A Framework for Preparing Experimental Evaluation of Rerouting Mechanisms

Reinaldo de B. Correia[1] Luiz F. Rust Carmo[1] Luci Pirmez[1] Luiz F. H. Bacellar[2]

reinaldo@posgrad.nce.ufrj.br *rust@nce.ufrj.br* *luci@nce.ufrj.br* *bacellar@utrc.utc.com*

[1] Núcleo de Computação Eletrônica
Universidade Federal do Rio de Janeiro,
P.O.Box 2324, 20001-970, Rio de Janeiro – RJ, Brasil

[2] United Technologies Research Center
411 Silver Lane, M/S 129-48, East Hartford,
CT – 06108, USA

Abstract

Among the various QoS schemes that have already been proposed, rerouting critical flows is the most promising for its inherent ability to interface easily with management systems. To enhance the responsiveness, the operations performed by rerouting mechanisms should be done pro-actively. A pro-active network management using rerouting mechanisms was proposed in [1]. Recently, the proposed approach was selected to be deployed in an experimental fiber-optic gigabit backbone (GIGA project - sponsored by the Brazilian Research Network - RNP). The fundamental problem is how to specify critical factors and to develop a consistent set of tests for a complete field evaluation of the implementation. The constraints associated with this process are reduced deployment time and restricted availability of the network infrastructure for testing. This paper discusses a framework for deploying the rerouting mechanisms proposed in [1] using DOE (design of experiments) analysis [11]. It is shown how rapidly some conclusions about performance, stability and interdependence upon the selection of design parameters can be taken through the use of DOE. Basically, MatLab models have been used to support the DOE analysis. This paper shows how the use of simple two and three-level factorial experimental designs can rapidly increase the designer's knowledge about the behavior of the mechanisms being studied.

1. Introduction

Rerouting may be understood as the set of actions carried out by networks to establish new routes for application flows. These actions are normally accomplished by the Network and Link layers, but some Physical-layer proposals exist [2]. The shortcomings of these three approaches are related to the rerouting latency as well as the number of application flows that are affected, that is, the granularity of the flows [5]. The higher the layer, the finer is the granularity and the higher is the latency. Whereas in the Network layer it is possible to reroute a particular application flow at the expense of a higher latency, in the Physical layer rerouting is done promptly but the number of flows involved is high.

Traditionally, network topology events such as devices and link failures have already been used for triggering rerouting operations in order to mitigate their effects. The goal is to provide a high degree of robustness [5], increasing system availability. By nature, this scheme is reactive as long as triggering occurs after fault occurrences. Since their creation, the Internet rerouting protocols, - RIP, OSPF and BGP -, have been employing reactive rerouting successfully. The key idea behind these protocols is to keep network working on top of prone-to-fail topologies. This scheme worked fine until multimedia applications were first introduced on the Internet. The stringent constraints of these time-sensitive applications and high latencies presented by Internet routing protocols have been restraining the large-scale deployment of these new applications. The high latencies for failure recovery arise due to the convergence time of these protocols, which are a function of network size (number of nodes).

To both address the QoS requirements from multimedia applications and circumvent the failure recovery delay of Internet routing protocols, additional rerouting mechanisms may be attached to network nodes, running concurrently with routing protocol processes. In addition to being responsive to topology failures, rerouting mechanisms should be able to react against QoS faults, that is, when multimedia QoS requirements are not met by the underlying network. In [1], a rerouting framework was introduced, which not only encompasses these features but also employs mechanisms in such a way that the rerouting operations can be performed pro-actively. The pro-active approach is achieved by executing all possible operations in advance, during the period when the applications flows are being monitored. In addition, the rerouting actions are started as soon as tendencies of QoS faults are detected.

These rerouting mechanisms were selected to be deployed in an experimental fiber-optic gigabit backbone (GIGA project - sponsored by the Brazilian Research Network -RNP). Although the feasibility of the adopted rerouting scheme was demonstrated in [1], its stability is still an issue. In order to come up with a solution, system dynamics should firstly be investigated. The first step is to find out critical variables (factors), which are essential to meet operational temporal requirements. It is fundamental for a designer to understand the impact of critical parameters and the respective tradeoffs to effectively allocate requirements and devise the best solutions. The next step comprises the development of a consistent set of experimental tests for field validation of those mechanisms. The constraints associated with this process are: (i) reduced development time and (ii) restricted availability of the network infrastructure for testing.

