NIST Publication #904961
Measurement Science for Complex Information Systems

AN EFFICIENT SENSITIVITY ANALYSIS METHOD FOR NETWORK SIMULATION MODELS

Kevin L. Mills
James J. Filliben

Information Technology Laboratory
National Institute of Standards & Tech.
Gaithersburg, MD 20899, USA

ABSTRACT

Simulation models for data communications networks encompass numerous parameters that can each take on millions of values, presenting experimenters with a vast space of potential parameter combinations. To apply such simulation models experimenters face a difficult challenge: selecting the most effective parameter combinations to explore, given available resources. This paper describes an efficient method for sensitivity analysis, which can be used to identify significant parameters influencing model response. Subsequently, experimenters can vary combinations of these significant factors in order to exercise a wide range of model behaviors. The paper applies the sensitivity analysis method to identify the most significant parameters influencing the behavior of MesoNet, a 20-parameter network simulator. The method and principles explained in this paper have been used to investigate parameter spaces for simulated networks under a variety of congestion control algorithms.

1 INTRODUCTION

Paxson and Floyd (1997) describe many difficult problems that impede simulation of large data communication networks, and recommend two main coping strategies: search for invariants and careful exploration of the parameter space. Unfortunately, typical network simulators (e.g., Fall and Varadhan 2009, SSFNet 2009, Tyan et al. 2009) use hundreds of parameters that can each take on millions of values. Several researchers (Riley et al. 2004, Yaun et al. 2003, Zeng et al. 1998) investigate parallel techniques as a means to simulate larger, faster networks. Unfortunately, such techniques do not reduce the parameter space, which remains difficult to configure and continues to require significant resources when conducting careful exploration. While reduced scale models, such as the 20-parameter MesoNet (Mills, Schwartz and Yuan 2010), can be easily configured, the parameter space still requires infeasible resources to explore fully. In this paper, we describe an efficient method for sensitivity analysis, which can be used to identify the most significant parameters influencing model behavior. This allows experimenters to explore a reduced set of parameter combinations by varying those parameters that contribute most to differences in model response. We demonstrate the sensitivity analysis method in the context of MesoNet, a network simulator implemented using SLX[1] (Henriksen 2000). Elsewhere (Mills et al. 2010) we use MesoNet to study a variety of congestion control algorithms proposed for the Internet. In that study, we conduct five simulation experiments, each of which subject the congestion control algorithms to only 32 simulated conditions. The parameter combinations that compose those conditions were selected based on findings from the sensitivity analysis described in this paper.

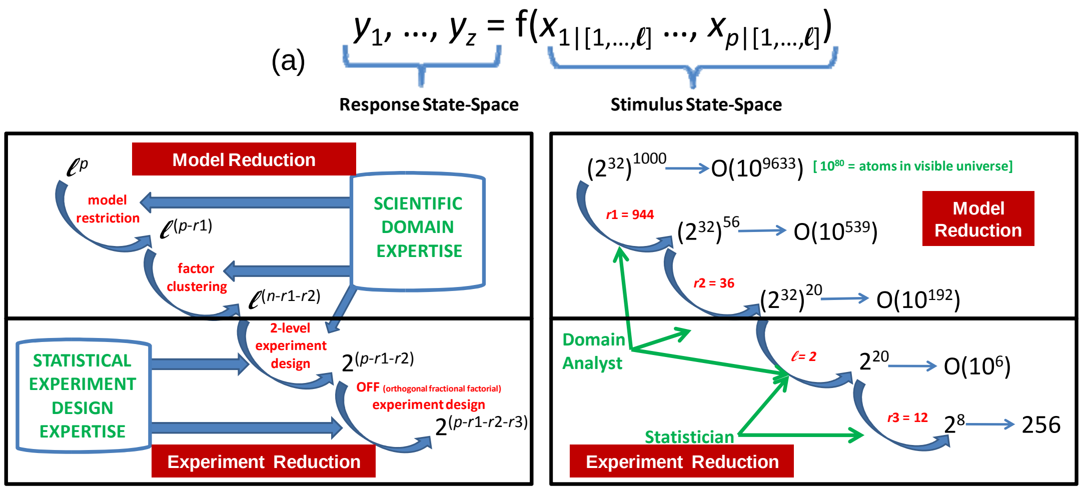
The paper makes two contributions: (1) describes an efficient method for sensitivity analysis of simulation models and (2) shows how the method can be applied to identify the most significant parameters influencing behavior in a network simulator. The ideas contained in this paper facilitate feasible exploration of the parameter space in large simulations and should improve the ability of researchers and practitioners to design efficient and effective simulation experiments.

The paper is organized in six main sections. In Sec. 2 we explain why the parameter space of simulation models can be difficult to explore and discuss some theoretical techniques for reducing the search space. We also show the substantial reduction we achieved applying our sensitivity analysis method to MesoNet. In Sec. 3 we identify and summarize MesoNet parameters described more fully elsewhere (Mills, Schwartz and Yuan 2010). Sec. 4 discusses the 2-level-per-factor orthogonal fractional factorial (OFF) experiment design technique that underlies our sensitivity analysis method. Sec. 5 explains how we applied 2-level OFF experiment design to conduct a sensitivity analysis of MesoNet. In Sec. 6 we describe three analysis techniques we used to characterize model behavior, and we identify the relative significance of MesoNet parameters. We conclude in Sec. 7.

[1] Any mention of commercial products within this paper is for information only; it does not imply recommendation or endorsement by NIST.

2 SEARCH-SPACE REDUCTION: THEORY & PRACTICE

As illustrated in Fig. 1(a), a simulation model can be viewed as a function transforming a set of p input parameters, x_1 to x_p, into a set of responses[2], y_1 to y_z. Each input parameter can take on a range of values, 1 to ℓ in our example, defining a parameter space of size ℓ^p, which can be very large. Fig. 1(c) shows the infeasible search space arising from a communication network model with p = 1000 parameters that can each take on $\ell = 2^{32}$ values.



(b) Theory of Search-space Reduction (c) Search-space Reduction Applied

Figure 1: (a) Functional representation of a simulation model; (b) Theoretical explanation of search-space reduction; (c) Search-space reduction applied to MesoNet simulation model

Fig. 1(b) illustrates two processes that can help reduce the search space: reduce the number of parameters in the model and reduce the number of parameter configurations through judicious experiment design. The process of model reduction involves two main steps. First, restrict model parameters to only that set of $(p - r1)$ factors relevant to the questions under investigation. Second, identify parameters that can be clustered together as facets of a single factor, leaving a reduced set of factors numbering $(p - r1 - r2)$. These two steps require expertise within the domain of investigation. In many cases, a reduced model parameter space remains infeasible to search, requiring two additional reduction steps to limit the number of experiments. The first step involves selecting only two levels to assign for each parameter – reducing ℓ to 2. Choosing appropriate levels requires domain knowledge. If the reduced search space of $2^{(p-r1-r2)}$ remains too expensive, then one can adopt an orthogonal fractional factorial (OFF) experiment design (Box, Hunter and Hunter 2005) to further reduce the space to $2^{(p-r1-r2-r3)}$, providing the most information possible for the available resources.

