectives of this thesis is to investigate this claim. The BGP specification lacks convergence behavioral and performance analysis. This thesis presents the analysis of the BGP convergence behavior and performance. The behavior of the protocol can be estimated in an experimental manner by means of simulations. The effect of network topology on the number of BGP routing updates and convergence latency is examined. The analysis in this thesis is based on data collected in a simulation environment. The best and the worst-case of BGP convergence models are simulated. This analysis shows that BGP has bouncing problem. In the case of a route failure event, the upper bound on volume of routing update messages is found to be factorial and convergence latency is linear with respect to the number of autonomous systems. In the case of a route announcement event, the upper bound on number of routing update messages is found to be exponential with respect to the number of autonomous systems. It is found that performing MinRouteAdvertisementInterval timer and loop detection on the receiver router significantly reduces the number of BGP routing updates.

Contents

List of Tables

List of Figures

Acknowledgements

I would like to express my sincere thanks to my supervisor, Dr. Mabo R. Ito, for his guidance, support and advice throughout this work. I am very grateful for his commitment. I would like to thank Mark McCutcheon for his assistance in solving technical problems. I would also like to thank all the members of the High Performance Computing Laboratory.

My greatest thanks go to my parents and my sister, for their love, support and encouragement. Completion of this thesis would have been impossible without their support.

I would also like to thank all of my friends for their very valuable comments and suggestions.

NEGAR NAVAI

The University of British Columbia
October 2000

To My Parents

Chapter 1

Introduction

This chapter introduces the thesis research topic. It explains why Border Gateway Protocol, BGP [20], is needed. The motivation of the research along with an outline of the accomplished work is then presented.

1.1 Internet Exterior Routing Protocol Development

By 1985, the ARPANET was heavily used and congested [7]. A large network was comprised of different hardware and software levels managed by different organizations and people. Management of an extremely large network challenged the networks rigidity and flexibility. This led to the hierarchy of the Internet into domains and creation of exterior routing protocols.

The collection of domains, their policies and peering relationships, and the address prefixes they advertise, defines the Internet inter-domain routing system. The first inter-domain protocol used to provide autonomous systems routing through the Internet was Exterior Gateway Protocol (EGP) [13]. EGP borrowed many of the characteristics of distance vector protocols [21]. The NSFNET used EGP [18] to exchange reachability information between the backbone and the regional networks. Although the use of EGP was widely deployed, its topology restrictions and inefficiency in dealing with routing loops and setting routing policies created a need for a new and more robust protocol [3]. Currently, BGP4 is the de facto standard for Internet routing.

BGP is built based on the learned experience with EGP in the NSFNET backbone. It is an advanced exterior routing protocol that is providing the Internet with a controlled and loop-free topology. BGP4 is the first version that provides a new set of mechanisms for supporting classless inter-domain routing and route aggregation [5].

1.2 Motivation and Objectives

Distance vector routing protocol, including RIP [12] can take a long time to converge after a topological change. This is due to routers having insufficient information to determine if the next hop will cause a routing loop. This problem is known as the count-to-infinity [21] problem. The result of count-to-infinity process is a bouncing effect. The bouncing effect is in fact a routing loop. A routing loop occurs when traffic circles back and forth between domains, never reaching its final destinations. Several solutions to the count-to-infinity problem have been proposed such as split horizon [1]. The adoption of the path vector [4] in BGP is believed to solve this problem. It is claimed that BGP converges faster than other traditional distance vector protocols including RIP [8]. One of the objectives of this thesis is to investigate this claim.

The BGP specification [20] has several ambiguities. It lacks in convergence behavioral analysis. The specification is quite ambiguous about BGP performance. The BGP specification calls for loop detection. However, there is no indication where loop detection should be performed.

The objectives of this thesis are to clarify the BGP specification explicitly and to develop models and a better behavioral analysis. Two characteristics of the routing system can impact the performance of BGP4 protocol:

- inter-domain topology: this is the graph of autonomous systems and peering relationships between them.

- route stability: the routing system experiences transient changes in routes, caused by router and link failures or router misconfiguration.

Parallel to the growth of the Internet, the routing system has also rapidly increased in size. Understanding the impact of size of routing system on BGP convergence behavior is important for BGP protocol evaluation.

1.3 Contributions

In order to analyze BGP convergence behavior, simulation software from the MRT [14] project has been used and a test bed has been constructed consisting 6 PCs each running the MRT's implementation of BGP.

To evaluate the protocol a set of metrics has been identified. Several routines have been added to the source of the simulation in order to measure those metrics.

A simplified model of BGP interconnectivity is used in this analysis. This model neglects the impact of routing policies and IBGP (Internal BGP) [9] interconnectivity on the process of convergence. The major results include:

- Although BGP adopts path vector algorithm, a router can learn about a new invalid path which results in bouncing contrary to the speculation expressed in [4].

- In the case of a route failure event, the global upper bound on volume of routing update messages is found to be factorial and convergence latency is linear with respect to the number of autonomous systems.

- In the case of a route announcement event, the global upper bound on number of routing update messages is found to be exponential with respect to the number of autonomous systems.

- It is found that performing MinRouteAdvertisementInterval timer and loop detection on the receiver router significantly reduces the number of BGP routing updates.

1.4 Outline of the Thesis

Chapter 2 provides an overview of routing protocols and present a study of the related work. Chapter 3 explains our methodology and simulation software. Both empirical observations as well as quantitative analysis of the relationship between specific Internet topological configurations and the rate of convergence are presented in chapter 4. Chapter 5 presents the conclusions inferred from the results and proposes future work for optimizing BGP, which if deployed, would significantly improve inter-domain routing convergence.

Chapter 2

Background

This chapter provides an overview of IP routing and describes how BGP works. It also provides a summary of works that have been done in the area of BGP analysis.

2.1 An Overview of IP Routing

Routers build routing tables that contain their collected information on all the best paths to all the destinations they know how to reach. They both announce and receive route information to and from other routers. This information goes directly into the routing tables. Routers develop a hop-by-hop mechanism by keeping track of next hop information that enables a data packet to find its destination through the network. A router that does not have a direct physical connection to the destination checks its routing table and forwards the packet to another next hop router that is closer to that destination. The process repeats until the traffic finds its way through the network to its final destination. Routing involves two basic activities:

1. Determining optimal routing paths

2. Transporting routing packets through an internetwork.

There are two primary types of routing: static and dynamic routing.

2.1.1 Static vs. Dynamic Routing

Static routing is the simplest form of routing and is preprogrammed. The tasks of discovering routes and propagating them throughout a network are left to the network's administrator. Static routing may be the preferred routing mechanism for very small networks. Network reachability in this case is not dependent on the existence and the state of the network itself. Whether a destination is up or down, the static routes would remain in the routing table, and the traffic would be sent toward that destination [7].

With a dynamic routing protocol, routers learn about the network topology by communicating with other routers. Each router announces its presence, and the routes it has available, to the other routers on the network. Therefore, if one adds a new router, the other routers will hear about the addition and adjust their routing tables accordingly. There is no need to reconfigure the routers to tell them that the network has changed. Similarly, if one moves a network segment, the other routers will hear about the change. In a dynamic routing protocol network reachability is dependent on the existence and state of network. If a destination is down, the route would disappear from the routing table, and traffic would not be sent toward the destination.

The main advantages of dynamic routing over static routing are scalability and adaptability. A dynamically routed network can grow more quickly and larger, and is able to adapt to changes in the network topology brought by this growth or by failure of one or more network components.

Each router participating in the dynamic routing protocol must decide exactly what information to send to other routers. More importantly, it must attempt to select the best route for reaching other destinations from the candidates it learns about from other routers. In addition, if a router is going to adapt to changes in the network, it must be prepared to remove old or unusable information from its routing table.

In order to communicate information about the topology of the network, routers must periodically send messages to each other using a dynamic routing protocol. These messages must be sent across network segments just like any other packets. But unlike other packets in the network, these packets do not contain any information to or from a user. Instead, they contain information that is only useful to the routers [2].

Dynamic routing protocols can be classified in several different ways: distance-vector versus link-state protocols, exterior gateway protocols (EGP) versus interior gateway protocols (IGP). The first classification has to do with the kind of information the protocol carries and the way each router makes its decision about how to fill in its routing table. The second classification is based on where a protocol is intended to be used.

2.1.2 Distance-Vector Protocols vs. Link-State Protocols

Distance vector routing algorithms operate by having each router maintain a table (i.e., a vector) giving the best known distance to each destination and the line to use to get there. These tables are updated by exchanging information with the neighbors [22]. In order to allow the information to propagate throughout the network, each router includes in its announcements all the destinations to which it is directly attached, as well as all destinations that it has heard about from other routers.

Distance vector protocols were mainly designed for small network topologies. These protocols do not support classless inter-domain routing (CIDR) and route aggregation.

Link state protocols work very differently from distance vector protocols. The way that routing information is communicated in a link state routing protocol is through link state advertisements. Link state advertisement contains identification information for the router generating it and information about the routers and networks to which it is connected, including the cost to get to those routers and networks. A router generates a link state advertisement for itself and sends it to all its neighbors. A router sends its link state advertisement when it initially comes up, whenever it experiences a topology change, or periodically to refresh the older link state advertisement. An algorithm runs in the network to ensure that every router's link state advertisement is delivered to every other router in the network. After a given router has received a complete set of link state advertisements for the network, that router can construct a map of the entire network and then perform computations on the map to decide the shortest path to every destination in the network [9].

Link state protocols support CIDR. Even though link state algorithms have provided better routing scalability, which enables them to be used in bigger and more complex topologies, they are still restricted to interior routing. Link state protocol cannot provide a global connectivity solution required for Internet inter-domain routing. In very large networks and in case of fluctuation caused by link instabilities, link state retransmission and recomputation will become too large for any router to handle [7].

2.1.3 Interior Gateway Protocols vs. Exterior Gateway Protocols

A routing algorithm within an autonomous system is called an interior gateway protocol, IGP ,also known as intra-domain routing. An algorithm for routing between autonomous systems is called an exterior gateway protocol, EGP, also known as inter-domain routing. Within a single autonomous system, the recommended routing protocol on the Internet is OSPF, between autonomous systems, BGP4 is used [22].

The classic definition of an autonomous system (AS) is a set of routers under a single technical administration, using an Interior Gateway Protocol to route packets within an autonomous system and using an Exterior Gateway Protocol to route packets to neighboring autonomous systems. Autonomous systems are assumed to be administered by a single administrative entity for the purposes of representation of routing information to the systems outside of the autonomous system [20].

2.1.4 Path Vector Protocol Overview

The routing algorithm employed by *path vector* bears a certain resemblance to the distance vector routing algorithm [4].

The *path vector* routing algorithm augments the advertisement of reachable destinations with information that describes various properties of the paths to these destinations. This information is expressed in terms of *path attributes.* To emphasize the tight coupling between reachable destinations and properties of the paths to these destinations, *path vector* defines a route as a pairing between a destination and the attributes of the path to that destination.

In *path vector* routing, a vector contains *paths* to set of destinations. The path, is expressed in terms of the domains traversed so far, is carried in a special path attribute which records the sequence of routing domains through which the reachability information has passed. Suppression of routing loops is implemented via this special path attribute.

Path vector does not require all routing domains to have homogenous criteria (policies) for route selection; therefore route selection policies used by one routing domain are not necessarily known to other routing domains.

To maintain maximum degree of autonomy and independence between routing domains, each domain that participates in *path vector* may have its own view of what constitutes an optimal path. This view is based solely on local route selection policies and the information carried in the path attributes of a route.

2.1.5 Classless Inter-Domain Routing

As the Internet has evolved and grown over in recent years, it has become painfully evident that it is soon to face several serious scaling problems. These include:

- Exhaustion of the class-B network address space. One fundamental cause of this problem is the lack of a network class of a size, which appropriate for mid-sized organization. Class-C, with a maximum of 254 host addresses, is too small while class-B, which allows up to 65534 addresses, is too large to be densely populated.

- Growth of routing tables in the Internet routers are beyond the ability of current software (and people) to effectively manage.

Classless inter-domain routing (CIDR) [5] attempts to deal with these problems by proposing a mechanism to slow the growth of the routing table and need for allocating new IP network numbers.

CIDR supports two important features that benefit the global Internet routing system:

1. CIDR eliminates the traditional concept of Class A, Class B, and Class C network addresses.

2. CIDR supports route aggregation where a single routing table entry can represent the address space of perhaps thousands of traditional classful routes. This allows a single routing table entry to specify how to route traffic to many individual network addresses.

2.2 Border Gateway Protocol Version 4

BGP is a path vector protocol used to carry routing information between autonomous systems. BGP went through different phases and improvements from its earlier version, BGP1, in 1989 to today's version, BGP4, deployment of which started in 1993. BGP4 is the first version that provides a new set of mechanisms for supporting CIDR and supernetting [20].

At a global level, BGP is used to distribute routing information among multiple autonomous systems. Figure 2.1 illustrates the information flows.

This diagrams points out that, while BGP alone carries information between autonomous systems, both BGP and IGP may carry information across an autonomous systems [19].

The network reachability information exchanged via BGP provides sufficient information to detect routing loops and enforce routing decisions based on performance preference and policy constraints. In particular, BGP exchanges routing information containing full autonomous system paths and enforces routing policies based on configuration information.

