Interdomain Routing for Mobile Nodes

Katie Schroth and Diane Kiwior
The MITRE Corporation
Bedford, MA

ABSTRACT

In this paper, we investigate the issues related to the use of a BGP (Border Gateway Protocol) backbone to provide connectivity between mobile nodes, with a specific focus on nodes within an airborne network domain. Research efforts have developed multiple MANET (Mobile Ad hoc Network) protocols to provide routing for mobile nodes. In an airborne network environment, however, there may not be a dense enough concentration of nodes within radio range to provide the connectivity needed for effective use of a MANET protocol. In addition, aircraft within radio range of other nodes will experience intermittent and varying quality radio signals due to banking, interference, or Doppler effects.

BGP is the de facto standard in use today to provide terrestrial internetworking routing among Autonomous Systems (AS) despite well known problems. BGP configuration can be complex and has convergence issues but the BGP capability to handle large numbers of routes makes it invaluable. In addition to its use in terrestrial internetworking, BGP has been identified as the routing protocol for the Transformational Satellite Communications System (TSAT) Network. Given the BGP networks in a satellite network above and a terrestrial network below an airborne network, it is important to understand the issues of connecting via BGP for airborne nodes.

This paper summarizes the results of lab experiments evaluating use of a BGP network for an alternate routing path between aircraft when there is no other connectivity within their airborne routing domain. Routing protocol overhead and convergence times are presented here along with an analysis of airborne nodes use of interdomain routing for connectivity.

1. INTRODUCTION

It is necessary to assume that an airborne network, as compared to a terrestrial network, will have frequent connectivity losses that will force it to use either a satellite network or a terrestrial network to maintain connectivity. Thus, it is important to study how fast an airborne network can regain connectivity through other Autonomous Systems, such as through satellite or terrestrial networks. The set of experiments discussed in the following pages focuses on airborne nodes and satellite nodes, where airborne nodes are located in one AS and the satellite portion of the network is another AS. The objective of the experimentation was to determine the convergence time and protocol overhead of the network if the airborne nodes lost connectivity and were forced to switch to a satellite link in order to regain connectivity. In these particular experiments, the airborne nodes are using the Open Shortest Path First (OSPF) protocol, while the satellite nodes are using BGP [1-2]. OSPF was used for the airborne nodes because it is one of the more popular terrestrial standard routing protocols; it is a link state routing protocol; and it is included on both the CISCO and Quagga routers (PCs running the Linux OS and the Quagga Routing Suite[3]) that were used in the experimentation. In reality, the routing protocol for an airborne network may be some type of MANET protocol that is specifically designed to provide routing for mobile nodes; candidates include the protocols being developed to support mobile ad hoc networking by extending OSPFv3 [4-5]. BGP, however, has been specifically called out as a preferred protocol for future military satellite constellation networks. As a result, these experiments aimed to determine how the BGP portions of the network would affect the overall convergence time and overhead given that the air-to-air link will quickly determine outages.

This paper presents a brief overview of the BGP protocol, which explains why the protocol is widely used as well as describes a number of shortcomings in regards to using BGP in a more dynamic environment. Section 3 describes in detail how an airborne and satellite network was emulated in a laboratory environment in order to measure network convergence times and overhead. Sections 4 and 5 present and explain the network convergence time and overhead results. Finally, conclusions are presented in Section 6.

2. BGP OVERVIEW

BGP is widely known to be a robust and scalable exterior routing protocol, which is used ubiquitously in

the Internet. BGP is a path vector protocol that allows Autonomous Systems to exchange routing information through messages sent over a TCP connection [6]. Although BGP is considered a robust exterior routing protocol, it is known to have long convergence times after routing changes [7-8]. Although BGP can have excessive convergence times, it is not generally considered a significant issue in terrestrial networks due to the fact that the links between Autonomous Systems are generally very stable. This may not be the case in a military environment where a single airborne node may be defined as an AS.

