QoS ENHANCEMENTS TO BGP IN SUPPORT OF MULTIPLE CLASSES OF SERVICE

Lotfi Benmohamed [1], Chieh-Jan Mike Liang [2], Eric Naber [1], Andreas Terzis [2]

[1] Johns Hopkins University Applied Physics Laboratory
Laurel, Maryland, USA

[2] Johns Hopkins University Computer Science Department
Baltimore, Maryland, USA

ABSTRACT

The Border Gateway Protocol (BGP) is the dominant inter-domain routing protocol in IP-based networks today. However, the requirements of emerging applications have exposed limitations in the current BGP protocol. In particular, future military IP networks, exemplified by the Global Information Grid (GIG), will carry a diverse mix of applications with widely different Quality of Service (QoS) requirements. At the same time, the GIG includes a diverse set of component networks, such as tactical ad-hoc networks, with highly dynamic QoS characteristics.

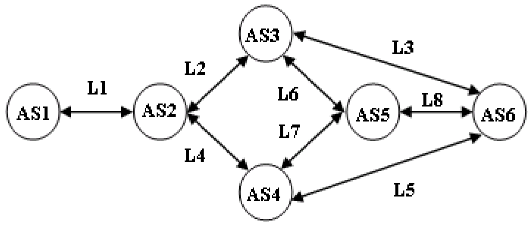
In this paper we investigate the problem of enhancing BGP to discover routing paths with QoS characteristics that match application requirements. We explore the requirements posed on multi-domain QoS routing protocols that provide multiple classes of service with multi-dimensional QoS requirements and present how these requirements map to BGP. We discuss enhancements to BGP that allow nodes to discover multiple paths with associated QoS attributes. In particular, we discuss a dominant path selection algorithm that allows nodes to discover the minimum set of paths needed to make QoS routing decisions. We present details of the proposed BGP changes and identify the modifications needed at each stage of the BGP path selection process. We implemented the proposed enhancements in the NS-2 simulator. Preliminary simulation results indicate the potential performance benefits of the introduced QoS enhancements to inter-domain routing.

1. INTRODUCTION

The Border Gateway Protocol (BGP) is the ubiquitous inter-domain routing protocol used to exchange reachability information among the Internet's Autonomous Systems (ASes). Given that BGP must operate in Internet-wide scale, it must generate minimal traffic overhead as well as have minimal routing state requirements. Moreover, it needs to account for restrictions imposed by commercial relationships among Internet Service Providers and between providers and their customers. Finally, the reluctance of competing service providers to share details of their network internals further limits the type of information that can be exchanged. For these reasons, BGP is a path vector protocol: the only information sent by an AS to its neighbors is the set of network prefixes reachable from that AS and, for each such prefix, the sequence of ASes on the path to that destination.

BGP is a single-path routing protocol, meaning that at most one route is advertised by an AS for any given destination. Specifically, after an AS (more accurately a router at the boundary between two ASes running BGP) receives multiple advertisements (UPDATEs in BGP parlance) from its upstream neighbors, it applies its routing policies to select the single neighbor used to reach that destination, and finally advertises this decision to its downstream neighbors. In this respect, BGP is application-agnostic because all traffic to a particular destination follows the same path. On the other hand, IP networks currently under development for civilian as well as military operations, will carry a mix of applications with diverse Quality of Service (QoS) requirements. At the same time, some of these internets will have a diverse set of component networks (wireless and wireline, fixed and mobile with different degrees of mobility, long lived and short term) and some of the component networks will be very dynamic in their service capabilities. Thus, different end-to-end routes between the same end points may offer very different QoS capabilities and these may vary over time. As a result, the ability to select among multiple routing paths based on the applications' QoS requirements will become an important need in emerging IP networks, especially in networks such as the Global Information Grid (GIG).