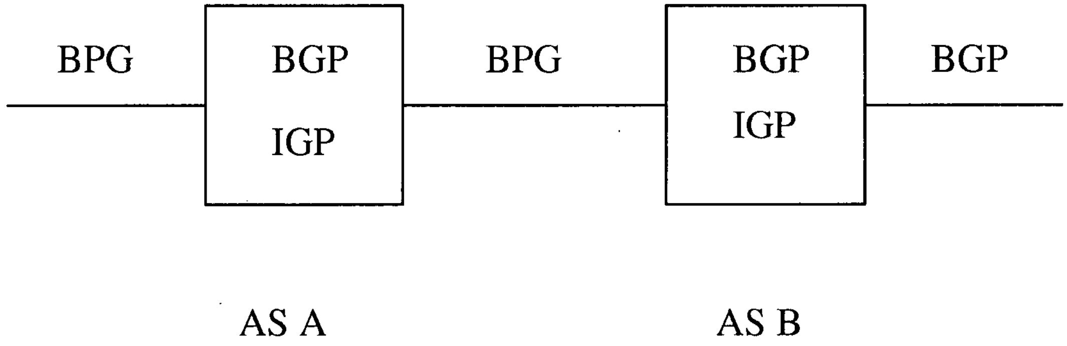


Figure 2.1: The relationship between autonomous systems

BGP runs over a reliable transport protocol. This eliminates the need to implement explicit update fragmentation, retransmission, acknowledgement, and sequencing. BGP uses TCP as its transport protocol [20].

BGP assumes that routing within an autonomous system is done by an intra-domain routing protocol. BGP does not make any assumptions about intra-domain routing protocols employed by the various autonomous systems. Specifically, BG-P does not require all autonomous systems to run the same intra-domain routing protocol [23].

Routers that communicate directly with each other via BGP are known as BGP speakers. BGP speakers can be located within the same autonomous system (I-BGP) or in different autonomous systems (E-BGP). BGP speakers in each autonomous system communicate with each other to exchange network reachability information based on a set of policies established within each autonomous system. For a given BGP speaker, some other BGP speaker with which the given speaker communicates is referred to as an external peer if the other speaker is in a different autonomous system, while if the other speaker is in the same autonomous system it is referred to as an internal peer [20].

There can be as many BGP speakers as deemed necessary within an autonomous system. Usually, if an autonomous system has multiple connections to other autonomous systems, multiple BGP speakers are needed. All BGP speakers representing the same autonomous system must give a consistent image of the autonomous system to the outside. This requires that the BGP speakers have consistent routing information among them. These gateways can communicate with each other via BGP or by other means. The policy constraints applied to all BGP speakers within an autonomous system must be consistent.

In the case of external peers, the peers must belong to different autonomous systems, but share a common Data Link subnetwork. This common subnetwork should be used to carry the BGP messages between them.

BGP supports four types of messages: *open*, *keepalive*, *update*, and *notification* messages [20].

- An *open* message is the first message sent after the TCP connection is established. The purpose of the *open* message is for each endpoint to identify itself to the other and to agree on protocol parameters, such as timers.

- BGP does not use any transport protocol-based keep-alive mechanism to determine if peers are reachable. BGP neighbors send a *keepalive* message to each other to confirm that the connection between them is still active.

- The *update* message is the primary message used to communicate information between two BGP speakers. When a BGP speaker advertises a prefix to a BGP neighbor or withdraws a previously advertised prefix, that BGP speaker uses an *update* message.

- If an error occurs during the life of a BGP session, the *notification* message can be used to signal the presence of such an error before the underlying TCP connection is closed. This arrangement allows the administrator of the remote system to receive an indication of why the BGP session was terminated. The BGP connection is closed immediately after sending it.

BGP peers initially exchange their full routing tables. To conserve bandwidth and processing power, BGP peers send incremental updates as routing tables change. BGP does not require periodic refresh of the entire BGP routing table. Therefore, a BGP speaker must retain the current version of the entire BGP routing tables of all of its peers for the duration of the connection. Keepalive messages are sent periodically to ensure the liveness of the connection. Notification messages are sent in response to errors or special conditions. If a connection encounters an error condition, a notification message is sent and the connection is closed.

Figure 2.2 is shown from the perspective of AS2, so only the routers in AS1 and AS3 that connect to AS2 are shown. This figure shows that AS2 has three routers, each of which has an I-BGP connection to all other routers. AS1 and AS2 are connected via an E-BGP connection between R1 and R2. AS2 and AS3 are connected via E-BGP session between R3 and R5. On the E-BGP session between AS1 and AS2, R1 advertises routes for prefixes within AS1, and R2 advertises routes for prefixes within both AS2 and AS3. R2 will have learned routes for prefixes within AS3 via the I-BGP session with R3. R3 will have learned these routes directly from R5 via the E-BGP session. Finally, R3 advertises to R5 routes for prefixes within both AS2 and AS1.

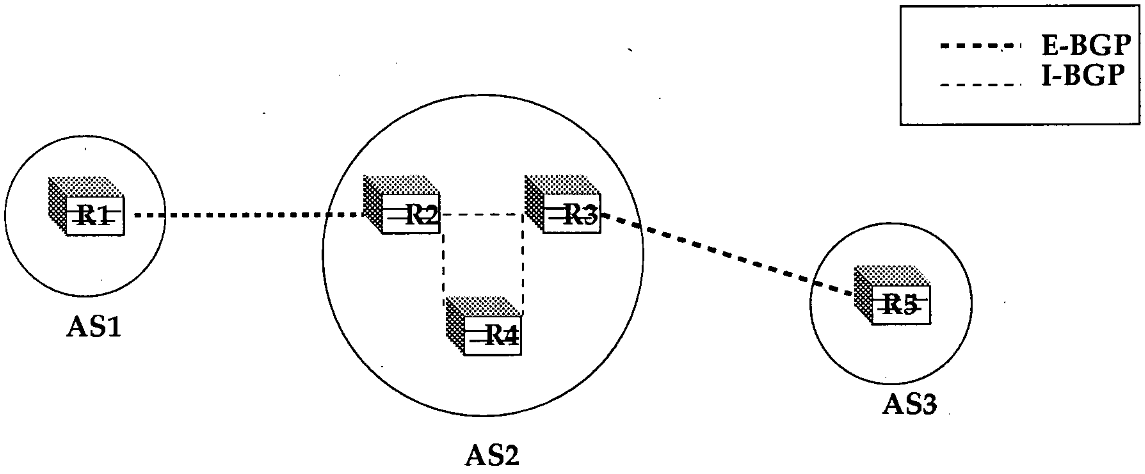


Figure 2.2: Complete BGP Example

BGP provides the capability for enforcing policies based on various routing preferences and constraints. Policies are not directly encoded in the protocol. Rather, policies are provided to BGP in the form of configuration information.

BGP enforces policies by affecting the selection of paths from multiple alternatives and by controlling the redistribution of routing information. Polices are determined by the autonomous system administration. Routing policies are related to political, security, or economic consideration [19].

2.3 How BGP Works

The update message is the primary message used to communicate information between two BGP speakers. Routing updates contain all the necessary information that BGP uses to construct a loop-free picture of the Internet. The following are the basic blocks of an update message [20]:

- Network Layer Reachability Information (NLRI)
- Path attributes
- Unreachable routes

The NLRI is an indication, in the form of an IP prefix route, of the networks being advertised. The path attributes list provides BGP with the capabilities of detecting routing loops and flexibility to enforce local and global routing policies. The third part of the update message, is a list of routes that have become unreachable - or in BGP terminology, *withdrawn.*

The IP prefix is an IP network address with an indication of the number of bits (left to right) that constitute the network number. The Network Layer Reachability Information (NLRI) is the mechanism by which BGP supports classless routing. The NRLI is the part of the BGP routing update that lists the set of destinations about which BGP is trying to inform its BGP neighbors.

Withdrawn routes provide a list of routing updates that are not feasible, or that are no longer in service and need to be withdrawn (removed) from the BGP routing tables. The withdrawn routes have the same format as the NLRI.

The BGP attributes are a set of parameters used to keep track of route-specific information such as path information, degree of preference of a route, next hop value of a route, and aggregation information. These parameters are used in the BGP filtering and route decision process. One of these parameters is AS-PATH attribute.

An AS-PATH attribute is a well-known mandatory attribute. It is a sequence of autonomous system numbers a route has traversed to reach a destination. The autonomous system that originates the route adds its own autonomous system number when sending the route to its external BGP peers. Thereafter, each autonomous system that receives the route and passes it on to the other BGP peers will prepend its own autonomous system number to the list. Prepending is the act of adding the autonomous system number to the beginning of the list. The final list represents all the autonomous system numbers that a route has traversed with the autonomous system number of the autonomous system that originated the route all the way at the end of the list. Figure 2.3 shows this list of autonomous systems. This type of AS-PATH list is called an AS-sequence, because all the autonomous system numbers are ordered sequentially.

BGP uses the AS-PATH attribute as part of the routing updates (update packet) to ensure a loop-free topology on the Internet. Each route that gets passed between BGP peers will carry a list of all autonomous system numbers that the route has already been through. If the route is advertised to the autonomous system that originates it, that autonomous system will see itself as part of the AS-PATH attribute list and will not accept the route. BGP speakers prepend their autonomous system numbers when advertising routing updates to other autonomous systems (external peers). When the route is passed to a BGP speaker within the same autonomous system, the AS-PATH information is left intact. AS-PATH information is one of the attributes BGP looks at to determine the best route to take to get to a destination. In comparing two or more different routes, given that all other attributes are identical, a shorter path is always preferred. In case of a tie, other attributes are used to make the decision.

BGP routers receive the update message, run some policies or filter over the updates, and then pass on the routes to other BGP peers. Cisco's implementation of BGP keeps track of all BGP updates in a BGP routing table separate from IP routing table. In case multiple routes to the same destination exist, BGP does not flood its peers with all those routes. Rather, it picks the best route and sends it.

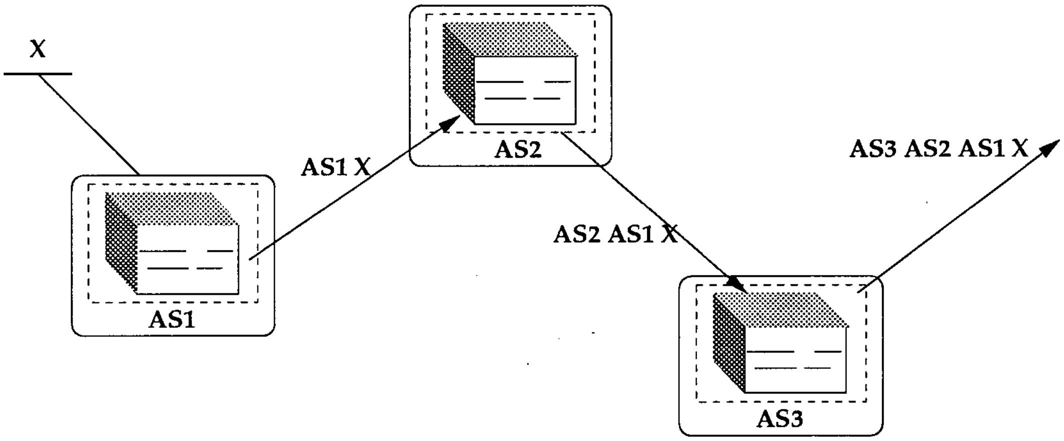


Figure 2.3: BGP routing

In addition to passing along routes from peers, a BGP router may originate routing updates to advertise networks that belong to its own autonomous system. Valid local routes originated in the system, and the best routes learned from BGP peers are then installed in the IP routing table. The IP routing table is used for the final routing decision [19]. Figure 2.4 [7] illustrates BGP routing process.

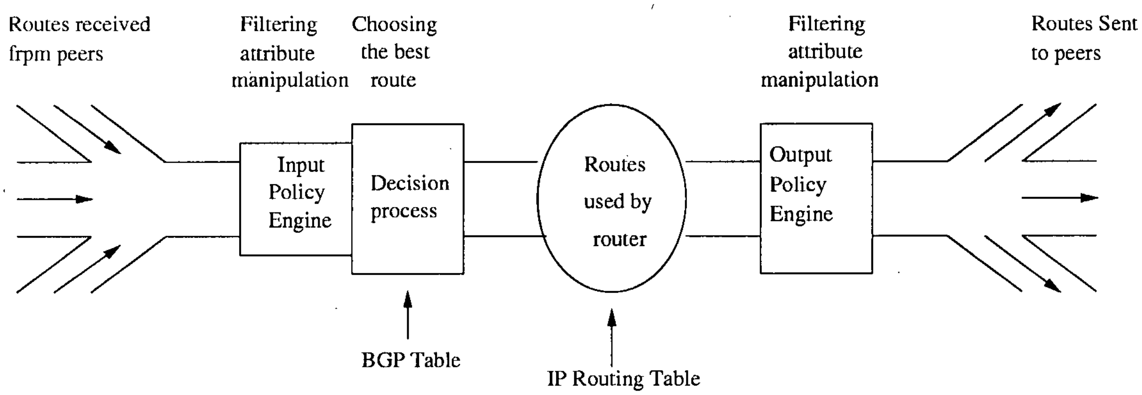


Figure 2.4: Routing process overview

In the following more details about each component are provided.

- Routes Received from Peers

  BGP receives routes from external or internal peers. Depending on what is configured in the Input Policy Engines, some or all of these routes will make it into the router's BGP table.

