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

Since the deployment of real-time sensitive applications in the Internet, the research community has been struggling to leverage the IP best-effort networks for providing QoS assurance. Among the various QoS schemes that had already been proposed, rerouting critical flows seems to be a good choice 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 proactive network management and rerouting mechanisms was proposed in [1]. Recently, these rerouting mechanisms were 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 to this process are reduced development time and restricted availability of the network infrastructure for testing. This paper discusses a framework for deploying our rerouting mechanisms in the light of DOE analysis [11]. It will be shown how rapidly some conclusions about performance, stability and interdependence upon the selection of design parameters can be taken by its use. Basically, MatLab models have been used to evaluate DOE analysis. This paper shows how simple two and three-level factorial experimental designs can rapidly increase the designer's knowledge about the behavior of the mechanism being studied. The final purpose is to reduce development time and mitigate risks by assessing rerouting scenarios with models.

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 network and link layers, but some physical-layer proposals exist. The shortcomings of these three approaches are related to the granularity of the flows that are effected due to rerouting operations and its latency. The higher the layer, the finer is the granularity and the higher is the latency. That is, in the network layer, specific flows may be rerouted with higher latencies. Physical-layer rerouting is done promptly but the number of flows involved is high. The best choice is, of course, application dependent.

Traditionally, network topology events such as devices and link failures already trigger rerouting operations in order to mitigate its effects. The goal is to provide a high degree of resilience, 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 -, has been employing reactive rerouting successfully. The key idea behind these protocols is to keep network working on top of dynamical topologies. This scheme worked fine until multimedia application was first introduced in 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. High latencies arise due to convergence time of these protocols, which are a function of network size (number of nodes).

To both address these new QoS requirements and circumvent the delay limitation, additional rerouting mechanisms may be attached to network nodes, running concurrently with routing protocol processes. In addition to being responsive to topology failures, rerouting mechanism should be able to react against QoS faults.

In [1], it was introduced a rerouting framework, 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 that flows are being monitored. In addition, the rerouting actions are started as soon as tendencies of QoS fault 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], the system 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 tradeoffs to effectively allocate requirements and envisage the best solutions. Next step consists of developing a consistent set of experimental tests for posterior field validation of those mechanisms. The constraints associated to this process are reduced development time and the restricted availability of the network infrastructure for testing.

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

Basically, the framework can be decomposed in three main steps, described above:

Model development and analysis

The performance of rerouting mechanisms is typically dependent upon the selection of different variables. It is necessary to carry out performance estimations related to the effect of different design options, like 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 we go to the expense and trouble of implementation. For instance, restricting the effective use of channels throughput could improve its responsiveness but might introduce significant switching delays. These models can be more or less detailed, given that they provide the right abstraction. Having a quality of service measure, which takes all timing effects into account, can be helpfully 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 sequences 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, experiments generate more-precise data while using substantially fewer experimental runs than alternative approaches. They lead to results that can be easily interpreted, in contrast to the information gathered in other studies, which can be exceedingly difficult to interpret. Another advantage of the factorial design is its efficiency, the 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, certainly there is a large amount of savings in testing time and cost. The rerouting models can also be used to assess the competence of factorial experimental design versus the full factorial.

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 costly requirements of large time consumption. Usually, porting any network mechanism to a real backbone will change the timing behavior previously obtained from models due to effects such as different traffic distribution, processing delays, etc. To reduce those differences, it is recommended to perform a real 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 two first steps of this framework developed as part of the GIGA project that is in charge for implementing rerouting capabilities in an optical research network.

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 and, finally, section 6 reports some of the conclusions of this paper.

2 Rerouting

Rerouting of traffic flows through the network 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 [2] 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 [3] are being deployed on IP networks to provide for the efficient designation, routing, forwarding, and switching of traffic flows through the network. 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 (LSPs), 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 [6].

Traditional rerouting approaches are reactive in nature since rerouting actions are only taken after a fault is detected. In contrast, proactive rerouting must be able to find new paths before faults occur. Such proactive operations lead to significant reduction of rerouting latency. Basically, the proactive 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 proactively: plain and partial rerouting. Plain rerouting aims at replacing the whole virtual circuit of a pre-established flow with a new redundant virtual circuit, while the partial one only replaces a section of the current virtual circuit.

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. It also demonstrated that it is suitable to reroute traffic flows through less congested virtual circuits on a Multiprotocol Label Switching (MPLS) network infrastructure.