Fig. 1(c) illustrates the practical reduction we achieved in constructing a communication network model intended to compare proposed congestion control algorithms for the Internet. Assuming a detailed network model requires 1000 parameters, we identified 64 parameters germane to our investigation, achieving an initial reduction of $r1$ = 936. Subsequently, we grouped some of the 64 parameters together to create a reduction of $r2$ = 44, leaving the 20-parameter model that we summarize below in Sec. 3. In Sec. 4 we explain the theory underlying 2-level OFF experiment design, which allows us to reduce the search space further to (2^{20-12} =) 256 parameter combinations. In Sec. 5 we use these 256 combinations to conduct a sensitivity analysis of the network model.

3 THE 20-PARAMETER NETWORK MODEL

Table 1 identifies the 20 parameters composing MesoNet, our model of a communication network. We organize the parameters into five categories: (1) network configuration, (2) user behavior, (3) sources and receivers, (4) protocols and (5) simulation and measurement control. Here, we provide a summary of the parameters – more details can be found elsewhere (Mills, Schwartz and Yuan 2010). We defer until Sec. 5 a discussion of the contents of the Minus and Plus columns in Table 1.

---

[2] Determining which responses to examine is an interesting problem in its own right. Though not detailed in this paper, we used correlation and principal components analyses to select the responses used in our sensitivity analysis.

Table 1: MesoNet Parameters

Category	ID	Name	Minus (-1) Level	Plus (+1) Level
Network Configuration	X1	Network Speed	800 packets/ms	1600 packets/ms
	X2	Propagation Delay	1	2
	X3	Buffer Provisioning	*RTT* x *C*/sqrt(*n*)	*RTT* x *C*
	X4	Topology	Abilene - SPF prop. delay	ISP - SPF assigned costs
User Behavior	X5	Web Object Size (a = 1.5)	75 packets	150 packets
	X6	Proportion/Size of Larger Files (*Fx* = 10 *Sx* = 1000 *Mx* = 10000)	*Fp* = 0.02 *Sp* = 0.002 *Mp* = 0.0002	*Fp* = 0.04 *Sp* = 0.004 *Mp* = 0.0004
	X7	User Think Time	2 s	5 s
	X8	User Patience	0.0 (Infinite)	1.0 (Finite)
	X9	Selected Spatiotemporal Congestion	4th Time Period	None
	X10	Long-lived Flows	3 Start in 3rd Time Period	None
Sources & Receivers	X11	Probability of Fast Interface	0.2	0.8
	X12	Number of Sources & Receivers	2	3
	X13	Distribution of Sources	Web Centric	Peer-to-Peer Centric
	X14	Distribution of Receivers	Web Centric	Peer-to-Peer Centric
Protocols	X15	Probability of Algorithms	*TCP* = 0.8 *CTCP* = 0.2	*TCP* = 0.2 *CTCP* = 0.8
	X16	Initial Congestion Window Size	2 packets	8 packets
	X17	Initial Slow Start Threshold	43 packets	1 073 741 823 packets
Simulation & Measurement Control	X18	Measurement Interval Size	200 ms	1 s
	X19	Simulation Duration	25 minutes	50 minutes
	X20	Source Startup Pattern	Exponential (mean = X7)	50 % start early

A ***network configuration*** requires a topology (parameter X4) of routers and links, as shown for example in Fig. 2, adapted from the Abilene backbone network (Kratz et al. 2001). MesoNet supports topologies with up to three hierarchical router tiers: backbone routers (A-K in Fig. 2), point of presence (PoP) routers (A1-K2) and access routers (A1a-K2d). To model heterogeneity in network access, MesoNet allows three different types of access routers: **D**-class (e.g., six red nodes in Fig. 2, which connect directly to backbone routers), **F**-class (e.g., 28 green nodes) and **N**-class (e.g., 105 small gray nodes). Classifying access routers enables different speeds to be assigned to each class. Sources and receivers compose a fourth tier distributed below access routers. Packets flowing between a source-receiver pair follow a single ingress/egress path between an access router and a top-tier backbone router. In MesoNet ingress/egress paths are not subject to propagation delays. Propagation delays on backbone links are an intrinsic property of all MesoNet topologies, as are the paths taken by packets flowing among backbone routers. Given a cost metric for each link, one can use Dijkstra's shortest-path first (or equivalent) algorithm to generate least-cost paths. Assuming a link cost equal to propagation delay, the topology in Fig. 2 generates 110 backbone paths with an average length of 3.51 router hops. Adding in the hops for sources and receivers to reach the backbone routers increases the average path length to 9.43 hops. To scale propagation delays in a topology, parameter X2 multiplies the delays assigned to each backbone link. Unlike real networks, where links have transmission speeds and associated buffers, MesoNet assigns speeds to routers. Each router multiplexes packet forwarding from a single buffer shared among all attached links. Because MesoNet packets have no size, router speeds are assigned in units of packets/millisecond. Parameter X1 defines the base speed of backbone routers and all other router classes operate at a proportion of that speed: PoP routers 25 %, **N**-class 2.5 % , **F**-class 5 % and **D**-class 25 %. To provision router buffers, MesoNet allows buffer size (in packets) to be selected using an algorithm, specified by parameter X3.

Given a three-tier topology of routers and links, the model constructs a fourth tier, where ***sources and receivers*** are distributed under (and attached to) access routers. The model includes a target number of sources and receivers which should be set to a value appropriate for the network speed. Model parameter X12 serves as a multiplier to scale the target number of sources and receivers. Parameter X13 specifies probabilities (not shown here) that bias the distribution of sources so that a higher or lower proportion of the target number appear under various classes of access router. Similarly, parameter X14 specifies probabilities that bias the distribution of receivers. Altering the distributions of sources and receivers adjusts the probability of flows transiting access routers of specific classes, where the slowest access router crossed by a flow determines the flow's path class: very fast (VF) for **D**-class routers, fast (F) for **F**-class or typical (T) for **N**-class. The final property of sources and receivers concerns the maximum speed at which they can transfer packets to the network. The model allows two speeds: normal (e.g., 8000 packets/second) and fast (e.g., 80 000 packets/second). Parameter X11 specifies the probability that a source or receiver connects at the fast speed. When a flow's receiver and source are both connected at the fast speed, a flow's maximum rate is fast (F); otherwise a flow's maximum rate is normal (N).

