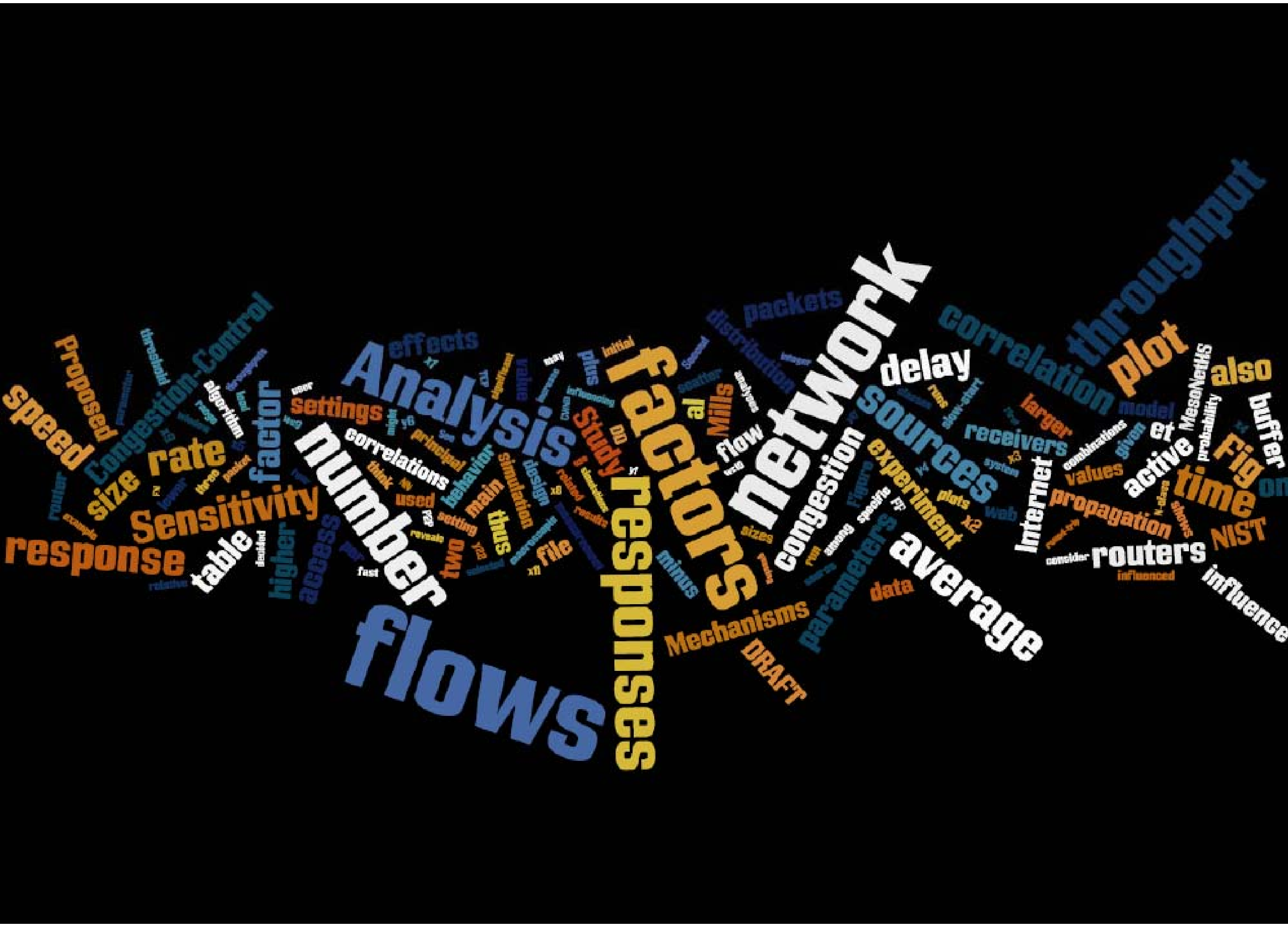
Chapter 4 – Sensitivity Analysis of MesoNet



4 Sensitivity Analysis of MesoNet

This section discusses a sensitivity analysis of MesoNet, along with related correlation and principal components analyses. Sensitivity analysis [94] varies settings of a model's input parameters and assesses resulting changes in model outputs. Correlation analysis [90] examines the way in which two outputs vary with relation to each other when exposed to the same conditions. Principal components analysis generates orthogonal linear combinations of weighted measures that account for variance in model outputs. Here, we conduct a sensitivity analysis to understand the behavior of MesoNet and to discover the most significant model inputs that influence model response. To assess relationships between model inputs and outputs we use a 10-step graphical analysis technique developed at NIST. In addition to allowing analysis of input-output relationships, exercising a model over a wide range of its parameter space helps to reveal software implementation errors, which can be corrected prior to applying the model in particular studies. We use correlation and principal components analyses to identify significant aspects of MesoNet behavior. In general, application of correlation and principal components analyses can help to reduce the number of responses that must be used in subsequent statistical analyses. The results of our analyses serve to validate that MesoNet reasonably represents the macroscopic behavior of a network of TCP flows. The results of our analyses also help to answer some questions raised in the literature regarding the applicability of particular findings from small-scale simulations to a larger network.

The current practice of network modeling omits the use of sensitivity analyses, despite the fact that network researchers understand the benefits of such analyses [70, 72]. Why is this so? Most network simulators [76-83] are quite detailed, involving hundreds of parameters with potential settings that can range over many values. Running such simulations with large topologies and billions of TCP flows can be a daunting task, requiring substantial computational resources. In addition, configuring the parameter settings in such simulations can be time-consuming and tedious. Sensitivity analysis requires running a simulation through many combinations of settings. Thus, configuring and computing the required combinations for a detailed model with a large parameter space is infeasible.

Recently, two groups of researchers developed hybrid models [71, 73] that aim to reduce the computational requirements and range of parameters necessary to simulate TCP flows in reasonably large topologies. MesoNet was motivated by the same aims: establishing a new class of network models that can simulate many flows operating over a large network topology, while maintaining reasonable configurability and computational requirements. For example, MesoNet has on the order of 20 fundamental parameters and, depending on specific parameter settings, can simulate tens of billions of flows in days or weeks of processing time on commercial servers using x86-compatiable chips with cycle speeds of 2.6 GHz to 3.66 GHz. Thus, it becomes possible to contemplate conducting sensitivity analyses for this new class of network models.

Still, 20 parameters, each with a large possible range (*n*) of values, can suggest a large space of (n^{20}) combinations to consider. To circumvent such a problem, experiment designs used in many scientific disciplines have long adopted an approach where the range of values for system parameters is limited to a small number, typically two or three, referred to by experiment designers as levels. As explained previously in Chapter 2, this approach, when applied to MesoNet, could limit the number of combinations to 2^{20} (just over a million). Still, running a million experiments would prove challenging when each experiment requires several days of processing time. Of course, individual combinations could be spread across independent processors to reduce the latency before all combinations are completed. For example, if 2^5 (32) processors were available, then 2^{20} simulations could be reduced to only 2^{15} serial executions. However, if each simulation required two days, then these simulations would still take 2^{16} days to complete. No one is willing to wait 180 years to collect data for a sensitivity analysis. Adding 32 additional processors to conduct the simulations would reduce the latency to around 90 years. Each additional 32 processors would cut the time further. Perhaps one day soon computation servers will offer 2^{16} processors in an affordable package. Such a computation engine would allow us to complete 2^{20} MesoNet simulations in about one month. In the meantime, we must adopt another approach to solve the problem.

Many scientific disciplines face situations where the number of desired experiments (even when considering only two levels per parameter) is unaffordable due to issues of cost or time. The best available practice in such situations is to use orthogonal fractional factorial (OFF) experiment designs [89] tailored to provide the maximum possible information from an affordable number of experiments. For example, if we could afford to run only 2^8 (256) MesoNet simulations, then we would use a 2^{20-12} OFF experiment design. Such a design would select 256 combinations of parameter settings that allow us to probe the parameter space in a balanced and orthogonal form. A balanced experiment design means that all combinations of parameter settings will be given an equal number of observations. An orthogonal experiment design means that observations will be spread equally throughout the space of possible parameter combinations. The properties of balance and orthogonality yield significant benefits when conducting statistical and graphical analyses of experiment data. In addition, properly selected combinations of parameters will limit the amount of confounding that arises when analyzing experiment data. When confounding arises a particular observed effect cannot be clearly attributed to a single factor or interaction of factors. Sometimes, domain knowledge can be used to resolve the uncertainty from confounding; however, one should strive to create an experiment design that eliminates confounding among at least the main effects[1] and as many two-parameter and three-parameter interactions[2] as possible. OFF experiment designs can be combined to good effect with a 10-step graphical analysis technique used in many scientific studies conducted at NIST. This 10-step technique is explained in Appendix D using detailed examples drawn from this sensitivity analysis.

The remainder of this chapter is organized into eight sections. Sec. 4.1 outlines our method for experiment design and analysis. Sec. 4.2 describes the specific experiment design used for the sensitivity analysis of MesoNet. Sec. 4.3 discusses the execution of the simulations and the data collection techniques. Sec. 4.4 presents a correlation analysis of 22 responses collected from each of the experiment executions.

[1] Main effects are changes in model response that can be attributed to changes in individual model parameters (or input factors).

[2] Interactions occur when changes in model responses can be attributed to simultaneous changes in multiple (e.g., two or three) model parameters.

Sec. 4.5 uses principal components analysis (PCA) as an alternate means to investigate relationships among the responses. Section 4.6 details the sensitivity analysis of MesoNet. In Sec. 4.7 we consider the effects of buffer sizing on network behavior. We conclude in Sec. 4.8.

4.1 Method

We use a method that involves five main elements. First, we use 2-level orthogonal fractional factorial experiment design (Sec. 4.1.1) to yield maximum information using the available computing resources. Second, we select candidate responses (Sec. 4.1.2) to analyze. Third, we employ correlation analysis and clustering (Sec. 4.1.3) along with principal components analysis (Sec. 4.1.4) to identify significant behaviors represented within the candidate responses. Third, we apply a 10-step graphical analysis (Sec. 4.1.5 and Appendix D) to provide insight into the main input parameters (or factors) driving the behavior of the simulation model. In addition, we augment our analyses with various exploratory plots (e.g., Sec. 4.1.6) designed to shed light on specific questions of interest. Below, we elaborate on these elements.

4.1.1 Experiment Design

We consider our model in the form of following equation: $[y_1, y_2, \dots y_M] = f(x_1, x_2, \dots x_N)$, which represents the model as a function transforming its N inputs (factors) into M outputs (responses). Designing an experiment consists of four main steps. First, we identify the N factors (model parameters) whose influence on system behavior we would like to investigate. Second, we select the number (L) of levels and then the settings ($s_1, s_2, \dots s_L$) for each level of each factor ($x_1s_1, x_1s_2, \dots x_Ns_L$). Third, we specify the combinations of factor settings that we intend to simulate. Fourth, we identify the M responses we are interested in investigating. We discuss each of these steps in turn.

4.1.1.1 Identify Factors. At a maximum, the factors include all parameters associated with a model of interest. Of course, this can be quite a large number, so one may wish to limit the specific parameters to investigate. Some parameters might specify control details, such as the number or granularity of measurement intervals and the seeds of random number generators. Typically, these may be fixed to specific values during a sensitivity analysis. Fixed parameters are not factors to be investigated in a set of experiments.

If the number of factors is still too large, other reduction steps may be adopted. For example, one may fix various factors and conduct sensitivity analyses with a limited number of runs. Repeating this process with various groupings of factors may identify some parameters as having limited influence on system behavior, at least for the range of settings envisioned for a particular experiment. Parameters that appear to have limited influence can be fixed during a sensitivity analysis that investigates more significant parameters. Domain expertise may also be applied to select various parameters to fix. One should exercise care in fixing particular parameters because some important elements of system behavior could be missed. Once parameters have been classified as fixed or variable for a given set of experiments, the variable parameters become the N factors for the experiment.

4.1.1.2 Select Number of Levels and Level Settings for Factors. Selecting the number of levels for an experiment determines the maximum number of combinations (L^N) that will be investigated. The most common practice in engineering experiments is to specify 2-level (L = 2) designs, which yield 2^N as the maximum number of combinations. 2^N designs result in nice properties of balance and orthogonality when OFF designs are used to reduce the number of combinations in a particular experiment. For this reason, we adopt L = 2 in our sensitivity analysis.

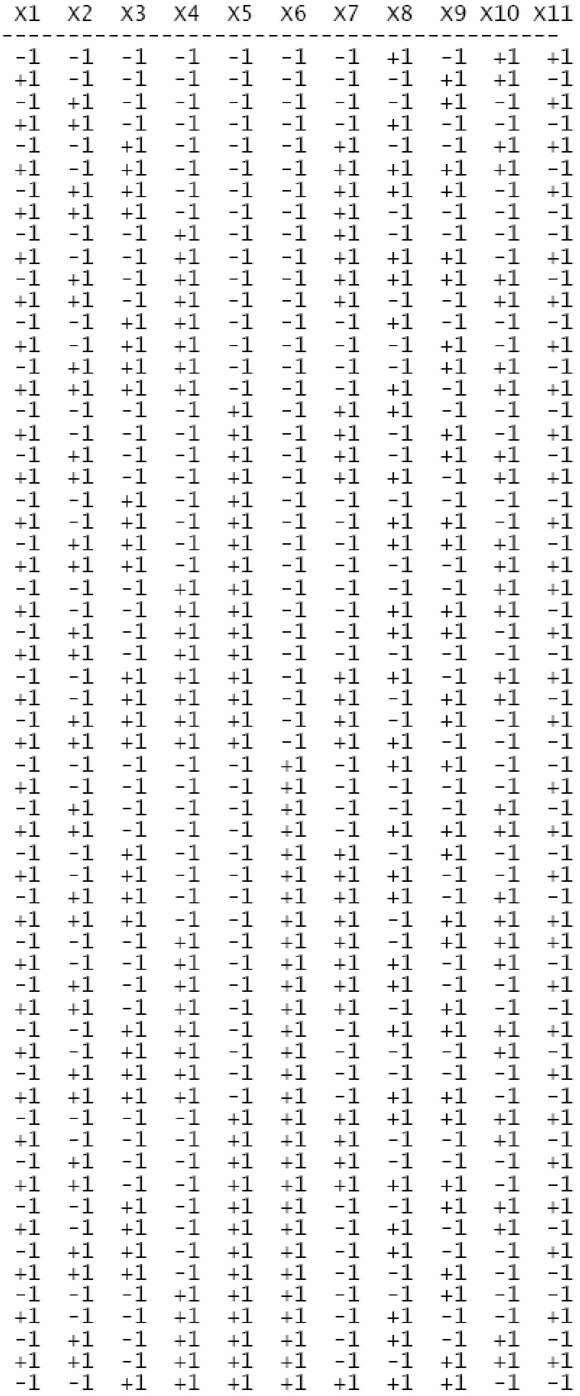
Given L = 2, one needs to select values for each of the N factors at each of two levels. This mapping of levels to factors yields specific parameter values to be used in a set of experiment executions. The two levels are typically encoded as a plus (+1) level and a minus (-1) level. This form of encoding simplifies many mathematical transformations that are applied during experiment design and data analysis. By convention, the larger value of a setting is assigned to the +1 level and the smaller value is assigned to the -1 level.[3]

Selecting the specific settings for the +1 and -1 levels of each factor is a key step that relies on domain expertise of an experimenter. Little general guidance exists because specific domains of investigation vary widely. In general, settings should be selected so as to be both realistic in the domain and also to stimulate the system sufficiently to reveal differences in response. When experiments are done using computers, preliminary simulations can be used to probe for the effects of varying specific parameters. No matter what settings are chosen, the analysis method relies upon an assumption that responses vary monotonically over the range of settings investigated. In cases where behaviors are non-monotonic, analysis of experiment data could completely miss significant and important behaviors. Further, the conclusions from data analysis are limited to (i.e., robust over) the range of settings investigated. For this reason, it is often prudent to run a second sensitivity analysis using different level settings to confirm conclusions from an initial sensitivity analysis. As discussed in Appendix C, we adopt this measure of prudence in our sensitivity analysis. Later, we plan to conduct a more complete sensitivity analysis with N = 20, covering the entire MesoNet parameter space.

4.1.1.3 Select Specific Combinations to Simulate. Ideally, one would run a full factorial experiment that simulates all 2^N possible combinations of level settings and factors. Often, though, executing 2^N runs would be unaffordable. For example, we selected (N =) 11 factors for our sensitivity analysis. A full factorial experiment would require 2^{11} (2048) runs. We could spread those runs over 16 processors, but each run requires between four and 10 hours of processing. We estimated that running a full-factorial experiment would require about 32 days of computing time plus overhead associated with managing the process. Such overhead includes configuring and monitoring simulations, collecting and summarizing data and recovering from various hardware and software

[3] Note that due to an encoding error in the design of our sensitivity analysis we inadvertently encoded higher network speed (our factor x2) as the -1 level and slower network speed as the +1 level. Unfortunately, this can lead to confusion when viewing some of the related plots. Despite this potential for confusion, the encoding approach works fine even when larger values are assigned to the -1 setting and smaller values to the +1 setting. Correcting this in our situation would require rerunning the related experiments, which would prove too costly.

failures that might arise. This illustrates that there may be other justified reasons for reducing the number of experiment runs even when one can afford the processing cost.



```
 X1  X2  X3  X4  X5  X6  X7  X8  X9 X10 X11
-----------------------------------------
 -1  -1  -1  -1  -1  -1  -1  +1  -1  +1  +1
 +1  -1  -1  -1  -1  -1  -1  -1  +1  +1  -1
 -1  +1  -1  -1  -1  -1  -1  -1  +1  -1  +1
 +1  +1  -1  -1  -1  -1  -1  +1  -1  -1  -1
 -1  -1  +1  -1  -1  -1  +1  -1  -1  +1  +1
 +1  -1  +1  -1  -1  -1  +1  +1  +1  +1  -1
 -1  +1  +1  -1  -1  -1  +1  +1  +1  -1  +1
 +1  +1  +1  -1  -1  -1  +1  -1  -1  -1  -1
 -1  -1  -1  +1  -1  -1  +1  -1  -1  -1  -1
 +1  -1  -1  +1  -1  -1  +1  +1  +1  -1  +1
 -1  +1  -1  +1  -1  -1  +1  +1  +1  +1  -1
 +1  +1  -1  +1  -1  -1  +1  -1  -1  +1  +1
 -1  -1  +1  +1  -1  -1  -1  +1  -1  -1  -1
 +1  -1  +1  +1  -1  -1  -1  -1  +1  -1  +1
 -1  +1  +1  +1  -1  -1  -1  -1  +1  +1  -1
 +1  +1  +1  +1  -1  -1  -1  +1  -1  +1  +1
 -1  -1  -1  -1  +1  -1  +1  +1  -1  -1  -1
 +1  -1  -1  -1  +1  -1  +1  -1  +1  -1  +1
 -1  +1  -1  -1  +1  -1  +1  -1  +1  +1  -1
 +1  +1  -1  -1  +1  -1  +1  +1  -1  +1  +1
 -1  -1  +1  -1  +1  -1  -1  -1  -1  -1  -1
 +1  -1  +1  -1  +1  -1  -1  +1  +1  -1  +1
 -1  +1  +1  -1  +1  -1  -1  +1  +1  +1  -1
 +1  +1  +1  -1  +1  -1  -1  -1  -1  +1  +1
 -1  -1  -1  +1  +1  -1  -1  -1  -1  +1  +1
 +1  -1  -1  +1  +1  -1  -1  +1  +1  +1  -1
 -1  +1  -1  +1  +1  -1  -1  +1  +1  -1  +1
 +1  +1  -1  +1  +1  -1  -1  -1  -1  -1  -1
 -1  -1  +1  +1  +1  -1  +1  +1  -1  +1  +1
 +1  -1  +1  +1  +1  -1  +1  -1  +1  +1  -1
 -1  +1  +1  +1  +1  -1  +1  -1  +1  -1  +1
 +1  +1  +1  +1  +1  -1  +1  +1  -1  -1  -1
 -1  -1  -1  -1  -1  +1  -1  +1  +1  -1  -1
 +1  -1  -1  -1  -1  +1  -1  -1  -1  -1  +1
 -1  +1  -1  -1  -1  +1  -1  -1  -1  +1  -1
 +1  +1  -1  -1  -1  +1  -1  +1  +1  +1  +1
 -1  -1  +1  -1  -1  +1  +1  -1  +1  -1  -1
 +1  -1  +1  -1  -1  +1  +1  +1  -1  -1  +1
 -1  +1  +1  -1  -1  +1  +1  +1  -1  +1  -1
 +1  +1  +1  -1  -1  +1  +1  -1  +1  +1  +1
 -1  -1  -1  +1  -1  +1  +1  -1  +1  +1  +1
 +1  -1  -1  +1  -1  +1  +1  +1  -1  +1  -1
 -1  +1  -1  +1  -1  +1  +1  +1  -1  -1  +1
 +1  +1  -1  +1  -1  +1  +1  -1  +1  -1  -1
 -1  -1  +1  +1  -1  +1  -1  +1  +1  +1  +1
 +1  -1  +1  +1  -1  +1  -1  -1  -1  +1  -1
 -1  +1  +1  +1  -1  +1  -1  -1  -1  -1  +1
 +1  +1  +1  +1  -1  +1  -1  +1  +1  -1  -1
 -1  -1  -1  -1  +1  +1  +1  +1  +1  +1  +1
 +1  -1  -1  -1  +1  +1  +1  -1  -1  +1  -1
 -1  +1  -1  -1  +1  +1  +1  -1  -1  -1  +1
 +1  +1  -1  -1  +1  +1  +1  +1  +1  -1  -1
 -1  -1  +1  -1  +1  +1  -1  -1  +1  +1  +1
 +1  -1  +1  -1  +1  +1  -1  +1  -1  +1  -1
 -1  +1  +1  -1  +1  +1  -1  +1  -1  -1  +1
 +1  +1  +1  -1  +1  +1  -1  -1  +1  -1  -1
 -1  -1  -1  +1  +1  +1  -1  -1  +1  -1  -1
 +1  -1  -1  +1  +1  +1  -1  +1  -1  -1  +1
 -1  +1  -1  +1  +1  +1  -1  +1  -1  +1  -1
 +1  +1  -1  +1  +1  +1  -1  -1  +1  +1  +1
 -1  -1  +1  +1  +1  +1  +1  +1  +1  -1  -1
```

Figure 4-1. Encoding Template for a 2^{11-5} Orthogonal Fractional Factorial Experiment Design

After estimating the time requirements for conducting a full-factorial experiment, an investigator must decide on the number of affordable runs. In making this decision, one should also consider the confounding effects that would arise for a particular choice.

For example, suppose we decided to limit the number of affordable runs to 64 instead of 2048. We would then need to select a subset of (2^6) combinations from among the complete set of 2^{11}. Experiment design theory [89] labels such a design as a 2^{11-5} (= 2^6) design. Experiment design theory also specifies which 64 combinations to select and reveals the resulting confounding structure for the experiment.

Fig. 4-1, taken from Dataplot [92], a software package available from NIST, shows the +1/-1 encoding of a 2^{11-5} OFF design as a matrix. Each row in the matrix represents a specific experiment run. Each cell in a given row specifies the level setting to be used for a designated factor (x1 through x11). Thus, having previously assigned +1/-1 level settings for each factor, an experimenter need only map level settings according to this table to create the specific combination of experiment parameters for each run.

Experiment design theory also specifies the precise confounding structure associated with this experiment design. A 2^{11-5} OFF design has no confounding of main effects with two-factor interactions, which is a desired property of an experiment design. Some main effects are confounded with some three-factor and higher interactions, but most systems are not driven by such higher interactions. From this we conclude that a 2^{11-5} OFF design would yield significant information for our sensitivity analysis.

A reasoning process such as outlined above should be used when selecting specific combinations to simulate. Of course, the reasoning process must be tailored to the specific number of factors and affordable runs. Experiment design theory provides appropriate algorithms to generate designs and determine associated confounding structures for any bounds of interest. The NIST Dataplot software [92] also provides encoded experiment designs and confounding structures for a range of typical OFF designs, as documented by Hunter and Box [89].

4.1.1.4 Select Responses to Examine. Often simulation models can measure system response through tens to hundreds of outputs, which might represent aspects of fewer significant underlying model behaviors. Usually, experimenters select a subset of model outputs to analyze because considering all available responses proves too time consuming, too costly or computationally infeasible. MesoNet, for example, can monitor the time-varying average aggregate behavior of the network for about 20 responses, can report about 6 time-varying properties for every router in a topology and can measure average throughputs experienced by users in six topologically determined flow classes. Summarizing and analyzing all of this data would prove time-consuming.

When choosing a subset of simulation outputs, experimenters may select outputs in a fashion that overemphasizes particular behaviors. These mistakes become particularly salient during careful exploration of a model's parameter space, where experimenters seek to understand the response of a model to changes in input parameters. Overweighting significant model behaviors can yield misleading conclusions, thus some method is required to identify precisely the model outputs that correspond to each significant behavior. Fodor [93] describes this mathematically as a dimension reduction problem: "given the *p*-dimensional random variable x = $(x_1,\ldots, x_p)^T$, find a lower dimensional representation, s = $(s_1,\ldots, s_k)^T$ with $k \leq p$, that captures the content in the original data, according to some criterion." Fodor goes on to survey numerous linear and non-linear techniques that may be applied to reduce the dimension of high-dimensional data sets. Adopting any of these techniques would provide a principled approach that experimenters could use to identify significant model behaviors from a large collection of model output data. Of course, one wonders whether some techniques are superior to others. Fodor identifies principal components analysis (PCA) as the best (in terms of mean-square error) linear dimension reduction technique. In our analysis we use two techniques to reduce dimension in MesoNet response data. We use PCA and we also combine correlation analysis and clustering (CAC). Applying both techniques allows us to compare and contrast their findings, which provides additional information about MesoNet behavior.

4.1.2 Candidate Responses

We chose to examine a set of 22 MesoNet responses to which we applied PCA and CAC to identify lower dimensional response spaces representing the most significant model behaviors. This information could help us validate our model and could also help us to reduce the number of responses to analyze in subsequent experiments. We selected our 22 candidate responses from among the measurements (see Sec. 3.3) provided by MesoNet. We selected responses in two classes: (a) responses that depict macroscopic behavior of the network and (b) responses that indicate user experience for various flow classes. We discuss these response classes in turn.

4.1.2.1 Responses Characterizing Macroscopic Network Behavior. We chose 12 fundamental responses to characterize macroscopic network behavior and we augmented those with four derived responses in order to investigate how well the fundamental responses represented the intended information. Table 4-1 lists the responses we used to characterize macroscopic behavior. MesoNet records each response as a value associated with each measurement interval, providing a time series for each response. To compute the fundamental responses that we analyzed, we discarded data from the first 3000 of 6000 measurement intervals recorded. We then averaged the remaining data (from the second 3000 intervals) to obtain a mean value for each response. To compute a derived response, we mathematically manipulated some combination of fundamental responses, sometimes including a factor setting. The details are given as appropriate in Table 4-1.

We tracked the number of active flows (y1) over time and used that number to indicate the general amount of user activity in the network. Because more potential sources might lead to more active flows, we chose also to consider (y2) what proportions of potential flows were represented by the active flows. In this way, we could investigate whether the number of possible flows was a key determinant in the number of active flows, or whether the number of active flows was driven primarily by other factors. We measured separately the number of data packets entering (y3) and leaving (y4) the network because we wanted to understand what relationship, if any, exists between the rate at which packets are injected into the network and the number of active flows. Given the rate of packets entering and leaving the network, we could also measure the loss rate (y5), which should give us some rough indication of the amount of network congestion.

While the rate of data packets leaving the network gives us some idea of aggregate throughput, we were also interested in investigating the ability of the network to complete flows (y6), which could be combined with the number of active flows to yield a flow-completion rate (y7). Since we implemented TCP connection establishment (explained in Chapter 5) procedures, congestion could lead connection establishment to fail. We measured the number of connection failures (y8) and also related the failures to the number of active flows (y1) to create a connection-failure rate (y9).

