*Proceedings of the 2004 Winter Simulation Conference*
*R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds.*

# A CASE STUDY IN META-SIMULATION DESIGN AND PERFORMANCE ANALYSIS FOR LARGE-SCALE NETWORKS

David Bauer
Garrett Yaun
Christopher D. Carothers

110 8th Street, Lally Building
Dept. of Computer Science
Rensselaer Polytechnic Institute
Troy, NY 12180, U.S.A.

Murat Yuksel
Shivkumar Kalyanaraman

110 8th Street, JEC Building
Dept. of Electrical, Computer, and Systems Engineering
Rensselaer Polytechnic Institute
Troy, NY 12180, U.S.A.

## ABSTRACT

Simulation and emulation techniques are fundamental to aid the process of large-scale protocol design and network operations. However, the results from these techniques are often view with a great deal of skepticism from the networking community. Criticisms come in two flavors: (i) the study presents isolated and potentially random feature interactions, and (ii) the parameters used in the study may not be representative of real-world conditions. In this paper, we explore both issues by applying large-scale experiment design and black-box optimization techniques to analyze convergence of network routes in the Open Shortest Path First protocol over a realistic network topology. By using these techniques, we show that: (i) the needed number of simulation experiments can be reduced by an order of magnitude compared to traditional full-factorial experiment design (FFED) approach, (ii) unnecessary parameters can easily be eliminated, and (iii) rapid understanding of key parameter interactions can be achieved.

## 1 INTRODUCTION

Performance analysis techniques are fundamental to the process of protocol design and network operations (Jain 1991, Floyd and Paxson 2001, Floyd 2001). The high-level motivation of these techniques is simple: to gain varying degrees of qualitative and quantitative understanding of the behavior of a system under-test. Systematic design-of-experiments (Jain 1991, Montgomery 2001) is a well studied area of statistics and performance analysis offering guidance in this aspect. Such systematic techniques (e.g.: factorial designs, large-scale search) have not been used in the protocol design process or network operations process except possibly by measurement specialists. This ad-hoc approach to organizing simulation or testbed experiments has worked when we design and examine a small number of features, network scenarios and parameter settings. However, this method is likely to be untenable as we design newer protocols that will rapidly be deployed on a large-scale, or have to deal with a combinatorial explosion of feature interactions in large operational inter-networks. This point has also been made in Floyd and Paxson (2001), where *scale, heterogeneity and rapid change* are listed as the the three reasons why it is hard to simulate large networks. The need for scalable simulation and meta-simulation tools is implicit in Floyd (2001): *"...we can't simulate networks of that size (global Internet). And even if we could scale, we would not have the proper tools to interpret the results effectively..."*

Beyond mere scaling of simulation platforms, our next need is meta-simulation capabilities, i.e. large-scale experiment design. Statistical experiment design considers the system-under-test as a black-box that *transforms input parameters to output metrics*, and aims to *characterize* and *optimize* the black-box system. While regression models for small dimensional parameter spaces can be built using simple factorial methods (Jain 1991, Montgomery 2001), these methods do not ramp up to large-scale situations. As a result, we replace detailed regression-like characterization with heuristic search methods. Many heuristic search algorithms have been proposed such as multi-start hill-climbing (Törn and Žilinskas 1989), genetic algorithms (Goldberg 1989) and simulated annealing (Aarts and Korst 1989). While these techniques tend toward the global optima in the limit, they do not have the property of finding good results quickly. We have recently proposed an efficient search algorithm, Recursive Random Search (RRS) (Ye and Kalyanaraman 2003), for efficient large-dimensional heuristic optimization, which has yielded very positive results in finding a "good" minima with few iterations.

The key focus here is a case study in the application of this meta-simulation technique to examine OSPFv2 convergence times during network link failures. This study includes OSPF optimizations for sub-second convergence, adapted from Alaettinoglu, Jacobson, and Yu (2000). Here, *convergence* is defined to be time at which *all* routers in the network have a synchronized routing table or put another way, a consistent view of the routing tables is shared by all routers. We explore the cases , i.e. large-scale experiment design and black-box optimization (i.e. large-dimensional parameter state space search) using realistic topologies with bandwidth and delay metrics to analyze convergence of network route paths in the Open Shortest Path First (OSPFv2) protocol.

By using Recursive Random Search (RRS) approach to design of experiments, we find: (i) that the number of simulation experiments that must be run is reduced by an order of magnitude when compared to full-factorial experiment design (FFED) approach, (ii) it allowed the elimination of unnecessary parameters, and (iii) it enabled the rapid understanding of key parameter interactions. From this design of experiment approach, we were able to abstract away large portions of the OSPF model that result in a 100 fold improvement in simulation execution time.