In our previous work, we proposed a set of extensions to the BGP protocol designed to expose paths with diverse QoS attributes to end-user applications [1]. Specifically, we allow for more than one route to be propagated in BGP UPDATE messages and the information propagated involves more QoS metrics than a simple sequence of ASes. In this follow-up work we report our results from an implementation of the proposed extensions based on the popular NS-2 network simulator. Our results indicate that

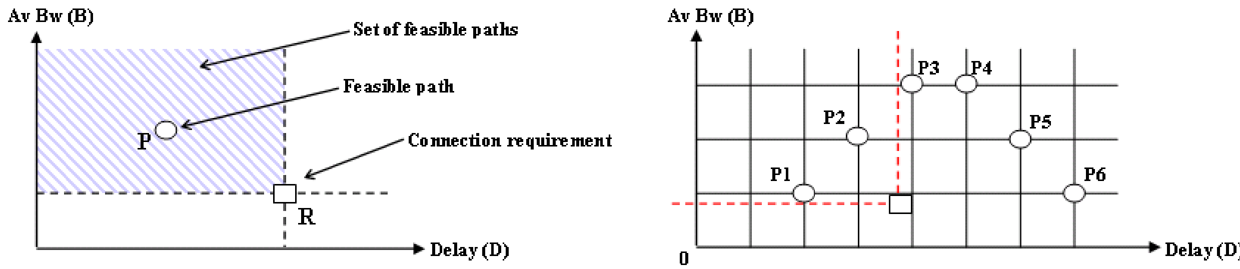


(a)

Link	L1	L2	L3	L4	L5	L6	L7	L8
Delay (D)	1	0	1	3	1	1	1	1
Av Bw (B)	4	4	1	4	3	4	4	2

Path	P1	P2	P3	P4	P5	P6
Hops	L1-2-3	L1-2-6-8	L1-2-6-7-5	L1-4-5	L1-4-7-8	L1-4-7-6-3
Delay	2	3	4	5	6	7
Av Bw (B)	1	2	3	3	2	1

(b)



(c) (d)

Figure 1. Support for an application class whose relevant metrics are bandwidth and propagation delay.

the proposed extensions can indeed calculate the sets of *dominant* paths to each destination with only a moderate increase in overhead.

The rest of the paper is structured as follows: In Section 2 we summarize the proposed extensions while in Section 3 we provide a proof showing that the extensions can accurately detect the list of dominant paths for each destination. Section 4 elaborates on our implementation and Section 5 presents our evaluation results. We close with a summary in Section 6.

2. BGP ENHANCEMENTS

There are five enhancements to BGP that enable multipath and QoS-aware routing: (1) exchanging potentially multiple paths per prefix, (2) maintaining QoS parameters for each path, (3) pruning the set of known paths to a dominant set while maintaining optimality, (4) choosing a particular path from this dominant set that best satisfies the unique QoS requirements of a particular application, and (5) enforcing the selected path. Enhancements (1), (2), (3), and (5) were discussed in-depth in our previous work [1]. As such, we will briefly review these four enhancements and then focus on fourth enhancement.

2.1. REVIEW

BGP restricts each router to advertise to its neighbors only one route per destination prefix. This information hiding behavior can prevent a router from learning the particular path that most appropriately provides the QoS requirements for a given traffic class. Enhancement (1) removes this limitation, allowing each BGP router to advertise a set of dominant paths. The notion of dominant paths is implemented through enhancement (3), and it prevents each BGP router from advertising every path it knows. Dominant paths are selected by the dominant path selection algorithm (DPSA), which is discussed further in Section 3. Enhancement (2) associates a list of QoS metrics with each path, which are then used in making routing decisions.

Exchanging additional paths with their associated QoS attributes enables QoS-aware routing to select appropriate paths which will then need to be enforced. Various mechanisms can be used to pin the selected path for a particular application's data flow. These options including MPLS are discussed in our previous work [1].

2.2. PATH SELECTION PER APPLICATION CLASS

The goal of enhancement (4) is to choose a routing path based upon the QoS requirements of a particular application class. BGP uses the coarse distance metric of hop count, among other things such as local policy, to decide on a routing path to any given destination. In order to affect QoS-aware routing, we alter this decision process so that we choose a path based upon the QoS metrics of each dominant path to the desired destination and the QoS requirements of the particular application class whose data path is being selected.

Given the four enhancements introduced above, the following information is known when making the decision at Source Node S regarding which path to select for traffic from application class A destined to destination D:

- All dominant paths from S to D.
- The QoS metrics of each dominant path from S to D.

- The minimum QoS requirements of traffic class A (note that the traffic class a packet belongs to can be inferred from the packet's header, such as the DSCP field). We assume that applications are aware of their minimum QoS requirements. For example, the quality of a VoIP call rapidly deteriorates when end-to-end delay is larger than 100 msec. This delay bound can then be used as a minimum QoS requirement for the class of VoIP flows.