- Input Policy Engine

  This engine handles route filtering and attribute manipulation. Filtering is done based on different parameters such as IP prefixes, AS-PATH information, and attribute information. BGP also uses the Input Policy Engine to manipulate the path attributes to influence its own decision process and hence affect what routes it will actually use to reach a certain destination.

- The Decision Process

  BGP goes through a decision process to decide which routes it wants to use to reach a certain destination. The decision process is based on the routes that made it into the router after the Input Policy Engine was applied. The decision process is performed on the routes in the BGP routing table. The decision process looks at all the available routes for the same destination, compares the different attributes associated with each route, and chooses one best route.

  One of the major tasks of a BGP speaker is to evaluate different paths from itself to a set of destination covered by an address prefix, select the best one, apply appropriate policy constraints, and then advertise it to all of its BGP neighbors. The key issue is how different paths are evaluated and compared. In traditional distance vector protocols (e.g., RIP) there is only one metric (e.g., hop count) associated with a path. As such, comparison of different paths is reduced to simply comparing two numbers. A complication in inter-domain routing arises from the lack of a universally agreed-upon metric among autonomous systems that can be used to evaluate external paths. Rather, each autonomous system may have its own set of criteria for path evaluation.

  A BGP speaker builds a routing database consisting of the set of all feasible paths and the list of destinations (expressed as address prefixes) reachable through each path. In most cases, it is expected to have only one feasible path. However, when this is not the case, all feasible paths should be maintained in case of the loss of the primary path. Only the primary path at any given time will ever be advertised.

The path selection process can be formalized by defining a complete order over the set of all feasible paths to a set of destinations associated with a given address prefix. One way to define this complete order is to define a function that maps each full autonomous system path to a non-negative integer that denotes the path's degree of preference. Path selection is then reduced to applying this function to all feasible paths and choosing the one with the highest degree of preference. In actual BGP implementation, the criteria for assigning degree of preference to path are specified as configuration information.

Although not specified in the BGP standard [20], most vendor implementations ultimately default to the best path selection based on AS-PATH length. The number of autonomous systems in the path is used in a manner similar to the metric hop count in the RIP protocol. The analysis in this paper is based on the default behavior of BGP, or constrained shortest path first policies.

- Routes Used by the Router

  The best routes, as identified by the decision process, are what the router itself uses and are candidates to be advertised to other peers and also to be placed in the IP routing table.

  In addition to routes passed on from other peers, the router originates updates about the networks inside its autonomous system. This is how an autonomous system injects its routes into the outside world.

- Output Policy Engine

  This is the same engine as the Input Policy Engine, applied on the output side. Routes used by the router (the best routes) in addition to routes that the router generates locally are given to this engine for processing. The engine might apply filters and might change some of the attributes (such as AS-PATH or metric) before sending the update. The Output Policy Engine also differentiates between internal and external peers.

- Routes Advertised to Peers

  This is the set of routes that made it through the Output Engine and are advertised to the BGP peers, internal or external.

2.4 Related Work

Govindan *et al.* [6] has shown that the Internet topology is growing increasingly less hierarchical with the rapid addition of new exchange points and peering relationships.

In [11], Labovitz *et al.* describe significant level of Internet routing instability by measuring the BGP updates generated by service provider backbone routers at the major U.S. public exchange points. The authors show that most Internet routing instability in 1997 was pathological and stemmed from software bugs and artifacts of router vendor implementation decisions.

Labovitz *et al.* in [10] examined the latency in Internet path failure, failover and repair due to the convergence properties of inter-domain routing. Their results showed that inter-domain routers in the packet switched Internet might take tens of minutes to reach a consistent view of the network topology after a fault. They claimed that these delays stem from temporary routing table oscillations formed during the operation of BGP path selection process on Internet backbone routers. They also claimed that the theoretical upper bound on the number of computational states explored during BGP convergence is $O(n!)$, where n is the number of autonomous systems in the Internet.

Their analysis indicated that there is no correlation between convergence latency and geographic or network distance. Their analysis found no temporal correlations between convergence delay and the time of day or week. They demonstrated that even moderate levels of routing table oscillation would lead to increased packet loss, latency and out of order packets. They found that for the worst case in the event of a route withdrawal and a route announcement, $(n-1)O((n-1)!)$ announcements are generated.

2.5 Summary

In this chapter an overview of IP routing is presented. BGP messages and operations are explained. The results of experiments by Labovitz *et al.* are introduced. Their results came out very recently. The work done for this project is independent of their work. In the next chapter the methodology and an overview of simulation software used in this study is presented.

Chapter 3

Methodology

Performance testing is concerned with the derivation of performance characteristics of a protocol implementation in normal or overload situations. Current performance testing is best covered by the notion of performance measurements.

This chapter defines the metrics that are related to the performance of BGP4 protocol. Also this chapter gives an overview of methodology and BGP models used for testing the performance of the BGP4 protocol.

3.1 Metrics

This section describes the range of parameters exercised. We begin by defining a set of metrics to analyze the protocol with respect to scaling and performance. To evaluate the capability of the BGP4, following metrics are identified:

- Volume of routing updates
- Convergence latency

Volume of routing updates is the overall number of update messages generated by all BGP routers in the system during convergence process. This includes both announcement and withdrawal messages. An update message is used to advertise a single feasible route to a peer or to withdraw multiple unfeasible routes from service. It may simultaneously advertise a feasible route and withdraw multiple unfeasible routes [20]. A route announcement indicates a router has either learned of a new network attachment, or has made a policy decision to prefer another route to a network destination. Route withdrawals are sent when a router makes a new local decision that a network is no longer reachable.

Convergence is the process of agreement, by all routers, on optimal routes [17]. When a network event causes routes either to go down or become available, routers distribute routing update messages that permeate networks, stimulating recalculation of optimal routes and eventually causing all routers to agree on these routes. *Convergence latency* is the time when a routing event happens and routing tables of all BGP routers reach a steady state for that event. Steady state is when all BGP routers send no more updates for those routes. Convergence is not just a time factor but also a CPU and memory issue on each router.

3.2 Measurement Infrastructure

Performance testing and evaluation of the protocol is carried out by means of statistical data gathering in a simulation environment. The suggested methodology in this work is an active one [15]. Software from the Multi-threaded Routing Toolkit, MRT, project [14] has been used. The experiments are performed on a test bed consisting six computers, five running Linux 6.0 and one running FreeBSD 3.3. These machines are configured as routers running MRT's implementation of BGP. One of these machines injects BGP faults that simulate two routing events:

- A previously unavailable route is announced available.

- A previously available route is withdrawn.

Each routing processor logs the time it receives an update message and the time it sends an update message. In this thesis convergence latency of each injected routing event is defined as the time between the injection of the fault and the time when all BGP routers make the final decision for the injected prefix. To measure convergence latency, times reported by each router are compared with the other routers reports.

The time of fault injection should be known too. It is important that clocks on routers be synchronized [16]. When computing convergence latency, we are interested only in the differences between clock values, not the values themselves. During the process of sending updates, each router logs the time the packet is sent, BGP peer ID, and AS-PATH attribute. Upon receipt of an update, each router logs BGP peer ID, AS-PATH attribute, NLRI information and time-stamps the update message.

At the end of the data collection period, all logs are transformed to a central database machine for analysis. Data collection is performed during period of the time where there are no known planned network outages. The measurement architecture is shown in Figure 3.1.

Evaluation of performance metric is done by analyzing data in log files. Network of different sizes are simulated by programmatically introducing delay in message propagation and processing. In order to measure convergence latency, Poisson sampling [16] is used to generate samples of BGP data. We must stress at this point that the router hardware (memory and CPU) play an important role in determining the convergence latencies observed in this study.

3.3 MRT, Simulation Software

MRT is written by the University of Michigan and Merit Network, Inc [14]. MRT includes:

- MRTd, which is a routing daemon supporting BGP4.
- BGPSim, which is a BGP4 traffic generator, simulator. It simulates complex BGP4 routing environments with possibly high levels of routing instability/change.

MRT libraries fall into two main categories:

- Lower level services and support routines (timer, interface, socket routines, etc.)
- Protocol modules (BGP, kernel routing table support, etc.)

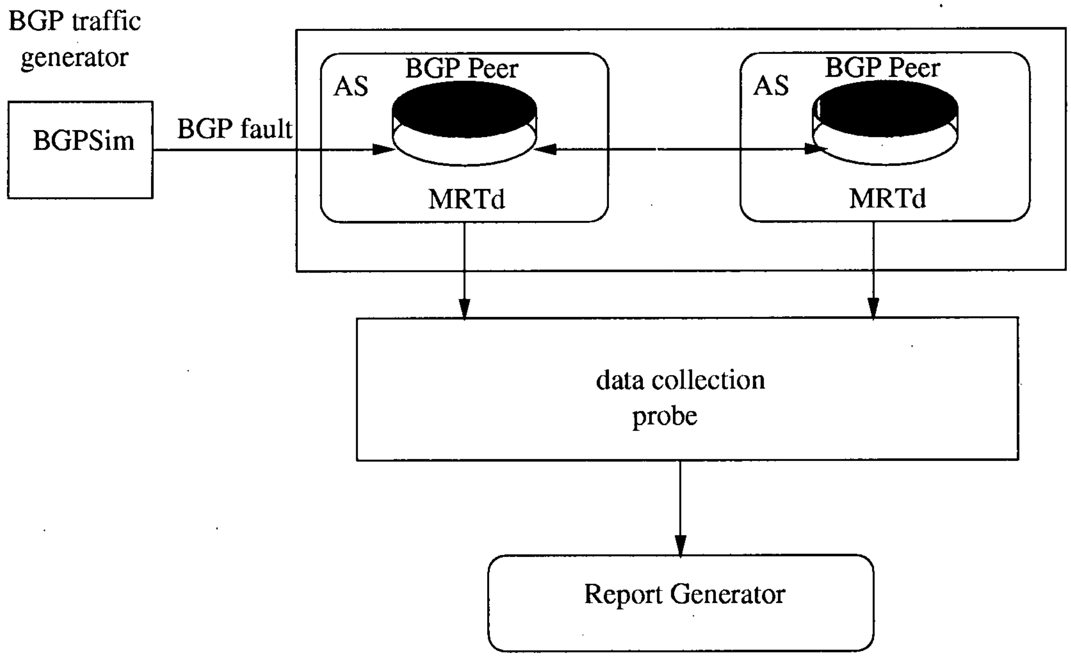


Figure 3.1: Diagram of fault injection and measurement infrastructure

MRT includes the following characteristics:

- Reads Cisco System-like router configuration file to configure routing protocols, routes peering and routing policy
- Scans the kernel for existing routes
- Scans the kernel interface list
- Initiates routing protocol communications

MRT provides an object-oriented, multi-threaded programming environment. It allows the user to define a configuration file. To be able to analyze the BGP convergence behavior several routines has been added to the source of the MRT:

- Routines for logging routing table changes
- Routines for tracing BGP routing updates

- It is indicated in the BGP specification [20] that AS-PATH attribute provides sufficient information to avoid routing information looping, but it does not specify where the detection should occur. A routine for performing loop-detection on the sender side has been written.
- routines for implementing MinRouteAdvertisementInterval timer

This approach requires a deep familiarity with the code and data structures.

3.4 An Abstract Model of BGP

This section presents an abstract model of BGP that is simulated to investigate properties related to protocol convergence. Graphs are commonly used to model the structure of the Internet [24]. In this study the Internet is modeled as an undirected graph, where a single node represents an autonomous system and edges represent peering relationships. All issues relating to IBGP are ignored. The impact of ingress and egress filters on BGP route propagation are excluded.

If loop-detection is performed on the sender side, then each router will check the AS-PATH attribute before advertising a prefix to its peer. If AS-PATH attribute includes the peer autonomous system number, then the router will send a withdrawal message to the peer for that prefix instead of sending an announcement. If loop-detection is performed on the receiver side, then each router upon receipt of an update message checks the AS-PATH attribute and invalidates any route which includes the router's own autonomous system number.

Figure 3.2 shows how peering sessions can be built among three BGP routers. To build EBGP session between two BGP routers, two different autonomous system numbers are assigned to each BGP router.