There are a number of reasons why BGP can have long convergence times. One reason for long times can be attributed to the Minimum Route Advertisement Interval (MRAI) [2], which is meant to limit the time between route advertisements in order to minimize the effects of route flapping, but in turn can create long convergence times [7]. Another reason for longer convergence times is that BGP has to deal with thousands of nodes on the Internet. In regards to using BGP in a satellite network, it is unclear how the severe propagation delay and variable processing/interleaving delay incurred by the space segment will affect the performance of BGP. Additionally, BGP relies on TCP for its operation, and TCP has known issues dealing with delay in a satellite channel. Thus, determining how BGP will affect the network's convergence time is vital in determining the overall network performance.

There are five timers associated with BGP which include the keepalive timer, hold timer, connect retry interval, minimum AS origination interval, and Minimum Route Advertisement Interval, mentioned above. The keepalive timer is the period between transmissions of keepalive packets, while the hold timer is maximum time allowed between messages before a link is declared down [2]. The connect retry interval is the time between a BGP neighbor's attempts to re-establish a connection [2]. The minimum AS origination interval is the minimum time between advertising changes within the router's AS, while the MRAI is the minimum time between successive route advertisements [2] The experiments being described assume that the satellite or ground network is stable. If the satellite or ground network is stable, then there is no reason to change the BGP timers. Also, by keeping the BGP timers constant while changing the airborne network protocol timers the variability of test results is limited. As a result, the default BGP timers are used in the satellite routers for every test. The default BGP timer settings are: keepalive timer = 30 seconds, hold timer = 90 seconds, connect retry interval = 120 seconds, minimum AS origination interval = 15 seconds, and MRAI = 30 seconds [2].

3. LABORATORY SETUP

The experimental setup described below is a standard configuration that is common to each test run that was conducted in the laboratory. Figure 3.1, shown below, is a basic diagram of the laboratory setup. It is assumed that the airborne nodes will not be affected by inter-beam or inter-satellite handovers.

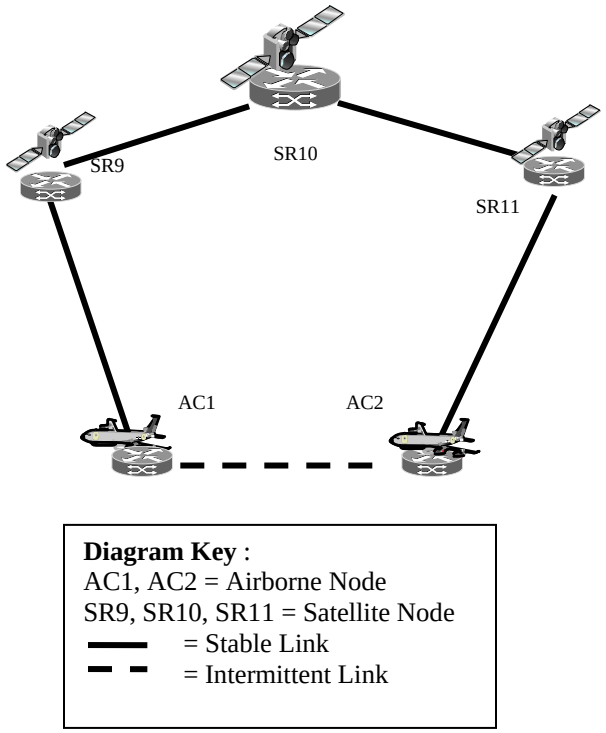

Figure 3.1: Airborne Network Laboratory Setup

The three routers, SR9, SR10, and SR11, shown in the upper half of Figure 3.1 represent satellite routers. The three satellite routers are configured as Autonomous Systems running exterior BGP (e-BGP). As previously mentioned, BGP is an exterior routing protocol that allows AS's to exchange routing information via messages sent over TCP connections. For some tests, the edge satellite routers, SR9 and SR11, utilized OSPF between themselves and the airborne nodes, while for other tests only e-BGP is run. AC1 and AC2 represent airborne nodes that run OSPF between each other and between the edge satellites for a series of tests and run e-BGP for another series of tests. There are a number of different laboratory setups that could have been used in the testing described herein. The

setup shown in Figure 3.1 was selected because it is a worst case scenario due the severe propagation delays found in satellite crosslinks.

Tables 3.1 and 3.2 identify the two configurations of routing protocols running on each link as shown in Figure 3.1.