Given this knowledge, we can choose the optimal path. Consider the network topology presented in Figure 1(a). Figure 1(b) enumerates all links in the network and their associated metric values, along with all of the possible loop-free paths from AS1 to AS6 and their associated metric values. Figure 1(c) depicts graphically the multiple QoS metrics; the y-axis represents bandwidth values, and the x-axis represents delay values. The point R in Figure 1(c) represents a set of minimum QoS requirements. Given that increases in bandwidth and decreases in delay are desirable, we see that the area of the graph that satisfies the minimum QoS requirements of R is the northwest quadrant of a shifted graph in which R is the origin. Any path whose QoS metric values fall within that quadrant belongs to the set of paths that satisfy the minimum QoS requirements of R and is referred to as the set of feasible paths.

During the route selection process, our BGP enhancements require two steps in addition to the standard BGP steps. The first step is to use the DPSA to prune the set of all known paths to the set of dominant paths. In the particular situation shown in Figure 1(d), P1, P2, and P3 comprise the dominant set among the six paths. This is easy to determine graphically by noting that P4, P5 and P6 all have another path in the northwest quadrant of a shifted graph where P4, P5, or P6, respectively, is the origin. The second step is the class-assignment algorithm in which at most one route is assigned to each class for every destination prefix. First, dominant paths with feasible QoS characteristics must be identified. In the example of Figure 1(d), these potential paths would be in the northwest quadrant of the request R represented by the square, and they are P1 and P2. If more than one dominant path is feasible, the algorithm chooses one path as follows. In our example of two QoS metrics, we choose the path with furthest distance from R in the quadrant of feasible paths, with the distance being the Euclidean distance from R to each point (*i.e.* P1 in Figure 1(d)). Using the furthest metric helps decrease route flapping as the path with the most "extra" resources (*e.g.* capacity) is used. Additionally, using the furthest metric is the easiest way to ensure that no routing loops are created. The proof of the non-existence of routing loops under this path selection rule will be detailed in a later paper. An alternative method to prevent routing loops is to make each node's class assignment known to its neighbors. This method however is less attractive since it creates additional message overhead.

This class-assignment algorithm is performed for each application class with different QoS requirements. Each optimal path determination is stored in the router's RIB (Routing Information Base) and used for subsequent routing lookups. Routers along the chosen path will forward the data packets along the pre-determined path in accordance with the option chosen to enforce the path.

It is possible for the DPSA and class-assignment algorithms to encounter ties in which two or more routes have equal metrics. The first tie-breaking rule is to pick the route with the fewest AS hop count. When there is also a tie in the hop count, the second rule is to pick the route with the lowest next-hop IP address.

3. DOMINANT PATH SELECTION ALGORITHM

We present a proof of convergence of the Dominant Path Selection Algorithm (DPSA) under the assumption of synchronous operation. Insights into the general asynchronous case will be provided through simulations presented in the following section. Under the synchronous model, enhanced BGP nodes update their dominant paths periodically under a common synchronous schedule as follows: at the beginning of a period, each node computes its set of dominant paths and exchanges it with its neighbors. Before the beginning of the next period all nodes would have received updates from all their neighbors and this cycle repeats for each consecutive period.

As discussed in the previous section, for a given set S of paths between a pair of nodes, if a path P in S is not dominated by any other path in S, P is said to be a dominant path of S. The set of all dominant paths of S is denoted dom(S). In order to show convergence, we need to first derive a set of properties of the dominance operator dom( ). The following three properties are needed:

$$P \in \mathrm{dom}(S) \Rightarrow P \in \mathrm{dom}(S^{'}) \quad \text{for any } S^{'} \subset S \text{ with } P \in S^{'} \qquad \text{(P0)}$$

$$\mathrm{dom}\left(S_1 \cup S_2\right) = \mathrm{dom}\left(\mathrm{dom}(S_1) \cup \mathrm{dom}(S_2)\right) \qquad \text{(P1)}$$

$$\mathrm{dom}\left(L \oplus S\right) = \mathrm{dom}\left(L \oplus \mathrm{dom}(S)\right) \qquad \text{(P2)}$$

Property P0 states that if path P is a dominant path of a set S then it is also a dominant of any subset of S that it belongs to. Property P1 states that the set of dominant paths of the union of two sets is equal to the set of dominant paths of the union of their dominant paths. For property P2, L is a link from source node to a neighbor node j, S is a set of paths from node j to the destination, and @ denotes concatenation, *i.e.*, L@S is the set of paths made up of paths of S concatenated with link L. Appendix A includes proofs for these properties.