Table 4-1. Responses Characterizing Macroscopic Network Behavior

Response	Definition
y1	Active Flows – flows attempting to transfer data
y2	Proportion of potential flows that were active: Active Flows/All Sources
y3	Data packets entering the network per measurement interval
y4	Data packets leaving the network per measurement interval
y5	Loss Rate: y4/(y3+y4)
y6	Flows Completed per measurement interval
y7	Flow-Completion Rate: y6/(y6+y1)
y8	Connection Failures per measurement interval
y9	Connection-Failure Rate: y8/(y8+y1)
y10	Retransmission Rate
y11	Congestion Window per Flow
y12	Window Increases per Flow per measurement interval
y13	Negative Acknowledgments per Flow per measurement interval
y14	Timeouts per Flow per measurement interval
y15	Smoothed Round-Trip Time
y16	Relative queuing delay: y15/(x1x41)

For active flows, we were interested in understanding the average level of congestion. We suspected that several measures of flow congestion should be correlated. We measured the average retransmission rate (y10) for flows, which we postulated should be about twice the loss rate. We also measured the average congestion window per flow (y11) – larger congestion windows indicate that flows should be receiving better throughputs. In addition to the congestion window size, we chose to measure the average number of window increases (y12) received per flow during each measurement interval. To determine to what extent retransmissions arose from indicated losses vs. timeouts, we measured the number of negative acknowledgments (y13) and number of timeouts (y14) per flow.

Finally, we were interested in monitoring smoothed round-trip time (y15), which might provide some indication of congestion. We also wanted to see how changing buffer sizes influenced round-trip time. We computed a relative queuing delay (y16) by factoring out propagation delay from the smoothed round-trip time. We computed y16 because we wished to discover if there would be any differences in the pattern between smoothed round-trip time and queuing delay.

4.1.2.2 Responses Characterizing User Experience. Aside from aggregate network behavior, we were interested in exploring the throughputs received for the six possible flow classes allowed by MesoNet. This required monitoring six additional responses, as shown in Table 4-2. Here, the measure gives average instantaneous throughput for a flow in each class, so the metric captures the throughput for active flows rather than flows that have finished. As with the aggregate measures, we computed the average value for each flow class over the final 3000 measurement intervals of each simulation run.

Table 4-2. Responses Characterizing Instantaneous Throughput for Active Flows by Flow Class

Response	Definition
y17	Average Throughput for Active DD Flows
y18	Average Throughput for Active DF Flows
y19	Average Throughput for Active DN Flows
y20	Average Throughput for Active FF Flows
y21	Average Throughput for Active FN Flows
y22	Average Throughput for Active NN Flows

We chose to examine the throughput of the various flow classes in order to determine whether or not different factors affect the throughput of flows transiting different types of access routers. We collected separate throughput data for flows that completed and for flows that completed in each flow class. For purposes of our sensitivity analysis, we decided not to analyze the throughput data for completed flows.

4.1.3 Correlation Analysis and Clustering

As part of our analysis we wish to investigate the sensitivity of various model responses to model inputs. Of course, we are also interested in learning relationships among the responses. A reasonable hypothesis might be that correlated responses are influenced by the same model inputs. Further, clustering correlated responses into significant model behaviors might allow us to reduce the number of responses analyzed in future experiments. To determine relationships among responses we conduct a correlation analysis using the techniques described in this section. First, we generate scatter plots among all response pairs. Second, we compute correlations among each pair of responses. Third, we combine the selected scatter plots and correlation values into a single visualization. The combined visual can be ordered using several techniques to reveal correlation groupings. Finally, we select a correlation threshold above which we wish to consider correlations, and then generate an ordered index-index plot to highlight correlation groups and to help select specific responses for further study. We explain these four steps below. To aid our explanation, we use designators for various responses. The designators are y*N*, where y denotes a response and *N* denotes the number of the response. Here, *N* may range from 1 to 22 to correspond with the 22 candidate responses described in Sec. 4.1.2.

4.1.3.1 Y-Y Scatter Plots. Scatter plots of each pair of responses can visually reveal linear correlations and can also suggest structure beyond correlation. Fig. 4-2 shows a sample scatter plot between two responses from our sensitivity analysis. The abscissa gives values for response y22 (average instantaneous throughput among typical flows) and the ordinate gives values for response y7 (flow-completion rate). Perhaps unsurprisingly, the scatter plot reveals a positive linear correlation among the two responses. Higher throughput for typical flows, which are most numerous, leads to higher flow-completion rate. Perhaps surprisingly, the scatter plot also reveals a bifurcation in correlation structure. Attributable to the properties of our OFF experiment design, the scatter plot can be augmented to reveal the cause underlying this bifurcation. We discuss the use of other exploratory plots and analyses below in Sec. 4.1.6.

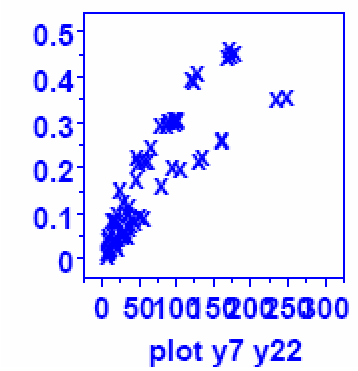
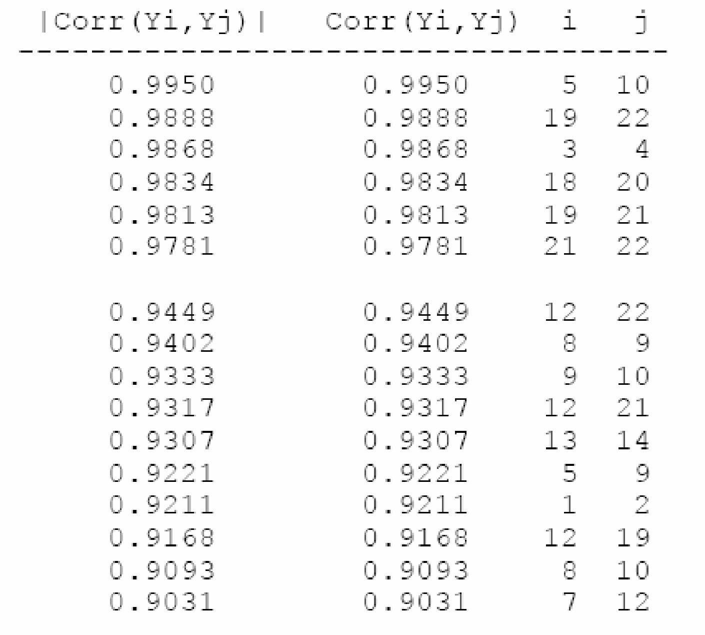


Figure 4-2. Enlargement of Sample Scatter Plot of Response y7 vs. y2 – y axis gives the flow completion rate (y7) as a proportion (ranging from 0 to 0.5, where each tick mark represents 0.05) and x axis gives average goodput of NN flows (y22) in packets per second (ranging from 0 to 300, where each tick mark represents 25 packets/second)

4.1.3.2 Correlation Computations. We also compute correlations among all pairs of responses generated from our sensitivity analysis. We compute the signed values, which separate positive and negative correlations, and the absolute values, which allow us to order correlations by magnitude. Fig. 4-3 provides a sample table of correlations, ordered by magnitude, where magnitude > 0.9. The table consists of four columns: (a) absolute value of the correlation between a pair of responses (Y*i* and Y*j*), (b) the signed value of the correlation, (c) the identifier (*i*) of the first response in the pair and (d) the identifier (*j*) of the second response in the pair. Here, two subgroups are shown: (1) correlations > 0.95 and (2) correlations > 0.9 and < 0.95. In this particular sample, all correlations are positive.



```
 |Corr(Yi,Yj)|   Corr(Yi,Yj)  i   j
-----------------------------------
      0.9950         0.9950    5  10
      0.9888         0.9888   19  22
      0.9868         0.9868    3   4
      0.9834         0.9834   18  20
      0.9813         0.9813   19  21
      0.9781         0.9781   21  22

      0.9449         0.9449   12  22
      0.9402         0.9402    8   9
      0.9333         0.9333    9  10
      0.9317         0.9317   12  21
      0.9307         0.9307   13  14
      0.9221         0.9221    5   9
      0.9211         0.9211    1   2
      0.9168         0.9168   12  19
      0.9093         0.9093    8  10
      0.9031         0.9031    7  12
```

Figure 4-3. Sample (and Partial) Table of Correlations among Response Pairs

We also plot a histogram (see Figure 4-4) of the absolute values of all pairs of correlations. This gives a concise view of the distribution of correlations. The histogram can help us select a threshold above which to consider the correlations. Fig. 4-4, for example, suggests that correlations greater than about 0.65 should be considered because there is a notable change above that value, appearing as a separate sub-distribution centered on a different mode. This sub-distribution includes around 40 of the 231 correlation pairs computed.

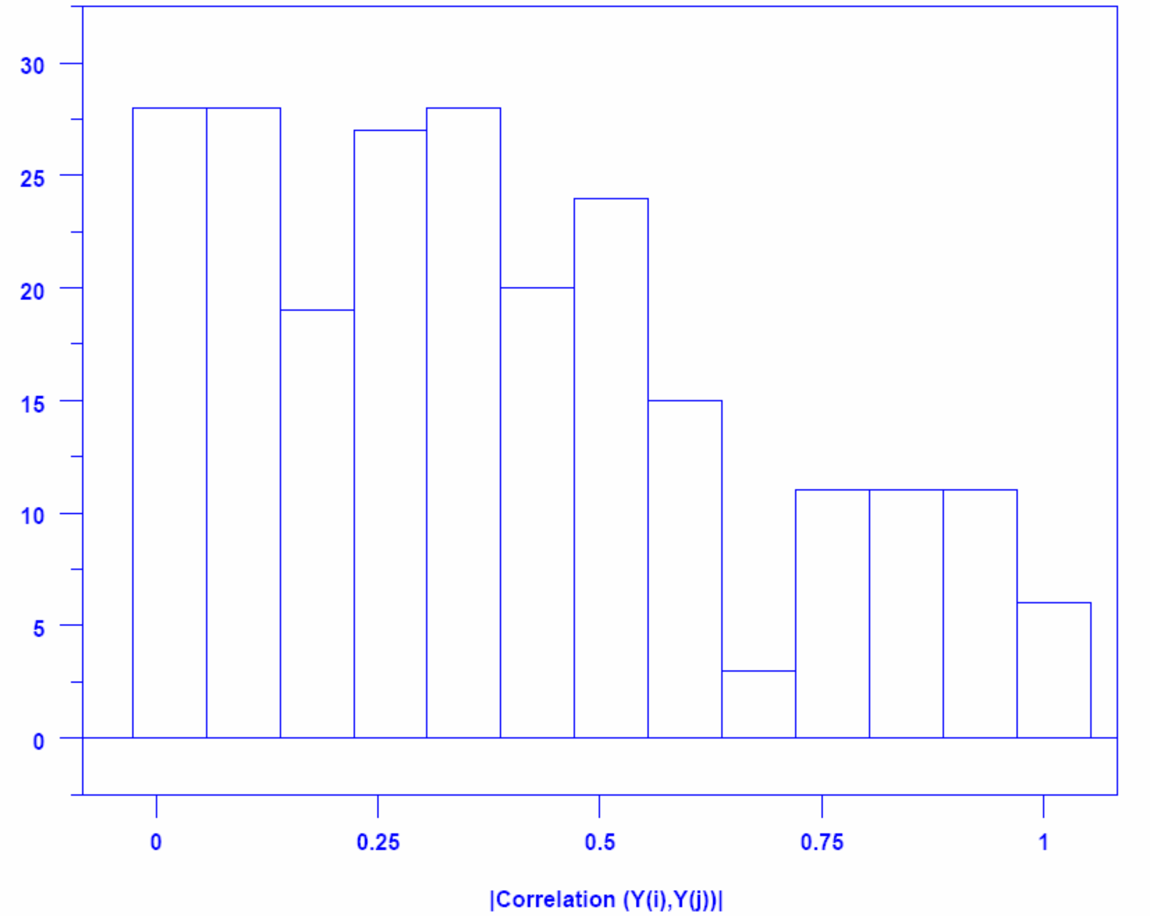
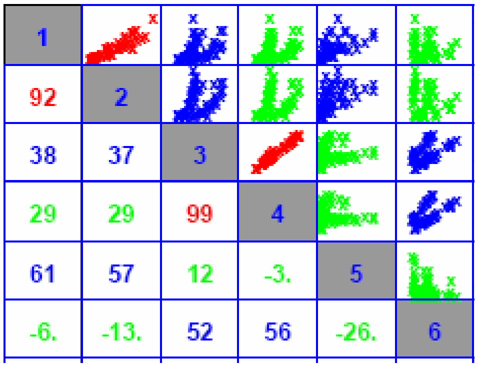


Figure 4-4. Sample Histogram of Correlation Magnitudes among Response Pairs (x axis depicts correlation strength divided into 13 bins, where each bin spans a range of size ~0.075 and y axis gives the count, or frequency, of correlation pairs appearing in each bin)

4.1.3.3 Combined Matrix Visualization. We can combine the response scatter plots and computed correlation values into a matrix visualization providing a concise view of all relevant information. Further, we can use color to highlight various correlation groupings. Fig. 4-5 gives a 6-x-6 subset taken from our complete matrix for all 22 responses.

The diagonal of the matrix identifies a particular response associated with each column and row. The scatter plots are displayed to the right and above the diagonal and the associated correlation values (multiplied by 100, rounded and truncated) are displayed to the left and below. For example, consider the response y3 (data packets input per time unit), which is third on the diagonal in Fig. 4-5. The scatter plot in the cell directly to the right of y3 and above y4 (data packets output per time unit) depicts the linear correlation between y3 and y4. The cell directly below y3 and to the left of y4 reads 99, which is the associated correlation value. Not surprisingly, the correlation is positive and quite high. Similarly, the scatter plot related to y2 (proportion of flows that are active) vs. y3 is shown in the cell directly above y3 and to the right of y2. The related correlation value (37) is given in the cell immediately below y2 and to the left of y3. Perhaps the weakness of this correlation is surprising. Other scatter plots and correlation values may be located similarly. For example, the scatter plot in the cell in the upper right-hand corner depicts y1 (number of active flows) vs. y6 (flows completed per time unit) and the related correlation value (-6) appears in the cell in the lower left-hand corner of the matrix. While the negative direction of the y1-y6 correlation is not surprising, the lack of correlation might be unexpected.



1					
92	2				
38	37	3			
29	29	99	4		
61	57	12	-3.	5	
-6.	-13.	52	56	-26.	6

Figure 4-5. Sample 6-x-6 Subset from a Combined Matrix of Scatter Plots and Correlation Values

Thresholds may be selected for coloring the scatter plots and correlation values in the combined matrix visualization. In Fig. 4-5 we chose three colors, based on the magnitude of the correlation values. For correlation magnitudes 80 and above, we colored the related cells red. We colored cells blue for correlation magnitudes below 80 and greater than or equal to 30. The green cells represent correlation magnitudes below 30. After coloring, one can scan the matrix to visually group correlations by their strengths. The diagonal of the colored matrix may also be reordered, along with the related scatter plots and correlation values. Such reordering may readily indentify correlation groupings. For example, Fig. 4-5 could be reordered by descending mean, median or maximum correlation of each given response with all other responses. Later, in Sec. 4.3, we order our matrix by descending mean correlation, which nicely groups correlations among response pairs.

4.1.3.4 Index-Index Plot. Fig. 4-6 shows an index-index plot involving all 22 responses from our sensitivity analysis. Guided by Fig. 4-4, we display only correlations with magnitudes above 0.65. The x and y axes in Fig. 4-6 both list all 22 responses in order of numerically increasing designator (N = 1 to 22). Then a grid is formed. A point is placed at each grid intersection when the magnitude of the correlation between the related pair of responses exceeds 0.65. In Sec. 4.3, we use this index-index plot but we reorder the axes in a different form. The resulting correlation groups, not obvious in Fig. 4-6, become quite apparent after the axes are reordered (for example, see Fig. 4-22).

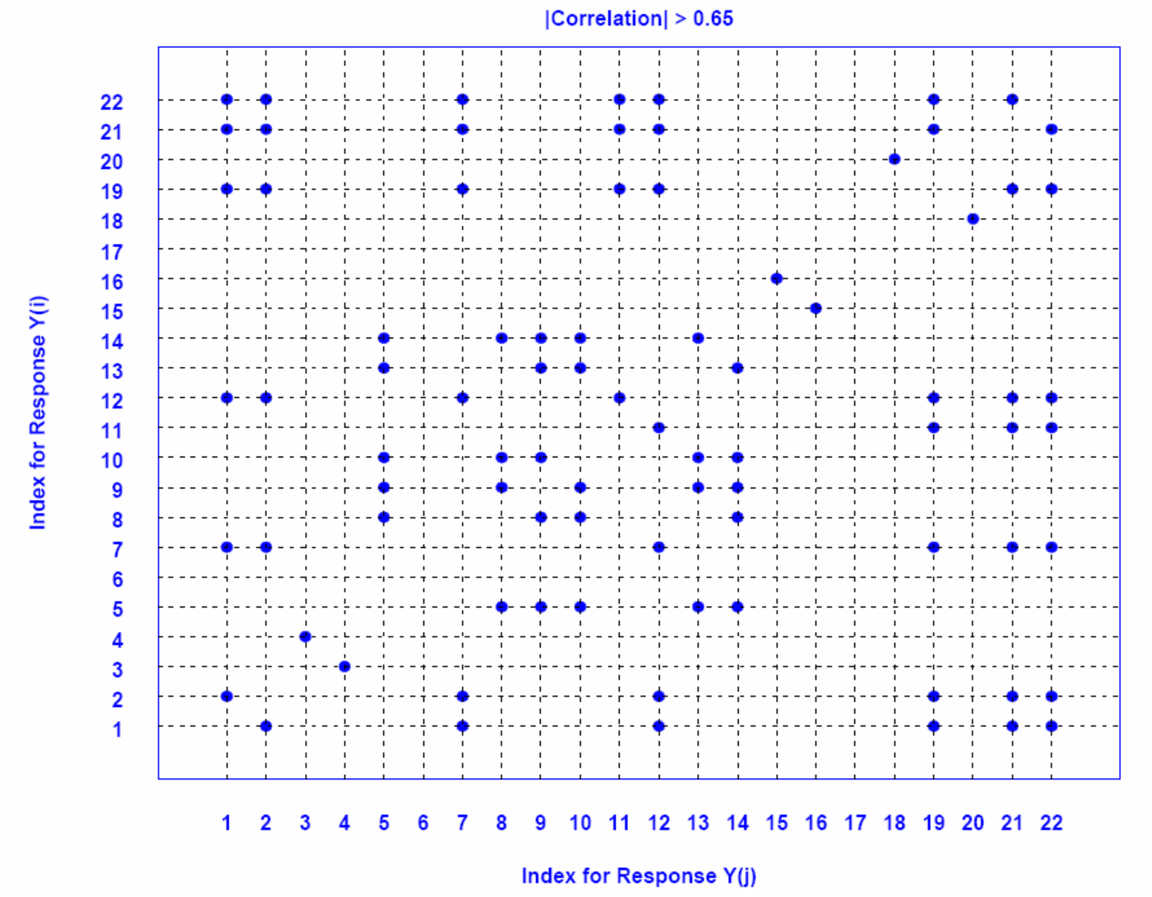


Figure 4-6. Index-Index Plot Identifying Response Pairs with Correlation Magnitude above 0.65

4.1.4 Principal Components Analysis

Principal Components Analysis (PCA) provides another approach to identify significant behaviors in model response data. PCA aims to reduce the dimensionality of model responses by finding orthogonal linear combinations (i.e., principal components, or PCs) of the responses that account for the largest variance. In essence, PCA identifies as many PCs as there are responses, with each PC being orthogonal to the others and with PC1 accounting for the largest variance in the data and PC2 second largest and so on to PC*n*, where *n* is the total number of responses. For many sets of responses, the first several PCs account for most of the variance in the data, and thus those PCs represent the most significant model behaviors.

In our case we have 64 samples (recall Fig. 4-1) for each of 22 response variables (recall Tables 4-1 and 4-2). Since variance depends upon the scale of each response, we must first normalize each response to have a mean of zero and a standard deviation of one. This can be done for a given response by subtracting the mean of the 64 samples from the response and then dividing by the standard deviation of the 64 samples. Such normalization will place all responses into comparable units.

In our application, each PC consists of a 22 dimensional weight vector representing the linear weighted combination of response variables necessary to generate the PC. For example, Fig. 4-7 depicts a graphical representation of the weight vector for the first PC (PC1) from one of our PCAs. The figure depicts the 22 response variables on the x axis, while the y axis gives the (positive or negative) weighting. A horizontal line denotes zero weight. Given a weight vector for a PC, it is often customary in heuristic interpretation to suppress consideration of the low-weighted variables. In Fig. 4-7 for example, we might choose to suppress the following variables: y3, y4, y6, y15, y16, y17, y18 and y20. It is also customary to differentiate between "average" weights and "contrasting" weights, though such differentiation is not warranted in Fig. 4-7.

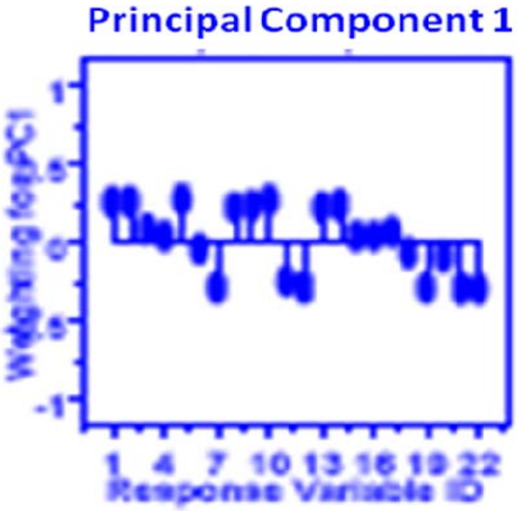


Figure 4-7. Enlargement of a Sample Weight Vector for First Principal Component (x axis identifies the response variable, ranging from 1 to 22 and y axis identifies weight, ranging from -1 to +1)

For each PC we can plot a histogram of the 64 values using appropriate components to represent the variance accounted for by the PC. For example, Fig. 4-8 gives the histogram corresponding to PC1. The x axis divides the standard deviation over the 64 values into appropriately sized bins and the y axis gives the count of values that fall into each bin. Above the plot we give the standard deviation accounted for by the PC. Given an entire set of such histograms, we can determine the relative variance accounted for by each PC by summing the standard deviations and then dividing each by that sum. For example, Fig. 4-23 recounts the 22 histograms representing each PC in one of our PCAs. In that case, the first four PCs account for about 86 % of the variance in the data.

4.1.5 10-Step Graphical Analysis of Selected Responses

Once we select the specific responses to examine, we can subject them to a 10-step graphical analysis regime developed at NIST. Each analysis step produces a different type of plot intended to reveal information about model responses. In this section we simply introduce the intent of each plot type, as shown in Table 4-3, which lists each of the ten plots and provides a summary of the purpose of each plot. In Appendix D we give detailed examples and explanation of each plot type. Here we introduce in detail only the main effects plot, which proved most insightful for purposes of our sensitivity analysis.

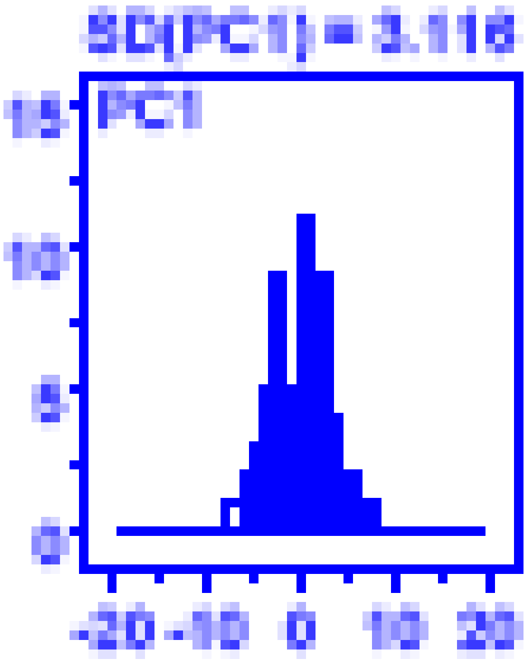


Figure 4-8. Enlargement of a Sample Histogram for First Principal Component (x axis identifies bins of normalized component values ranging from -20 to +20 and y axis the count of values within each bin). Above the plot is the standard deviation in the data accounted for by the Principal Component.

We illustrate the main effects plot using response variable y11, average congestion window (CWND) size. The transmission control protocol (TCP) manages a congestion window variable that represents the number of packets that can be sent prior to receipt of an acknowledgment. The larger the congestion window, the more packets that can be sent per unit time and thus the greater will be the transmission rate. For that reason, a network with a high average congestion window (y11) will be able to transmit more packets than a network with a lower average congestion window. In general, a congestion window is reduced when packets are lost, usually due to congestion. Lowering the congestion window slows the rate of packet transmissions in the network and thus should reduce congestion. Subsequent to a reduction, TCP allows the congestion window to increase linearly and so the rate of packet transmissions in the network should also increase. Once the transmission rate becomes too high, packets are lost and congestion windows are reduced and the rate of transmission slows and so on. Thus the average congestion window size might be used to represent the level of congestion in a network.

Fig. 4-9 gives a sample main effects plot, which is the most essential plot to identify the factors and settings driving a system's response. The x axis identifies each of 11 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. 4-9 shows that the mean CWND size was about under 8.5 packets when network speed (X1) was high (-) and was about 4 packets under low 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 value. Two numbers are reported just above each parameter label. The top number gives the effect in raw terms (e.g., CWND size of 4.33 fewer packets under lower network speed) and the bottom number gives the change relative to (i.e., as a % of) the mean response, which is about 6.2 packets in Fig. 4-9 (i.e., the 4.33 packet change in CWND size is 70 % of the mean, which is called the relative effect). The plot also gives the number of parameters (k = 11) and observations (n = 64).

Table 4-3. Identity and Purpose of 10 Plots in the 10-Step Graphical Analysis (For sample and explanation of each plot see Appendix D)

Plot	Purpose
Ordered Data Plot	Reveal how combinations of parameter settings influence response
Multi-factor Scatter Plot	Reveal influence of individual parameter levels on response distribution
Main Effects Plot (see Fig. 4-8)	Reveal individual parameters having greatest influence on response
Interaction Effects Matrix	Reveal degree of influence of parameter pairs on response
Block Plot	Test robustness of statistically significant parameters in light of secondary or nuisance factors
Youden Plot	Reveal parameters and parameter pairs with greatest influence on response
|Effects| Plot	Reveal magnitude of a change in response due to specific parameters and parameter interactions
Half-Normal Probability Plot of |Effects|	Separate influential parameters and parameter interactions from those that are not influential
Cumulative Residual SD Plot	Provide information sufficient to construct a linear model to represent response data
Contour Plot	Suggest how alterations in parameter settings could influence system response in predictable directions.

Fig. 4-9 reveals that the most influential factor in determining CWND is network speed (70 % of mean) followed by three closely grouped factors: buffer-sizing algorithm (54 %), initial slow-start threshold and think time (53 % each). The distribution of sources also has a significant (50 %) influence. Notice that the plot reveals a smaller number of sources and receivers (x8 = -) leads to a (1.7 packet) larger average CWND than a larger number. A domain expert will understand that fewer sources sharing the same network mean that each source may transmit faster, which is reflected in a larger CWND. Thus, the main effects plot clearly reveals the nature of the influence of the factors and settings on the response.

In thinking about the main effects, an experimenter with domain knowledge might be quite pleased with the meaning of these results regarding the validity of the model. Fewer, simultaneously active, flows (x5 = +, x8 = - and x9 = -), higher network speeds (x2 = -) together with more buffers (x3 = +) should permit higher CWND. Under these circumstances, the ability to increase the CWND to a higher threshold via initial slow-start (x11 = +) should also lead to higher CWND, because CWND increases faster during initial slow start.

