TECHNIQUES FOR ENABLING DYNAMIC ROUTING ON AIRBORNE PLATFORMS

Jared Burdin, Jeffrey D'Amelia, Elizabeth Idhaw, and Jack Shaio

The MITRE Corporation
202 Burlington Road
Bedford, MA 01730

ABSTRACT

The Department of Defense is developing a global information grid (GIG) that will fully interconnect warfighters, policy makers, and support personnel. To provide the end-to-end connectivity envisioned by the GIG, an airborne networking layer is required. Current technologies must be evolved to develop this capability.

An evolution of sorts has already begun, utilizing proven protocols and existing commercial routers. Commercial routers have been used for testing and experimentation at live-fly events such as Joint Expeditionary Force Experiment (JEFX) and Empire Challenge (EC). Until recently, static routes have been used to enable IP routing between airborne platforms during these exercises. However, static routes are a less than ideal solution for airborne networking, because the routes cannot be adjusted in response to topology changes caused by platform mobility. Consequently, experiments utilizing dynamic routing have been conducted at recent live-fly events.

This paper explores approaches to dynamic routing used at recent live-fly experiments including protocol configuration optimizations and internetworking with routing enabled terminals. It reports on performance measurements obtained through lab testing and at live-fly exercises and presents proposed enhancements, including load balancing techniques, for future dynamic routing capabilities.

1. INTRODUCTION

Air Force (AF) operations are transitioning to a network-centric model that provides seamlessly internetworked systems like those found on the Internet. To facilitate this transition, there is a desire to leverage proven networking standards like the Internet Protocol (IP) [1] to provide connectivity to tactical edge platforms such as AWACS and JSTARS. In the desired architecture, airborne nodes will be connected via IP networks and dynamic routing will be utilized to determine the appropriate path for data flows to traverse. However, the airborne environment presents challenges and requirements to dynamic routing that are not typical of a fixed infrastructure network. Our research addresses some of these problems and demonstrates near term solutions to them using commercial-off-the-shelf (COTS) routers.

Prior to our demonstrations, only static routes for IP networking were used to connect airborne platforms to each other and the ground network at both Joint Expeditionary Force Experiment (JEFX) and Empire Challenge (EC) live-fly exercises. At EC08, the Open Shortest Path First (OSPF) [2] interior gateway routing protocol was used to enable dynamic routing for an airborne network that included both line-of-sight (LOS) and beyond line-of-sight (BLOS) links. Lessons learned from EC08 were then used to deploy a dynamically routed airborne network at JEFX in 2009. In this paper, we discuss the architectures used at these two exercises, show how dynamic routing worked within these architectures, and offer further enhancements that could improve performance for future airborne networks. In addition, we present the results of laboratory testing and compare it with those obtained from both EC08 and JEFX09 to illustrate the improvements from our work.

2. DYNAMIC ROUTING FOR AIRBORNE PLATFORMS WITH OSPF AND COMMERCIAL-OFF-THE-SHELF ROUTERS

Dynamic routing is a necessity for the successful operation of future airborne networks. The ability to react to degrading link metrics and changing topologies in a dynamic fashion allows for minimal disruption in a quickly changing environment and can provide a marked improvement over the statically routed architectures employed in previous exercise networks. This improvement is two-fold: limiting the amount of preplanning necessary to introduce IP routing and allowing the routing topology to dynamically handle unforeseen situations. When attempting to gain this improvement by implementing dynamic routing in airborne networks, however, one is constrained by the terminals provided on the airborne platforms and the planned network architecture.

A. DYNAMIC ROUTING AT EMPIRE CHALLENGE 2008

The primary goal of the dynamic routing portion of the LOS/BLOS initiative of Empire Challenge 2008 was to demonstrate the advantages of using a dynamic routing

protocol across a black side airborne network comprised of both line-of-sight (LOS) and beyond-line-of-sight (BLOS) links. Commercial-off-the-shelf (COTS) networking equipment was used in conjunction with LOS and BLOS radio terminals on each node. The network architecture consisted of an airborne component made up of three nodes as well as three ground entry nodes with reachback to the terrestrial CAOC network as depicted in Figure 1 below. Dynamic routing was only utilized on the black side. Red side routers used a static routing topology.