RA's configuration file:

```
router bgp 345
neighbor 137.82.52.86 remote-as 678
neighbor 137.82.52.127 remote-as 123
```

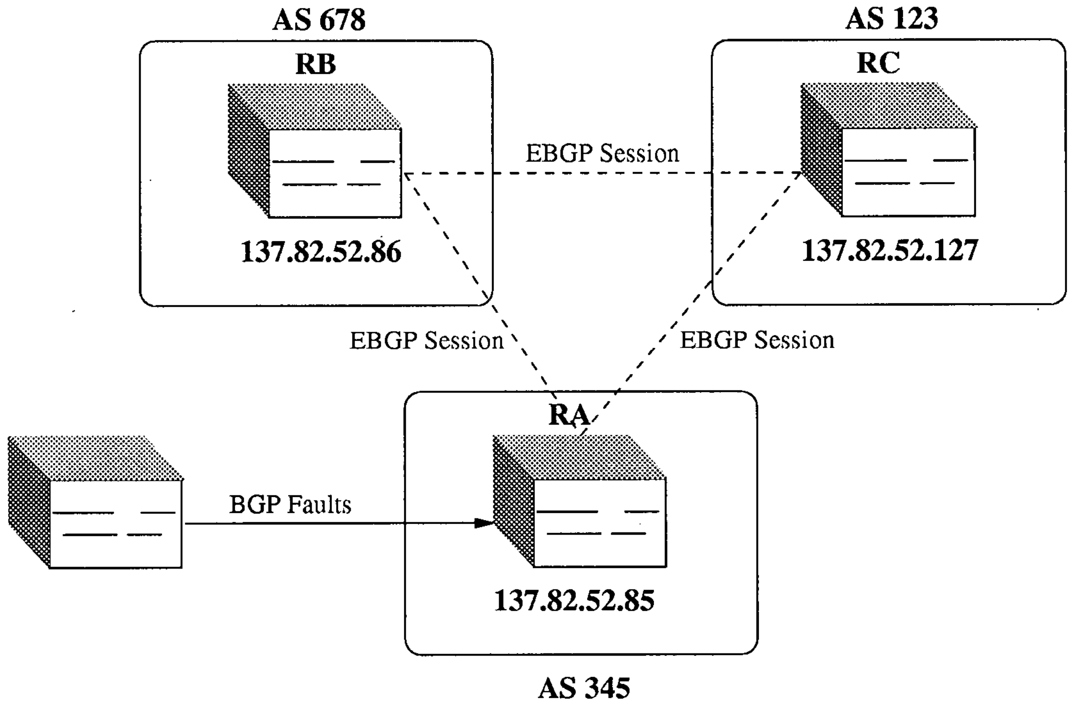


Figure 3.2: EBGP session

RB's configuration file:

```
router bgp 678
neighbor 137.82.52.85 remote-as 345
neighbor 137.82.52.127 remote-as 123
```

RC's configuration file:

```
router bgp 123
neighbor 137.82.52.85 remote-as 345
neighbor 137.82.52.86 remote-as 678
```

At first RA, RB, and RC are in *idle* state. They send an open message. Each router sends a keepalive message to its peers after receiving an open message from its peers. If a keepalive message is received, the state will go to *established.* At this stage they can start exchanging update messages.

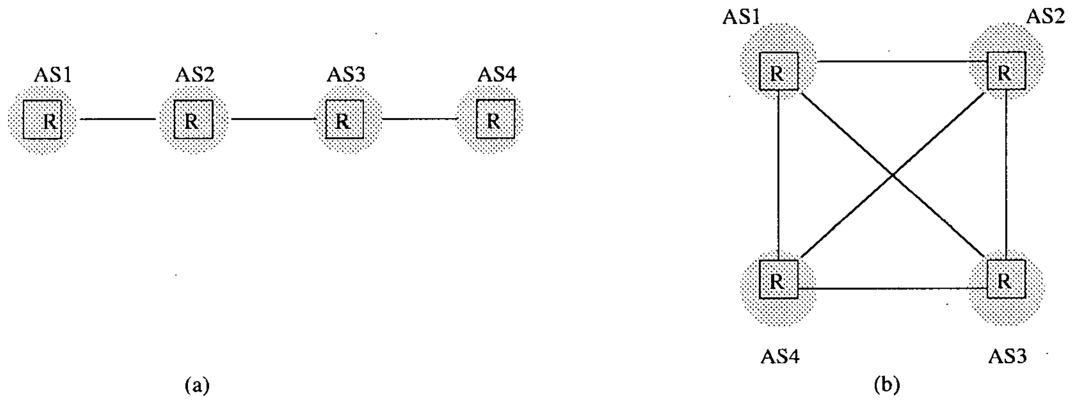


Figure 3.3: The best and worst-case topology for 4 BGP routers

BGP faults are injected into RA. RA after receiving these updates and processing them, will forward its best routes to its peers. It is possible that each router has several paths to the same prefix. In this case the best one is the primary path and others are alternate paths.

3.5 BGP Convergence Model

The BGP specification [20] does not indicate how to model BGP and it lacks behavioral and performance analysis. To study BGP convergence behavior, arbitrary network topologies have been modeled and simulated. These simulations lead to the best and the worst-case model for BGP convergence.

Figure 3.3a shows a four linearly connected BGP routers. This model, chain topology, provides the best-case complexity for BGP convergence; therefore it establishes the lower bound on the number of BGP routing updates.

Figure 3.3b shows a four-node mesh (a fully connected graph) topology. This topology makes the worst-case complexity for BGP convergence; therefore it gives the upper bound on the number of routing updates.

These models and simulations lead to inductive analysis. Next chapter provides further analysis of these models in more depth.

3.6 Summary

This chapter has provided an overview of the proposed approach for evaluating BGP4 protocol. The evaluation metrics and the BGP model used in this study are introduced. The empirical observations as well as the theoretical upper bounds and lower bounds on specified metrics are presented in the next chapter.

Chapter 4

Analysis of Routing Information

In this chapter, the inductive analysis of data collected with the experimental measurement infrastructure described in the previous chapter is presented. The number of BGP routing updates triggered by each injection of a routing event is examined. The relationship between specific Internet topological configurations and the rate of convergence is explored. Upper bounds and lower bounds on the volume of routing updates and convergence latency are developed.

4.1 Volume of routing updates when a new route is announced

In this section, the impact of network topology on number of BGP routing updates generated by announcing a new route is examined.

For this purpose the following assumptions are made:

- Expedient update message propagation (zero or minimum delay in sending update messages

- Link among autonomous system peers have the same latency. All autonomous system peers have the same processing delay.

- Decision process selects the path with the shortest AS-PATH length.

- If the AS-PATH length is the same, the BGP router ID will be a tie breaker. In this analysis the autonomous system number is used as the BGP router ID.

To develop the lower bound on the volume of routing updates, the BGP convergence behavior is studied for the best-case topology. The best-case topology is a linear connection between autonomous systems. Figure 4.1 shows the best-case topology for four BGP routers. A four-node chain topology is demonstrated in this figure. Injection of a single routing event (a route announcement) generates six update messages.

Node 1 sends an announcement for X to its neighbor, node 2, after route X is injected into node 1. Node 2 adds this new route into its routing table. Node 2 will disseminate this route to each peer located in neighboring autonomous systems (node 1 and node 3). If the receiver node performs the loop detection process, then node 1 will detect a looped path from node 2 and it will invalidate this route. If the sender node performs the loop detection process, then node 2 should send a withdrawal message to node 1. Therefore, if node 2 previously had announced a route with the same destination to node 1, node 1 must remove this route from its routing table. Other nodes will exhibit similar behavior. Finally the system will converge after generating six update messages.

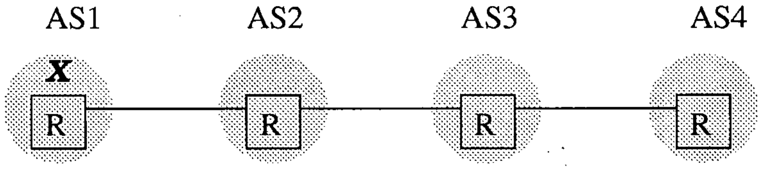


Figure 4.1: Four-node chain topology

It is found from the results of simulations that in general in a network of n autonomous systems where all autonomous systems are connected linearly, $2(n-1)$ update messages are generated until convergence. Assume node 1 is the only node that is directly connected to route X. The first and the last nodes in this network have one neighbor. Thus, they will send out one announcement for X to their neighbor. Other nodes have 2 neighbors and each will send out 2 announcements to their neighbors. Following shows the number of routing updates each node generates:

$$
\begin{aligned}
vol_1 &= 1 \\
vol_2 &= 2 \\
vol_3 &= 2 \\
&\vdots \\
vol_{n-1} &= 2 \\
vol_n &= 1
\end{aligned}
$$

Therefore, in general the total number of routing updates is given by:

$$\sum_{i=1}^{n} vol_i = 2(n-1) \tag{4.1}$$

Formula 4.1 shows the lower bound on number of routing updates (announcement and withdraws) for n linearly connected BGP routers when a new route is injected. That is the lower bound for the best-case topology; thereby it is called the global lower bound.

To develop the lower and upper bounds for the worst-case topology, the convergence behavior of BGP is examined in a fully connected graph topology. Figure 4.2 shows the worst-case topology for four BGP routers. It shows a full mesh, complete graph, of E-BGP connections among four BGP routers. Node 1 gets connected to route X. Experimental results show that this system converges after propagating twelve update messages.

At first node 1 adds X to its routing table. Then node 1 announces this new route to its peers in neighboring autonomous systems. Node 1 prepends its own autonomous system number to the AS-PATH attribute of all update messages sent to its peers. Node 2, node 3, and node 4 will receive this update message from node 1. Since this new route is not present in their routing tables, it will be placed in their routing tables. These nodes will add a new entry to their routing table for route X.

Table 4.1 shows the routing table of node 2, node 3, and node 4 after receiving the announcement for route X from node 1. The active route is denoted with an asterisk.

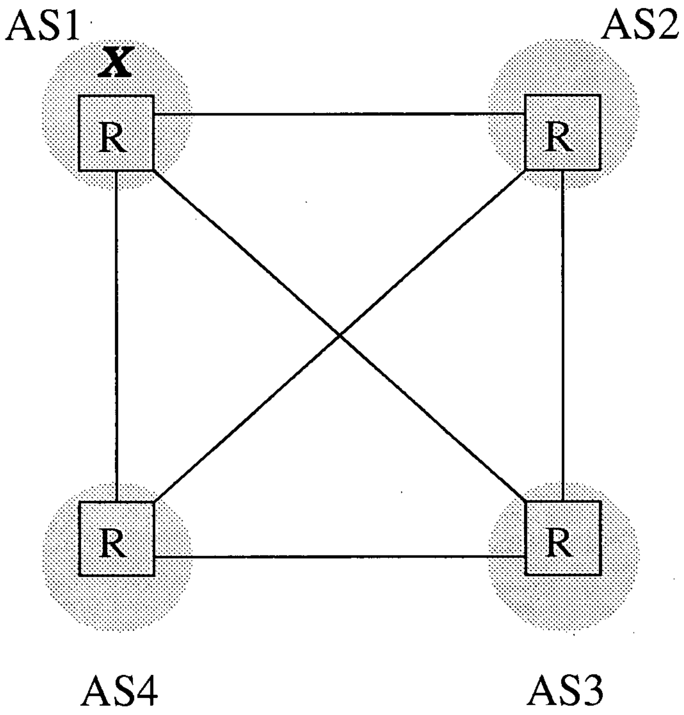


Figure 4.2: Four-node mesh topology

Table 4.1: Routing tables after receiving an announcement from node 1.

Node2	Node3	Node4
*1X	*1X	*1X

These three nodes also announce this new active route to each of their neighbors. If the sender node performs the loop detection process, these nodes will send a withdrawal message to node 1 instead of sending an announcement. If the receiver node performs the loop detection process, then these nodes will announce their active route to node 1 and node 1 will invalidate these routes after detecting loop in their AS-PATH attribute.

Now consider messages transmitted to node 2. Node 2 upon receipt of an announcement from node 1, stores (1X) as its primary path. When node 2 receives an announcement from node 3, it runs its decision process to choose the best route out of these two available paths to destination X. Decision process selects the route with the shortest AS-PATH length, (1X). Since the primary path to X is not changed, node 2 does not generate new update messages. The routing tables of these three nodes is demonstrated by Table 4.2 after convergence.

Table 4.2: Routing tables after convergence.

Node2	Node3	Node4
*1X	*1X	*1X
3-1X	2-1X	2-1X
4-1X	4-1X	3-1X

The first entry is their active route and the second and the third entries are backup paths. It can be concluded from these results that in general for a complete graph of n BGP autonomous systems, $n(n-1)$ update messages are generated. Each BGP router has $(n-1)$ neighbors. Each announcement of a new route is forwarded to all $(n-1)$ neighbors of an autonomous system, thereby generating $n(n-1)$ messages until convergence. It is found that $n(n-1)$ is the lower bound on volume of routing updates under the assumption of unbounded delay for the worst-case topology, a fully connected graph.

To develop the worst-case complexity for BGP convergence, link delays are generated among autonomous system peers. Analysis of collected data shows that link delay can affect the ordering of messages. Such an ordering represents the upper bound on number of routing updates for the worst-case topology. In [11], Labovitz *et al.* did not consider the impact of link delays on the ordering of messages.

In Figure 4.2, link delays are generated among four BGP routers. This topology and condition make the worst-case complexity for the BGP convergence. The system converges after propagating twenty-four update messages. It is found that the number of BGP routing updates is exponential with respect to the number of autonomous systems.

Node 1 sends announcements to its peers located in neighboring autonomous systems after it learns about route X. Node 3 will receive the announcement for route X from node 2 before it receives it from node 1. Since there is no path to route X in node 3's routing table, node 3 will add a new entry to its routing table for route X. This entry shows a path of length 2, (2-1X), to reach destination X. This path is its primary path to destination X. Node 4 will receive this announcement from node 3, before its gets this announcement from node 1 and node 2. This announcement shows a path of length 3, (3-2-1X). Node 4 adds (3-2-1X) to its routing table as its primary path to reach X. Then node 3 receives the announcement from node 1 and adds path (1X) to its routing table. Now there are two feasible paths to X. These two feasible paths (2-1X and 1X) should be maintained. The path with the shortest AS-PATH length is the primary path. Decision process chooses (1X). Since node 3 routing table is changed, it should announce this replacement route to its peers. Node 4 upon receipt of this announcement from node 3, runs its decision process and chooses (3-1X) as its primary path. Then node 4 announces this replacement to its peers. Node 2 keeps path (4-3-1X) as a backup path to destination X. Then node 4 receives an announcement (2-1X) from node 2. Node 4 runs its decision process. Path (2-1X) becomes the active path to X. Node 4 will announce this update to its peers. Finally node 4 receives node 1 announcement and chooses path (1X) as its best path to X. It then announces its best path to its peers located in neighboring autonomous systems. Table 4.3 shows changes in routing table of each node. A withdrawal message is denoted by W and an announcement with A.