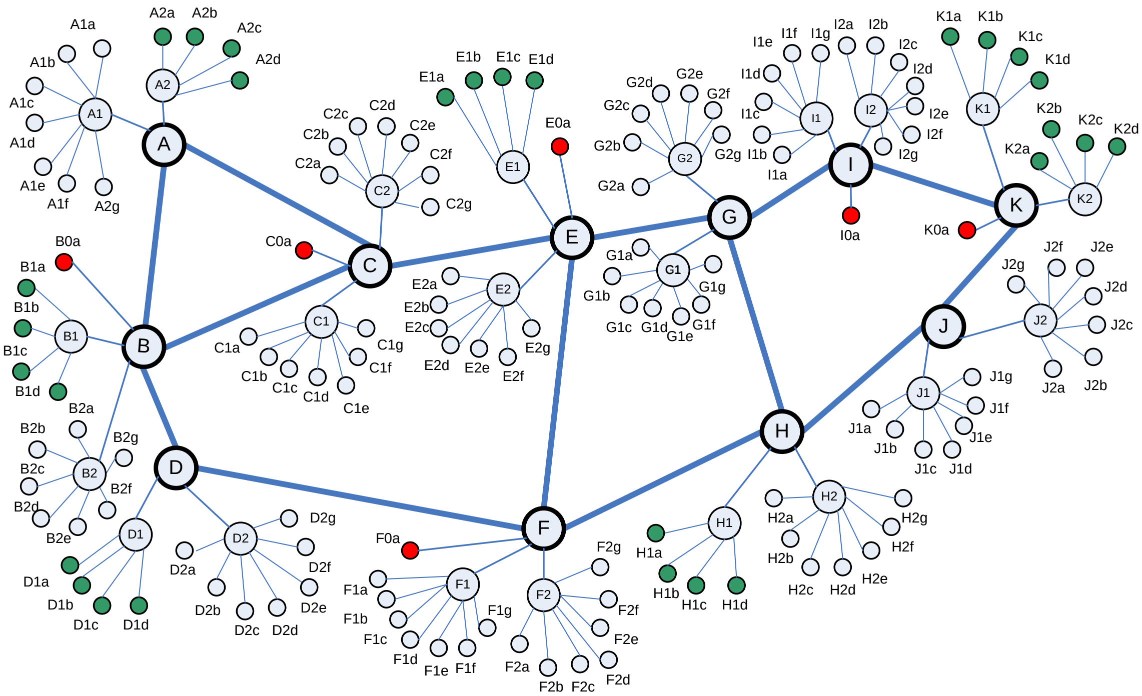


Figure 2: Three Tier Topology with 11 Backbone Routers (A-K), 22 Point of Presence Routers (A1-K2) and 139 Access Routers (A1a-K2d) – 6 red and 28 green Access Routers may operate at different speeds from the 105 others

User behavior is modeled through periodic activity by sources, which cycle between thinking, connecting and sending. Prior to entering the thinking state, a source selects a random residence time from an exponential distribution with a mean given by parameter X7. Upon expiration of residence, the source enters the connecting state, where a connection is attempted to a randomly selected receiver. If a connection attempt succeeds, the source enters the sending state, where a flow of packets is transmitted. Once all packets in a flow are acknowledged, the source reenters the thinking state. If a connection attempt fails, the source reenters the thinking state without sending. Sources may have finite or infinite patience. Parameter X8 specifies the probability that a source has finite patience, where short flows must be completed within a reasonable time and long flows must progress at a reasonable rate or else a source aborts the flow and reenters the thinking state.

Prior to sending, a source selects a Web object size (in packets) from a Pareto distribution with a mean defined by parameter X5. Through parameter X6, the model allows sources to transmit larger files in three categories: documents, software updates and movies, with corresponding multipliers (*Fx*, *Sx* and *Mx*) that scale the selected Web object to a larger size with a corresponding probability (*Fp*, *Sp* and *Mp*) for each category. The model also allows simulation of spatiotemporal congestion by specifying (parameter X9) a time period during which every flow transiting a very fast (VF) path will have the file size multiplied by 10, which creates spatiotemporal congestion on VF paths. The model also accommodates simulation of long-lived flows that, once activated, send as many packets as possible in the course of a simulation. Parameter X10 specifies the number, location and starting time for any long-lived flows included in an experiment.

The rate of each flow is regulated by ***protocols***. Upon connecting to a receiver, a source first sends a number of packets, known as the initial congestion window (*cwnd*), specified by parameter X16. As acknowledgments arrive from the receiver, the source increases *cwnd* exponentially. Upon the first lost packet, the source adopts procedures associated with a specific congestion avoidance algorithm implemented by the source. Model parameter X15 specifies the probabilities that a given source uses each of the congestion avoidance algorithms simulated by MesoNet. If there are no losses, a source switches to its congestion avoidance algorithm once the *cwnd* reaches an initial slow start threshold (*sst*), defined by parameter X17.

MesoNet measures numerous aspects of model behavior during each ***simulation*** run. Most ***measurements*** are made as time series, which sample system states at periodic intervals defined by parameter X18. Model parameter X19 ***controls*** the duration of a simulation run. Model parameter X20 determines the rate at which sources initially enter the sending state.

4 TWO-LEVEL ORTHGONAL FRACTIONAL FACTORIAL EXPERIMENT DESIGN THEORY

Even a simulation model as concise as MesoNet presents experiment designers with a significant challenge: deciding which combination of parameters to simulate from the $O(10^{192})$ possibilities. Two-level experiment designs, where each parameter (X_k, k=1,...,p) is assigned only two of its possible values, provide an immediate reduction in the search space to 2^p combinations. Restricting parameters to only two values has obvious limitations: only a small number of parameter values are explored and extrapolating from the results assumes a model behaves monotonically in the range between chosen values. On the other hand, adopting a two-level design provides some advantages (Box, Hunter and Hunter 2005): (1) requires few runs per parameter, (2) facilitates interpretation of response data, (3) identifies promising directions for future experiments (and may be augmented with thorough local explorations), (4) fits naturally into a sequential strategy, which supports the scientific method and (5) forms the basis for further reduction in parameter combinations through use of fractional factorial designs.

While two-level designs reduce the required number of simulation runs, a full factorial search of 2^p parameter combinations may still be infeasible. For example, a full factorial experiment design with MesoNet (p = 20) would require (2^{20} =) 1,048,576 simulations. Given that an average MesoNet simulation requires about 28 processor hours, and assuming that 48 processors are available, a full factorial experiment would take (2^{20} simulations x 28 processor hours/simulation over 48 processors =) about 611,670 hours, which is about 70 years. Adding processors could reduce the latency, e.g., to 3.4 years for 1000 processors or to 4 months for 10,000 processors, but the expense would remain constant, e.g., just under \$3M assuming processors cost \$0.10/hour. Thus the expense of a full factorial experiment would prove infeasible for most researchers, which means the number of simulations must be reduced to fit within time and budget constraints.

Reducing the time and cost of a full factorial experiment requires adopting a fractional factorial design, which simulates only a 2^{p-r}[3] subset of parameter combinations. While many experimenters adopt ad hoc techniques, such as factor-at-a-time (FAT) design, to select subsets of parameter combinations, orthogonal fractional factorial (OFF) theory (Box, Hunter and Hunter 2005) provides a principled approach to create designs, where the choice of 2^{p-r}[3] parameter combinations is made to achieve balance and orthogonality, which provide superior coverage and robustness and minimize variance in estimated effects[3]. Fig. 3 illustrates the difference in coverage between a 2^{7-4} FAT and 2^{7-4} OFF experiment design, where the sets of cubes represent a seven dimensional parameter space of (2^7 =) 128 possibilities and each red dot represents one of eight simulation runs comprising a specific combination of parameters from the space. The FAT design provides a localized exploration of the design space, while the comparable OFF design spreads the eight runs more globally throughout the space, thus injecting an element of robustness into conclusions derived from the experiments.