Table 3.1 Configuration A

Link	Routing Protocol
AC1 – SR9	OSPF
SR9 – SR10	e-BGP
SR10 – SR11	e-BGP
SR11 – AC2	OSPF
AC2 – AC1	OSPF

Table 3.2 Configuration B

Link	Routing Protocol
AC1 – SR9	e-BGP
SR9 – SR10	e-BGP
SR10 – SR11	e-BGP
SR11 – AC2	e-BGP
AC2 – AC1	e-BGP

Because airborne links are inherently unstable it was necessary to break airborne link, AC2 – AC1, and then restore it for some arbitrary amount of time. This occurs through use of the MITRE-developed NetEmulator, a software package running on a PC, which is used to emulate the links between the routers [9]. Although not shown in Figure 3.1, there is an Etherenet connection between each pair of nodes to a PC that runs the NetEmulator in order to control and emulate link characteristics. A link scenario file was written to create link outages and recoveries at various times for the airborne link. This scenario file is a set of text-based sequences of updates that are read and executed by the NetEmulator during a test. Each line within the scenario file contains a time-stamp indicating when the command is to be executed and information defining the characteristics of a link, such as Bit Error Rate (BER), data rate, and delay. A link is broken by defining a bit error rate equal to 50%. The length of time a link is down is also defined in the Network Emulator scenario file. In general the airborne link was broken for a minute or more, and then restored for the same amount of time.

The BER of 10^{-8} was used as a reasonable estimate of an air-to-air link performance during periods of connectivity, and was derived from typical BERs found in current satellite systems. The delay between the satellites as well as the delay between the airborne nodes and satellites was determined by using approximate distances between the nodes. Table 3.3 identifies the BER and link delays used for each node.

Table 3.3: BER Characteristics and Link Delay

Link	BER	Delay (ms)
AC1 – SR9	10^{-8}	120
SR9 – SR10	10^{-8}	200
SR10 – SR11	10^{-8}	200
SR11 – AC2	10^{-8}	120
AC2 – AC1	10^{-8}	15

These laboratory experiments take a simple approach to calculating convergence time. A DOS Batch program was written to send a ping from the airborne node, AC1, to the other airborne node, AC2, every two seconds. AC1 was configured to log all Internet Control Message Protocol (ICMP) traffic as well as to log OSPF notifications. A system logger was used to capture all the logging information. The captured logging information was then exported into Excel where the convergence time was calculated by using the logged ICMP traffic. The log file shows ICMP replies being received and the receiving port. The log then notes that the link has been declared down. The next ICMP reply is received on another port because the router is now using the path through the satellite routers. Thus, the convergence time is the time it takes to receive an ICMP reply from the satellite path minus the time of the last received ICMP reply from the airborne link. It should be noted that this method of calculating convergence time is not exact. The pings are sent every two seconds, thus the actual convergence time could be up to two seconds less than the calculated convergence time depending on when the last ping was sent and when the link actually went down. Normally, convergence time would be defined as the time the link went down to the time it takes for all the routers to be updated with that information. Unfortunately, it is difficult to measure convergence time by the strict definition. The method of measuring convergence time as described above allows the network convergence time to be measured with an acceptable error of two seconds.

Ethereal, a packet capture program was used to measure the OSPF overhead during test runs. The overhead was calculated as a throughput.

Assuming that the airborne link is unstable while the satellite path is stable, it is reasonable to test how changing the protocol timer values on the intermittent airborne link may effect the convergence time of the overall network. Some of the OSPF parameters that can be manipulated within the router include: Hello interval, Router Dead interval, Shortest Path First (SPF) Delay, and SPF Hold time. The Hello interval is the period between transmissions of Hello packets [1]. The default Hello interval in the router is 10 seconds. Decreasing the Hello interval may aid in improving the convergence time of the network, but at the cost of increasing routing overhead. Conversely, increasing the Hello interval will reduce routing overhead, but the network will react more slowly to link outages. The Router Dead interval is the period that the router waits to hear a Hello from a connected neighbor before it declares the link is down [1]. The default Router Dead interval is 40 seconds. Decreasing the Router Dead interval will allow the network to react faster to link outages reducing packet loss. The main disadvantage of decreasing the Dead interval time is that if a link goes down for only a few seconds longer than the Dead interval, then the routers must send twice as many updates for a short outage period. The SPF Delay is the time between a link state update and the SPF calculation [1]. The SPF Hold time is the minimum time allowed between SPF calculations [1]. Table 3.4 contains the three set timer values that were used during test runs.