The analysis shows that node 2 sends announcements for X just one (2^0) time. Path (1-X) is its primary path and announcements for X from its peers do not change its primary path. It is observed that node 3 sends announcements for X two (2^1) times, first time by receiving an announcement from node 2 and second time from node 1. Node 4 receives updates for X from node 3 two times and one time from node 2 and node 1. Therefore node 4 sends updates for X four (2^2) times.

If the number of autonomous system increases, volume of routing updates will grow exponentially. This is shown in Table 4.4.

It is inferred from these results that in the worst-case in a complete graph of n autonomous systems $(n-1)*(2^{(n-1)})$ update messages are generated until convergence. Assume node 1 is the only node that is directly connected to route X. Each node has $(n-1)$ neighbors. Therefore, node 1 will send $(n-1)$ updates to announce X to its neighbors. Node 2 will announce X one time. Therefore, it will generate $(n-1)$ updates. Node 3 will receive announcement for X first from node 2 and then from node 1. Node 3 will generate $2*(n-1)$ updates. Following shows the number of routing updates each node generates:

$$\begin{aligned}
vol_1 &= n-1 \\
vol_2 &= 2^0 * (n-1) \\
vol_3 &= 2^1 * (n-1) \\
vol_4 &= 2^2 * (n-1) \\
vol_5 &= 2^3 * (n-1) \\
&\vdots \\
vol_n &= ((2^{(n-2)}) * (n-1))
\end{aligned}$$

$$\begin{aligned}
\text{volume of routing updates} &= \sum_{i=1}^{n} vol_i \\
&= (n-1) + \sum_{i=0}^{n-2} 2^i \\
&= (n-1) + ((2^{n-1}) - 1)(n-1)
\end{aligned}$$

$$\text{volume of routing updates} = ((n-1) * (2^{(n-1)})) \tag{4.2}$$

Formula 4.2 shows the upper bound on number of routing updates (announcement and withdraws) in a system of n BGP routers when a new route is injected. It

Table 4.3: Changes in routing tables of 3 nodes for convergence-worst case.

Node2	Node3	Node4
* A(1X)		
	* A(2-1X)	
		* A(3-2-1X)
	* A(1X)	
A(3-1X)		* A(3-1X)
A(4-3-1X)		
		* A(2-1X)
W(4-3-1X)	A(4-2-1X)	
		* A(1X)
A(4-1X)	A(4-1X)	

Table 4.4: Changes in routing tables of 4 nodes for convergence-worst case.

Node2	Node3	Node4	Node5
* A(1X)			
	* A(2-1X)		
		* A(3-2-1X)	
			* A(4-3-2-1X)
	* A(1X)		
A(3-1X)		* A(3-1X)	
			* A(4-3-1X)
A(5-4-3-1X)		* A(2-1X)	
	A(4-2-1X)		* A(4-2-1X)
W(5-4-3-1X)	A(5-4-2-1X)	* A(1X)	
A(4-1X)	A(4-1X)		* A(4-1X)
A(5-4-1X)	A(5-4-1X)		* A(3-1X)
A(5-3-1X)	W(5-4-1X)	A(5-3-1X)	* A(2-1X)
W(5-3-1X)	A(5-2-1X)	A(5-2-1X)	* A(1X)
A(5-1-X)	A(5-1X)	A(5-1X)	

is the upper bound for the worst-case topology; therefore it is called the global upper bound.

The relationship between number of autonomous systems and number of routing updates is quantitatively illustrated in Figure 4.3. The vertical axis provides the total numbers of routing updates during the process of BGP convergence, and the horizontal axis shows the number of autonomous systems. The first row of bars, light gray bars, shows the lower bound for the best-case topology, which is the global lower bound, the third row of bars, dark gray bars, shows the upper bound for the worst-case topology, which is the global upper bound. The second row of bars, medium gray bars, shows the lower bound for the worst-case topology.

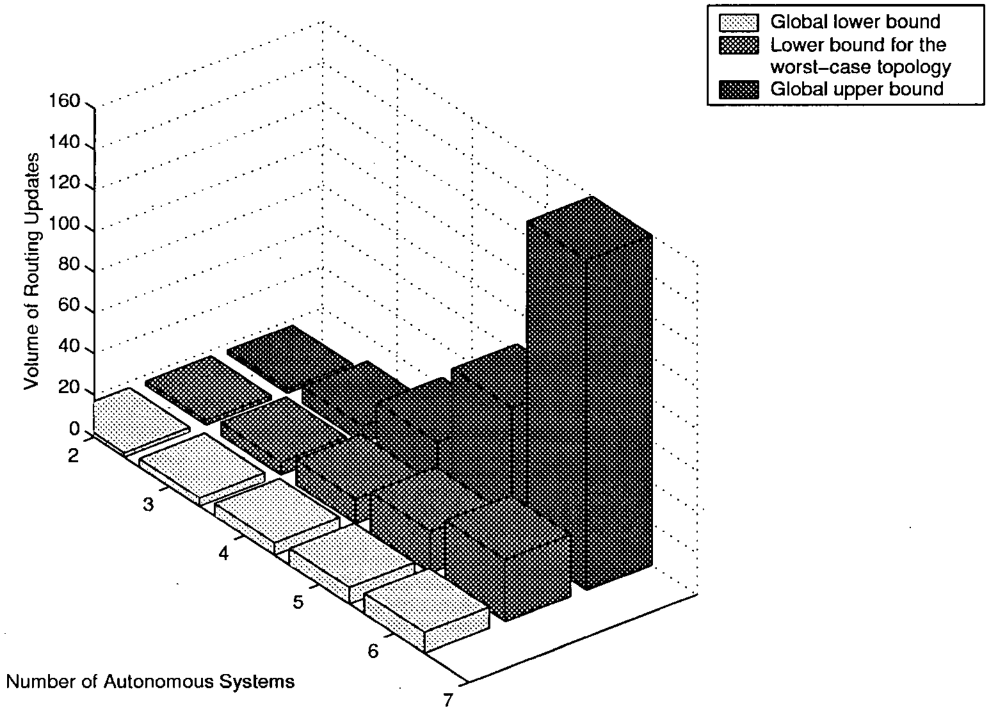


Figure 4.3: Volume of routing updates triggered by a route announcement

4.2 Volume of routing updates when an available route is withdrawn

In this section the impact of network topology on number of BGP routing updates generated by withdrawing a previously available route is explored

Following assumptions are made:

- Expedient update message propagation
- Link among autonomous system peers have same latency and all autonomous system peers have same processing delay
- Decision process selects the path with the shortest AS-PATH length.
- If the AS-PATH length is the same, the BGP router ID will be a tie breaker. In this analysis the autonomous system number is used as the BGP router ID.

The BGP convergence behavior is studied for the best-case topology in order to develop the lower bound on the volume of routing updates. As demonstrated before in Figure 4.1, node 1 is initially connected to route X. Route X is withdrawn following a fault. Node 1 withdraws its directly connected path from its routing table. Lacking a valid route to X, node 1 then sends out a withdrawal message to its neighbor, node 2. Upon receipt of this withdrawal message, node 2 invalidates its path to X. Node 3 and node 4 exhibit similar behavior. This system converges after propagating six update messages.

This topology, a chain, generates same number of routing updates, $2(n-1)$, upon receipt of either a route failure and or a route announcement. $2(n-1)$ is the lower bound on volume of routing updates for the best case, linearly connected, topology. Thereby it is the global lower bound.

In the worst case, a fully connected graph topology, link or processing delays can affect the ordering of update messages. To evaluate the volume of routing updates, link and processing delays are generated in the network. It is found by experimentation that volume of routing updates is factorial with respect to the numbers of nodes in the network.

Figure 4.2 shows the worst-case topology for four BGP routers. It shows a complete graph of four nodes, where node 1 is initially directly connected to route X. Route X is withdrawn following a failure. The empirical results show that this system converges after propagating sixty-three update messages.

The initial routing tables at steady state prior to route withdrawal are shown in Table 4.5.

Table 4.5: Stage 0-Routing tables before any routing event

Node2	Node3	Node4
*1X	*1X	*1X
3-1X	2-1X	2-1X
4-1X	4-1X	3-1X

Node 1 sends withdrawals messages to its peers. Node 2, 3,and 4 invalidate the first entry of their routing tables. They all run their decision process since the previously announced route is no longer available for use. Each node in addition to its primary path to destination X, stores alternative paths at most one per neighbor. These nodes choose the secondary entry in their routing table. Node 2 selects (3-1X), node 3 selects (2-1X), and node 4 selects (2-1X).

These three nodes announce their active route to each peer located in neighboring autonomous systems. If receiver nodes perform loop detection, then all nodes can send their announcements to their peers located in neighboring autonomous systems. BGP convergence behavior when the receiver router performs loop detection is presented in the rest of this section.

Table 4.6: Stagel-Routing tables

Node2	Node3	Node4
*3-1X	*2-1X	*2-1X
4-1X	4-1X	3-1X

Messages generated at this stage		
$2 \rightarrow 1(A\ 231X)$	$3 \rightarrow 1(A\ 321X)$	$4 \rightarrow 1(A\ 421X)$
$2 \rightarrow 3(A\ 231X)$	$3 \rightarrow 2(A\ 321X)$	$4 \rightarrow 2(A\ 421X)$
$2 \rightarrow 4(A\ 231X)$	$3 \rightarrow 4(A\ 321X)$	$4 \rightarrow 3(A\ 421X)$

In stage 1, each node (except node 1) announces path of length 3. Node 2 sends out (2-3-1X), node 3 sends out (3-2-1X), and node 4 sends out (4-2-1X). In the next stage, messages generated by node 2 are processed. The newly announced path for node 3 creates a routing loop. Therefore, node 3 rejects it and deletes the path (2-1-X) from its routing table. Node 3 selects its alternate path and sends out new updates for X, (3-4-1X). The announcement from node 2 replaces an existing path in the routing table of node 4. The path being placed is shorter than the path replacing it. Therefore, node 4 selects the other alternate path, (3-1X), and sends out new announcements.

The routing table of each node, processed messages and propagated messages by each node at each stage for convergence are shown in the following.

Table 4.7: Stage 2-Routing tables

Node2	Node3	Node4
*3-1X	*4-1X	*3-1X
4-1X	-	2-3-1X

Messages processed	Messages generated at this stage	
$2 \rightarrow 1(A\ 231X)$	$3 \rightarrow 1(A\ 341X)$	$4 \rightarrow 1(A\ 431X)$
$2 \rightarrow 3(A\ 231X)$	$3 \rightarrow 2(A\ 341X)$	$4 \rightarrow 2(A\ 431X)$
$2 \rightarrow 4(A\ 231X)$	$3 \rightarrow 4(A\ 341X)$	$4 \rightarrow 3(A\ 431X)$

In stage 3, update messages from node 3, (3-2-1X), are processed. Node 2 eliminates (3-1X) from its routing table and chooses (4-1X) as its primary path to X. Therefore node 2 sends out new announcements. Node 4 upon receipt of a new announcement from node 3 replaces (3-1X) by (3-2-1X) and chooses (2-3-1X) as its primary path to X. Node 4 sends new announcements for X to its neighbors.

Table 4.8: Stage 3-Routing tables

Node2	Node3	Node4
*4-1X	*4-1X	*2-3-1X
-	-	3-2-1X

Messages processed	Messages generated at this stage	
$3 \rightarrow 1(A\ 321X)$	$2 \rightarrow 1(A\ 241X)$	$4 \rightarrow 1(A\ 4231X)$
$3 \rightarrow 2(A\ 321X)$	$2 \rightarrow 3(A\ 241X)$	$4 \rightarrow 2(A\ 4231X)$
$3 \rightarrow 4(A\ 321X)$	$2 \rightarrow 4(A\ 241X)$	$4 \rightarrow 3(A\ 4231X)$

In stage 4, update messages from node 4, (4-2-1X), are processed. Node 2 eliminates (4-1X) from its routing table. Lacking a valid path to X, node 2 sends out withdrawal messages to it peers. Node 3 upon receipt of a new announcement from node 4 replaces (4-1X) by (4-2-1X) and sends new announcement for X to its neighbors.

In stage 5, update messages from node 3, (3-4-1X), are processed. Node 2 adds (3-4-1X) to its routing table and sends out new announcements. Node 4 cannot

Table 4.9: Stage 4-Routing tables

Node2	Node3	Node4
-	*4-2-1X	*2-3-1X
-	-	3-2-1X

Messages processed	Messages generated at this stage
$4 \rightarrow 1(A\ 421X)$	$2 \rightarrow 1W \quad 3 \rightarrow 1(A\ 3421X)$
$4 \rightarrow 2(A\ 421X)$	$2 \rightarrow 3W \quad 3 \rightarrow 2(A\ 3421X)$
$4 \rightarrow 3(A\ 421X)$	$2 \rightarrow 4W \quad 3 \rightarrow 4(A\ 3421X)$

receive this update from node 3 because of the link delay. Therefore at this stage the routing table of node 4 does not change.