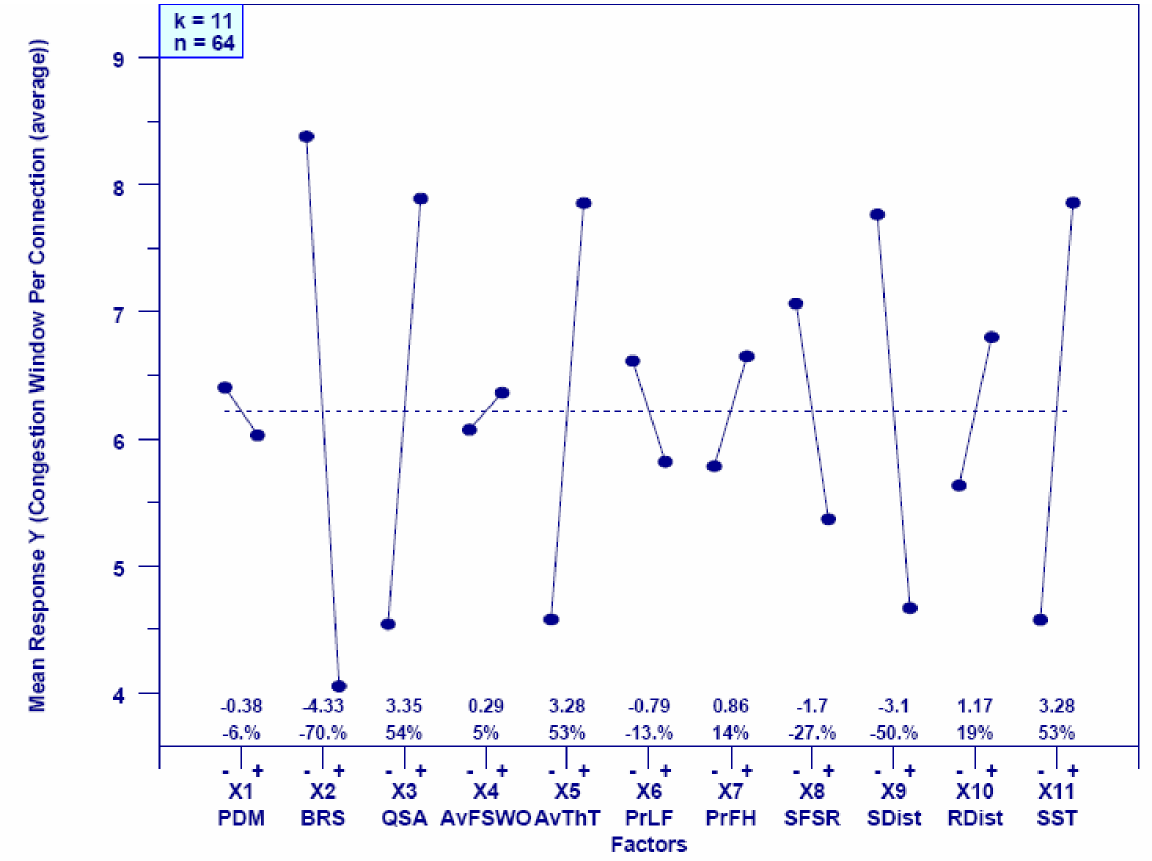


Figure 4-9. Sample Main Effects Plot for Response y11, average congestion window size, x axis lists 11 model parameters with a – and + value for each parameter, y axis gives average congestion window (CWND) size in packets, two average CWND sizes are given for each parameter, one size when the parameter is set to its – level and one size when the parameter is set to its + level, and a line connects the pair of average sizes for each parameter. Dashed line is the overall average CWND size (about 6.2 packets)

In Appendix D we illustrate the application of the entire 10-step graphical analysis technique to analyze model parameters influencing CWND size. An experimenter might also apply the 10-step graphical analysis technique to examine influences on principal components. We give an example of this technique in Sec. 4.6.2.

4.1.6 Other Exploratory Plots and Analyses

Using an orthogonal fractional factorial (OFF) design opens the possibility for a range of exploratory plots and analyses to supplement the correlation and clustering analysis, the principal components analysis and the 10-step graphical analysis presented so far. For example, bifurcations in response-response scatter plots can be explored by altering the scatter plot symbols to reflect factor settings. As a sample, recall Fig. 4-2, a scatter plot of y7 vs. y22, which revealed a bifurcation. One means to explore the underlying reason for the bifurcation is to plot points using symbols, e.g., - when associated with minus settings for each factor, and as +, when associated with plus settings. Fig. 4-10 illustrates twelve scatter-plots for y7 vs. y22. The first plot, upper left-hand corner, repeats the scatter plot from Fig. 4-2. The remaining plots encode the plus (in blue) and minus (in red) settings responsible for the responses given each of the 11 factors (x1 through x11).

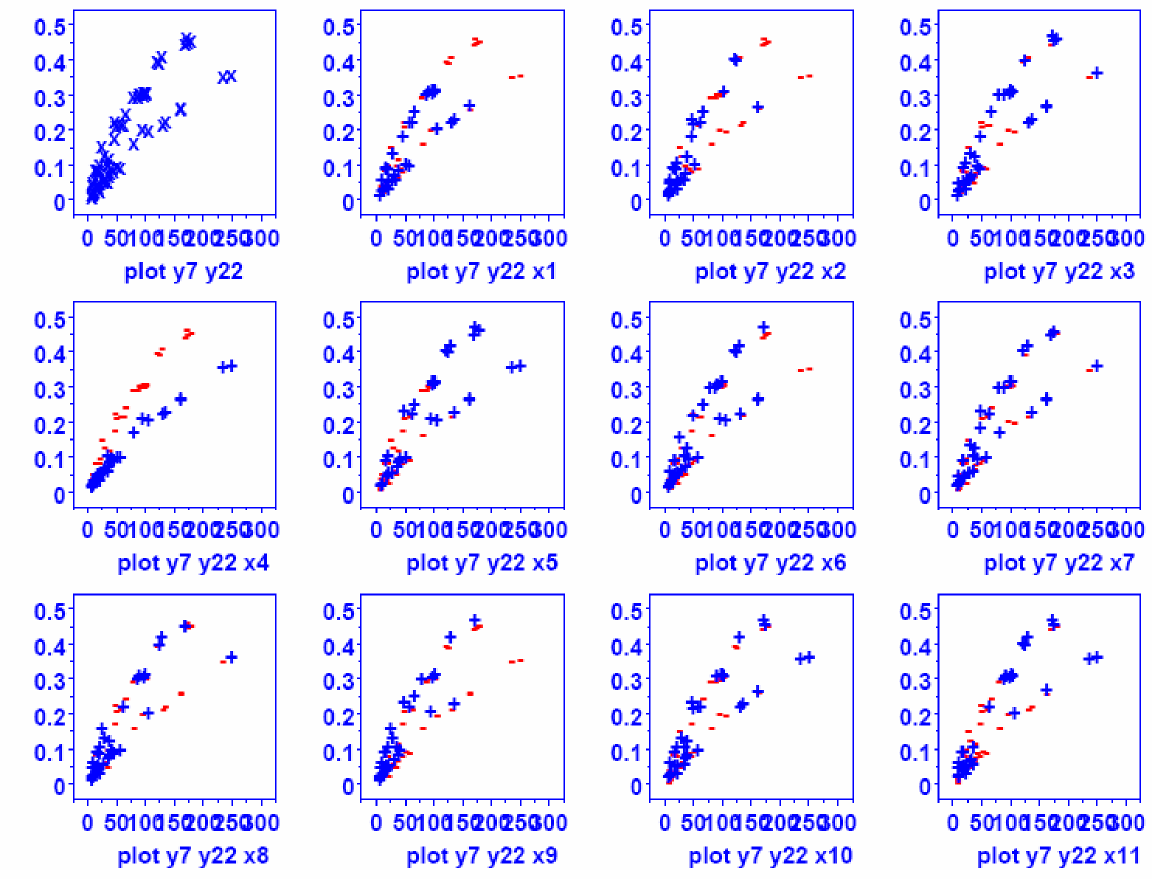


Figure 4-10. Sample Y-Y-X plot for Responses y7 and y22 – configuration of each plot is as explained for Fig. 4-2 – x axis is goodput in packets/second on NN flows ranging from 0 to 300 and y axis is flow completion rate ranging from 0 to 0.5

Examining Fig. 4-10, one can appreciate that factor x4 (average file size) is responsible for the bifurcation. Shorter file sizes result in higher completion rates (y7) and yet lead to lower average throughputs for typical flows (y22). Thinking this through reveals a sensible explanation. Shorter files spend a higher percentage of their transfer in TCP slow start, during which throughputs are lower. On the other hand, shorter files are generally transferred more quickly because they involve fewer packets. Since shorter files are transferred more quickly, more flows are completed per unit of time, so the flow completion rate is higher. Longer files spend a higher percentage of their transfer beyond TCP slow start, during which throughputs are higher. On the other hand, longer files require transferring more packets. Since it takes longer to transfer more packets, fewer flows are completed per unit time. Thus, the explanation for the bifurcation, as revealed in the y7-y22-x4 plot in Fig. 4-10, matches an explanation that appears reasonable to a domain expert. Such analysis and reasoning can help to verify a model's correctness, or to reveal flaws.

Additional exploratory plots and analyses are also possible. For example, one can combine factor settings to create additional conditions and then compare the relative effects of varying each of the combined factor settings on the ordering of selected responses. More specifically, one could combine the 2-level settings for factors x1 through x3 (propagation delay, network speed and buffer size) to create $2^3 = 8$ conditions, and then examine the relative influence of varying each of the factors on selected responses. We use such an approach in Sec. 4.7 to explore the relative effects on system response due to changing network speed, propagation delay and buffer size. We defer a more detailed explanation of the technique to Sec. 4.7.

4.2 Experiment Design for MesoNet Sensitivity Analysis

This section outlines the experiment design used for the MesoNet sensitivity analysis, and explains the rationale underlying the design. The design consists of a 2^{11-5} orthogonal fractional factorial (OFF) design, which requires 64 simulation runs. We compared the results of the simulation runs across 22 responses (described in Sec. 4.1.2). Below, we summarize MesoNet parameters and we identify the 11 parameters chosen as factors in our OFF design. We then explain the levels, and related settings, chosen for each factor. We summarize the 64 specific combinations simulated.

4.2.1 MesoNet Factors

For this sensitivity analysis, MesoNet parameters may be divided into six general categories: (1) simulation-control parameters, (2) parameters controlling user behavior, (3) parameters adapting the characteristics of the network, (4) parameters altering the properties of sources and receivers, (5) parameters controlling the startup pattern of sources and (6) parameters related to TCP operation. We describe the specific parameters in each category. Parameter descriptions may identify a parameter's type as an integer or a float. In such cases, one should assume that an integer may take on values between -2^{31} and $+2^{31}$. A float may take on values in the range of 1.797^{-308} to 1.797^{+308}. The range of values for any other parameter types will be given explicitly. For more detail on these parameters see Sec. 3.2.

As we discuss the MesoNet parameters, we identify (highlighted in blue bold) which were chosen as factors for our sensitivity analysis and we give our reasoning. In each table, we also give (highlighted in red) the fixed values assigned to the excluded parameters. At the end of the section, we recap the parameters included as factors in the sensitivity analysis.

4.2.1.1 Simulation Control Parameters. Simulation control is affected by six parameters, as defined in Table 4-4. The sensitivity analysis will not consider the response of MesoNet to variations in simulation-control parameters. Thus, five of these parameters will simply be fixed (to the values shown in Table 4-4) across all experiment runs. The maximum propagation delay in our experiments will be around 200 time steps, which we select for our basic measurement interval duration. We set our fundamental time-step duration to 1 millisecond, so each measurement interval captures about 200 milliseconds (i.e., five measurement intervals cover one second). We run each simulation for 6000 measurement intervals, which is (6000/5 =) 1200 seconds (20 minutes). We set our random number seed to a fixed value for all simulations because we are interested in capturing changes due to parameter variations, and not variations due to randomness. We vary the run number from 1 to 64 to identify the particular configuration of factors used in specific experiments.

Table 4-4. Simulation-Control Parameters

Parameter	Definition	Type
P1	Number of time steps in a measurement interval (**200**)	Integer
P2	Number of measurement intervals (**6000**)	Integer
P3	Number of measurement intervals in a measurement buffer (**6000**)	Integer
P4	Run number (**1 to 64**, signifying a combination of factors)	Integer
P5	Random number seed (**200000**)	Integer
P6	Duration of each time step (**0.001 s**)	Float

4.2.1.2 Parameters Controlling User Behavior. Eight parameters, shown in Table 4-5, determine how individual users (sources) behave over the course of a simulation. The MesoNet sensitivity analysis considers only typical Web traffic, so parameters (P12-P14) dealing with jumbo file transfers will be assigned fixed values (see Table 4-5) that cause them to be deactivated. The five remaining parameters (P7-P11) are candidates to include as factors in the experiments.

Table 4-5. Parameters Controlling User Behavior

Parameter	Definition	Type
P7	Shape parameter for the distribution of web-object sizes (**1.5**)	Float
P8	**Average size (in packets) of web objects**	**Integer**
P9	**Average think time (in time steps) between web clicks**	**Integer**
P10	**Probability a user decides to download a larger document**	**Float**
P11	Factor by which web-object size is multiplied if it is a larger document (**10**)	Integer
P12	Proportion of simulation time that elapses before jumbo file transfers begin (**1.0**)	Float
P13	Proportion of simulation time that elapses before jumbo file transfers end (**1.0**)	Float
P14	Factor by which web-object size is multiplied if it is a jumbo file (**100**)	Integer

We decided to fix the value of parameters P7 and P11, so we selected only three parameters to control user behavior during the sensitivity analysis. We chose to fix P7 (= 1.5) because experiments with P7 set to 1.2 and 1.5 revealed little difference in results. We chose to fix P11 (= 10) because varying the size of Web objects (P8) will also implicitly vary the size of larger documents. Further, preliminary sensitivity analyses with P11 set to either 5 or 10 showed little influence on the results.

4.2.1.3 Parameters Adapting Network Characteristics. Nine parameters, shown in Table 4-6, may be varied to adapt characteristics of a network topology defined for a MesoNet simulation. We decided to vary only three of these parameters during the sensitivity analysis. We wanted to be able to vary propagation delay, network speed and buffer sizes. Parameter P15 can alter the base propagation delays defined in a network topology. Network speed can be influenced by six parameters (P16-P21). We chose only to vary the backbone speed (P17), which has the effect of varying the speeds of the other routers because their speeds are expressed in terms of backbone-router speed. Thus, even though we fix parameters P18-P21 to the values given in Table 4-6, the speeds of the associated routers vary as we vary P17. We chose not to vary the speedup of backbone routers (P16 is fixed to 1) because our anticipated scenario of simulated Web traffic was unlikely to overwhelm the backbone routers with traffic.

Table 4-6. Parameters Adapting Network Characteristics

Parameter	Definition	Type
P15	**Factor by which to multiply basic propagation delays defined within a simulated topology**	**Float**
P16	Multiplier used to speed up backbone routers (**1**)	Integer
P17	**Backbone router speed (in packets per time step)**	**Integer**
P18	Divisor used to reduce the speed of POP routers relative to backbone routers (**4**)	Integer
P19	Divisor used to reduce the speed of access routers relative to POP routers (**10**)	Integer
P20	Multiplier used to increase the speed of directly connected access routers over typical access routers (**10**)	Integer
P21	Multiplier used to increase the speed of fast access routers over typical access routers (**2**)	Integer
P22	**Identification of a specific buffer-sizing algorithm to adopt**	**Integer (1 to 3)**
P23	Multiplier used to increase or reduce buffer sizes as computed by the algorithm selected by parameter P22 (**1.0**)	Float

Buffer sizes can be varied by choosing among several algorithms to calculate buffers in each router. This choice is controlled by P22, which we varied for our sensitivity analysis. Another parameter, P23, may be used to refine buffer sizing, either increasing or decreasing the basic buffer sizes computed by a chosen algorithm. For our sensitivity analysis, we decided to stick with the choice among alternate algorithms, so we fixed the value of P23 to 1.0.

4.2.1.4 Parameters Altering Properties of Sources and Receivers. Nine parameters, shown in Table 4-7, control the properties of sources and receivers within the model. Controllable properties include: the network interface speeds of hosts on which sources and receivers operate, the relative number of sources and receivers and the distribution of sources and receivers within the network topology.

In our sensitivity analysis, we are interested in examining the effects on responses as the number and speeds of sources and receivers is changed and as the distribution of sources and receivers is altered in the topology. This requires varying the six parameters (P26, P28-P32) highlighted in Table 4-7. We decided there is no need to vary the speeds of either basic or fast hosts, so we simply fix the speed of each (P24 = 1 and P25 = 8). Varying the probability a host is fast and the number of sources and receivers in the network should provide sufficient variation in the number of fast and slow hosts in the network. Similarly, we decided not to vary the base number of sources (P27 = 100) under an access router because the number of sources and receivers are determined by multiplying the base number by a scaling factor (P28). Thus, varying P28 achieves sufficient variability among the number of sources and receivers in a topology.

Table 4-7. Parameters Altering Properties of Sources and Receivers

Parameter	Definition	Type
P24	Speed (packets per time step) of basic host (**1**)	Integer
P25	Speed (packets per time step) of fast host (**8**)	Integer
P26	**Probability a host is fast**	**Float**
P27	Base number of sources under an access router (**100**)	Integer
P28	**Multiplier by which to scale the base number of sources**	**Float**
P29	**Probability source is located under a typical access router**	**Float**
P30	**Probability source is located under a fast access router**	**Float**
P31	**Probability receiver is located under a typical access router**	**Float**
P32	**Probability receiver is located under a fast access router**	**Float**

MesoNet permits the distribution of sources and receivers to be varied by reallocating some sources and receivers among the three classes of access router (normal, fast and directly connected). Two parameters (P29 and P30) control the allocation of sources (note that the probability a source is allocated to a directly connected access router is equal to 1 – P29 – P30). Similarly, two parameters (P31 and P32) control the allocation of receivers. We chose to combine the three probabilities associated with sources into a single factor and also to combine the three probabilities associated with receivers into a single factor. Thus, the six highlighted parameters in Table 4-7 will comprise only four factors in our sensitivity analysis.

4.2.1.5 Parameters Controlling Source Startup Pattern. Sources are started randomly in stages: some portion start in the ON state, some portion enter the ON state after about 1/3 of the average think time, some portion enter the ON state after about 2/3 of the average think time and the remaining sources enter the ON state after about the average think time. This startup pattern is controlled by three parameters (P33-P35) as shown in Table 4-8. Subtracting the value of these three parameters from one reveals that half of the sources start after about the average think time: 1 – 0.25 – 0.08 – 0.17 = 0.50.

Table 4-8. Parameters Controlling Source Startup Pattern

Parameter	Definition	Type
P33	Portion of sources that start ON (**0.25**)	Float
P34	Portion of sources that come ON after about 1/3 average think time (**0.08**)	Float
P35	Portion of sources that come ON after about 2/3 average think time (**0.17**)	Float

For two reasons, we decided not to vary parameters controlling source startup pattern. First, we discard the first half of our observations and consider only the second half. Thus, the influence of startup pattern should not be evident in the data. Second, we conducted preliminary sensitivity analyses where we varied the startup pattern, along with other parameters, and found that such variations had no influence on the long-term results.

4.2.1.6 Parameters Related to TCP Operation. MesoNet includes only three parameters, given in Table 4-9, controlling the operation of standard TCP. Given lack of widespread agreement on the choice of initial slow-start threshold for TCP, we were interested in exploring the influence of the threshold on network performance. We decided to fix the other two parameters: initial congestion window (P36 = 2) and threshold (P38 = 100) for switching from exponential slow-start increase to logarithmic increase.

Table 4-9. Parameters Related to TCP Operation

Parameter	Definition	Type
P36	Initial TCP congestion window (**2**)	Integer
P37	**Initial slow-start threshold**	**Integer**
P38	Threshold for switching from exponential to logarithm slow-start (**100**)	Integer

Parameter P38 influences slow-start operation only if the value of the initial slow-start threshold (P37) exceeds the value of P38. Assuming this condition, the congestion window begins at the value of P36 and then increases exponentially with each round-trip time until reaching the value of P38, after which the congestion window increases logarithmically until reaching P37 and then linearly. Assuming that P37 < P38, the congestion window increases exponentially until reaching P37 and then linearly. Of course, under either assumption, whenever a loss is encountered, slow-start is abandoned and the congestion window increases linearly when standard TCP is being simulated.

4.2.1.7 Summary of Factors Selected for Sensitivity Analysis. Table 4-10 recaps the eleven factors selected for the sensitivity analysis and the relationship of those factors to MesoNet parameters. Parameters not included in Table 4-10 are assigned fixed values, as indicated in Tables 4-4 through 4-9.

Table 4-10. Recap of Sensitivity Analysis Factors and Mapping to MesoNet Parameters

	Factor	Definition	MesoNet Parameter(s)
Network Factors	x1	Propagation delay	P15
	x2	Network speed	P17
	x3	Buffer sizing	P22
User Factors	x4	Average file size for web pages	P8
	x5	Average think time between web clicks	P9
	x6	Probability a user opts to transfer a larger file	P10
Source & Receiver Factors	x7	Probability a source or receiver is on a fast host	P26
	x8	Scaling factor for number of sources & receivers	P28
	x9	Distribution of sources	P29 & P30
	x10	Distribution of receivers	P31 & P32
Protocol Factors	x11	Initial TCP slow-start threshold	P37

As Table 4-10 demonstrates, the sensitivity analysis is designed to consider the influence of four main classes of factors: (1) network factors, (2) user factors, (3) factors affecting sources and receivers and (4) protocol factors. The factors are fairly balanced with three or four in each category, except for a single protocol factor. In general, the three protocol-related factors might have been fixed, but the inclusion of the initial slow-start threshold as a factor was driven by a specific question, about which the related literature [6, 7, 10] indicates there is no widespread agreement. Now that values have been assigned to 25 fixed parameters, it remains to select the number of levels and settings for the 11 factors identified in Table 4-10. We address that topic next.

4.2.2 Number of Levels and Settings for MesoNet Factors

Adopting the convention of a two-level experiment allows us to produce the kind of OFF designs often used in engineering studies [88, 89, 95] and to benefit from the positive effects such designs have on related analysis techniques. For this reason, we decided to choose two levels for each factor in our sensitivity analysis. Of course, doing so limits our conclusions to the range of settings chosen for our (robustness) factors. Even here we are assuming that the system behaves monotonically in the range between any two settings. If we have reason to believe that behavior is non-monotonic between particular settings, then we should not select such settings for our sensitivity analysis. To extend confidence in the findings produced by our sensitivity analysis, we should explore different specific values for our settings and see whether or not our conclusions also hold. We adopted this supplementary exploration. Here, we focus on our initial sensitivity analysis. We present our supplementary sensitivity analysis in Appendix C.

In choosing specific settings for our two levels (plus and minus) of each factor, we were guided by a desire to complete our 64 experiment runs within a week or so of computing time. We had already determined that computation time in our model was influenced by the number of packets that need to be processed during a simulation. Given that we had decided to fix our simulation to a 20-minute period of network operation, this meant that the computational requirements of our model would be driven largely by the number of sources and the network speed. For this reason, we chose to restrict our simulated network to a few tens of thousands of potential sources and to restrict our network speed to about 10 Gbps in the backbone. Increasing the number of potential sources and the network speed would increase our computational requirements. We decided that increasing the number of sources and the network speed would not be necessary for our sensitivity analysis. Of course, we verified this decision by using more sources and higher backbone speeds in Appendix C.

4.2.2.1 Two-Level Factor Settings. Table 4-11 presents settings chosen for the plus and minus levels for all eleven factors in the sensitivity analysis. Next, we discuss the reasons underlying our choices and the ramifications for the related simulations.

4.2.2.2 Rationale for (and Ramifications of) Network Factor Settings. The topology used in our experiments (recall Fig. 3-1) has defined link propagation delays (recall Table 3-1) that lead to specified minimum round-trip times on designated routes (see Table 3-2). We decided to assign one setting (x1 = 1) to indicate the propagation delays defined in this topology and a second setting (x1 = 2) that doubles those propagation delays. With the minus setting, paths in the topology average a round-trip propagation delay of 41 time steps and a maximum round-trip propagation delay of 100 time steps. This is consistent with a network spanning the United States. When the plus setting is used, average and maximum propagation delays increase to 81 and 200 time steps, respectively. The increased propagation delays are consistent with a network that spans from the west coast of Asia across the United States and into Europe. Note that the setting for propagation delay also influences buffer sizes because the average round-trip propagation delay makes up the *RTT* component of the buffer-sizing algorithms.

Table 4-11. Two-Level Settings for Each of 11 Factors in Sensitivity Analysis

	Factor	Plus	Minus	Parameter Mapping
Network Factors	x1	2	1	+(P15 = 2) or –(P15 = 1)
	x2[4]	400 p/ms	800 p/ms	+(P17 = 400) or –(P17 = 800)
	x3	RTTxC	*RTTxC*/SQRT(*n*)	+(P22 = 1) or +(P22 = 2)
User Factors	x4	100 packets	50 packets	+(P8 = 100) or –(P8 = 50)
	x5	5000 ms	2000 ms	+(P9 = 5000) or –(P9 = 2000)
	x6[5]	0.01	0.02	+(P10 = 0.01) or –(P10 = 0.02)
Source & Receiver Factors	x7[6]	0.2	0.4	+(P26 = 0.2) or –(P26 = 0.4)
	x8	3	2	+(P28 = 3) or –(P28 = 2)
	x9	P2P	WEB	+(P29 = 0.33 and P30 = 0.33) or –(P29 = 0.13 and P30 = 0.53)
	x10	P2P	WEB	+(P31 = 0.33 and P32 = 0.33) or –(P31 = 0.5 and P32 = 0.25)
Protocol Factors	x11	1.07x10^9 packets	43 packets	+(P37 = 1.07x10^9) or –(P37 = 43)

For network speed, we chose to consider a backbone operating near 10 Gbps. Thus, we chose 800 p/ms (packets per millisecond – 8 x 10^5 packets per second) as the top speed of our backbone routers. (8 x 10^5 packets per second x 12 x 10^3 bits per packet = 9.6 Gbps). Of course, modern backbone routers operate at many times this speed; however, we were interested in keeping our simulation time within reason, while still providing some level of load to the simulated network. We chose to define our slower network speed as half our higher speed; thus, we chose 400 packets per millisecond, which equates to a 4.8 Gbps backbone. Note that the choice of backbone router speed determines the choice of router speeds for the other five router types, as shown in Table 4-12. In addition, router speeds influence buffer size because router speed equates to the capacity (*C*) component of the buffer-sizing algorithm.

For buffer sizes, we chose two algorithms. One algorithm, *RTTxC*, instantiates the conventional wisdom [40] regarding how to select buffer sizes to match the expected round-trip time of routes transiting the router and also the capacity of links attached to the router. The second algorithm, *RTTxC*/SQRT(*n*), incorporates an alternate proposal suggesting that one can reduce buffer capacity proportional to the square root of the expected number of flows transiting a router. In the paper proposing the second algorithm

[4] Unfortunately, we coded an increased network speed under the minus setting (and a lower network speed under the plus setting). The reader should bear this in mind when interpreting the results in following sections. Changing this coding would necessitate rerunning the experiment, which would be rather costly.

[5] We also coded this setting incorrectly. Fortunately, this factor doesn't have a large influence on model response, so it does not become confusing in the discussion.

[37], it was left as future work to assess the influence of this algorithm in a large network. This open research question motivated us to include the second buffer-sizing algorithm as an alternative to the typical algorithm.

Table 4-12. Relationship among the Speed of Backbone Routers and Other Router Types (all values given in packets per millisecond)

Router Type	Plus	Minus
Backbone	400	800
POP	100	200
Typical Access	10	20
Fast Access	20	40
Directly Connected Access	100	200

4.2.2.3 Rationale for (and Ramifications of) User Factor Settings. Defining user behavior required selecting three parameters: average file size, average think time and likelihood of downloading a larger file. These parameters are meant to characterize Web users who click from Web page to Web page and occasionally download a picture or a paper or a music file. Previous research [33-36] has established that Internet file sizes exhibit a long-tailed distribution that can be approximated with a Pareto distribution with a shape parameter below 2. We adopted this approach. On the other hand, we need to select an average for the distribution (factor x4). We chose 100 packets (100 packets x 1500 bytes per packet = 1.5 x 10^5 bytes per Web page) as a reasonable size for typical Web pages. We decided to also consider Web pages at half that size: 50 packets (7.5 x 10^4 bytes).