This paper deals with an integrated framework, based on the use of design of experiments (DOE) technique, as an appropriated methodology for (i) functional analysis and test definition, and (ii) experimental evaluation of rerouting mechanisms. This framework is part of a larger investigation being carried out at the NCE/UFRJ in Brazil, which applies modeling and simulation to the analysis and requirements definition of new network mechanisms for management and routing.

Basically, the framework proposed here can be decomposed in three main phases, as follows.

Model development and analysis

The performance of rerouting mechanisms is typically dependent upon the selection of different design parameters, variables. It is necessary to carry out performance estimations related to the effect of different design options (e.g., switching threshold) in order to obtain implementation guidelines. Models are an effective means of understanding quantitative issues, and predicting the interesting properties or different design alternatives - and thereby minimize risk – before entering the expensive and time consuming implementation phase. For instance, restricting the number of redundant channels for rerouting, the higher throughput achieved could improve the responsiveness but at the same time might introduce significant switching delays. These models can be more or less detailed, given that they provide the right level of abstraction. Having a QoS measure, which takes all timing effects into account, can be helpful when designing complex rerouting mechanisms. MatLab models have been developed and used to determine the variation of rerouting metrics due to some strategic controllable factors.

Test sequence generation and validation

In the past decade, the application of DOE has gained acceptance as an essential tool for systems evaluation and validation. Properly designed and executed, partial factorial experiments generate data with acceptable precision while using substantially fewer experimental runs than alternative approaches. The partial factorial experiments lead to results that are easily interpreted, in contrast to the information gathered in other studies, which sometimes are difficult to be interpreted. Another advantage of the partial factorial design is its efficiency; the partial factorial design allows each factor to be evaluated with the same precision as in the one-factor-at-a-time experiment, but with a smaller number of runs. When DOE is used for testing / evaluation of rerouting mechanism implementation, it reduces considerable testing time and costs. The rerouting models can also be used to assess the competence of partial factorial experimental design versus the full factorial approach.

Testing and evaluation of the rerouting implementation

The bottom line is how to perform a consistent evaluation of the rerouting mechanism when using the real infra-structure (optical backbone), without incurring large testing and deployment time. Usually, porting any network mechanism to a real backbone may change the timing behavior previously obtained during the model development and analysis phase due to effects such as different traffic distributions, processing delays, etc. To reduce those differences, it is recommended to carry out field traffic estimation as the first evaluation step, and use it to perform a more accurate simulation, before running DOE test sequence.

This paper describes the first two phases of this framework developed as part of the GIGA project, which aims at implementing rerouting and management capabilities in an optical research network in Brazil.

This paper is organized as follows. Section 2 presents the rerouting concepts necessary to understand the modeling approach. Section 3 introduces the proposed model architecture. Section 4 describes the use of DOE technique inside the proposed framework for preparing experimental evaluation of a rerouting mechanism. Section 5 points out the simulations performed over the model and analyses the results. The conclusions of this work are described in Section 6.

2. Rerouting

Rerouting flows through networks is being considered as an important component to improve the availability and response time of distributed systems. Since the convergence time of IP routing algorithms is usually very high [3] and is dependent on network size, rerouting real-time multimedia traffic flows is unfeasible. To address this problem, virtual circuit technologies such as the Multiprotocol Label Switching (MPLS) framework [4] are being deployed on IP networks to provide for the efficient designation, routing, forwarding, and switching of traffic flows. These technologies are critical to increase the probability that application QoS needs will be satisfied by the network infrastructure.

Virtual circuit driven networks (ATM, Frame Relay and MPLS) stem on swapping of labels to forward packets. In particular, MPLS defines virtual circuits, known as Label Switched Paths (LSP's), over connectionless environments such as IP networks in order to support connection-oriented like services. Rerouting in virtual circuit networks is defined as the set of operations necessary for redirecting pre-established flows through redundant routes. Rerouting is generally employed to: (i) support administrative policies, (ii) to establish traffic profiles and (iii) to increase the degree of fault tolerance.