Next, we address the problem of calculating the set of dominant paths from every node to a given destination. Without loss of generality we denote node 1 as a generic destination node. Let S_i^h be the set of all paths from node i to node 1 having a hop distance (or hop count) less than or equal to h. Let D_i^h be the set of h-dominant paths from node i to node 1, which is defined as the set of dominant paths from node i to node 1 with a hop count less than or equal to h, i.e. $D_i^h = \text{dom}(S_i^h)$.

Theorem: D_i^h can be generated iteratively as follows:

$$D_i^{h+1} = \text{dom}\left(\bigcup_{j \in N(i)} \{L_{ij}\} \oplus D_j^h \right) \quad \forall i \neq 1 \quad \text{(E1)}$$

$$D_i^0 = \varnothing \quad \forall i \neq 1 \quad \text{(E2)}$$

where (E2) is the starting initial condition.

Proof: We use induction to prove the theorem. We first show that (E1) is true for h=0. In this case we expect D_i^1 (the set of dominant paths from i to 1 which are one hop long) to be equal to the single-link path between nodes i and 1 (if it exists). Indeed this is the case since

$$\begin{aligned} D_i^1 &= \text{dom}\left(\bigcup_{j \in N(i)} \{L_{ij}\} \oplus D_j^0 \right) \\ &= \text{dom}\left(\left(\bigcup_{j \in N(i), j \neq 1} \{L_{ij}\} \oplus D_j^0 \right) \bigcup \left(\{L_{i1}\} \oplus D_1^0 \right) \right) \\ &= \text{dom}\left(\text{dom}\left(\bigcup_{j \in N(i), j \neq 1} \{L_{ij}\} \oplus D_j^0 \right) \bigcup \text{dom}(\{L_{i1}\}) \right) \\ &= \text{dom}\left(\{L_{i1}\}\right), \quad \text{since } D_j^0 = \varnothing \ \forall \ j \neq 1, \text{ and } D_1^0 = \{\text{node } 1\} \\ &= \{L_{i1}\} \end{aligned}$$

Suppose that D_i^k is the set of h-dominant paths from i to 1 for all $k <= h$, for $k=h+1$ we have:

$$\begin{aligned} D_i^{h+1} &= \text{dom}\left(\bigcup_{j \in N(i)} \{L_{ij}\} \oplus S_i^h \right) \\ &= \text{dom}\left(\bigcup_{j \in N(i)} \text{dom}\left(\{L_{ij}\} \oplus S_i^h \right) \right) && (a) \\ &= \text{dom}\left(\bigcup_{j \in N(i)} \text{dom}\left(\{L_{ij}\} \oplus \text{dom}(S_i^h) \right) \right) && (b) \\ &= \text{dom}\left(\bigcup_{j \in N(i)} \{L_{ij}\} \oplus \text{dom}(S_i^h) \right) && (c) \\ &= \text{dom}\left(\bigcup_{j \in N(i)} \{L_{ij}\} \oplus D_i^h \right) && (d) \end{aligned}$$

where (a) is due to property P1, (b) is due to P2, (c) is due to P1, and (d) is due to the induction hypothesis. ■

If DPSA terminates after H iterations, we must have

$$D_i^h = D_i^H \text{ for all } i, \text{ and for all } h \geq H$$

At the h-th iteration, the algorithm computes all dominant paths that have a hop count <= h. In a network of N nodes, the longest path has N-1 hops, therefore

$$D_i^h = D_i^{N-1} \text{ for all } i, \text{ and for all } h \geq N$$

Hence, the synchronous algorithm converges within N-1 iterations.

When the synchronous version of DPSA (equation E1) converges, we have

$$D_i = \text{dom}\left(\bigcup_{j \in N(i)} \{L_{ij}\} \oplus D_j \right)$$

This equation corresponds to the asynchronous implementation of the algorithm: Each node j asynchronously transmits its D_j to its neighbors when a change occurs and each node i asynchronously executes the above iteration using the latest D_j received from its neighbors.