In the next section we describe the term meta-simulation and it's relation to design of experiments. Then in Section 3, we present the OSPFv2 model, and the environment in which we generated our results. In Section 4, we explain how Recursive Random Search allows us to generate more detailed results with fewer experiments. In the final sections, we detail the results of our experiment designs and what we have learned from them.

## 2 META-SIMULATION: LARGE-SCALE EXPERIMENT DESIGN AND ANALYSIS

### 2.1 Overview of FFED

Design of Experiments or "experiment design" is a well known branch of performance analysis, specifically, a sub branch of statistics. It has been used extensively in areas such as agriculture, industrial process design and quality control (Montgomery 2001) , and has been introduced to the area of practical computer and network systems design by Jain (1991). Statistical experiment design views the system-under-test as a black-box that transforms input parameters to output metrics. The goal of experiment design is to maximally characterize (i.e. obtain maximum information about) the black-box with the minimum number of experiments. Another goal is robust characterization, i.e., one that is minimally affected by external sources of variability and uncontrollable parameters, and can be specified at a level of confidence.

The underlying premise of experiment design is that each experiment has a non-negligible cost. Simple designs like "best-guess" or "one-factor-at-a-time" designs are less favored in complex situations since they do not provide information about the interactions between parameters. Designs like full-factorial and fractional factorial (also called orthogonal designs), appropriately subjected to replication, randomization and blocking are preferred. The usual end-goal of formulating regression models is to observe the effects of both individual parameters and parameter interactions. Techniques like blocking and analysis of covariance are used to explicitly handle measurable, but uncontrollable (a.k.a. "nuisance") factors. Transforms on data (e.g., Box-Cox power-law family of transformations) can effectively aid in producing a family of non-linear regression models and stabilizing the variance of the response (Jain 1991, Montgomery 2001).

The next step beyond characterization (i.e. developing input-output regression models) is optimization, i.e. to determine the region in the important factors that leads to best-possible response. The output (i.e. response) in general will have an unknown surface topology, also known as "response surface". The approach typically used involves quickly traversing the surface sequentially (by using lower-order models built with fractional factorial experiments) to reach interesting areas where more detailed (higher-order) characterization is done.

As well known, one of the significant drawbacks of the FFED approach is the exponential increase in the number of experiments that must be run as a function of the number of data points per parameter. To vastly reduce the number, heuristic search algorithms must be used, such as RRS.

### 2.2 Overview of The RRS Algorithm

The key idea behind RRS is to maintain the initial efficiency of random sampling by "restarting" it before its efficiency becomes low. However, unlike the other methods, such as hill climbing, random sampling cannot be restarted by simply selecting a new starting point. Instead we accomplish the "restart" of random sampling by *changing its sample space*. We perform random sampling for a number of times, then move or resize the sample space according to the previous samples and start another random sampling in the new sample space. Given a black-box objective function, a desired optimization process should start with inspecting macroscopic features of the objective function, and then look further into microscopic features in selected promising areas. The search process of RRS algorithm is fully consistent with this idea. In the beginning of the search, RRS performs sampling from the whole parameter space and thus examines the overall structure of the objective function. With the search continuing and the sample space gradually shrinking, the search gets more and more details of the objective function until it finally converges to a local optimum.

A stochastic search algorithm usually comprises two elements: *exploration* and *exploitation*. Exploration examines the macroscopic features of the objective function and aims to identify promising areas in the parameter space, while exploitation focuses on the microscopic features and attempts to exploit local information to improve the solution quickly. Various search techniques can be used for these two purposes. The RRS algorithm uses random sampling for exploration and recursive random sampling for exploitation. Ideally it should only execute the exploitation procedure in promising areas. However, it is difficult to determine which areas are more promising and should be exploited. Many algorithms, such as multi-start, do not differentiate areas and hence may waste time in trivial areas. For the algorithmic details and implementation of RRS we refer the interested reader to Ye and Kalyanaraman (2003).

## 3 DESIGN OF EXPERIMENTS FOR OSPFv2 CONVERGENCE

The goal of our design of experiments was to understand the factors determining the amount of wall-clock time required for a network of routers to detect and propagate a link state failure. Our experiment is a simulation of a network of Internet routers all operating the OSPFv2 protocol as described in Moy (1998). For an example router network, we selected the AT&T network, as described by Rocketfuel data (Rocketfuel 2002). This network description was determined by using various network probing techniques (i.e, traceroute). The AT&T network is challenging because of it's size and complexity.

### 3.1 OSPFv2

OSPFv2 is a link-state routing protocol designed to be run internal to a single Autonomous System. Each OSPFv2 router maintains an identical database describing the internal network's topology (i.e. an Autonomous System (AS)). 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. OSPFv2 is classified as an Interior Gateway Protocol (IGP). This means that it distributes routing information between routers belonging to a single Autonomous System. An example of an Autonomous System is the AT&T network, which is AS number 7018. Routing between ASs is handled by an external protocol, such as Border Gateway Protocol (BGP) (Stewart 1999).