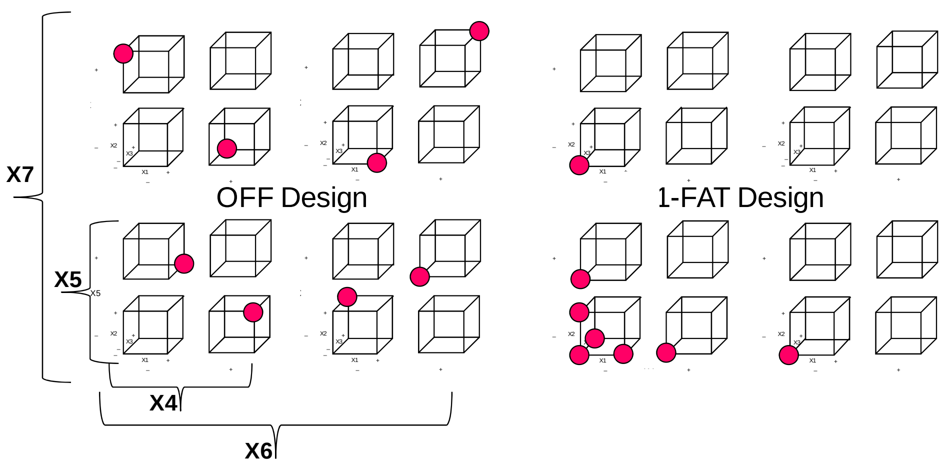


Figure 3: Schematic comparison of a 2^{7-4} OFF design and a 2^{7-4} FAT design – cubes represent the space of possible parameter combinations and red dots represent combinations chosen for simulation – each cube represents three of seven dimensions

In constructing an OFF design, an experimenter must ensure exploration of a sufficient number of parameter combinations to prevent confounding, which can cause confusion about the specific parameters responsible for variations in model responses. As a rule of thumb, experimenters should strive for at least "Resolution IV" (Box, Hunter and Hunter 2005) designs, which ensure no confusion among effects attributable to individual parameters and also prevent confusion about whether effects are caused by individual parameters or by interactions among parameter pairs. Further, Resolution IV designs

[3] In a two-level experiment, an effect is the mean model response when a parameter is set to one level minus the mean response when the parameter is set to the other level.

specify precisely which parameter pairs are confounded with which other parameter pairs. Typically, confusion involving specific parameter pairs can be resolved by a domain expert. A Resolution IV design must provide a sufficient number of simulations (n) to estimate a leading constant, each parameter (p) and each pair of parameters (p choose 2), or

$$n = 1 + p + \binom{p}{2}. \tag{1}$$

For example, MesoNet (p = 20) requires at least 211 (n = 1 + 20 + 190) simulations to construct a Resolution IV design. For a two-level design, choose the next higher power of 2 above n, i.e., 256 runs, which identifies the need for a 2^{20-12} design. Box, Hunter and Hunter (2005) give an algorithm for choosing combinations of parameters for various two-level designs. Applying the algorithm yields a design template, such as the partial (8 of 256 combinations) matrix shown in Fig. 4, where each simulation run (row) is defined as a combination of levels (either -1 or +1) for each parameter (column). For each parameter, an experimenter selects a specific value corresponding to each of the two levels and then substitutes the appropriate parameter value into the template to generate specific combinations to simulate. The resulting experiment design exhibits balance, i.e., each parameter has 128 -1 settings and 128 +1 settings, and orthogonality, i.e., each pair of parameters has 64 settings at each of (-1, -1), (-1, +1), (+1, -1) and (+1, +1). Balance minimizes variance in estimated effects, as illustrated in Fig. 5, which graphs the standard deviation for various 2^{20-12} fractional factorial designs.

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16	x17	x18	x19	x20
1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
2	1	-1	-1	-1	-1	-1	-1	-1	1	1	1	1	1	1	-1	-1	-1	-1	-1	-1
3	-1	1	-1	-1	-1	-1	-1	-1	1	1	1	1	1	1	1	1	1	1	1	-1
4	1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	1	1	1	1	-1
5	-1	-1	1	-1	-1	-1	-1	-1	1	-1	-1	-1	-1	-1	1	1	1	1	1	1
6	1	-1	1	-1	-1	-1	-1	-1	-1	1	1	1	1	1	1	1	1	1	1	1
7	-1	1	1	-1	-1	-1	-1	-1	-1	1	1	1	1	1	-1	-1	-1	-1	-1	1
8	1	1	1	-1	-1	-1	-1	-1	1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1

Figure 4: Template defining the first eight of 256 parameter combinations for a 2^{20-12} Resolution IV experiment design

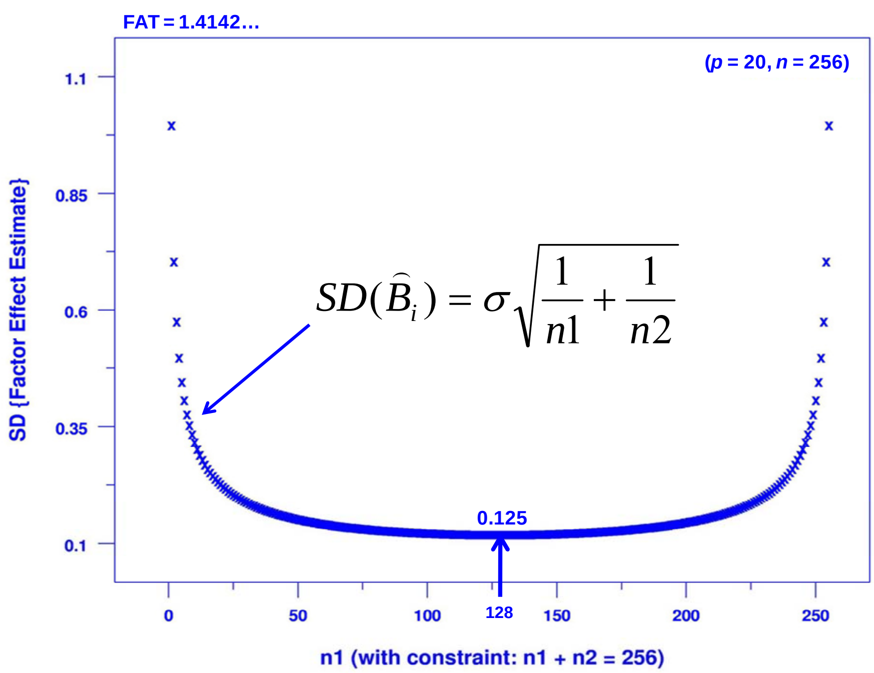


Figure 5: Standard Deviation (SD) in estimated effects for various combinations of the number ($n1$) of -1 settings and the number ($n2$ = 256 – $n1$) of +1 settings for a 2^8 fractional factorial design covering a 2^{20} space of parameter combinations – the standard deviation for a balanced design is 0.125, as compared to 1.4142… (i.e., √2) for a factor-at-a-time (FAT) design

5 EXPERIMENT DESIGN FOR MESONET SENSITIVITY ANALYSIS

Given the template from Fig. 4 and the Minus (-1) and Plus (+1) values from Table 1, we generated 256 parameter combinations, yielding a 2^{20-12} OFF design to form the basis for a sensitivity analysis of MesoNet. Most of the parameter mappings from Table 1 are straightforward. Here, we discuss a few mappings that merit more explanation. We also introduce the response variables used in the sensitivity analysis.