In stage 6, update messages from node 4, (4-3-1X), are processed. Node 2 adds a new entry to its routing table and stores (4-3-1X) as an alternate path to X. Node 3 upon receipt of a new announcement from node 4, detects a looped path. Node 3 does not have any path to X and thereby sends out withdrawals messages to its peers.

In stage 7, update messages from node 2, (2-4-1X), are processed. Node 3 adds a new entry to routing table and sends a new announcement, (3-2-4-1X), to its peers. Node 4 eliminates (2-3-1X) from its routing table and chooses (3-2-1X) as the active path to X.

In stage 8, update messages from node 4, (4-2-3-1X), are processed. Node 4 receives the update message from node 3 that advertise (3-4-1X). It detects the looped path and sends out withdrawals to its peers. Node 2 cannot receive the update form node 4 because of the link delay.

In stage 9, the withdrawals from node 2 are processed. Node 3 removes the only path to X and sends out withdrawals to its peers.

In stage 10, update messages from node 3, (3-4-2-1X), are processed. Node 2 removes (3-4-1X) from its routing table and chooses the only alternate path as the primary path to X.

Table 4.10: Stage 5-Routing tables

Node2	Node3	Node4
*3-4-1X	*4-2-1X	*2-3-1X
-	-	3-2-1X

Messages processed	Messages generated. at this stage
$3 \rightarrow 1(A\ 341X)$	$2 \rightarrow 1(A\ 2341X)$
$3 \rightarrow 2(A\ 341X)$	$2 \rightarrow 3(A\ 2341X)$
	$2 \rightarrow 4(A\ 2341X)$

Table 4.11: Stage 6-Routing tables

Node2	Node3	Node4
*3-4-1X	-	*2-3-1X
4-3-1X	-	3-2-1X

Messages processed	Messages generated at this stage
$4 \rightarrow 1(A\ 431X)$	$3 \rightarrow 1W$
$4 \rightarrow 2(A\ 431X)$	$3 \rightarrow 2W$
$4 \rightarrow 3(A\ 431X)$	$3 \rightarrow 4W$

Table 4.12: Stage 7-Routing tables

Node2	Node3	Node4
*3-4-1X	*2-4-1X	*3-2-1X
4-3-1X	-	-

Messages processed	Messages generated at this stage	
$2 \rightarrow 1(A\ 241X)$	$3 \rightarrow 1(A\ 3241X)$	$4 \rightarrow 1(A\ 4321X)$
$2 \rightarrow 3(A\ 241X)$	$3 \rightarrow 2(A\ 3241X)$	$4 \rightarrow 2(A\ 4321X)$
$2 \rightarrow 4(A\ 241X)$	$3 \rightarrow 4(A\ 3241X)$	$4 \rightarrow 3(A\ 4321X)$

Table 4.13: Stage 8-Routing tables

Node2	Node3	Node4
*3-4-1X	*2-4-1X	-
4-3-1X	-	--

Messages processed	Messages generated at this stage
$4 \rightarrow 1(A\ 4231X)$	$4 \rightarrow 1W$
$4 \rightarrow 3(A\ 4231X)$	$4 \rightarrow 2W$
$3 \rightarrow 4(A\ 341X)$	$4 \rightarrow 3W$

Table 4.14: Stage 9-Routing tables

Node2	Node3	Node4
*3-4-1X	-	-
4-3-1X	-	-

Messages processed	Messages generated at this stage
$2 \rightarrow 1W$	$3 \rightarrow 1W$
$2 \rightarrow 3W$	$3 \rightarrow 2W$
$2 \rightarrow 4W$	$3 \rightarrow 4W$

Table 4.15: Stage 10-Routing tables

Node2	Node3	Node4
*4-3-1X	-	-

Messages processed	Messages generated at this stage
$3 \rightarrow 1(A\ 3421X)$	$2 \rightarrow 1(A\ 2431X)$
$3 \rightarrow 2(A\ 3421X)$	$2 \rightarrow 3(A\ 2431X)$
$3 \rightarrow 4(A\ 3421X)$	$2 \rightarrow 4(A\ 2431X)$

Table 4.16: Stage 11-Routing tables

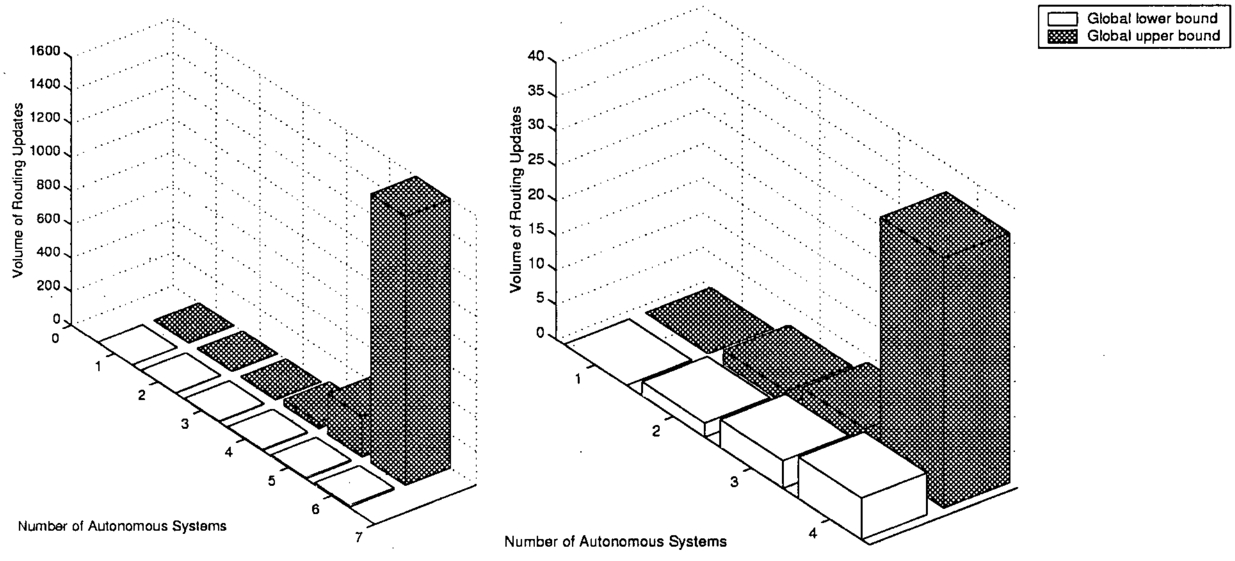
Node1	Node2	Node3
-	-	-

Messages processed	Messages generated at this stage
$2 \rightarrow 1(A\ 2341X)$	$2 \rightarrow 1W$
$2 \rightarrow 3(A\ 2341X)$	$2 \rightarrow 3W$
$2 \rightarrow 4(A\ 2341X)$	$2 \rightarrow 4W$
$4 \rightarrow 2(A\ 4231X)$	

In stage 11, update messages from node 2, (2-3-4-1X) are processed. Node 2 receives the update message, (4-2-3-1X), from node 4. In this stage node 2 eliminates the only path to X from its routing table and sends out withdrawals to its peers.

Processing other messages in the queue of each node does not change their routing table. Finally the system converges with all routes withdrawn.

Figure 4.4a shows the simulation resulting from injecting a route failure in system ranging size from 1 to 6. Figure 4.4b shows the simulation results in system ranging size from 1 to 4 with a different vertical scale. These figures quantitatively show the relationship between number of updates and number of autonomous systems. The vertical axis demonstrated the total number of announcements observed during the process of BGP convergence and the horizontal axis shows the number of autonomous systems. The first row bar, white bars, in these figures shows the lower bound for the best-case topology which is the global lower bound and the second row bar,gray bars, shows the upper bound for the worst-case topology, which is the global upper bound.



(a)

(b)

Figure 4.4: Volume of routing updates triggered by a route withdrawal

Each node after receiving withdrawal update from node 1, explores all $x = 2, 3, ..., n-1$ length paths until convergence. In a complete graph of n nodes, $(n-1)(n-2)$ paths of length 3 exist to reach a particular destination in a graph. Any other path of length greater than 3 must use one of these $(n-1)(n-2)$ paths as the last hop in order to reach that destination.

Therefore in general total number of paths to reach a destination is given by:

$$\begin{aligned}
&(n-1)(n-2)+(n-1)(n-2)(n-3)+\cdots+(n-1)! \\
&=\frac{(n-1)!}{(n-3)!}+\frac{(n-1)!}{(n-4)!}+\cdots+\frac{n!}{2!}+(n-1)! \\
&=(n-1)! * \sum_{i=1}^{n-3}\frac{1}{i!} \qquad (4.3)
\end{aligned}$$

Each BGP router will send an announcement of a new path (backup path) to its $(n-1)$ neighbors.

Therefore:

$$\text{Total Updates (announcements)} = (n-1)(n-1)! * \sum_{i=1}^{n-3}\frac{1}{i!} \qquad (4.4)$$

Knowing:

$$\mathrm{e}=\sum_{k=1}^{\infty}\frac{1}{k!}$$

Equation 4.4 for $n > 3$ can be rewritten as:

$$\text{Total Updates (announcements)} \approx (n-1) * ((n-1)!e) \qquad (4.5)$$

Formula 4.5 shows the upper bound on volume of routing updates when a route is withdrawn. Since $n!$ is approximately n^n, $O(n^n)$ is found to be a good approximation for the upper bound on volume of updates. This is the upper bound for the worst-case topology; therefore it is the global upper bound on volume of routing updates.

It is demonstrated in this example that large number of messages are generated. It shows that BGP has bouncing problem in contrary to what is believed in [4] . At stage 1 node 2, invalidates its primary path that used node 1 to reach X, and then runs it decision process. This process chooses the alternative path,(3-1X), although node 3 cannot reach X through node 1. Node 2 sends out a new announcement upon this change to its peers. Node 3 and node 4 also invalidates their first entry in their routing tables and selects the second entry as their best path to X. Node 3 and node 4 receive the update message from node 2, and they change their primary path again. Node 3 chooses (4-1X) and node 4 chooses (3-1X). These nodes select paths, which use node 1 to reach X. These paths are all invalid. This process continues till all nodes eliminate all entries in their routing table.

To solve the bouncing problem we suggest that all node 1's peers in neighboring autonomous systems upon receipt of the withdrawal message from node 1 search their routing table and eliminate all the entries, which their AS-PATH attribute includes node 1 to reach X. If this optimization gets implemented on routers, the system will converge faster. In this experiment, the system will converge by generating 3 update messages.

4.3 Volume of routing updates under MinRouteAdvertisementInterval

In the previous sections, the behavior of BGP convergence under the assumption of expedient update message propagation was described. This section analyzes BGP convergence behavior when all messages propagate within the bounds of the MinRouteAdvertisementInterval timer.

The BGP protocol constraints the amount of routing traffic (update messages) in order to limit both the link bandwidth needed to announce update messages and the processing peer needed by the decision process to digest the information contained in the update messages. The parameter MinRouteAdvertismentInterval determines the minimum amount of time that must elapse between announcements of routes to a particular destination from a single BGP speaker. This rate limiting procedure applies on a per-destination basis, although the value of MinRouteAdvertisementInterval is set on a per BGP peer basis [20].

Two update messages sent from a single BGP speaker that announce feasible routes to some common set of destinations received from BGP speakers in neighboring autonomous systems must be separated by at least MinRouteAdvertisementInterval. Clearly, this can only be achieved precisely by keeping a separate timer for each common set of destinations.

Since fast convergence is needed within an autonomous system, this procedure does not apply for routes received from other BGP speakers in the same autonomous system. To avoid long-lived black holes, the procedure does not apply to the explicit withdrawal of unfeasible routes.

This procedure does not limit the rate of route selection, but only the rate of route announcement. If new routes are selected multiple times while awaiting the expiration of MinrouteAdverisementInterval, the last route selected must be announced at the end of MinrouteAdvertisementInterval [20].

This technique is capable of reducing router-processing load. A BGP implementation must be prepared for a large volume of routing traffic. MinRouteAdvertisementInterval timer limits the propagation of unnecessary changes as routing topology grows.

The suggested value for the MinRouteAdvertisementInterval is 30 seconds [20]. Labovitz *et al.* in [10] indicate that the vast majority of BGP messages propagate within 30 seconds. To understand the effect of MinRouteAdvertisementInterval timer, BGP convergence is studied for the worst-case topology. Figure 4.5 shows the simulation results by injecting a route failure in a system of ranging sizes from 3 to 6 autonomous systems with and without implementing MinRouteAdvertisementInterval timer. Figure 4.6 shows the worst-case topology. MinRouteAdvertisementInterval timer is implemented on each router. It is assumed that all BGP messages propagate within 30 seconds.