The OSPFv2 protocol is based on link-state or shortest-path-first (SPF) technology. In a link-state routing protocol, each router maintains a database describing the Autonomous System's topology. This database is referred to as the link-state database. Each participating router has an identical database. Each individual piece of this database is a particular router's local state (e.g., the router's usable interfaces and reachable neighbors). The router distributes its local state throughout the Autonomous System via flooding. All routers run the exact same algorithm, in parallel. From the link-state database, each router constructs a tree of shortest paths with itself as the root. This shortest-path tree gives the route to each destination in the Autonomous System (Moy 1998). OSPFv2 routers employ the HELLO protocol for establishing and maintaining communications with adjacent routers. Adjacencies are established between two routers when a HELLO protocol packet is received by one of the two routers connected by a link. HELLO packets are then sent at regular intervals between adjacent routers. Upon receiving a HELLO packet from a neighboring router, an inactivity timer is set for that router. If another HELLO packet is not received from that router before the timer expires, then the adjacency is broken and that router should no longer be used to route IP packets.

All of these aspects are modeled. However, multiple areas within a single OSPF domain is not currently modeled. In the experiments presented here, we configure OSPF to be a single large area. This was done because there is an interest in determining where OSPF ceases to execute in an efficient manner. This was a sub-goal of our experimentation.

### 3.2 AT&T Network Topology

For our network topology we selected the AT&T network, which contains 11964 router nodes and 7491 links. Internet topologies like the AT&T network are interesting from a modeling prospective because of their sparseness and power-law structure (Rocketfuel 2002). This structure allows for a greater range of convergence times compared to fully connected networks. The OSPFv2 update packets require multiple hops in order to reach the outer edges of the network.

In performing a breadth-first-search of the AT&T topology, there are eight distinct levels. A number of routers were not directly reachable and thus were removed. Those routers are likely connected by transit routes. In total there are 3371 backbone routers and at the successive levels there are 8593 routers. The 4 ms delay that was chosen for the backbone core routers was in-line with the delays that Rocketfuel had associated with the Telstra topology backbone. An order of magnitude higher delay was selected for all lower level routers.

The bandwidth and delay for the AT&T topology is as follows:

- **Levels 0 and 1 routers:** 155 Mb/sec and 4 ms delay
- **Levels 2 and 3 routers:** 45 Mb/sec and 4 ms delay
- **Levels 4 and 5 routers:** 1.5 Mb/sec and 10 ms delay

- **Levels 6 and 7 routers:** 0.5 Mb/sec and 10 ms delay.

Our experiments focused on the convergence time metric. We defined convergence to be the time at which all routers on the network have received an update corresponding to a link status change, and have recomputed their forwarding tables. In order to clearly state convergence intervals, in our simulations we have only a single link state failure per simulation and all of the OSPFv2 routers were started in a converged state. We defined the input plane to this experiment design to be composed of the HELLO Interval (HELLO-Int), HELLO Inactivity Timer (HELLO-IT), Shortest Path First Computation Interval (SPF-CI), Acknowledgment Timer (ACK) and Maximum Transmission Unit (MTU). The response plane is the convergence time from the link state failure.

The goal for our design of experiments was to adapt some of the convergence optimizations in Alaettinoglu, Jacobson, and Yu (2000) for the IS-IS protocol to the OSPFv2 RFC (Moy 1998) protocol. IS-IS is a link state protocol for Cisco routers. Suggestions for lowering convergence times were: to queue HELLO packets in front of data packets, use a modern shortest-path-first (SPF) algorithm, and to give a higher priority to link state packet (LSP) propagation over SPF computation. We took the following steps to adapt the optimizations.

We did not model the data plane in our OSPFv2 routers so that HELLO packets would always be at the front of the queue. It is still possible for other control plane packets to queue in front of the HELLO packets. To facilitate a higher priority for link state propagation over SPF computation, we remove the LSP propagation timer from the OSPFv2 protocol. Now, LSP propagation will always occur immediately, and the SPF computations will always occur later. In addition, modern SPF computations would only add a small amount of time to overall convergence interval. We modeled this by adding in the amount of time stated by Alaettinoglu, Jacobson, and Yu (2000) for a topology of our size.

## 4 RRS RESULTS

As previously discussed, Recursive Random Search (RRS) is a heuristic search algorithm for black-box optimization problems. This algorithm is specifically designed to optimize dynamic network protocol parameterizations with an emphasis on obtaining “good” solutions within a limited time frame. RRS does not attempt to find a full optimization of the parameter space. The RRS algorithm maintains the high efficiency property of random sampling by constantly restarting random sampling but with adjusted parameter spaces. Because it shares this property with random sampling, it is also highly robust to the effect of random noises in the objective function. It also performs efficiently when handling an objective function that contains negligible parameters.