Table 3.4: OSPF Timer Values

Test Set	Hello Interval	Dead Interval	SPF Delay	SPF Hold
OSPF1	10	40	5	10
OSPF2	10	40	1	4
OSPF3	1	3	1	4

For the series of tests that used e-BGP on the airborne link, the Keepalive and the Hold Timers are analogous to the OSPF Hello and Dead Intervals. As a result, when e-BGP was run on the airborne link the Keepalive Timer and the Hold Timer mirrored the Hello Interval and the Dead Interval as shown in Table 3.5.

Table 3.5: e-BGP Timer Values

Test Set	Keepalive Timer	HoldTimer
BGP1	10	40
BGP2	1	3

In order to create a more realistic test network, a Smartbits 6000 was used to inject a large number of BGP routes into the network. The injected BGP routes create large routing tables, which aids in determining how the routing protocols perform in a real-world environment. Essentially, injecting BGP routes makes the test network look larger than it actually is and more representative of the current Forwarding Information Base (FIB) size for routing tables, which are approximately 175,000 unique IP prefixes. It is also likely that a Global Information Grid (GIG) will require similar routing table sizes.

4. CONVERGENCE TIME RESULTS

A number of different sets of tests were conducted using the laboratory setup described in Section 3. The first series of tests used Quagga routers running on Linux computers to measure the network convergence time of the three sets of OSPF timers defined in Table 3.4. There were sets of tests performed that used the BER and delays defined in Table 3.3, and sets of baseline tests that had no added BER or delay. The primary objective in conducting this series of tests was to determine how adding BER and delays impacted the network convergence time. Figure 4.1 illustrates the average convergence time for each trial for the OSPF timer values shown in the OSPF1 row of Table 3.4 with no BGP injected routes. Figure 4.1 clearly shows that on average, when BER and delay is included, there is an additional two seconds added to the network convergence time. Although it is not desirable to have an increased convergence time, an additional two seconds is not as severe as one might have expected given the severe propagation delays added.

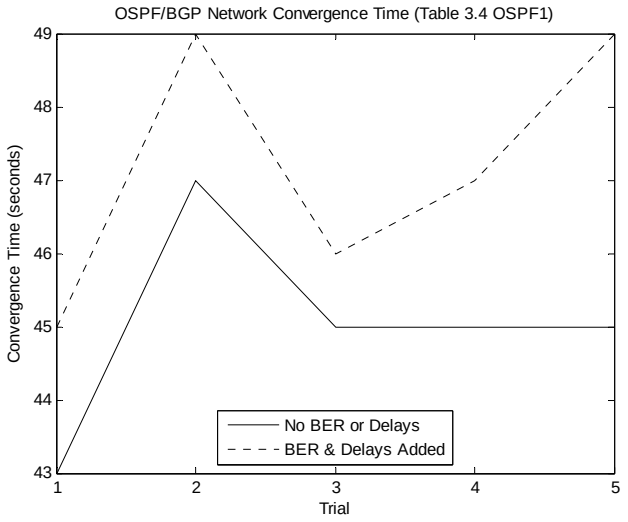

Figure 4.1 OSPF/BGP Convergence Time (OSPF1)

Figure 4.2 presents the average convergence time for each trial for the OSPF timer values shown in the OSPF2 row of Table 3.4 with no BGP injected routes. The timer values in OSPF2 decrease the SPF Delay and SPF Hold Time, which resulted in slightly faster convergence times on average by three seconds. Yet, Figure 4.2 is similar to Figure 4.1 in that the additional BER and delay only increased the convergence by two seconds on average.