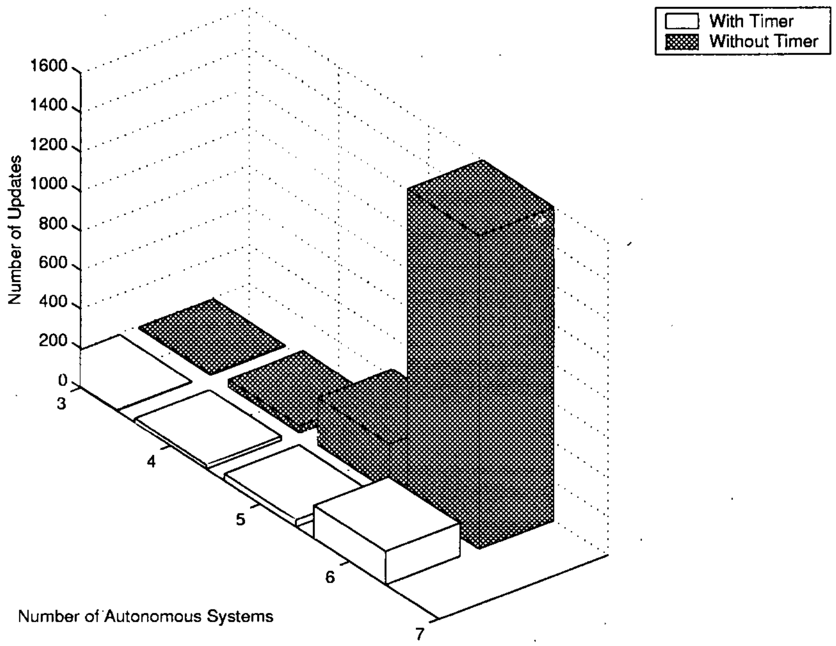


Figure 4.5: Simulation Results due to MinRouteAdvertisementInterval

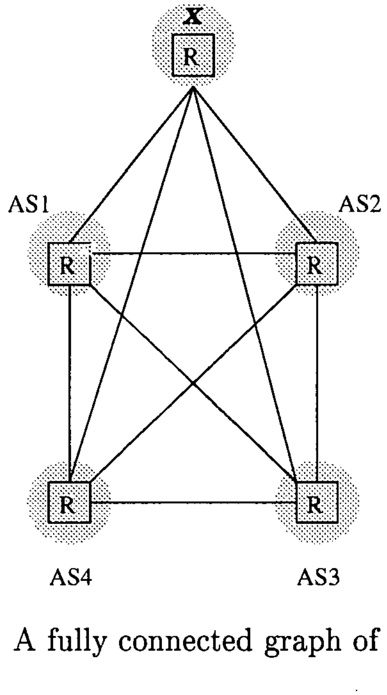


Figure 4.6: A fully connected graph of five routers

When route X is withdrawn, each node invalidates its primary path to X and selects a new active route to X. Then each node announces its active route to its peers located in neighboring autonomous systems. Upon receipt of these update messages, each node selects another path to X, but they cannot announce their new active routes until MinRouteAdvertisementInterval timer expires. It is possible that new routes be selected multiple times while awaiting the expiration of MinRouteAdvertisementInterval timer. However each node announces the last route selected after the timer expires.

Stages for system convergence are shown in the following (considering only receiver nodes perform loop detection):

In stage 1 all nodes send out announcements to their peers for route X. They cannot send a new announcement for X to their peers until MinRouteAdvertisementInterval timer expires. Each node will receive 3 announcements from its neighbor. At stage 5, node 1 sends out withdrawals messages to its neighbors that it has no valid path to X. MinRouteAdvertisementInterval timer expires and node 2 and 3 send out new announcements. Node 2 at stage 9, node 4 at stage 11 and node 3 at stage 12 send out withdrawals messages.

Table 4.17: Stage 1-Routing tables

Node1	Node2	Node3	Node4
*2X	*1X	*1X	*1X
3X	3X	2X	2X
4X	4X	4X	3X

Messages generated at this stage			
$1 \rightarrow 2(A\ 12X)$	$2 \rightarrow 1(A\ 21X)$	$3 \rightarrow 1(A\ 31X)$	$4 \rightarrow 1(A\ 41X)$
$1 \rightarrow 3(A\ 12X)$	$2 \rightarrow 3(A\ 21X)$	$3 \rightarrow 2(A\ 31X)$	$4 \rightarrow 2(A\ 41X)$
$1 \rightarrow 4(A\ 12X)$	$2 \rightarrow 4(A\ 21X)$	$3 \rightarrow 4(A\ 31X)$	$4 \rightarrow 3(A\ 41X)$

Table 4.18: Stage 2-Routing tables

Node1	Node2	Node3	Node4
*2X	*3X	*2X	*2X
3X	4X	4X	3X
4X	-	1-2X	1-2X

Messages processed at this stage
$1 \rightarrow 2(A\ 12X)$
$1 \rightarrow 3(A\ 12X)$
$1 \rightarrow 4(A\ 12X)$

Table 4.19: Stage 3-Routing tables

Node1	Node2	Node3	Node4
*3X	*3X	*4X	*3X
4X	4X	2-1X	1-2X
-	-	1-2X	2-1X

Messages processed at this stage
$2 \rightarrow 1(A\ 21X)$
$2 \rightarrow 3(A\ 21X)$
$2 \rightarrow 4(A\ 21X)$

Table 4.20: Stage 4-Routing tables

Node1	Node2	Node3	Node4
*4X	*4X	*4X	*1-2X
-	3-1X	1-2X	2-1X
-	-	2-1X	3-1X

Messages processed at this stage
$3 \rightarrow 1(A\ 31X)$
$3 \rightarrow 2(A\ 31X)$
$3 \rightarrow 4(A\ 31X)$

Table 4.21: Stage 5-Routing tables

Node1	Node2	Node3	Node4
-	*3-1X	*1-2	*1-2X
-	4-1X	2-1X	2-1X
-	-	4-1X	3-1X

Messages processed
$4 \rightarrow 1(A\ 41X)$
$4 \rightarrow 2(A\ 41X)$
$4 \rightarrow 3(A\ 41X)$

Messages generated at this stage			
$1 \rightarrow 2W$	$2 \rightarrow 1(A\ 231X)$	$3 \rightarrow 1(A\ 312X)$	$4 \rightarrow 1(A\ 412X)$
$1 \rightarrow 3W$	$2 \rightarrow 3(A\ 231X)$	$3 \rightarrow 2(A\ 312X)$	$4 \rightarrow 2(A\ 412X)$
$1 \rightarrow 4W$	$2 \rightarrow 4(A\ 231X)$	$3 \rightarrow 4(A\ 312X)$	$4 \rightarrow 3(A\ 412X)$

Table 4.22: Stage 6-Routing tables

Node1	Node2	Node3	Node4
-	*3-1X	*2-1X	*2-1X
-	4-1X	4-1X	3-1X

Messages processed at this stage
$1 \rightarrow 2W$
$1 \rightarrow 3W$
$1 \rightarrow 4W$

Table 4.23: Stage 7-Routing tables

Node1	Node2	Node3	Node4
-	*3-1X	*4-1X	*3-1X
-	4-1X	-	2-3-1X

Messages processed at this stage
$2 \rightarrow 1(A\ 231X)$
$2 \rightarrow 3(A\ 231X)$
$2 \rightarrow 4(A\ 231X)$

Table 4.24: Stage 8-Routing tables

Node1	Node2	Node3	Node4
*4X	*4-1X	*4-1X	*2-3-1X
			3-1-2X

Messages processed at this stage
$3 \rightarrow 1(A\ 312X)$
$3 \rightarrow 2(A\ 312X)$
$3 \rightarrow 4(A\ 312X)$

Table 4.25: Stage 9-Routing tables

Node1	Node2	Node3	Node4
-	-	*4-1-2X	*2-3-1X
-	-	-	3-1-2X

Messages processed	Messages generated at this stage
$4 \rightarrow 1(A\ 412X)$	$2 \rightarrow 1W \quad 3 \rightarrow 1(A\ 3412X) \quad 4 \rightarrow 1(A\ 4231X)$
$4 \rightarrow 2(A\ 412X)$	$2 \rightarrow 3W \quad 3 \rightarrow 2(A\ 3412X) \quad 4 \rightarrow 2(A\ 4231)$
$4 \rightarrow 3(A\ 412X)$	$2 \rightarrow 4W \quad 3 \rightarrow 4(A\ 3412X) \quad 4 \rightarrow 3(A\ 4231X)$

Table 4.26: Stage 10-Routing tables

Node1	Node2	Node3	Node4
-	-	*4-1-2X	*3-1-2X

Messages processed at this stage
$2 \to 1W$
$2 \to 3W$
$2 \to 4W$

Table 4.27: Stage 11-Routing tables

Node1	Node2	Node3	Node4
-	-	*4-1-2X	-

Messages processed	Messages generated at this stage
$3 \to 1(A\ 3412X)$	$4 \to 1W$
$3 \to 2(A\ 3412X)$	$4 \to 2W$
$3 \to 4(A\ 3412X)$	$4 \to 3W$

Table 4.28: Stage 12-Routing tables

Node1	Node2	Node3	Node4
-	-	-	-

Messages processed	Messages generated at this stage
$4 \to 1(A\ 4231X)$	$3 \to 1W$
$4 \to 2(A\ 4231X)$	$3 \to 2W$
$4 \to 3(A\ 4231X)$	$3 \to 4W$

In the last stages withdrawals from node 3 and node 4 are processed. The results show that 39 messages are generated until convergence.

The first six stages for system convergence considering only sender nodes perform loop detection are shown in the following.

In stage 1 node 1 sends out a new announcement to its peers. Node 1 cannot send its active path to node 2, instead it sends out a withdrawal message to node 2. Node 2, 3, and 4 exhibit similar behavior.

Table 4.29: Stage 1-Routing tables

Nodel	Node2	Node3	Node4
*2X	*1X	*1X	*1X
3X	3X	2X	2X
4X	4X	4X	3X

Messages generated at this stage			
$1 \rightarrow 2W$	$2 \rightarrow 1W$	$3 \rightarrow 1W$	$4 \rightarrow 1W$
$1 \rightarrow 3(A\ 12X)$	$2 \rightarrow 3(A\ 21X)$	$3 \rightarrow 2(A\ 31X)$	$4 \rightarrow 2(A\ 41X)$
$1 \rightarrow 4(A\ 12X)$	$2 \rightarrow 4(A\ 21X)$	$3 \rightarrow 4(A\ 31X)$	$4 \rightarrow 3(A\ 41X)$

In stage 2 messages sent by node 1 are processed. Node 2,3, and 4 upon receipt of this update, eliminate the first entry in their routing tables and select the second entry. Node 2 cannot send a new announcement to node 3 and node 4 until MinRouteAdvertisementInterval timer expires. Node 2 sends out the new announcement to node 1. Node 3 and node 4 also send the new announcement to node 1.

In stage 3 updates from node 2 are processed. Only node 1 can send a new announcement to node 2.

In stage 4, all nodes are still waiting for the MinRouteAdvertisementInterval timers to expire; therefore they cannot advertise for route X.

In stage 4 messages from node 4 are processed. MinRouteAdvertisementInterval timers expire; therefore each node can send out new update messages.

Table 4.30: Stage 2-Routing tables

Node1	Node2	Node3	Node4
*2X	*3X	*2X	*2X
3X	4X	4X	3X
4X	-	1-2X	1-2X

Messages processed	Messages generated at this stage
$1 \rightarrow 2W$	$2 \rightarrow 1(A\ 23X)$
$1 \rightarrow 3(A\ 12X)$	$3 \rightarrow 1(A\ 32X)$
$1 \rightarrow 4(A\ 12X)$	$4 \rightarrow 1(A\ 42X)$

Table 4.31: Stage 3-Routing tables

Node1	Node2	Node3	Node4
*3X	*3X	*4	*3X
4X	4X	1-2X	1-2X
-	-	2-1X	2-1X

Messages processed	Messages generated at this stage
$2 \rightarrow 1W$	$1 \rightarrow 2(A\ 13X)$
$2 \rightarrow 3(A\ 21X)$	
$2 \rightarrow 4(A\ 21X)$	

Table 4.32: Stage 4-Routing tables

Node1	Node2	Node3	Node4
*4X	*4X	*4	*1-2X
-	3-1X	1-2X	2-1X
-	-	2-1X	3-1X

Messages processed	Messages generated at this stage
$3 \rightarrow 1W$	
$3 \rightarrow 2(A\ 31X)$	
$3 \rightarrow 4(A\ 31X)$	

Table 4.33: Stage 5-Routing tables

Node1	Node2	Node3	Node4
-	*3-1X	*1-2X	*1-2X
-	4-1X	2-1X	2-1X
-	-	4-1X	3-1X

Messages processed
$4 \rightarrow 1W$
$4 \rightarrow 2(A\ 41X)$
$4 \rightarrow 3(A\ 41X)$

Messages generated at this stage
$1 \rightarrow 2W \quad 2 \rightarrow 1W \quad 3 \rightarrow 1W \quad 4 \rightarrow 1W$
$1 \rightarrow 3W \quad 2 \rightarrow 3W \quad 3 \rightarrow 2W \quad 4 \rightarrow 2W$
$1 \rightarrow 4W \quad 2 \rightarrow 4(A\ 231X) \quad 3 \rightarrow 4(A\ 312X) \quad 4 \rightarrow 3(A\ 412X)$

Table 4.34: Stage 6-Routing tables

Nodel	Node2	Node3	Node4
*2-3X	*3-1X	*1-2X	*1-2X
-	4-1X	2-1X	2-1X
-	-	4-1X	3-1X

Messages processed	Messages generated at this stage
$2 \rightarrow 1(A\ 23X)$	$1 \rightarrow 2W$
	$1 \rightarrow 3W$
	$1 \rightarrow 4W$

It is learned from these experimentations that MinRouteAdvertisementInterval timers limit the number of BGP update messages. Without MinRouteAdvertisementInterval over 200 announcements are propagated. Performing MinRouteAdvertisementInterval and receiver side loop detection significantly reduces the number of routing updates.

4.4 Convergence Latency

The objective of this section is to understand the relation between convergence latency and number of BGP routers. Convergence latency is a function of many parameters such as link and router processing delay.