Table 1: Input Plane Parameters for RRS Experiments

| **Parameters** | **Unit** | **Minimum** | **Maximum** |
|---|---|---|---|
| HELLO-Int | seconds | 0.5 | 10.0 |
| HELLO-IT | seconds | 1.5 | 8.0 |
| ACK | seconds | 0.5 | 10.0 |
| MTU | bytes | 500 | 1500 |
| SPF-CI | seconds | 0.5 | 10.0 |

Table 2: RRS Linear Regression Model

Coefficients

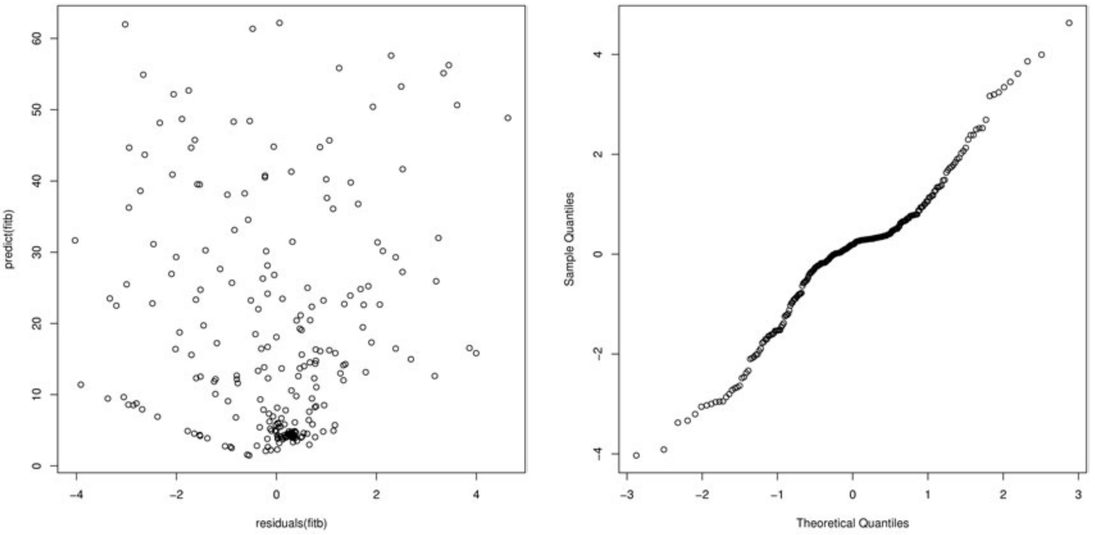
| | **Estimate** | **t value** | $Pr(>\|t\|)$ |
|---|---|---|---|
| (Intercept) | -19.838684 | -12.816 | $< 2e-16$ |
| HELLO-Int | 4.426648 | 36.415 | $< 2e-16$ |
| HELLO-IT | 4.507337 | 26.623 | $< 2e-16$ |
| ACK | 0.089194 | 0.788 | 0.432 |
| MTU | -0.001568 | -1.368 | 0.173 |
| SPF-CI | 0.717926 | 5.909 | 1.16e-08 |

Residuals

| Min | 1Q | Median | 3Q | Max |
|---|---|---|---|---|
| -15.7286 | -1.0311 | 0.1805 | 1.4811 | 11.5511 |

As shown in Table 1, we chose a wide range of input parameters for the RRS algorithm: HELLO-Int, HELLO-IT, ACK, MTU, SPF-CI. We allowed RRS to search for 250 experiment runs, specifying a desired confidence level of 99%. RRS generated a convergence minimum after only 7 executions of 4.07 seconds. We fitted a linear regression model to our data using a tool called R (2004), and generated the co-efficients shown in Table 2. After analyzing the variance on the inputs, we found the parameters that had the greatest impact on the model to be the HELLO-Int, HELLO-IT, and the SPF-CI.

After considering the simulation model, we realized that the HELLO polling interval is set to be HELLO-Int multiplied by HELLO-IT. These two parameters have an impact on convergence because they determine the time to detect a link state failure. The other factor of convergence time is the time to propagate the link state failure to the remainder of the routers in the network. The update propagation time is defined by flooding packets throughout the network. The router which detects the failure informs all remaining neighbors, who notify their neighbors, and so on, until eventually all routers in the network have received the update. The propagation delay on the updates is bounded by the amount of time it takes for the update to travel across the diameter of the network. Recall also from our definition



(a) Scatter plot.

(b) Q-Q plot.

