A Case Study in Understanding OSPF and BGP Interactions Using Efficient Experiment Design

David Bauer†, Murat Yuksel‡, Christopher Carothers†and Shivkumar Kalyanaraman‡
†Department of Computer Science
‡Department of Electrical, Computer, and Systems Engineering
Rensselaer Polytechnic Institute
110 8th Street, Troy, NY 12180, USA.
{bauerd, chrisc}@cs.rpi.edu, {yuksem,shivkuma}@ecse.rpi.edu

Abstract

In this paper, we analyze the two dominant inter- and intra-domain routing protocols in the Internet: Open Shortest Path Forwarding (OSPFv2) and Border Gateway Protocol (BGP4). Specifically, we investigate interactions between these two routing protocols as well as overall (i.e. both OSPF and BGP) stability and dynamics. Our analysis is based on large-scale simulations of OSPF and BGP, and careful design of experiments (DoE) to perform an efficient search for the best parameter settings of these two routing protocols.

1 Introduction

Understanding routing protocol dynamics and interactions on a large-scale is an important problem due to its immediate affect on current practice of inter- and intra-domain routing [1]. Network simulation allows us to consider multiple Autonomous Systems (AS), and to quantify the possible effects both from within and from outside a particular domain. However, because of the computational complexity within such models, the simulation community has primarily focused on tools which allow for large-scale parallel and/or distributed experimentation [2, 3, 4]. But beyond just model complexity, this problem equally has a Design of Experiment (DoE) complexity problem [4]. To address that problem in a real-world case study, we apply ROSS.Net in an attempt to begin to understand complex protocol interactions between the BGP4 and OSPFv2 routing protocols.

In this paper, we focus on characterizing Internet routing protocol performance response by the number of update messages generated by each routing protocol model (i.e., OSPF and BGP) as a function of protocol timers, variables, and algorithm decisions. Measuring protocol response as a function of update messages generated is important because this is where the interactions between protocols are defined. For example, in a network where route flapping is occurring, routers may converge quickly between the two routes as they change. Measuring convergence as the response would lead us to believe there are no negative effects on OSPF from BGP. Similarly, measuring link congestion does not lead to the negative effect because we may observe only a fractional difference in bandwidth consumption over time. Each route removal/installation can be directly measured within the OSPF domain. By measuring the number of updates generated by the OSPF domain, a clearer picture of the negative effects emerges. Of course these negative effects lead to slower convergence times and greater link utilization, but these are secondary measures. By measuring the interactions directly (i.e., updates messages) we are able to quantify the direct impact on the network without having to separate out other effects. This allows us to begin answering the questions, *does my intra-AS management policy adversely affect my inter-AS policies, and vice-versa?* and *which is the best approach to minimizing negative effects between protocols?*

1.1 Why are Protocol Interactions Harmful?

Network protocol weaknesses are not fully understood until they have been deployed in large-scale production environments. There is probably no better example of this than the BGP protocol. Clear limitations of this protocol have been illustrated since its introduction (e.g. BGP storms [6, 7], the stability problem [8, 9]), and several solutions (e.g. route reflection) have been proposed and implemented to overcome them.

These investigations have typically focused on the individual effects of the parameter settings, and neglected the external effects on protocol performance. The problem that we see is that there are two conflicting views of the network: intra-domain and inter-domain routing. Our concern is that decisions made to efficiently route data *within* a domain are directly affecting the ability of the network to route data *across* the domain.

One immediate cause for concern is **Hot Potato Routing** [10], though some researchers have similarly voiced concerns over Cold Potato routing [11]. Hot Potato routing is interesting because it allows a router which does not necessarily contain an up-to-date view of the internal network to make decisions about how to route traffic through that network. As a practical example, the BGP protocol makes a decision about which routes to install based on the distance of each competing intra-domain route. The problem arises when this information is not stable. BGP routers typically are responsible for generating large flows of traffic data into and out of the network. The major concern is that a *small* degree of unstable routing information may inversely impact a *large* amount of network traffic.

Traffic shifts because of OSPF-BGP interactions happen typically at ASes with multiple paths to another ISP. More than half of the non-ISP ASes have such multiple paths to a tier-1 ISP [12]. Previous work indicates that Hot Potato changes can cause major shifts in routing and network traffic. In addition, hot-potato routing may add to the degradation of forwarding plane convergence and generate temporary forwarding plane loops. Finally, Hot Potato routing leads to measurement inaccuracies in probes of the forwarding plane, and the external visibility of BGP routes.

1.2 Our Contributions

Our main goal is to minimize the number of negative interactions between the OSPF and BGP protocols in a multi-AS environment. In particular, the number of OSPF updates caused by BGP protocol dynamics, and the number of BGP updates caused by OSPF protocol characteristics should be *minimized*. Past investigations relying on measurement data have been constrained to one-way analysis of either BGP dynamics on the OSPF protocol, or OSPF dynamics on the BGP protocol, and then only for a single AS.

Our major contributions can be itemized as follows:

A framework to optimize OSPF and BGP protocol response: Based on a controlled large-scale simulation of OSPF and BGP, we present a framework to optimize a particular protocol performance metric over possible parameter search space through a heuristic search algorithm. We particularly use the number of updates with various original causes (i.e., OSPF-caused, BGP-caused) as metrics to optimize several OSPF and BGP parameters used in practice. Instead of measurement-based estimation and matching methods, we leverage a controlled simulation environment to trace exact causes of each routing update in the system.