The -1 value for parameter X4 entails using the Abilene topology shown in Fig. 2. For the +1 value of X4 we used a larger topology adapted from a commercial Internet Service Provider (ISP). The ISP topology has more routers (16 backbone, 32 PoP, 8 **D**-class, 40 **F**-class and 122 **N**-class), more backbone links (24) and thus additional least-cost paths (240) in the backbone. The increased number (170) of access routers implies that the +1 topology will also have more sources and receivers than the -1 topology. Values for the X12 parameter scale the target number of sources and receivers under each access router in the selected topology. Backbone paths in the +1 topology are determined based on costs assigned by the ISP in order to achieve specific traffic engineering objectives. Both the -1 and +1 topologies have propagation delays corresponding to the physical length of backbone links. Values for the X1 parameter scale all router speeds in the selected topology. Values for the X2 parameter scale propagation delays on all backbone links in the topology. Values for the X3 parameter scale buffer sizes for all routers in the topology. The +1 value for X3 selects a buffer provisioning algorithm that corresponds to recommended practice (Bush and Meyer 2003), i.e., a router's buffer size in packets is the average round-trip time (*RTT*) in a topology multiplied by the router's speed (*C*). Following the suggestion of some researchers (Appenzeller et al. 2004), the -1 value for X3 divides a router's computed buffer size by the square root of the expected number (*n*) of flows transiting the router.

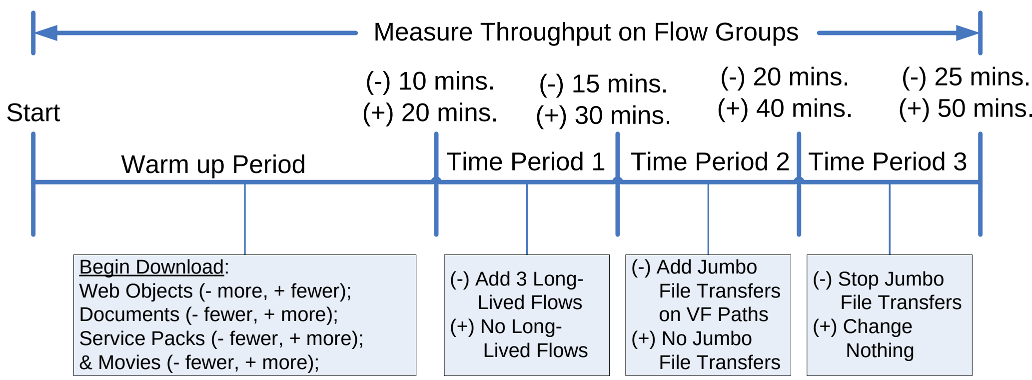


Figure 6: Possible traffic scenarios generated by various combinations of values for parameters X5, X6, X9, X10 and X19

Several parameters influence network traffic generated by sources, as illustrated in Fig. 6. Each simulation run can be viewed through a time line with length corresponding to the simulation duration assigned via parameter X19: 25 (-1 level) or 50 (+1 level) minutes. The simulation begins with sources sending files of various sizes, as determined by the values of parameters X5 and X6. The -1 value for X6 denotes transfer of fewer larger files, i.e., documents, software service packs and movies, which implies the transfer of more Web objects. The +1 value for X6 increases the number of transfers of larger files and decreases the number of Web objects. After a warm up period of either 10 (-1 for X19) or 20 (+1 for X19) minutes, the scenario unfolds over three additional time periods, each with a duration of either 5 (-1 for X19) or 10 (+1 for X19) minutes. At onset of the first time period three long-lived flows are started if X10 is -1. The long-lived flows are not started if X10 is +1. At onset of the second time period transfer of jumbo files may be started (-1) or not (+1) on VF paths, depending on the level of X9. At the onset of the third time period no further jumbo files will be initiated.

A few other parameters merit mention. Parameters X13 and X14 vary the distribution of sources and receivers in a topology, which influences the proportion of flows transiting specific access router classes. The -1 value for these parameters create Web centric traffic, which means an increase in proportion of flows transiting **D**-class and **F**-class access routers. The +1 value for these parameters increase the proportion of flows that transit **N**-class access routers, which is more consistent with peer-to-peer traffic. For this experiment, sources may regulate flow transmission rate using one of two congestion avoidance algorithms: the standard transmission control protocol (TCP) or compound TCP (CTCP), an alternative developed by researchers (Tan et al. 2006) at Microsoft. A -1 value for parameter X15 deploys more TCP sources in a topology, while a +1 value deploys more CTCP sources. Finally, a -1 value for parameter X20 causes sources to leave the initial thinking state after exponential delays with a mean determined by parameter X7. The +1 value for X20 causes 25 % of sources to start in the connecting state and 25 % to leave the initial thinking state early, while the remaining 50 % leave after a normal delay.

As shown in Table 2, we characterized MesoNet behavior by measuring 18 macroscopic responses, summarizing network state in six categories, and by averaging throughput (in packets/second) for each of 24 flow groups, where a flow group is defined by three dimensions: (1) file size, (2) path class and (3) maximum transfer rate. We averaged each macroscopic response separately in the three time periods identified in Fig. 6, yielding a total of (3 x 18 =) 54 macroscopic responses. We computed throughput per flow group separately for sources using TCP and for those using CTCP, yielding a total of (2 x 24 =) 48 flow-group throughput measurements. Thus the number of computed responses totaled (54 + 48 =) 102.

Table 2: Responses measured during the sensitivity analysis of MesoNet

Macroscopic Responses			Flow Groups for Throughput Averages			
Category	Identity	Definition	Number	File Size	Path Class	Max. Rate
Flow State	Y1	Average # sources connecting	1	Movie	VF	F
	Y2	Average # sources sending	2		VF	N
	Y3	% sending flows in initial slow start	3		F	F
	Y4	% sending flows in standard congestion avoidance	4		F	N
	Y5	% sending flows in alternate congestion avoidance	5		T	F
			6		T	N
Conges-tion	Y6	Retransmission rate	7	Software Service Pack	VF	F
	Y7	Average congestion window size (packets)	8		VF	N
	Y8	Aggregate # connection failures	9		F	F
			10		F	N
Delay	Y9	Average round-trip time (ms)	11		T	F
	Y10	Average queuing delay (ms)	12		T	N
			13	Docu-ment	VF	F
Work	Y11	Average # flows completed per second	14		VF	N
	Y12	Average # packets output per second	15		F	F
			16		F	N
Long-Lived Flows	Y13	Average throughput on long-lived flow #1	17		T	F
	Y14	Average throughput on long-lived flow #2	18		T	N
	Y15	Average throughput on long-lived flow #3	19	Web Object	VF	F
			20		VF	N
Flows by Path Class	Y16	Average throughput on flows transiting VF paths	21		F	F
	Y17	Average throughput on flows transiting F paths	22		F	N
	Y18	Average throughput on flows transiting T paths	23		T	F
			24		T	N