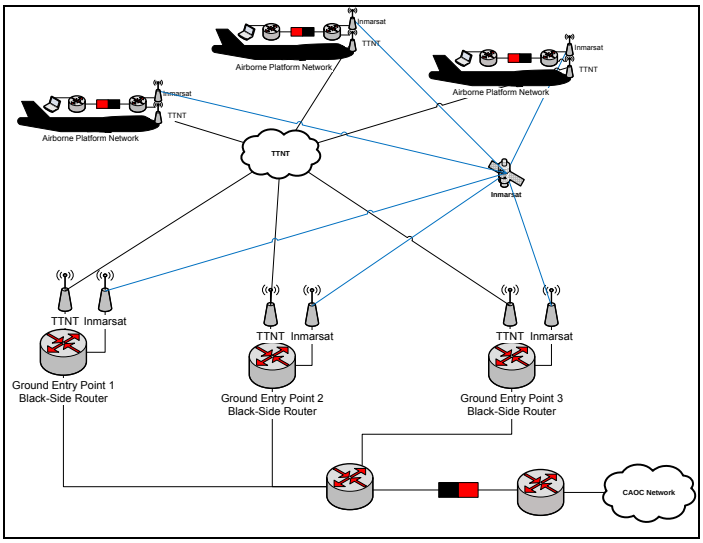

Figure 1 Notional network diagram for LOS/BLOS initiative of Empire Challenge 2008

The black side routing infrastructure consisted of one COTS router (Cisco 3800 series) per platform running the Open Shortest Path First (OSPF) dynamic routing protocol. Each black side router was connected to one or more other black side routers over different types of terminals such as the Rockwell-Collins Tactical Targeting Network Technology (TTNT) terminal [3], Inmarsat Swift-64 terminal [4], and Inmarsat Broadband Global Area Network (BGAN) terminal [5]. In addition, each router was connected to a red side enclave through a High Assurance IP Encryption (HAIPE) [6] device.

Dynamic Routing Design Considerations

In setting up the routing topology for Empire Challenge 2008, there were several considerations that needed to be taken into account in order to achieve optimal performance from the IP network. There were two main criteria for designing the network; the first objective was to ensure that OSPF worked correctly over all types of links, including but not limited to TTNT, Inmarsat BGAN, and Inmarsat Swift-64. The second objective was to ensure that the settings used for OSPF configuration, namely the hello-interval and dead-interval, were configured in such a way as to obtain acceptable responsiveness while ensuring as little excess control overhead as possible.

Routing Topology Considerations

The first challenge to achieving these goals was finding a way for OSPF to operate correctly over each of the different types of links. Because Inmarsat Swift-64 links provide direct ISDN connectivity between endpoints, OSPF ran directly over these links without any problems. However, this was not the case for the Inmarsat BGAN or TTNT terminals. In the case of Inmarsat BGAN, the connection does not terminate locally, but is instead sent over the Internet to its final destination, preventing OSPF from running over this link. In the case of TTNT, while the terminals can form a network amongst themselves using a proprietary routing protocol, they do not provide support for internetworking with baseband equipment. This inability to act as a gateway between their mobile routing protocol and a standard routing protocol like OSPF prevents each network from learning about the other. Additionally, TTNT's inability to correctly deal with multicast traffic, limiting it to a single TTNT hop, prevents OSPF from correctly traversing the TTNT network. Because of these things, Generic Routing Encapsulation (GRE) [7] tunnels were established between black side routers for each of these link types to allow OSPF to correctly operate over these terminals.

By utilizing the GRE tunnels, we were able to exploit the terminals' unicast capabilities. By tunneling routing protocol traffic to unicast destinations, routing adjacencies can be formed and topological information exchanged. The drawback with this approach is the additional overhead associated with sending duplicate unicast copies of OSPF messages instead of a single multicast version of the same message. This added overhead can be minimized and effectively balanced against the benefit of improved network performance by optimizing the protocol configurations to provide the best compromise between performance and excess overhead.

OSPF Configuration Optimizations

It is essential, considering the relatively small amount of bandwidth available on the wireless links in the network, to investigate the overhead that the tunneling and routing protocols impose on an airborne network and to configure the protocols to minimize overhead while maximizing responsiveness to topology changes. To determine the optimal protocol settings, we developed an airborne networking test bed, which emulated the Empire Challenge LOS/BLOS network shown in Figure 1, over which we ran GRE tunnels and the OSPF routing protocol.

The OSPF protocol's overhead consists of Hello Message, LS Acknowledgement, LS Request, LS Update, and Database Description packets, which detect the presence of remote routers and distribute routing topology information throughout the network. The frequency at which these messages are sent relates both to protocol configuration settings (balancing responsiveness and overhead), and the frequency with which the network topology changes (motivating the distribution of new network state information). Additionally, the GRE protocol's overhead consists of keep-alive messages, which are sent over each tunnel at a specified frequency to detect whether the underlying network is functional, and encapsulation packaging, which consists of approximately 20 additional header bytes added to each data packet traversing the tunnel.