Experiment design approach to understand OSPF and BGP interactions: We devise a systematic design of experiments methodology to investigate particular effects of three classes of protocol parameters in the total number of negative interactions between OSPF and BGP. We measure the negative interactions as the total number of OSPF-caused BGP updates and BGP-caused OSPF updates. We investigate three classes of parameters as factors into the negative interactions: (i) OSPF timers, (ii) BGP timers, and (iii) BGP decision making attributes.

Large-scale OSPF and BGP simulation: We present large-scale simulation of OSPF and BGP in a single model. Our simulation model uses realistic inter- and intra-domain topology generated from Rocketfuel [5] measurement data and nearly complete RFC implementations of the OSPF and BGP protocols.

2 Related Work

Analysis of interactions between inter- and intra-domain routing protocols has been an attractive research topic. In [13], through analysis of data from AT&T's BGP and OSPF traffic measurements, authors showed that majority of BGP updates are because of Hot Potato decision-making practices of ISPs. The main difference in our work is that we do not need any matching or estimation technique to determine OSPF-caused BGP updates or vice versa. Since our large-scale simulation environment is fully controlled, we can easily trace the causes of updates.

In [10], as a follow-up to their previous work [13], authors modeled sensitivity of BGP (and the network in general) to IGP-caused Hot Potato changes. The bottom line result is one needs to enumerate all possible Hot Potato IGP changes to perform BGP analysis.

Another major work on analysis of OSPF and BGP interactions was presented in [12]. In contrast to the research direction on analyzing effects of intra-domain changes on inter-domain routing as in [13, 10], the main goal of the work in [12] was to determine if BGP dynamics effect intra-domain routing behavior and in turn effect the traffic engineering of it.

There has been significant research on convergence time and stability of both BGP and OSPF [14, 15, 16, 17]; BGP security and misconfiguration [18, 19, 20] and BGP quality of service extensions [21, 22]. Our work can potentially enhance the key results already generated by these efforts.

In placing this work in the context of the larger modeling and simulation community, we are driven by the need to "obtain good results, fast". This performance driven need has been attacked on many fronts. Clearly one of the most popular approaches is the application of many processors to speedup the execution of a simulation, such as done with such systems as [23, 24, 25, 26, 27]. In this case, we employ a technique that greatly reduces the numbers the experiments that must be run and aides in the search process for the set of parameters that provides "good" protocol performance.

3 ROSS.Net

The goal of our black-box simulation was to simulate a portion of the Internet by simulating multiple ASes, with multiple OSPF areas per AS, and multiple sub-networks per area. The investigation centers around link weight changes at the OSPFv2 routers, and link status changes which occur globally throughout the topology. This section details how the model was built and what assumptions were made during model construction.

We use the ROSS.Net framework to perform systematic analysis of OSPF and BGP dynamics and interactions. Our framework includes (i) a simulation component for simulating large-scale network protocol scenarios, and (ii) a Unified Search Framework which currently employs the heuristic search algorithm, Recursive Random Search (RRS) [28], to seek the best parameter settings of the system under consideration. In our experiments, we use the count of routing updates as the metric to optimize, and several OSPF and BGP protocol parameters as the parameters of the optimization.

3.1 Network Protocol Simulation Models

The network protocols simulated include: BGP4, OSPFv2, and IPv4. For scalability, the TCP layer was not simulated for the BGP routers. The assumptions made for scalability and time available for development are addressed in this section.

OSPFv2 is a link-state routing protocol designed to run internal to a single AS. Each OSPFv2 router maintains an identical database describing the AS network's topology. From this database, a routing table is calculated by constructing a shortest-path tree. OSPFv2 recalculates routes quickly in the face of topological changes, utilizing a minimum of routing protocol traffic.

We developed our OSPFv2 simulation model as a nearly complete RFC implementation citeRFC2328. The only exception is that Network LSAs were not modeled, and so the topology was not configured with stub networks. Our **BGP4 model** simulates both eBGP and iBGP according to [29]. External peers were defined by connections to BGP routers outside the current AS. Internal peers were fully connected to all eBGP routers in a given AS. With the exception of the error notification details, we have nearly fully modeled the BGP4 protocol per the RFC specification. In BGP, the decision making process occurs in three phases. The first phase is related to calculating route preferences according to owner-defined policies. The second phase selects the best route to each destination based upon the route attributes, and installs those routes into the local RIB (Loc-RIB). Phase 3 involves route aggregation and dissemination, which we did *not* model since route aggregation and information reduction are not described in the RFC and are optional and commercially dependent. Routes are disseminated as appropriate for eBGP and iBGP, and external control traffic minimized by using the MinRouteAdver-

Table 1: Stages of the BGP decision algorithm for route selection.

BGP Decision Algorithm
1. Highest Local Preference
2. Lowest AS Path Length
3. Lowest Origin Type (0 iBGP, 1 eBGP, 2 Incomplete)
4. Smaller MED (iff next hops equal)
5. Lowest IGP Cost
6. Lowest Next Hop
7. Lowest BGP Identifier
8. Vendor-dependent Tie Break

tisementInterval (MRAI). The specification simply calls for MRAI seconds to elapse between successive route updates between any two eBGP speakers.