A prototype was implemented to evaluate the performance of the proposed proactive rerouting architecture. One of the metrics adopted was the *Redirecting Flow - RF* phase latency. Two cases were considered: on-demand label assignment and Early label assignment. For on-demand label assignment, the latency reached 98 ms. Early label assignment scheme makes RF phase latency be dependent on alternative path length due to the additional operations needed. To reroute a flow (RF phase), it is necessary 834.5 ms for a 5-hop-length alternative path [1].

3 Model Overview

The models developed so far 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 a stream 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 the functional decomposition of as rerouting mechanism as shown in the SIMULINK model. 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 function as shown in Figure 1 is given:

Traffic Generator: Represents the source of background traffics. An independent random number generator, meaning the instantaneous traffic, represents each channel. Currently only Gaussian and Uniform distributions are modeled.

Switch commander: Commands a change of the active channel (representing a rerouting activation) as function of the instantaneous utilization rate and the switching threshold.

Arbitrator: Implements the switching algorithms, which decide which virtual channel must be in use at a given time. Actually, the used algorithm choices the least used channel at the moment when a rerouting operation is commanded.

Virtual Channel(i): Translates data rate into data delay using a exponential function.

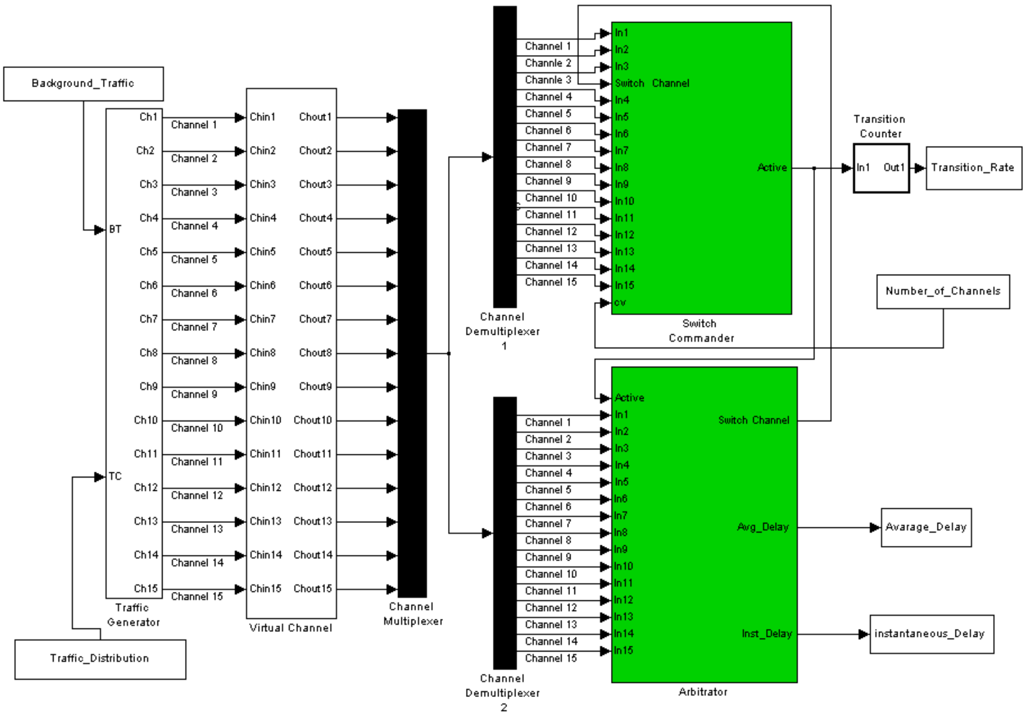

Figure 1: Functional Model of Rerouting mechanism

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 every possible way so that all combinations will be evaluated.

Carefully planned, designed experiments offer clear advantages over traditional one-factor-at-a-time alternatives. These techniques are particularly useful for rerouting studies, where the effects of various factors on the design must be determined. Not only is the DOE concept easily understood, the factorial experiment designs are easy to construct, efficient, and capable of determining interaction effects.

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

1. Establish the set of factor and number of states necessary to fully represent: (i) the parameters related to the functional choices (controllable), and (ii) 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 dowselect 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 use context (uncontrollable) are:

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

Uncontrollable Factors

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

Scenario

The target scenario consists of a CBR stream transmission, consuming 25% of a channel capacity, through an elected set of 5, 10 or 15 reroutable channels. Each channel has the same capacity and is fed by an independent Gaussian distributed 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 reevaluation 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.

Next table summarizes all the factors chosen to evaluate the rerouting operation. Three levels (states) had been picked for each factor.

No.	Controllable Factors	Levels			Units
		1	2	3	
a	Number of virtual channels	5	10	15	-
b	Switching threshold	0.25	0.5	0.75	100%

No.	Uncontrollable Factors	Levels			Units
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
A	Transitions (count)
B	Average delay (t.u.)