To determine which settings provide the best compromise between additional overhead burden and network convergence time, which is defined as the amount of time it takes for a route change to be propagated to all routers in the network [8], laboratory testing was performed using a variety of different parameter settings for both GRE and OSPF. For each group of settings, all network traffic was collected and analyzed to determine the rate at which overhead was introduced into the network. During each trial, a link failure that caused a routing topology change was induced on the network and the consequent convergence time was measured. Figure 2 illustrates the results of the testing, in which the bars indicate the overhead and the line denotes the convergence time.

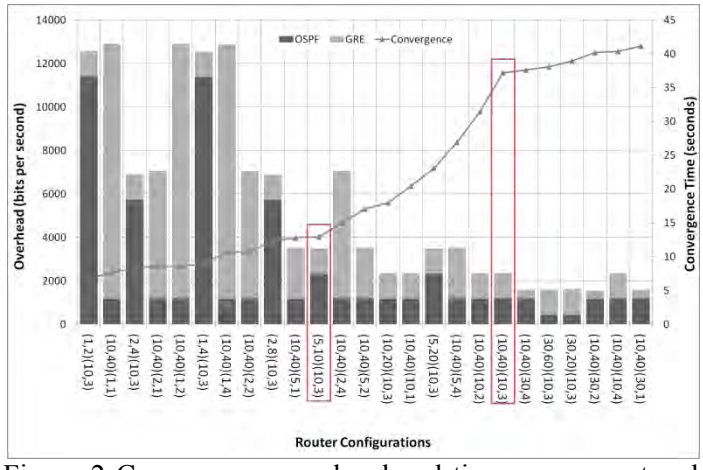

Figure 2 Convergence overhead and time versus protocol configuration, in which the tick marks represent OSPF Hello Interval, OSPF Dead Interval, GRE Keep-alive Interval, and GRE Retry Count from top to bottom, from laboratory experimentation

As shown in Figure 2, protocol parameter settings exert a strong influence on the resulting overhead and convergence time. Faster OSPF and GRE polling intervals result in greater amounts of overhead but shorter convergence times; slower polling intervals result in longer convergence times but a significant decrease in overall overhead. As would be expected with our small topology, our results show that the convergence time is approximately equal to either the OSPF dead interval or the total time for the GRE retries, whichever is shorter. In a larger network, the convergence time would be the amount of time to detect a link failure plus the time to propagate that change throughout the network.

From our observation of the data, the best compromise between overhead and convergence time is utilizing the protocol settings shown in Table 1 which are also the leftmost highlighted configuration set in Figure 2. These settings result in approximately 4 kbps of overhead and approximately 13 seconds for convergence time. The configuration settings in Table 1 were selected for use in the Empire Challenge 2008 (EC08) LOS/BLOS experiments.

Table 1 Optimized OSPF & GRE Protocol Settings

Protocol	**Variable**	**Value**
OSPF	Hello Interval	5 seconds
	Dead Interval	10 seconds
GRE	Keep-alive Interval	10 seconds
	Retry Count	3 retries

Table 2 Cisco IOS default OSPF & GRE Protocol Settings

Protocol	**Variable**	**Value**
OSPF	Hello Interval	10 seconds
	Dead Interval	40 seconds
GRE	Keep-alive Interval	10 seconds
	Retry Count	3 retries

EC08 Experimental Results

The benefit provided by the OSPF and GRE optimizations was measured during live-fly experiments at Empire Challenge 2008. The results from our lab experimentation were confirmed during the live-fly. A link outage was generated by turning off the transmit switch on the TTNT terminal at one of the ground entry points. The black side router on the airborne platform using that particular ground entry point detected the link outage due to the lack of OSPF hello messages from the ground entry point. Once detected the effected airborne platform's black side router

would re-route to one of the two remaining ground entry points. Figure 3 below shows the average convergence time measured for a TTNT link outage using both Cisco IOS default settings [9], shown in Table 2, and our optimized settings. When Cisco default settings were used on the black side routers the effected airborne platform's router would converge to the other ground entry point in an average of 36.6 seconds; when our optimized OSPF settings were used the average convergence time was 11.8 seconds.

The live-fly test results are remarkably close to the lab test results, demonstrating the value of pre-exercise lab testing. Additionally, the results illuminate the importance of optimizing OSPF for an airborne platform. The optimized parameter settings effected an improvement in the convergence time by nearly 25 seconds. It is expected that a tactical network implementing the parameter optimizations will experience improved application performance because the network will adjust quickly to the frequent link outages experienced with airborne nodes.