We have modeled all other parts of the decision making algorithm, illustrated in Table 1, and the default decision if a tie exists at all levels is to keep the existing route. Conversely, if no route exists, the route in question is always added to the RIB. In our model the vendor dependent tie breaking decision is to keep the existing route.

In order to reduce the complexity of the design of experiments, input parameters are configurable at the AS level. Some of the variables in the BGP decision stage are simply on/off, such as MED and Hot Potato routing, which signify that these features are either enabled or disabled in the AS throughout the simulation. Other variables can take a value in a range, such MED, path padding and Local-Pref, which indicate the values of each feature, if enabled. So if MED is enabled in an AS, we can also select a value for it's policy. Conversely, if MED is disabled, then the AS will have no MED attribute set. Additional parameters involve the timers in BGP model, such as the MRAI which significantly affects the total number of update events in the system. KeepAlive messages are affected by the interval at which KeepAlives are sent; the HoldInterval, or just Hold, determines how many KeepAlive messages can be missed before a connection is disabled.

The IP simulation model we developed is very simple and only responsible for keeping statistical data such as packets forwarded, dropped or completed. The main function of the IP model is to determine the destination port for each packet in the system. The IP model determines which port should be used by first determining if a link to the destination exists for the packet. If not, then a routing table lookup is done. If this also fails, then the packet is dropped by the network.

4 Large-Scale Network Configuration

We used the topological data measured by the University of Washington's Rocketfuel project [5]. In totality, the Rocketfuel data identifies internal AS topologies for 10 major ISPs which cover a large area of the world. Our simulation included 5 of these ISP topologies. We chose the ISPs based upon reachability, ISP size and number of external routers detected. For example, for scalability purposes, the AT&T and Sprint topologies were excluded from this study, both having greater than 10,000 routers. The Verio map was not used because the high number of external routers would have required greater than 3.5 million iBGP connections. Finally, the Telstra and VSNL maps covering India and Australia were not used because they could not be connected to the remaining ISPs from the available data.

Rocketfuel data lists routers as having both internal and external ISP connections. We established an OSPFv2 router at each router in the topology. Also, we determined that routers which contained one or more external connections to be BGP4 routers in addition to an OSPFv2 router. This means that some routers were running only OSPF, while others modeled both BGP and OSPF protocols. Each ISP was configured as a single AS, and within routers were broken down into two additional levels: areas and subnets. To determine the areas and the subnets, we used the IP addresses of the individual machines. For example, if two machines shared the same class A, B and C address, then we placed them into the same subnet. Areas were determined in the same way using the class A and B prefixes.

Because the Rocketfuel data relies on traceroute to determine the ISP topologies, certain limitations had to be addressed. Rocketfuel data does not define the bandwidth, speed or delay of the links. Link bandwidth and delay classes were defined for the different Rocketfuel-determined router levels. The bandwidth and delay for the topology is as follows:

- **Level 0 routers:** 9.92 Gb/sec and 1 ms delay
- **Level 1 routers:** 2.48 Gb/sec and 2 ms delay
- **Level 2 routers:** 620 Mb/sec and 3 ms delay
- **Level 3 routers:** 155 Mb/sec and 50 ms delay
- **Level 4 routers:** 45 Mb/sec and 50 ms delay
- **Level 5 routers and below:** 1.55 Mb/sec and 50 ms delay

Table 2 outlines the details of the multiple AS topology. BGP routers within an AS are fully connected to form the iBGP domain. The degrees are listed for each AS to every other AS, and the total number of BGP connections is listed along the diagonal.

Table 2: Rocketfuel ISP Topology Parameters

ISP	#iBGP	AS0	AS 1	AS2	AS3	AS4
AS0: AboveNet	2,500	199	8	12	18	161
AS1: EBONE	16,384	8	38	6	12	12
AS2: Exodus	50,176	12	6	53	9	26
AS3: Tiscali	441	18	12	9	50	11
AS4: Level 3	7921	161	12	26	11	210

5 OSPF and BGP Interactions

We have generated 4 designs of experiments. The first design optimizes across 3 classes of parameters: BGP timers, OSPF timers, and BGP decision parameters. Taking these classes in different combinations will allow us to quantify the specific parameter effects first, then the feature interactions. The second design investigates the effects of cold versus Hot Potato routing in the BGP protocol model. This is significant because Hot Potato routing in BGP relies on information from the OSPF protocol. Because there is a direct correlation between the models, we expect there to be a direct feature interaction as well. The third design investigates the effects of various network management policies on the response. Broadly, there

are two approaches to network management at the AS level: greedy and cooperative. We define a greedy strategy as one in which the management policy promotes efficiency within the AS without consideration of the effects on the surrounding ASes. By contrast, a cooperative strategy is one in which the efficiency goal is considered across all of the ASes first. The final design considers the effects of the parameters on the response with varying degrees of network robustness. Here we perform a full-factorial on the topology parameters: link stability and link weight changes. The goal is to determine the range of the parameters and their interactions under varying network conditions.