Figure 1: RRS Linear Regression Model

Table 3: Re-parameterized RRS linear Regression Model

Coefficients

| | **Estimate** | **t value** | $Pr(>\|t\|)$ |
|---|---|---|---|
| (Intercept) | -0.31820 | -0.581 | 0.5617 |
| HELLO-Int (1) | -0.19646 | -1.750 | 0.0814 |
| HELLO-IT (2) | 0.13986 | 1.047 | 0.2962 |
| SPF-CI | 1.0743 | 5.055 | 5.27e-06 |
| (1) × (2) | 0.92009 | 37.773 | $< 2e-16$ |

Residuals

| Min | 1Q | Median | 3Q | Max |
|---|---|---|---|---|
| -18.6508 | -0.6173 | 0.23916 | 0.5768 | 5.3645 |

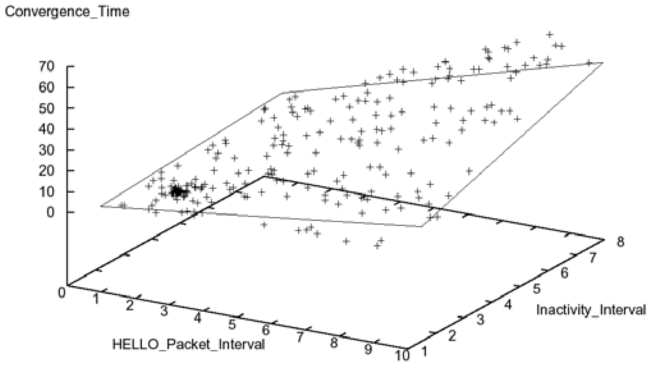


Figure 2: Re-parameterized RRS: Demonstration of the Major Role Played by the HELLO Interval and HELLO Inactivity Timer

of convergence that each router must have also recomputed their forwarding tables. So the convergence time is compounded by either how long it takes for the final router to update it's table, or by the longest SPF computation interval.

After analyzing the variance we fitted a new linear regression model to SPF-CI and the cross between HELLO-Int and HELLO-IT, as shown in Table 3. This regression produced an adjusted R-squared value of 98%. In order to verify the accuracy and correctness in our model we created a scatter plot of the errors versus predicted responses. The scatter plot did not show any trends in the data. The next step in our verification was to create a quantile-quantile plot of the residual errors. We observe a linear relationship between sample error and theoretical. From Figure 1, the linear model assumptions of normality appear to be valid.

In order to gain more detail about the interesting parts of the design, we re-executed the model a second time with only those input parameters. Specifically, we used only HELLO-Int, and HELLO-IT in the same ranges as shown in Table 1. Again, we allowed RRS to search for 250 iterations and with a confidence interval of 99%. This experiment generated a convergence minimum after only 129 executions of 0.93 seconds.

In this series of experiments, a sub-second range for the convergence interval is observed. Figure 2 shows that HELLO-Int and HELLO-IT form a plane with one corner tilting downward toward the smaller values. This low corner is anchored by the best convergence result given by RRS. We report a clustering effect occurring on the graph, which is attributed to the RRS algorithm centering upon a given input. It appears that RRS was successful in isolating HELLO-Int and HELLO-IT at the low end of their ranges.

The initial goal of our design was to determine if we could adapt some of the ideas in Alaettinoglu, Jacobson, and Yu (2000) to the OSPFv2 protocol and achieve convergence times on the order of magnitude in the tens of milliseconds.

Table 4: Re-scaled RRS input plane parameters.

| Parameters | Minimum | Maximum |
|---|---|---|
| HELLO-Int | 0.03 | 1.0 |
| HELLO-IT | 1.5 | 2.0 |

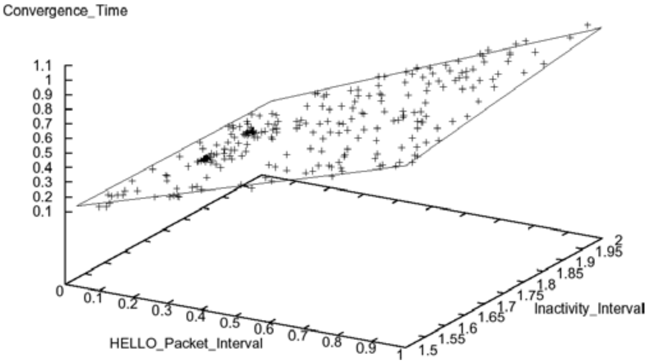


Figure 3: Re-scaled RRS: Re-run of RRS on the Specific Region of Interest Where the Convergence Time is Lower.

Having isolated the effective parameters of the simulation model, we being to see points in the sub-second range. So we re-scaled the experiment into the range of the input values that generated those results, and noticed that HELLO-Int was in the range suggested by Alaettinoglu, Jacobson, and Yu (2000). Table 4 and Figure 3 show the results of this experiment. All of the convergence times are below a second, and the best values are in range of tens of milli-seconds.

## 5 OSPF MODEL CRITICAL PATH ANALYSIS

Berry and Jefferson (1985), developed a technique called *Critical Path Analysis* to determine the optimal parallel simulation execution time. Armed with the results from RRS, we apply this technique here to examine what the critical path for OSPF convergence within the context of our model.