The think time (x5) between Web clicks could be chosen in two different ways. One way is to imagine how long a user typically dwells on a page, while perusing it. Another way is to choose times to obtain a desired load of active users on the network. We took this second approach. A more heavily loaded network would be represented by sources that clicked on a Web link every 2000 milliseconds (x5 = 2 seconds), while we modeled a more lightly loaded network through sources that clicked on a Web link every 5000 milliseconds (x5 = 5 seconds). Of course, this factor interacts with the number of potential users. Many potential users clicking very often create a heavier load and fewer potential users clicking less often create a lighter load. And combinations would fall in between. Note that a heavily loaded network would require users to take longer to transfer their files and thus would mean that users might not be able to arrive for additional transfers quickly because they are slower with ongoing transfers. This implies that there is some dependency-based feedback inherent in the model. Such feedback is probably congruent with the same type of feedback inherent in real networks. The overall effect of this technique for modeling network traffic is not clear, but one must bound the number of simulated users in some fashion.

The probability for a user to decide to download a larger document (x6) represents the possibility that, after looking at a Web page, the user decides to download a paper or a photo or some other document that is larger than a typical Web page. Since we set that file size multiplier to a fixed value (10), a user will download files with an average size of 1.5 Mbytes (x4 = 100) or 750 Kbytes (x4 = 50). As with normal Web objects, these larger documents will be distributed according to a Pareto distribution, which gives a long tail. Lacking concrete measurements, we chose to imagine that a user might download a larger document once in every 100 clicks, so we could set x6 = 0.01.

We also decided to consider the situation where a user downloads a document twice as often (x6 = 0.02), or twice in every 100 clicks.

4.2.2.4 Rationale for (and Ramifications of) Source & Receiver Factor Settings. Defining parameters for sources and receivers required deciding how fast each source or receiver could operate, determining how many sources and receivers existed in the topology, and also indicating the distribution of sources and receivers. These decisions influenced the number of potential active flows and the probability of flows between various classes of access router. We begin by noting that computers connected to the Internet are in transition from slower speed connections (e.g., 100 Mbps) to higher speed connections (e.g., 1 Gbps), so sources and receivers operate on computers with different network-connection speeds. To reflect this, we decided to experiment with two different mixes of computer speeds: 20 % fast computers (x7 = 0.20) and 40 % fast computers (x7 = 0.40).

We began by fixing the base number of sources (P27) under each access router to 100. Since each router has on average four times as many receivers as sources, the base number of receivers becomes 400. We decided to investigate two scaling factors for the number of sources and receivers; we set the scaling factor to either two (x8 = 2) or three (x8 = 3). A scaling factor of two implies that each access router will have around 200 sources and 800 receivers, while a scaling factor of three implies that each access router will have about 300 sources and 1200 receivers. Thus, the total number of potential sources in the network will vary from around 18.56 x 10^3 to 41.7 x 10^3 and the total number of potential receivers will vary from around 111.2 x 10^3 to 219.6 x 10^3.

The average number of sources and receivers under each access router (and also total sources and receivers in the network) will be further adjusted through the distribution pattern assigned to sources (x9) and receivers (x10). The combination of distribution patterns will also affect the number of sources and receivers under each access router and throughout the network. (Sec. 3.2.4 explains the specific relationships that determine the resulting distribution of sources and receivers.)

To recap, given a specified base number of sources and receivers, a scaling factor and a distributional pattern for sources and for receivers, MesoNet populates the network topology with a specified number of sources and receivers and distributes those sources and receivers in the required proportion under each class of access router: normal (N-class[6]) routers, fast (F-class) routers and directly connected (D-class) routers. Table 4-13 shows the resulting distribution of sources for each combination of relevant factors (x8, x9 and x10) used in our sensitivity analysis. Table 4-14 shows the resulting distribution of receivers. A given distribution of sources and receivers also leads to a particular apportioning of flows among the three classes of access router, as shown in Table 4-15.

As the tables indicate, the distributional factors (x9 and x10) control the probability that flows go between specific combinations of access router classes: directly connected to directly connected (DD), directly connected to fast (DF), directly connected to normal (DN), fast to fast (FF), fast to normal (FN) and normal to normal (NN). The scale factor (x8) coupled with the fixed base sources parameter (P27) determines the number of potential active flows (which is also the number of sources).

[6] We continue our convention of color coding designators for access-router classes to match the colors used in Fig. 3-1.

Table 4-13. Relation between Factors and Number and Distribution of Sources

x8	x9	x10	Total Sources	% under D Routers	% under F Routers	% under N Routers
2	P2P	P2P	27.8×10^3	4.32	20.14	75.54
3	P2P	P2P	41.7×10^3	4.32	20.14	75.54
2	WEB	WEB	18.56×10^3	6.46	48.27	45.25
3	WEB	WEB	27.84×10^3	6.46	48.27	45.25
2	P2P	WEB	27.8×10^3	4.32	20.14	75.54
3	P2P	WEB	41.7×10^3	4.32	20.14	75.54
2	WEB	P2P	18.56×10^3	6.46	48.27	45.25
3	WEB	P2P	27.84×10^3	6.46	48.27	45.25

Table 4-14. Relation between Factors and Number and Distribution of Receivers

x8	x9	x10	Total Receivers	% under D Routers	% under F Routers	% under N Routers
2	P2P	P2P	111.2×10^3	4.32	20.14	75.54
3	P2P	P2P	166.8×10^3	4.32	20.14	75.54
2	WEB	WEB	146.4×10^3	2.45	11.47	86.06
3	WEB	WEB	219.6×10^3	2.45	11.47	86.06
2	P2P	WEB	146.4×10^3	2.45	11.47	86.06
3	P2P	WEB	219.6×10^3	2.45	11.47	86.06
2	WEB	P2P	111.2×10^3	4.32	20.14	75.54
3	WEB	P2P	166.8×10^3	4.32	20.14	75.54

Table 4-15. Relation between Factors and Distribution of Flow Classes

x8	x9	x10	% DD Flows	% DF Flows	% DN Flows	% FF Flows	% FN Flows	% NN Flows
2	P2P	P2P	0.186	1.74	6.52	4.05	30.43	57.06
3	P2P	P2P	0.186	1.74	6.52	4.05	30.43	57.06
2	WEB	WEB	0.159	1.92	6.67	5.53	46.74	38.95
3	WEB	WEB	0.159	1.92	6.67	5.53	46.74	38.95
2	P2P	WEB	0.106	0.99	5.57	2.31	26.00	65.01
3	P2P	WEB	0.106	0.99	5.57	2.31	26.00	65.01
2	WEB	P2P	0.279	3.38	6.83	9.72	45.58	34.18
3	WEB	P2P	0.279	3.38	6.83	9.72	45.58	34.18

One final note: the number and distribution of sources and receivers also influences the determination of router buffer sizes when using the *RTT*x*C*/SQRT(*n*) algorithm. The *RTT*x*C* algorithm computes buffer sizes based on multiplying the average round-trip propagation delay in the network by the capacity of each router. Table 4-16 shows the results for this algorithm when using the factor values adopted in this sensitivity analysis. When switching to the *RTT*x*C*/SQRT(*n*) algorithm, the values in

Table 4-16 are divided by the estimated average number of active flows expected to transit each router. This estimate depends on the number and distribution of sources and receivers throughout the topology. In general, using the *RTT*x*C*/SQRT(*n*) algorithm reduces buffers within routers by one or two orders of magnitude.

Table 4-16. Buffers for Combinations of Round-Trip Propagation Delay (x1) and Capacity (x2)

x1	x2	Backbone Router Buffers (avg.)	POP Router Buffers (avg.)	Access Router Buffers (avg.)
1	400	16.277 x 10^3	4.070 x 10^3	647
2	400	32.553 x 10^3	8.139 x 10^3	1.294 x 10^3
1	800	32.553 x 10^3	8.139 x 10^3	1.294 x 10^3
2	800	65.106 x 10^3	16.277 x 10^3	2.588 x 10^3

4.2.2.5 Rationale for (and Ramifications of) Protocol Factor Settings. After investigating the literature, we came to the realization that there is no consensus value to use for the initial TCP slow-start threshold. Some authors [4] suggest using the receive window provided by a corresponding TCP entity. Some authors [10] suggest picking a small value. Some authors [6] suggest picking a very large number. A colleague, Mark Carson (personal communication, November 12, 2008) indicated that some operating systems select this value based upon characteristics of the local network card. Given this general lack of consensus, we decided to include the initial TCP slow-start threshold as a factor (x11) in our sensitivity analysis. We decided there were two main schools of thought about choosing a value: choose a small value and choose a large value. To represent the small-value school of thought, we chose (x11 =) 43 packets, which was recommended by Stevens [10]. To represent the large-value school of thought, we chose an arbitrarily large value of (x11 =) 1.07x10^9 packets, as suggested by Fall [6]. We also adopted the recommendation of Floyd [7], where a flow increases its sending rate exponentially up to a congestion window of 100 and then logarithmically until a higher threshold is reached or loss encountered. The rationale for choosing a large value derives from the purpose of initial slow-start: to quickly determine how fast a source may send on a given path. Choosing a small value could lead a flow to switch to a linear increase prior to achieving its maximum transmission rate, so a flow might end before maximum rate is achieved. We decided to see what difference the choice of initial TCP slow-start threshold would make given our other factors and parameter settings.

4.2.3 Specific Combinations Simulated

Given 11 factors, each with two possible levels, a full factorial experiment would require (2^{11} =) 2048 simulation runs. Assuming an average run takes about 8.5 processor hours, conducting all these simulation runs would require 17.408 x 10^3 processor hours. If we split these among 24 processors, we could complete the work in about 725 hours – or 30 days. We preferred to be able to complete our simulations within a week, so we adopted a 2^{11-5} orthogonal fractional factorial (OFF) design that required only 64 simulation runs. The design can be found in Fig. 4-1. To generate our parameterized runs, we set our fixed factors to the values indicated in Tables 4-8 through 4-9 and then we generated 64 configuration files that varied the factors (x1 to x11) as instructed by Fig. 4-1 – taking from Table 4-11 the PLUS values to substitute for the +1 designators in Fig. 4-1and the MINUS values to substitute for the -1 designators in Fig. 4-1.

4.3 Experiment Execution

The experiment plan required 64 simulation runs, each simulating a different combination of factor settings (recall Fig. 4-1). We had 28 physical processors[7] on which we could run our experiments, so we could conduct simulations in parallel. However, we were sharing these processors with other projects, so we could not always use all of the available processors. Below, we give a brief discussion of the resource requirements for the simulations and then we recount our approach to data collection and summarization.

4.3.1 Resource Requirements for Simulations

Table 4-17 reports the characteristics of the 28 processors available for our sensitivity analysis. Since MesoNet is implemented in SLX, each of the processors had access to an SLX simulation environment. SLX comes in two varieties: one configured to run in a 32-bit address space and one configured to run in a 64-bit address space. Some of the available processors were configured with a 64-bit operating system, which could support both the 32-bit and 64-bit versions of SLX. We chose to run all our simulations using the 32-bit version of SLX. We made this choice because our simulations could easily fit within a 32-bit address space and 32-bit simulation runs faster than 64-bit simulation. This is true largely because 64-bit simulation requires the use of 64-bit arithmetic when manipulating pointers that address simulation objects. Also 64-bit simulation requires more memory than 32-bit simulation because of the doubling of size for address pointers. For these reasons, 64-bit simulation should be reserved for situations where the size of the simulation cannot be contained within a 32-bit address space.

Table 4-17. Characteristics of Processors Executing Simulation Runs

Node	Physical Processors	Speed (GHz)	Hyperthreaded	Memory (GB)	Operating System
ws7	4	3.66	Yes	20	Windows Server 2003 R2 x64 Edition SP2
ws8	4	3.66	Yes	20	Windows Server 2003 R2 x64 Edition SP2
ws9	8	2.6	No	32	Windows Server 2003 R2 x64 Edition SP2
ws10	8	2.6	No	32	Windows Server 2003 R2 x64 Edition SP2
DT	4	3.2	No	3	Windows XP SP2

We executed the simulations in three rounds (runs 1-35, runs 36-50 and runs 51-64) over about one week. All simulation runs required a similar amount of memory: on the order of 120 Mbytes. On the other hand, simulation runs required varying amounts of

[7] Hyperthreading^{TM} was enabled on 8 of these physical processors. Hyperthreading creates two independent logical threads on a single physical processor. With hyperthreading the number of available logical processors totaled (28 + 4 x 2 =) 36. On hyperthreaded processors, our simulations ran at (or below) half the speed that was possible without using hyperthreading. The reader should take this into account when interpreting the execution time requirements given in Table 4-18.

processor time, depending on the specific combination of factors and on the specific node used to execute the simulation. Table 4-18 recounts the execution time used for each simulation run. Executing all 64 runs required a total of 537.6 hours of processing time, which amounts to 8.4 hours on average per run. However, due to the fact that ws7 and ws8 used hyperthreading, this figure is somewhat misleading.

Table 4-18. Execution Time (Hours) Required for Each Simulation Run

Run	Node	Time	Run	Node	Time	Run	Node	Time	Run	Node	Time
1	ws9	7.7	17	ws7	13.8	33	DT	10.7	49	DT	6.8
2	ws9	6.2	18	ws7	12.2	34	DT	6.5	50	DT	2.4
3	ws9	3.8	19	ws7	11.5	35	DT	5.2	51	ws9	2
4	ws9	4.3	20	ws7	12.9	36	ws9	4.5	52	ws9	3.6
5	ws9	4.9	21	ws7	8.8	37	ws9	7.3	53	ws9	2.8
6	ws9	9.2	22	ws7	15.6	38	ws9	6.9	54	ws9	3.1
7	ws9	5.1	23	ws7	15.4	39	ws10	5.7	55	ws9	3
8	ws9	4.1	24	ws7	8.4	40	ws9	4.9	56	ws9	3.2
9	ws10	7.5	25	ws8	16.7	41	ws9	8.2	57	ws9	5.7
10	ws10	8.8	26	ws8	24.6	42	ws10	8.3	58	ws10	5.6
11	ws10	6.1	27	ws8	19.1	43	ws10	4.9	59	ws10	5
12	ws10	4.3	28	ws8	16.4	44	ws9	4.8	60	ws10	4.1
13	ws10	10.2	29	ws8	24.7	45	ws10	9.2	61	ws10	7.5
14	ws10	8.5	30	ws8	22.5	46	ws9	8.2	62	ws10	4
15	ws10	5.6	31	ws8	19	47	ws10	5.1	63	ws10	3.8
16	ws10	5.1	32	ws8	19.9	48	DT	6.8	64	ws10	4.9

Considering the processing time required for runs on individual nodes, runs on ws9 averaged 5.2 hours, runs on ws10 averaged 6.2 hours, runs on DT averaged 6.4 hours, runs on ws7 averaged 12.3 hours and runs on ws8 averaged 20.4 hours. Grouping nodes into those that were not hyperthreaded (ws9, ws10 and DT) and those that were hyperthreaded (ws7 and ws8), we found that the hyperthreaded nodes required an average of 16.3 hours per run, while the non-hyperthreaded nodes required an average of 5.8 hours per run. Thus, the hyperthreaded nodes took an average of 2.8 times longer than the non-hyperthreaded nodes to execute a simulation run. This suggests that the hyperthreaded processors ran at about 36 % the speed of the non-hyperthreaded processors. Of course, to gauge the effects due to hyperthreading alone, one must account for the fact that the processor speeds of the hyperthreaded nodes were different than the processor speeds of the non-hyperthreaded nodes.

Given that the processor speed of ws7 (and ws8) is 3.66 GHz, one would expect hyperthreading to provide half the processing speed, or (3.66/2 =) 1.83 GHz, to each logical thread. Thus, one might expect that it would take (2.6/1.83 =) 1.42 times longer to run simulations on ws7 (and ws8) than on ws9 (and ws10). We found that on average it took 2.8 times longer to run simulations on the hyperthreaded nodes. These findings do not provide a complete characterization of differences between hyperthreaded and non-hyperthreaded operations. First, we did not run the same workload on both types of processors, as we split various experiment configurations among the processors. Second, the hyperthreaded processors employed chip architectures (Intel Xenon MP) different from some of the non-hyperthreaded processors (ws9 and ws10 used AMD Opteron 8218 and DT used Intel Xenon).

4.3.2 Data Collection and Summarization

MesoNet records response data as time series. This allows monitoring response changes over time. For example, Fig. 4-11 shows the time series for the number of active flows (response y1) during run 64 of our sensitivity analysis. As shown, the time series reports the number of active flows (y axis) at the end of each of measurement interval for each of 6000 measurement intervals (x axis) recorded during the simulation run. To facilitate our analyses, we summarize each response to an average value for each run. As illustrated in Fig. 4-11, we do this by discarding the first half of the data (measurement intervals 1 to 3000) and then computing the average value for the remaining data (measurement intervals 3001-6000). As illustrated in Fig. 4-11, discarding the first 3000 measurement intervals eliminates transient startup effects and enables us to retain behavior representative of the model operating in steady-state. In this case, for run 64, the mean value of y1 over measurement intervals 3001-6000 is 214.676 x 10^2 flows.

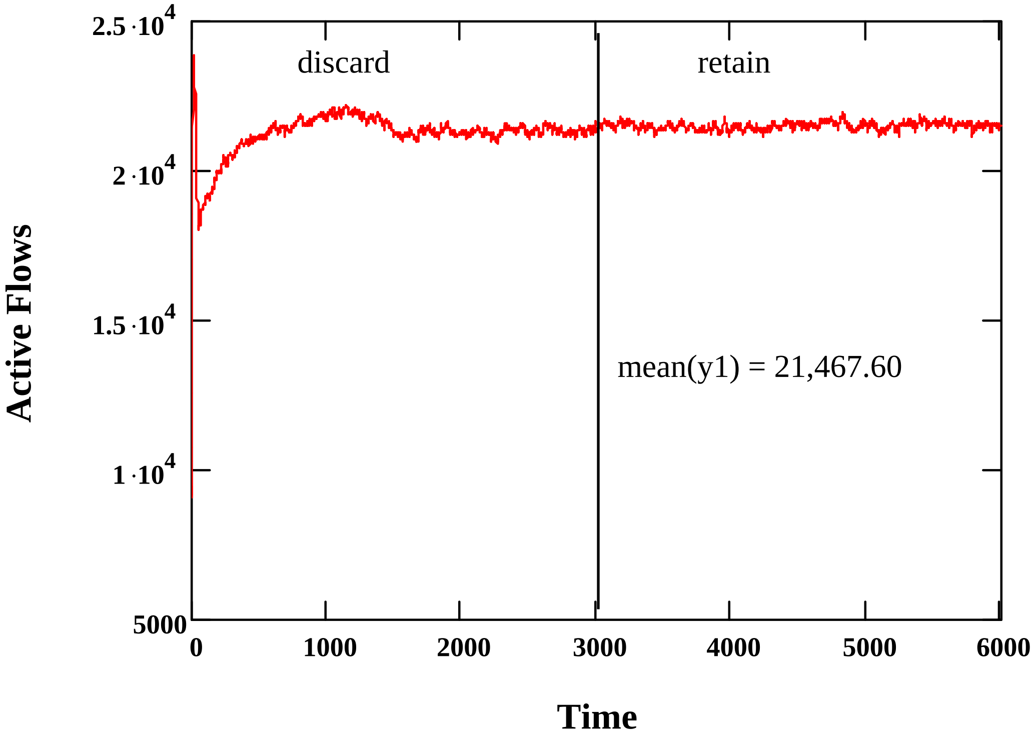
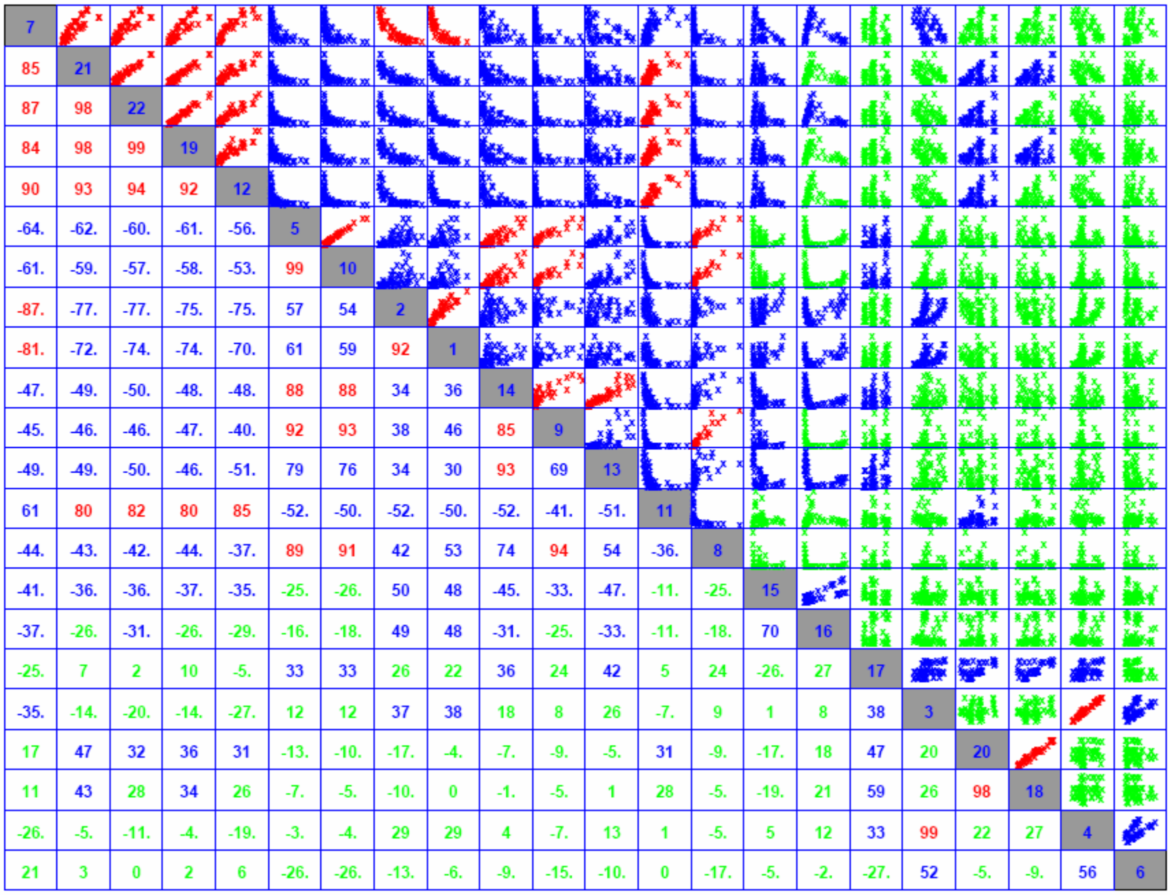


Figure 4-11. Example Illustrating the Technique used to Summarize System Responses (x axis gives the number of active flows and y axis gives time in 200 ms intervals)

We conduct such a summarization for all 22 responses under each of the 64 conditions and collect the summarizations into a table, as shown in Fig. 4-12. For example, the value placed in the cell for response y1 and run 64 in Fig. 4-12 is the value we computed in Fig. 4-11. The summarization table forms the basis for all of our analyses.

Run	y1	y2	…	y21	y22
1	4680.619	0.168126	…	92.034	89.785
2	6654.512	0.239371	…	72.596	57.738
3	9431.405	0.339259	…	29.569	13.963
4	11565.81	0.415439	…	23.427	19.882
…	…	…	…	…	…
61	10319.55	0.247471	…	87.969	41.573
62	1738.469	0.093668	…	159.298	161.602
63	1783.509	0.096094	…	148.395	161.36
64	21467.6	0.514811	…	26.159	9.981

Figure 4-12. Sample Data Summarization: 22 Responses for each of 64 Simulation Runs



7																					
85	21																				
87	98	22																			
84	98	99	19																		
90	93	94	92	12																	
-64.	-62.	-60.	-61.	-56.	5																
-61.	-59.	-57.	-58.	-53.	99	10															
-87.	-77.	-77.	-75.	-75.	57	54	2														
-81.	-72.	-74.	-74.	-70.	61	59	92	1													
-47.	-49.	-50.	-48.	-48.	88	88	34	36	14												
-45.	-46.	-46.	-47.	-40.	92	93	38	46	85	9											
-49.	-49.	-50.	-46.	-51.	79	76	34	30	93	69	13										
61	80	82	80	85	-52.	-50.	-52.	-50.	-52.	-41.	-51.	11									
-44.	-43.	-42.	-44.	-37.	89	91	42	53	74	94	54	-36.	8								
-41.	-36.	-36.	-37.	-35.	-25.	-26.	50	48	-45.	-33.	-47.	-11.	-25.	15							
-37.	-26.	-31.	-26.	-29.	-16.	-18.	49	48	-31.	-25.	-33.	-11.	-18.	70	16						
-25.	7	2	10	-5.	33	33	26	22	36	24	42	5	24	-26.	27	17					
-35.	-14.	-20.	-14.	-27.	12	12	37	38	18	8	26	-7.	9	1	8	38	3				
17	47	32	36	31	-13.	-10.	-17.	-4.	-7.	-9.	-5.	31	-9.	-17.	18	47	20	20			
11	43	28	34	26	-7.	-5.	-10.	0	-1.	-5.	1	28	-5.	-19.	21	59	26	98	18		
-26.	-5.	-11.	-4.	-19.	-3.	-4.	29	29	4	-7.	13	1	-5.	5	12	33	99	22	27	4	
21	3	0	2	6	-26.	-26.	-13.	-6.	-9.	-15.	-10.	0	-17.	-5.	-2.	-27.	52	-5.	-9.	56	6

Figure 4-13. Combined Matrix of Scatter Plots and Correlation Values for 22 Responses

4.4 Correlation Analysis and Clustering

Given 64 average values (one per run) for 22 responses, correlation analysis investigates the degree to which pairs of responses are correlated. Recall that Tables 4-1 and 4-2 identify the 22 responses. We begin by generating a scatter plot and computing the correlation for each pair of responses. Then we plot (in Fig. 4-13) the results as a combined matrix of scatter plots and correlation values. We order the diagonal by decreasing average correlation for each response with the 21 other responses. The highest average correlation is for response y7 and the lowest is for response y6. Correlations of .8 and above are colored red, correlations between .3 and .79 are colored blue and correlations below .3 are colored green.

Fig. 4-13 reveals some correlation groupings. For example, responses y7, y21, y22, y19, y12, y11, y1 and y2 show mutual correlations. Responses y5, y10, y14, y8 and y9 also exhibit mutual correlations. Strong correlations appear between selected pairs of responses: y21 and y22; y22 and y19; y5 and y10; y1 and y2; y8 and y9; y13 and y14; y18 and y20; y3 and y4. These mutual correlations suggest that it should prove feasible to reduce the number of responses examined from 22 to some lower dimension. On the other hand, a few responses (e.g., y6 and y17) appear largely uncorrelated with other responses.

