SANDIA REPORT
SAND2021-12193
Printed September 2021

Sandia National Laboratories

Foundations of Rigorous Cyber Experimentation

Michael Stickland, Justin D. Li, Laura Swiler, and Thomas Tarman

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico
87185 and Livermore,
California 94550

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & Engineering Solutions of Sandia, LLC.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@osti.gov
Online ordering: http://www.osti.gov/scitech

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5301 Shawnee Rd
Alexandria, VA 22312

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.gov
Online order: https://classic.ntis.gov/help/order-methods/

ABSTRACT

This report presents the results of the "Foundations of Rigorous Cyber Experimentation" (FORCE) Laboratory Directed Research and Development (LDRD) project. This project is a companion project to the "Science and Engineering of Cyber security through Uncertainty quantification and Rigorous Experimentation" (SECURE) Grand Challenge LDRD project. This project leverages the offline, controlled nature of cyber experimentation technologies in general, and emulation testbeds in particular, to assess how uncertainties in network conditions affect uncertainties in key metrics.

We conduct extensive experimentation using a Firewheel emulation-based cyber testbed model of Invisible Internet Project (I2P) networks to understand a de-anonymization attack formerly presented in the literature. Our goals in this analysis are to see if we can leverage emulation testbeds to produce reliably repeatable experimental networks at scale, identify significant parameters influencing experimental results, replicate the previous results, quantify uncertainty associated with the predictions, and apply multi-fidelity techniques to forecast results to real-world network scales. The I2P networks we study are up to three orders of magnitude larger than the networks studied in SECURE and presented additional challenges to identify significant parameters.

The key contributions of this project are the application of SECURE techniques such as UQ to a scenario of interest and scaling the SECURE techniques to larger network sizes. This report describes the experimental methods and results of these studies in more detail. In addition, the process of constructing these large-scale experiments tested the limits of the Firewell emulation-based technologies. Therefore, another contribution of this work is that it informed the Firewheel developers of scaling limitations, which were subsequently corrected.

ACKNOWLEDGEMENTS

This work was funded by the Laboratory Directed Research and Development program at Sandia National Laboratories. The authors gratefully acknowledge the support of the National Security Program Investment Area team for their support of this project. We acknowledge the help of Kasimir Gabert, our colleague who, among other things, implemented the Snark torrenting capability in our I2P model, and Corithian Williams, a 2019 summer student in the Center for Cyber Defenders program, who wrote the initial version of the data analytics scripts. Finally, we acknowledge the help of our colleagues supporting various cyber emulation initiatives, including Steven Elliott and Chris Symonds.

CONTENTS

TABLE OF FIGURES

LIST OF TABLES

This page left blank

ACRONYMS AND DEFINITIONS

Abbreviation	Definition
AS	Autonomous System (a subset of the global internet)
BW	Bandwidth
BW_DistType	Bandwidth Distribution Type
CHR	Confirmed Hit Rate (a probability of de-anonymization success)
CPU	Central Processing Unit
DES	Discrete Event Simulation
FORCE	Foundations for Rigorous Cyber Experimentation
HPC	High-Performance Computing
I2P	Invisible Internet Project
IP	Internet Protocol [address]
IQR	Interquartile Range
Kbps	Kilobytes per second
LDRD	Laboratory Directed Research and Development
MFUQ	Multifidelity Uncertainty Quantification
RAM	Random Access Memory
RI	RouterInfo (contains identifying information about an I2P router)
SA	Sensitivity Analysis
SECURE	Science and Engineering of Cyber security through Uncertainty quantification and Rigorous Experimentation
UQ	Uncertainty Quantification
V&V	Verification & Validation
VIF	Variance Inflation Factor
VM	Virtual Machine

1. INTRODUCTION

This report summarizes the work performed under the "Foundations of Rigorous Cyber Experimentation" (FORCE) Laboratory Directed Research and Development (LDRD) project. This report describes an application of uncertainty quantification (UQ) in a large-scale study of de-anonymization of user activity in a distributed, anonymity preserving peer-to-peer network, the Invisible Internet Project (I2P). Uncertainty is inherent in peer-to-peer, Internet overlay networks and may affect results from studies conducted on the Internet which cannot control many variables. Cyber testbeds running in emulated environments allow us to vary certain parameters of interest, observe how the network responds to changes, and perform UQ. We study the impact of uncertainty in de-anonymization success.

This project complements, and leverages work done in the "Science and Engineering of Cyber security through Uncertainty quantification and Rigorous Experimentation" (SECURE) Grand Challenge LDRD project. The SECURE Grand Challenge has identified experimental design and uncertainty quantification as pillars of rigorous cyber experimentation. In the FORCE project, we are focused on one in-depth case study where we demonstrate the use of experimental design and UQ at scale on large cyber networks. The networks being emulated in FORCE are one to three orders of magnitude larger than those studied in SECURE. We also demonstrate the use of multi-fidelity experimentation and analysis using regression and discrete event simulation (DES) to project emulated experimental results to higher network scales, and apply the multifidelity uncertainty quantification (MFUQ) technique developed by the SECURE team to reduce the variance of results.

We performed experimentation using a Firewheel [12] cyber emulation model of an I2P network to understand a de-anonymization attack formerly presented in the literature. We demonstrate we can identify parameters influencing the results, and quantify uncertainty associated with de-anonymization attributions. Further, we present the use and usefulness of statistical analysis on emulations of large-scale, distributed cyber networks, and highlight the value of applying UQ in cyber experimentation.

This study provides us with a rich set of questions to investigate not only the potential for replication and studying a previous work, but also to consider the following:

- Verification and Validation (V&V) of emulation model. The study performed by Egger et al. involved live experimentation on the internet. The studies we performed were entirely emulated in the Firewheel [12] environment. This comparison is important for both verification and validation of emulator performance in modeling the I2P network.
- Uncertainty quantification (UQ) and sensitivity analysis (SA). The studies documented in this report focus on "forward UQ" which refers to propagating uncertainties in input parameters of a model to the corresponding uncertainties in the responses from that model. We examined a variety of configuration and topology uncertainties relating to I2P. Other uncertainties in cyber experiments could involve environment, user, and threat uncertainties. We performed a variety of SA studies.
- Advanced experimental design topics. With the experimental platform we have set up now and the Firewheel I2P framework, we plan to address multifidelity UQ. In multifidelity UQ, the idea is to run a small number of expensive, high-fidelity runs (such as Firewheel runs involving thousands of routers over several days) with many runs of a lower-fidelity

model. The lower-fidelity model does not need to be very accurate; its results simply need to be correlated with the high-fidelity model. Combining results from two fidelities of a model can result in statistical estimators that are lower-variance and more efficient than simply calculating the statistics based on the high-fidelity model alone.

The report outline is as follows: Section 1.1 presents project background information, and Section 1.2 discusses related work in cyber experimentation and the science of cybersecurity and how this work integrates this related work. Section 2 describes how the FORCE I2P model evolved during the three years of FORCE research. Sections 3 and 4 provide an overview of the I2P case study and the experimental configurations for the studies. Section 5 describes the data analytics that were developed for this research, to process the TB size datasets and to calculate the mean confirmed hit rate (CHR) across many groupings and configurations (per node, per parameter setting type, etc.). Section 6 describes the emulation results obtained throughout the FORCE project. Section 7 discusses multifidelity UQ and its demonstration on the I2P network study; Section 7 also discusses optimal experimental design. Section 8 addresses lessons learned and best practices. Conclusions are summarized in Section 9. Finally, Appendix A provides an overview of definitions related to experimentation, uncertainty quantification, and sensitivity analysis.

1.1. Background

This project's proposal was originally submitted as a backup plan for the SECURE Grand Challenge in case that proposal was rejected. Instead, the SECURE Grand Challenge proposal and this project's proposal were both accepted. As a result, this project focuses on the application of SECURE techniques such as UQ to a scenario of interest and scaling the SECURE techniques to larger network sizes. We take a deep dive into one specific cyber case study, while the SECURE Grand Challenge project focuses on advancing the theoretical foundations of the science and engineering of cyber security through uncertainty quantification and rigorous experimentation.

Cyber testbeds have been established to provide a platform for research and experimentation on networks [6]. These testbeds often deploy many virtual machines (VMs), running on individual or clusters of powerful host computers, to provide greater network sizes at lower costs. Uses of cyber testbeds include test and evaluation, identification of network performance, cyber security investigation, and training [20,41]. Some examples of testbeds include LARIAT [36], Emulab [30,43], DETER [4, 31], and DARPA's National Cyber Range [10]. In this work, we use the Firewheel emulation testbed [12], which was developed at Sandia National Laboratories (SNL) as part of the Emulytics™ (Emulation and Analytics) program.

1.1.1. Case Study Selection

Our selected cyber case study started with a paper called "Practical Attacks against the I2P Network" by Christoph Egger et al. in 2013 [9]. I2P refers to the Invisible Internet Project [22] which is an anonymous network built on top of the internet to allow protected communications. It is a type of "darknet" technology which is described in more detail in Section 3.

In this report, we present the results of a series of experiments performed on large, emulated networks to study a particular attack on the I2P network described in [9], and herein referred to as the I2P Case Study.

We selected the I2P Case Study for the following three reasons:

1. Egger et al. utilized a multi-phased approach for predicting the probability of successfully de-anonymizing a target victim in a large-scale, distributed anonymization network, and this study involved live experimentation on the Internet. In contrast, our studies use an offline, emulated testbed environment.
2. Part of their process produced a very specific measurement (*u*, an empirically measured property of the I2P network) that was derived by determining the probability of successfully correlating sets of stochastic events generated by the network. Much of the work presented here involves understanding the uncertainty associated with that key parameter, *u,* which was found to have a value of 0.52 in Egger's work [9]. The studies documented in this paper focus on propagating I2P configuration uncertainties to the corresponding uncertainty in *u* and beyond. Note that *u* is also referred to herein as mean Confirmed Hit Rate or mean CHR in later chapters.
3. Egger et al. then used this specific measurement (*u*) in their calculations when making de-anonymization attributions about their attack's intended victims, regarding otherwise anonymized communications. This provides a rich example to show how uncertainties in one parameter or one part of the system propagate to downstream or aggregate predictions.

1.1.2. Preexistent I2P Model

Previous work by colleagues on the Firewheel team [12] had already produced a Firewheel emulation model of the I2P network. It contained all the fundamental building blocks for the creation of basic I2P experimental networks, however, it was now unusable as it was built to run on an early version of Firewheel v1.0 and wasn't compatible with the newest (at the time) Firewheel v2.0. Even so, it could provide us with an established I2P emulation model that we could reengineer to work on the newer version of Firewheel, and then be expanded on to support all our experimental needs.

While not the deciding factor, the availability of this existing I2P emulation model did weigh considerably on our decision to select the I2P Case Study over other candidate case studies not involving the I2P network.

1.2. Previous Work

During the last 15 years there has been much work in developing virtual testbed technologies to conduct cyber experimentation at scales that exceed what can be achieved using physical testbeds [4, 10, 12, 31, 43]. These technologies include tools for defining topologies, deploying experiments on one or more physical computing nodes, orchestrating experiments, collecting results, and analyzing collected data. In parallel, related efforts in virtualization research and development led to today's cloud computing technologies, which are now used in a variety of applications where computing at scale is needed, including cyber testbeds.

Also during this time there developed an increased interest in the science of cybersecurity [3, 8, 18, 27] due to increasing concern about the lack of scientific rigor in cyber experiments. Often such experiments would be conducted without an articulated, falsifiable hypothesis, and would not be documented sufficiently to facilitate experiment reproduction by other research teams. These shortcomings in experimental methodology can lead to confusion about experimental conclusions and courses of action that should be taken as a result of these conclusions. For decisions regarding

high consequence cyber systems such as the Nuclear Command, Control, and Communications (NC3) system, more rigor in cyber experimentation is needed to produce scientifically meaningful results that can inform these decisions.

Recent work in this LDRD, and the companion SECURE Grand Challenge LDRD have addressed this gap between mature experimental testbed technologies and nascent work in the science of cybersecurity. For example, formal uncertainty quantification and sensitivity analysis methods were recently used for cyber experimentation (see [46] for an example of a DNS amplification attack scenario, ranking the most important parameters affecting Central Processing Unit (CPU) utilization and victim response rate). However, experimental design and uncertainty/sensitivity analyses methods are not yet used widely in the cyber emulation community. This is a contribution of SECURE and FORCE.

2. OVERVIEW AND EVOLUTION OF RESEARCH

To achieve our goals, we need our I2P emulation model to run reliably repeatable experiments at network scales and runtime durations not previously achieved using the Firewheel testbed. We set out to verify that our inherited (newly ported to Firewheel 2.0), original I2P model could produce reliably repeatable experiments. This chapter discusses the topologies generated for these experiments as well as the parameters we investigated.

2.1. Network Topology Generation

Our I2P model generates network topologies that attempt to notionally replicate the Internet structure that the real I2P network runs on. To this end, as shown in Figure 2-1, our model includes an emulated tier 1 Internet backbone, tier 2 routing, and distributions of autonomous system (AS) subnets and the I2P routers within each subnet. The subnet and I2P router distributions i.e., number of subnets with I2P routers and number of I2P routers per subnet, are determined per AS by sampling from cumulative distributions constructed using data from previous I2P measurement studies [25].

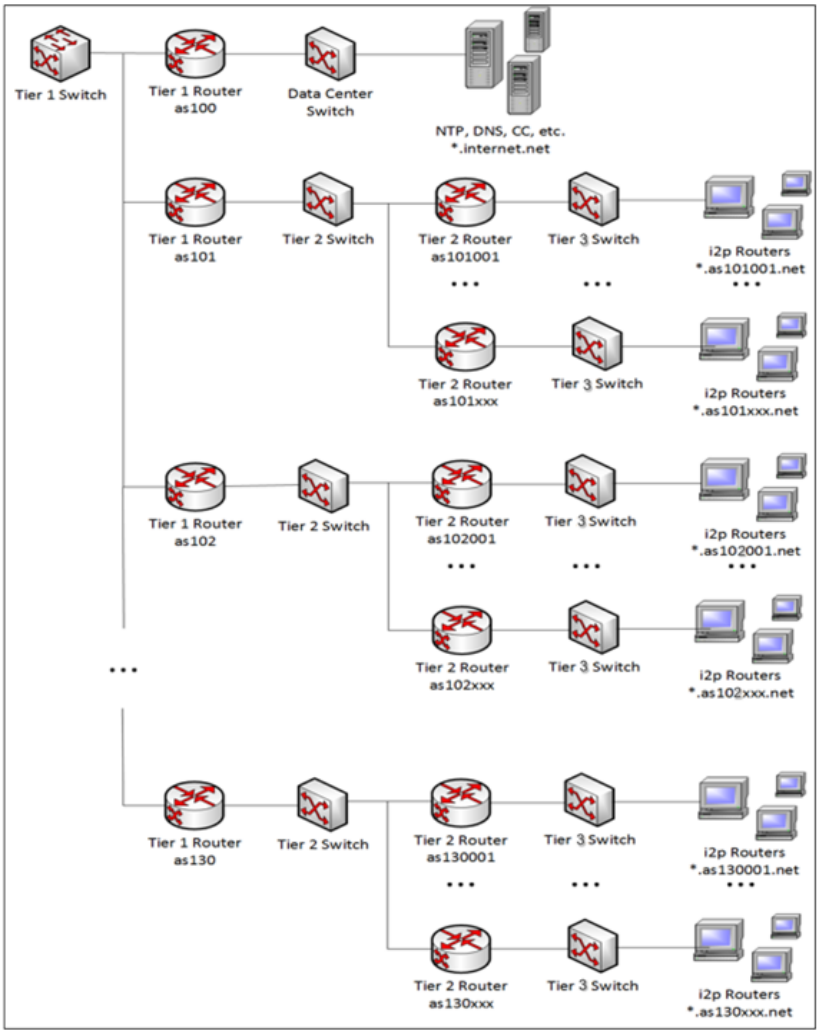

Figure 2-1. Notional Internet Topology

All Tier 1 BGP Routers are linked in a full mesh across a single Tier 1 Switch, and all Tier 2 BGP Routers are linked in a full mesh across their respective tier 2 Switches. Finally, we also include a data center network with DNS and NTP servers, I2P and Snark torrenting bootstrap servers, and an Eepsite server (an I2P hidden service website used for downloading files via the I2P network.)

2.1.1. Original Dynamic Model

The I2P model we inherited used a dynamic network topology generation algorithm, as described above, but one which did not generate the same network topology across subsequent runs using the same experiment parameters. While this approach did produce a notional Internet and a I2P overlay network of roughly a specified size, these were not reliably repeatable as is needed for our work.

2.1.2. Static Model

To remedy the non-repeatability problem of the original model, we developed a new static network topology generation algorithm. As its name implies, this model's topology generation algorithm generated the same network topology repeatedly and reliably. All the Internet backbone, tier 2 and 3 routers, and the number of subnets and I2P routers per subnet were always the same. For any given network size, all the topology of a smaller network size was replicated identically. For instance, a 500-node network would contain the exact 400-node sub-topology as was generated for a 400-node network, and so forth. While this approach did fulfill our reliability of repeatability requirement, since larger networks always contained exact subsets of all smaller networks the algorithm would generate, we felt it was too deterministic and wouldn't afford us the opportunity to examine any possibility of effects caused by differences in I2P overlay network topology.

2.1.3. New Dynamic Model

We then reengineered the original dynamic model's network topology generation algorithm to make it reliably repeatable for any specific network size, but still generate randomly different I2P overlay topologies for different network sizes. We felt that this compromise between the deterministic nature of our static model and the stochasticity of the original dynamic model would satisfy both of our requirements i.e., reliable repeatability and variability in I2P network topology.

2.1.4. Network Topology Generation V&V

To ensure that our changes to the network topology generation algorithm did not significantly affect the mean CHR metric we are focused on, we conducted a series of network scale experiments using each of the different models. The results of these experiments (Figure 2-2) showed that all three models' results corresponded and tracked closely with one another across varying network scales. However, our new dynamic model's results corresponded most closely with the deterministic, static model's results, thus giving us confidence that our changes had not adversely affected the outcomes for our primary experimental metric of interest.