After examining the results from our designs, we observed that the two main components of convergence are *detection* and *propagation*. Detection is simply the amount of time that elapses between a link state change and the time at which the inactivity timer fires. The second component, propagation is determined by the longest path the link state update travels through the network. The longest path is not immediately determined by the number of hops between the originating router and the final router to receive the update. It is possible for an update to take many hops over high-speed links and still not be on the longest path. Conversely, it is possible to take only a small number of hops over very low speed links and be on the longest path. Realizing that these two factors have the highest impact on convergence time, we observed that to accurately model convergence in any network, only the set of nodes which encompass the longest path through the network require simulation. This observation has been used on other OSPF optimizations (Goyal, Ramakrishnan, and Feng 2003).

We simulated the AT&T network which contained almost 12,000 routers. The model was instrumented so that each router would keep track of which routers they received link state updates from. Once the update reached the final router in the network, we then backtracked this path to find the longest path in the network.

In order to validate the modeling optimization, we analyzed the OSPF convergence on the VSNL (India) topology by applying FFED. This topology contained only 291 routers, which allowed us to compare the optimization to the full model simulation results. We tried all combinations possible within the range of each parameter, and the optimization results generated exactly the same output for the optimization as we would have received had we modeled the entire topology.

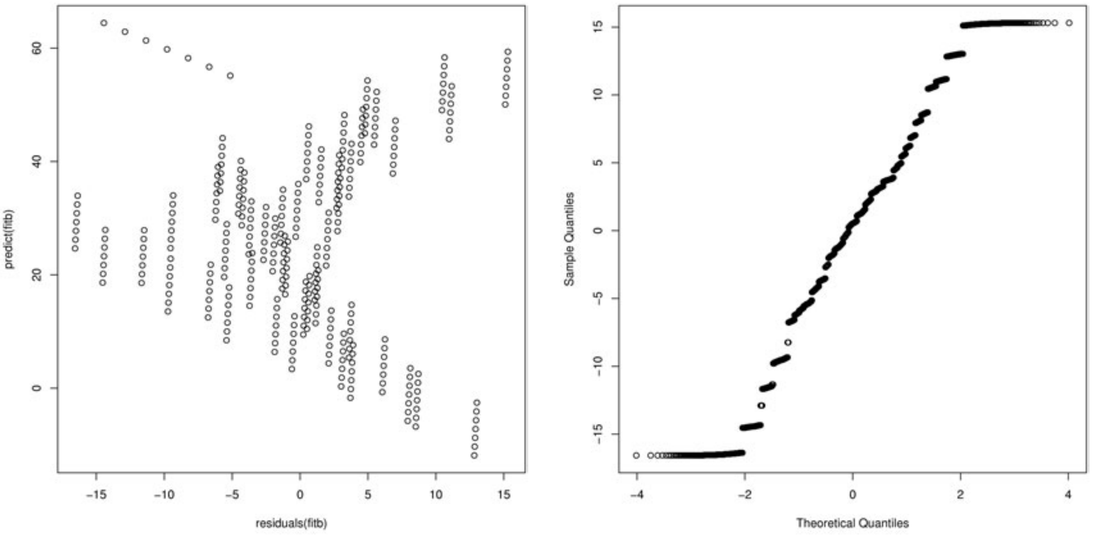
Simulations on the full AT&T network required anywhere from one half hour to initialize to several hours depending on whether the routing tables need o be computed. In addition to the time required to initialize the simulation, execution time took on average one second of wall clock time to simulate one second of simulated time. In other words, to simulate an hour of OSPFv2 traffic required almost one hour of real time.

After determining this optimization, we computed the longest paths through the AT&T network for each of the updates generated from a single link failure. These paths we 11 and 12 hops long respectively, and the paths only varied at a single node. This means that in order to simulate the entire AT&T network for convergence times only required actually simulating 13 total routers. Obviously, this reduced the time required to run the simulation to the order of seconds. At this stage, the simulation requires a second or so to initialize, and on an average execution was complete in 0.0006 seconds. The simulation results we presented in this paper required 12,000 events to generate the convergence times in a simulation of 100 seconds. Using this optimized model, we are now able to compare FFED approach to RRS.

## 6 COMPARISON OF RRS TO FFED

In the previous section we showed how meta-simulation can reduce the amount of time to acquire meaningful results from our models by employing algorithms such as Recursive Random Search. Using RRS, we were able to generate all of our results in only 750 experiments, or simulation executions. However, RRS had not sampled a large area of the state space, so how confident can we be in the results?

We applied FFED model in order to validate the results we gained. We used the same five input parameters in



(a) Scatter plot.

(b) Q-Q plot.