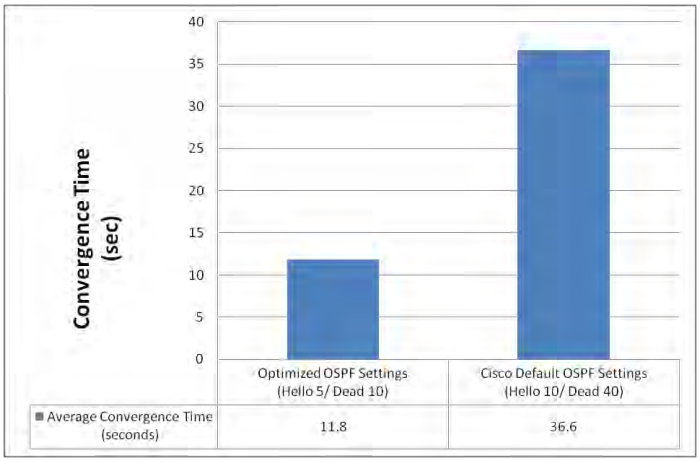

Figure 3 Average convergence time (seconds) when a linked failed at Empire Challenge 2008 using default and optimized OSPF settings.

B. DYNAMIC ROUTING AT JEFX 2009

The primary goal of the dynamic routing portion of JEFX09 was to allow for dynamic node entry and exit, compensate for link outages, reduce black side pre-configuration, and offer dynamic selection of ground entry points. This was in contrast to JEFX08, at which static route entries were used network wide. What differentiated the dynamic routing trials at JEFX09 from those at EC08 was the elimination of GRE tunnels to support the multicast traffic of OSPF. GRE tunneling was not feasible at JEFX09 because the architecture did not support it on all nodes.

The black side routing infrastructure was similar to the Empire Challenge 2008 black side network shown in Figure 1. However there were a couple of key differences, one was JEFX09 did not have a BLOS data link and another difference was most of the platforms did not have a black side router. The nodes without a black side router directly connected their TTNT to their HAIPE device, where as nodes with a router connected their TTNT to the router and then to the HAIPE device. The absence of black side routers on some nodes was one of the reasons, in addition to the scalability problems, why GRE tunnels were not a reasonable solution for the JEFX09 architecture. Open Shortest Path First (OSPF) was also run on the COTS black side routers that some platforms included as part of their on-board architecture. In addition, each node utilized a TTNT radio terminal, on which the OSPF dynamic routing protocol was run.

Running OSPF on the TTNT Terminals

Enabling OSPF on TTNT allows the TTNT terminal to form adjacencies with baseband side routers and to learn about the larger operational topology. One drawback to this approach is that it requires running OSPF over TTNT's RF interface. This is inefficient because OSPF is running over the air in addition to TTNT's own proprietary routing protocol. Because the terminal cannot redistribute its routes out the baseband connection, there is no way to incorporate the terminals into the routing topology without this inefficiency.

In order to run OSPF on the TTNT terminals, the open source GNU Quagga [10] implementation, which includes Zebra [11], an open source OSPF daemon, was cross compiled to run on each TTNT terminal. By doing this, each route learned by the OSPF daemon was inserted in the TTNT's routing table, allowing the terminals to have knowledge of the subnets that resided behind each radio. This replaced the approach taken at previous JEFX exercises where routes where statically added to the terminals' route tables.

An important part of the OSPF configuration on TTNT is the usage of the point-to-multipoint network type. Typically, OSPF would use a broadcast network, especially when the network uses a broadcast link layer such as Ethernet. With an OSPF broadcast network type, two nodes in the network are elected to be the designated router (DR) and backup designated router (BDR) [2]. Their role in the network is to flood link state update messages to everyone via multicast when the routing topology changes. However, TTNT only supports one-hop multicast preventing some TTNT nodes from receiving link state update messages when a broadcast network type is used. By using a point-to-multipoint network type, there

is no longer a DR and BDR in the OSPF network and each node sends out link state update messages. The advantage is that every TTNT radio will receive updates despite the cost of additional network overhead.

JEFX09 Results

Figure 4 shows a graph of the traffic associated with OSPF on one of the ground entry points and all mobile nodes within range of it during one day of testing at JEFX09. The OSPF Hello packets represent a relatively small, steady rate of traffic, while the link-state updates and database transfers create traffic spikes depending on when nodes enter or exit the OSPF network. While spikes of OSPF traffic exceeded 50 kbps, the overall traffic associated with OSPF averaged across the live fly period was ~400 bps, this is an improvement from EC08, which saw OSPF overhead average ~4 kbps, because of the GRE tunnels.