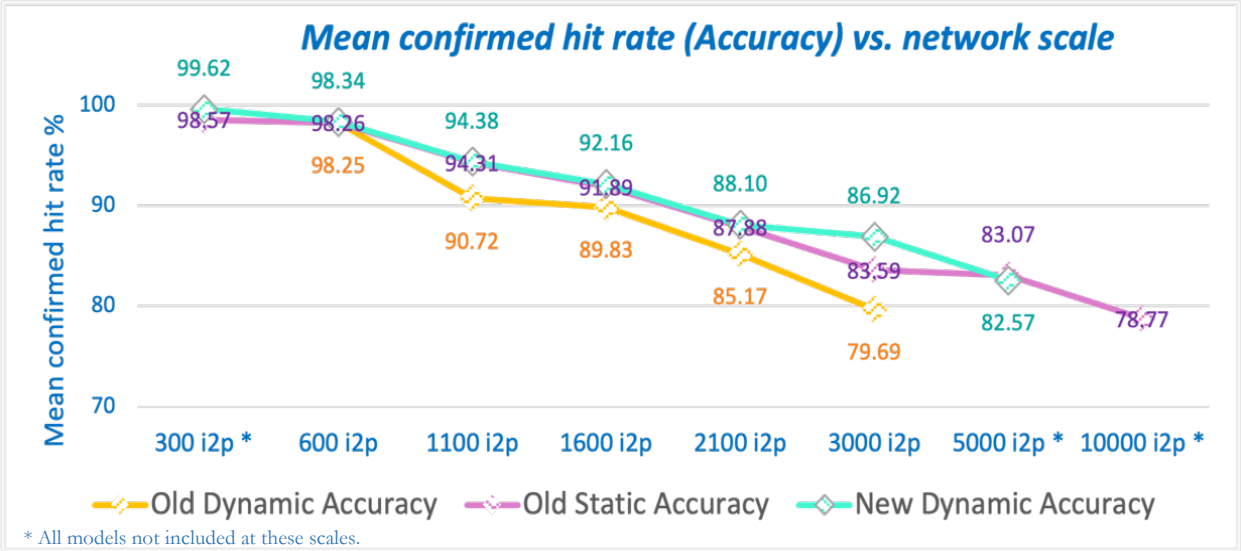

Figure 2-2. Comparison of Network Topology Generation Algorithms

2.2. Parameter Selection

Our work is focused on how experimental inputs affect outputs, as well as predictions and analyses based on those outputs, so selecting input parameters that matter is vital. Unfortunately, we didn't know much about how I2P worked when we started this project, and as a result we had to discover which parameters would be significant.

We had discovered early on that network scale i.e., the overall number of I2P routers included in each of our emulated I2P experiment networks, had an observable effect on an experiment's mean CHR. In addition, network scale also affected the variance of the CHR [see Section 6, Figure 6-1, and Figure 6-14]. But we were also interested in finding parameters that would affect the CHR of specific, targeted routers.

2.2.1. Local Parameters for Local Effects

We initially operated under the assumption, and hope, that we could cause changes to an individual I2P router's CHR by making changes to its local operating environment. This was a naïve assumption, based on a shallow (or lack of) understanding of the inner workings of the I2P routing software, and was eventually proven to be incorrect. However, the idea was that if a router being targeted by an attacker was configured in a way that significantly altered its CHR, then this could disrupt the effectiveness of the attack. Therefore, we conducted numerous experiments which varied parameters on individual routers and measured the resulting CHR for them before we became much more fluent in the workings of I2P routers. During this phase, we varied such parameters as a VMs available bandwidth, the percentage of that bandwidth it shared with its I2P router software, the number of virtual CPU cores (sockets) and the amount of Random Access Memory (RAM) that it had. We also looked at whether the number and/or length of a router's encrypted tunnels would affect its local CHR.

We developed an automated capability for configuring sets of targeted routers, called Victim Groups since they were intended to be victims of the de-anonymization attack, so we could have

multiple sets of similarly configured routers per experiment. This also allowed us to configure different sets of routers with different configuration profiles, as well as to specify the number of duplicate sets for each configuration profile. As stated previously, this approach didn't identify any local parameters that could reliably, if at all, predict a change in a that router's CHR.

By this time, though, we had observed enough of the I2P network's behaviors and become more familiar with the I2P software, and as such had developed a different way of thinking about experimental parameters.

2.2.2. Global Parameters for Global Effects

As mentioned above, we discovered early on that network scale had an observable effect on an experiment's mean CHR, and on the variance of the CHR. Network scale is a property of the network, and not of an individual node, and this is the shift in thinking about experimental input parameters that we pursued from here on out. This led us to look at the global distribution of bandwidths across the real I2P network, and to derive a cumulative distribution from the real-world data that we could apply proportionately to our experimental networks.

Other global parameters we experimented with include the percentage of the overall number of I2P routers per experiment that were using Snark, I2P's native torrent-based file sharing application, the size range of the torrent files being shared on the network, and other globally set torrent-related router behaviors.

Using this global approach to input parameters, we started seeing effects on mean CHRs, and over time we discovered a few global input properties we could set that would affect global experimental outputs.

3. I2P CASE STUDY DESCRIPTION

The details of the I2P Case Study are presented below, including the study motivation, the experimental configuration used, the parameters that were varied, and the experimental results. Additionally, information is provided about a process called *forward uncertainty quantification*, which is the propagation of uncertainties in input parameters or configuration settings through the emulated experiment to determine uncertainty in the corresponding quantities of interest.

3.1. Invisible Internet Project (I2P)

I2P [22, 23] is an anonymous network consisting of peers (also called routers) running I2P routing software that allows them to communicate with one another with a degree of anonymity. As shown in Figure 3-1, anonymization involves multi-hop encryption in both directions to anonymize both clients and servers. Previous investigations of the I2P network include the impact of a DNS misconfiguration [11] and a study of I2P performance [19].

When an I2P router wants to make its presence known to the network, it publishes its RouterInfo (RI) to the netDb, a Kademlia [28] distributed hash table (DHT) database, which is managed by a subset of about 6% of all I2P peers, called floodfill nodes. RI data contains information needed to contact a router within the I2P network. It is accessed by querying floodfill nodes and used by all peers when building their encrypted tunnels. All routers rebuild their encrypted tunnels about every ten minutes, and also they all repeatedly store their RI data using this same frequency. Once a router stores its RI data to a floodfill node, the floodfill then 'floods' this data to other floodfill nodes in the network, thus replicating RI data to other netDb nodes. The netDb also maintains LeaseSet (LS) data, containing information needed to contact hidden destination sites within the I2P network, called Eepsites.

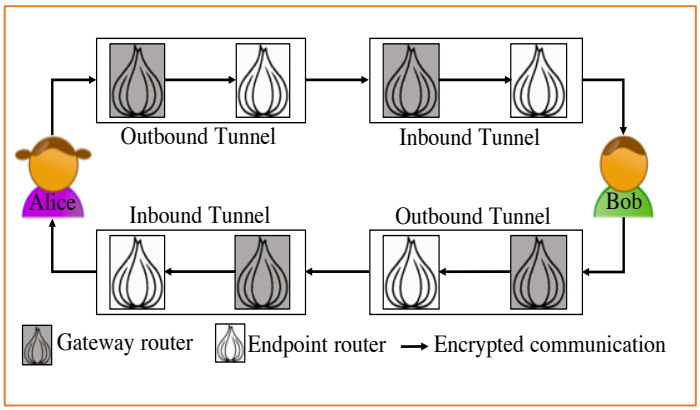

Figure 3-1. I2P communications using inbound and outbound tunnels (from Hoang, P.H., et al., An Empirical Study of the I2P Anonymity Network and its Censorship Resistance. arXiv:1809.09086v2, 2018.)

Before we continue, we need to point out that some features and default behaviors of the I2P network have changed over time. Of particular importance to our work is that when the I2P Case Study was conducted, each router performed what's called a verification lookup step after completing their RI *store* step. This *verify* step is critical in the de-anonymization attack described in [9], and can still be performed by today's I2P routers, though it's now been disabled by default (as of I2P ver. 0.9.7 [23]). In our work we reenabled the *verify* step on all I2P routers used in our experiments, and we describe this behavior herein as it worked when Egger et al did their research.

Each peer I2P router populates the netDb as follows (see Figure 3-2):

1. Select one of today's closest floodfill nodes; send the floodfill a request to *store* its RI using a plaintext channel (no encryption; RIs includes peers' cleartext IP address, so it's already known.)
2. Wait about 20 seconds.
3. Select a different one of today's closest floodfill nodes; send the second floodfill a request to *lookup* its RI and hence *verify* successful *replication*, this time using an encrypted tunnel (this ensures the *verify* lookup is indistinguishable from any other RI *lookup*).

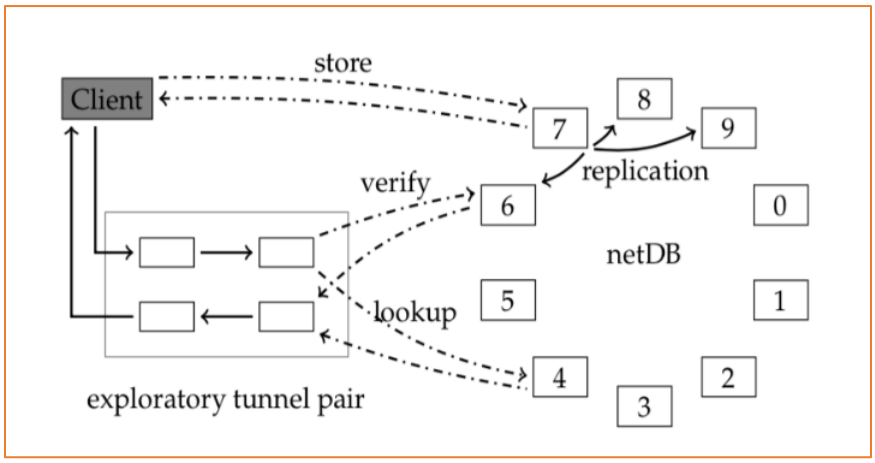

Figure 3-2. netDb store, verify, and lookup operations (from Egger, C., et al. Practical Attacks against the I2P Network. 2013. Berlin, Heidelberg: Springer Berlin Heidelberg.)

In their 2013 paper [9], Egger et al. investigated a multi-phased set of attacks that resulted in the attacker's ability to de-anonymize a targeted victim on the I2P network. We note that their work resulted in several changes to subsequent versions of the I2P software, which corrected these attacks, including the disabling of the RI verification lookup step mentioned above.

Their attack allowed them to control all the floodfill nodes utilized by their victim I2P routers and maintained this advantage even as the victims selected a different set of floodfills to use each day. As such, whenever they observed a victim *store* their RI (to one of the floodfills they controlled) and then saw a *lookup* for the same RI around 20 seconds later (again on a floodfill node they controlled) they would assume the *lookup* was the *verify* step done by the victim and call that a *hit*, and associate the tunnel used for the *lookup* with the victim who performed the *store*.

During their study, using ground truth data obtained from their victim I2P routers, they observed an empirical success probability of 0.52 for successfully associating a victim's *store* with its related *verify* lookup. The reason they were not 100% successful at associating these events is because all other routers in the network *lookup* peers' RI when periodically rebuilding their encrypted tunnels. If a peer requested a *lookup* of the victim's RI from one of the floodfill nodes controlled by the attacker, at around 20 seconds after an RI *store* done by the victim, but before the victim did their *verify* lookup, then the attacker would erroneously associate this other peer's tunnel with the victim. We hereafter refer to this success probability as the Confirmed Hit Rate and denote it as u.

4. I2P CASE STUDY EXPERIMENTAL GOALS AND METHODOLOGY

We wish to understand how uncertainties in I2P network properties (e.g., network size, I2P router bandwidth, number of CPU cores, percent of peers generating encrypted traffic, peer traffic profiles, etc.) contribute to uncertainty in de-anonymizing a victim. Answering this question involves three steps: an experimental step to propagate input uncertainties to the emulated network; data collection and analysis to determine the effects of these changes on the output parameter u; and an analytical step to compute the de-anonymization uncertainty as a function of uncertainty in u.

Our motivating questions pertaining to the study described in [9] are the following:

1. Is the confirmed hit rate probability, $u = 0.52$, stable?
2. Which I2P router and network environment parameters might affect the value of this probability?
3. Are the de-anonymization results (i.e., attribution probabilities) that rely on u robust when there are uncertainties in input parameters?

We address question 1 by constructing an I2P topology model and running it in an experimental environment, so we can vary several local router configuration settings and global network properties. The I2P routers are instrumented to collect data that is then used to determine confirmed hit rates during post-experiment analysis. To address question 2, we perform a sensitivity analysis on the I2P topology to assess the level of influence various input parameters have on the confirmed hit rates of individual routers, and on the population.

Once a set of parameters that impact confirmed hit rates are identified, the third question is addressed by varying these parameters' values, measuring the variation in the confirmed hit rates, and then using the range of confirmed hit rate values to calculate attack metrics of interest. These metrics include:

- Probability of successful attribution within k trials
- Probability of successfully attributing k connections over N observations

Our approach to the third question (an uncertainty quantification question) is shown in Figure 4-1. It shows forward propagation from uncertain network conditions to uncertainty in de-anonymization attribution as a two-step process. Uncertainty in the confirmed hit rate u is determined via experimentation using an emulation-based testbed, with a set of inputs that are fixed for each experiment and a set that are uncertain (or varied) during each experiment (e.g., number of I2P routers, router bandwidths, and user behaviors). Experimentation is required to calculate u because, due to the complexities and scale associated with the I2P network, u cannot be calculated analytically. Once a range of values of u is determined, these values are used in post-experimental analysis to calculate the attribution probabilities listed above.

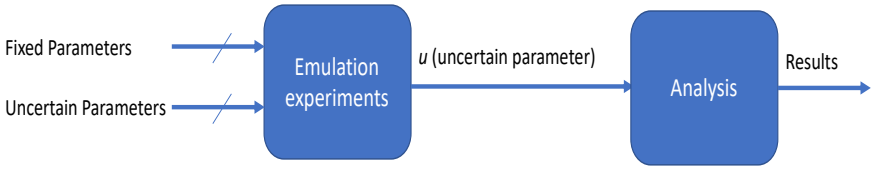

Figure 4-1. Forward propagation of uncertainty in our I2P study

4.1. Experimental Network Design

Our emulated I2P network is modeled using Firewheel, a cyber experiment orchestration tool [12] which assists a user in building and controlling repeatable experiments of large-scale distributed network systems. Our Firewheel model allows us to specify the size of the I2P network for each experiment run as well as other network conditions (see Figure 2-1). The emulated Internet backbone, tier 2 routing, the distribution of tier 3 autonomous system (AS) routers, the I2P router distributions within each AS i.e., number of subnets with I2P routers and number of I2P routers per subnet, are generated based on data from previous I2P measurement studies [25]. We also include a data center network with DNS and NTP servers, I2P and Snark torrenting bootstrap servers, and an Eepsite to download files from.

All I2P routers run a modified copy of I2P version 0.9.29, installed on Ubuntu 14.04 Desktop (which were both the latest versions when our I2P model was originally built). The I2P software was modified to collect ground truth data about RI *store* and *verify* lookup events on all non-floodfill (client) routers, and to collect all observed RI *store* and all *lookup* event data on all floodfill (server) routers. This information is used to calculate the confirmed hit rate, u, as well as other statistics reported herein.

The emulated experiments are run on a High-Performance Computing (HPC) platform, due to the experimental network sizes. Each HPC host machine has 32 CPU cores across 2 sockets, 512 Gigabytes of RAM, and 100 Gigabit Ethernet. The provisioning of virtual machines and virtual networking is performed across an experiment's cluster of HPC nodes using another technology, called minimega [5,29], a tool for launching and managing virtual machines on a laptop or cluster, which is available as open source. minimega can accept a network topology from Firewheel and deploy it in a matter of minutes.

4.2. Experiment Configuration for Experimental Propagation of Uncertainty

We performed four types of emulation experiments, varying different configuration parameters relating to I2P in each. The first three emulation experiment types are exploratory in nature. That is, we conducted them to identify which parameters, when changed, resulted in observable changes to confirmed hit rates. The final emulation experiment type is to conduct a sensitivity analysis on a final set of parameters chosen from amongst those evaluated during the exploratory experiments.

The experimental results are presented in detail in Section 5. In Table 4-1, we present a summary of the four experiment types and parameters that were varied, including the values used. Note that for these experiments, we discretized the input uncertainties to take a set of specific values enumerated in Table 4-1 and did not treat the uncertain inputs with parametric distributions such as normal, Weibull, etc. Some of the variables are inherently discrete (e.g. number of CPU cores or bandwidth) and so continuous distributions are not appropriate. Additionally, a select number of discrete values per input allows one to perform a "main effects analysis" which tests if the mean response value changes significantly as the input parameter varies over the input domain.

Note that generating samples of uncertain parameters and running the emulation with those sampled values is a way to perform uncertainty analysis in emulated systems. This is often the only tractable approach to perform UQ on problems which cannot be solved analytically. We used this approach to understand the uncertainty in u (mean CHR) which we then carried forward to the uncertainty in de-anonymization probability (Section 6.2). We stress that by traditional computational modeling standards, the experimental design in terms of the number of uncertain

parameters and the total number of parameters is straightforward for these studies (Table 4-1 and Section 7.3). As such, the total number of experiments would, traditionally, be considered low. However, each particular experimental I2P network run—with specific configuration parameters—produced a significant amount of data: One emulation run provided hundreds or thousands of nodes, each with its own confirmed hit rate per day. We partition the confirmed hit rate results by various node attributes including bandwidth and percent of I2P share and also study the confirmed hit rate per node configuration per day. Overall, this gives us rich datasets with which to test our hypotheses.

Experiment Type	Parameters varied and range
(Exploratory) Type I: Network Scales	Number of i2p routers: {150, 300, 450, 600, 1100, 1600, 2100, 3000, 5000, 10000}
(Exploratory) Type II: Victim Groups	Bandwidth (BW): {8, 12, 16,3 2, 64, 128, 256, 512} Kbps I2P Share%: {0, 10, 50, 100} Number of CPU cores per router: {1 or 8} Note: *Only routers in sampled victim groups vary parameters using these settings, all others are fixed:* {*BW 40 Kbps, Share 100%, cores 1*}.
(Exploratory) Type III: Global Distributions	Bandwidth distribution notionally derived from I2P Metrics data [21]: {48 Kbps (50%), 64 Kbps (11%), 128 Kbps (17%), 256 Kbps (2.8%), 512 Kbps (4.4%), 1024 Kbps (4.3%), 2048 Kbps (3.5%), and 4096 Kbps (6.5%)} Percent of routers participating in Snark Torrenting: (0%, 20%, 80%, 100%) Snark Torrenting Roles: {Generators, Seeders}
Type IV: Sensitivity Analysis	Network size {500, 2000, 2100, 3000, 4000, 5000, 6000, 10000 nodes}, BW distribution type [explained in Section 6.3] {0, 1, 2}, Percent of routers participating in Snark Torrenting {0, 20%, 80%, 100%}, number of tunnels {3, 6}, length of tunnels {3,4}, number of torrents {0, 50, 100}, frequency of torrents {0, 0.0017, 0.0033}, frequency of new torrent subscriptions {0, 0.0008, 0.0033}, minimum torrent size {0, 10K, 1M bytes}.