5.1 Response Surface

Our response surface is defined by the number of network topology update messages exchanged by the BGP and OSPF protocols in the control plane. There are four types of update messages possible:

- OSPF caused OSPF updates (OO)
- BGP caused BGP updates (BB)
- OSPF caused BGP updates (OB)
- BGP caused OSPF updates (BO)

There are two types of changes which may occur in the topology: link status changes and link weight changes. The OSPF protocol detects link status changes via the HELLO protocol, and the BGP protocol via the KeepAlive Timer. Link weight changes are only detected by the OSPF protocol and are detected directly. When the OSPF protocol detects a change in the topology, it creates new LSAs appropriate for the cause and floods them throughout the OSPF domain. As the new LSAs are flooded they are accounted for in the "OSPF caused OSPF updates" statistic. The same is true for BGP caused BGP updates, and we do not distinguish between eBGP and iBGP route updates.

OSPF caused BGP updates are measured when the connection between two iBGP peers changes. This signals a change in the underlying OSPF network between the peers, and so the cause of the subsequent updates are attributed to the OSPF protocol. For example, a link which was previously down in the intra-AS domain becomes available again, and the OSPF network rebuilds the corresponding routing tables. The new routing tables allow BGP KeepAlive messages to suddenly start getting through again, and reachability information is exchanged via update messages.

BGP caused OSPF updates are measured when an eBGP router creates or installs a new route to a destination IP prefix. The AS External LSA created by the IGP domain is tagged as being caused by BGP and at every hop throughout the flood is measured as such. Not all AS External LSAs are caused by BGP. OSPF routers must exchange their entire LSA database when a link becomes available, and these LSAs must be flooded throughout a domain according to the OSPF RFC.

Because we are specifically interested in feature interactions between the OSPF and BGP protocols, our main response surface is defined as $BGP\ caused\ OSPF\ +\ OSPF\ caused\ BGP\ Updates$. Also, because the interactions are implicit in the models, specific code had to be added to the models to detect and mark updates as to their cause, and tracked throughout the system for quantification purposes.

5.2 Network Topology Stability

Recall that our network protocol models start in the converged state for each experiment generated by the optimization. In steady state, no control plane update messages are exchanged, other than periodic OSPF LSA refreshing. BGP does not require refreshing of the RIB. In order to generate update messages in the system, two types of network events were modeled: link status changes and link weight changes.

Link statuses are either *up or down* and occur with a uniform random probability over the simulation endtime. These events can model either link congestion in the data plane or actual link availability on a given timeline. The probability that a link status may change in the given simulation endtime is varied to model different levels of network topology stability. The stability levels are: 1%, 10% and %15 over runtime. While it was shown in [31] that some links fail far more frequently than others over a given interval, generalizing link failures uniformly allows us to investigate varying degrees of network topology stability. While the system has the capability of modeling individual links, creating a more "realistic" link failure model is beyond the scope of this investigation.

Link weight changes follow the same uniform random probability over the simulation endtime, but rather than act as up/down events, they affect the network by varying the metric on the links. Also, link weight change events are delivered directly to the affected OSPF routers and are modeled as network administration events which occur through either human contact or programmatically. Each router originating an LSA containing the affected link refreshes the LSAs containing the link in question. Each new link metric is chosen randomly over the ranges: $\pm$ 10, 25, and 50 units.

6 Design of Experiments

We present here three investigations with the goal of generally characterizing the system under test in variety of conditions. The first experiment design considers varying network management perspectives. These perspectives each attempt to minimize the response as related to either a global or local perspective. One example of a local perspective is optimizing the OSPF domain without considering the impact on the BGP domain. The global perspective implies all ISPs working together to reduce control plane traffic.

The second design investigates cold- versus hot-potato routing policies within an AS. This investigation focuses on the BGP attribute, MultiExitDiscriminator (MED) for cold-potato routing and the IGP hop count for hot-potato routing.

Design 3 analyzes the performance of protocol models under varying degrees of network stability and link weight management. Network stability is determined by the frequency and duration of link outages in the network.

For each experiment conducted, an efficient RRS search was performed for the given response value, and each RRS search generated 200 simulation samples. We then performed a multiple linear regression on the results of the RRS search. Please note that only $AdjustedR^2$ results are shown because experiments may have different input parameters. The $AdjustedR^2$ value indicates the degree to which the input parameters are related to the response. In each experiment the P value was always < 0.0001, indicating in each case that the regression model predicted the response in a statistically significant manner. In other words, in each experiment the predictions of the model are better than chance alone. In addition, the Degrees of Freedom are not reported per experiment. In each experiment the degrees of freedom was high, > 100. Finally, multi-collinearity was not observed to be a problem in

any of the experiments (i.e., all R^2 with other X values were < 0.75.

6.1 Input Parameter Classes

Table 3: Detail of parameter space for the large-scale OSPF and BGP experiment designs.

Input Parameter Classes	Min, Max, Step	Defaults
OSPFv2 Timer Class:		
OSPF Hello Interval	[1,4,1] secs	2
OSPF Inactivity Interval	[2,5,1] multiplier	4
OSPF Flood Interval	[1,4,1] secs	1
BGP4 Timer Class:		
BGP KeepAlive Interval	[25,35,2] secs	30
BGP Hold Interval	[36,56,4] secs	45
BGP Min Update Interval	[20,40,4] secs	30
BGP Policy Routing Class:		
MED	ON/OFF	ON
Hot Potato	ON/OFF	ON
BGP Decision Algorithm Class:		
Local-Pref	{low, med, high}	low
MED	{low, med, high}	low
AS-PATH Padding	{low, med, high}	low

The system under test can be characterized as different classes of input parameters. The four classes shown in Table 3 represent timers for OSPF and BGP, the BGP route selection policies and the BGP decision algorithm. Each class is defined at the AS level and the values generated are determined by the efficient search algorithm, RRS.