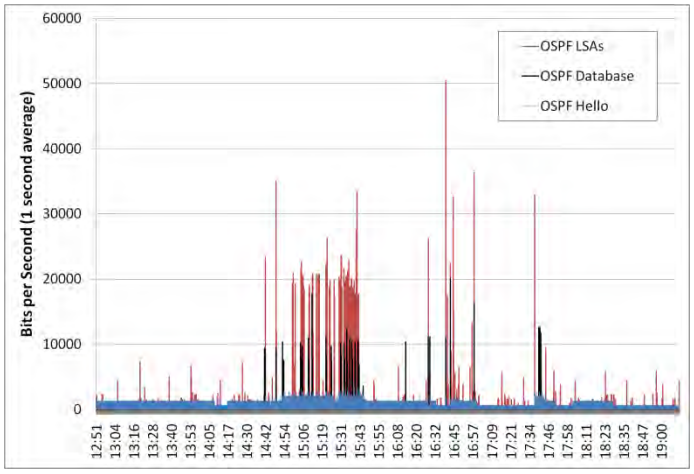

Figure 4 OSPF overhead measured at a ground entry point at JEXF09

We did not measure convergence time at JEFX09, as we did at EC08, due to differences (no BLOS link or black side router) in network architecture and the overlap of OSPF with TTNT's mobile routing protocol. Ultimately, the tests would have only measured the convergence time of TTNT's mobile routing protocol and would not have been comparable to the Empire Challenge results.

C. FUTURE DYNAMIC ROUTING ENHANCEMENTS

In addition to our dynamic routing work at both EC08 and JEFX09, there is still more that should be done to enhance dynamic routing for airborne platforms. First, routing enabled terminals, like TTNT, need to have a mechanism to redistribute routes in order to eliminate tunnels from the network architecture. Second, a load balancing capability is needed to distribute traffic over disparate links and multiple ground entry points.

Route Redistribution

Ideally IP networks would run a single routing protocol; however that is not always possible. In those cases, when you have multiple routing protocols in the same domain, route redistribution is used to exchange the external route information [9]. This is the case when using TTNT in an airborne network. TTNT uses its own mobile routing protocol to route data between TTNT nodes, and the ground network uses a standard routing protocol, like OSPF, to route data through its network. Therefore the TTNT network and ground network each have their own routing topology and do not know about each other.

At JEFX09 we enabled OSPF on the baseband and RF side of TTNT, thus effectively using a single routing protocol for the entire exercise network at JEFX09, and eliminating the need to distribute routes between OSPF and TTNT's mobile routing protocol. However, TTNT's mobile routing protocol was still running and it would be more efficient to utilize it for the RF network versus OSPF.

While running OSPF on the TTNT terminals proved successful, a future enhancement to this architecture would be to redistribute TTNTs internal routes into OSPF and for TTNT to redistribute OSPF routes into its mobile routing table. Besides TTNT providing a route redistribution capability with OSPF, it is important that emerging routing enabled terminals also implement a route redistribution capability, such that its own routing protocol can exchange routes with a standards based routing protocol. This would make routing in large heterogeneous airborne networks more feasible while reducing overhead even further compared to the approaches taken at both Empire Challenge 2008 and JEFX09.

Load Balancing

At JEFX09 we were unable to select which ground entry point the airborne platform would use to get to the ground network. In past JEFXs when static routes were used, the network administrators simply set the default gateway on the airborne platform to use one of the ground entry points and made sure to balance out the platforms across all the available ground entry points. With dynamic routing the airborne platform is going to route data to the ground network via the shortest path and what typically happened was one ground entry point was the shortest path for all airborne platforms, thus creating a congestion point in the network. In addition to the desire to distribute traffic over multiple ground entry points, as was the case with the JEFX09 network, it is also desirable to distribute traffic over multiple links if available, which was the case with the EC08 network. At EC08 both LOS and BLOS links

were available on the airborne platforms, but during the exercise only the link that was part of the shortest path could be used, as a result the other links on the platform were underutilized. In order to improve link utilization, we developed a load balancing technique for airborne platforms. Our load balancing technique focuses on distributing traffic across multiple links, but it could also be adapted to distribute traffic across ground entry points.

3. LOAD BALANCING FOR AIRBORNE PLATFORMS

In addition to studying the tradeoffs of particular OSPF protocol settings and how to best optimize these settings to perform in the exercise architectures, we explored ways of load balancing over multiple disparate wireless links. While the COTS routers natively supported load balancing with OSPF over equal cost links, we present a technique for load balancing across a collection of non-equal cost links, such as TTNT (LOS) and Inmarsat (BLOS). This study was conducted in our lab environment and was not tested at any live-fly exercises. However, we feel that utilizing multiple links to load balance traffic in a bandwidth deficient environment will improve network performance.

Equal cost multipath (ECMP) load balancing allows a router to balance the traffic load among multiple routes of equal cost; however it is rare for an airborne platform to offer equal cost paths due to the disparate link types typically available on platforms, which offer varying link capacities and delays. Therefore to improve dynamic routing on airborne platforms we researched ways to load balance traffic over disparate link types using COTS routers. Then, we created a non-equal cost multipath routing capability that will integrate with COTS routers thus providing load balancing and improved bandwidth utilization.