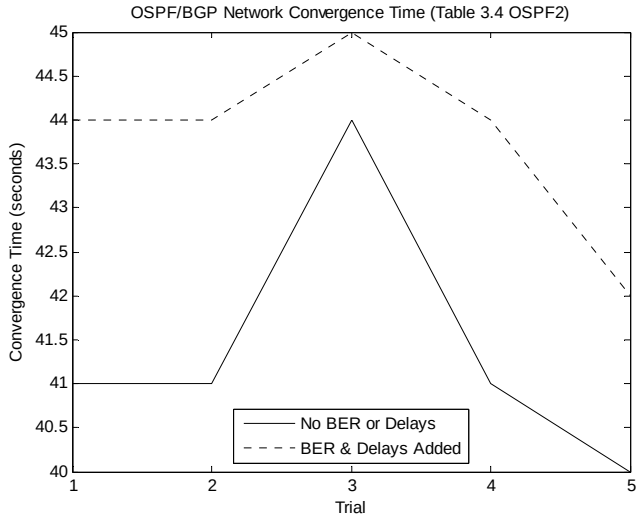

Figure 4.2 OSPF/BGP Convergence Time (OSPF2)

Figure 4.3 displays the average convergence time for each trial for the OSPF timer values shown in the OSPF3 row of Table 3.4. The timer values in OSPF3 decrease the Hello Interval, Dead Interval, SPF Delay and SPF Hold Time, which resulted in significantly faster convergence times of 34 seconds on average from the OSPF2 timers and 37 seconds on average from the OSPF1 timers. Yet, Figure 4.3 is also similar to Figures 4.1 and 4.2 in that the additional BER and delay only increased the convergence by three seconds on average.

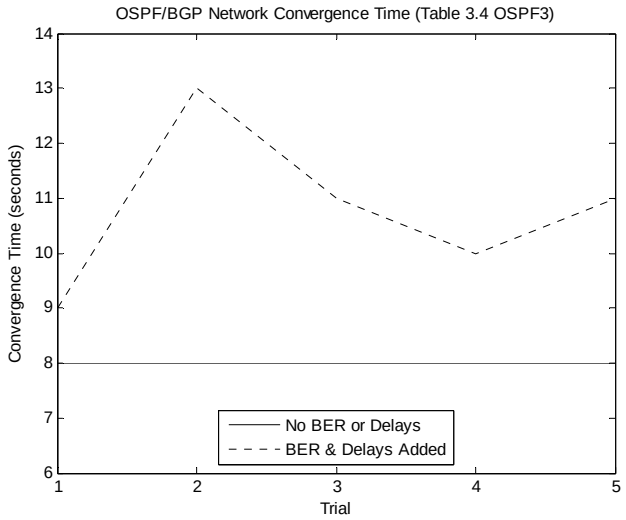

Figure 4.3 OSPF/BGP Convergence Time (OSPF3)

Because adding BER and delays did not have significant impact to the combined OSPF/BGP network convergence times, a second series of tests were conducted that used e-BGP on all the network links, including the intermittent airborne link. The primary objective in running e-BGP on all five nodes was to evaluate how well e-BGP would converge if each airborne node were declared to be an Autonomous System. Figure 4.4 shows the average convergence times for each trial for the protocol timer values shown in the OSPF1 row of Table 3.4 and the BGP1 row of Table 3.5. In this small network, there is no difference between the combined OSPF/BGP convergence times and the BGP only convergence times. On average, the network convergence was equal. It should be noted that a five node network is small and may not be yield realistic results. Therefore, a third series of tests was conducted in which a large number of BGP routes were injected to make the network look larger.

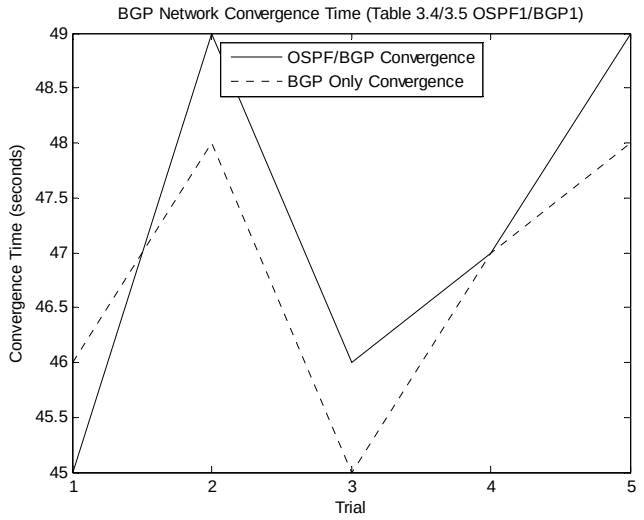

Figure 4.4 Convergence BGP only vs OSPF/BGP (OSPF1/BGP1 from Tables 3.4/3.5)