The BGP and OSPF timer classes represent router timers and the values they may have during each simulation run. The specified ranges and steps for each timer value determines the search sample space. The defaults shown are the values used per AS when a given class is not searchable within a given design.

The BGP Policy Routing Class allows hot- and cold-potato routing to be enabled/disabled within an AS. The ROSS.Net framework allows any of the stages in the BGP decision algorithm to be disabled, however these are the two of interest in this paper.

The BGP Decision Algorithm Class provides specific values for the AS routes. For example, if cold-potato routing is enabled within an AS, then the MED value is defined for routes created by that AS. In this paper we investigate cold-potato routing so must define MED values for those ASes where cold-potato routing in enabled. Values are *low, medium and high* and correspond to varying levels of aggressiveness within each AS. Recall that during the BGP decision algorithm, stage 1, we install the route with the higher Local-Pref value, so each AS must define this attribute for each route created. When these stages are enabled, but not searched by the experiment design, the default values are used.

When all of the input parameter classes are searched the sample space is greater than 14 million. Heuristic search algorithms such as RRS allow us to search this sample space efficiently, i.e., using a proportionally small number of experiments, while still achieving highly correlated results (high $AdjustedR^2$ values).

6.2 Experiment Design 1: Management Perspective

Our first investigation focuses on the role of network management perspectives in the response plane. We identify two disparate approaches to network management: local and global. The local approach involves performance tuning an AS domain without knowledge or concern for the impact on neighboring ASes, or even other protocols within the AS. The global approach attempts to optimize all of the ASes simultaneously and is semantic to optimal performance with respect to the internetwork as a whole. Here information about each neighboring AS is openly available and the optimization goal is across all ASes. BB, OO, OB, BO and BO+OB are considered local policies and the global policy is the addition of all update messages (BB+OO+OB+BO).

This design focuses on multiple response surfaces, as shown in Table 4 and optimizes across all input parameter classes. Each Experiment conducted generates a unique response plane corresponding to a network management perspective. For example, Experiment 1 generates a response plane where OSPF caused OSPF updates were minimized. The optimal response column indicates that of the 200 simulation runs, the minimum number of OO updates obtained was 27,424. The BO+OB column indicates the number of interactions that occurred between the OSPF and BGP protocols. A value of 59,429 indicates that minimizing OO updates does not greatly increase the number of updates between OSPF and BGP when compared to the other perspectives. The $AdjustedR^2$ value of 88% indicates that the search parameters highly correlated to the response, and the optimal values were 3 seconds for the OSPF HELLO timer, 4 seconds for the OSPF Flood timer, and 56 seconds for the BGP HOLD interval.

Table 5: Variation in the optimization of different perspectives. This table illustrates the tradeoffs made for each particular optimization. Bold values are optimization results.

Exp	ΣBB	ΣOO	ΣBO	ΣOB	$\Sigma BO + OB$	$\Sigma Global$
0	**1,938**	27,624	20	77,004	77,024	106,586
1	9,565	27,864	245	**52,700**	52,945	90,374
2	2,507	27,672	**18**	75,481	75,499	105,678
3	2,574	**27,424**	20	59,409	59,429	89,427
4	8,619	27,888	211	52,748	**52,959**	89,466
5	2,687	27,847	24	52,703	52,727	**83,261**

How efficient is each management perspective? Table 5 lists the results from each of the experiments. We see that the lowest number of interactions occurred in Experiment 1 where OSPF caused BGP updates were optimized. Because we were optimizing OB updates, and OB updates account for greater than 99% of BO+OB updates, this result make sense. Experiment 5 generated the least number of updates overall and was 7-27% better than the local perspectives. Not only does optimizing globally lower the number of overall updates, it also lowers the number of interactions between the protocols, within $<$ 1% of the best case. So it is clear that maintaining privacy between ISPs leads to an increase in the amount of update messages in the network.

Each row of the table represents the optimal value generated by the Experiment. Each column indicates the total number of each type of update message generated for those parameters. Experiments 1, 3 and 4 generated the least number of updates overall locally, and the least number of interactions. Each of these local policies where within 7% of global.

Of the different types of update messages, BB and BO were insignificant in respect to the global number of updates. Conversely, OO and BO were a significant fraction of all update messages, but the OO updates varied little. This leaves OSPF caused BGP (OB) update messages as the significant response to optimize when attempting to minimize both feature inter-

Table 4: Design 1: Search varying network management perspectives. The optimal response column relates to the specific management goal searched. The BO+OB column represents the interactions between protocols that occurred.

Design 1: Management Perspectives									
	Response Surfaces								
Experiment	BB	OO	OB	BO	**Optimal Response**	**BO+OB**	$Adj\ R^2$	**Effects: optimal values**	
0	+	-	-	-	1,938	77,024	0.30	Inactivity: 3	Keep: 26
1	-	+	-	-	27,424	59,429	0.88	Hello: 3	Flood: 4
2	-	-	+	-	52,700	52,945	0.88	Flood: 1	Keep: 34
3	-	-	-	+	18	75,499	0.18	MRAI: 34	
4	-	-	+	+	52,959	52,959	0.91	Keep:34	Hold: 45
5	+	+	+	+	83,261	52,727	0.52	Flood: 1	Keep: 34
	Sample Space Size: 14,348,907								*+ = searched*

actions and the overall number of update messages in the network.