Our non-equal cost multipath (Non-ECMP) routing consists of two commonly available COTS router features. The first is a rate-limit policy [12], which looks at the rate of arriving traffic and sets the DiffServ Code Point (DSCP) [13] field to indicate if the traffic conforms to or exceeds the specified rate. Conforming traffic would be marked "green", while traffic exceeding the rate-limit would be marked "yellow" or "red". The traffic markings applied by the rate-limit could then be processed by the second feature, policy-based routing [14]. Policy-based routing utilizes packet classifications instead of the standard destination address to determine the packet's next-hop. The policy routes send traffic to a *recursive* next-hop; in other words, the router forwards the IP packet to the IP next-hop for the route to the recursive next-hop (which might be many IP hops away). This makes the policy routes more robust to link failures and route changes: as long as the router has a route to the recursive next-hop, it will be able to send the packet towards it. If there is no route, the policy route is ignored and the packet follows the normal IP routed path.

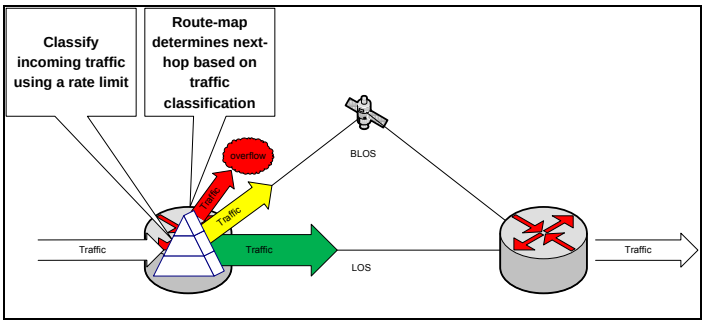

Figure 5 Non-Equal Cost Multipath Routing Architecture

Figure 5, shows a graphical representation of our Non-ECMP routing architecture. In addition to developing the Non-ECMP architecture, we also tested the architecture to verify the throughput gains and to understand its impact on TCP [15].

Implementation

The Non-ECMP routing capability was implemented using Cisco 2800 routers running IOS version 12.4. All the policy routing on an incoming interface for these tests is selected by the DSCP value alone. The configuration resulted in the behaviors identified in Table 3. The DSCP values used were just for the purposes of our proof-of-concept testing. Other values could be substituted and still result in the same behavior.

Table 3 DSCP values and routing behavior

DSCP value	Behavior
AF22	Packet is processed by rate-limit, which may change the DSCP value depending on the outcome
CS2	Selects policy route 1; this DSCP is set by the rate limit if the AF22 traffic is between the rate limit's sustained rate (cir) and the maximum rate (pir) configured values.
CS6	Selects policy route 2; this DSCP value is set by the rate limit if the AF22 traffic is above the maximum rate (pir) configured in the rate limit.
CS7	Any incoming packet with this DSCP is sent on policy route 1; its DSCP is left unchanged
AF21	Any incoming packet with this DSCP is sent on policy route 2; its DSCP is left unchanged.
All other DSCP values	Packet follows the IP routed path

This configuration allows both dynamic packet-based and static flow-based load balancing to be tested. For packet-based testing, all packets are sent with DSCP=AF22. The packets leave the rate limit with one of three possible

DSCP values, depending on the traffic load seen by the rate limit: AF22, CS2, or CS6. Table 3 describes the routing behavior for each value.

For flow-based load balancing, the sender decides whether to use DSCP=CS7 (for policy route 1) or DSCP=AF21 (for policy route 2); any other DSCP (except AF22, CS2, CS6) sends the packets on the IP routed path, skipping policy routing. Note that the sender could also use DSCP=CS2 or CS6 to select policy routes, although these tests set aside those two values for the policy routes chosen by the rate-limit and used the different values AF21/CS7 to select those same routes statically.

A single configuration allowed both packet and flow based load balancing to be tested on the same interface. This requires, however, that the DSCP markings applied by the rate limit be applied to the packet before the DSCP field is examined to select the policy route. The specific Cisco router used does not allow serializing these two operations; instead the rate-limit markings are applied in parallel with other packet processing, so that policy routing sees only the DSCP value originally on the packet, before the rate limit changed it.

These tests achieved serialized DSCP marking and policy routing of packets, first by the rate limit and then by policy routing. This was done by re-circulating packets twice through the router; the first time the rate-limit applies its DSCP changes (if any) and the packets are forwarded out an interface that loops back to the same router. The second time the packets are processed by policy routing and carry any DSCP changes that were applied to them in their first pass through the router.