4. SIMULATION TOOLS

We begin by describing our high-level requirements for a simulation package. To evaluate the network-level impacts of the proposed BGP extensions, the simulation tool should be capable of representing a high fidelity BGP model. At the same time, to evaluate the impact of these decisions to end-user applications, the simulator must couple routing decisions made at the control plane with packets transported by the (simulated) data plane.

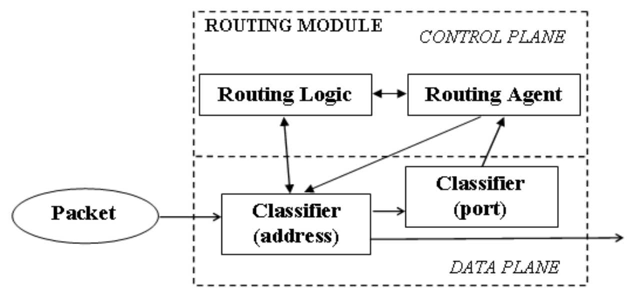


Figure 2. Interactions inside a typical NS2 routing module.

The combination of NS-2 [2] and BGP++ [3] simulators provide the best match for these requirements. NS-2 is a popular event-driven network simulator, and it offers a broad support of network protocols. BGP++ is an extension to NS-2 that provides a BGP simulation model. Since BGP++ is a port of the GNU Zebra BGP Daemon [4], it inherits most of Zebra functionalities and flexibility.

We modified these tools to resolve the following shortcomings. First, although BGP++ supports CIDR IPv4 addressing scheme (e.g. the set of 256 addresses from 192.168.1.1 to 192.168.1.255 is represented as 192.168.1/24), NS-2 supports only a flat addressing scheme and a three-layer hierarchical addressing scheme. Second, BGP++ was primarily developed to study the routing control plane, so the interface between data plane and control plane was not implemented. Finally, implementing the proposed enhancements requires modifications to packet structures, RIB, and the forwarding decision processes.

Central to most modifications is a new NS-2 routing module we developed. As Figure 2 illustrates, such a module manages three functional blocks of a routing protocol: the routing agent, the routing logic, and the classifiers. The routing agent and routing logic represent the control plane which is responsible for exchanging routing messages and maintaining routing tables, also known as RIBs. Most control plane modifications focused on altering the decision process and extending the RIB. Specifically, we incorporated the dominant path selection algorithm as part of the BGP route selection process. To do so, we extended the RIB to store multiple paths and route assignments for each QoS class under a destination prefix. In NS-2, every node consists of one or more classifiers responsible for determining where packets should be forwarded next. As mentioned above, NS-2 does not support IPv4 addressing scheme, therefore the data plane cannot reference routing tables maintained by the control plane. We resolved this limitation by modifying the BGP classifier to add support for a CIDR-based addressing scheme. Specifically, the BGP classifier is able to perform longest-prefix-matching on the packet destination address. In addition, the BGP classifier incorporates QoS class-based forwarding decisions, which selects the next hop according to the packets' class information. With these modifications the BGP classifier allows NS-2 to consult the BGP Routing Information Base (RIB) for forwarding decisions and successfully couple the control plane routing decision with the data plane forwarding decision.

In order to test the end-user performance provided by the proposed BGP extensions we need an end-user application model. Rather than building an application model from scratch, we opted to modify the UDP agent distributed with NS-2. We modified this constant bit rate agent in the following ways: First, the modified agent provides interfaces to specify the QoS class information for the packets it transmits. Second, the sequence numbers embedded in these UDP packets make it possible to track the progress of these packets in the network throughout the simulation. Finally, a new NS-2 command allows users to assign IPv4 addresses to nodes. These addresses are then used as destinations for packets sent by the simulated sources and are used by the BGP classifier for its forwarding decisions.