Which protocol parameters effect the response? If we choose to minimize the number of updates and/or interactions in the network by minimizing OB updates, then Table 4 suggests settings for the OSPF Hello interval and Flood interval be set high. In our search, settings of 3 seconds for the Hello interval and 4 seconds for the Flood interval suggest that OSPF convergence times be lengthened in order to minimize overall updates. Generally, slow convergence is not a desirable feature in OSPF networks as it can lead to losses in the data plane. However, slower detection in OSPF may reduce the effects of highly unstable links.

An alternative is to optimize for one of the other local perspectives which prescribe aggressive OSPF convergence settings. In Experiments 2, 4 and 5 the important parameter appears to be the BGP KeepAlive timer. In each case, this timer is set to a high value. Since iBGP connections far outweigh eBGP connections, it makes sense then that by setting the KeepAlive timer to a high value would minimize the effects of highly unstable links in the path between iBGP neighbors.

6.3 Experiment Design 2: Cold vs Hot Potato Routing

Table 6: Design 2: analyze protocol performance under competing goals of Hot and Cold Potato routing. **Sample Space Size: 14,348,907**, **+ = searched**.

Design 2: Cold vs Hot Potato Routing				
	BGP Decision Classes			
Exp.	Hot Potato	MED	**Opt. Response**	$Adj\ R^2$
0	-	-	52,722	0.91
1	+	-	52,494	0.91
2	-	+	52,675	0.91
3	+	+	52,908	0.91

When two otherwise equal routes are being considered for addition to the BGP RIB, and those routes are both from iBGP peers, the route selected should be from the nearest peer. To determine which peer is the shortest distance away, the IGP hop count path is considered. This is the definition of Hot Potato routing, and was highlighted as a potential cause of many OSPF caused BGP updates in [13]. In that study it was noted that it was not possible to quantify the causes of the updates through measurement data. Also, protocol timer settings in routers throughout the network were not known. Simulation allows us to have a global view of the network, and complete topological information. Searching the sample space allows us to quantify the causes of the updates as well as determine the effects of any potentially influential protocol parameters.

Now that we have a validation that the OSPF domain adversely impacts the BGP domain, we can begin to focus our experiments on the hypothesized cause of the interruptions. In Table 6 we investigate the effects of cold versus hot potato routing. In this design we perform a simple full-factorial of RRS optimizations, turning Hot Potato routing on/off, and the MED on/off within the BGP decision algorithm.

If the goal of Hot Potato routing is to transit data through the network by the shortest paths possible, the goal of cold potato routing is the opposite. Cold potato routing is employed when end-to-end quality of service is of importance to an ISP. By carrying data longer in the network, an ISP can exert more control over the data before handing it off to another ISP. The MED accomplishes this goal by advertising to an AS the *preferred* routes data should take. *Preferred* is a term which is open to interpretation, but in this sense it implies "highest quality" ingress points to a neighboring AS [32]. An ISP implements cold potato routing by setting the MED parameter.

Table 7: This table illustrates the steps used in the BGP decision algorithm for route updates. Each entry illustrates how many times a particular step resulted in a tie-breaking event.

BGP Decision Algo	**Hot Potato**	**MED**	**Neither**	**Both**
Local-Pref	**6383**	**1,714**	**767**	**885**
AS Path	**15,251**	**5,503**	**2,240**	**5,874**
Origin	1	8	50	204
MED	OFF	4	OFF	0
Hot Potato	199	OFF	OFF	1,229
Next Hop	123	369	175	113
Default	476	778	272	635
Total	22,433	8,376	3504	12,444
% Hot Potato	0.8	-	-	9
% MED	-	$\ll 1$	-	0

Which steps in the BGP decision algorithm are most important? Table 7 quantifies the tie-breaking steps in the BGP decision making algorithm. We expected MED and Hot Potato to play a larger role in the algorithm, based on previous work [13, 12]. In our model it appears that Local-Pref and AS Path Padding play a much larger role in the decision process. In practice, these parameters may not be implemented in some or all ISP networks. Clearly, these parameters do play an important role in dampening the effects of both Hot and Cold Potato routing.

While our statistical models show a high correlation between the input parameters and the response ($AdjustedR^2$ = 91%), we believe that this design is only an initial step towards

systematic questioning of the BGP decision algorithm. For example, when hot-potato routing only is enabled, the number of times the AS Path length was the tie-breaker increased from about 2,000 to over 15,000. Clearly, hot-potato routing is generating longer AS Path lengths in the routes. But it is unclear why there would be a corresponding 10-fold increase in the number of times the Local Pref tie-breaker was used. When just cold-potato routing was employed, these tie-breakers only doubled, which indicates that cold-potato routing has the same problem, but to much less a degree. More importantly, Table 7 indicates that when both policies are enabled, cold-potato routing can dampen the negative effects of hot-potato routing.

We did not expect these policies to have such a large effect on the other stages in the BGP decision algorithm. More insight into these results may be gained by future designs which takes this into account.