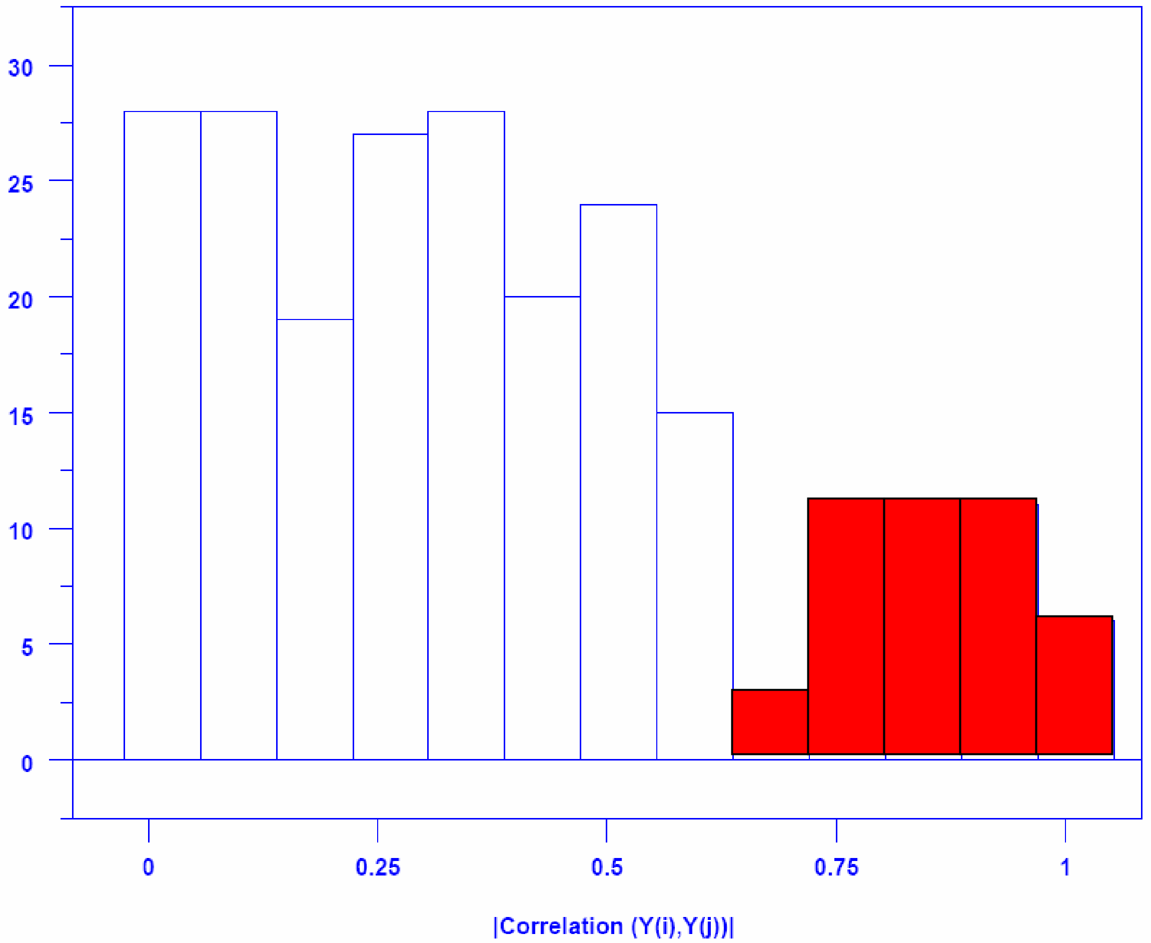


Figure 4-14. Frequency Distribution of the Absolute Value of Correlations for All Pairs of Responses

To identify particular correlation groups, we need to select a threshold for the absolute value of correlations such that above that threshold we will consider correlations sufficiently strong to warrant inclusion in further analyses, while we will discard correlations below that threshold. To help identify a reasonable threshold, we plot (in Fig. 4-14) a frequency distribution of the absolute values of all correlation pairs.

In Fig. 4-14, we emphasize (in red) the range of correlations that appear most significant because there is a notable change above that value, appearing as a separate sub-distribution centered on a different mode. The range of correlations emphasized in Fig. 4-14 run from about 0.65 to 1.0, so we decided to use correlations whose absolute value exceeds 0.65. We discard correlations with lower absolute values. For the correlations retained, we produced an index-index plot (recall Fig. 4-6). In Fig. 4-15 we reorder the indices from Fig. 4-6 (on both the ordinate and abscissa) by increasing total number of variables exhibiting above threshold correlation with the designated variable. Where the count of mutual correlations is the same, our order is arbitrary. We begin with responses y6 and y17, which have no retained correlations. For those responses, we order y17 first because it has only one mutual correlation > 0.5, while y6 has two such correlations – thus, y17 is somewhat less correlated with other responses than is y6.

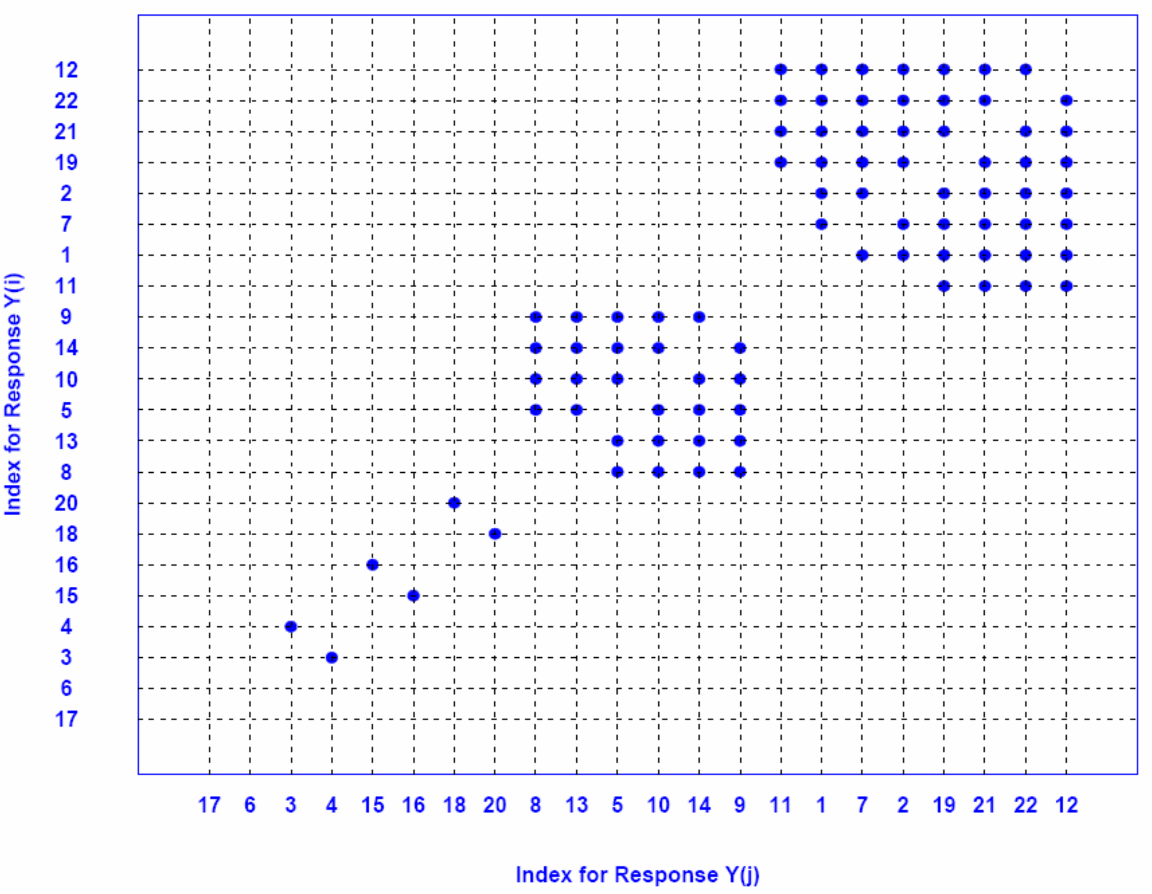


Figure 4-15. Index-Index Plot for Correlation Pairs where |Correlation (Y_i, Y_j)| > 0.65

Fig. 4-15 identifies seven clear correlation groups: y17 (no correlations); y6 (no correlations); y3 and y4 (pair-wise correlation); y15 and y16 (pair-wise correlation); y18 and y20 (pair-wise correlation); y8, y13, y5, y10, y14 and y9 (28 mutual correlations); y11, y1, y7, y2, y19, y21, y22 and y12 (50 mutual correlations). This suggests that we can characterize system response through seven, rather than 22, responses. Next, we address the issue of whether or not the seven correlation groupings make sense from the perspective of the network simulation model. We also discuss what information may be conveyed by lack of correlation. We begin our discussion with the three mutually correlated pairs and then consider the group with 28 mutual correlations, followed by the group with 50 mutual correlations. We close by considering the two uncorrelated responses.

The rate of data packets injected into the network (y3) is highly correlated (0.99) with the rate of packets leaving (y4). This strong correlation is expected because packets must enter the network before they can exit and the rate of entry and exit should be balanced (unless many packets are lost within the network). Perhaps more surprising is the fact that the rate of packets entering and exiting the network is not strongly correlated with any other responses. The closest correlation (around 0.5) is with the rate of flow completions (y6), which is largely uncorrelated with any other responses. One might expect correlation between the number of active flows (y1) and the number of packets entering and leaving the network, but this is not the case. From this, we conclude that the rate of packets flowing through the network is influenced by factors different from those influencing the number of active flows. Thus, our sensitivity analysis needs to consider either the rate of packets entering or leaving the network but not both.

The (SRTT) smoothed round-trip time (y15) and relative queuing delay (y16) are somewhat correlated (0.7). This makes sense because the relative queuing delay is computed by transforming the SRTT. The correlation is not particularly strong because the relative queuing delay factors out the propagation delay and gives enhanced weight to time spent in buffers. Buffer size has a greater influence on y16, while that influence is somewhat diluted (by propagation delay) in y15. The low strength of the correlation suggests that our sensitivity analysis should consider both y15 and y16. On the other hand, the reasons underlying the correlation suggest that perhaps we could use only y15, which captures influences due to both propagation delay and queuing delay.

The average instantaneous throughput for DF flows (y18) is strongly correlated (0.98) with the throughput for FF flows (y20). This reflects the fact that throughput is constrained by the capabilities of the slower of the two access-router classes over which such flows transit. This strong correlation implies that we need only consider one of these two responses for our sensitivity analysis.

The next correlation group in Fig. 4-15 consists of 28 mutual correlations among six responses: loss rate (y5); connection failures (y8) and connection-failure rate (y9); retransmission rate (y10); negative acknowledgment rate (y13) and timeout rate (y14). Most of these mutual correlations exceed 0.8. The correlations among these responses appear reasonable because packet losses have numerous consequences: negative acknowledgments or timeouts, connection failures and retransmissions. The strongest correlation (0.99) exists between loss rate and retransmission rate. In fact, since both data packets and acknowledgments may be lost, one would expect the retransmission rate to be about twice the loss rate. The (0.89) correlation between loss rate and connection failures is lower because connection attempts are retried; three connection attempts must be lost before a connection fails. The (0.79) correlation between loss rate and negative acknowledgment rate (as seen by sources) is lower because negative acknowledgments may also be lost; losses push up the rate at which receivers send negative acknowledgments but also increase probability that negative acknowledgments are lost. When acknowledgments (negative or positive) are lost, the rate of timeouts increases, so there is a higher correlation (0.88) between loss rate and timeouts. The six responses in this correlation group are measures of packet losses and the ensuing consequences for the network. Our sensitivity analysis need only consider one of these responses, such as retransmission rate (y10), which reflects both packet losses and the packets resent to recover from losses.

The final correlation group consists of 50 mutual correlations among eight responses: active flows (y1) and proportion of possible flows that are active (y2); flow-completion rate (y7); average congestion window (y11) and window-increase rate (y12); and average instantaneous throughput for DN (y19), FN (y21) and NN (y22) flows. Most correlations, which are negative, stem from sharing network resources. Increasing active flows leads to decreases in flow-completion rate, the average congestion window, window increase rate and instantaneous throughput for flows transiting normal access routers. Flows (DN, FN and NN) transiting normal access routers are most numerous; sharing access routers affects the throughput of these flows. As the number of flows transiting an access router increases, each flow receives a lower share of the bandwidth and so will receive lower throughput. Lower throughput implies smaller congestion windows. Smaller congestion windows imply a slower rate of window increases. More active connections also imply a lower rate of connection completion. Note, however, that the (-0.5) correlation between active connections and average congestion window is not strong enough to be included in this correlation group. Stronger correlations exist between average congestion window and flow throughputs (about 0.8) and window increase rate (0.85). This suggests that congestion window size is influenced by factors not solely related to the number of active connections. In fact, in Sec. 4.1.5 we showed that congestion window size is influenced by network speed, buffer-sizing algorithm and initial slow-start threshold, as well as by factors that influence the number of active connections. For our sensitivity analysis we can select one response (such as y22) to reflect the degree of sharing among common network resources. We should probably also include the number of active flows in order to investigate what factors influence the need to share resources.

The two remaining responses, flows completed (y6) and average instantaneous throughput for DD flows (y17), are uncorrelated with other responses. Apparently, the number of flows completed is driven by factors different from the factors driving other responses. The reason for this is not obvious. Throughput for DD flows is also driven by factors different from the factors influencing throughput for other flow classes. The reason for this appears straightforward. First, DD flows are relatively few in number, when compared with other flow classes. Second, DD flows cross high-speed access routers that are connected directly to backbone routers, so DD flows see less contention for bandwidth on the ingress and egress paths of the network. Since y6 and y17 are uncorrelated with other responses, we must include them in our sensitivity analysis.

To recap, Table 4-19 identifies the responses we chose to investigate during our sensitivity analysis. Correlation analysis suggested that we could characterize system response using only seven of 22 responses. We decided to include an eighth response: the number of active flows (y1). This added response allowed us to consider which factors lead to an increased number of active flows, a main influence on the degree of resource sharing required within a network. Two responses deal with the aggregate throughput of packets (y4) and flows (y6). One response (y10) reflects the degree and consequences of packet losses. One response (y15) mirrors the degree of network delay. The remaining responses gauge throughput for flows constrained by transiting directly connected (y17), fast (y20) or normal (y22) access routers.

Table 4-19. Responses Selected for Investigation in Sensitivity Analysis

Response	Definition
y1	Average number of active flows
y4	Average number of packet output per measurement interval
y6	Average number of flows completed per measurement interval
y10	Average retransmission rate
y15	Average smoothed round-trip time
y17	Average instantaneous throughput for DD flows
y20	Average instantaneous throughput for FF flows
y22	Average instantaneous throughput for NN flows

4.5 Principal Components Analysis

Principal components analysis (PCA) is an alternative (or complementary) technique often used to assess the covariance structure of a set responses [96]. In this section, we describe the findings of a PCA applied to the 22 responses from our sensitivity analysis simulation runs. As the first step in the PCA, we transform our data responses into a standardized form by subtracting the mean value (over all 64 conditions) from each response to yield (22 x 64 =) 1408 normalized data points, as discussed previously in Sec. 4.1.4. In this way, all responses are placed on an equivalent scale with respect to variance around the mean. Next, we find a weight vector that yields the maximum possible variance (or standard deviation), subject to the constraint that the sum of all weights (with each weight squared) is equal to one. We repeat this process, possibly up to the total number of responses, and each time require the weights selected to be orthogonal to the weights used in previous steps. Using this technique we are looking for the largest sources of variation in different directions through the data with each step. Each different direction through the data is considered a principal component. The amount of variation accounted for diminishes with each principal component considered. At some point, most of the variance will be accounted for and one could stop the analysis.

For example, consider Figure 4-16, which displays the results of a PCA for the 22 responses from our sensitivity simulations. The detailed layout of each sub-plot was explained previously in Sec. 4.1.4 (recall Fig. 4-8). The upper left-hand plot depicts the standard deviation (SD) across all normalized responses (y1 through y22). The remaining 22 plots show the standard deviation accounted for by each of 22 principal components in decreasing order of magnitude. We note that most (about 86 %) of the variation in the data is accounted for by the first four principal components. Next, we plot the weights associated with each response in each of the first four principal components. Figure 4-17 shows this information. The detailed layout of each sub-plot was explained previously in Sec. 4.1.4 (recall Fig. 4-7).

The results of the PCA suggest that the behavior of our model can be represented with as few as four (statistically uncorrelated) responses, instead of the seven responses suggested by our correlation analysis. Further, these four principal components, or PCs, are linear combinations of many regular responses. Extracting responses from each PC in Fig. 4-17 using heuristics mentioned in Sec. 4.1.4, we can group responses by principal component, as shown in Tables 4-20 through 4-23.

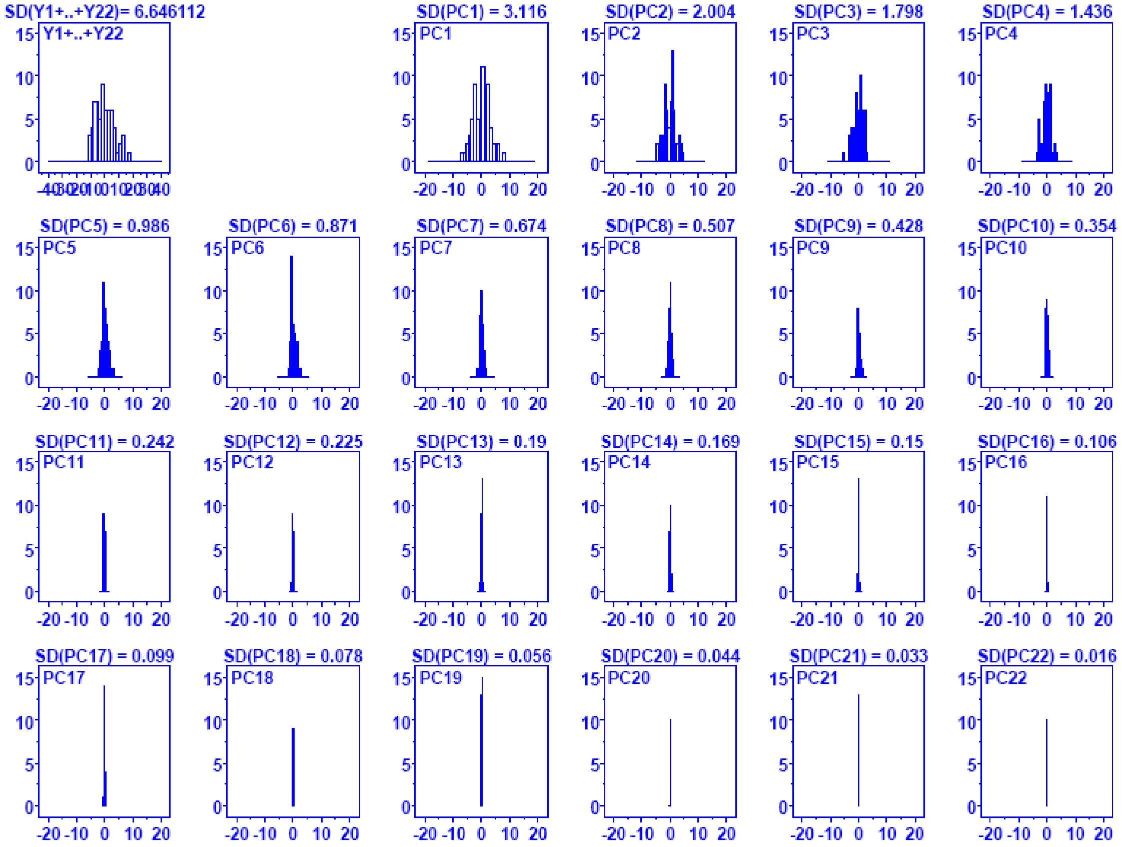


Figure 4-16. Histograms for 22 Principal Components (x axis of each sub-plot identifies bins of normalized component values ranging from -20 to +20 and y axis the count of values within each bin). Above each sub-plot is the standard deviation in the data accounted for by the Principal Component. The first sub-plot gives the distribution of the normalized responses.

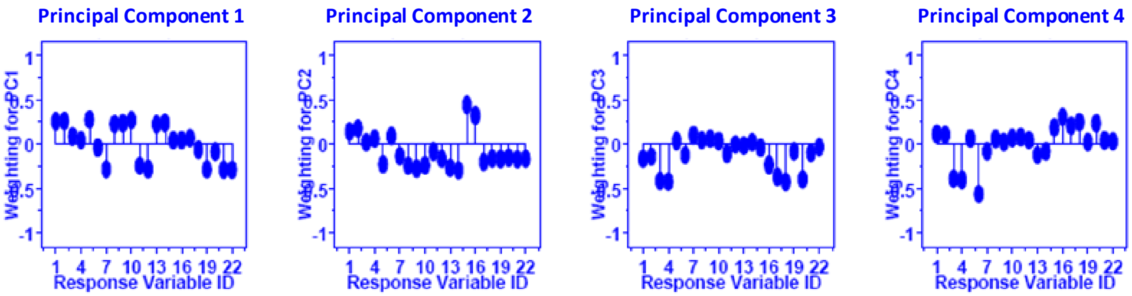


Figure 4-17. Weight Vectors for the First Four Principal Components

The first principal component (Table 4-20) combines two response groups identified in the correlation analysis. One group represents the effects of resource sharing and congestion on the throughput of flows that transit typical access routers. Such flows are most numerous in any given simulation. The second group represents the level of congestion present in the network. Congestion occurs most often at access routers. Thus, the PCA finds that the largest source of variance in the 22 responses arises from the level of congestion at access routers in the simulated network.

Table 4-20 Responses Composing Principal Component One

Correlation Cluster	Response	Definition
Effects of Resource Sharing and Congestion on Throughput in Flows Transiting Typical Access Routers	y1	Average number of active flows
	y2	Proportion of possible flows that are active
	y7	Flow-completion rate
	y11	Average congestion window
	y12	Window-increase rate
	y19	Average instantaneous throughput for DN flows
	y21	Average instantaneous throughput for FN flows
	y22	Average instantaneous throughput for NN flows
Overall Network Congestion	y5	Loss rate
	y8	Connection failures
	y9	Connection-failure rate
	y10	Retransmission rate
	y13	Negative-acknowledgment rate
	y14	Timeout rate

Table 4-21. Responses Composing Principal Component Two

Response	Definition
y15	Smoothed round-trip time
y16	Relative queuing delay

The second principal component (Table 4-21) corresponds to a pair of responses (y15 and y16) grouped together by the correlation analysis. These responses represent the level of delay within the network.

Table 4-22. Responses Composing Principal Component Three

Correlation Cluster	Response	Definition
Packet Throughput	y3	Packets input
	y4	Packets output
DD-flow Throughput	y17	Average instantaneous throughput for DD flows
DF- & FF-flow Throughput	y18	Average instantaneous throughput for DF flows
	y20	Average instantaneous throughput for FF flows

The third principal component (Table 4-22) unites three separate groupings found in the correlation analysis. One group represents the number of data packets flowing in and out of the network, which correlation analysis suggested were not strongly correlated with other responses. Note, though, that there were moderate correlations with throughput on faster flows (y17, y18 and y20) and with the number of flows completing (y6). In fact, the PCA assigns similar weights for y3 and y4 in both principal components three and four. (For this reason, we also include y3 and y4 in the grouping associated with principal component four.) Responses relating to throughputs for flows transiting only fast and directly-connected routers were grouped together by the principal components analysis, while correlation analysis separated these responses. Principal component three also seems to include the effects of the higher throughput flows on packets flowing into and out of the network.

The fourth principal component represents the ability of the network to complete flows. Included in this component is the association with packets entering and leaving the network. If called upon to place y3 and y4 into only a single principal component, we would choose to place them into PC4. On the other hand, as shown in Table 4-23, PC4 unites two separate groupings found in the correlation analysis.

Table 4-23. Responses Composing Principal Component Four

Correlation Cluster	Response	Definition
Packet Throughput	y3	Packets input
	y4	Packets output
Flow Throughput	y6	Flows completed per measurement interval

The principal components analysis both confirms the findings of the correlation analysis and also provides additional information. For example, the PCA groups together the symptoms and effects of congestion. This appears sensible. The PCA also reveals a connection between packets in and out and two other groupings: throughput on high-throughput flows and the number of connections completed. The correlation analysis hinted at these connections. The PCA suggests that throughput on DD flows should be grouped together with throughput on DF and FF flows; the correlation analysis indicated that DD flows should be studied separately. We will use findings from both the correlation and principal components analyses as we investigate the sensitivity of model responses to input parameters.

4.6 Sensitivity Analysis

In this section, we use the experiment design, the model responses and the results of the correlation and principal components analyses to assess the sensitivity of MesoNet to changes in eleven input factors. We begin by exploring how model inputs affect the eight responses identified by our correlation analysis (recall Table 4-19). Subsequently, we consider how the four main principal components (recall Tables 4-20 through 4-23) vary with changes in input factors.

4.6.1 Sensitivity Analysis Guided by Correlation Analysis

We begin by exploring how model inputs affect three, congestion-related responses: number of active flows (y1), retransmission rate (y10) and average instantaneous throughput for NN flows (y22). Subsequently, we consider the five remaining responses in the following order: average smoothed round-trip time (y15), rate of data packets output (y4), number of flows completed per measurement interval (y6) and average instantaneous throughput for DD flows (y17) and for FF flows (y20).

4.6.1.1 Congestion-Related Responses. For the topology and experiment design we adopted, flows transiting through the slowest (N-class) access routers were most numerous. For this reason, congestion tends to occur most often in N-class access routers, which affects the throughput of flows transiting such routers. The affected flows include DN, FN and NN flows, which our analysis showed to be significantly correlated. We selected NN flows as a representative flow class to consider. The throughput experienced by NN flows is likely to be affected by the number of active flows transiting N-class access routers and by the retransmission rate on those flows. Since flows transiting N-class access routers are most numerous, macroscopic measures of the number of active flows and the retransmission rate network-wide should be indicative of the level of congestion experienced by NN flows. For these reasons, we decided to consider y1, y10 and y22 as a related set of responses.

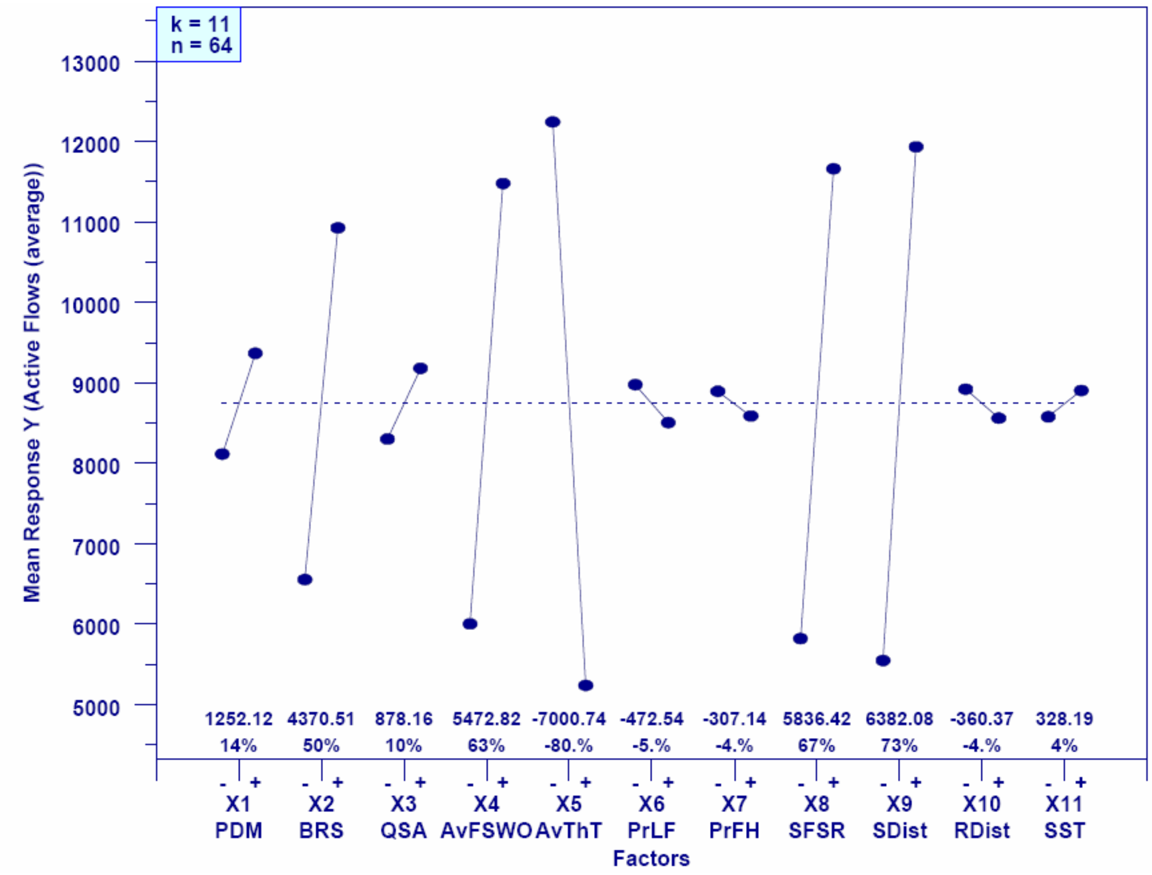


Figure 4-18. Main-Effects Plot for Response y1 (Average Number of Active Flows)

We decided to first examine factors that influence the number of active flows in the network, since the number of active flows is likely to affect congestion. Fig. 4-18 gives the main-effects plot highlighting factors influencing the number of active flows. The main factors appear to fall into three categories: (a) number of sources underneath N-class access routers, (b) idle interval for those sources and (c) duration for which flows remain active. The think time of sources (x5) is the main influence on the number of active flows. The shorter the think time the more often sources become active and attempt to transfer files (i.e., sequences of packets). Naturally, the more sources that exist under N-class access routers, the greater will be the effects of shorter think time. The number of sources under N-class access routers is influenced by two factors: the base number (x8) of sources used to populate the topology and the distribution (x9) of those sources. The plus setting for x9 increases the probability flows will exist between sources and receivers under N-class access routers. This setting gives the network a bit of a peer-to-peer (P2P) character.