Table 4-1. Experimental Parameter Settings Used in Section 5

5. DATA ANALYTICS

Once we finish running the Firewheel emulated experiments, we process and extract the CHR data from the experiment logs, and then analyze the CHRs across nodes and across experiments to identify significant effects.

5.1. Data Extraction from Running Firewheel Experiments

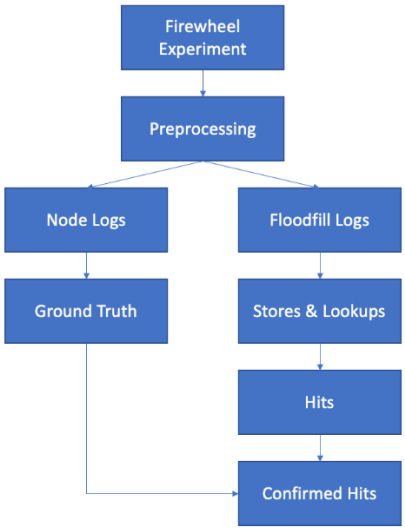

Figure 5-1. Process flowchart for data extraction

A process flowchart for the data processing performed to extract information from the I2P experiments in shown in Figure 5-1. The data analysis begins with processing the generated log files produced by the nodes in the Firewheel experiment. During each experiment run, the client I2P nodes capture timestamped records of their own RI *store* and *verify* lookup operations, including information about the RI key of interest and the node performing the operation. Each floodfill server provides records of all node *store* and *lookup* operations (both *verify* and normal *lookups*) that it observes, while each node logs the ground truth *store* and *verify* operations that it performs.

These log files are then processed using a combination of SQL (Structured Query Language) and Python to extract every *store* operation seen by the floodfill servers and then find the first *lookup*, seen by one of the floodfills being used that day by the node that performed the *store* (i.e. the floodfills being controlled by the attacker targeting that node). This pair of *store* and *lookup* operations, which occur together within an empirically determined 18 to 27 second window, are collectively called a hit. If this *lookup* corresponds to the ground truth *verify* operation recorded in the originating node's log, this hit is labelled as a confirmed hit. Otherwise, this *lookup* comes from a different node and the timing association was not successful, resulting in an unconfirmed hit. The output of this data extraction for all of the non-floodfill nodes is a table summarizing the total number of hits and confirmed hits for each node. This process also logs information about node log errors, as well as information for verifying the correct expected operation of the I2P network.

We also provide additional options for preprocessing the data to skip nodes with various types of errors in the logs. Logs can have errors due to data being corrupted, failure to complete operations

resulting in repeated *store* or *verify* attempts, unexpected termination of the node's host machines, or nodes not properly initializing and participating in the I2P network.

5.2. Data Analysis of the Confirmed Hit Rate per Node

We calculate the confirmed hit rates from the output of the data extraction process for each Firewheel experiment and analyze them to characterize the statistical behavior, focusing on three different sets of analyses that address three different aspects of the data: node and experiment characteristics, experiment runtime dependencies, and experimental configuration effects.

First, the initial analysis helps examine the standard statistical properties of all nodes in the experiment. Due to observations of non-normal behavior, these properties cover the population CHR mean, variance, standard deviation, minimum and maximum values, as well as median and first and third quartiles. We support these numerical results with histogram and bar plot visualizations to examine the distribution. These visualizations reveal the extent of the CHRs at the edges and the occurrence of unexpected bimodal behavior.

Second, with the dynamic behavior of the Firewheel I2P model, we consider the possibility of initialization effects and the time delay for the network to converge to more realistic behavior. This increases the runtime cost, as we must run experiments over longer durations, 3-7 days. We can then compare the statistical properties as before, for each day, to determine the minimum duration for experiments and to examine any changes in behavior over time. To do so, we augment our comparisons of experiment statistics from before with statistical tests and regression analysis to evaluate if the behavior is statistically different and to determine the trend over time. Specifically, we use the Tukey test for multiple comparison, the pairwise Kolmogorov–Smirnov test, and simple linear regression models.

Third, we explore different ways of parsing and grouping the data to analyze the effects of different experiment configuration settings, as enumerated in Table 4-1. With the difficulty in setting up experiments at the same size as the real I2P network, whose size also fluctuates, we set scale size as a primary parameter of interest. Moving beyond that, to have a sufficient number of parameters to motivate optimal experimental design considerations, we survey a range of nodes settings: bandwidth, I2P share percentage, number of cores, and torrenting role. We then apply the same statistical analysis and tests from the first two types of analysis above to groups of nodes with these varying parameters to determine the effect on CHR. Similarly, we also examine how the global configuration of an experiment affects the CHR. Some of these global parameters overlap with the node configuration, such as the distribution of node bandwidths and the percentage of nodes in each torrenting role. Other parameters explore the possible effects from the activity and behavior of the network, such as the number and lengths of tunnels used by the nodes, the number and size of the torrents, the frequency of torrent generation and requests.

6. EMULATED EXPERIMENTAL RESULTS, ANALYSIS, AND OBSERVATIONS

Our goal is to understand the variability in the confirmed hit rate and what factors might influence it. To do that, we performed several studies where we varied the size of the I2P network that was emulated, as well as configuration parameters such as the bandwidths for each node, the number of CPU cores, the way I2P encrypted traffic was generated, and other I2P configuration settings like the number and length of encrypted tunnels maintained by each I2P node. The results below present these experiments.

6.1. Exploratory Experimental Results

6.1.1. Type I – I2P Network Scales

First, we examined the effect of different network scales (numbers of routers) on confirmed hit rate, ranging from networks with as few as 150 nodes up to 10,000 nodes. To reduce the effects from other parameters, we set all the I2P routers to have the same bandwidth, 100 Kbps with 100% I2P share, and background traffic consisted of routers periodically downloading files from an Eepsite.

Figure 6-1 shows how *the median population confirmed hit rate decreases while the variance increases as the network scale becomes larger*. For each scale size, the median is shown in orange, while the variance is captured by the box showing interquartile range (IQR) as well as by the whiskers going out to 1.5 times the IQR. For clarity, we do not plot the outliers.

The median population confirmed hit rate decreases from 100% accuracy at the smallest scale (150 nodes) to 80% accuracy at our largest scale (10,000 nodes), while the interquartile range increases significantly from 0.6% up to 17.2%. We expect this trend to continue as network scale continues to increase. Network scale affects measurements of the confirmed hit rate, with Egger et al. conducting their measurement of u over different times of the day to mitigate the fluctuation in network size throughout a day (from 18,000 to 28,000 active routers) [9]. Such a measurement then could have an underlying uncertainty spanning a range of potential variances.

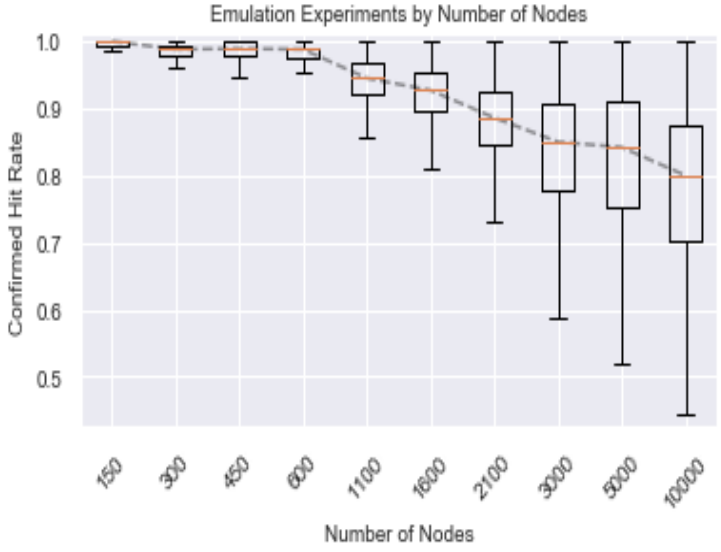

Figure 6-1. Median confirmed hit rate vs. network scale

6.1.2. Type II – Victim Groups

We consider the potential effects of local I2P router configuration parameters, given that the experimental setup for Egger et al. used groups of floodfill and victim nodes which were identically configured [9] against the context of the live I2P network. We examine how the local parameters for a small number of nodes representing a Victim Group might affect the confirmed hit rates of that group.

For tractability, we look at a single 3000 scale experiment using data from the third day to reduce boundary effects from initializing the system. We set around 2000 identically configured nodes with 40 Kbps bandwidth, 100% I2P share, and 1 CPU core for the default network behavior, and configure the remaining nodes into 42 groups of around 30 nodes with unique configuration combinations of bandwidth (ranging from 12 Kbps to 512 Kbps), I2P share percentage (between 10% and 100%), and number of CPU cores (1 or 8). To isolate the potential effects of these parameters, no background traffic occurs in this experiment.

As Egger et al. used a group of 6 victim nodes to verify the attack, we similarly sample nodes from each configuration combination to form and calculate the median confirmed hit rate. Given that our emulation enables control over the entire network, we sampled multiple times to produce 3 independent groups of 6 nodes for each configuration combination, as shown in Figure 6-2. The median confirmed hit rate for each Victim Group of 6 identically configured nodes is plotted against the effective bandwidth (router bandwidth * I2P share %) for each group, with color and shape denoting the number of cores. We see that the number of cores does not appear to show any significant effect, while a sharp difference appears to divide the confirmed hit rates for lower performance and higher performance nodes. Here, we can characterize performance in terms of effective bandwidth, which we compute by multiplying the bandwidth and the I2P share percentage. This represents the amount of bandwidth actually available for use in supporting the I2P network and is what governs the ordering of the parameters on the x-axis in ascending order. Using this effective bandwidth approach, we categorize nodes as lower or higher performing based on a threshold of 12 Kbps effective bandwidth. Within these two groupings, there does not appear to be any other dependence of the confirmed hit rates on the configuration.

The effect of lower versus higher performing nodes matches well with the I2P code itself, which marks these lower performing nodes as not being effective for supporting the network function. This places them in a special category and minimizes their use by peers within tunnels, which results in fewer *lookups* of these nodes. Here, based on that observation along with the observed higher CHRs, we hypothesize that one mechanism driving confirmed hit rates is the number of *lookups* for a node. While this particular set of parameters may be limited and the effect is very distinct, the implication is that the *local individual node configurations can impact the experimental confirmed hit rate.* Furthermore, by looking at the three different groups sampled from the same configuration, we observe that *there is a certain degree of variability inherent to the network itself.* While the lower performing nodes show more consistent median confirmed hits with smaller spreads of around 5%, the higher performing nodes can show a spread of over 10%. Consequently, a single Victim Group based on a single configuration could have a significant amount of underlying uncertainty, which might not be captured by one snapshot from the real system. One intrinsic benefit to emulation environments is the ability to collect data from all of the nodes in the experiment and to run replicate experiments to better assess the variability.

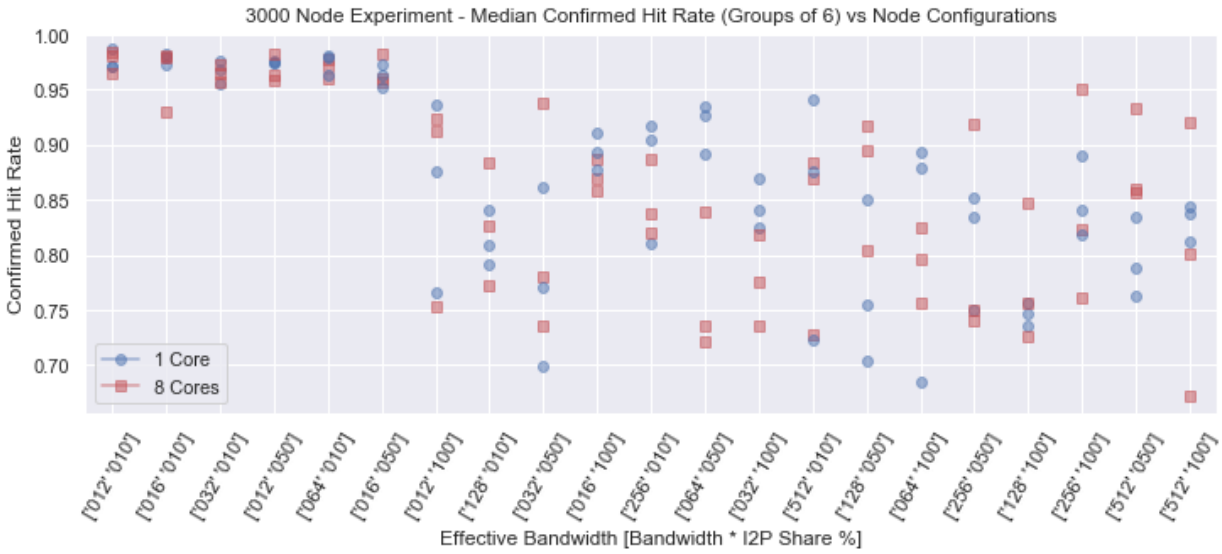

Figure 6-2. Median confirmed hit rates for sampled groups of 6 victim routers for each effective bandwidth rate.

6.1.3. Type III – Global Distributions

Given the greater degree of flexibility and control with using an emulation environment, we further extend our analysis to consider how the overall configuration of the entire I2P network affects observations of confirmed hit rates *u*. We again present results from the data of the third day from a single 3000 scale experiment, where we globally set the bandwidth parameter for all of the nodes within the network to approximate the actual bandwidth distributions of the real I2P network.

In addition, we include background traffic by configuring a high percentage of nodes (80%) to participate in I2P Snark torrenting. We set 20% of the nodes to act as torrent generators, who produce and share files but don't download them from others, and 60% as seeders, who both download files and share them with others, but don't produce new ones. Non-torrenting nodes are labelled as "I2P" nodes in the data. Guided by recent data from early 2021 about the distributions of the I2P network bandwidths [21], we proportionally place the nodes into categories with different performance specifications: 48 Kbps (50%), 64 Kbps (11%), 128 Kbps (17%), 256 Kbps (2.8%), 512 Kbps (4.4%), 1024 Kbps (4.3%), 2048 Kbps (3.5%), and 4096 Kbps (6.5%), all with I2P Share set at 100%. Given the results from Section 4.1.2, we do not include any lower bandwidths.

The boxplots for each combination of these parameters are shown in Figure 6-3, where the label indicates the corresponding parameter configurations. Here, we see that *nodes with bandwidths less than 512 Kbps tend to exhibit a higher confirmed hit rate than nodes with bandwidths equal or greater than 512 Kbps*. The lower bandwidth nodes also show a weak dependency on the bandwidth, while the higher bandwidth nodes all appear to exhibit a similar range of confirmed hit rates. We verify the effect of the global bandwidth parameter, in driving different confirmed hit rates of differently configured nodes within the overall population, visually but also using various statistical tests, as described in Section 5.2.

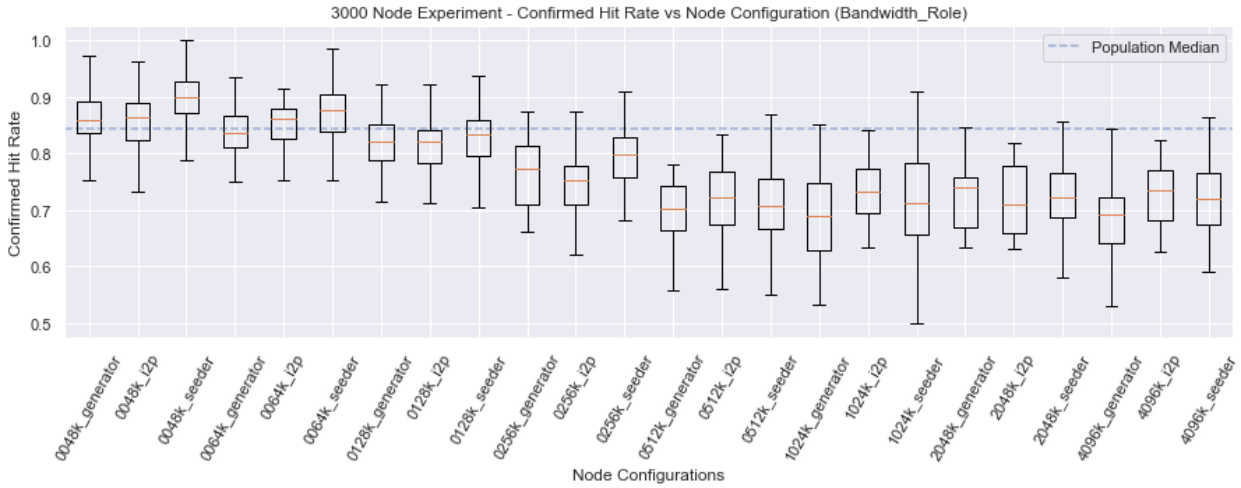

Figure 6-3. Confirmed hit rate boxplots based on the global parameter composition of the network. Outliers exceeding the range defined by the 1.5 times IQR whiskers have been removed, for clarity.

In Table 6-1, we show the first 10 rows of the results from applying the pairwise Tukey's honest significance test [47,48]. The Tukey method is an example of the main effects analysis mentioned earlier. We want to understand if the mean confirmed hit rate is statistically significantly different across the various node configurations shown in Figure 6-3. While there are some nuances, due to differences in the number of nodes for each parameter grouping, the results largely agree with the qualitative observation of an s-shaped curve from Figure 6-3. Of the 276 pairwise comparisons, 162 comparisons result in rejecting the null hypothesis and finding that there is a statistical difference in behavior between the parameter groupings. Similarly, other approaches such as the Kolmogorov–Smirnov statistical test (not shown) reveal comparable behavior.