6.4 Experiment Design 3: Network Robustness

Table 8 illustrates our third design. The purpose of this experiment design is to ascertain the effects of network robustness on our characterization of the system under test. Network robustness is varied in two dimensions: link stability and link weight changes. Link stability was varied randomly over the intervals 1, 10 or 15% and link weights randomly over the intervals 10, 25 or 50 units. The design computes a full factorial over the two parameters of network robustness.

Which parameters were most important in reducing interactions? We report that the liveness timers are the important parameter settings and are related to minimizing OSPF caused BGP updates (OB). KeepAlive is maximized in BGP, and the InactivityInterval is maximized in OSPF. In OSPF, the flood timer, when important is optimized to a value of 2 leading to slower convergence. As the network becomes less stable however, we begin to see that other parameters are having more of an impact on the response. In OSPF we begin to see the Hello frequency becoming more important, and maximized. This is interesting because delaying detection allows OSPF to aggregate (implicitly) more changes into a single LSA update, which would act to minimize the overall number of updates generated. This implicit aggregation is occurring in the BGP domain as well by setting the KeepAlive interval to 34 seconds and the Hold Interval to 45-55 seconds. By detecting fewer link status changes the models are generating fewer control plane update messages.

Table 9: Improvements over average BO+OB, Global and Defaults in design 3.

LW/LS	Optimal	Avg BO+OB	Avg Global	Defaults
±10,1%	50,450	17%	19%	18%
±10/10%	73,196	17%	8%	
±10/15%	99,564	36%	34%	
±25/1%	54,254	10%	13%	18%
±25/10%	75,819	17%	17%	
±25/15%	100,493	22%	21%	
±50/1%	52,959	19%	14%	20%
±50/10%	76,346	17%	19%	
±50/15%	110,009	18%	17%	

Table 9 shows that we continue to receive consistent improvements in the response over the average regardless of the robustness in the network. We see that the optimal simulation experiments are simply setting the link failure detection parameters in either protocol to their slowest convergence settings. By not detecting link status changes quickly, the number of updates generated can most effectively be minimized. The table compares the amount of improvement over the average cases of BO+OB and the global response, as well as over the default settings. Generally, this approach to minimizing updates yielded a 20% improvement over the average. This figure is primarily related to the intervals chosen for the protocol parameters. In the future we could relate the improvements to the rate of convergence, which would be a more meaningful representation of the trade-off.

From the table we also see that the response is independent of the link weight changes. Each link weight interval varies by $< 5\%$ for each fixed link stability interval. This is surprising since aggressive link weight policies are known to produce routing loops among other problems. While aggressive changes impact the OSPF domain internally, those updates do not appear to be propagating into the BGP domain via OSPF. We theorize that the link status changes have a much greater impact on the OB response because they have a direct impact on the iBGP connections which dominate the model.

7 Conclusions

In this paper, we have used the design of experiments tool in ROSS.Net [4, 33] to characterize OSPF and BGP behavior in combination as well as their interactions. Based on the Rocketfuel data repository, we have developed a "more realistic" large-scale simulation of these two dominant inter- and intra-domain routing protocols. We then employed an efficient heuristic search algorithm, RRS, to search for best protocol optimizations and parameter settings. The protocol parameters we investigated included OSPF timers, BGP timers and BGP decision algorithm attributes. We defined the number of routing updates as the metric to minimize in our heuristic search for the best parameter settings. We also classified the routing updates into four categories to help design our experiments more flexibly.

We found that in order to minimize the interactions between BGP and OSPF the OSPF caused BGP updates should be optimized, as they account for the largest percentage of overall updates in the system and are the best candidate for minimization. In our second design we were able to verify past results which showed that hot-potato routing does in fact have an impact on the control plane, however we have quantifiably shown the AS PATH padding and Local Preference route attributes to have a greater impact. In our final design we found that link status changes propagated heavily from the OSPF domain into the iBGP domain, and that the effects of link weight changes were relatively insignificant in comparison.

References

[1] A. Papachristodoulou, L. Li, and J. C. Doyle, "Methodological frameworks for large-scale network analysis and design," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 3, pp. 7–20, October 2004.

[2] M. Liljenstam, et. al., "Rinse: the real-time interactive network simulation environment for network security exercises," in *Proceedings of the 19th Workshop on Parallel and Distributed Simulation*, June 2005, pp. 119 – 128.

[3] A. Park, R. Fujimoto, and K. Perumalla, "Conservative synchronization of large-scale network simulations," in *Proceedings of the Workshop on Parallel and Distributed Simulation*, May 2004, pp. 153 – 161.

Table 8: Design 3: Analyze performance of protocol models under varying degrees of network stability and link weight management.

Design 3: Network Robustness							
	Topology Parameters						
Experiment	Link Stability (LS)	Link Weight (LW)	**Optimal Response**	$Adj\ R^2$	**Effects: optimal values**		
0	1%	$\pm$ 10	50,450	0.89	Keep: 34		
1	1%	$\pm$ 25	54,254	0.91	Keep: 32	Hold: 46	MRAI: 33
2	1%	$\pm$ 50	52,959	0.91	Keep: 34	Hold: 45	
3	10%	$\pm$ 10	73,196	0.87	Keep: 34	Hold: 39	Inactivity: 4
4	10%	$\pm$ 25	75,819	0.88	Keep: 34	Hello: 4	Inactivity: 5
5	10%	$\pm$ 50	76,346	0.87	Keep: 34	Hold: 55	
6	15%	$\pm$ 10	99,564	0.78	Keep: 35	Flood: 2	MRAI: 28
7	15%	$\pm$ 25	100,493	0.78	Keep: 34		
8	15%	$\pm$ 50	110,009	0.900	Keep: 34	Hello: 4	
Sample Space Size: 14,348,907							

[4] D. Bauer, et. al., "A case study of meta-simulation and performance analysis of large-scale networks," in *Proceedings of Winter Simulation Conference*, 2004.