Figure 4: FFED Linear Regression Model

Table 5: FFED Linear Regression Model

Coefficients

| | **Estimate** | **t value** | $Pr(>\|t\|)$ |
|---|---|---|---|
| (Intercept) | -2.132e+01 | -86.26 | $< 2e-16$ |
| HELLO-Int | 3.844e+00 | 236.49 | $< 2e-16$ |
| HELLO-IT | 4.690e+00 | 197.41 | $< 2e-16$ |
| ACK | -1.528e-17 | -9.40e-16 | 1 |
| MTU | -1.335e-19 | -8.64e-16 | 1 |
| SPF-CI | 9.796e-01 | 60.26 | $< 2e-16$ |

Residuals

| Min | 1Q | Median | 3Q | Max |
|---|---|---|---|---|
| -16.567 | -4.262 | 0.523 | 3.762 | 15.306 |

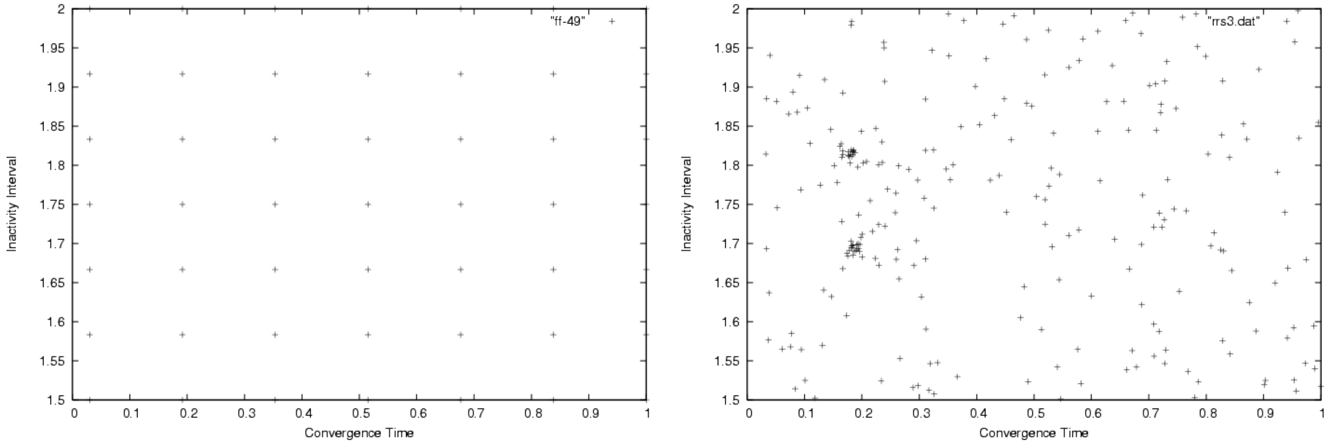
Table 1, as in the RRS experiments. In Table 5, we fitted a linear regression model to the data and found that the same three input parameters had the most effect on the model. The FFED model produced an adjusted R-squared value of 85%. We were confident that RRS was properly modeling the same parameter space as we would have explored had we done a more detailed FFED. Plotting the same two most effective parameters as we modeled in RRS, the scatter plot did not show any trends in the data, as shown in Figure 4. The final step in our verification was to create a quantile-quantile plot of the residual errors. We observed a linear relationship between sample error and theoretical. From Figure 4, the linear model assumptions of normality appear to be valid for the FFED model.

FFED generated 16,807 experiment runs, 20 times more than the RRS design, and yielded less information in the areas that we were interested in studying. Figure 5 illustrates the amount of detail generated by RRS versus the FFED for convergence times in the sub-second range. RRS also generate a "good" value for the convergence time 0.11 seconds, which was within 7% of the FFED best value. While we could have generated a FFED using the final RRS input parameter ranges, we would not have had the benefit of the knowing that was in fact the area of interest, beyond our ability to analyze the system. In fact, we purposely chose FFED presented here because we wanted to be certain about the nature of the system. It was necessary to explore a large range in order to validate our results in the RRS design.

## 7 CONCLUSIONS

In this paper, we demonstrate the efficacy of the Recursive Random Search (RRS) technique when applied to large-scale meta-simulation of OSPF routing networks. We found that: (i) the number of simulation experiments is reduced by an order of magnitude when compared to full-factorial experiment design (FFED) approach, (ii) this approach enabled the rapid elimination of unnecessary parameters, and (iii) RRS enabled the rapid understanding of key parameter interactions. By using RRS we made the interesting observation that when modeling only OSPF control-plane dynamics we were able to shrink the number nodes down to that subset that was only needed for determining convergence times. This reduction resulted in models that execute 100 times faster than their full topology counterparts.

In the future, we plan to leverage our experience here to examine potential optimization to the OSPF protocol that may decrease convergence times over what has been previously reported.