The observation of this effect aligns with the special case noted in Section 6.1.2, where I2P minimizes the use and *lookups* to nodes with bandwidths under some threshold. With the wider range of bandwidths used here, we can refine our hypothesis and say that *the difference in node performance results in a difference in the number of lookups, which drives different confirmed hit rates.* As nodes operate on the network, they learn over time which other nodes provide tunnels with better performance (by Day 3 in our data). This learned preference drives biases in selecting tunnel routes, resulting in more *lookups* to nodes with higher performance specifications and fewer *lookups* to nodes with worse performance. As a result, as shown in Figure 6-3, the *higher bandwidth nodes exhibit lower confirmed hit rates, while the lower bandwidth nodes have higher confirmed hit rates.* Similarly, examining the full tables of results from our statistical tests shows that the higher bandwidth configurations do not show statistical differences among them, whereas the lower bandwidth configurations do tend to show significant statistical differences in behavior. It can also be noted that there is an unclear dependency on the torrenting role for routers, with some groupings of roles within the same bandwidth rejecting the null hypothesis. This indicates potential finer-grained configurations and additional parameters to consider which could affect the confirmed hit rate.

group1	group2	meandiff	p-adj	reject
0048k_generator	0048k_i2p	-0.0047	0.9000	False
0048k_generator	0048k_seeder	0.0368	0.0010	True
0048k_generator	0064k_generator	-0.0324	0.1056	False
0048k_generator	0064k_i2p	-0.0282	0.3720	False
0048k_generator	0064k_seeder	0.0107	0.9000	False
0048k_generator	0128k_generator	-0.0462	0.0010	True
0048k_generator	0128k_i2p	-0.0532	0.0010	True
0048k_generator	0128k_seeder	-0.0380	0.0010	True
0048k_generator	0256k_generator	-0.1002	0.0010	True

Table 6-1. First 10 rows of the Tukey test over configurations. Red highlights examples where the null hypothesis is rejected

Overall, we find through our emulation experiments that a number of factors can affect the confirmed hit rate and increase the degree of uncertainty in interpreting the power of an attack against the I2P network. Although some factors, as reported by Egger et al [9], such as the location and proximity of nodes do not appear to affect the confirmed hit rate, *network size has a dominant effect on confirmed hit rate* and effective bandwidth is also an important parameter. Interestingly, the effective bandwidth parameter shows an effect both as part of the configuration of nodes relative to the global configuration of the network, as well as for certain local configurations on their own. Most importantly, we find that the network itself exhibits a significant amount of inherent variability, which may add to the uncertainty in observations.

6.1.4. Exploratory Post-processing Parameters

In addition, post-processing assumptions within the data analysis approach can also affect the confirmed hit rates. We empirically determine the valid time window for tracking *hits*, finding that over 98% of all *lookups* occur between 18 and 27 seconds after the *store*, as shown in Figure 6-4. In comparison, Egger et al. note only that the *verify* lookup commences 20 seconds after the *store* event [9]. If we assume that their *hit* time window covers a range of 20 to 27 seconds, we can rerun our analysis and evaluate if this 2 second difference in the time window size matters.

To evaluate the effect of altering this post-processing parameter, we use the same data from Section 6.1.3 As shown in Figure 6-5, the 20 to 27 second time window (orange) results in slightly higher average confirmed hit rates as compared with our empirically determined 18 to 27 second time window (blue) across every configuration setting, along with a nearly uniform corresponding increase in the quartiles and the interquartile ranges. While the effect is comparatively small, the fact that we observe it impacting every node category in the experiment indicates that the decisions underlying the post-processing can also contribute to the overall experimental uncertainty.

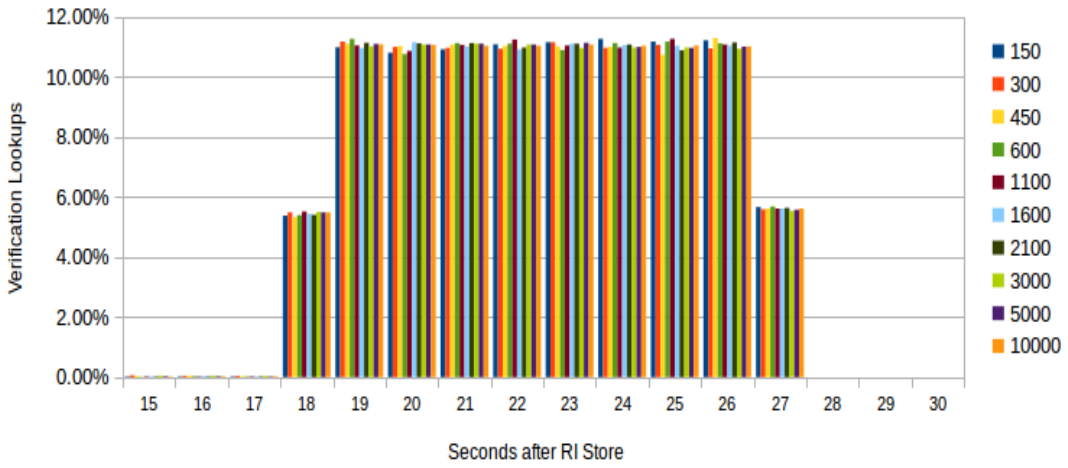

Figure 6-4. Percent of I2P verification lookups per second after RI store event, for network scales from 150 to 10,000 nodes.

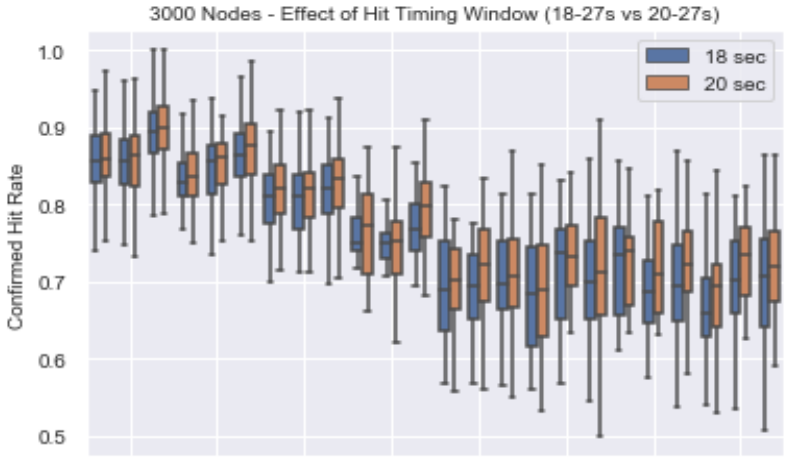

Figure 6-5. Comparison of the confirmed hit rates from Exp. III with post-processing analysis hit time windows of 18-27 seconds (blue) and 20-27seconds (orange).

6.2. Exploratory Analytical Results

The sections above describe the experimentation required to quantify the uncertainty in the confirmed hit rate, *u*. Once the uncertainty of *u* is characterized, it can be analytically used to calculate attribution probabilities. In the following we describe the analysis and results for two attribution probabilities: the probability of successful attribution within *k* observations of a *store*/*verify* pair, and the probability of observing *k* connections between an I2P client and a server in a 24-hour period.

6.2.1. Successful Attribution in k Trials

The confirmed hit rate *u* discussed in the above sections is the probability of successfully associating a netDb *store* request with a subsequent *verify* lookup request within a few second time window. Here, we are interested in understanding how multiple *store*/*verify* pairs would affect the

probability of successful de-anonymization over a time period such as a day. More specifically, we would like to know the probability that, given *n* observations of *store*/*verify* pairs, what is the probability that at least one observation is correct? This probability is represented by the following equation:

$$\Pr\{success\ in\ k\ trials\} = 1 - (1-u)^k,$$

where *u* is the probability of successful attribution per trial, and *k* is the number of trials.

Figure **6-6** shows the probability of successful attribution for the following cases: using Egger's observed result, i.e. $u = 0.52$, and using the uncertainty bounds from Section 6.1.3, i.e. $u = [0.693, 0.919]$. The horizontal dotted line shows the 95% success probability, and the vertical dashed lines show the number of trials required to achieve 95%. Rounding these values up to the next integer, we see that with Egger's result, 5 trials are required. With the experimental results described earlier, between 2 and 3 trials are required to achieve 95% confidence. It is important to note, however, that while attribution is 95% probable within 2, 3, or 5 trials, we do not know *which* trials are successful.

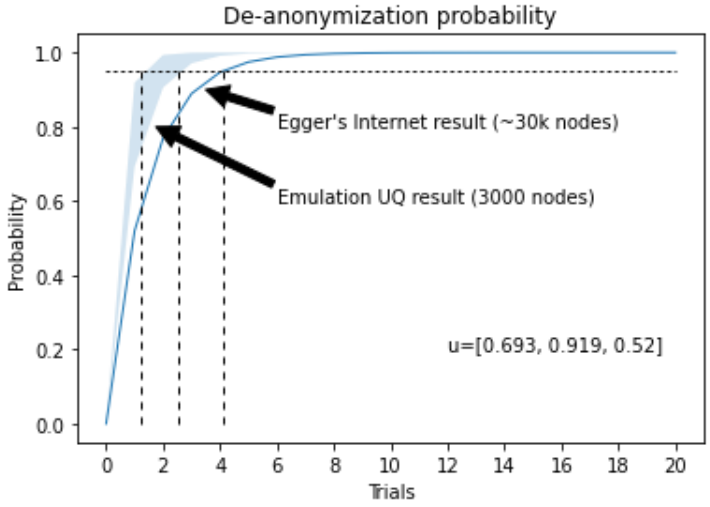

Figure 6-6. **Probability of successful attribution in k trials.**

6.2.2. Probability of k Connections in One Day

A more meaningful question to ask is, given the *store*/*verify* pair observations, what is the probability of observing a user accessing a given resource *k* times in one day? Egger et al. [9] models the probability of seeing *k hits* in *N* time slots as a binomial distribution:

$$Prob(k\ hits) = \binom{N}{k} x^k (1-x)^{N-k}.$$

In this equation, *x* is the probability of a *hit*, and includes a "correct" *hit* and a "false negative" *hit*. That is, $x = u * p + (1-u) * q$ where *p* is the fraction of time slots that the person accesses a resource R, and *q* is the probability that any other random user accesses R. In this work, we assume p=0.05 and q=0.001 following [9] but we also vary *p* parametrically to see the effect of different resource usage.

Figure 6-7 shows the effect of including uncertainty in *u* in the plots showing the probability of *k* or more *hits* as a function of the number of *hits* observed under different assumed resource usage values, *p*. Note that the probability of *k* or more *hits* can be calculated by summing terms from the previous formula:

$$Prob(k\ or\ more\ hits) = 1 - \sum_{j=0}^{k-1} \binom{N}{j} x^j (1-x)^{N-j}.$$

The results are shown in Figure 6-7. The upper plot has the deterministic value of $u = 0.52$ from Eggers' paper and the lower plot has the uncertainty bounds which incorporate the uncertainty in *u* from the analyses in Section 6.1.3. Specifically, we use the 10th and 90th percentile of *u* as well as the median: $u = [0.693, 0.844, 0.919]$. In this analysis, N=144 (one day of observations with 10-minute timeslots). Figure 6-7 shows that *including the uncertainty in u adds significant uncertainty in successfully attributing k connections in N observations, especially for larger values of p.*

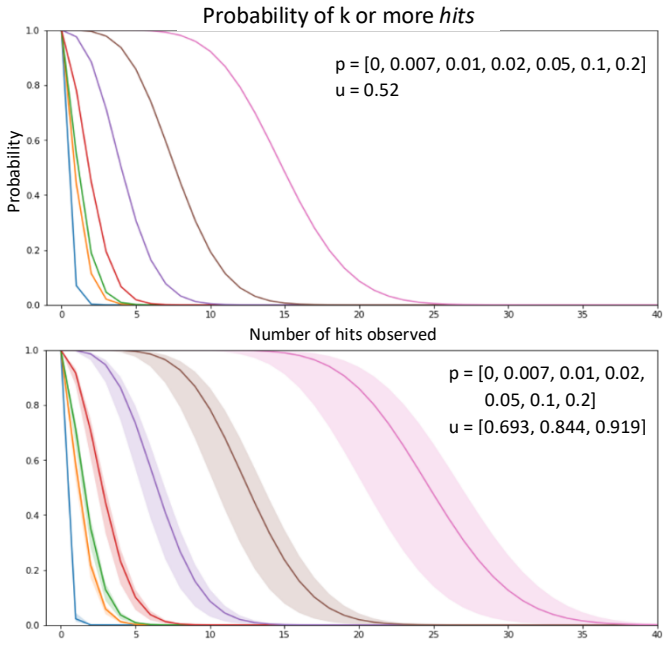

Figure 6-7. Probability of k or more hits in a day.

6.3. Sensitivity Analysis Experimental Results

In the spring of 2021, we added two capabilities to the Firewheel I2P model: torrenting, which represents Snark file sharing traffic on the I2P network, and the ability to change the distribution of bandwidths across the network according to a global distribution. Both of these global settings (the percentage of nodes engaging in Snark Torrenting, called the global snarking percentage, and global bandwidth distribution) appeared to influence the mean CHR [see Figure 6-13 and Table 6-3]. Because of the important role torrenting played, we further examined the number of torrents, the size of the torrents, the frequency of initial torrents and the frequency for new torrenting subscriptions. The effect of these variables is confounded (e.g. increasing the frequency of torrents can have the same effect as increasing the number of torrents and/or the size). However, we were able to observe a relationship where higher torrenting activity correlated with lower mean CHR,

as shown in the scatterplots in Figure 6-13 and the high correlation coefficients between the torrenting parameters and mean CHR as shown in the last row of Table 6-3.

A summary of the experiments included in our sensitivity analysis is shown in Table 6-2. Note that the experiments numbered 0 through 15 are experiments performed in which we did have the Snark torrenting ability for the I2P model. The previous experiments that were scaling experiments and did not involve torrenting traffic on the I2P network are listed as experiments number 1000-1003. These were included because they are important for scaling, with some of our few results at 5000 and 10K nodes.

Some explanation for the experimental settings listed in Table 6-2. Number of nodes refers to the number of routers on the I2P network. BWDistType refers to the bandwidth distribution type, where BWDistType = 0 is the empirical distribution observed in [21]: {48 Kbps (50%), 64 Kbps (11%), 128 Kbps (17%), 256 Kbps (2.8%), 512 Kbps (4.4%), 1024 Kbps (4.3%), 2048 Kbps (3.5%), and 4096 Kbps (6.5%)}, BWDistType = 1 is equal percentages over all the bandwidths, and BWDistType = 2 is the distribution we used in the original scaling studies (no global parameter tuning of bandwidth). Global Snarking is the percentage of traffic that was devoted to Snark Torrenting (e.g. 80 = 80% Snarking traffic). Number of torrents is the number of torrents used in Snark torrenting, and the frequency of torrents or of new torrent subscriptions is given as number of torrents/second. The minimum torrent size represented a lower bound on the size of the torrents (in bytes). We also varied number of tunnels and tunnel lengths in these experiments. However, the number of tunnels and tunnel lengths were not determined to be significant in terms of mean CHR, so we did not list them explicitly in Table 6-2.

Exp_No	Num of Nodes	BWDist Type	Global Snarking	Num of Torrents	Freq of Torrents	Min Size Torrents	Freq New Torrents	mean CHR
0	2000	0	100	50	0.0017	10000	0.0008	0.8852
1	2000	1	80	50	0.0017	10000	0.0008	0.9074
2	2000	0	80	100	0.0033	10000000	0.0033	0.6735
3	2000	0	20	50	0.0017	10000	0.0008	0.8830
4	2000	0	80	50	0.0017	10000	0.0008	0.8734
5	2000	0	80	50	0.0017	10000000	0.0008	0.7426
6	2000	0	80	50	0.0033	10000	0.0033	0.6871
7	2000	0	80	50	0.0017	10000	0.0008	0.8939
8	2000	0	80	50	0.0017	10000	0.0008	0.8978
9	2000	0	100	100	0.0033	10000000	0.0033	0.7016
10	4000	0	80	50	0.0017	10000	0.0008	0.7893
11	500	0	80	50	0.0017	10000	0.0008	0.9817
12	4000	0	20	50	0.0017	10000	0.0008	0.7986
13	2000	0	20	100	0.0033	10000000	0.0033	0.8101
14	4000	0	20	100	0.0033	10000000	0.0033	0.6960
15	6000	0	20	50	0.0017	10000	0.0008	0.7562
1000	2100	2	0	0	0.0000	0	0.0000	0.8742
1001	3000	2	0	0	0.0000	0	0.0000	0.8321
1002	5000	2	0	0	0.0000	0	0.0000	0.8257
1003	10000	2	0	0	0.0000	0	0.0000	0.7845

Table 6-2. Parameter settings and results for experiments used in sensitivity analysis

6.3.1. Summary of Results for Experiments 0-15

We analyze the data using the same tools as before, to extract the CHRs for each node in the network as well as the statistics characterizing the behavior of the overall network. Here, while we can again separate nodes based on their particular configuration settings, the focus is on how the torrenting configuration and global network parameters drive changes in the mean CHR.

First, as a summary of all of these experiments, we can plot the CHRs for the nodes and the mean CHR for each experiment, as shown in Figure 6-8. Each blue point represents the mean CHR for a node, although markers may overlap, while the orange marker indicates the mean CHR for all of the nodes in the experiment. Although this representation is not as detailed as histograms for showing the distribution of CHRs, the figure provides a concise summary of the behavior of all experiments. This side-by-side overview of experiments allows us to compare and make high-level evaluations of the effects of given parameters.

As a result, we can also note the interesting and somewhat unexpected effect of parameter combinations. For example, Experiment #9 is a 100%-high torrenting experiment with longer and more tunnels. Yet, its mean CHR falls between that for Experiment #2 (an 80%-high torrenting experiment) and Experiment #13 (a 20%-high torrenting experiment) and with less variance. We hypothesize that the increased tunnel setting actually reduces the effect of torrenting.