Normally this approach would lead to degradation in forwarding performance because each packet requires two IP lookups, one IP lookup on each pass through the router, but in the airborne network context the bottleneck imposed by the limited radio link capacity is so great that the forwarding inefficiency of a second pass through the router and two IP lookups instead of one is not noticeable. All the tests run with this configuration were able to achieve the maximum capacity provided by the radio links, showing that throughput was not impacted by passing twice through the router.

Performance Measurements

As stated earlier, we developed a non-equal cost multipath (Non-ECMP) routing capability for COTS routers to increase network throughput by improving bandwidth utilization. Further, we tested our Non-ECMP implementation to verify and measure the additional throughput it provides. Figure 6, shows the results of these tests.

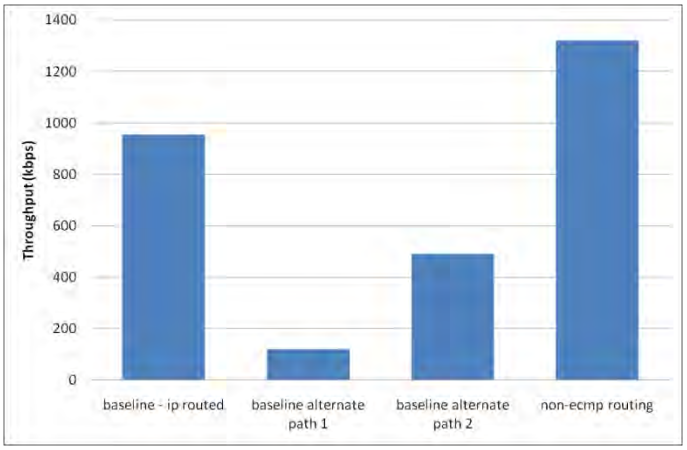

Figure 6 Throughput results for Non-Equal Cost Multipath Routing

The process that we used to evaluate our Non-ECMP technology was to first measure the throughput of the IP routed path, which is the green traffic from Figure 5. Then we measured the throughput available in just the alternate paths, which is the yellow and red traffic in Figure 5. Finally, we measured the throughput using Non-ECMP routing, which means the green traffic followed the IP routed path and the yellow and red traffic followed their respective policy-based routed paths. Figure 6, shows the IP routed path had a bandwidth of 955kbps and the two alternate paths had an additional bandwidth of 610kbps. The resulting throughput for Non-ECMP, which uses the bandwidth from the IP routed path and that of the alternate paths was 1320kbps, which is approximately the sum of all the path bandwidths. Theoretically, the achieved throughput should have been 1565kbps, however after analyzing our tests we discovered that we set the rate-limit to 800kbps for the IP routed traffic; therefore approximately 155kbps of bandwidth available on the IP routed path was unused. Once modified our Non-ECMP routing throughput should increase and be closer to the theoretical max bandwidth.

In general, the percentage of throughput gains of Non-ECMP routing is calculated using the following equation.

$$\textbf{percentage improvement} = 100 * \frac{\textbf{policy based routed path}}{\textbf{IP routed path}}$$

In our case, we primarily used an emulated 1Mbps TTNT link for the IP routed path and a 432kbps emulated SATCOM link for the policy-based routed paths. Therefore, our percentage of improvement should be approximately 40% and our results above show we measured a 38% throughput gain.

Lastly, it is important to note that our Non-ECMP routing capability used a rate-limit on the black side router to

determine the amount of arriving traffic that should be marked green, yellow, or red. However, by doing the traffic marking on the black side we are not able to mark all the packets from the same TCP flow with the same color, because the IP addresses and TCP ports are encrypted. Therefore, it is possible that some packets in the TCP flow could take one of the alternate paths, arriving out-of-order at the destination. The out-of-order packets can significantly degrade TCP flow performance. The best solution to this issue is to do the packet marking on the red side where we can identify packets by flow. This is typically referred to as flow-based load balancing versus packet-based load balancing. We used packet-based load balancing to prove Non-ECMP routing does increase throughput, however it should be implemented using flow-based load balancing by marking the packets on the red side router.

Load balancing tests based on TCP flows are more difficult than the plain throughput tests above because TCP has a built-in congestion control algorithm that will force the sender to slow down when packet timeouts or retransmissions are encountered. Load balancing can magnify those effects by sending packets from the same TCP stream over multiple paths.

Three tests were run, each with 10 independent TCP flows:

1. IP routing baseline test, without load balancing
2. Dynamic load balancing where only the rate limit approach was used
3. Static load balancing, where 60% of the TCP flows used the IP routed path, 30% used policy route 1 and 10% used policy route 2, as defined in the previous section.