For flows active between N-class access routers, the longer it takes for flows to complete, the more likely the number of active flows will increase. There is a bit of reinforcement at work here. The more active flows that transit a given access router, the lower will be the throughput of each flow and the longer it will take for each flow to finish transferring its packets. Thus, the higher the arrival rate of flows (i.e., the lower the source think time) the larger the number of active flows. Two other factors have significant influence on the time taken to complete flows. The first factor is the average file size (x4). Larger files take longer to transfer because more packets must be relayed and acknowledged. The second factor is network speed (x2): a slower network (plus setting) will take longer to transfer files of any particular size. Fig. 4-18 reveals this complex collection of related and reinforcing influences on the number of active flows.

Some other plots (not reproduced here) from the ten-step analysis also reveal interactions between number and distribution of sources (x8/x9), file size and distribution of sources (x4/x9) and think time and distribution of sources (x5/x9). These interactions make sense given the discussion contained in the previous paragraph. The effects from these factor interactions are much less significant than the main factors alone. In fact, the analysis of all 22 responses reveals that MesoNet simulations are driven by main factors and not by interactions among factors.

Congestion at N-class access routers could certainly lead to packet losses, which would stimulate retransmissions and cause flows to take longer to complete because the required number of packet transmissions would increase. Given this reasoning, one would expect many of the same factors influencing the number of active flows to also influence the retransmission rate. Fig. 4-19 displays the main-effects plot for network-wide retransmission rate (y10). Comparing Fig. 4-18 and Fig. 4-19 one can certainly see significant overlap in the main factors: number (x8) and distribution of sources (x9), average file size (x4) and think time (x5) and network speed (x2). In fact, the same settings for these factors that lead to increased number of active flows also lead to increased retransmission rate. The main difference is that retransmission rate is influenced most significantly (and equally) by network speed (x2) and buffer sizing algorithm (x3). Fewer buffers (minus setting) and lower network speed (plus setting) lead to increased probability of packet losses, which stimulate retransmissions.

The fact that buffer size was not so important with respect to the number of active flows reflects TCP congestion control. Given a larger number of flows, the TCP congestion control mechanism reacts to losses by adjusting flow sending rate: slowing packet transmissions, which leads to lower throughputs but also mitigates packet losses. Fig. 4-19 shows that mitigating packet losses becomes more difficult when buffer sizes are severely restricted. One would expect the main effects influencing retransmission rate to be identical to the main effects influencing loss rate (y5). Our review of the main-effects plot (not reproduced here) for loss rate confirms this expectation.

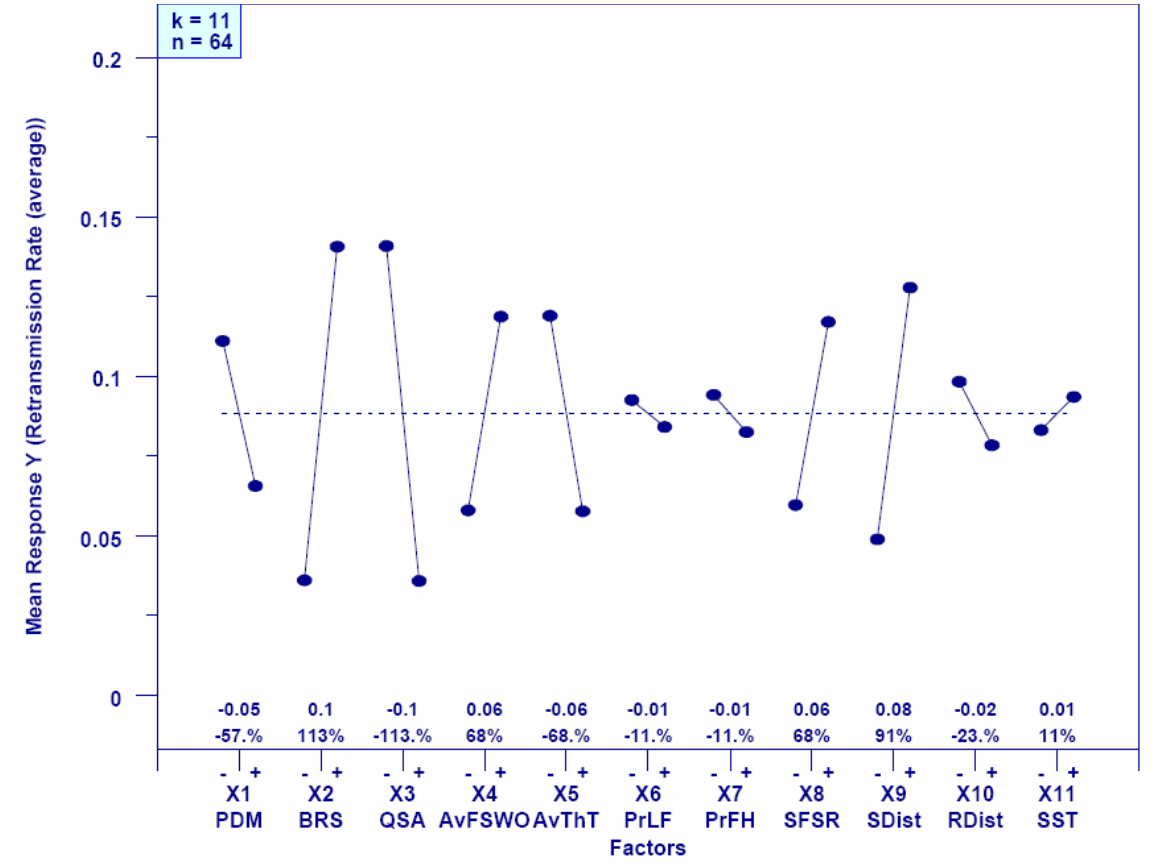


Figure 4-19. Main-Effects Plot for Response y10 (Average Retransmission Rate)

Given the analysis related to number of active flows and retransmission rate, one would expect throughput[8] on NN flows to be driven primarily by a relationship between available bandwidth (network speed) and number of active flows. The main-effects plot, Fig. 4-20, supports this expectation. Factors leading to fewer active flows include a lower number of sources (x8 minus) and a distribution that leads to fewer NN flows (x9 minus), as well as longer think time (x5 plus). Setting x9 to minus increases the probability that sources under N-class access routers will exchange data with receivers under F-class access routers, which gives the network a bit of a Web-centric character. With fewer active NN flows and higher network speed (x2 minus), the throughput achieved by NN flows is higher; under reverse conditions the throughput is lower.

Fig. 4-20 also reveals some subtle, although less significant, effects. Shorter propagation delay (x1 minus) yields higher throughput. This occurs because sources receive feedback more quickly and timeout values remain lower. Perhaps unexpectedly, throughput is higher when file sizes are smaller (x4 minus). This appears related to reducing the number of active flows, as flows complete more quickly when fewer packets must be transferred. Finally, larger buffer sizes (x3 plus) lead to higher throughput. This appears due to experiencing fewer losses, which requires fewer retransmissions and timeouts.

[8] Note that, though we use the term *throughput* when discussing flow classes, what we actually measure is often referred to as *goodput*; thus, retransmissions are not considered to be throughput.

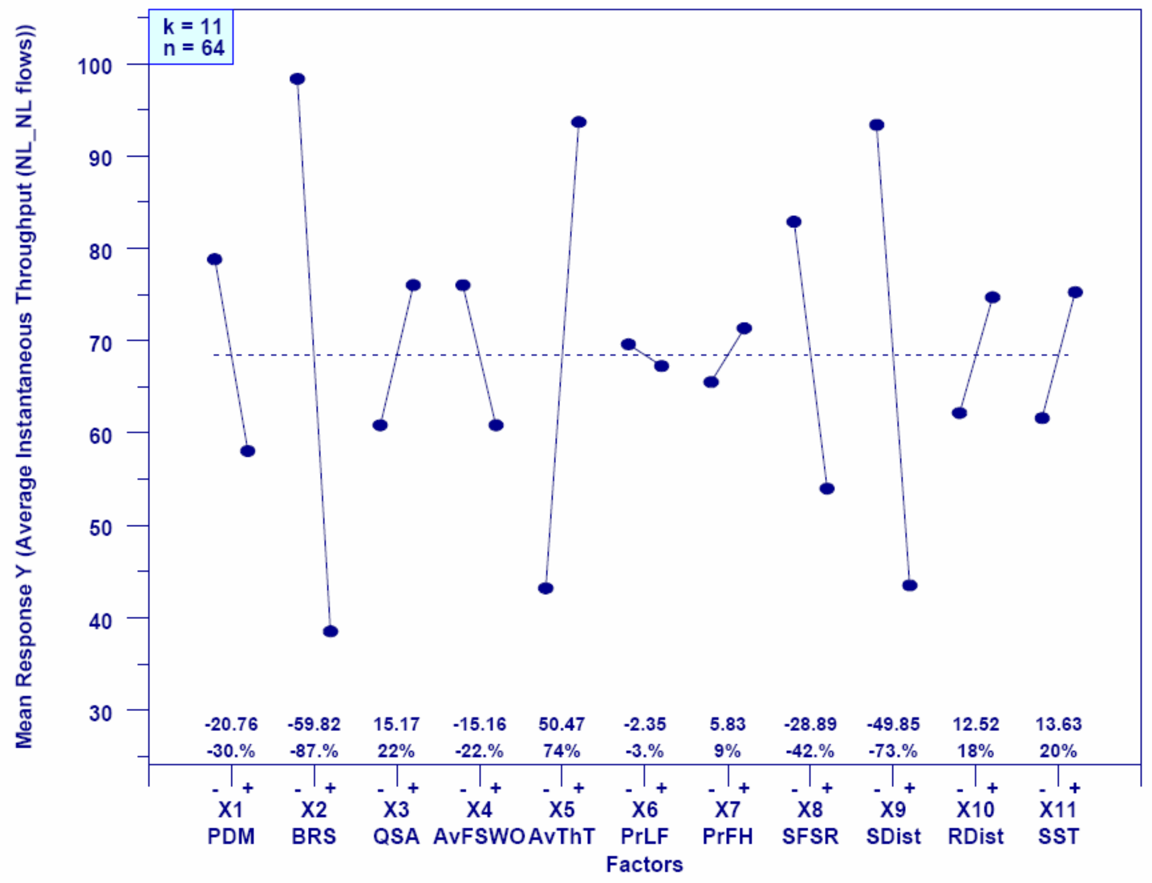


Figure 4-20. Main-Effects Plot for Response y22 (Average Instantaneous Throughput for NN Flows)

4.6.1.2 Delay-Related Responses. We selected average, smoothed, round-trip time (SRTT) as the response (y15) reflecting changes in network delay. Fig. 4-21 gives the related main-effects plot, which reveals that buffer-sizing algorithm and propagation delay are the main factors influencing SRTT. This makes eminent sense: higher propagation delay (x1 plus) and larger buffer sizes (x3 plus) lead directly to increase in SRTT. Larger buffer sizes permit bigger queues of packets, which increases queuing delay. Fig. 4-21 also reveals some minor effects, which suggest that congestion influences SRTT. This makes sense: more congestion leads to more packets residing in the bigger buffers.

4.6.1.3 Responses Related to Macroscopic Throughput. To represent the macroscopic throughput of the network, we selected two responses: data packets output per interval (y4) and flows completed per interval (y6). The first response represents the rate at which packets are flowing through the network, while the second response represents the rate at which flows are being completed by the network. We begin by considering the rate of packet output.

Fig. 4-22 reveals that the main influence on the rate of packet output is network speed: higher network speed (x2 minus) means a greater rate of packet output. This stands to reason in a network with a sufficient number of active flows. The combination of shorter think times (x5 minus) and more sources (x8 plus) leads to an increase in the number of flows and the higher network speed implies that each flow can transmit faster, so the aggregate rate of packet output should be greater under these circumstances. File size is another factor significantly affecting the rate of packet output. Larger file sizes (x4 plus) lead to greater throughputs because a smaller portion of the transfer occurs during slow-start, the transfer phase during which a flow's congestion window is lowest. Flows transferring with a larger congestion window achieve higher throughput, which helps to increase the aggregate network throughput.

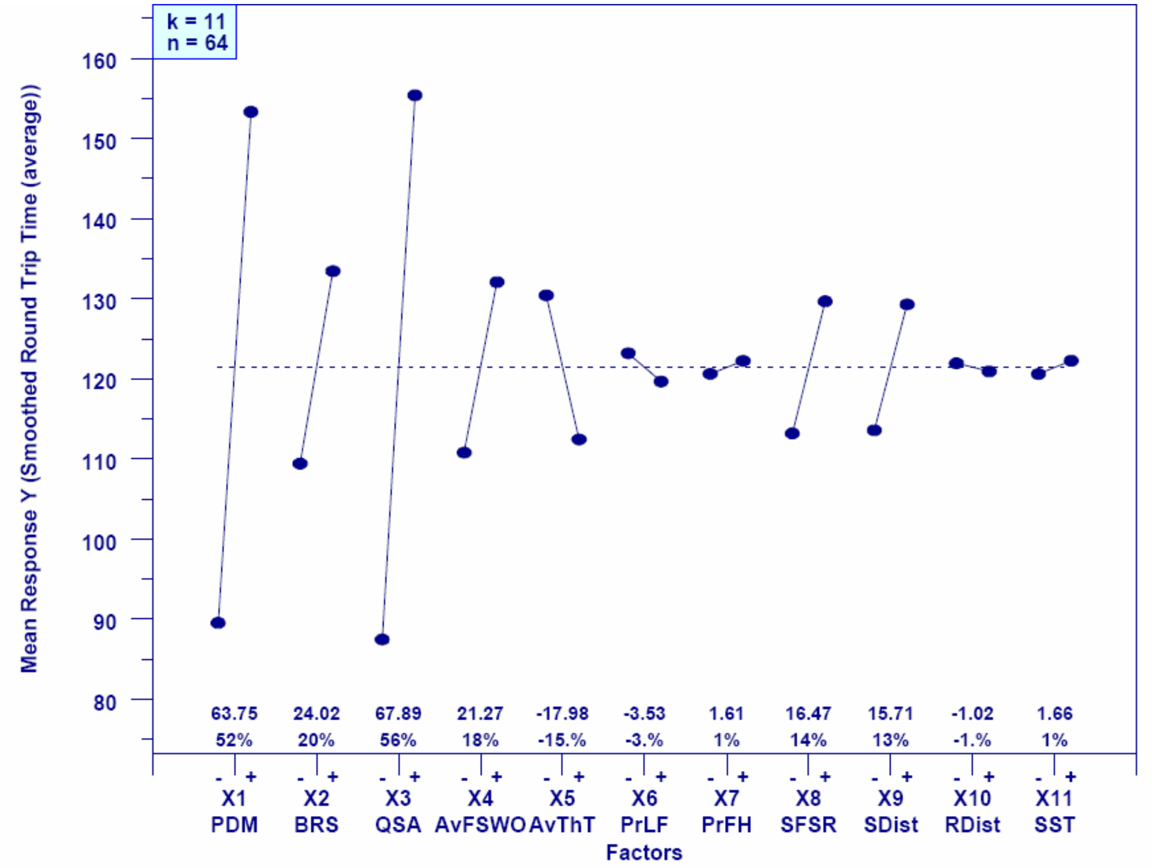


Figure 4-21. Main-Effects Plot for Response y15 (Average Smoothed Round-Trip Time)

As shown in Fig. 4-23, with one major exception, the story regarding the rate of flow completions is quite similar to the story regarding the rate of packet outputs. A sufficient number of connections (x5 minus and x8 plus) combined with higher network speed (x2 minus) contributes to a higher rate of flow completion. The exception involves file size (x4). In the case of packets output, larger file sizes (x4 plus) led to higher throughputs and thus to more packets output. On the contrary, for flows completed, smaller file size led to a higher completion rate. This stands to reason; smaller flows will be completed sooner. The sooner flows can be completed, the more flows can be completed per unit of time.

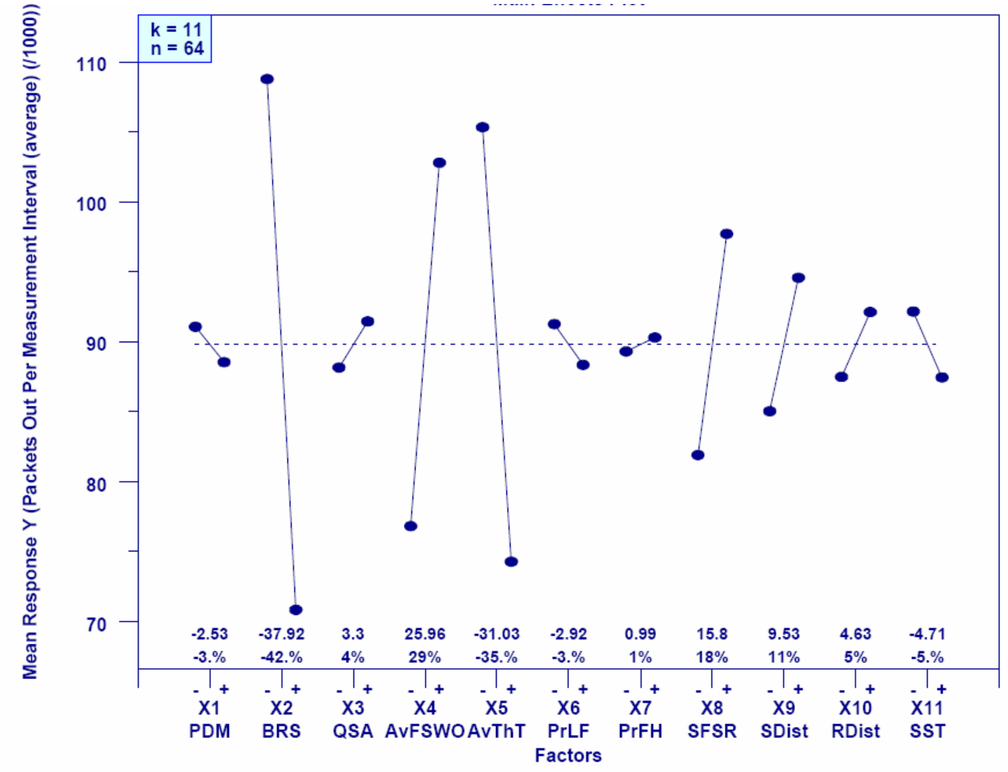


Figure 4-22. Main-Effects Plot for Response y4 (Average Packets Output per Measurement Interval)

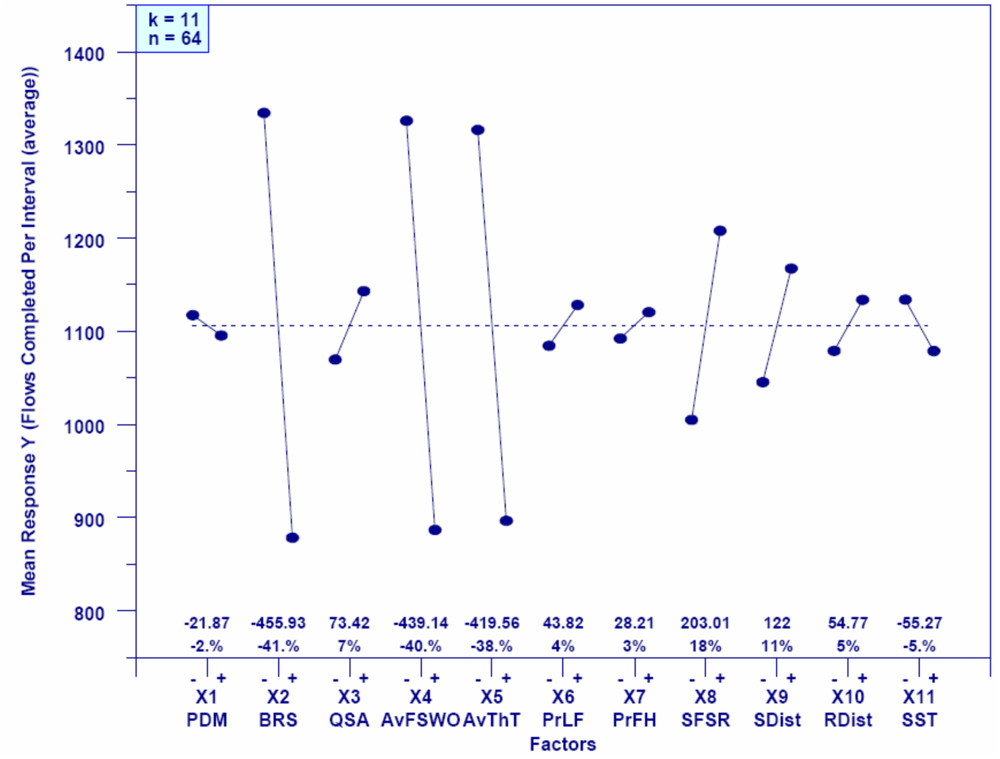


Figure 4-23. Main-Effects Plot for Response y6 (Flows Completed per Measurement Interval)

4.6.1.4 Responses Related to Advantaged Flow Classes. The final two responses we investigate represent throughputs achieved over advantaged flow classes, which are flows that transit between sources and receivers located under directly-connected and fast access routers. We examine the average instantaneous throughput of DD flows (y17) and FF flows (y20).

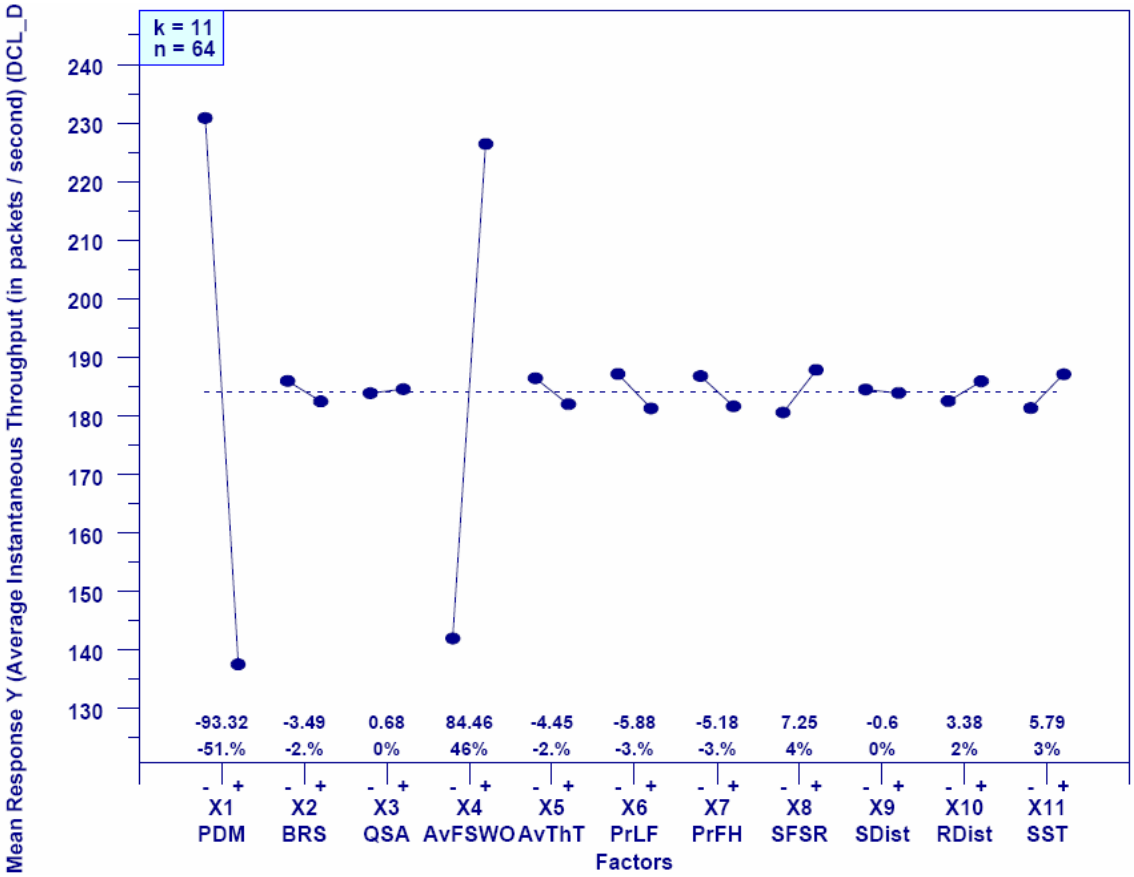


Figure 4-24. Main-Effects Plot for Response y17 (Average Instantaneous Throughput of DD Flows)

Each DD flow transits across a pair of D-class access routers, which are directly connected to backbone routers. D-class access routers are comparable in speed to POP routers, which are 10 times faster than N-class access routers. Given these factors, DD flows are the most advantaged in the simulation and should be able to achieve highest throughputs under the traffic scenario adopted for the sensitivity analysis. Few factors should impede the throughput of DD flows. Fig. 4-24 reveals that the throughput of DD flows is influenced by only two factors: propagation delay (x1) and file size (x4). This makes sense. Shorter propagation delay (x1 minus) permits faster feedback on DD flows, which allows the congestion window to increase more quickly. The rate of feedback is most important during the initial slow-start phase, where the congestion window starts at a small size but doubles with each acknowledgment received. The influence of file size is also clear. Larger file sizes (x4 plus) allow more of the packets in a file to be transferred after the flow reaches its peak sending rate. Smaller file sizes (x4 minus) imply that more of the packets in a file will be sent early in the slow-start phase, when a flow is building up toward its peak sending rate. Throughput early in slow-start will be much smaller than throughput after a flow reaches its peak rate.

Two other classes of advantaged flows are those where a source or receiver is under an F-class access router and its correspondent is under either an F-class or D-class access router. These flows comprise the following classes: DF flows and FF flows. The throughput achievable on these flows is constrained by the F-class access routers, which operate at twice the speed of N-class access routers. DF and FF flows are less advantaged than DD flows. We use the throughput on FF flows (response y20) to represent both classes.

Fig. 4-25 shows the main effects influencing throughput on FF flows. FF flows are influenced by a more complex mix of factors than DD flows. The significance of propagation delay (x1) and file size (x4) are two clear common factors among all advantaged flow classes. Shorter propagation delay means quicker feedback, which leads to faster increase in the congestion window for flows that are not impeded by congestion. Larger file sizes allow more of a flow's packets to be transferred at a higher sending rate. Less advantaged (DN, FN and NN) flows are influenced mainly by congestion, so propagation delay has less effect on those flows.