Traditional rerouting approaches are reactive in nature since rerouting actions are only taken after a fault is detected. In contrast, pro-active rerouting must be able to find new paths before faults occur. Such pro-active operations lead to significant reduction of rerouting latency. Basically, the pro-active rerouting scheme consists of three set of operations: (i) identification of alternative paths: (ii) generation of local identifiers (labels) on all nodes that belong to all virtual circuits just discovered and (iii) redirection of the flow by replacing the current label.

In this context, faults take place when QoS constraints are not met due to lack of resources on the current virtual circuit of an application flow. There are two approaches for redirecting flows pro-actively: plain and partial rerouting. Plain rerouting replaces the whole virtual circuit of a pre-established flow with new redundant virtual circuit, while the partial one only substitutes a section of the current virtual circuit. The partial rerouting approach speeds up the operations and turns the rerouting process independent of the virtual circuit length. Latencies on the order of several seconds are common for virtual circuits larger than 4 hops [6].

The partial approach presumes there is a section that must be identified in the current virtual circuit. This section, the critical section, is the one, which is the bottleneck of the virtual circuit considering the QoS metric relevant to the flow (e.g., lowest bandwidth, higher delay). The scheme restrains the searching area so that the number of nodes and links involved in rerouting operations becomes small. The final result is a lower processing time and consequently, lower rerouting latency. Moreover resource consumption in rerouting tasks is also reduced turning the scheme more scalable.

The rerouting strategy described above was adopted by a proactive rerouting architecture introduced in [1]. Its goal is to avoid QoS failures by redirecting application traffic flows through redundant routes. Test results showed that it is feasible to employ rerouting over a Java-based middleware using the mobile agent paradigm [7]. It also demonstrated that it is suitable to reroute traffic flows through less congested virtual circuits on a Multiprotocol Label Switching (MPLS) network infrastructure.

3. Model Overview

The models developed are intended to represent the essence of the rerouting mechanism. Typically, performance metrics that are associated with rerouting are (i) the number of rerouting operations which occur during the lifetime of the application flow transmission (Transitions) and (ii) the average delay of such transmission (Average delay). The basic objective was to start from a representative scenario that could be successively enhanced as new inputs (traffic parameters, rerouting algorithms, etc) were being identified.

Figure 1 exhibits a SIMULINK model, which represents the functional decomposition of the rerouting mechanism. It is important to note that this decomposition is intended to be scalable to any number of channels but currently only accounts for those mentioned later. As the system scope changes, the details of the functions shown below will change, however the functional decomposition should not. In the following, a short description of each functional block of Figure 1 is given.

Traffic Generator: Implements the background traffic sources of each virtual channel as well as the source of the application traffic. Each background traffic source generates independent random traffic for each virtual channel. Only Gaussian and Uniform distributions are currently modeled. The application traffic source was configured to produce a constant bit rate (CBR) traffic.

Switch commander: Commands a change from the channel that is currently in use, the active channel, to one of the available redundant channels (Transition). These transitions are carried out in accordance with the instantaneous utilization rate and the switching threshold previously set. It also implements the logic necessary to guarantee that only one channel is active at any time.

Arbitrator: Implements the switching algorithms, which selects one of the redundant channels that must be in use at a given time. Actually, the used algorithm chooses the least used redundant channel at the moment when a rerouting operation is commanded.

Virtual Channel: Translates data rate into data delay using an exponential function.

every possible way so that all combinations will be evaluated.

Carefully planned and designed, the DOE experiments offer clear advantages over traditional one-factor-at-a-time alternatives. This technique is particularly useful for rerouting studies, where the effects of various factors on the design must be determined. Besides being easily understood, the DOE concept employs factorial experiment designs, which are easy to construct, efficient, and capable of determining interaction effects [8].