Figure 4.5 displays the average convergence times for the shortest protocol timers found in Tables 3.4 and 3.5. There is a significant difference between the convergence times for the BGP only network versus that of the combined OSPF/BGP network. The BGP only network, on average, converges 12 seconds slower than the combined OSPF/BGP network. It should be noted that the shorter protocol timers have smaller variances in convergence times per test run. For example, for the OSPF1 timers the convergence times vary from 40 seconds to 50 seconds for any given test run, whereas the OSPF3 timers only vary from 8 seconds to 10 seconds. As a result, it is more likely to have the average convergence time for OSPF3 timers be equal, which is the case in Figures 4.3 and 4.5 when there are no BER or delay.

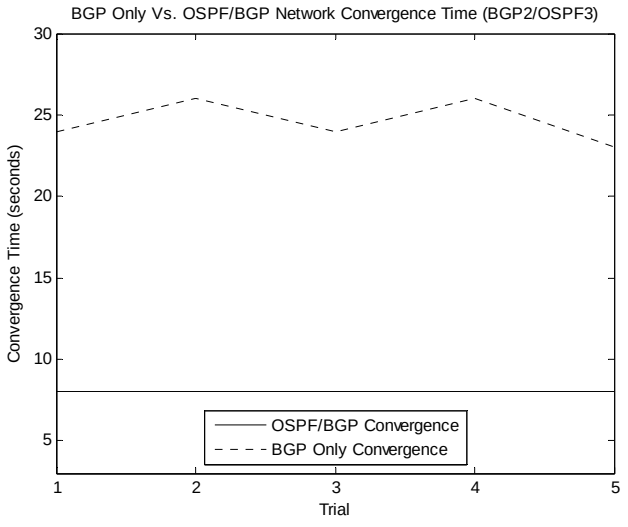

Figure 4.5 Convergence BGP Only vs OSPF/BGP (BGP2/OSPF3)

The first two series of tests use realistic BERs and delays, but the test setup was only a five node network. Test results may change significantly as the network grows larger. As a result, a third series of tests used the Quagga routers to measure the convergence times and overhead of the three sets of OSPF timers defined in Table 3.4, while injecting 100,000 BGP routes. Figure 4.6 presents the average convergence times of the test network with injected BGP routes versus no route injection for the OSPF1 and OSPF2 timers found in Table 3.4. For the protocol timers in OSPF1, the convergence time is generally four seconds slower when BGP routes are injected into the network versus when there is no route injection. For the slightly faster protocol timers in OSPF2, not only is the convergence time generally 14 seconds slower when BGP routes are injected into the network versus no route injection, but the convergence time is also significantly slower than it was for the slower OSPF1 timers. This result conflicts with the earlier presented results where faster protocol timers equated to faster convergence times. Making the routing tables large had a significant impact on the test results, which is why laboratory test setups need to be as realistic as possible to achieve legitimate results.

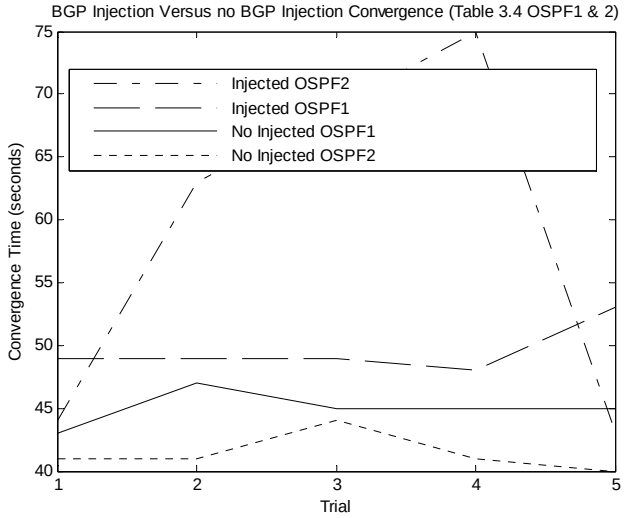

Figure 4.6 BGP Route Injection versus no Route Injection Convergence Times (OSPF1/OSPF2)