A few responses require brief explanation. Recall that sources cycle through three states: thinking, connecting and sending. We measured the average number of connecting (Y1) and sending (Y2) sources; other sources are thinking. Sending flows begin operating under initial slow start rules and may then move to congestion avoidance, where sources implementing CTCP may cycle between standard and alternate rules. We used responses Y3, Y4 and Y5 to measure the proportion of sending flows operating under each rule set. Since lost packets must be resent, we computed retransmission rate (Y6) as a ratio: file size to data packets sent on a flow before receiving the last acknowledgment. We measured the average work/second accomplished in flows (Y11) and packets (Y12) for each time period. With responses Y13, Y14 and Y15 we estimated instantaneous throughput in each time period for individual long-lived flows transiting specified paths in the network. Similarly, we used responses Y16, Y17 and Y18 to estimate instantaneous throughput on each path class in each time period regardless of differences in file size and maximum transfer rate. To estimate instantaneous throughput we divided the number of acknowledgments sent in a measurement interval by the interval size. For flow groups, we computed throughput measures by dividing file size (in packets) by the time interval between sending the first packet and receiving acknowledgment for the last packet.

6 SELECTED ANALYSIS TECHNIQUES IDENTIFY SIGNIFICANT MESONET PARAMETERS

We analyzed the main effects of each MesoNet parameter on all 102 measured responses. We also analyzed all two-parameter interactions for each response. We conveyed these analyses through custom plots. For example, Fig. 7 shows a main effects plot for response Y2 (average number of sending sources) in Time Period 2. The x axis identifies each of the 20 MesoNet parameters and the y axis gives the mean response. For each parameter the plot gives two means: (1) when the parameter is set to -1 value and (2) when set to the +1 value. Fig. 7 shows that the mean number of sending sources was just under 13 000 when network speed (X1) was low (-) and was about 8500 under high network speed (+). For each parameter, a line connects the two means to indicate direction and magnitude of the effect when changing the parameter from its -1 to +1 values. Two numbers are reported just above each parameter label. The top number gives the effect in raw terms (e.g., 4145 fewer sending flows under higher network speed) and the bottom number gives the percentage change (e.g., 39 % fewer sending flows under higher network speed), which is called the relative effect. We subjected each effect to a t-test for statistical significance, and inserted two asterisks (**) for effects, such as network speed, with statistical significance > 0.99. We insert one asterisk (*) for effects with significance > 0.95 and ≤ 0.99. Interpreting Fig. 7, we find that shorter think time (X7 -) coupled with more sources (X12 +) distributed in a peer-to-peer pattern (X13 +) induce the largest of the six statistically significant effects on the number of sending sources. The effect of these parameters is followed by larger file size (X5 +) and larger topology (X4 +). The indicated values for these five parameters increase demand on the network. The remaining significant increase in sending sources arises from lower network speed (X1 -). In short, greater demand offered to a slower network increases congestion, which causes longer file transfer times for sources, leading more sources to be in the sending state. Similar analyses are possible for the remaining 101 responses.

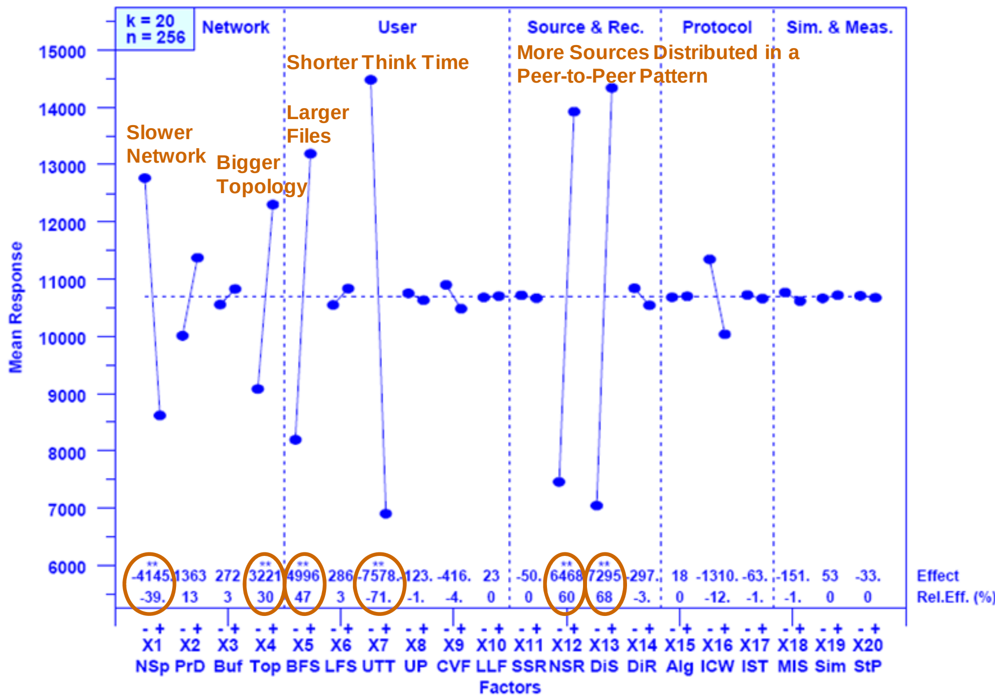


Figure 7: Main effects plot showing absolute and relative influence of each MesoNet parameter (x axis) on mean (y axis) number of sending flows (response Y2) during the 2nd time period, and identifying six statistically significant parameters: network speed (X1), topology size (X4), file size (X5), think time (X7) and number (X12) and distribution (X13) of sources

We created five (only one shown here) summary tables: three tables (one per time period) report statistically significant effects of parameters on the 18 macroscopic responses and two tables (one for TCP and one for CTCP) report statistically significant effects of parameters on throughput for each of the 24 flow groups. For example, Table 3 provides a summary for the 18 macroscopic responses in the 2nd time period – the row for Y2 was created from the main effects plot in Fig. 7. Cells in Table 3 highlighted in blue denote ** effects and those highlighted in tan denote * effects. Each highlighted cell also includes either a + or – to indicate which value for the corresponding parameter (column) led to a higher value in the response (row). For Y2 for example, we find slower (-) network speed (X1), larger (+) topology (X4), bigger (+) file sizes (X5) , shorter (-) think times (X7) and more (+) sources (x12) distributed in a peer-to-peer (+) pattern (X13) led to a larger number of active flows. This corresponds to the information given in Fig. 7. A quick scan of Table 3 shows that network speed had significant influence on all 18 responses during the 2nd time period. Other significant parameters can also be identified, as well as those that had little or no significant influence on the responses. Other patterns can also be discerned, such as the influence of particular sets of parameters on responses associated with network congestion.

We also used response data to explore the influence of two-parameter interactions on model behavior. This allowed us to establish that MesoNet is driven primarily by main effects rather than by interactions. We investigated two-parameter interactions using custom plots, such as the sample shown in Fig. 8, which reports interactions associated with response Y2 (number of sending sources) during the 2nd time period.