5. SIMULATION RESULTS

5.1. DYNAMIC BEHAVIOR

In this section, we demonstrate the ability of the proposed enhancements to choose routes with desirable QoS attributes as network conditions change. To do so, we simulated a network topology with seven nodes and three paths between the data source and the destination. Figure 3 illustrates the link characteristics of the topology. In addition, path 1-5-6-7 was modeled as an intermittent path by initially taking down all links on this path and periodically altering their state between the *up* and *down* states every 200 seconds. An UDP agent was attached to node 1 as the source, and it was configured to inject a 500-byte packet into the network every 200 milliseconds. The data source simulates an application that requires minimal delay from the network (*e.g.* VoIP). A sink was attached to node 7 as the destination of the packet flow.

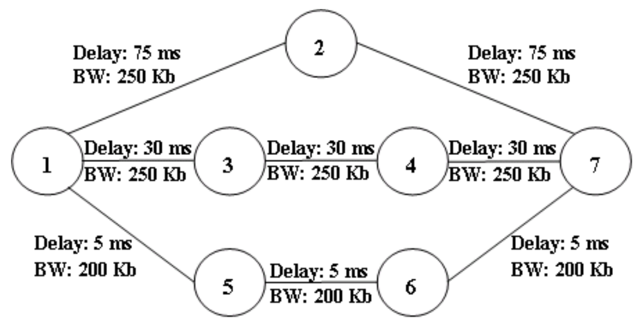


Figure 3. Network topology used in the simulation of dynamic behavior.

Figure 4 shows the delay that packets experience on the network path from the source to the destination when paths are selected by standard BGP and when the QoS-enhanced BGP is used.

BGP bases its route selection decision on path attributes. One of the decisive path attributes is AS_PATH, in which a path with lower AS_PATH count is preferable. Figure 4 shows that BGP chose path 1-2-7 throughout the simulation and as a result packets experience 180ms of network delay (propagation plus transmission delay). Our QoS-enhanced BGP alters the decision process and chooses the path with the lowest end-to-end delay. This behavior is shown by the sudden drop in delay at the 219th second. Initially, path 1-3-4-7 was chosen because it had the lowest delay between the two available paths. However, when path 1-5-6-7 becomes available, the route-assignment algorithm realizes that a better path, given the class requirements, is feasible and routes packets accordingly. Conversely, when path 1-5-6-7 was disconnected, the route-assignment algorithm switched back to the previous path.

The initial network convergence time is 8.59 seconds. Given that path 1-5-6-7 alternates between the *up* and *down* states every 200 seconds, we can estimate the protocol convergence time from Figure 4 as the intermittent link changes its state. It took approximately 19 seconds and 7 seconds for the network to converge the first time and the second time path 1-5-6-7 was brought up respectively. On the other hand, the network converged 178 seconds after path 1-5-6-7 was taken down, and this is due to the large default value of the BGP hold time of 180 seconds.

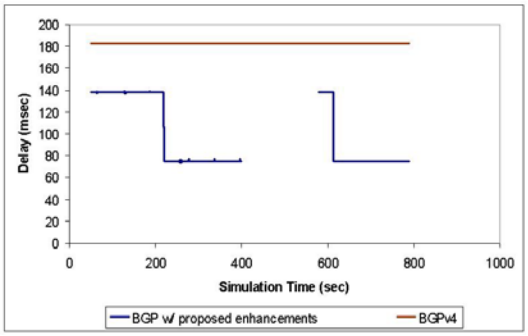


Figure 4. Delay experienced by packets vs. packet departure time in the simulation of dynamic behavior.

5.2. SCALABILITY

The proposed enhancements allow BGP routers to advertise multiple routes to a given prefix, thus generating additional network overhead. In this section, we evaluate the scalability of the proposed modifications in terms of the number of BGP update messages as well as protocol convergence times.

To automate the process of topology description for NS-2, we base our topologies on the architecture of the Global Information Grid (GIG) network, an IP network currently under development by the U.S. government. To generate these topologies, we modified a PERL script[1] that takes as input several GIG parameters that characterize the network, such as the number of nodes in each sub-network and the number and types of sub-networks. Since we focus on the network as a whole, most sub-networks were approximately of the same size. We experimented with three network topology sizes: 48 nodes / 88 duplex links, 108 nodes / 198 duplex links, and 153 nodes / 300 duplex links. By increasing the size of sub-networks by a factor of two, the network topology grows by a factor of two.