The process used to the evaluation of rerouting mechanism was performed in six steps:

1 - Establish the set of factors and the number of states necessary to fully represent: (i) the parameters related to the functional choices (controllable) and (ii)

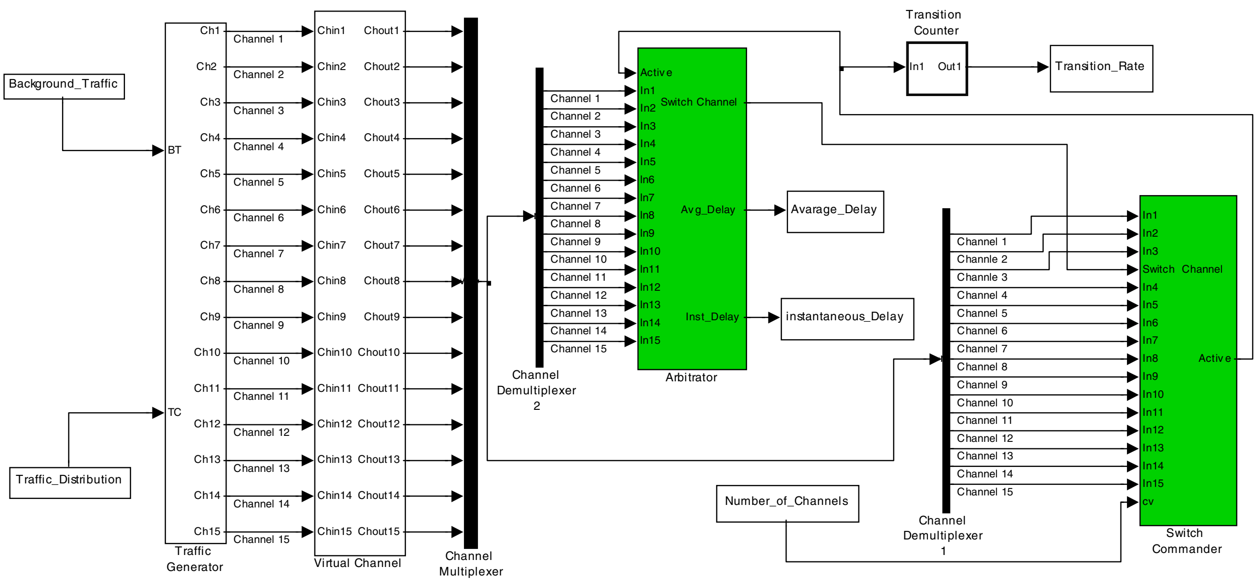

Figure 1. Functional model of the rerouting mechanism

4. Design of experiments process

Traditional methods of experimentation evaluate only one variable (or factor) at a time (sensitivity analysis): all of the variables are held constant during test runs except the one being studied. This type of experiment reveals the effect of the chosen variable under set conditions; it does not show what would happen if the other variables also changed.

The idea behind DOE is based on the fact that is much better to vary all the factors at once using a factorial design, in which experiments are run for all combinations of levels for all of the factors. With such a study design, testing will reveal what the effect of one variable would be when the other factors are changing. The job is to attempt to break the system in

the parameters related to the use context of the mechanism (uncontrollable);

2 - Choose the best Orthogonal Array-Based Test Case in order to fully cover the number of factors and states previously selected; taking into account the capability of evaluating interaction between two or more factors;

3 - Perform a full factorial exploration of design space and fill the orthogonal array test case (DOE analysis);

4 - Calculate the average effect of each factor for both the full-factorial and the orthogonal test case in order to determine how sensitive metrics are to change in factors, and to compare and validate the reduced matrix of experiments;

5 - Determine interaction between parameters and down-select the most sensitive ones;

6 - Use response surface to estimate isometric curve to meet requirement for critical metrics.

4.1. Simulation Matrix

The set of factors and states necessary to fully represent the design choices of the rerouting mechanism and the parameters related to the rerouting application context are described as follows.

Controllable Factors:

•***Number of virtual channels*** - denotes the maximal number of channels used during a rerouting operation. This factor restrains the searching area of nodes and links involved in rerouting operations. The recommended number to be used is about 0 to 15.