Table 3: Significance of influence of 20 MesoNet parameters (columns) on 18 macroscopic responses (rows) during 2nd time period: blue cells indicate significance > 0.99 and tan cells indicate significance > 0.95 and ≤ 0.99, where a – or + in highlighted cells indicates the parameter setting that causes an increase in the corresponding response

			Network				User Behavior						Source/Receiver				Protocol			Sim. Control & Meas.		
			X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	X16	X17	X18	X19	X20
Category	ID	Short Name	NSp	PrD	Buf	Top	FS	LFS	ThT	UP	CVF	LLF	SSR	NSR	DiS	DiR	CCA	ICW	IST	MIS	DUR	StP
Flow State	Y1	# Connecting	-**		+**	+*			-**					+**	+**			+**				
	Y2	# Active	-**			+**	+**		-**					+**	+**							
	Y3	% ISS	+**	+**	+**		-**		+**					-**	-**				+**			
	Y4	% NCA	-**	-**	-**		+**		-**					+**	+**		-*					
	Y5	% ACA	+**		+**				+**					-**	-**		+**	+*	-**			
Congestion	Y6	Retrans. Rate	-**	-**	-**		+**		-**					+**	+**			+**				
	Y7	*cwnd* Size	+*																			
	Y8	# conn. fails	-**	-**	-**		+**							+**	+**			+**				
Delay	Y9	SRTT	-**	+**	+**									+*	+*							
	Y10	Queue Delay	-**	+**	+**				-*					+**	+**							
Work	Y11	Flows/sec	+**	-*		+**	-**		-**					+**	+**							
	Y12	Packets/sec	+**		+**	+**	+**		-**		-**			+**								
Long-Lived Flows	Y13	LLF 1	+**		+*	+*					+**	-**										
	Y14	LLF 2	+*			+**					+**	-**										
	Y15	LLF 3	+**								+**	-*				-*						
Flows by Path Class	Y16	VF Paths	+**	-**		-**			+*		+**			-**		-*			+**	-*		
	Y17	F Paths	+**	-**		+*			+**		+**			-**	+**	-**		+**		-*		
	Y18	N Paths	+**	-**		-**			+**					-**	-**			+**				

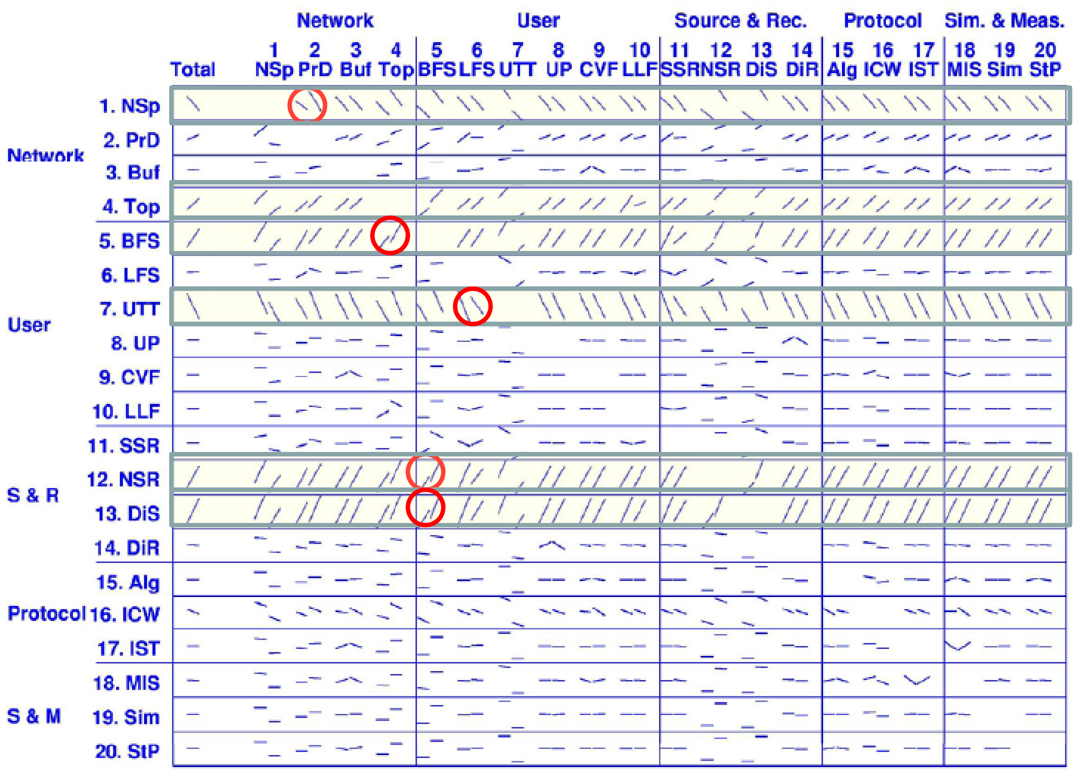


Figure 8: Two-parameter interaction effects plot for the average number of sending sources (Y2) during 2nd time period: highlighted rows indicate significant parameters from Fig. 7 and circled pairs of traces identify possible parameter interactions, which appear rather minor

Each row of Fig. 8 corresponds to a different parameter. The first column in each row provides a reference trace representing the main effects for the parameter. Each subsequent pair of traces in the same row shows the parameter effect when each other parameter takes on each of its two levels (first –1 and then +1). Traces with slopes differing from the reference identify the existence of interactions. For example, the first red circle in Fig. 8 suggests that when X2 = -1 (i.e., shorter propagation delay) then network speed (X1) has less influence on the response (number of sending sources). Similarly, the second red circle suggests that when X4 = -1 (smaller topology) then file size has less influence on the response. File size also interacts with number and distribution of sources. All these interactions appear rather minor. Overall, Fig. 8 instills confidence that the response is driven by single parameters rather than by two-parameter interactions. This was largely the case for all responses, thus we can summarize the sensitivity of MesoNet using only main effects analyses.