Each topology was simulated over three phases, and the protocol overhead was measured during each phase. The first phase starts when the simulation starts and lasts until the protocol converges (*i.e.* no more BGP UPDATEs are sent). The second phase starts when a link is removed and lasts until the next network convergence. Finally, the third phase starts when the same link is brought back up until the final network convergence. The choice of the link that is removed affects the convergence time. The reason is that as the number of dominant paths using the intermittent link increases, the amount of update message traffic generated after each link state change increases as well. To compensate for the inherent randomness of this process we simulated each topology size five times choosing a random link to remove for each simulation run.

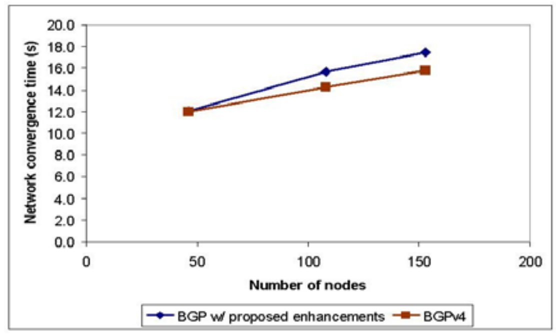


Figure 5. Relationship between the size of network topology and network convergence time in phase 1.

Figure 5 illustrates the relationship between the network topology size and the convergence time in phase 1. It took 12.078, 15.682, and 17.453 seconds for the QoS-enhanced BGP to converge as we increased the network topology size. Figure 5 also compares BGP to BGP with our enhancements. Both flavors of BGP have the same increasing trend, but BGP converged slightly faster. In the network with 153 nodes, BGP converged approximately 10% faster than BGP with our enhancements.

Table 1. Average number of BGP update messages exchanged during each of the three phases.

# Nodes	BGP	BGP w/ proposed enhancements		
	Phase 1	Phase 1	Phase 2	Phase 3
48	38174	38188	2790	1477
108	307469	335729	2384	1943
153	468398	484657	1081	1757

Table 2. Maximum and minimum number of BGP update messages exchanged during each of the three phases.

# Nodes	Phase 1	Phase 2	Phase 3
48	46074/34150	6877/0	3297/223
108	375213/285737	9553/0	5176/256
153	519780/434930	2783/128	3648/224

[1] Original script provided by Bob Cole of JHU/APL.

Table 1 shows the average number of BGP UPDATE messages exchanged during each of the three phases. Given the inherent variability of the underlying process (*i.e.* the number of messages exchanged depends on the number of dominant paths that use the link that is removed and re-inserted) it is also interesting to count the minimum and maximum number of messages for each phase. Table 2 summarizes these statistics for the different network sizes. For example, in the network with 48 nodes, the third simulation run exchanged a relatively high number of BGP update messages because it had more dominant paths using the intermittent link. On the other hand, the fourth simulation run did not have any dominant paths using the periodic link, which resulted in no BGP update messages being exchanged in phase 2. Table 2 shows that similar result variations also exist in phase 3, but we expect that every simulation run to generate some update message traffic in this phase because the two nodes of the periodic link have to exchange BGP updates after the periodic link is up.

6. SUMMARY

While the existing BGP protocol provides only a single path to each network destination, emerging applications need QoS support from the network and require that multiple paths with diverse QoS characteristics be exposed. In this work we evaluate the feasibility of the extensions to the BGP protocol proposed in [1] through simulation. Our results are very encouraging: the proposed extensions expose multiple routing paths with diverse QoS attributes, from which applications can select the ones that fit their needs. Furthermore, the additional overhead associated with providing these paths is only moderate, indicating that such a solution is feasible for future networks such as the Global Information Grid.

REFERENCES

[1] L. Benmohamed, B. Doshi, T. DeSimone, R. Cole, “Inter-Domain Routing with Multi-Dimensional QoS Requirements”, IEEE Milcom 2005.

[2] NS2, "The Network Simulator – ns-2", USC - ISI, http://www.isi.edu/nsnam/ns/, March 2003.

[3] X. Dimitropoulous and G. Riley, "The BGP++ simulation", Georgia Tech University, February 2005.

[4] GNU Zebra, "The GNU Zebra Routing Demon - Border Gateway Protocol", http://www.zebra.org/, January 2005.