(a) FFED

(b) RRS

Figure 5: Comparison of FFED and RRS: FFED Samples Uniformly While RRS First Samples Randomly and then Focuses on Specific Regions

## ACKNOWLEDGMENTS

This research is supported by an NSF CAREER Award CCR-0133488, the DARPA Network Modeling and Simulation program, contract #F30602-00-2-0537 and a AT&T University Relations Program Grant.

## REFERENCES

Aarts, E., and J. Korst. 1989, January. *Simulated annealing and boltzmann machines: A stochastic approach to combinatorial optimization and neural computing*. Paperback ed. John Wiley and Sons.

Alaettinoglu, C., V. Jacobson, and H. Yu. 2000, October. Towards millisecond igp convergence. In *NANOG Meeting*. Available online via <http://www.nanog.org/mtg-0010> [accessed August 3, 2004].

Berry, O., and D. R. Jefferson. 1985, January. Critical path analysis of distributed simulation. In *Proceedings of SCS Multiconference on Distributed Simulation*, 57–60.

Floyd, S. 2001, January. Simulation is crucial. *IEEE Spectrum* 38 (1): Sidebar.

Floyd, S., and V. Paxson. 2001, August. Difficulties in simulating the Internet. *IEEE/ACM Transactions on Networking* 9 (4): 392–403.

Goldberg, D. 1989. *Genetic algorithms in search, optimization, and machine learning*. Addison Wesley.

Goyal, M., K. K. Ramakrishnan, and W. Feng. 2003, May. Achieving faster failure detection in ospf networks. In *Proceedings of the International Conference on Communications (ICC)*, Volume 1, 296–300.

Jain, R. 1991. *The art of computer systems performance analysis: Techniques for experimental design, measurement, simulation, and modeling*. Wiley - Interscience.

Montgomery, D. C. 2001. *Design and analysis of experiments*. John Wiley and Sons.

Moy, J. 1998, April. OSPF version 2. Technical Report RFC 2328, IETF.

R 2004. The r project for statistical computing. Available online via <http://www.r-project.org> [accessed August 3, 2004].

Rocketfuel 2002. Rocketfuel internet topology database. Available online via <http://www.cs.washington.edu/research/networking/rocketfuel> [accessed August 3, 2004].

Stewart, J. 1999. *Bgp-4 inter-domain routing in the internet*. Addison Wesley.

Törn, A., and A. Žilinskas. 1989. *Global optimization*, Volume 350 of *Lecture Notes in Computer Science*. Springer-Verlag.

Ye, T., and S. Kalyanaraman. 2003. A recursive random search algorithm for large-scale network parameter configuration. In *Proceedings of SIGMETRICS*, 196–205.

## AUTHOR BIOGRAPHIES

**DAVID BAUER** is a Ph.D. candidate in Computer Science Department of Rensselaer Polytechnic Institute (RPI). His research interests include parallel and distributed systems and network simulation with a focus on performance optimizations. His e-mail address is <bauerd@cs.rpi.edu>.

**GARRETT YAUN** is a Ph.D. candidate in the Department of Computer Science at Rensselaer Polytechnic Institute (RPI). His research interests include parallel and distributed systems, networking, modeling, and simulation. His e-mail address is <yaung@cs.rpi.edu>.

**CHRISTOPHER D. CAROTHERS** is an Associate Professor in the Computer Science Department at Rensselaer Polytechnic Institute. He received the Ph.D., M.S., and B.S. from Georgia Institute of Technology in 1997, 1996, and 1991, respectively. Prior to joining RPI, he was a research scientist at the Georgia Institute of Technology. His research interests include parallel and distributed systems, simulation, networking, and computer architecture. His e-mail address is <chrisc@cs.rpi.edu>.

**MURAT YUKSEL** is a Post-Doctoral Research Associate at ECSE Department of Rensselaer Polytechnic Institute (RPI), Troy, NY. He received a B.S. degree from Computer Engineering Department of Ege University, Izmir, Turkey in 1996. He received M.S. and Ph.D. degrees from Computer Science Department of RPI in 1999 and 2002 respectively. His research include routing in wireless ad-hoc networks, large-scale network simulation, mobile free-space optical networks, network economics, and performance analysis. His e-mail address is <yuksem@ecse.rpi.edu>.

**SHIVKUMAR KALYANARAMAN** is an Associate Professor at the Department of Electrical, Computer and Systems Engineering at Rensselaer Polytechnic Institute in Troy, NY. He received a B.Tech degree from the Indian Institute of Technology, Madras, India in July 1993, followed by M.S. and Ph.D. degrees in Computer and Information Sciences at the Ohio State University in 1994 and 1997 respectively. His research is in topics such as congestion control architectures, quality of service (QoS), free-space optical networking, network management, multicast, pricing, multimedia networking, and performance analysis. His e-mail address is <shivkuma@ecse.rpi.edu>.