•**Switching *threshold*** - defines the utilization limit of the active channel throughput before firing a rerouting operation. The Switching threshold is represented by a percentage number, ranging from 0 to 100%.

Table 1. Selected Factors–Gaussian distribution

No.	Controllable Factors	Levels			Units
		1	2	3	
A	Number of virtual channels	5	10	15	-
B	Switching threshold	0.250	0.5	0.75	100%
No.	**Uncontrollable Factors**	**Levels**			**Units**
		1	2	3	
C	Mean of background traffic	0.15	0.25	0.35	100%
D	Variance of background traffic	0.025	0.05	0.075	100%
No.	**Metrics**				
Tc	Transitions (count)				
Ad	Average delay (t.u.)				

Uncontrollable Factors: for the purposes of the rerouting analysis and design decisions, it is important to evaluate different channel background traffics. As the Gaussian distribution was selected to model the background traffic, the two uncontrollable factors are the mean and variance of the traffic, both represented by percentage numbers.

Scenario: the target scenario consists of a CBR stream transmission, consuming 25% of the channel capacity, through an elected channel from a set of 5, 10 or 15 redundant channels. Each channel has the same capacity and is fed by an independent Gaussian background traffic (using a different seed). In fact, the use of a homogeneous set of traffic sources is a direct consequence of major stability concerns.

The fundamental point is not to analyze convergence time to the slightest used channel, but to observe system behavior whenever the contour conditions benefit a maximal number of rerouting operations. The total extension of observable process is 20 units of time (ut) and the re-evaluation of rerouting conditions is done each 0.1 ut (discrete model). The use of a discrete model limits the amount of transitions during the stream transmission, given by the maximal number of eligible rerouting operations (in this case 201), but it is the best representation of the implementation context. Table 1 summarizes all the factors chosen to evaluate the rerouting operation. Three levels (states) had been picked for each factor.

4.2. Orthogonal Array-Based Test Cases

Next the best Orthogonal Array-Based Test Case must be chosen in order to fully cover the number of factors and states previously selected, taking into account the capability of evaluating interaction between two or more factors [9]. Table 2 shows a fractional design using OA (Orthogonal Array) L9. It has nine rows and four columns.

Table 2. OA L9 matrix

Test Plan Using OA L9				
Test Number	**A**	**B**	**C**	**D**
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

The rows correspond to the experiments and the columns correspond to the factors. Thus, the first experiment comprises Level 1 for each parameter, i.e., it represents the combination A1, B1, C1, D1. The second test case comprises combination A1, B2, C2, D2, etc. An orthogonal array has the balancing property that, for each pair of columns, all parameter-level combinations occur an equal number of times. In OAL9, there are nine parameter-level combinations for each pair of columns, and each combination occurs once. By conducting the nine experiments indicated by L9, we can accomplish the following: (i) detect the effects of all single-mode factors; (ii) detect the effects of all double-mode factors, i.e. interactions between two factors.

5. Simulation Results

The first step in the rerouting analysis consisted of performing a full factorial exploration of the design space, filling the orthogonal array test case. Thereafter the partial factorial was computed through the use of OAL9 shown in Table 2, leading to the results shown in the matrix of Table 3.

Table 3. OAL9 results

Simulations					Results	
Test Number	Number of Channels	Switching threshold	Mean of background traffic	Variance of background traffic	Transitions	Average delay (t.u.)
1	5	25%	15%	2.5%	160	1.37
2	5	50%	25%	5.0%	101	1.52
3	5	75%	35%	7.5%	60	1.97
4	10	25%	25%	7.5%	177	1.33
5	10	50%	35%	2.5%	149	1.61
6	10	75%	15%	5.0%	13	1.76
7	15	25%	35%	5.0%	187	1.38
8	15	50%	15%	7.5%	72	1.41
9	15	75%	25%	2.5%	12	2.00

The values in Table 3 were obtained by running each experiment 100 times with different seeds for the background traffic source. This analysis process involves taking the average response of a particular factor at the three levels (states) and calculating the larger difference between those factors. This difference is referred to as the effect of that factor. As an example, in Table 3, the average response of the *Number of Channels* at the level 1 would be adding up the 3 averages corresponding to the 5 indications in the *Number of Channels* column and dividing by 3. This process of estimating the factors effects is sometimes called ANalysis Of Means (ANOM).