APPENDIX A

Property P0: To show that A→B we show that nonB→nonA instead:

$$P \notin \text{dom}(S') \Rightarrow \exists\, P' \in S' \text{ s.t. } P' \text{ dominates } P$$
$$\Rightarrow \exists\, P' \in S \text{ s.t. } P' \text{ dominates } P \text{ (since } S' \subset S)$$
$$\Rightarrow P \notin \text{dom}(S)$$

Property P1: We will show that if a path P belongs to D_1 then it must also belong to D_2, and vice versa, where:

$$D_1 = \text{dom}\left(S_1 \cup S_2\right) \text{ and } D_2 = \text{dom}\left(\text{dom}(S_1) \cup \text{dom}(S_2)\right)$$

If a path P belongs to D_1 then:

there does not exist $P' \in S_1 \cup S_2$ s.t. P' dominates P (*),
also $P \in \text{dom}(S_1) \cup \text{dom}(S_2)$ because otherwise:
$P \notin \text{dom}(S_1) \Rightarrow \exists\, P'' \in S_1$ s.t. P'' dominates P, which contradicts (*).
Since $\left(\text{dom}(S_1) \cup \text{dom}(S_2)\right) \subset \left(S_1 \cup S_2\right)$ and P is domiant in $S_1 \cup S_2$,
then using property P0 we also have $P \in D_2$

If a path P belongs to D_2 then:

$P \in \text{dom}(S_1) \cup \text{dom}(S_2)$, and (1)
$\exists\!\!/\; P' \in \text{dom}(S_1) \cup \text{dom}(S_2)$ s.t. P' dominates P (2)
(1) $\Rightarrow P \in S_1 \cup S_2$ (1')
(2) $\Rightarrow \exists\!\!/\; P' \in S_1 \cup S_2$ s.t. P' dominates P (2'), because otherwise if
$\exists\, P' \in S_1 \cup S_2$ s.t. P' dominates P, then either
$P' \in S_1 \Rightarrow \exists\, P'' \in \text{dom}(S_1)$ that dominates P, or
$P' \in S2 \Rightarrow \exists\, P'' \in \text{dom}(S_2)$ that dominates P,
implying $\exists\, P'' \in \text{dom}(S_1) \cup \text{dom}(S_2)$ that dominates P, contradicting (2).
(1') and (2') $\Rightarrow P \in D_1$

Property P2: We will show that if a path P belongs to D_1 then it must also belong to D_2, and vice versa, where:

$$D_1 = \text{dom}\left(L \oplus S\right) \quad \text{and} \quad D_2 = \text{dom}(L \oplus \text{dom}(S))$$

If a path P belongs to D_1 then:

$P = L \oplus \tilde{P}$, $\tilde{P} \in S$, and $\exists\!\!/\; P' \in L \oplus S$ s.t. P' dominates P (*),
also $P \in L \oplus \text{dom}(S)$ because otherwise:

$\tilde{P} \notin \text{dom}(S) \Rightarrow \exists\; \tilde{P}' \in S$ s.t. $\tilde{P}'$ dominates $\tilde{P} \Rightarrow$

$L \oplus \tilde{P}' \in L \oplus S$ dominates $P = L \oplus \tilde{P}$, which contradicts (*).
Since $\left(L \oplus \text{dom}(S)\right) \subset \left(L \oplus S\right)$ and P is domiant in $L \oplus S$,
then using property P0 we also have $P \in D_2$

If a path P belongs to D_2 then:

$P = L \oplus \tilde{P} \in L \oplus \text{dom}(S)$, and (1)
$\exists\!\!/\; P' \in L \oplus \text{dom}(S)$ s.t. P' dominates P (2)
(1) $\Rightarrow P \in L \oplus S$ since $\left(L \oplus \text{dom}(S)\right) \subset \left(L \oplus S\right)$ (1')
(2) $\Rightarrow \exists\!\!/\; P' \in L \oplus S$ s.t. P' dominates P (2'), because otherwise if

$\exists\, P' = L \oplus \tilde{P}' \in L \oplus S$ s.t. P' dominates P, then

$\tilde{P}' \in S$ dominates $\tilde{P} \Rightarrow \exists \tilde{P}'' \in \text{dom}(S)$ that dominates $\tilde{P} \Rightarrow$

$P'' = L \oplus \tilde{P}'' \in L \oplus \text{dom}(S)$ dominates P, contradicting (2).
(1') and (2') $\Rightarrow P \in D_1$