While conducting test runs using the OSPF3 timers found in Table 3.4 it was discovered that network was not converging. The captured data indicates that the network never realized that the airborne link was lost. Again, this outcome points out the fact that how the network is setup has a significant impact on test results. Therefore, it is vital to have as realistic a test setup as possible in order to have valid results.

5. ROUTING OVERHEAD RESULTS

To better understand the increase in convergence times it is necessary to study the routing overhead. Figure 5.1 displays the OSPF overhead for the protocol timers in Table 3.4. Figure 5.1 compares the overhead of the test runs that included injected BGP routes to that of the overhead for test runs that did not include injected routes. It should be noted that the figure looks as though the overhead is equal for all test runs that did not include injected BGP routes. In reality, the faster protocol timers in OSPF3 of Table 3.4 did have an increase in overhead throughput by a factor of three, but due to the scale of the figure this is not shown well. Figure 5.1 clearly shows when a large number of routes are injected into the test network the protocol overhead increases by factors on the order of hundreds as the protocol timers are decreased. The result is a longer convergence time as the protocol timer values decrease, which significantly differs from the results presented from the five node network with no injected BGP routes.

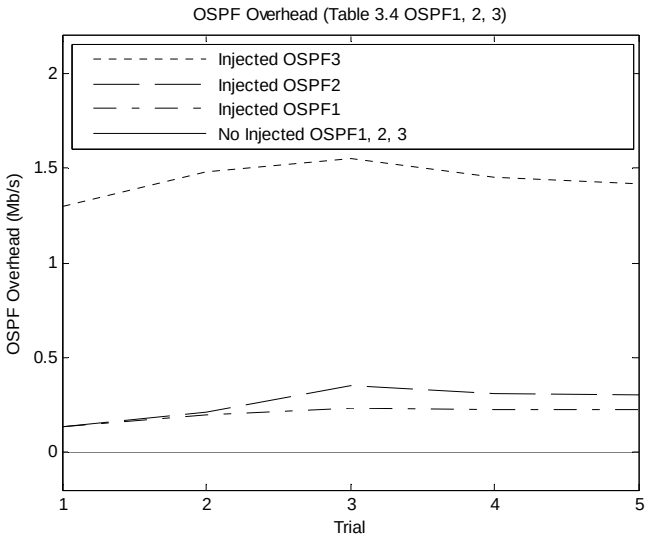

Figure 5.1 OSPF Overhead BGP Injected Routes Vs. No Injected Routes

In order to fully understand the significant increase in convergence time when large quantities of BGP routes are injected into the network, it is necessary to further study why the routing overhead significantly increases as BGP routes are being injected into the network. There are numbers of different types of routing overhead packets, each with a specific purpose. When BGP routes are injected into the network, the results indicate that there is a significant increase in the OSPF Link State Advertisement (LSA) Type 5 packets. LSA Type 5 packets are generated by Autonomous System Border Routers, and are used to redistribute routes into OSPF [1]. Figure 5.2 shows the percentage of overhead packets that are LSA Type 5 packets that were captured during test runs with no routes injected versus with routes injected. Figure 5.2 indicates that the increase in overhead that occurs due to BGP route injection is caused by the LSA Type 5 packets.