The next two figures show the main effects plot for Transitions (Figure 2) and Average delay (Figure 3). The main effects plots show the strengths of the various effects. Those plots are constructed by plotting the transitions mean and the average delay of a particular factor at the three levels [10]. The results confirmed that the Switching threshold has the largest effect and it is the dominant factor for both metrics, and consequently, is also the most critical controllable factor.

However, a subtle and unexpected system behavior is revealed: the *Number of channels* is the least significant factor, meaning that the size of the rerouting search area is not a major issue. Figures 4 and 5 show the main effects plot generated according to the OAL9 reduced set of experiments (table 3). Both OAL9 and full factorial experiments led to results which allow reaching the same conclusions.

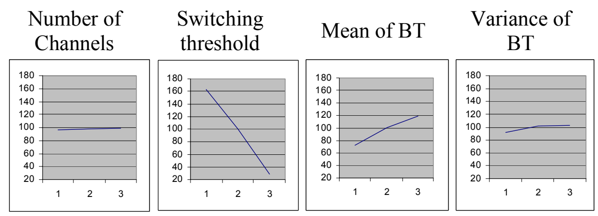

Figure 2. Transitions-main effects (full factorial)

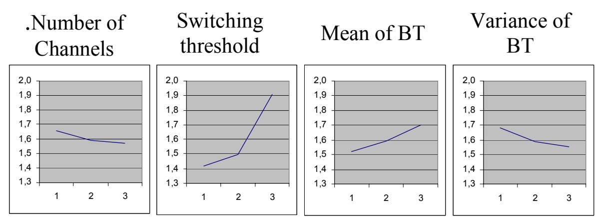

Figure 3. Average Delay-main effects (full factorial)

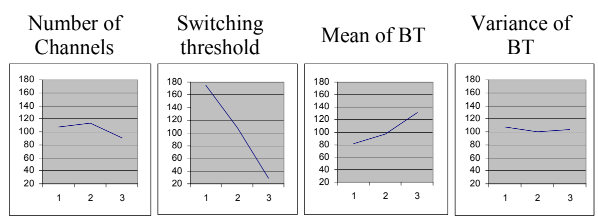

Figure 4. Transitions - main effects (OAL9)

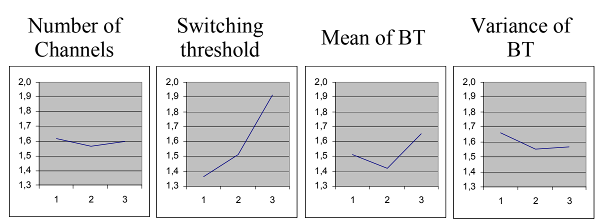

Figure 5. Average Delay - main effects (OAL9)

5.1. Interactions Plots

Another effect that we have to be concerned with is the interactions between two or more factors. Interactions occur in the two factors interaction case, when the response difference depends on the setting of another factor [11]. In order to illustrate how these interaction plots are generated, Table 4 shows the *Number of Channels* (Ch) and the *Switching threshold* interactions (ST) for the number of transitions. Notice that the various levels for the (Ch x ST) interactions are arrived at by calculating the average of every experiment showing the expected combination of factors levels: Ch1xST1, Ch1xST2…Ch3xST3.

Table 4. Partial Extract of the full factorial table

	Ch 1	Ch 2	Ch 3
ST 1	159.6	162.6	166.9
ST 2	100.1	100.5	100.9
ST 3	28.5	28.9	28.8

The interactions plots among the dominant factor Switching threshold and all the others factors for both transitions and average delay metrics are shown in Figures 6 and 7, respectively. Almost all the interactions plots show that the interactions are small by the fact that the line pairs are nearly parallel. An example of a two factors that have very little to no interaction is shown in the Number of Channels / Switching threshold interactions plot for transition metric. Notice that the change in maximum deviation going from the ST1 setting to the ST2 setting in Ch1 is about the same for either setting of the Ch2.