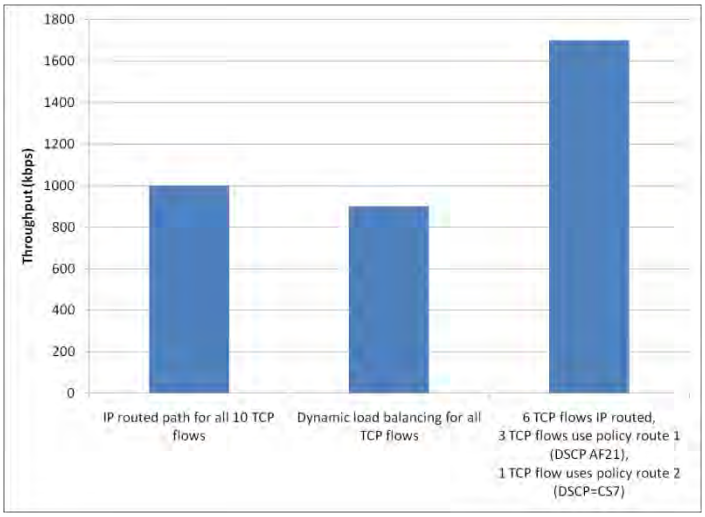

Figure 7 TCP Throughput results for Non-Equal Cost Multipath Routing

As expected, the dynamic load balancing approach caused underperformance due to TCP congestion control, while the static load balancing allowed greater network throughput than either the IP routed or the dynamic load balancing approach. These results, in Figure 7, show that with multiple TCP flows, the extra flexibility of being able to route some of the IP flows off the IP routed path, which is determined by the dynamic routing protocol, increased total network throughput. With the current COTS equipment and configuration, this can only be done statically; in the event that the policy route is not available, the router will use the IP routed path so the TCP flow is not dropped.

4. CONCLUSION

In this paper, we discussed the dynamic routing architectures for airborne platforms that were implemented at recent live-fly exercises, which utilized commercial-off-the-shelf (COTS) routers to enable a near-term airborne network solution. We showed how our lab experimentation enabled us to find optimal OSPF and GRE configurations that were then used at Empire Challenge 2008 to show the benefits of dynamic routing in an airborne network. We further optimized dynamic routing for airborne platforms by integrating and testing an OSPF implementation for TTNT that enabled dynamic routing at JEFX09 without using GRE tunnels. The elimination of the tunnels improved OSPF scalability for airborne networks that include TTNT by reducing OSPF overhead. One lesson learned from our work with dynamic routing on airborne platforms is routing enabled terminals such as TTNT, should support route re-distribution.

Lastly, we offered another enhancement to COTS routers with load balancing, which further improves the current near term solution to airborne networking, which uses COTS routers and currently available radio terminals.

5. REFERENCES

1. Information Sciences Institute, University of Southern California, “Internet Protocol Darpa Internet Program Internet Specification”, IETF RFC 791, September 1981.
2. J. Moy, “OSPF Version 2,” IETF RFC 2328, April 1998.
3. http://www.rockwellcollins.com/products/gov/airborne/cross-platform/comm-systems/ttnt/index.html
4. http://www.inmarsat.com/Downloads/English/Aero/Swift64_fact_sheet_EN.pdf?language=EN&textonly=False
5. http://www.inmarsat.com/Downloads/English/BGAN/Collateral/bgan_overview_brochure_EN.pdf?language=EN&textonly=False

6. M. Mirhakkak, P. Ta, G. Comparetto, and V. Fineberg, “Modeling and Simulation of Haipe”, IEEE, MILCOM 2006, October 2006.
7. D. Farinacci, T. Li, S. Hanks, D. Meyer, and P. Traina, “Generic Route Encapsulation (GRE)”, IETF RFC 2748, March 2000.
8. John Moy, OSPF: Anatomy of an Internet Routing Protocol, Addison-Wesley, 1998.
9. Jeff Doyle & Jennifer Carroll, CCIE Professional Development Routing TCP/IP Volume 1, Second Edition, 2006.
10. http://www.quagga.net/
11. http://www.zebra.org/
12. Cisco IOS Quality of Service Solutions Command Reference, http://www.cisco.com/en/US/docs/ios/qos/command/reference/qos_cr.pdf, March 2009.
13. K. Nichols, S. Blake, F. Baker, D. Black, “Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers”, IETF RFC 2474, December 1998.
14. Policy Based Routing, Cisco White Paper, 1998, http://www.cisco.com/en/US/products/ps6637/products_ios_protocol_option_home.html.
15. Information Sciences Institute, University of Southern California, “Transmission Control Protocol Darpa Internet Program Protocol Specification”, IETF RFC 793, September 1981.