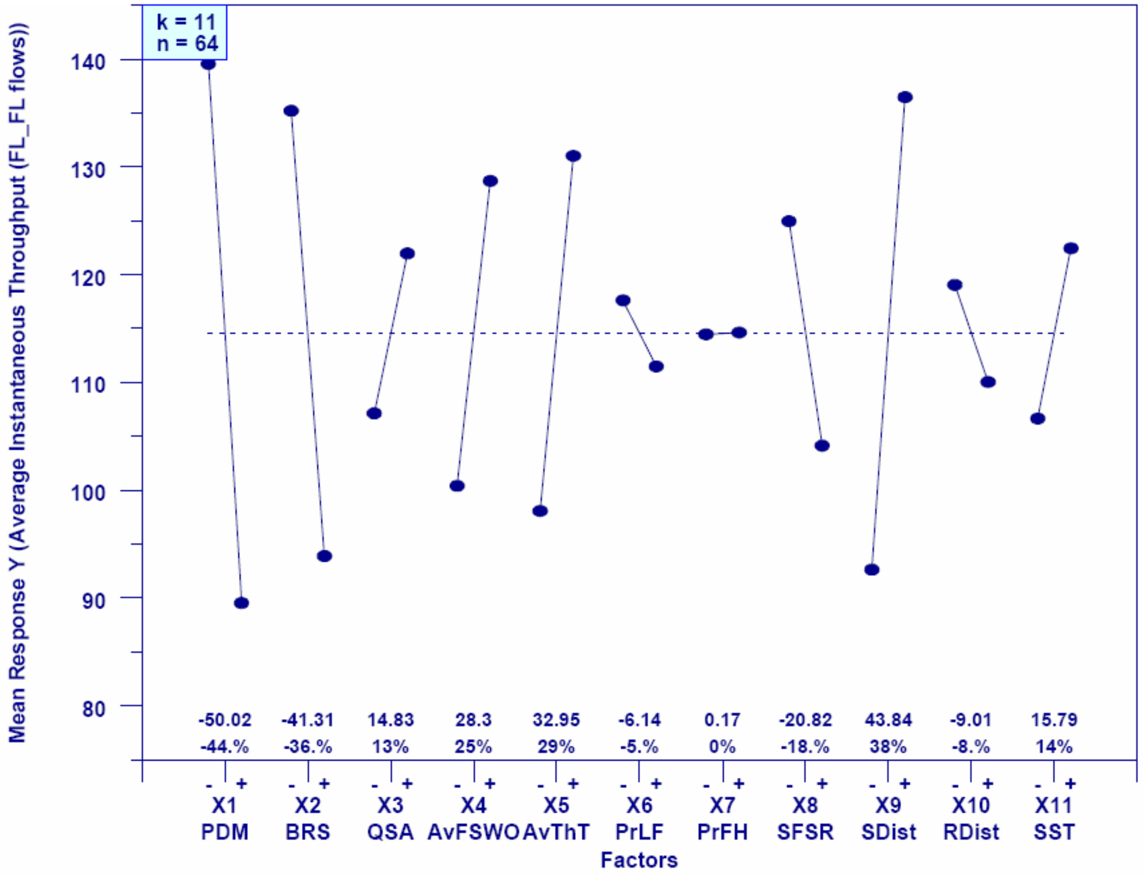


Figure 4-25. Main-Effects Plot for Response y20 (Average Instantaneous Throughput of FF Flows)

Unlike DD flows, FF flows can face some congestion because selected source distributions lead to higher numbers of FN flows. Specifically, a source distribution (x9 minus) that gives the network a Web-centric characteristic leads to more FN flows, which compete for throughput with FF and DF flows. In addition, lower average think time (x5 minus) leads to more active flows that can compete for throughput. Under these circumstances, higher network speed (x2 minus) allows competing flows to achieve higher throughputs. The influence of all these factors is evident in Fig. 4-25.

Our investigation of throughput reveals three general categories of flows. Throughput in one category, which includes the most numerous (DN, FN and NN) flows, is influenced mainly by congestion and network speed. Throughput in a second category, which includes only the most advantaged and least numerous DD flows, is influenced mainly by propagation delay and file size. Throughput in the remaining category (DF and FF flows) is influenced by a combination of the factors influencing the other two categories.

4.6.2 Sensitivity Analysis Guided by Principal Components Analysis

In this section we examine the sensitivity of the principal components (PCs) to variations in model inputs. Recall from our PCA (Sec. 4.5) that we identified four main principal components accounting for most variation in the model's 22 responses. We viewed these PCs, summarized in Table 4-24, as groupings of responses representing different aspects of the model's behavior. The reader may note a correspondence between the groupings by principal component and the groupings used (in Sec. 4.6.1) to describe the sensitivity analysis of responses guided by correlation analysis. Below, we report the results of applying main-effects analysis to the PCs identified in Table 4-24.

Table 4-24. Definition of Major Principal Components in Model Response

Principal Component	Responses in Principal Component
Congestion (PC1)	y1, y2, y5, y7, y10, y11, y12, y13, y14, y19, y21, y22
Delay (PC2)	y15, y16
Throughput for Advantaged Flows (PC3)	y17, y18, y20
Macroscopic Throughput (PC4)	y3, y4, y6

4.6.2.1 Congestion. Given that PC1 represents the effects of network congestion, one would expect significant congruence between factors affecting PC1 and factors affecting responses driven by congestion. Previously, we analyzed three congestion-related responses: number of active flows (y1), retransmission rate (y10) and average instantaneous throughput for NN flows (y22). We also noted that loss rate (y5) and retransmission rate were related closely. Analysis of PC1 should show that the same factors influence PC1 as influence responses y1, y5, y10 and y22. Fig. 4-26 displays the main-effects plot for PC1. The key factors influencing PC1, in order of significance, include: network speed (x2), think time (x5), distribution (x9) and number (x8) of sources, file size (x4) and buffer size (x3). This set of factors is also the union of factors most significantly driving responses y1, y10 and y22. Further, given insights from our previous analyses, we conclude that Fig. 4-26 illustrates the following factors induce network congestion and its consequences: slower network speed (x2 plus), smaller buffer sizes (x3 minus), larger file size (x4 plus), shorter think time (x5 minus), more sources (x8 plus) and a P2P-like distribution of sources (x9 plus). Reversing these factors eases network congestion and its consequences.

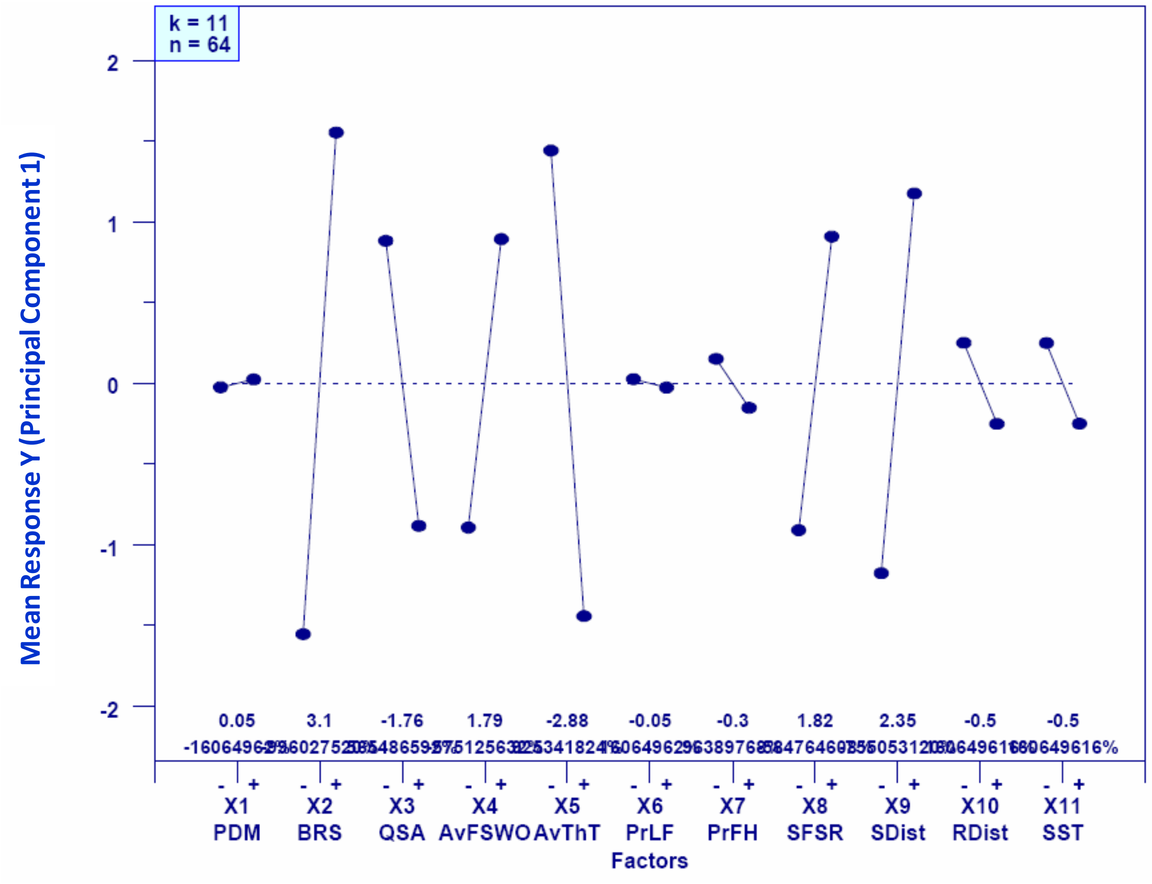


Figure 4-26. Main-Effects Plot for PC1 (Network Congestion) – the % computations in Main-Effects plots for PCs appear unreadable (and meaningless) because PC values are normalized to a 0 mean. The reader may simply use the magnitude and slope of the lines connecting the – and + level of each of the 11 factors to provide information that may be compared with previous Main-Effects plots applied to individual responses rather than PCs, which are linear combinations of many individual responses

The reasons that these factors modulate network congestion have already been explained in Sec. 4.6.1.1. Our analysis suggests that factors modulating network congestion will influence all responses grouped under PC1. For example, increasing network congestion lowers the average congestion window (y11), decreases the rate of congestion window increases (y12), increases the rate of negative acknowledgments (y13) and timeouts (y14) and reduces average throughput for DN (y19), FN (y21) and NN (y22) flows. For the traffic scenario adopted, the macroscopic pattern of network congestion relates primarily to the most numerous types of flows, which transit the most numerous N-class access routers. Other factors influence less numerous, more advantaged flows, as discussed below when considering PC3.

4.6.2.2 Delay. Given that PC2 represents effects on network delay, one would expect significant congruence between factors affecting PC2 and factors affecting responses driven by delay. Previously, we analyzed one delay-related response: average smoothed round-trip time – SRTT (y15). SRTT was driven primarily by two factors: buffer size (x3) and propagation delay (x1). Note that SRTT is significantly correlated (0.70) with relative queuing delay (y16), which is driven mainly by one factor: buffer size (x3). One would expect PC2 to be driven by the same factors that drive SRTT and relative queuing delay. Fig. 4-27 depicts the main-effects plot for PC2. The plot shows that PC2 is mainly influenced by two factors: buffer size and propagation delay. Fig. 4-27 also reveals a minor influence of think time (x5). Interpreting Fig. 4-27 shows that average network delay increases with increases in propagation delay (x1 plus) and buffer size (x2 plus). Further, Fig. 4-27 suggests that decreasing think time (x5 minus) tends to increase delay; this is likely due to the fact that more flows are active simultaneously, which helps to fill the larger available buffer space.

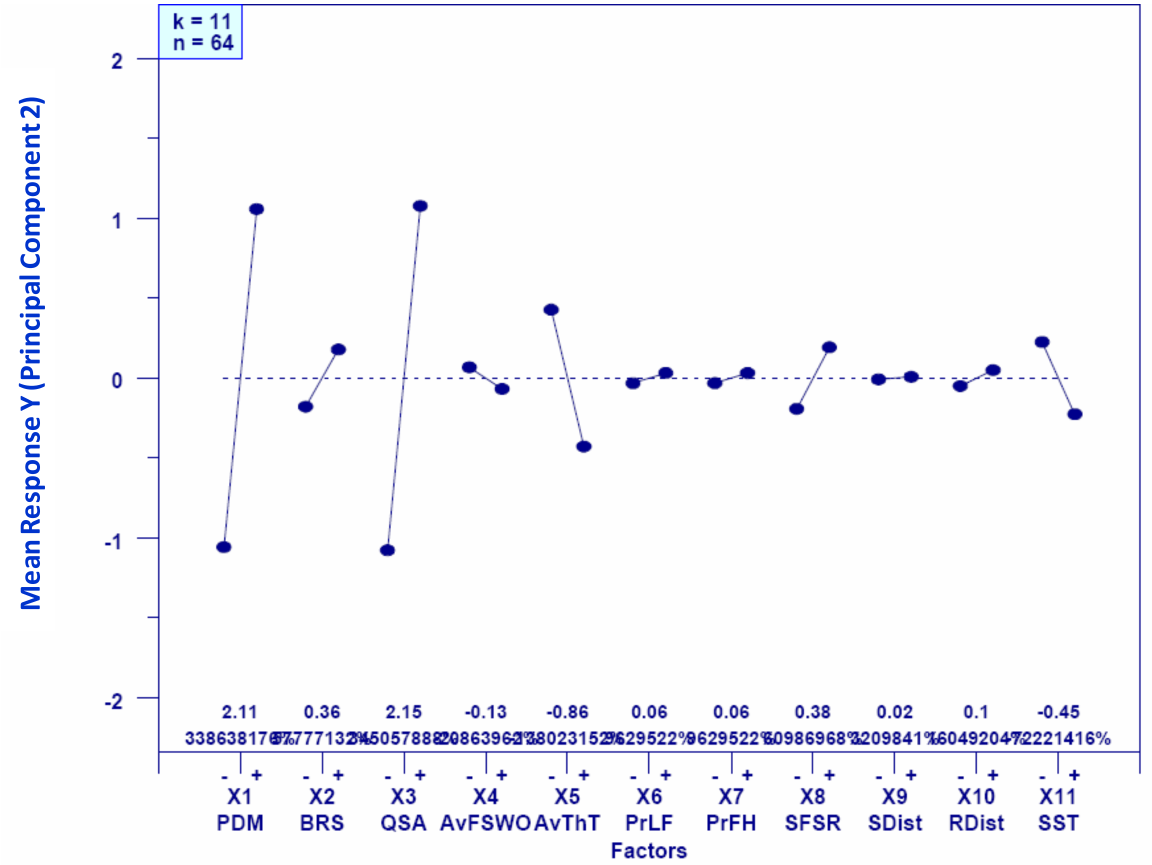


Figure 4-27. Main-Effects Plot for PC2 (Network Delay)

These results indicate that, for the traffic scenario used here, network delay is largely orthogonal to network congestion. Why might this be so? Under congestion, the TCP congestion control mechanism causes flows to reduce their sending rate. This adapts the flow of packets into the network in accordance with perceived congestion. The feedback rate for the congestion control mechanism depends largely on network delay, which is due to two factors: propagation delay and queuing delay. Propagation delay is modulated by the distance packets must travel, and queuing delay is modulated by the size of buffers in network routers. Congestion cannot affect the distance that packets must travel. When buffers are small, congestion cannot affect the queuing delay because queues will be small. Only when buffers are large can the degree of congestion influence network delay, but in this case TCP congestion control reacts to reduce the rate of traffic entering the network, which tends to limit the number of packets in the network. For these reasons, results suggesting lack of correlation between network congestion and delay appear reasonable.

4.6.2.3 Throughput for Advantaged Flows. Given that PC3 represents the effects on throughput for advantaged flows, one would expect significant congruence between factors affecting PC3 and factors affecting throughput for DD (y17), DF (y18) and FF (y20) flows. Previously, we analyzed factors influencing throughput on DD and FF flows. Factors influencing throughput on DF flows (not included in this report) are identical to the factors influencing FF flow throughput. The main factors influencing throughput for advantaged flows include: propagation delay (x1), file size (x4), network speed (x2), distribution of sources (x9) and think time (x5). Review of Fig. 4-28 shows that the same factors influence PC3. PC3 is also driven to some extent by buffer size (x3), which was not a significant factor in the previous analysis of throughput for DD or FF flows.

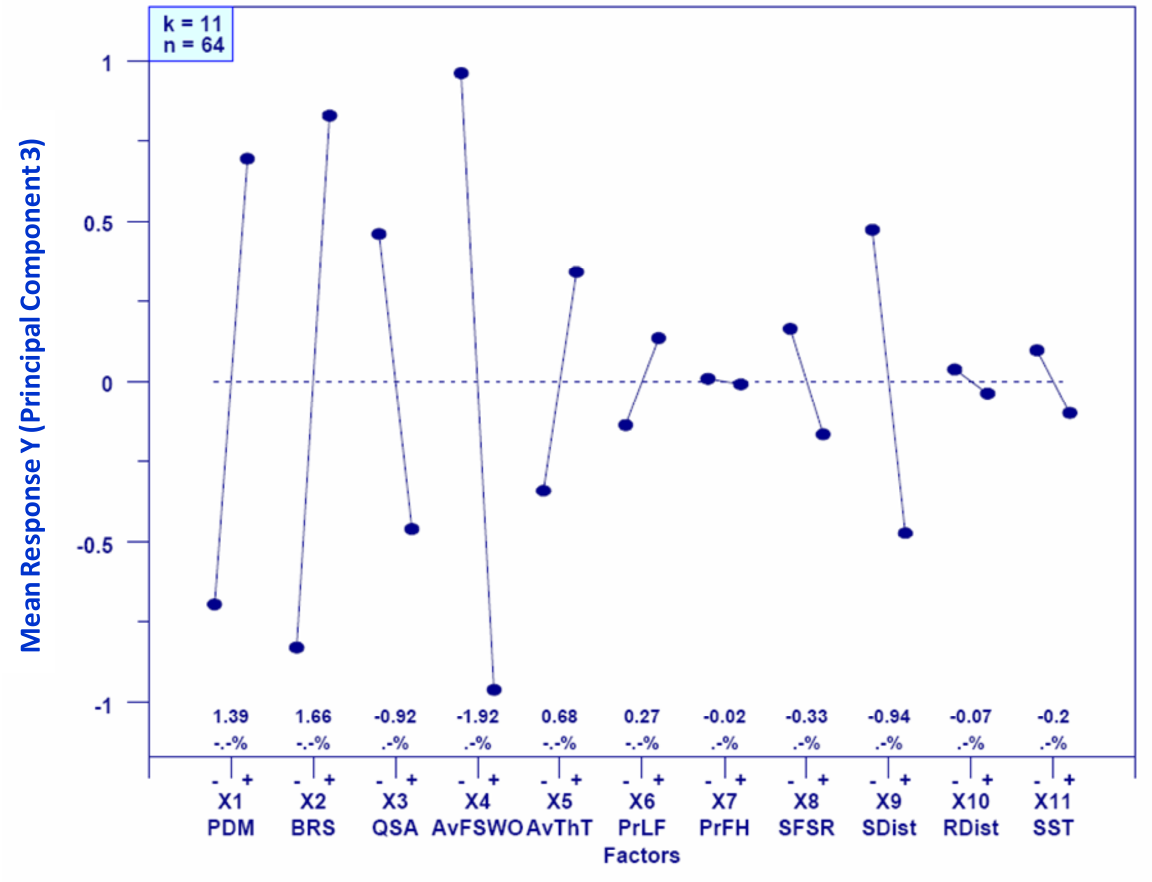


Figure 4-28. Main-Effects Plot for PC3 (Throughput for Advantaged Flows)

One could interpret Fig. 4-28 as depicting lower throughput above the zero line and higher throughput below the zero line. Using this interpretation, Fig. 4-28 indicates that higher throughput on advantaged flows results from: larger file sizes (x4 plus), higher network speed (x2 minus), shorter propagation delay (x1 minus), and more P2P-like network traffic (x9 plus). Reversing the settings for these factors would lead to lower throughputs. These findings are consistent with our previous analysis of the factors influencing throughput for DD (y17) and FF (y20) flows.

One somewhat new piece of information is revealed by Fig. 4-28: the influence of buffer size on throughput for advantaged flows. In our previous analyses, buffer size had a more modest influence on throughput. The influence that was present indicated that smaller buffers led to lower throughputs and larger buffers led to higher throughputs. This seems to make sense because small buffers lead to increased losses, which lead to increased retransmissions, which lead to longer file transfer times, which results in lower throughputs. Fig. 4-28 shows the same influence, so the interpretation of PC3 remains consistent with the earlier results for DD and FF flows.

4.6.2.4 Macroscopic Throughput. Given that PC4 represents effects on macroscopic (network-wide) throughput, one would expect significant congruence between factors affecting PC4 and factors influencing the rate of data packets leaving the network (y4) and the flow-completion rate (y6). Recall, though, that one factor, file size (x4), had the opposite influence on y6 and y4. With this information, we should be able to determine which aspect of macroscopic throughput is represented by PC4. Sec. 4.5 suggested that PC4 represents macroscopic throughput of flow completions.

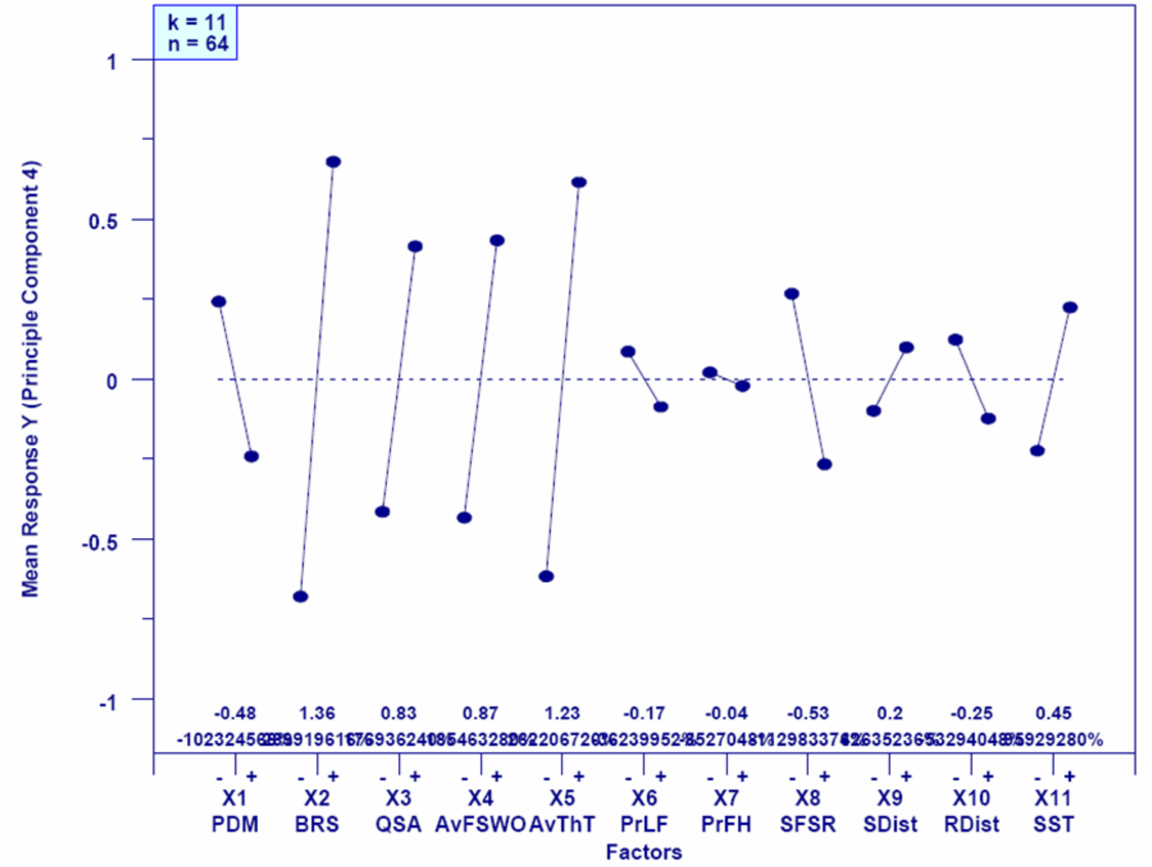


Figure 4-29. Main-Effects Plot for PC4 (Macroscopic Throughput)

Fig. 4-29 displays the main-effects plot for PC4. The primary factors influencing PC4, in order of significance, include: network speed (x2), think time (x5) and file size (x4). Interpreting Fig. 4-29 suggests that higher network speed (x2 minus) and shorter think time (x5 minus) increase macroscopic throughput.[9] These findings are consistent with the factors influencing both the rate of packet output (y4) and the rate of flow completion (y6). Fig. 4-29 shows that smaller file size (x4 minus) causes variation in the same direction as higher network speed and shorter think time. From this, we conclude that PC4 represents macroscopic throughput of flow completions. This agrees with the previous PCA.

4.6.3 Summary of Findings from the Sensitivity Analysis

Results from the sensitivity, correlation and principal components analyses increased our confidence in MesoNet. The behavioral patterns and relationships revealed by the sensitivity analysis aligned with our expectations. Further, the sensitivity analysis provided significant insight into model operation. We review our key findings here. We begin with a summary of the main aspects of model behavior. Second, we characterize the major factors influencing model behavior. Third, we identify and discuss factors that appear to have little influence on model behavior.

4.6.3.1 Main Aspects of Model Behavior. The sensitivity analysis revealed the model to have six main behavioral aspects: congestion, delay, throughout for DD flows, throughput for DF and FF flows, packet throughput and flow completion throughput. We discuss each of these in turn.

Congestion. The largest behavioral aspect of the model relates to macroscopic congestion, which occurs primarily in the slowest (N-class) access routers. In the topology employed for the sensitivity analysis, most users (represented as model sources) accessed the network through the (105) N-class routers. This is analogous to business and home users who connect to a network via limited bandwidth links. Higher network tiers (represented in the model by 22 POP and 11 backbone routers) typically operate at speeds sufficient to support traffic entering the network from the access tier. The topology used in the sensitivity analysis reflects this fact of network design. The model's heterogeneous topology allowed selected (D-class and F-class) access routers to operate at higher speeds. (Twenty-eight) F-class access routers represented larger businesses that might support Web sites, which could be accessed by many users, most of whom connect to the network through N-class routers. (Six) D-class access routers represented research institutions and very large corporations that connect directly to the network backbone.

The net result of this topology is that most active flows transit N-class access routers because most users reside underneath such routers. These flows include NN, FN and DN flows. Since these flows are most numerous, their behavior tends to drive macroscopic congestion, which occurs at the network edge (i.e., in the access tier). Of course, this is also due in part to the homogeneous Web-like traffic model employed during the sensitivity analysis. Regardless of traffic model, one should expect network congestion to arise primarily at the access tier because transit networks are continuously

[9] Note that PCA involves normalizing and transforming responses to a scale differing from the scales of the original responses. This means that interpretation of the main-effects plots for principal components must be aided by context provided from previous interpretation of main-effects influencing particular responses.

monitored by network providers and the bandwidth in the POP and backbone tiers is provisioned to meet traffic demands from the access tier.

Throughput available to individual NN, FN and DN flows is constrained by the bandwidth of N-class routers. This means that lower network speed (x2 plus) will reduce flow throughputs, while higher speed (x2 minus) will increase flow throughputs. Further, the more flows (y1) that transit an N-class router, the lower will be the individual throughputs for each flow. This behavior is revealed by the related response variables: y19, y21 and y22. The number of active flows transiting an access router is influenced primarily by three factors: number (x8) and distribution (x9) of sources and average idle time (x5) between source transfers. Increasing the number of sources leads to increased congestion within a fixed topology of N-class access routers. In the sensitivity analysis, the FN to NN ratio (in number of flows) shifts depending on the distribution of sources. With the plus setting for x9 the pattern of flows takes on a P2P-like characteristic, where the FN to NN ratio decreases. With the minus setting the flow pattern adopts a Web-centric characteristic, where the FN to NN ratio increases. Since NN flows take slightly longer to complete than FN flows, the P2P pattern tends to result in more flows transiting N-class routers at any given instant. And, of course, the shorter the idle time between transfers the more sources will arrive in any given period.

Network congestion also influences macroscopic responses, including: loss (y5) and retransmission (y10) rates, congestion window (y11) and its rate of increase (y12), and rate of negative acknowledgments (y13) and timeouts (y14). As with flow throughputs, these responses can be primarily attributed to the relative number of flows simultaneously transiting N-class access routers, as well as to the speed of N-class routers. The existence of fewer simultaneous flows, combined with higher speed, creates a better experience for individual flows and less congestion at the network edge.