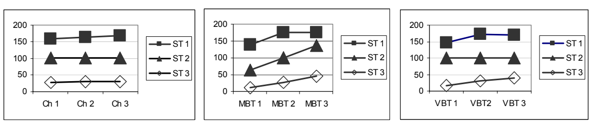

Figure 6. Transitions interactions plots

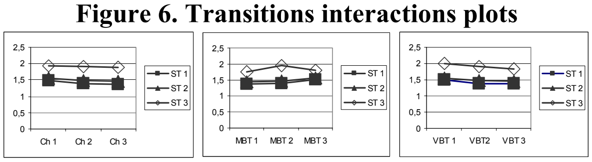

Figure 7. Average delay interactions plots

Graphically when there is an interaction, the lines on the interaction plot stop being parallel to each other (some times called synergistic interaction), and for very strong interactions the lines could cross (antisynergistic interaction). However, even that not so pronounced, there is some divergences between the Switching threshold and Mean in both transitions and average delay plots, and also between Switching threshold and Variance plots for average only, meaning the existence of insipid interactions.

Resuming, the conclusions driven by the main effect and interaction plots so far are:

•The *Switching threshold* has the largest effect and it is the dominant factor (and also the most critical controllable factor).

•There is light interaction between the *Switching threshold* and the *Mean Background traffic* for both Transitions and Average delay metrics as the *Switching threshold* decreases, the interaction becomes more pronounced.

•There is also a light interaction between the *Switching threshold* and the *Variance Background traffic* for Average delay metric.

5.2. Response Surface

The construction of the response surface for Switching threshold have been done by running 100 different simulations, covering a range of values for Switching threshold from 0.1 sec to 1.0 (100%) - with a 0.1s step, and a range of values for Mean of background traffic from 0.05 to 0.5 (100%) - with a 0.1 step. Figure 8 presents the transition response surface and isometric lines as a function of the *Switching threshold* and the *Mean of Background traffic*.

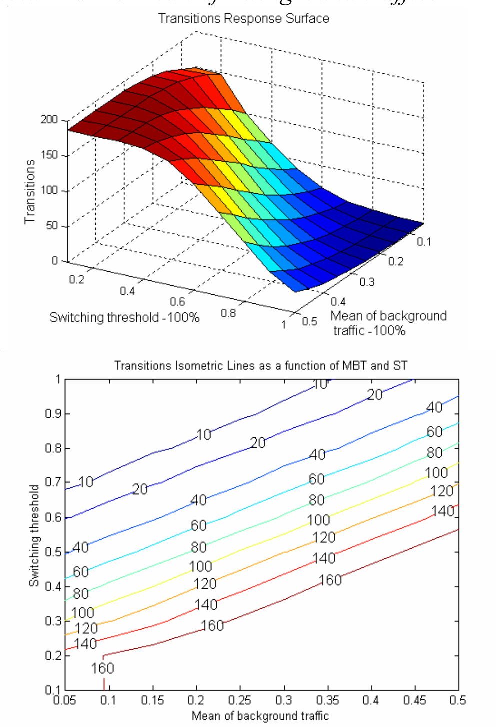

Figure 8. Response surface and Isometric lines for Transitions

These plots show the effect of the *Switching Threshold* on the rerouting mechanism in different background traffics (expressed by different Mean values). The isometric lines allow us to meet requirements for critical Transitions values. This is an important and known effect on rerouting, as each transition operation adds a fixed delay in the active stream transmission. Figure 9 presents the Average delay response surface and isometric lines as a function of Switching threshold and Mean background traffic. The visual inspection of both Transitions and Average delay isometrics lines confirms our early conclusion that stated that the interaction between the Switching threshold and the Mean background traffic is very light.