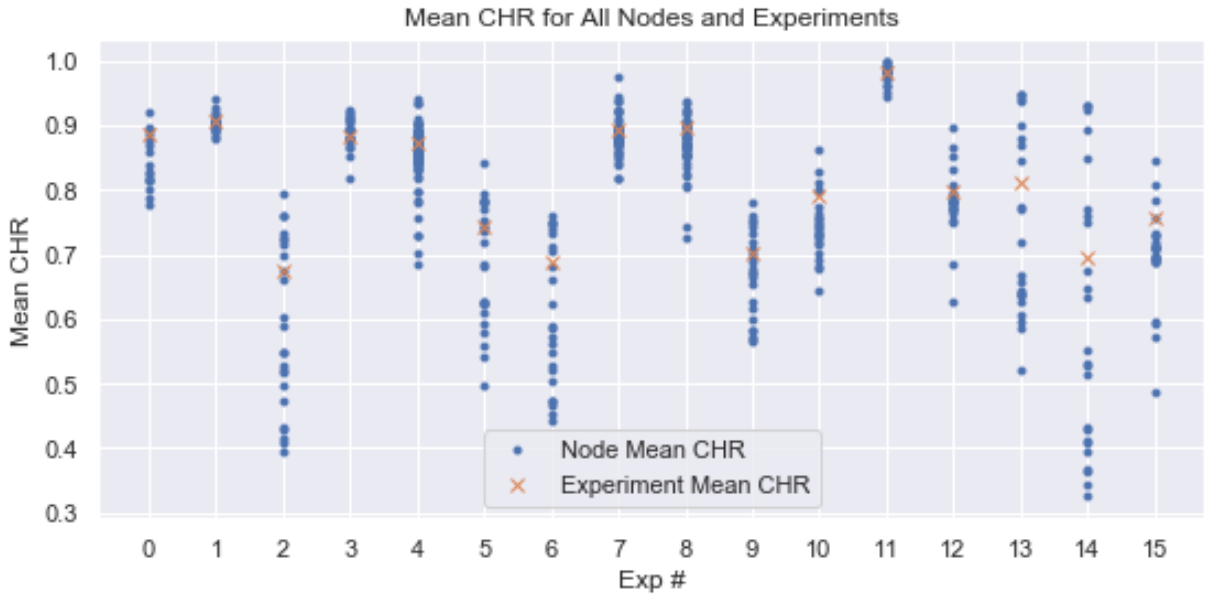

Figure 6-8. CHRs for nodes and the mean CHR for all sensitivity analysis (Type IV) experiments

A more straightforward and expected comparison would be evaluating the effect of the size of the experiment. With all other settings being identical, Experiments #3 and #15 differ only by the size of the experiment, 2000 nodes versus 6000 nodes respectively. As shown in Figure 6-9, the increase in scale size clearly results in a decrease in the mean confirmed rate, along with an apparent spread in the distribution of node CHRs. This effect is supported by the exploratory studies, where scale size was a significant driver of experimental CHR, as shown in Figure 6-1. For the studies with Snark torrenting (experiments 0-15 as listed in Table 6-2), most of the

experiments were performed with 2000 nodes, so node scale did not exhibit such a dominant role. When we examine both the exploratory studies and the Snark torrenting studies together, we do find node scale to be significant. This is discussed in Section 6.3 and in Figure 6-14.

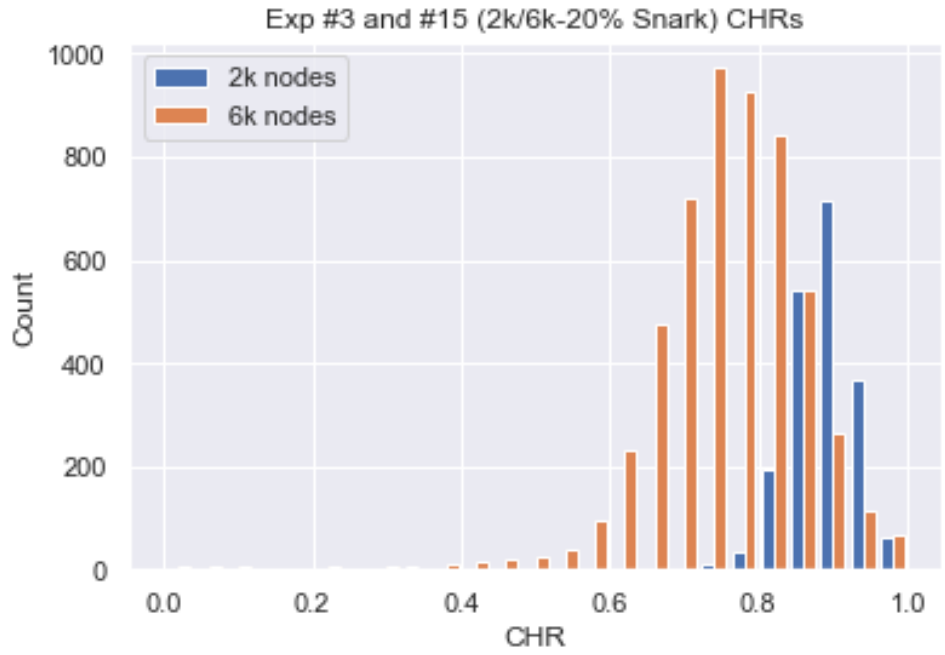

Figure 6-9. Histogram of node CHRs for Experiments #3 and #16

Similarly, another clear pattern shows that higher torrenting activity, with larger torrents generated and requested more frequently, significantly affects the resulting distribution in CHRs. We show a comparison, in Figure 6-10, between Experiments #12 and #14, which differ only by the amount of torrenting activity but show significant changes in the mean and variance of the node CHRs. Similar behavior is observed with other pairs of experiments, such as Experiments #3 and #13. Furthermore, examination of the data in Figure 6-8 shows that differences even in one aspect of torrenting also produces a similar effect. Experiments #5 (larger torrent sizes), #6 (torrents generated and requested more frequently), and #2 (80% of nodes participating in torrenting vs 20% for #13) likewise exhibit lower mean CHRs and higher variance.

Based on these experimental results, which covered a wider range of the experiment parameter space, we return to our earlier hypothesis that the number of *lookups* determines the success of the timing attack and whether the *store* and *verify* lookup activity is correctly associated. To validate our hypothesis, we extract all the *lookups* for each node in all of the Type IV experiments (as well as the Type I experiments with more than 2000 nodes). We then calculate the average number of *lookups* per node to remove the bias due to the size of the experiments and compare that against the mean CHR for that experiment.

As shown in Figure 6-11, there appears to be a strong inverse linear relationship, with a statistically significant ($p = 1.4e\text{-}6$) correlation coefficient of –0.86. This result, showing that more *lookups* per node correlates with lower CHRs, supports our hypothesis. The experiments with more *lookups* per node also suggest that scale size (10k nodes with 5,262 *lookups* per node) and torrenting

activity (higher torrenting frequency Experiment #6 with 6,283 *lookups* per node) are the primary drivers behind this mechanism.

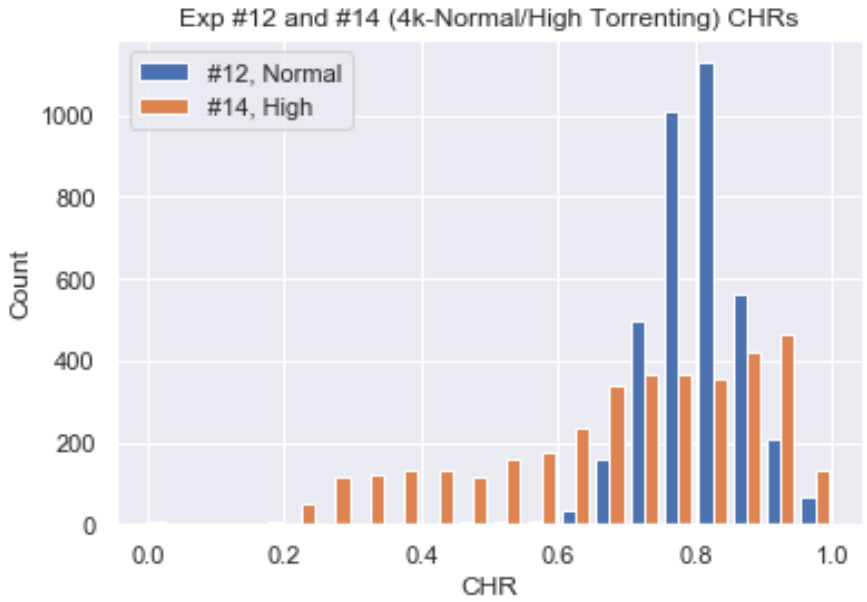

Figure 6-10. Histogram of node CHRs for Experiments #12 and #14

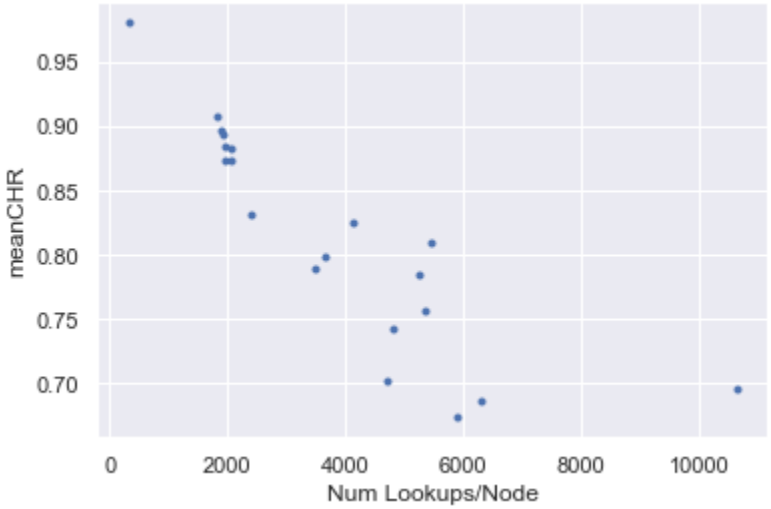

Figure 6-11. Mean CHR vs. Number of lookups per node

While examining the data, we note that there appears to be an outlier for Experiment #14, which has an average of 10,640 *lookups* per node. After considering the different experiment settings, we hypothesize that the effect results either from a combination of larger experiment size along with more torrenting activity or from having a higher percentage of nodes participating in torrenting. Based on the amount of available data, we focus on the second hypothesis, torrenting participation.

We split the data into 20% and 80% torrenting participation experiments and recalculate the correlation with the mean CHR. As shown in Figure 6-12, (which also includes the Type I 0% torrenting experiments and the Type IV 100% torrenting experiments for completeness), we note that the different amounts of torrenting participation appear to result in different groupings of

experiments which are each more strongly correlated. For 20% torrenting experiments, we obtain a correlation coefficient of -0.91 (p=0.03) and for 80%, we obtain a correlation of -0.99 (p=1.5e-7). These improvements in the correlation, which are both still statistically significant, suggest that Experiment #14 is not an outlier, but rather that our second hypothesis on torrenting participation is a likely explanation. Given the number of experiments for each torrenting participation setting, further experiments and analysis would be required to confirm our hypothesis.

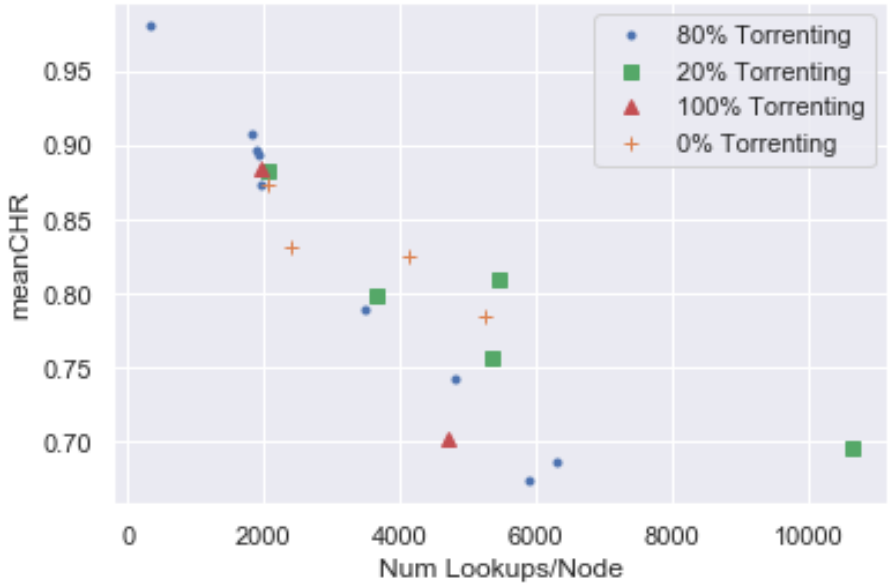

Figure 6-12. Mean CHR vs. Number of Lookups/Node, identified by percent torrenting

Note that measurement of *lookups* per node is an experimental outcome, strongly correlated with the confirmed hit rate, rather than an input parameter. As such, it is not used as part of the regression analysis. Instead, it supports our understanding of the mechanism behind the success of the timing attack and allows us to examine the relevant parameters, which affect the number of *lookups* performed.

6.4. Regression Analysis for Extrapolation

This recent set of experiments with torrenting was very useful because the data showed the significance of the parameters relating to torrenting and volume of traffic. It also allowed us to build a regression model to perform extrapolation to larger size networks than we were able to emulate on Carnac.

In this section, we present two analyses of the recent experimental dataset: (1) sensitivity analyses which identify significant parameters affecting the mean confirmed hit rate and (2) a regression model for extrapolation. In the graphs and analyses below, we use all of the data in Table 6-2 except experiment 11 with 500 nodes, because that was considered too small to be a realistic I2P network as evidence by a mean CHR of nearly 1 (0.98).

6.4.1. Sensitivity Analysis

The scatterplots shown in Figure 6-13 show the raw data: mean CHR is plotted against the various parameter values that were varied in these experiments as detailed in Table 6-2. Note that the point with the highest mean CHR is experiment 1, with the red square indicating an equal distribution of nodes in the I2P network across the various BW levels. The BW levels are 48 Kbps, 64 Kbps, 128 Kbps, 256 Kbps, 512 Kbps, 1028 Kbps, 2048 Kbps, and 4096 Kbps. Each of these 8 levels had 12.5% of the nodes for the "equal" Bandwidth distribution type (BW_DistType) = 1. In contrast, a BW_DistType of 0 (data shown in blue dots) had a distribution across nodes given by the I2P data that was available [21]. Note that the major difference between BW_DistType = 0 and 1 is that in the 0 setting, most of the nodes are very low bandwidth (e.g. 78% of the nodes are either 48 Kbps, 64 Kbps, or 128 Kbps). Finally, the BW_DistType = 2 represents the earlier scaling studies performed where the bandwidth distribution was sampled at levels between 8 and 512 Kbps. In summary, BW_DistType 0 is the typical distribution used for torrenting, 1 is equal distribution, and 2 is the scaling experiments with no torrenting.

Figure 6-13 shows a negative slope for all the parameters with respect to mean CHR. That is, as the input parameter increases (such as number of nodes, percent global snarking, number of torrents, etc.), the mean CHR decreases. We expect this, but the scatterplots and corresponding analyses support it. We also see that the slope of the line representing the relationship between the frequency of new torrent subscriptions and mean CHR (the right most box) is the largest negative slope: this variable has the most significant effect as will be also discussed in the regression plot. The parameters relating to torrenting have similar slopes, with the lines relating number of nodes to mean CHR and percent global snarking to mean CHR (two leftmost boxes) are less significant. Finally, we note that the trend lines shown in red in Figure 6-13 are univariate trends: they show only the trend of each parameter individually with respect to mean CHR. Because each subplot shows only 19 data points and the trend lines need to fit results that straddle a large range of mean CHR, these trend lines capture overall trends but do not predict individual experiments accurately in many cases.

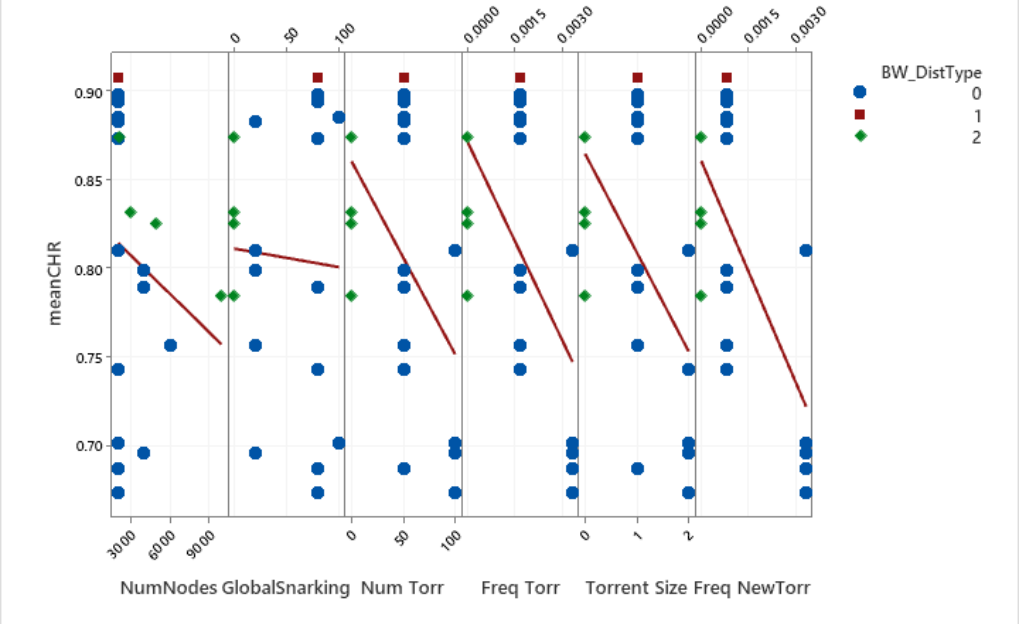

Figure 6-13. Scatterplots of inputs (x-axes) vs. mean Confirmed Hit Rate

Correlation coefficients are another method used for sensitivity analyses. Correlation coefficients vary in value between -1 (perfect negative trend, one variable decreases as another increases) to 1 (perfect positive trend: one variable increases as another variable does). A zero value of correlation typically implies that knowing the value of one variable does not help inform the value of another. The correlation coefficients for the data in Table 6-2 are shown in Table 6-3 below.

	Num_ Nodes	*BW_DistType*	*Global Snarking*	*Num Torr*	*Freq Torr*	*Torrent Size*	*Freq NewTorr*
Num_Nodes	1.00						
BW_DistType	0.44	1.00					
Global Snarking	-0.54	-0.63	1.00				
Num Torr	-0.41	-0.79	0.48	1.00			
Freq Torr	-0.43	-0.79	0.51	0.95	1.00		
Torrent Size	-0.43	-0.79	0.51	0.95	0.89	1.00	
Freq NewTorr	-0.33	-0.57	0.33	0.87	0.94	0.79	1.00
meanCHR	-0.19	0.24	-0.05	-0.46	-0.56	-0.50	-0.68

Table 6-3. Correlation Coefficients between I2P experiment parameters and mean CHR

In Table 6-3, the parameters highlighted in yellow show inputs which are strongly correlated. The bright yellow color indicates correlations with an absolute value greater than 0.5, while the pale yellow shows correlations with absolute value between 0.25 and 0.5. Generally, correlations greater than 0.25 are considered significant, with larger values being more significant. The bottom row of Table 6-3 shows the most important correlations between inputs and mean CHR. As the scatterplots indicate, the frequency of new torrent subscriptions is most strongly negatively correlated with mean CHR. Other parameters relating to torrents (number of torrents, frequency of torrents, and torrent size) are also strongly negatively correlated with mean CHR. The number of nodes, BW_DistType, and global Snark torrenting percentage are not as strongly correlated with mean CHR in this set of experiments.