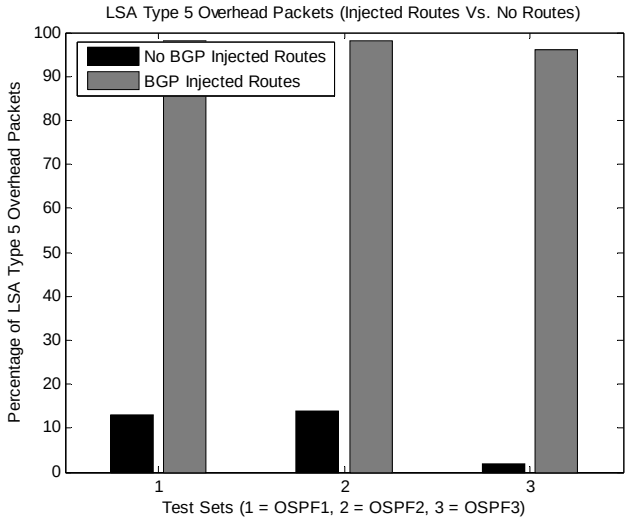

Figure 5.2 Percentage of LSA Type Overhead Packets (Injected Routes Vs. No Routes)

6. CONCLUSIONS

BGP is the de facto standard in use today to provide terrestrial internetworking routing among Autonomous Systems despite well known problems. BGP configuration can be complex and has convergence issues but the BGP capability to handle large numbers of routes makes it invaluable. In addition to its use in terrestrial internetworking, BGP has been identified as the routing protocol for future military satellite networks. Given the BGP networks in a satellite network above and a terrestrial network below an airborne network, it is important to understand the issues of connecting via BGP for airborne nodes.

The sets of experimental results discussed in Sections 4 and 5 focused on airborne nodes and satellite nodes, where the satellite portion of the network was another AS. The objective of the experimentation was to determine the convergence time and protocol overhead of the network if the airborne nodes lost connectivity and were forced to switch to a satellite link in order to regain network connectivity.

The initial experimentation indicated that decreasing the protocol timer values would also decrease the network convergence time, which is desirable. Unfortunately,

these results change significantly as the size of the network increases. When a large number of BGP routes are injected into the network, the results indicate that decreasing the protocol timer values significantly increases the protocol overhead, which in turn actually causes longer convergence times. These results point out how vital it is to accurately emulate or simulate a network environment in order to achieve realistic results.

The results indicate that decreasing protocol timers in a large network will actually degrade the network performance by increasing both the protocol overhead and the network convergence time. Because airborne links are more susceptible to link outages, it is desirable to have faster convergence times, but the results indicate that achieving faster convergence times is not a simple issue to resolve. Additional laboratory testing is necessary to find a more appropriate solution to interdomain routing for an airborne network. One possible solution may be using Virtual Private Networks (VPNs) over the satellite network rather than relying on a typical routing protocol, as suggested in [10]. A logical next step in testing is to use a simulation tool to test possible VPN solutions using a similar network setup as shown in Figure 3.1 that includes large routing tables. The objective would be to measure convergence time and protocol overhead of VPN solutions for comparison with the results presented in Sections 4 and 5.

7. SOURCES

[1] J. Moy, "OSPF Version 2," RFC 2328, April 1998.

[2] Y. Rekhter, T. Li, "Border Gateway Protocol," RFC1771, March 1995.

[3] Quagga Software Routing Suite. Site: http://quagga.net

[4] M. Chandra, A. Roy, "Extensions to OSPF to Support Mobile Ad Hoc Networking," Internet-Draft draft-chandra-ospf-manet-ext, January 2007.

[5] R. Ogier, "MANET Extension of OSPF using CDS Flooding," Internet-Draft draft-ogier-manet-ospf-extension, October 2006.

[6] W. Stallings, High Speed Networks and Internets Performance and Quality of Service, Prentice Hall 2002. Pages 444-445.

[7] T. Griffin, B. Premore, "An Experimental Analysis of BGP Convergence Time," Proceedings of ICNP, 2001.

[8] D. Pei, X. Zhao, L. Wang, D. Massey, A. Mankin, S. Wu, L. Zhang, "Improving BGP Convergence Through Consistency Assertions," in Proceedings IEEE INFOCOM, 2002.

[9] Open Channel Foundation: NetEmulator, 2005. Site: http://www.openchannelfoundation.org/projects/netemulator

[10] E. Ertekin, C. Christou, "Applying 4364 Virtual Private Networks to the Global Information Grid," MILCOM 2006.