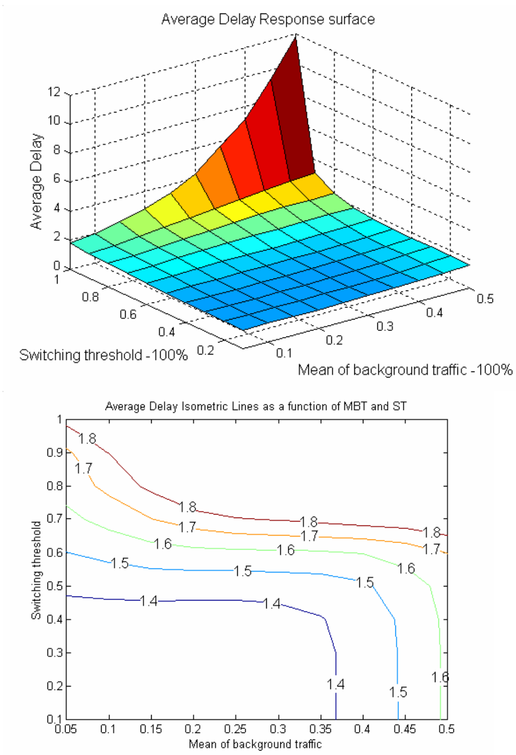

Figure 9. Response Surface and Isometric lines for Average delay

6. Conclusions

The use of DOE analysis to determine which parameters are most important for rerouting mechanism design decisions was extremely effective. The combination of main effect plots and interaction plots allows one to quickly identify the most influential parameters and understand whether that influence is isolated or not. Both OAL9 and full factorial sets of experiments produced the same conclusions, enforcing the use of OAL9 for the field evaluation phase. The use of response surface methods allows understanding the behavior of those influential parameters identified previously.

The most influential parameter is the switching threshold. The Number of channels is the least significant factor, meaning that the size of a rerouting search area is not a major issue. If a strict definition of stability is applied on the rerouting algorithm, it would be always considered stable as its discrete proprieties implies a finite amount of transitions at all times. However, the stability concept must be seen differently in order to incorporate applications requirements, that is, whenever the system generates a higher amount of transitions than the application can support is enough to be classified as instable. The isometric lines enable to derive requirements for limiting the amount of transitions and the average delay in function of the *switching threshold* and the *mean of Background traffic*.

Future work includes (i) performing real traffic estimation as the first step of a field evaluation for a more accurate simulation, before applying DOE test, and (ii) upgrading the model to take into account other constraints such as the delay introduced whenever a rerouting operation is fired.

7. References

[1] Reinaldo Correia, Edmundo Cecílio, Luci Pirmez, Luiz Rust and Luiz Bacellar, “A Proactive Management and Rerouting Framework for QoS Critical Distributed Applications Using Active Technology”, WORDS 2003F, Italy, 2003.

[2] Andrea Fumagalli et al, “IP Restauration vs. WDM Protection: Is there an Optimal Choice”, IEEE Network magazine, November / December 2002.

[3] Labovitz C. et al, “Delayed internet routing convergence”, In Proc. ACM SIGCOMM '00 pp. 175–187, Stockholm, Sweden, 2000.

[4] Rosen E. et al, “MPLS Architecture”, RFC–3031, IETF January 2001.

[5] Autenrieth, A. and Kirstdter, A. “Fault-Tolerance and Resilience Issues in IP-Based Networks, Second International Workshop on the Design of Reliable Communication Netwoks (DRCN), april 1995.

[6] G. Apostolopoulos et al, “Intradomain QoS Routing in IP Networks: A Feasibility and Cost/Benefit Analysis”, IEEE Network Magazine, 1999.

[7] Bieszczad, A., Pagurek, B. and White, T., “Mobile Agents for Network Management”, IEEE Communication Surveys, Fourth Quarter, 1999.

[8] Kim, John S. and Kalb, James W., “Design of Experiments: An Overview and Application Example”, http://devicelink.com/mddi/archive/96/03/011.html.

[9] Montgomery, Douglas, “Design and Analysis of Experiments”, Wiley, 2001.

[10] Phadke, Madhav S., “Quality Engineering Using Robust Design”, Prentice–Hall, 1989.

[11] Phadke, Madhav S. - Design of Experiment for Software Testing –Six Sigma, http://www.isixsigma.com/library/content/c030106a.asp.