Table 4: Count (zeroes left blank) of macroscopic responses in each of 3 time periods and flow group throughputs for each of 2 congestion avoidance algorithms for which each MesoNet parameter had statistical significance > 0.99 and between 0.95+ and 0.99; counts are totaled for responses in each of the five sets and for all responses

Responses Covered	t-test Statistic	Network				User Behavior						Source/Receiver				Protocol			Sim. Control & Meas.		
		X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15	X16	X17	X18	X19	X20
		NSp	PrD	Buf	Top	FS	LFS	ThT	UP	CVF	LLF	SSR	NSR	DiS	DiR	CCA	ICW	IST	MIS	DUR	StP
Time Period #1	>0.99	17	9	10	8	8		11			3		12	11	2	1	7	2	1		
	>0.95≤0.99	1	1	3	2	2		2					3	2	1		2	1	1		
	Total	18	10	13	10	10		13			3		15	13	3	1	9	3	2		
Time Period #2	>0.99	16	9	9	6	7		10		6	2		13	11	1	1	5	3			
	>0.95≤0.99	2	1	1	2			2			1		1	1	2	1	1		2		
	Total	18	10	10	8	7		12		6	3		14	12	3	2	6	3	2		
Time Period #3	>0.99	17	9	11	6	9		12		4	3		12	11	2	1	5	3	1		
	>0.95≤0.99	1	2		3	1		1		1			3	2					2		
	Total	18	11	11	9	10		13		5	3		15	13	2	1	5	3	3		
TCP	>0.99	19	16	12	8	11		10		1			4	16			8	16	1		
	>0.95≤0.99		2	3	3	5		4					2								
	Total	19	18	15	11	16		14		1			6	16			8	16	1		
CTCP	>0.99	19	18	10	9	15		13		1			8	16			7	12			
	>0.95≤0.99			2	3	1		3					6		4		1				
	Total	19	18	12	12	16		16		1			14	16	4		8	12			
Total	>0.99	88	61	52	37	50		56		12	8		49	65	5	3	32	36	3		
	>0.95≤0.99	4	6	9	13	9		12		1	1		15	5	7	1	4	1	5		
	Total	92	67	61	50	59		68		13	9		64	70	12	4	36	37	8		

Table 4 condenses our five tables summarizing main effects for macroscopic responses (in each of three time periods) and for throughput per flow group (under each of two congestion avoidance algorithms). Each column in Table 4 denotes one parameter. The table shows three rows for each of the five response sets. The final row of the table reports the total number of the 102 responses for which each parameter showed a statistically significant effect > 0.95. Network speed influenced 90 % of responses, while the number and distribution of sources, along with think time and file size influenced between 58 % and 69 % of responses. Propagation delay influenced 66 % of responses and buffer sizing influenced 60 %. Eight parameters influenced a relatively low proportion of responses, and five influenced no responses. The table also reveals that initial slow start threshold (X17) had significant influence on flow group throughput, especially under TCP.

We find that MesoNet is driven mainly by network capacity, demand (i.e., number, distribution and activity of sources), propagation delay and buffer sizing. This finding suggests that MesoNet can create significant variation in behavior under experiments that use only 7 of its 20 parameters. Further, since per-flow throughput can be influenced significantly by initial slow start threshold, throughput experiments should take that parameter into account. These findings suggest that MesoNet can be used to evaluate proposed congestion control algorithms with 2^7 parameter combinations, which can be reduced to only 32 combinations using a 2^{7-2} Resolution IV OFF design.

7 CONCLUSIONS

We showed how 2-level-per-factor designs can be used to construct experiments for simulation models. We discussed the application of orthogonal fractional factorial (OFF) experiment design theory to reduce the number of parameter combinations simulated. We exploited this reduction to define an efficient sensitivity analysis method for simulation models, which we applied to MesoNet, where we simulated only 2^8 of 2^{20} possible parameter combinations. We introduced effective techniques to concisely analyze the influence of parameters on main effects and to investigate two-parameter interactions. We showed that

MesoNet is driven primarily by 7 of its 20 parameters, which influence main effects rather than interactions. Based on these findings we identified that a 2^{7-2} Resolution IV OFF design could be constructed to generate significant variations in MesoNet using only 32 parameter combinations. Elsewhere (Mills et al. 2010) we use these insights to compare seven proposed Internet congestion control algorithms.

REFERENCES

Appenzeller G., I. Keslassy and N. McKeown. 2004. Sizing Router Buffers. In *Proceedings of ACM SIGCOMM*, 34:4, 281-292.

Box, G., J Hunter,. and W. Hunter, 2005. *Statistics for Experimenters*. 2nd ed. Hoboken, New Jersey: Wiley.

Bush R. and D. Meyer. 2003. Some Internet Architectural Guidelines and Philosophy. *RFC 3439*.

Fall, K. and K. Varadhan, eds. 2009. The *ns* Manual. Available via <http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf> [accessed December 2, 2009].

Henriksen, J.O. 2000. SLX: the X is for extensibility. In *Proceedings of the 32nd Winter Simulation Conference*,183-190.

Kratz, M., M., Ackerman, T. Hanss and S. Corbato. 2001. NGI and Internet2: Accelerating the Creation of Tomorrow's Internet. in *MEDINFO 2001*, IOS Press, Vol. 84, 28-32.

Mills, K., E. Schwartz and J. Yuan. 2010. How to model a TCP/IP network using only 20 parameters. Submitted to *the 41st Winter Simulation Conference*.

Mills, K., J. Filliben, D. Cho, E. Schwartz and D. Genin. 2010. *Study of Proposed Internet Congestion-Control Mechanisms*. NIST Special Publication 500-282.

Paxson, V. and S. Floyd. 1997. Why we don't know how to simulate the Internet. In *Proceedings of the 1997 Winter Simulation Conference*, ed. S. Andradottir, K. J. Healy, D. H. Withers, and B. L. Nelson, 1037-1044.

Riley, G., M. Ammar, F. Fujimoto, A. Park, K. Perumalla and D. Xu. 2004. A Federated Approach to Distributed Network Simulation. In *ACM Transactions on Modeling and Computer Simulation*, 14:2, 116-148.

SSFNet. 2009. How to use SSFNet. Available via <http://www.ssfnet.org/internetPage.html> [accessed December 2, 2009]

Tan, K., J. Song, Q. Zhang and M. Sridharan. 2006. A Compound TCP Approach for High-Speed and Long Distance Networks. In *Proceedings of the 25th Annual Joint Conference of the IEEE Computer and Communications Societies* (INFOCOM 2006), pp. 1-12

Tyan, H-Y., A. Sobeih and J. Hou. 2009. Design, Realization and Evaluation of a Component-based, Compositional Network Simulation Environment. In *Simulation*, 85:3, 159-181.

Yaun, G., D. Bauer, H. Bhutada, C. Carothers, M. Yukel and S. Kalyanaraman. 2003. Large-Scale Network Simulation Techniques: Examples of TCP and OSFP Models. In *SIGCOM Computer Communications Review*, 33:3, 27-41.

Zeng. X., R. Bagrodia and M. Gerla. 1998. GloMoSim: a Library for Parallel Simulation of Large-scale Wireless Networks. In *Proceedings of the 12th Workshop on Parallel and Distributed Simulations*, 154-161.

AUTHOR BIOGRAPHIES

KEVIN L. MILLS is a Senior Research Scientist at the U.S. National Institute of Standards and Technology (NIST). He received his Ph.D. in information technology from George Mason University (GMU). From 1996 to 2006 he served on the adjunct faculty of the department of computer science at GMU. From 1996 to 1999 he was a program manager at the Defense Advanced Research Projects Agency (DARPA). His research interests include complex distributed systems. His email is <kmills@nist.gov>.

JAMES J. FILLIBEN is a Mathematical Statistician at NIST. He received his Ph.D. in statistics from Princeton University. He has collaborated with dozens of NIST scientists in a wide range of disciplines, for which and his achievements have been recognized with Gold, Silver and Bronze medals awarded by the U.S. Department of Commerce. His research interests include statistical experiment design and exploratory analysis of multidimensional data. His email is <james.filliben@nist.gov>.