6.4.2. Regression Model and Use in Extrapolation

We perform a set of regressions with various combinations of input parameters considered as independent predictors in a linear model to predict the mean CHR. For example, Table 6-4 shows that if we only have one independent variable in the prediction model, the frequency of new torrents is the best variable to choose as shown by an X in the first row: that generates a linear regression with an R-squared value of 46%. The R-squared value indicates the fraction of the variance in the output (in this case, mean CHR) that can be explained by the regression model based on the selected inputs. An R-squared value of 100% is the highest possible: that would indicate that the entire output variance can be attributed to a certain combination of inputs in the regression model. R-squared is a common "goodness of fit" measure used in regression analysis.

If we look down the rows of Table 6-4, we find there is a sweet spot where we are not including parameters with negligible effect, but we are achieving a high R-squared value. This occurs with

the following 4 variables as highlighted in yellow: number of nodes, number of torrents, torrent size, and frequency of new torrent subscriptions. The R-squared value for this regression is 80.3%.

Response is meanCHR

Vars	R-Sq	R-Sq (adj)	R-Sq (pred)	Mallows Cp	S	Num Nodes	BW_Dist Type	Global Snarking	Num Torr	Freq Torrent	Torrent Size	Freq New Torr
1	46.1	42.9	33.6	18.2	0.059293							X
1	30.9	26.9	14.9	27.5	0.067122					X		
2	65.2	60.9	47.0	8.4	0.049106	X						X
2	53.4	47.5	34.8	15.7	0.056863	X				X		
3	67.3	60.8	45.5	9.1	0.049166	X			X			X
3	66.1	59.3	43.1	9.8	0.050050	X				X		X
4	80.3	74.7	*	3.1	0.039460	X			X		X	X
4	69.5	60.8	4.6	9.8	0.049138	X				X	X	X
5	80.6	73.1	*	5.0	0.040720	X	X		X		X	X
5	80.4	72.9	*	5.0	0.040863	X		X	X		X	X
6	81.8	72.7	*	6.2	0.041009	X	X		X	X	X	X
6	80.6	70.9	*	6.9	0.042367	X	X	X	X		X	X
7	82.1	70.7	*	8.0	0.042457	X	X	X	X	X	X	X

Table 6-4. Subset regression indicating best variables to choose depending on the number of variables included in the regression equation.

The regression equation for the regression highlighted in yellow in Table 6-4 is shown below:

Regression Equation

meanCHR = 0.9219 - 0.000018 NumNodes + 0.00346 Num Torr - 0.1270 Torrent Size - 74.0 Freq NewTorr

Specific information about each coefficient estimated in the regression model is shown below:

Coefficients

Term	Coef	SE Coef	T-Value	P-Value	VIF
Constant	0.9219	0.0285	32.30	0.000	
NumNodes	-0.000018	0.000005	-3.65	0.003	1.23
Num Torr	0.00346	0.00107	3.22	0.006	14.76
Torrent Size	-0.1270	0.0417	-3.05	0.009	9.97
Freq NewTorr	-74.0	14.8	-5.01	0.000	4.15

We note that the p-values for all terms are very small and all less than 0.05. This indicates that all terms would be considered to have strong statistical significance in this model. Finally, we note that two of the terms, number of torrents and torrent size, have a Variance Inflation Factor (VIF) value that is high. The VIF is an indicator of multi-collinearity in a regression model. That is, it indicates there is correlation amongst the predictor variables of the model. We are aware of this issue: we saw high correlation amongst several of the predictors in Table 6-3. A high VIF (greater

than 5) indicates high correlation between predictors, causing the resulting regression to be less reliable. A regression result is considered more robust when it is built over independent input values with zero correlation. However, in this situation, we had limited time and experimental budget, so we will proceed with this particular regression model.

6.4.3. Extrapolation

We now present some results of the regression model when we use it to predict the mean CHR for larger scale experiments which we did not have time or resources to implement. That is, we plug in number of nodes = 10K, 15K, 20K, 25K, 30K in the regression model shown above, along with two sets of values for the other parameters relating to torrenting. We use a nominal setting of the torrenting values which corresponds to number of torrents = 50, torrent size = 10Kbytes, and frequency of new torrent subscriptions = 0.00083/sec. Then, we use a no torrent set of values which corresponds to number of torrents = torrent size = frequency of new torrent subscriptions = 0. For each of these parameter settings, we vary the number of nodes as indicated above to generate predictions. The predictions are shown in Figure 6-14.

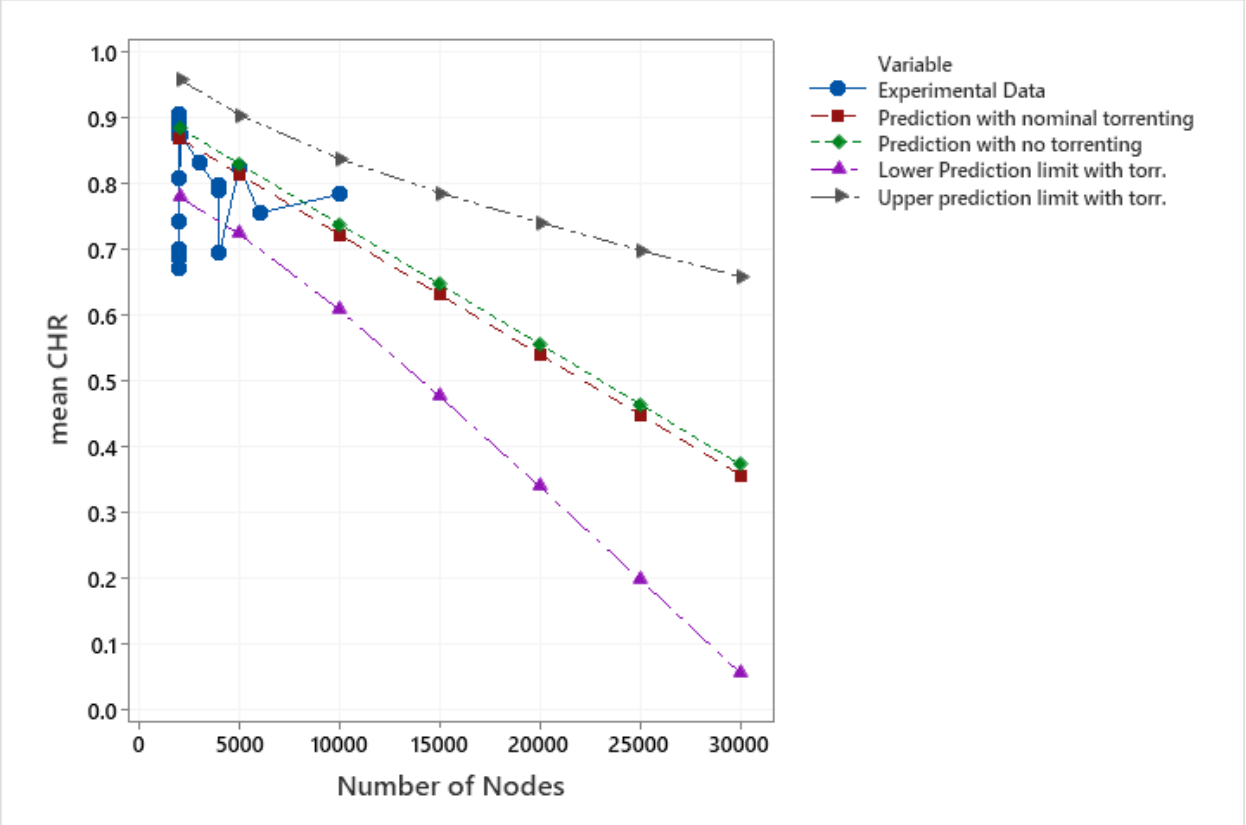

Figure 6-14. Regression predictions for two cases: nominal torrenting and no torrenting

As shown in Figure 6-14, the experimental data is shown in the blue dots. The prediction with nominal torrenting is given by the red line and the prediction with no torrenting is given by the green line. The prediction limits for the regression with nominal torrenting are shown in the grey and purple lines. These limits give the bounds of where a future realization of mean CHR may fall for a particular number of nodes. For example, at 15000 nodes, the prediction of the expected value of the mean CHR is 0.632 according to the regression model. However, this prediction has

a high uncertainty: a particular instance of the mean CHR at 15000 nodes may fall between 0.477 and 0.787. The prediction limits for the no torrenting case were very similar to the case with nominal torrenting: we did not plot them to simplify the graph.

Based on the regression equation, we calculate that a mean CHR of 0.522 will occur at 21000 nodes, with a prediction interval of [0.330, 0.715] around this estimate. Thus, the 52% value reported in Egger's paper using a realistic scale network is consistent with this estimate. Egger reported a mean CHR of 52% for a network size that fluctuated between 18K and 28K nodes.

7. MULTIFIDELITY EXPERIMENTS

Often, uncertainty quantification is challenging to perform because of the large number of samples that must be run through a cyber model, which can be computationally expensive. However, in multifidelity uncertainty quantification (MFUQ), many samples from one or more low-fidelity models (such as a discrete event simulation) are fused with a few runs of a high-fidelity cyber model (in this case, Firewheel) to decrease the estimator variance and obtain more reliable statistics. While we may only be able to run a few dozen samples of a high-fidelity model, we assume the cost of the low-fidelity model is much cheaper and so we can generate many low-fidelity samples for the cost of one high-fidelity model evaluation. The papers by Geraci et al. [13, 14] present the theory behind multifidelity UQ as well as demonstration of the methods to network applications.

In MFUQ, the multifidelity estimator for a mean of response quantity Q can be built starting from the single fidelity Monte Carlo (MC) mean estimate from the high-fidelity model and adding a weighted unbiased term to it:

$$\widehat{Q^{MF}} = \frac{1}{N}\sum_{i=1}^{N} Q_{high}^{(i)} + \alpha\left(\frac{1}{N}\sum_{i=1}^{N} Q_{low}^{(i)} - \frac{1}{r\times \mathrm{N}}\sum_{j=1}^{r\times N} Q_{low}^{(j)}\right) \qquad (7\text{-}1)$$

$$\widehat{Q^{MF}} = \widehat{Q_{high}} + \alpha\widehat{\Delta_{low}},$$

In Equation 7-1, N is the number of high-fidelity runs, and *r* is the oversampling ratio that allows for a maximization of the efficiency of the estimator by defining the optimal number of low-fidelity model evaluations as (N+1) x *r*. The first term on the right-hand side is just the usual mean estimate from the high-fidelity model. The second term is the low fidelity estimate "corrected" so that it is unbiased. Note that the second term has many more samples: this contributes to the variance reduction of the MF estimator. For a MF estimator with a single low-fidelity model, the coefficient α is obtained in closed form as function of the correlation and estimated variance of the two models.

Section 7.1 discusses the low-fidelity discrete event simulation model (DES) that was constructed to model the I2P network. Section 7.2 presents results from the MFUQ study and Section 7.3 provides an overview of optimal experimental design.

7.1. Discrete Event Simulation Model

Based on our high-fidelity model observations, we assume some simplified mechanisms driving the CHR and build a statistical DES model in Python to provide low-fidelity samples. Our model is a DES as it models *lookup* interactions in fixed, discrete time steps, and it is statistical in probabilistically mapping these modeled behaviors to CHRs. While this model runs significantly more quickly than the Firewheel experiments, the runtime increases with larger experiment scale sizes. Furthermore, we must tune the various parameters and hyperparameters of the DES model to ensure that the behavior is correlated with the high-fidelity model.

The statistical DES model has two phases: setup and run. The setup phase configures the number of nodes in the model, the distribution of node performance effects, the distribution of torrenting roles, the effects of torrenting on performance, the number of *lookups* done each round based on the number of tunnels, and local groups of nodes within which activity occurs. We use the

performance attribute here to directly determine the nodes to be preferentially selected for *lookup* operations, based on the observation that higher bandwidth nodes exhibit lower CHRs.

In the run phase, the simulation approximates the results from a 1-day long experiment, with 144 discrete rounds. Each round corresponds with a 10-minute time window to match the specification of having a pair of RI *store* and *verify* operations every 10 minutes. Within a single round, each node conducts *lookups* against nodes in their local group, weighted by their performance attribute. We assume three mechanisms driving the number of *lookups* that occur. First, we have the *store* and *verify* operation itself, which we define to have occurred and do not directly model. Second, we add *lookups* based on the number of nodes, to represent some *lookups* from background traffic needed to maintain and operate the network. Third, we increase *lookups* due to network traffic activity, based on the amount of torrenting. At the end of the round, we take the total number of *lookups* and use a statistical likelihood parameter to determine whether these additional *lookups* prevented the correct matching of the *verify* and *store* operation, resulting in an unconfirmed hit. We then aggregate these hits and confirmed hits over the entire simulation to produce CHRs for each node, with which we can conduct the same statistical analysis as with the high-fidelity Firewheel experiments for comparison.

7.2. Multifidelity Results for I2P

For a multifidelity framework to be effective, two conditions should be met: (1) the high and low fidelity models should show a strong correlation (e.g. if they are run with the same sample settings, the response quantities should be correlated) and (2) the cost of the low-fidelity model should be much cheaper.

First, we investigate the correlation. Recall we had 20 Firewheel runs as shown in Table 6-2. We ran the DES model at those parameter settings. The results are shown in Table 7-1 below, with the scatterplot in Figure 7-1 showing the results graphically. If the DES model were perfectly correlated with the Firewheel model, the dots would fall on a straight line. The correlation for these 20 points is 0.807. That is a reasonable correlation, large enough to proceed with the MF study. Two other details: the DES model had similar input parameters as Firewheel but they were not exactly the same. Thus, we had to map some of the settings in Table 6-2 to those in the DES model. Also, we note that the quantity of interest in this study is itself a mean (e.g. each row in Table 7-1 is the mean confirmed hit rate, where the mean is the average over the nodes studied in the experiment). We are interested in the overall mean (the average over all 20 samples as highlighted in the bottom of Table 7-1). That is, we are focused on the "mean of the mean CHR."

	HIGH FIDELITY	LOW FIDELITY
	Firewheel	DES
	meanCHR	meanCHR
	0.8852	0.8138
	0.9074	0.8468
	0.6735	0.6394
	0.8830	0.9345
	0.8734	0.8361
	0.7426	0.6335
	0.6871	0.6336
	0.8939	0.8452
	0.8978	0.8419
	0.7016	0.5573
	0.7893	0.7026
	0.9817	0.9598
	0.7986	0.8509
	0.8101	0.8567
	0.6960	0.7925
	0.7562	0.7613
	0.8742	0.9386
	0.8321	0.9069
	0.8257	0.8303
	0.7845	0.6746
Overall mean	**0.8147**	**0.7928**

Table 7-1. 20 samples of Firewheel and corresponding samples of DES model

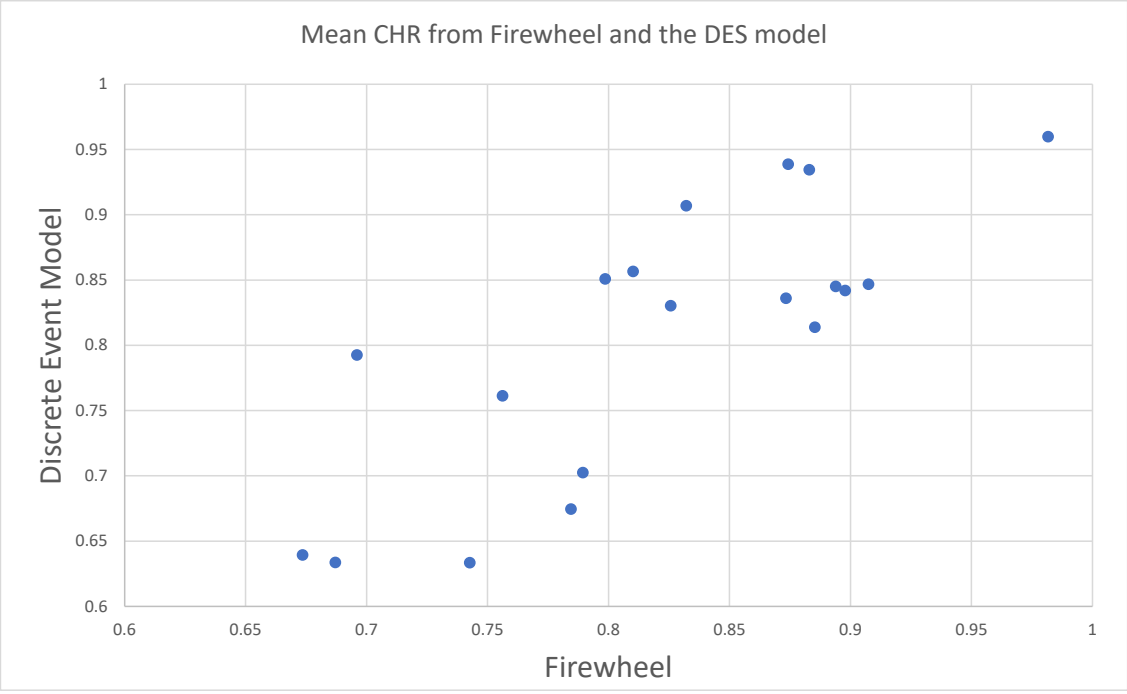

Figure 7-1. Scatterplot of 20 data points listed in Table 8-1.

Next, we consider the cost ratio. The cost to run the Firewheel experiments and process the results is approximately four days. The cost to run the DES model is 6.6 minutes, on average. This cost ratio only considers wall-time cost, not processor cost. Firewheel must be run on dozens of HPC nodes on the Carnac machine, whereas the DES model can be run on a laptop.

With this information, we proceed to the multifidelity calculations. The results are shown in Table 7-2 and Figure 7-2. In Table 7-2, we see that the mean of the high fidelity estimate of mean CHR is 0.8147, whereas the multifidelity estimate is higher, 0.8256. The variance reduction achieved by this process is approximately 58%. The coefficients α and $\widehat{\Delta_{low}}$ from Equation 7-1 are listed, along with the 99.7% confidence intervals on the mean estimate of the mean CHR (this is the mean value +/- 3σ. These confidence intervals are shown graphically in Figure 7-2. The cost of the Firewheel runs was 3 days for the first 16 experiments and 1 day for the last four experiments, resulting in (3*16+4)*24 = 1248 hours. We note that the cost of the 20 high-fidelity Firewheel runs only was 1248 hours, whereas the 220 low-fidelity DES simulations took 34 hours total. The cost of the low-fidelity model adds only 2% of the cost of the high-fidelity model while substantially reducing the variance of the mean estimate and narrowing its confidence interval. Note that the results from multifidelity uncertainty quantification are dependent on the relative costs of the models as well as their correlation: a low-fidelity model with even higher correlation that was even cheaper would result in larger variance reduction.