[5] U. of Washington, "Rocketfuel internet topology database," 2002.

[6] J. Cowie, A. Ogielski, B. J. Premore, and Y. Yuan, "Global routing instabilities triggered by code red ii and nimda worm attacks," Renesys Corporation, Tech. Rep., 2001.

[7] L. Wang, et. al., "Observation and analysis of bgp behavior under stress," in *Proceedings of ACM SIGCOMM Workshop on Internet Measurement*, 2002.

[8] T. G. Griffin, F. B. Shepherd, and G. Wilfong, "The stable paths problem and interdomain routing," *IEEE/ACM Transactions on Networking*, vol. 10, no. 2, pp. 232–243, April 2002.

[9] T. G. Griffin and G. Wilfong, "A safe path vector protocol," in *Proceedings of INFOCOM*, 2000.

[10] R. Teixeira, A. Shaikh, T. Griffin, and G. M. Voelker, "Network sensitivity to hot-potato disruptions," in *Proceedings of SIGCOMM*, 2004.

[11] T. G. Griffin and G. Wilfong, "Analysis of the med oscillation problem in bgp," in *Proceedings of ICNP*, 2002.

[12] R. Agarwal, C. N. Chuah, S. Bhattacharyya, and C. Diot, "The impact of bgp dynamics on intra-domain traffic," in *Proceedings of SIGMETRICS*, 2004.

[13] R. Teixeira, A. Shaikh, T. Griffin, and J. Rexford, "Dynamics of hot-potato routing in ip networks," in *Proceedings of SIGMETRICS*, 2004.

[14] A. Basu and J. G. Riecke, "Stability issues in ospf routing," in *Proceedings of SIGCOMM*, 2001.

[15] D. Obradovic, "Real-time model and convergence time of bgp," in *Proceedings of INFOCOMM*, 2002.

[16] A. Shaikh and A. Greenberg, "Experience in black-box ospf measurement," in *Proceedings of the SIGCOMM Internet Measurement Workshop*, 2001.

[17] A. Shaikh, R. Dube, and A. Varma, "Avoiding instability during graceful shutdown of ospf," in *Proceedings of INFOCOMM*, 2002.

[18] T. W. Chim and K. L. Yeung, "Time-efficient algorithms for bgp route configuration," in *Proceedings of the IEEE International Conference on Communications*, 2004, pp. 1197–1201.

[19] R. Mahajan, D. Wetherall, and T. Anderson, "Understanding bgp misconfiguration," in *Proceedings of SIGCOMM*, 2002.

[20] O. Nordstrom and C. Dovrolis, "Beware of bgp attacks," in *ACM Computer Communications Review*, 2004.

[21] J. Feigenbaum, C. Papadimitrious, R. Sami, and S. Shenker, "A bgp-based mechanism for lowest-cost routing," in *Proceedings of the ACM Sysposium on Principles of Distributed Computing*, 2002.

[22] L. Xiao, K.-S. Liu, J. Wang, and K. Nahrstedt, "Qos extension to bgp," in *Proceedings of the Internation Conference Network Protocols*, 2002.

[23] D. Nicol and J. Liu, "Composite synchronization in parallel discrete-event simulation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 5, pp. 433–446, May 2001.

[24] R. Fujimoto and M. Hybinette, "Computing global virtual time in shared-memory multiprocessors," *ACM Transactions on Modeling and Computer Simulation*, vol. 7, no. 4, pp. 425–446, 1997.

[25] J. Cowie, et. al., "Towards realistic million-node internet simulations," in *Proceedings of the 1999 International Conference on Parallel and Distributed Processing Techniques and Applications*, 1999.

[26] R. Fujimoto, et. al., "Large-scale network simulation – how big? how fast?" in *IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer Telecommunication Systems*, 2003.

[27] D. M. Nicol and G. Yan, "Simulation of network traffic at course time-scales," in *Proceedings of the 2005 Workshop on Principles of Advanced and Distributed Simulation*, 2005.

[28] T. Ye and S. Kalyanaraman, "A recursive random search algorithm for large-scale network parameter configuration," in *Proceedings of SIGMETRICS*, 2003, pp. 196–205.

[29] Y. Rekhter and T. Li, "A border gateway protocol 4 (bgp-4)," IETF, Tech. Rep. RFC 1771, March 1995.

[30] R. Teixeira, K. Marzullo, S. Savage, and G. M. Voelker, "In search of path diversity in ISP networks," in *Proceedings of IMC*, 2003.

[31] A. Markopoulou, et. al., "Characterization of failures in an ip backbone network," in *Proceedings of IEEE INFOCOM 2004*, March 2004.

[32] Cisco Systems Inc., "How the bgp deterministic-med command differs from the bgp always-compare-med command," March 2005, document ID: 16046.

[33] G. Yaun, et. al., "Large scale network simulation techniques: Examples of tcp and ospf models," in *ACM SIGCOMM Computer Communication Review*, 2003.