First an example is provided to illustrate BGP convergence behavior. Figure 4.7 shows one possible topology for n BGP routers. Assume node 1 gets connected to route X. It runs its decision process and sends a new announcement to node 2. Node 2 processes this message and after learning its best path to X, transmits a new announcement to its peers. All other nodes exhibit similar behavior.

If node 1 at time 0 gets connected to X, then:

$$
\begin{aligned}
Convergence_d &= (n * R_d) + (n * L_d) \\
&= n * (R_d + L_d)
\end{aligned}
$$

Where:

$Converge_d$: Convergence latency

R_d: Average router processing delay

L_d: Average link delay between any two BGP peers

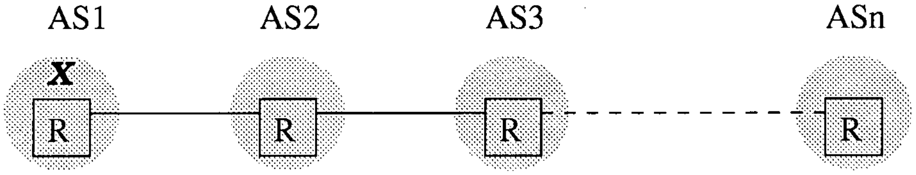


Figure 4.7: n-node chain topology

To understand the relationship between convergence latency and number of BGP routers, BGP convergence is studied for a complete graph of autonomous systems.

In a complete graph of n nodes, assume one node (node 1) gets connected directly to destination X. This node announces route X to its $(n-1)$ peers. Each of $(n-1)$ nodes upon receiving this update and processing this message, selects its active path to X and then sends an announcement for X to its peers. The system will converge after all nodes send out their active path to their peers. It is found that the upper bound on convergence latency for a route announcement depends on the number of hops between any two nodes, since path with fewer hops delivers the update message faster than paths with more hops.

Data in Figure 4.8 shows the relationship between the longest AS-PATH length and the average convergence latency for a route withdrawal event. The vertical axis provides the average convergence latency and the horizontal axis provides the longest path observed during the process of BGP convergence. A direct line is included in the graph to better illustrate this relationship. Analysis of the convergence latencies shows that the average convergence latency for a route failure corresponds to the length of the longest possible backup path allowed by policy and topology between two peers. Although the data in Figure 4.8 contains significant variability, but it is observed that a linear relationship between the longest path and the average convergence latency exists. A probable explanation for the variability in Figure 4.8 is differences in routes processing and link delays.

In a complete graph of n nodes when these n BGP routers reach steady state, they all have 1-length path to X as their primary path. When route X is withdrawn, each node eliminates its primary path and chooses its best alternate path to X. Then each node announces 3-length paths to its peers. This process continues until nodes announce paths of length n and finally all nodes withdraw all the entries in their routing tables.

Convergence latency in case of route failure is dependent on the length of the longest path. The length of the longest path is linear with respect to the number of nodes in the network; therefore convergence latency is dependent on the number of BGP routers in the network.

4.5 Summary

In this chapter the volume of BGP routing updates and convergence latency triggered by each injection of a routing event has been examined. The impact of MinRouteAdvertisement timer on volume of routing updates has been explored. Also the relationships between network topology and convergence latency have been found.

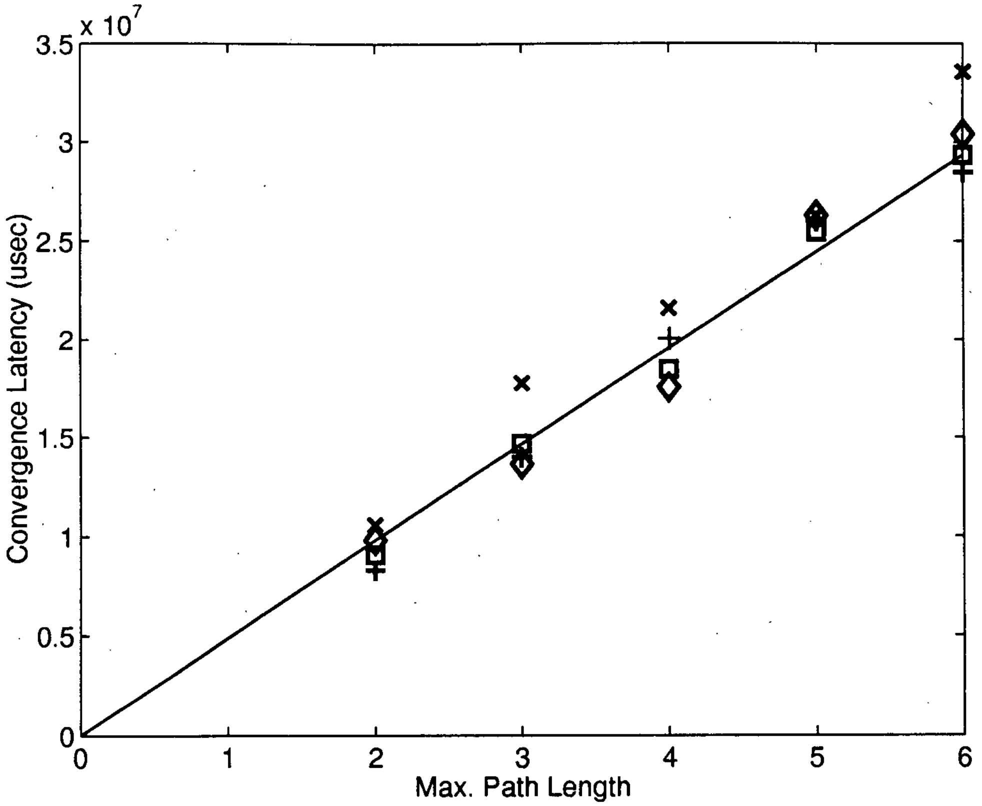


Figure 4.8: Relationship between convergence latency and max. path length

Chapter 5

Conclusions and Future Works

In this chapter a summary of the BGP protocol evaluation and analysis is provided, and also some tasks for optimizing the BGP protocol are suggested.

5.1 Conclusions

In this thesis, performance of the BGP4 protocol is evaluated. The BGP specification lacks behavioral and performance analysis. To clarify the BGP specification, the BGP convergence behavior under topological changes and routing instability is studied. In order to analyze the protocol functionality a test bed has been built which integrates six PCs and simulation software from the MRT project. To evaluate the capabilities of BGP4, a set of metrics is identified.

To analyze the BGP convergence behavior several routines are added to the source of simulation software. These routines implement:

- logging BGP routing update messages
- tracing BGP routing update messages
- loop detection on sender/receiver BGP routers
- MinRouteAdvertisementInterval timer

In this approach arbitrary network topologies have been modeled and simulated which lead us to the best and the worst-case model for BGP convergence. Our model neglected the impact of routing policies and intra-domain connectivity on the process of BGP convergence. The BGP specification [20] calls for AS-PATH loop detection, however it does not specify if detection should be performed by the sender or the receiver router. Labovitz *et al.* observations in [10] is based on receiver-side loop detection. In this project, BGP convergence behavior is analyzed by performing loop detection either on receiver side or sender side. It is found that performing MinRouteAdvertisementInterval timer and sender-side loop detection reduces the number of routing updates. Performing loop detection on the receiver-side significantly reduces the number of BGP routing updates.

In this project the lower bounds and upper bounds on volume of routing updates and impact of specific topological factors on convergence latency is explored. Labovitz *et al.* in [10] assumed that in the worst case for a route announcement, long link or processing delays can result in ordering of messages such that BGP would explore all possible paths of all possible lengths. It is shown in this project that such an ordering will not occur.

The analysis in this thesis demonstrated that for the event of a route announcement:

- Convergence latency is dependant on the maximum number of hops between paths in the Internet.

- The global lower bound on volume of routing updates is $2(n-1)$, where n is the number of autonomous systems. This is the lower bound for the best-case topology. The best-case topology is a linear connection between autonomous systems. The lower bound on volume of updates for the worst-case topology is $n(n-1)$. The worst-case topology is a fully connected graph of autonomous systems.

- The global upper bound on volume of routing updates is $(n-1)(2^{n-1})$. The volume of routing update messages grows exponentially with the addition of a new autonomous system.

This study showed that for the event of a route failure:

- Convergence latency will grow linearly with the addition of a new autonomous system.
- The global lower bound on volume of routing updates is $2(n-1)$.
- The global upper bound on volume of routing updates is approximately $(n-1)(n-1)!e$. Volume of routing update messages is factorial with respect to the number of autonomous systems in the Internet.

The adoption of path vector as a means of resolving the RIP routing table problem where a given node reuses information in a new path that the node itself originally initiated was explored. However path vector does not prevent a node from learning of a new and invalid path. It is observed that BGP still has bouncing problem contrary to the speculation expressed in [4].

5.2 Future Work

In our approach we have not considered the impact of IGP on BGP convergence. It is essential to consider the impact of routing policies and IBGP connectivity on the BGP convergence delay. Loop detection process was implemented either on sender router or receiver router. We suggest that loop detection process be implemented on both sender and receiver routers. It was demonstrated that BGP has bouncing problem. We suggest that each router upon receiving a withdrawal message for a given prefix, search its routing table and eliminates all entries for that prefix which include autonomous system number of the sender router in the AS-PATH attribute of those entries. If this optimization gets implemented on routers, the bouncing problem will be fixed and the BGP convergence behavior will improve.

References

[1] J. Garcia Aceves. Loop-free Routing Using Diffusing Computation. *IEEE/ACM Transactions on Networking*, pages 130–141, February 1993.

[2] Scott M. Ballew. *Managing IP Networks with Cisco Routers.* O'Reilly and Associates, 1997.

[3] Tim Bass. Internet Exterior Routing Protocol Development: Problems, Issues, and Misconceptions. *IEEE Network*, pages 50–55, 1997.

[4] D. Estrin, Y. Rekhter, and S. Hotz. A Scalable Inter-Domain Routing Architecture. In *Proceedings of ACM SIGCOMM Symposium on Communication Architectures and Protocols*, pages 40–52, August 1992.

[5] V. Fuller, T. Li, J. Yu, and K. Varadhan. Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation. *RFC 1519*, September 1993. Available at `http://www.ietf.org/rfc/rfc1519.txt`.

[6] Ramesh Govindan and Anoop Reddy. An analysis of internet inter-domain topology and route stability. In *Proceedings of IEEE INFOCOMM*, 1997. Available at `http://citeseer.nj.nec.com/govindan97analysis.html`.

[7] B. Halabi. *Internet Routing Architecure.* Cisco Press, 1997.

[8] John Hawkinson. Cisco Networking FAQ . Technical report, Cisco, May 1995. Available at `http://www.landfield.com/cisco-networking-faq/`.

[9] John W. Stewart III. *BGP4-Interdomain Routing in the Internet.* Addison Wesley Longman, 1999.

[10] Craig Labovitz, Abha Ahuja, Abhijit Bose, and Farnam Jahanian. Delayed Internet Routing Convergence . In *Proceedings of ACM SGCOMM*, pages 175–187, August 2000.

[11] Craig Labovitz, Robert Malan, and Farnam Jahanian. Internet routing instability. *IEEE/ACM TRANSANCTIONS*, pages 515–527, October 1998.

[12] Gary Scott Malkin. RIP Version2. *RFC 2453*, November 1998. Available at http://www.ietf.org/rfc/rfc2453.txt.

[13] D. Mills. Exterior Gateway Protocol Formal Specification. *RFC 904*, April 1984. Available at http://www.ietf.org/rfc/rfc0904.txt.

[14] University of Michigan and Merit Network. Multithreaded Routing Toolkit (M-RT) project. Available at http://www.merit.net.

[15] V. Paxson. Towards a framework for defining internet performance metrics. In *Proceedings of INET'96*, June 1996. Available at http://www.advanced.org.Almes/Inet96/Paxson/inet96-html.html.

[16] V. Paxson, G. Almes, J. Mahdavi, and M. Mathis. Framework for ip performance metrics. *RFC 2230*, May 1998. Available at http://www.ietf.org/rfc/rfc2230.txt.

[17] Cisco Press Publications. Routing Basics. December 1999. Available at http://www.cisco.com/cpress/cc/td/cpress/fund/ith2nd/it2405.htm.

[18] Jacob Rekhter. EGP and Policy Based Routing in the New NSFNET Backbone. *RFC 1092*, February 1989. Available at http://www.ietf.org/rfc/rfc1092.txt.

[19] Y. Rekhter and P. Gross. Application of the border gateway protocol in the internet. *RFC 1772*, March 1995. Available at http://www.ietf.org/rfc/rfc1772.txt.

[20] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP4). *RFC 1771*, March 1995. Available at http://www.ietf.org/rfc/rfc1771.txt.

[21] George C. Sackett and Christopher Y. Metz. *ATM and Multiprotocol Networking.* McGraw-Hill, 1997.

[22] Andrew S. Tanenbaum. *Computer Networks.* Prentice Hall PTR, 1997.

[23] P. Traina. BGP-4 Protocol Analysis. *RFC 1774*, March 1995. Available at http://www.ietf.org/rfc/rfc1774.txt.

[24] E. W. Zegura, K. L. Calvert, and Samarat Bhattacharjee. How to model internetwork. In *Proceedings of IEEE INFOCOMM*, September 1996. Available at http://citeseer.nj.nec.com/zegura96how.html.