Q=mean CHR	Result	Number of High Fidelity Samples	Number of Low Fidelity Samples	Cost (hours)
High Fidelity Mean of Q: $\widehat{Q_{high}}$	**0.8147**	20		1248
High Fidelity Variance of $\widehat{Q_{high}}$	0.0003689			
Multifidelity Mean of Q: $\widehat{Q^{MF}}$	**0.8256**	20	220	1248+34
Multifidelity Variance of $\widehat{Q^{MF}}$	0.0001504			
Variance Reduction Achieved	58.02%			
Alpha coefficient α	-0.5993			
Delta estimate $\widehat{\Delta_{low}}$	-0.0182			
99.7% Confidence Interval for $\widehat{Q_{high}}$	[0.7579 , 0.8715]			
99.7% Confidence Interval for $\widehat{Q^{MF}}$	[0.7889 , 0.8624]			

Table 7-2. Results from Multifidelity estimate of the mean of the mean CHR

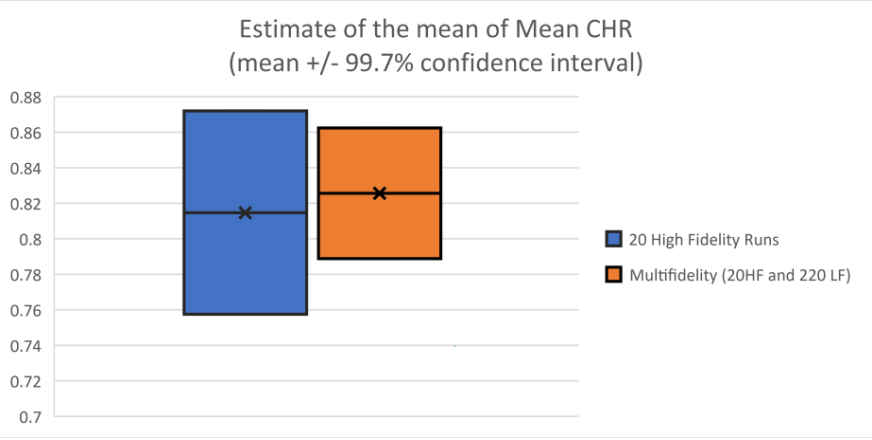

Figure 7-2. Mean estimate of mean CHR (middle line with X) and 99.7% Confidence Intervals

7.3. Optimal Experimental Design

As mentioned in Appendix A, section A.2, optimal experimental design refers to the creation of a run matrix (parameter values at which the emulation should be run) to optimize some property. For example, one property may be "space filling": one selects a design which creates sample points in the parameter domain so they fill out the space well (there are no "bunches" of points piled up in one location). Other properties involve maximizing some property of a surrogate which is constructed based on the points, for example, to minimize the maximum variance of the prediction values from a regression fit. This section presents what some OED designs would look like for the I2P study and why we were not able to use them (primarily because of computational cost of running the I2P experiments at scale).

In this section, we use the same notation as Appendix A. We denote the run matrix, X, to be of size n x p, where the n rows represent n runs, each with p parameter values. For the I2P case study, there are seven parameters, $p = 7$, if we remove the number of tunnels and tunnel lengths. We first consider a full factorial design. Full factorial designs can identify the main effects of each of the factors (parameters) on the outcome as well as all of the interaction effects between variables. Full factorial designs involve all combinations of all parameters at each of the parameter levels. For example, if there were two parameters and one had three levels and the other had four levels, a full factorial design would involve 3 x 4 or 12 runs. For our scenario, each of the parameters has three levels (e.g. number of torrents is either 0, 50, or 100), except number of nodes, which has 6 (2000 nodes, 3000, 4000, 5000, 6000, and 10000 nodes). Thus, the total number of combinations for a full factorial design is 6*36 = 4374. We ran a few dozen experiments through the course of this LDRD: 4374 is simply not feasible.

One class of experimental designs is called response surface designs. For example, to create designs which support linear models, parameters are sampled only at a high level and a low level (two values or settings allowed per parameter). The two values allow for a linear model. Most response surface designs use three levels per factor, to estimate quadratic effects in each parameter. Some examples of response surface designs are central composite designs and Box-Behnken designs. These both have parameters sampled at a central value and at high and low values. To illustrate such a design looks like, we created a Box Behnken design for 7 parameters in Minitab.

Note that the Box-Behnken design assumes that there are only 3 levels per parameter, so it is not strictly comparable to the full factorial design we calculated above: ideally, we would like the number of nodes to have more settings. But assuming that number of nodes would be limited to three settings (e.g. 2000, 6000, and 10000 nodes), the Box-Behnken design is shown in Table 7-3 below. It has 57 runs of the 7-parameter space. The -1 indicates the parameter value at its low value, 0 indicates a central value, and 1 indicates an upper or high value of the parameter. This 57 run design would allow us to create a model with quadratic effects in contrast to the regression model built in Section 6.4 which was a linear model in 7 parameters built over 19 points. Some of the interaction terms would be confounded, but the Box Behnken design would allow for some statistical analyses of significant effects. A 3-parameter version of Box-Behnken is shown in Figure 7-3. Note that the Box-Behnken design does not include corner points which may involve high values of one parameter as well as high values from another. This is a drawback to this design as compared with a full factorial, but it comes with much more efficiency.

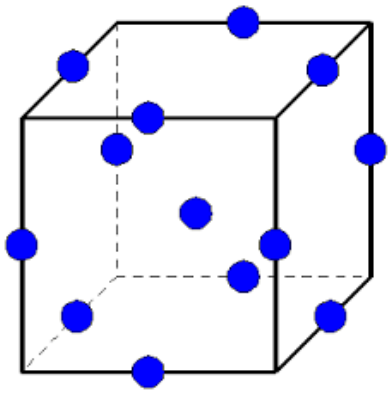

Figure 7-3. 3 parameter Box-Behnken design [from NIST, Section 5.3.3.6.2, https://www.itl.nist.gov/div898/handbook/pri/section3/pri3362.htm

Num_Nodes	BW_DistType	GlobalSnarking	Num_Tor	Freq_Tor	SizePackets	Freq_NewTor
1	0	1	0	1	0	0
-1	1	0	1	0	0	0
0	0	1	-1	0	0	-1
-1	0	1	0	1	0	0
1	0	-1	0	1	0	0
0	0	0	1	-1	-1	0
0	1	0	0	-1	0	-1
0	0	0	-1	-1	-1	0
0	1	1	0	0	-1	0
0	-1	1	0	0	-1	0
-1	0	0	0	0	1	1
-1	-1	0	-1	0	0	0
0	1	-1	0	0	-1	0
1	0	0	0	0	1	-1
1	0	0	0	0	-1	-1
0	0	0	-1	1	1	0
-1	0	0	0	0	1	-1
-1	0	0	0	0	-1	-1
0	0	0	1	1	-1	0
-1	-1	0	1	0	0	0
0	0	-1	1	0	0	1
0	0	0	1	-1	1	0
0	1	0	0	1	0	1
0	1	0	0	-1	0	1
0	0	0	1	1	1	0
0	1	1	0	0	1	0
0	-1	1	0	0	1	0
0	-1	0	0	-1	0	-1
-1	1	0	-1	0	0	0
0	1	0	0	1	0	-1
-1	0	-1	0	1	0	0
0	0	-1	-1	0	0	-1
1	1	0	-1	0	0	0
-1	0	-1	0	-1	0	0
0	-1	0	0	1	0	-1
1	0	1	0	-1	0	0
1	0	-1	0	-1	0	0
1	-1	0	1	0	0	0
1	-1	0	-1	0	0	0
1	1	0	1	0	0	0
0	-1	0	0	-1	0	1
0	0	1	1	0	0	1
-1	0	1	0	-1	0	0
0	0	1	1	0	0	-1
0	0	0	-1	-1	1	0
0	0	1	-1	0	0	1
0	1	-1	0	0	1	0
1	0	0	0	0	-1	1
1	0	0	0	0	1	1
0	-1	-1	0	0	-1	0
0	0	0	0	0	0	0
0	0	0	-1	1	-1	0
-1	0	0	0	0	-1	1
0	-1	0	0	1	0	1
0	0	-1	1	0	0	-1
0	0	-1	-1	0	0	1
0	-1	-1	0	0	1	0

Table 7-3. Example Box-Behnken Design for I2P study with 57 model runs.

For the I2P studies, we performed the experiments manually, typically changing one or two parameters which we thought would have the most significant effect. This is not ideal and is not a comprehensive experimental design approach. However, the cost of setting up the I2P model and debugging/deploying the runs on Carnac was very expensive: often it would take a week or two for each experiment (with 4-5 days of actual experiment time and the remainder postprocessing or debugging). Thus, we were not able to run the Box-Behnken study with 57 runs shown in Table 7-3, much less a full factorial with 4374 runs. Finally, we note that classical experimental designs tend to be insufficient for more than five to eight parameters. They have the additional limitation that the parameters are varied between a high and low settings (2 levels) or perhaps three levels, but not more than that. This is a limitation when deploying experimental design to practical applications.

8. LESSONS LEARNED AND BEST PRACTICES

Based on the three years of FORCE LDRD research and the effort designing, running, and analyzing the Firewheel emulation experiments, we present several lessons learned and suggestions for best practices:

- There are significant challenges with scaling emulation experiments to these sizes (2000-10000 nodes). One may only be able to afford a very small number of experimental runs, even smaller than the number of runs from an optimal experimental design.
- Instrumenting just one experiment (e.g. one run of the Firewheel I2P model with 3000 nodes, for example) on Carnac is a major effort requiring significant subject matter expertise e.g. for how many hosts to partition the experiment across, how many VMs to place on each host, for debugging the experiment to ensure the protocols are running properly, that the simulated IP networking is working properly, and that the log files are being created and extracted properly, etc. It's important to have a working relationship with colleagues who maintain the Firewheel, minimega, and Carnac technologies so you have someone to call on when things go wrong.
- Each emulated experiment requires that all the experimental input parameters are properly configured to ensure results represent those of the desired experiment. This can get complicated and sometimes confusing, so having a clear and complete experimental design for a suite of experiments (e.g. for a parameter sensitivity study) is critical to help keep the research on track.
- Although the number of experiments may be small, each experiment itself is rich in data. One can slice data by hour or day, by node or type of node, by type of traffic, by role of a component, etc. We obtained thousands of data points from each experiment: the question was how to aggregate these.
- When one parameter such as network scale dominates, it can be challenging to find other parameters which significantly affect a quantity of interest. We needed to rethink how we defined input parameters, from local to global, before we were able to identify additional parameters that significantly affected results, so be open to rethinking how you approach the problem.
- The size of the network made it difficult to see changes caused by various parameter settings if a parameter setting only affected a fraction of the nodes. We used coded hostnames and incorporated that info into logfile names to identify which nodes were configured with which parameter settings, then extracted that information from each node's logfile name when analyzing results.
- The I2P process we examined was sufficiently complicated that we could not easily create a mathematical model of it like we did with the Nmap scanning process under SECURE. Nmap was a fairly simple protocol for which we understood all of the "knobs" controlling it; we examined port scanning on a small number of nodes (24) where scanning was the only activity happening. This contrasts with thousands of nodes in the I2P network which were constantly changing their preferred neighborhood of other nodes to work with, the floodfill nodes they used, and the tunnels they used, etc. We could not scale down to 50 nodes because the global behavior of I2P would not be realistic at that scale. Determining

the scale at which a particular cyber model will behave as intended, and how long experiments must run before they can converge on behaviors approaching the real world dynamics of the system is very important to ascertain as quickly as possible, so you don't waste time running experiments that aren't big enough or for a long enough duration to be meaningful for your research.

- Statistical comparison tests such as main effects and Tukey multiple mean comparisons are useful for identifying statistically significant response effects as a function of changing input parameters. The tests are particularly helping when comparing large number of parameters with many settings.
- Traditional sensitivity analysis methods such as correlation coefficients, scatterplots, and regression modeling also were helpful to identify important parameters and develop models for extrapolation.
- More advanced UQ methods, such as multifidelity modeling, were viable for reducing the variance in the mean estimate of the quantity of interest, mean CHR. The exercise of constructing a low fidelity model was useful for the multifidelity exercise but also very helpful for extrapolatory studies.
- The discrete event simulation (the "lower fidelity") I2P model required significant tuning to work well. Efforts on improving its performance also motivated further research into the mechanism of the de-anonymization attack. Tuning the parameters associated with the associated mechanism resulted in analyzing the number of lookups/node, a feature that was found to be strongly correlated with the mean CHR.

9. SUMMARY

The FORCE LDRD focused on one specific case study: de-anonymization in the I2P network. Under this LDRD, we generated emulations at much larger scales (thousands of nodes) than the companion SECURE Grand Challenge. This allowed us to see challenges associated with these large emulations and address some of the issues relating to scaling. While the goals of FORCE and SECURE are similar, FORCE was complementary to SECURE in its focus on very large emulations. We anticipate that the framework established under FORCE for running such large emulations and processing, extracting, and analyzing extremely large datasets will be useful for the Emulytics community in years to come.

In terms of specific results for the I2P network, we present the following results:

- Using an emulated cyber experimentation environment, we have shown that there is inherent stochastic variation in confirmed hit rates (u) within an I2P network, regardless of configuration parameters, and thus the value of u is not stable for all routers across the network.
- We have confirmed that network scale has a large effect on the population median of u, as Egger et al. expected, but also on the variance of u. We have shown that some local router configuration changes (i.e. very low bandwidths) can have an impact on confirmed hit rates, while others do not (e.g. CPU cores, higher bandwidths), at least not when measured in isolation i.e. with no background traffic.
- Including the uncertainty in u adds significant uncertainty in successfully attributing k connections in N observations, especially for larger values of p, fraction of time slots that the person accesses a resource.
- We demonstrated that global changes (e.g. bandwidth distribution, encrypted traffic volume) can have a statistically significant impact on confirmed hit rates of like configured routers.
- We demonstrated that lookups/node is highly correlated with mean CHR, supporting our hypothesis for the mechanism driving the success of this de-anonymization attack
- Based on a regression equation constructed from our emulation studies, we calculate that a mean CHR of 0.522 will occur at 21000 nodes, with a prediction interval of [0.330, 0.715] around this estimate. Thus, the 52% value reported in Egger's paper using a realistic scale network is consistent with this estimate. Egger reported a mean CHR of 52% for a network size that fluctuated between 18K and 28K nodes.
- Finally, we demonstrated that changes in inputs do affect the analytical de-anonymization outputs, and that applying UQ to cyber experimentation can produce greater insights and higher confidence in analytical results.

REFERENCES

1. AIAA Standards: *Guide for the Verification and Validation of Computational Fluid Dynamics Simulations* (AIAA G-077-1998(2002)) Computational Fluid Dynamics Committee. https://doi.org/10.2514/4.472855
2. ASME V&V 20-2009 *Standard for Verification and Validation in Computational Fluid Dynamics and Heat Transfer.* https://www.asme.org/products/codes-standards/v-v-20-2009-standard-verification-validation
3. Balenson, D., L. Tinnel, and T. Benzel. "Cybersecurity Experimentation of the Future (Cef): Catalyzing a New Generation of Experimental Cybersecurity Research." *SRI International, Tech. Rep.* (2015).
4. Benzel, T., R. Braden, T. Faber, J. Mirkovic, S. Schwab, K. Sollins, and J. Wroclawski. *Current developments in DETER cybersecurity testbed technology*. In 2009 Cybersecurity Applications &Technology Conference for Homeland Security, pages 57–70. IEEE, 2009.
5. Crussell, J., Erickson, J., Fritz, D., and Floren, J. minimega v. 3.0, Computer software. Sandia National Laboratories, Albuquerque, NM, December 18, 2015. https://www.osti.gov//servlets/purl/1312788.
6. Davis, J. and S. Magrath. *A survey of cyber ranges and testbeds.* Technical report, Cyber and Electronic Warfare Division, Defence Science and Technology Organization, Australian Government, 2013.
7. Diegert, K., Klenke, S., Novotny, G., Paulsen, R., Pilch, M. and T. Trucano. *Toward a More Rigorous Application of Margins and Uncertainties within the Nuclear Weapons Life Cycle – A Sandia Perspective*. Sandia Technical Report SAND2007-6219.
8. Dykstra, J. *Essential Cybersecurity Science: Build, Test, and Evaluate Secure Systems.* " O'Reilly Media, Inc.", 2015.
9. Egger, C., J. Schlumberger, C. Kruegel, and G. Vigna, "Practical Attacks Against the I2P Network." In: Stolfo S.J., Stavrou A., Wright C.V. (eds) *Research in Attacks, Intrusions, and Defenses*. RAID 2013. Lecture Notes in Computer Science, vol 8145. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-41284-4_22
10. Ferguson, B., A. Tall, and D. Olsen. *National Cyber Range Overview*. In 2014 IEEE Military Communications Conference, pages 123–128. IEEE, 2014.
11. Gabert, K. "The Case of Misconfigured DNS in I2P", working document 2017.
12. Gabert, K., A. Vail, I. Burns, M. McDonald, S. Elliott, J. Montoya, J. Kallaher, T. Jones, and T. Thai, Firewheel – A Platform for Cyber Analysis, SAND2015-10324PE. Sandia National Laboratories, Albuquerque, NM. 2015. https://www.osti.gov/servlets/purl/1333803.
13. Geraci, G., Crussell, J., Swiler, L.P. and Debusschere, B. J. "Exploration of Multifidelity UQ Sampling Strategies for Computer Network Applications." *International Journal of Uncertainty Quantification*, Jan. 2021. Pp. 93-118. DOI: 10.1615/Int.J.UncertaintyQuantification.2021033774
14. Geraci, G., L.P. Swiler, J. Crussell, and B. Debusschere. *Exploration of Multifidelity Approaches to Uncertainty Quantification in Network Applications.* UNCECOMP ECCOMAS 2019. Thematic Conference on Uncertainty Quantification in Computational Sciences and Engineering. SAND2019-3274C.
15. Ghanem, R. And P. Spanos, *Stochastic Finite Elements: A Spectral Approach.* New York, New York: Springer Verlag (2002).