To summarize, congestion occurs at the network edge. The primary effects of congestion are due to flows transiting N-class access routers. Higher speed N-class routers mitigate congestion to some extent. The macroscopic effects of congestion are due to the most numerous flow types: NN, FN and DN flows.

Delay. Network delay, measured as smoothed round-trip time (SRTT), is influenced by two main factors: propagation delay (x1) and buffer-sizing algorithm (x3). As one would expect, longer propagation delay and larger buffer sizes lead to increased SRTT (y15). Further, relative queuing delay (y16) is driven only by buffer size. These relationships are as expected. Perhaps unexpected is that network delay is largely uncorrelated with congestion. We attribute this to the fact that the TCP congestion control mechanism responds to network congestion by slowing the rate of packets injected into the network and thus limiting the number of packets that might otherwise be sitting in network buffers.

Throughput for DD Flows. For DD flows, both the source and receiver reside under D-class access routers, which connect directly to backbone routers and operate at the same speed as POP routers. Further, the number of simultaneously active DD flows is typically quite small, relative to other flow classes. Given these facts, DD flows should be able to achieve throughputs constrained only by the minimum of the speeds of the source and receiver. The sensitivity analysis revealed that, unique among flow classes, throughput of DD flows is influenced by only two factors: propagation delay (x1) and file size (x4).

A DD flow must transit through TCP slow-start before reaching its maximum achievable throughput. The time taken to reach maximum throughput then depends upon the feedback rate on the flow. The feedback rate is determined mainly by propagation delay. Longer propagation delay lengthens time taken to achieve maximum throughput. Further, for larger files, more packets may be transferred at maximum throughput, so average throughput is higher. For the traffic patterns used in the sensitivity analysis, no other factors influenced throughput of DD flows. For this reason, DD flows must be given separate consideration from other flow classes.

Throughput for DF and FF Flows. DF and FF flows could potentially achieve maximum throughputs constrained only by the minimum of the speeds of the source and receiver, but some other factors can interfere. For example, DF and FF flows compete for throughput with FN flows, which might be smaller or larger in number, depending upon various factors discussed previously. When the relative number of FN flows is smaller, then DF and FF throughputs are influenced mainly by propagation delay and file size, as is the case for DD flows. When the relative number of FN flows is larger, then DF and FF throughputs are influenced more by the factors that influence throughput of FN flows.

Packet Throughput. Packet throughput (y4) is influenced primarily by network speed (x2), idle time of sources (x5) and file size (x4). When network speed is faster (x2 minus), flow congestion windows are larger, and so flows can send more packets per unit of time. When idle time is smaller (x5 minus), more flows tend to be active, which means more flows are injecting packets into the network. Finally, when file sizes are larger (x4 plus), then more flows are operating at their maximum achievable throughputs, so more packets are being injected into the network. The higher the network speed and the more packets being injected into the network, the greater the number of packets leaving the network. These factors determine packet throughput. Of course, when there are many active flows and lower network speed, then congestion increases and the TCP congestion control mechanism slows the rate of packets entering the network, which also slows the rate of packets exiting the network.

Flow-Completion Throughput. Flow-completion throughput (y6) is also influenced primarily by network speed, idle time of sources and file size. In this case, however, smaller file sizes (x4 minus) lead to shorter file-transfer times, which increases the number of flows completed in a given time period. Of course, since each file transfer spends a lower proportion of its duration at maximum achievable throughput, the number of packets injected will be lower than if the file size is smaller. Thus, to some extent, variations in file size lead to a tradeoff between packet throughput and flow-completion throughput.

4.6.3.2 Major Factors Influencing Model Behavior. Based on the results of the sensitivity analysis, we can identify the major factors influencing the behavior of MesoNet. We consider the results of the analysis by a domain expert and also the PCA. We begin with the results from a domain expert, which are based on six main behavioral characteristics, as identified in the preceding section.

We use one response to represent each characteristic: packet throughput (y4), flow-completion throughput (y6), congestion (y10), delay (y15) and throughput of DD (y17) and FF (y20) flows. Table 4-25 shows the result of a rank analysis, where the relative influence of each factor on each of the six responses is assigned a rank from one (most influential) to 11 (least influential) based upon the degree to which the factor altered the response when moving from a plus to a minus setting. The average rank is computed for each factor, and then the average rank is converted into an integer ranking based on ordering the factors from most (one) to least (11) influential. The table shows that network speed (x2) is the most influential factor, followed by file size (x4) and think time (x5). Next is number of sources (x8), followed by propagation delay (x1) and distribution of sources (x9). Buffer-sizing algorithm (x3) ranks seventh. The remaining factors lag: initial slow-start threshold (x11), distribution of receivers (x10), probability that a flow transfers a larger document (x6) and then probability that a host is fast (x7).

Table 4-26 gives a similar analysis based on the top four principal components identified by the PCA. The PCA squeezes out some redundancy included in the analysis conducted by the domain expert. In particular, the domain expert separated throughput for advantaged flows into two responses (y17 and y20) based upon an observation that different factors drove the two responses. In addition, the domain expert noted that packet throughput (y4) was driven in two different directions, depending upon file size (x4); thus, decided to retain packet throughput as a separate response. The PCA amalgamated y17 and y20 (in PC3) and also combined y4 with y6 (in PC4).

Table 4-25. Rank Analysis based on Domain Expertise

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11
y4	9.5	1	8	3	2	9.5	11	4	5	6.5	6.5
y6	11	1	6	2	3	9	10	4	5	7.5	7.5
y10	7	1.5	1.5	5	5	10	10	5	3	8	10
y15	2	3	1	4	5	8	10	6	7	10	10
y17	1	8	10.5	2	8	5	5	3	10.5	8	5
y20	1	3	8	5	4	10	11	6	2	9	7
Average Rank	5.25	2.92	5.83	3.50	4.50	8.58	9.50	4.67	5.42	8.17	7.67
Ordinal Rank	5	1	7	2	3	10	11	4	6	9	8

Table 4-26. Rank Analysis based on Principal Components Analysis

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11
PC1	10.5	1	6	5	2	10.5	9	4	3	7.5	7.5
PC2	2	6	1	7	3	8.5	8.5	4	11	10	5
PC3	3	2	5	1	6	8	10.5	7	4	9	10.5
PC4	5	1	4	3	2	10	11	7	9	8	6
Average Rank	5.13	2.50	4.00	4.00	3.25	9.25	9.75	5.50	6.75	8.63	7.25
Ordinal Rank	5	1	4	4	2	10	11	6	7	9	8

Comparing Tables 4-25 and 4-26 reveals similarities (in the main) and a few differences. Both tables rank factors x6, x7, x10 and x11 as not very influential on system response. Both tables identify network speed (x2) as the main factor driving response and both tables also rank file size (x4) and think time (x5) as significant factors. Both tables also agree on the relative influence of propagation delay (x1). The PCA suggests that buffer size is fairly influential, while the domain expert finds buffer size to be less significant. When the redundancies (y17 and y4) are removed from Table 4-25, the significance of buffer size is comparable for both analyses. Overall, the analyses are quite consistent.

4.6.3.3 Factors Exhibiting Little Influence on Model Behavior. The sensitivity analysis, whether based on domain expertise or principal components, shows that system response is little influenced by four factors: probability of transferring a larger file (x6), probability that a source is on a fast host (x7), distribution of receivers (x9) and initial slow-start threshold. The experiment design varied the probability of transferring a (10x) larger file from 0.01 (x6 plus) to 0.02 (x6 minus). Apparently, the difference in these probabilities was small enough that the system response was not influenced. Results might have proven different if the higher probability were increased, but this would imply that a lower proportion of file transfers were reserved for simple Web browsing activities.

The probability that a source is on a fast host (x7) makes little difference in system response because most sources existed underneath N-class access routers, which had limited bandwidth to share among those sources. Apparently, due to multiplexing with other sources, most sources were seldom able to realize their maximum potential transmission rate.

The distribution of receivers (x10) had little influence on results because no matter the setting, most receivers resided under N-class access routers. The proportion of receivers under N-class access routers varied from 76 % (x10 plus) to 86 % (x10 minus). The distribution of receivers had more bearing on the number of receivers under D-class access routers (4 % and 2 % for x10 plus and minus, respectively) and also under F-class access routers (20 % and 12 % for x10 plus and minus, respectively).

The initial slow-start threshold (x11) had significant influence on only one response: average congestion window size (y11). Absent losses, the congestion window on a flow can become very high even though flow throughput will be limited by the minimum of the maximum speeds of the source and receiver. Setting a high initial slow-start threshold (x11 plus) allows flows to increase their congestion window very quickly to a large size. Setting a lower initial slow-start threshold (x11 minus) permits quick increase to a small size and then linear increase afterward. Thus, when congestion is light, using a large threshold for initial slow-start permitted the average congestion window size to become much larger, even though there was little influence on flow throughput.

4.7 Exploring Effects of Buffer Sizing

As we explained in Sec 4.1.6, the experiment design approach we used can facilitate additional exploratory analyses that were not necessarily planned at the time the experiment was undertaken. Here, we demonstrate this feature of our approach by using model response data to investigate the importance of buffer sizing relative to network speed and propagation delay. Our experiment design used two algorithms for buffer sizing. One algorithm, which is recommended practice [40], sets buffer size by multiplying average estimated round-trip time by capacity. The second algorithm, suggested by McKeown and colleagues [37], divides buffer size computed from the first algorithm by the square root of the expected number of flows. This second algorithm requires much less buffer space in network routers. McKeown and colleagues conducted an analytical study and empirical experiment that found similar performance when using either buffer sizing algorithm. The McKeown study, which was limited to a small number of flows transiting a few routers, suggested that network providers could deliver reasonable performance while requiring much less memory in routers. The study left for future work consideration of the effects of the alternate buffer-sizing algorithm in a network-wide context. Our sensitivity analysis was not intended explicitly to study detailed effects of buffer-sizing algorithms, but the experiment design used does provide information that could shed some light on the topic.

In this section, we use the results from our sensitivity experiment to explore the effects of buffer-sizing algorithm on overall network behavior and user experience. Our goal is to develop evidence relevant to the findings of the McKeown study. First, we consider the effects that choice of buffer-sizing algorithm has on smoothed round-trip time (SRTT) and on relative queuing delay. In previous discussions, we showed that buffer-sizing algorithm influenced both these aspects of network delay. Here, we look a bit more explicitly at the data. Second, we conduct a rank analysis based on selected responses chosen to characterize overall network behavior and user experience.

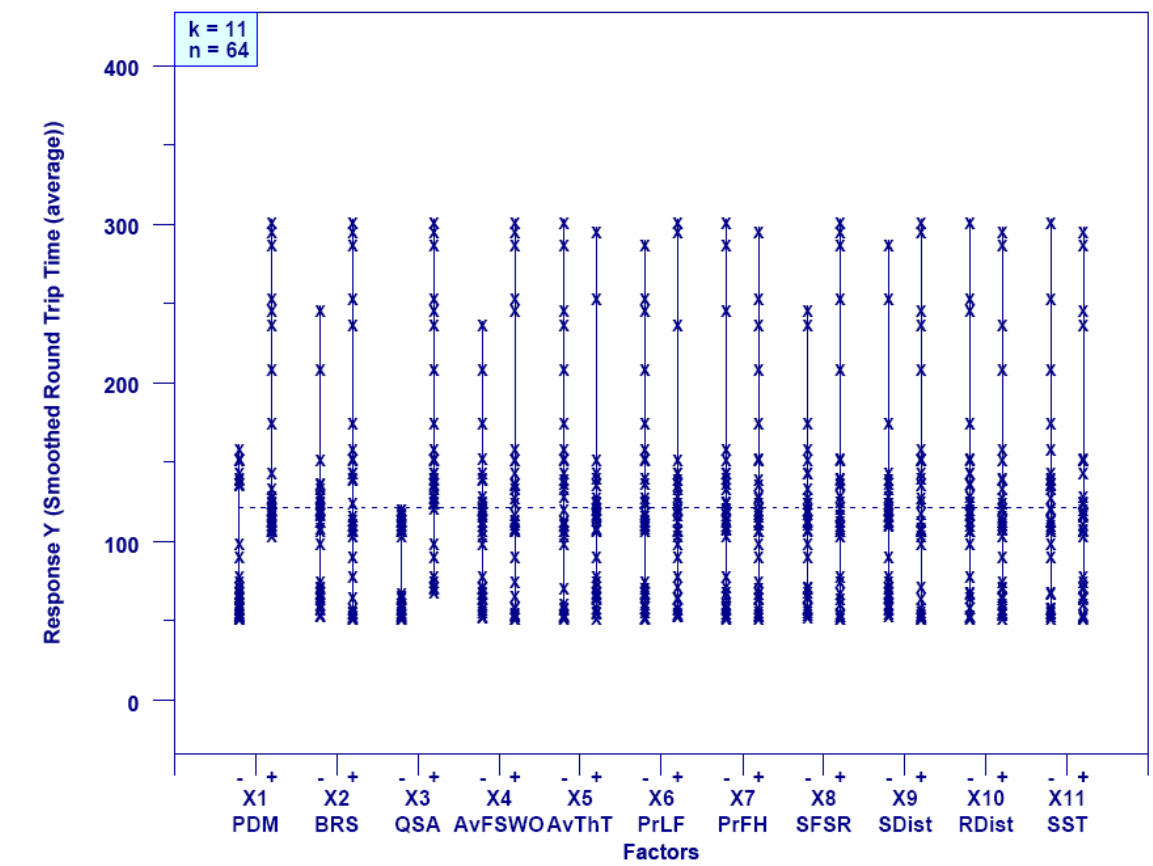


Figure 4-30. Multi-factor Scatter Plot of Smoothed Round Trip Time (y15) for each of the 11 Experiment Factors

4.7.1 Effects on Delay Variation. Fig. 4-30 presents a multi-factor scatter plot (explained in Appendix D.2) for SRTT. Fig. 4-31 shows a similar plot for relative queuing delay. Each plot depicts how the related response varies with the two settings of each of the 11 factors used in our experiments. For the current discussion we are interested in the buffer-sizing algorithm (factor x3). Fig. 4-30 shows that the choice of buffer-sizing algorithm shifts the pattern of SRTT. The McKeown algorithm restricts variation in SRTT. This occurs because buffer sizes are much smaller and thus queuing delays must also be smaller.

Fig. 4-31 illustrates clearly that reduced buffer size restricts the range of queuing delay that packets experience. Reducing variation in queuing delay within a network leads to more predictable SRTT and also to faster feedback regarding congestion. These traits might be considered valuable for selected networks and applications. On the other hand, one wonders whether more predictable delay might come at the cost of worsening behavior in other aspects of the network. The McKeown study suggested that smaller buffer size would not detract from user experience. We investigate this question next.

4.7.2 Effects on Other Aspects of Network Behavior. In this section, we consider the relative influence of propagation delay (x1), network speed (x2) and buffer-sizing algorithm (x3) on selected responses, chosen to represent macroscopic network behavior and user experience. To represent macroscopic behavior, we use packet throughput (y4), flow-completion throughput (y6), retransmission rate (y10) and relative queuing delay (y16). To represent user experience, we use average throughput from three different flow classes: DD flows (y17), FF flows (y20) and NN flows (Y22). We aim to determine which of the three factors (x1, x2 or x3) has largest influence on the combined responses.

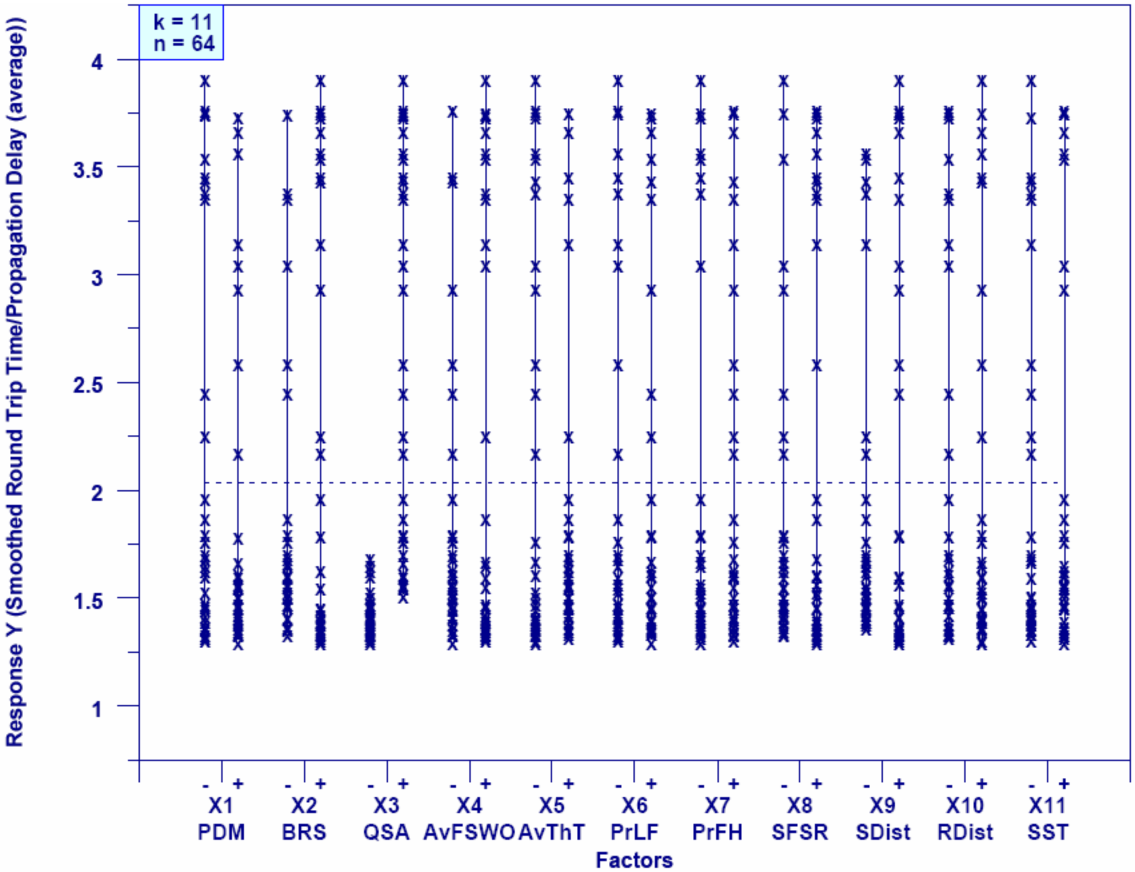


Figure 4-31. Multi-factor Scatter Plot of Relative Queuing Delay (y16) for Each Experiment Factor

We use a rank analysis to study the effects of our chosen factors on our selected responses. In this particular analysis, we elected to use a larger number to indicate higher rank and a smaller number to indicate lower rank. We begin by combining our three factors into a condition that can be assigned one of eight settings, as illustrated in Table 4-27. Next, we compute the average value for each of our responses under each condition. Table 4-28 displays the results of this averaging.

Table 4-27. Mapping of Factor Settings to Eight Conditions
(M means the – level of a factor and P means the + level of a factor)

		Values		
Condition	Factor Settings x1:x2:x3	Propagation Delay Multiplier	Backbone Router Speed	Buffer Sizing Algorithm
C1	M:M:M	1	800	*RTT*x*C*/SQRT(*n*)
C2	P:M:M	2	800	*RTT*x*C*/SQRT(*n*)
C3	M:P:M	1	400	*RTT*x*C*/SQRT(*n*)
C4	P:P:M	2	400	*RTT*x*C*/SQRT(*n*)
C5	M:M:P	1	800	*RTT*x*C*
C6	P:M:P	2	800	*RTT*x*C*
C7	M:P:P	1	400	*RTT*x*C*
C8	P:P:P	2	400	*RTT*x*C*

Table 4-28. Average Response Values for Each Condition

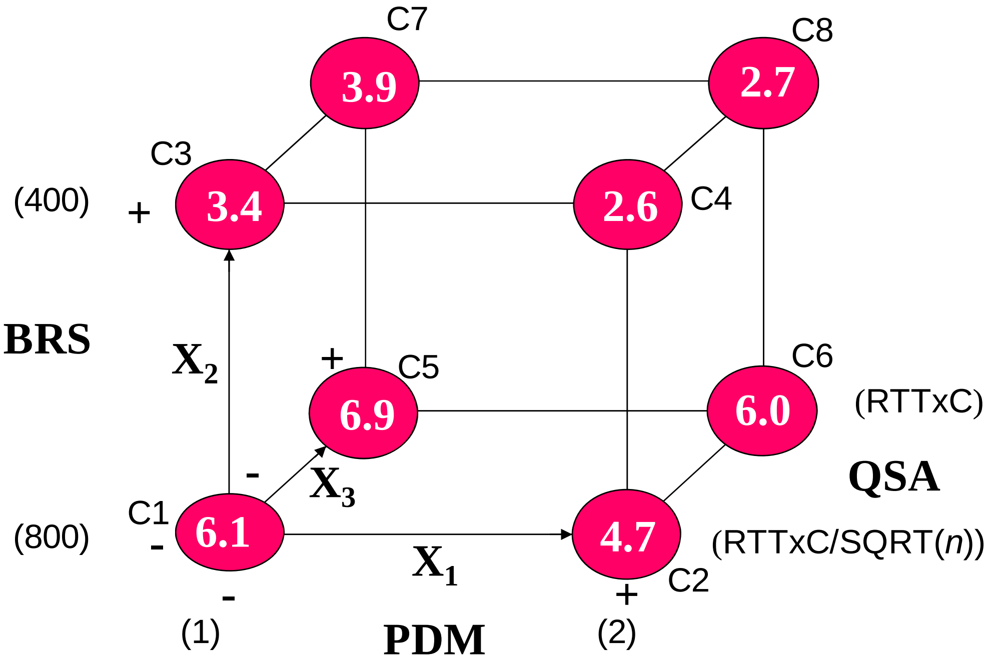
	Response						
Condition	y4	y6	y10	y16	y17	y20	y22
C1	109 863.71	1 384.55	0.07	1.53	229.88	167.12	107.06
C2	104 049.61	1 218.14	0.05	1.45	138.02	97.66	70.59
C3	68 721.38	803.31	0.27	1.41	229.82	89.81	37.48
C4	69 996.51	872.54	0.17	1.39	137.65	73.92	28.25
C5	111 195.23	1 324.93	0.02	2.52	237.65	169.29	119.39
C6	109 949.76	1 409.62	0	1.91	138.17	106.73	96.33
C7	74 509.45	956.32	0.08	3.27	226.01	131.99	51.3
C8	70 170.53	881.34	0.04	2.83	136.26	79.82	37.04

Using the average responses from Table 4-28, we next rank each condition from high (8) to low (1) for each response, based on the appropriate ordering criteria. For retransmission rate (y10) and relative queuing delay (y16) a lower value would be ranked higher. For the other five responses in Table 4-28 a higher value would be ranked higher. After ranking the conditions with respect to each response, we compute an average ranking. The results of our ranking are shown in Table 4-29.

We can assign the average rank for each condition to the vertex of a cube, where each vertex represents a specific combination of settings for propagation delay, network speed and buffer size. Fig. 4-32 shows the cube corresponding to Table 4-29. Moving along the edges among the vertices on the cube allows us to determine changes in ranking attributable to each factor. The change in each factor (x1, x2 and x3) across all conditions is represented by a set of four different edges from among the 12 edges contained in the cube. We extract the relevant changes in ranking and display them in Table 4-30.

Table 4-29. Ranking for Each Condition vs. Each Response

	Condition							
Response	C1	C2	C3	C4	C5	C6	C7	C8
y4	6	5	1	2	8	7	4	3
y6	7	5	1	2	6	8	4	3
y10	4	5	1	2	7	8	3	6
y16	5	6	7	8	3	4	1	2
y17	7	3	6	2	8	4	5	1
y20	7	4	5	1	8	5	6	2
y22	7	5	3	1	8	6	4	2
Average Rank	6.1	4.7	3.4	2.6	6.9	6.0	3.9	2.7



4-32. Average Condition Ranking Displayed on Vertices of a Cube

Table 4-30. Changes in Ranking Attributable to Each Factor

	Propagation Delay (x1)	Network Speed (x2)	Buffer Sizing (x3)
Edge 1	1.4	2.7	0.8
Edge 2	0.8	2.1	0.5
Edge 3	0.9	3	1.3
Edge 4	1.2	3.3	0.1

Interpreting Table 4-30 we see that changing network speed has the largest effect on the responses we selected. Changing propagation delay has the second largest effect. Changing buffer-sizing algorithm has the smallest effect. Further, Fig. 4-31 shows that changing from fewer to more buffers has a larger effect when network speed is high and propagation delay is long. This makes intuitive sense because more packets could potentially be inside the network when speed and propagation delay increases, so a higher proportion of the increased buffers would likely be occupied.

While our examination of the effect of buffer-sizing algorithm should not be considered definitive, the results extracted from our sensitivity experiment tend to support the findings of McKeown and colleagues. For the topology and traffic patterns used in our study, reducing buffer sizes by the square root of the expected number of active flows transiting each router had little overall effect on macroscopic network behavior and user experience. On the other hand, we found that reducing buffer sizes can markedly restrict the range of variation in queuing delay and thus in round-trip times. Reducing variance in round-trip times allows faster feedback on losses and permits the TCP flows to adapt more quickly, which could offset some of the losses that might otherwise occur due to reducing the number of buffers.

4.8 Conclusions

We described a method for conducting sensitivity analyses for simulations of large, complex systems, such as communications networks, computing grids and service-oriented architectures. The method included: orthogonal fractional factorial (OFF) design of two-level experiments, correlation and principal components analyses and a ten-step graphical analysis. We applied the method to gain an understanding of MesoNet (described in Chapter 3). Correlation and principal components analyses revealed the main dimensions of MesoNet behavior to include: (1) congestion, (2) delay, (3) throughput of advantaged flows and (4) aggregate rate of flow completions.

Sensitivity analysis identified the main factors influencing each aspect of MesoNet behavior. Congestion is influenced primarily by network speed, number and distribution of sources and average idle time for sources. Delay is influenced primarily by buffer size and propagation delay. Advantaged flows come in two categories: DD flows and FF (and DF) flows. Throughput for DD flows is influenced by two factors: propagation delay and file size. While propagation delay and file size prove influential, throughput for FF flows is also affected by network speed and distribution and idle time of sources. These additional factors reflect situations where increased numbers of FN flows compete with DF and FF flows. The aggregate rate of flow completions is influenced by network speed, source idle times and file size. Using rank analysis, we found the order of overall influence exerted by key factors. From more to less influential, the overall influence of factors was ordered as follows: network speed, file size and idle time, number of sources, propagation delay, distribution of sources and buffer size.

Sensitivity analysis also identified four factors that had little influence on the behavior of MesoNet. These factors included: probability of electing to download a larger file, probability of sources and receivers residing on fast hosts, distribution of receivers and initial slow-start threshold.

We extended our analysis to investigate explicitly the relative influence of network speed, propagation delay and buffer size on overall behavior of the model. We found that network speed had greatest influence and buffer size had least influence. We also showed that very small buffer sizes restrict the range of variance in smoothed round-trip times. Further, we found that buffer size has greater influence on model behavior when network speed and propagation delay are larger.

We also conducted a second sensitivity analysis (see Appendix C) that used the same 2^{11-5} OFF experiment design template documented in Sec. 4.1, but that changed the specific values assigned to each of the 11 experiment factors so that network size and speed were increased and so that the distances between the – and + levels for each parameter were expanded. The sensitivity analysis documented in Appendix C found the same main factors driving MesoNet behavior as we documented in Chapter 4, though the influence of propagation delay was increased due to the expanded distance between the values chosen for the two levels. In general, the network simulated in Appendix C exhibited lower correlations because overall congestion was lower. This was also reflected in changes in principal components, which became more difficult to interpret.

Overall, analyses conducted on MesoNet increase our confidence in the model's correctness and reasonableness. The fundamental characteristics of the model and the topology, as investigated here and in Appendix C, provide a reasonable basis for comparing the effects of alternate congestion control algorithms on macroscopic network behavior and user experience. In the next section, we discuss the congestion control algorithms we will study and we show how we modeled those algorithms.