16. Hedayat, A.S., Sloane, N.J.A., and J. Stufken. *Orthogonal Arrays: Theory and Applications.* Springer Series in Statistics, 1999.
17. Helton, J.C. *Conceptual and Computational Basis for the Quantification of Margins and Uncertainty.* Sandia Technical Report 2009-3005.
18. Herley, C., and P. C. van Oorschot. "Science of Security: Combining Theory and Measurement to Reflect the Observable." *IEEE Security & Privacy* 16, no. 1 (2018): 12-22. https://doi.org/10.1109/MSP.2018.1331028.
19. Hoang, N., P. Kintis, M. Antonakakis, and M. Polychronakis. 2018. "An Empirical Study of the I2P Anonymity Network and its Censorship Resistance." In *Proceedings of the Internet Measurement Conference 2018* (IMC '18). Association for Computing Machinery, New York, NY, USA, 379–392. DOI:https://doi.org/10.1145/3278532.3278565
20. Hussain, A., D. DeAngelis, E. Kline, and S. Schwab. *Replicated testbed experiments for the evaluation of a wide-range of DDOS defenses.* In 2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), pages 46–55. IEEE, 2020.
21. I2P Metrics. https://i2p-metrics.np-tokumei.ne.
22. I2P. https://en.wikipedia.org/wiki/I2P
23. I2P. Invisible Internet Project. http://www.i2p2.de/.
24. Jones, S.T., Gabert, K. G., and T. D. Tarman. *Evaluating Emulation-based Models of Distributed Computing Systems.*SAND2017-10634. https://www.osti.gov/biblio/ 1398865-evaluating-emulation-based-models-distributed-computing-systems.
25. Liu, P., Wang, L., Tan, Q., Li, Q., Wang, X., & Shi, J. (2014). Empirical Measurement and Analysis of I2P Routers. *J. Networks, 9*, 2269-2278.
26. Maricq, A., Duplyakin, D., Jiminez, I., Maltzahn, C., Stutsman, R. and R. Ricci. *Taming Performance Variability*. 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI). 2018
27. Maxion, R. "Reproducibility: Buy Low, Sell High." *IEEE Security & Privacy* 18, no. 6 (2020): 33-41. https://doi.org/10.1109/MSEC.2020.3005077.
28. Maymounkov, P. and D. Mazières. Kademlia: A peer-to-peer information system based on the XOR metric. In Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02), pages 53-65, March 2002.
29. minimega. https://minimega.org.
30. Mirkovic, J., Bartlett, G. and J. Blythe. *DEW: Distributed Experiment Workflows.* USC Information Sciences. Proceedings from USENIX/CSET 2018 Conference.
31. Mirkovic, J., T. V. Benzel, T. Faber, R. Braden, J. T. Wroclawski, and S. Schwab. *The DETER project: Advancing the science of cyber security experimentation and test.* In 2010 IEEE International Conference on Technologies for Homeland Security (HST), pages 1–7. IEEE, 2010.
32. Morgan, M. G. and M. Henrion. *Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis.* Cambridge University Press, 1990.
33. National Research Council. *Assessing the Reliability of Complex Models: Mathematical and Statistical Foundations of Verification, Validation, and Uncertainty Quantification.* Washington, DC: The National Academies Press, 2012. https://doi.org/10.17226/13395.

34. Oberkampf, W.L. and C.J. Roy. Verification and Validation in Scientific Computing. Cambridge University Press, 2010.
35. Rasmussen, C.E. And C.K.I. Williams, *Gaussian Processes for Machine Learning*. MIT Press (2006).
36. Rossey, L.M., R. K. Cunningham, D. J. Fried, J.C. Rabek, R. P. Lippmann, J. W. Haines, and M.A. Zissman. *Lariat: Lincoln adaptable realtime information assurance testbed.* In Proceedings, IEEE Aerospace Conference, volume 6, pages 6–6. IEEE, 2002.
37. Sacks, J., W.J. Welch, T.J. Mitchell, and H.P. Wynn. *Design and analysis of computer experiments. Statistical Science*, 4(4):409–435, 1989.
38. Saltelli, A., Chan, K., Scott, E.M. *Sensitivity Analysis*. New York: Wiley; 2000.
39. Saltelli, A., Tarantola, S., Campolongo, F., Ratto, M. *Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models*. New York: Wiley; 2004.
40. Santner, T., B. Williams, And W. Notz, *The Design and Analysis of Computer Experiments*. New York, New York: Springer (2003).
41. Schwab, S. and E. Kline. *Cybersecurity experimentation at program scale: Guidelines and principles for future testbeds.* In 2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), pages 94–102. IEEE, 2019.
42. Seber, G.A.F. And C.J. Wild, *Nonlinear Regression*. New York, New York: Wiley & Sons (2003).
43. Siaterlis, C., A. Perez Garcia, and B. Genge. *On the use of Emulab testbeds for scientifically rigorous experiments.* IEEE Communications Surveys & Tutorials, 15(2):929–942, 2012.
44. Simpson, T.W., V. Toropov, V. Balabanov, and V. F.A.C. Design and analysis of computer experiments in multidisciplinary design optimization: A review of how far we have come or not. in *Proceedings of the 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. 2008. Victoria, British Columbia, Canada. AIAA Paper 2008-5802.
45. Stickland, M., J. Li, T.D. Tarman, L.P. Swiler. *Uncertainty Quantification in Cyber Experimentation*. SAND2021-5710C.
46. Swiler, L.P., Stickland, M. and T.D. Tarman. *Design of Experiments for Cyber Experimentation.* SAND2019-5640 C.
47. Tukey, John W. Comparing Individual Means in the Analysis of Variance. *Biometrics*, vol. 5, no. 2, 1949. pp. 99–114. *JSTOR*, www.jstor.org/stable/3001913.
48. Wikipedia contributors. "Tukey's range test." *Wikipedia, The Free Encyclopedia*. Wikipedia, The Free Encyclopedia, 19 Sep. 2020. Web.
49. Xiu, D., *Numerical Methods for Stochastic Computations: A Spectral Method Approach*. Princeton University Press (2010).

APPENDIX A. EXPERIMENTAL DESIGN TERMINOLOGY

In this section, we present definitions for experimental design that can support cyber experimentation on emulation platforms. We draw on the history of experimentation on computational simulations with the goal of providing guidance on how one specifies the parameters to be investigated, how one generates the sample set, and how one analyzes the results.

Currently the state of the art is to either do a "hero" calculation on a cyber testbed with an experiment that is run once or just a few times, or to perform a "grid" study where the parameter settings are discretized, and each combination of parameter settings is run. This latter can be computationally expensive due to the curse of dimensionality and thus methods requiring one to run only a subset of the complete enumeration of the parameter space are necessary. We discuss grid studies (e.g., full factorial designs) but also suggest other options in instances where the number of runs is much less than the number of parameter combinations that is possible. This reduces computational cost at the expense of understanding interaction effects between variables. Finally, some cyber experiments are run multiple times at the same settings (sometimes called replications or iterations) to understand the stochastic behavior of the system. The optimal number of replications is also a topic of interest, and this is a topic little discussed in the cyber experimentation community although there is some recent work demonstrating number of samples required to achieve a particular confidence interval for the mean or median of a set of runs [26].

A.1. Experimental Design

This is an overloaded term. It can be used to refer to how one selects input parameter settings for one experiment (e.g., which virtual environment to use, how many virtual machines, what protocols to run, etc.) The one experiment may involve multiple replications where the experiment is run multiple times at the same settings of the governing parameters.

However, Experimental Design is typically used to refer to the broader problem of selecting a set or suite of experimental parameter settings at which one will run the cyber experimental model (e.g., various choices of number of cores per machine, protocols and environment settings, packet size of traffic, bandwidth of links, etc.) This process of generating an ensemble of runs is also called Design of Experiments (DoE). The parameter values for each run should be carefully chosen to extract as much trend data from a parameter space as possible using a limited number of sample points. Additionally, each run may involve replicates if the emulation model is stochastic and exhibits random behavior upon repeated iterations of the same model settings. The ensemble of runs is used to perform sensitivity analysis and uncertainty quantification studies. For example, one might want to know which variables contribute the most to packet response time or determine the distribution of quantities of interest such as network latency, bandwidth, and memory.

A.2. Optimal Experimental Design

The selection of input parameters for design of experiments may be done in many ways. There are several criteria one can choose to optimize when selecting a design. For example, Monte Carlo sampling over the parameter domain generally tries to select points with good "space filling" properties. There are several "alphabet-optimal" designs such as A-optimal, B-optimal, D-optimal, G-optimal, and I-optimal. These designs all involve optimizing some property of the run matrix.

Let us denote the run matrix, X, to be of size n x p, where the n rows represent n runs, each with p parameter values. The information matrix is denoted by the inverse of the variance matrix, or $[X^T X]^{-1}$. An A-optimal design minimizes the trace of the inverse of the information matrix which results in minimizing the average variance of the estimates of regression coefficients built on the dataset X. A D-optimal design minimizes the determinant of the information matrix which results in maximizing the information content of the parameter values (this is also has the effect of good "space filling" properties). A G-optimal design minimizes the maximum variance of the predicted values from a regression fit built on the dataset X, etc. [40]

The above designs are fixed designs that seek to optimize a particular property. There are also adaptive designs in which an initial experiment is run and then an optimization procedure identifies the "next best" experiment to run to optimize some objective. Typically, the objective involves improving the parameters of the model and "gaining the most information" possible. For example, Bayesian optimal experimental design has become popular, with the goal of determining experiments which most inform the posterior distribution inferred on model parameter values. [40]

A.3. Uncertainty Quantification

Uncertainty Quantification (UQ) is the process of characterizing all uncertainties that could affect the results of the cyber experimental runs. Once the uncertainties are identified and characterized as "input uncertainties", they are propagated (e.g., mapped) through the experiment to obtain uncertainties on the results ("output uncertainties").

UQ is a closely related activity to V&V and essential for verifying and validating computational models. The goal of UQ is to propagate input distribution uncertainty through the model to generate distributions on the model responses. This can then be used to understand the mean and variance of the output, calculate the probability that the response is less than or greater than a particular threshold value, etc. UQ, along with V&V, enables modelers and analysts to make statements about the degree of confidence they have in their simulation or emulation-based predictions. Uncertainty quantification has been a fundamental capability supporting nuclear reactor safety studies, performance assessment of repositories for the disposal of nuclear waste, computational fluid dynamics for aircraft design, and climate model predictions [7-33]. We anticipate more widespread use of UQ in the cyber emulation community to address questions about the performance and confidence in mitigation strategies for network attacks, for example. However, emulated cyber environments are different from physics simulation models used in many risk assessments of engineered systems. We need to understand how typical UQ methods work in the presence of stochastic network behavior, and how to use UQ methods to identify "edge case" behavior where software, hardware, network topology, and vulnerabilities interact in unforeseen ways.

A.4. Sensitivity Analysis

Sensitivity analysis (SA) is the process of identifying the most significant factors or variables affecting the uncertainty of the Emulytics model predictions [38,39]. This can help identify where to most effectively place cyber threat mitigations or invest in resources. SA can be used to identify model inputs in which a reduction of uncertainty would most reduce the uncertainty of the model output, or to identify model inputs that could be fixed to simplify the calculation, or to identify general trends between inputs and outputs. SA can be performed using local or global methods.

A.5. Verification and Validation (V&V)

Over the past few decades, the computational simulation community has developed a strong emphasis on Verification and Validation activities to build credibility in scientific computing. A study by the National Research Council at the National Academies issued a report outlining the mathematical and statistical foundations of V&V and UQ as primary activities supporting the reliability of computational models [33]. A number of professional societies have developed guidelines and standards for V&V activities [1,2]. We take as definitions those outlined in [34]:

• *Verification* is the process of assessing software correctness and numerical accuracy of the solution to a given mathematical model.

• *Validation* is the process of assessing the physical accuracy of a mathematical model based on comparisons between computational results and experimental data.

Verification provides evidence that the model and the equations are correctly solved. In computational simulations, it deals with the adequacy of the numerical algorithms to provide accurate numerical solutions to the discretized partial differential equations. In cyber experimentation, it can refer to how accurately the virtualized software and hardware components represent their physical counterparts. Validation addresses a different question: the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model. Validation provides evidence that the cyber experiment is appropriate for the problem of interest. Validation typically involves measuring agreement between the experimental outcomes and "gold standard" outcomes from appropriately designed validation experiments running on actual networks or physical testbeds with no emulation. The extent to which validation can be performed on cyber experiment models and how to do it is an open research question [24].

A.6. Parameter Study

Typically, a parameter study means the same thing as an experimental design: it specifies a number of runs which involve varying the allowable levels of the parameters in a structured way.

A.7. Factorial Design

A factorial design is an experimental design that samples the full combination of all parameters. Thus, if there were 3 parameters and each had 5 allowable values or levels, a full factorial design would involve 5*5*5 = 125 runs. A fractional factorial design only involves a subset of the full factorial. The subset is typically chosen to best estimate the main effects of the parameter values.[16] There is a rich statistical literature based on orthogonal arrays that involves determining fractional designs. The approaches typically involve substantial computation, rely on libraries of pre-generated orthogonal arrays, and are mainly valid for combinations of variables only with two or three levels.

A.8. Replicates

A replicate refers to running the same set of experimental settings multiple times to see how the response varies within that setting. A replicate can also be called an iterate. The idea of replicates comes from the early experimental design literature. A common example is that of crop yields, where the parameter of interesting might be the application of fertilizer. A "replicate" would be one of several plots to which fertilizer was applied or one of several plots to which it was not

applied. Replication is needed when there can be significant variation within a treatment or combination of parameter settings.

A.9. Surrogates

Experimental design and UQ can both require huge numbers of model evaluations to generate accurate statistics or to perform sensitivity analysis. For this reason, the computational science community has embraced the notion of "surrogate models", also called emulators, meta-models, or response surface approximations. Surrogate models are used in computational models for physics and engineering applications to replace the "full physics code runs" which involve the solution of partial differential equations over very large (e.g., > 1M elements) meshes. In the past two decades, surrogate modeling for computational science problems has become an active research field.[44] Some of the most common surrogate models involve regression [42], Gaussian processes [35,37], and polynomial chaos expansions [15,49].

Cyber testbeds themselves may be considered surrogates for real-world environments. However, it is also possible to think of surrogates or lower-fidelity models for cyber virtualized experiments or emulations. For example, such surrogates could be regression models or other statistical data-fit models such as Gaussian processes. But surrogates for cyber experiments might also involve discrete event simulators such as the NS3 network traffic simulator. Finally, surrogates for cyber experiments might involve analytic formulas. One use of surrogates for cyber experiments is for "multi-fidelity" UQ. In this approach to UQ, a low-fidelity model is run thousands of times, where a high-fidelity model may be run a few times. The results are combined to produce a high-fidelity estimate which has the benefit of low variance from the large number of low-fidelity runs and improved accuracy from the high-fidelity runs which reduce bias in the estimate [14, 13].

A.10. UQ vs. Experimental Design

Note that there can be a subtle difference in how one treats the results of an experimental design study and a UQ study. Typically, uncertainty quantification requires the user to specify probability distributions on the input parameters (e.g., normal, Weibull, exponential, etc.). Then, samples are taken according to the probability distributions and the model is run at those settings to produce a distribution on results. Thus, uncertainty quantification focuses on mapping input distributions to output distributions: the goal is understanding the probability distribution of the output and associated statistics such as mean, variance, and percentiles of the output. Historically, experimental design methods do not require distributions. They are more focused on the influence of the input settings (often taken to be binary or discrete levels). Thus, the goal of experimental design is to say something like "the application of fertilizer results in a mean crop yield that is statistically significantly higher than without the fertilizer." Parameter studies, factorial studies, and parameter sweeps over levels of an input parameter typically are not focused on "distribution of inputs to distribution of outputs mapping" but instead on "what is the difference in response under various experimental settings?" or "what is the trend in the response as we increase the value of an input parameter?"

A confusing aspect of the distinction outlined above is that Monte Carlo sampling may be used for both UQ and experimental design. That is, often Monte Carlo methods are used to generate realizations of input parameters for UQ. However, Monte Carlo methods may be used to generate a small number of samples from a high dimensional space when a full factorial design or complete enumeration is too expensive. In the latter case, one does not necessarily impose a distribution

structure on the outputs. It is acceptable to use Monte Carlo methods for both UQ and experimental design studies, but the analyst should carefully state what assumptions are being made on the input distributions or levels of parameter values.

A.11. DoE for Physical vs. Computational Experiments

Statisticians classify DoE approaches into two different areas: classical Design of Experiment methods and the more modern design and analysis of computer experiments (DACE) methods. Classical DoE techniques arose from technical disciplines that assumed some randomness and non-repeatability in field experiments (e.g., agricultural yield, experimental chemistry). DoE approaches such as central composite design, Box-Behnken design, and structured factorial designs have approaches to generate and handle replicate runs. These designs also put sample points at the extremes of the parameter space, since such designs offer more reliable trend extraction in the presence of non-repeatability.

DACE methods are distinguished from DoE methods in that the non-repeatability component is omitted for computer simulations which are deterministic (e.g., one set of input parameters always results in the same output. This is usually the case for the partial differential equations models used to solve physical problems). Thus, for DACE experiments, there are no replicates. In these cases, space-filling designs and Latin hypercube sampling are more commonly employed to accurately extract trend information. Quasi-Monte Carlo sampling techniques which are constructed to fill the unit hypercube with good uniformity of coverage are also used for DACE. Space filling designs are also employed when constructing surrogate models, and much of the early DACE work centered around sampling to construct Gaussian process models [35,37]. Note that cyber experimentation involves aspects of both DoE (e.g., possible randomness and non-repeatability from stochastic network traffic, delays, timings, etc.) and DACE (large numbers of simulation parameters, need for good space filling designs).

In this report, we used simple experimental designs, mainly parameter studies, because of the large cost of the I2P Firewheel runs. We did perform replicate studies (repeating the experiment two more times) early in the project. This helped verify the repeatability of network configurations (topology, global and local settings, etc.), the repeatability of results (mean CHR), and allowed us to determine experiment run times (typically, we saw that response metrics were stable enough to be sampled on a daily basis but we ran the experiments for 3-5 days to ensure we had stable mean CHR results. Also note that each experiment itself involves a significant amount of random behavior across all the nodes that are tracked. By taking averages of activity within a node and across nodes, we account for the stochasticity of the experiments. Further discussion of particular experimental aspects and challenges is discussed in Section 8.3.

DISTRIBUTION

Email—Internal

Name	Org.	Sandia Email Address
Laura Swiler	1463	lpswile@sandia.gov
Daniel Turner	1463	dzturne@sandia.gov
Derek Hart	5621	derhart@sandia.gov
Michael Stickland	5682	mgstick@sandia.gov
Thomas Tarman	5682	tdtarma@sandia.gov
Jorge Urrea	5682	jmurrea@sandia.gov
Kasimir Gabert	5689	kggaber@sandia.gov
Justin Li	5953	jdli@sandia.gov
Kristina R. Czuchlewski	5953	krczuch@sandia.gov
Ali Pinar	8762	apinar@sandia.gov
Technical Library	01977	sanddocs@sandia.gov

This page left blank

This page left blank

Sandia National Laboratories

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.