Table 1: Selected Factors for Rerouting evaluation – Gaussian distribution Scenario

4.2 ORTHOGONAL ARRAY-BASED TEST CASES

Next step consisted of choosing 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;

The next table shows a fractional design using OA (Orthogonal Array) L9. It has nine rows and four columns. The rows correspond to experiments; 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 OA L9, 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 effect of all single-mode factors. Detect all double-mode factors, i.e. interaction between two factors.

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

Table 2: OA (Orthogonal Array) L9

5 Simulation Results

The first step in the rerouting analysis consisted of performing a full factorial exploration of design space, filling the orthogonal array test case and also the OAL9 results matrix (Table 3). Those values are obtained by running each experiment 100 times with different source seeds

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

Table 3: OAL9 results

This analysis process will involve 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 the table above, the average response of *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 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* mean of a particular factor at the three levels.

The results confirmed that *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: *Number of channels* is the least significant factor, meaning that the size of a rerouting search area is not a major issue. Figures 4 and 5 show the main effects plot for generated from the OAL9 reduced set of experiments (table 3). Both OAL9 and full factorial sets of experiments produced the same conclusions.

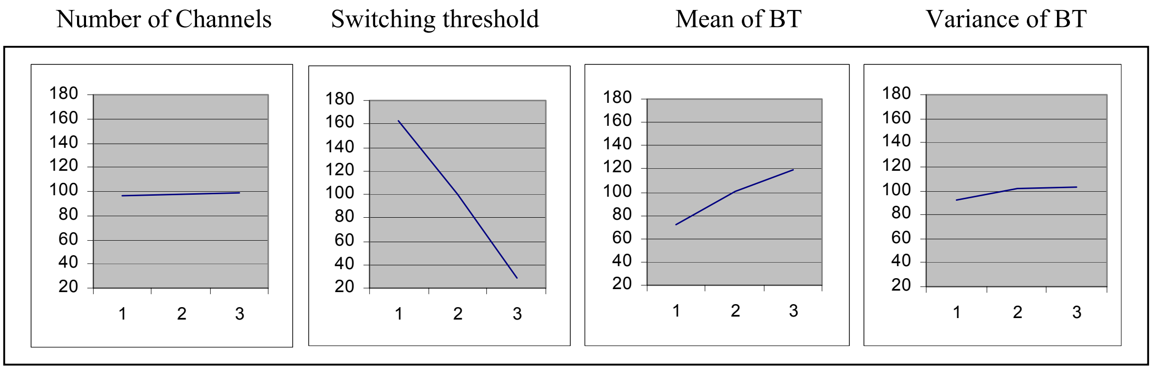

Figure 2 – Transitions - main effects plot

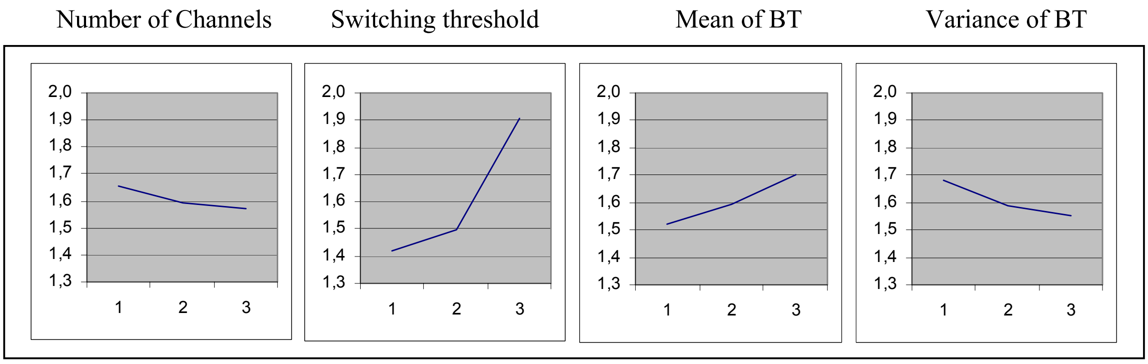

Figure 3 – Average Delay - main effects plot

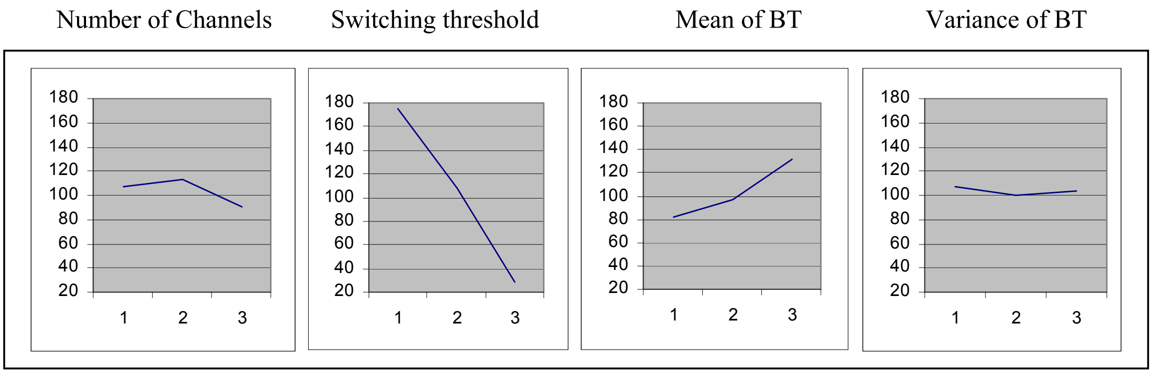

Figure 4 – Transitions - main effects plot using OAL9 experiments

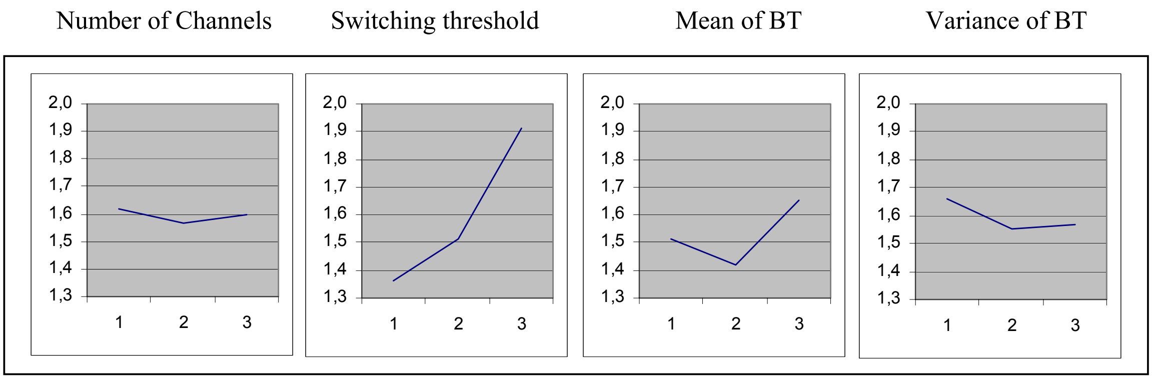

Figure 5 – Transitions - main effects plot using OAL9 experiments

5.1 Interactions Plots

Another effect that we have to be concerned with is 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. In order to illustrate how these interaction plots are generated, look at the example below showing the *Number of Channels* and *Switching threshold* interactions (Ch x ST) for maximum deviation during an unloading scenario (Table 4). 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.

	Ch 1	Ch 2	Ch 3
ST 1	159,5567	162,5711	166,9356
ST 2	100,1456	100,5222	100,8722
ST 3	28,53	28,89444	28,75222

Table 4 – Partial Extract of the full factorial table

Figures 6 and 7 show the interactions plots among the dominant factor *Switching threshold* and the all the others factors (for both transitions and average delay metrics). Almost all the interaction 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 transitions 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. Graphically when there is 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.

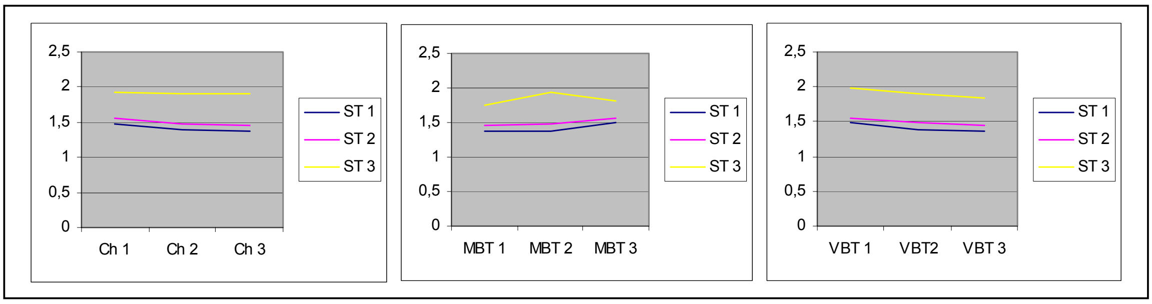

Figure 6 – Transitions interactions plots

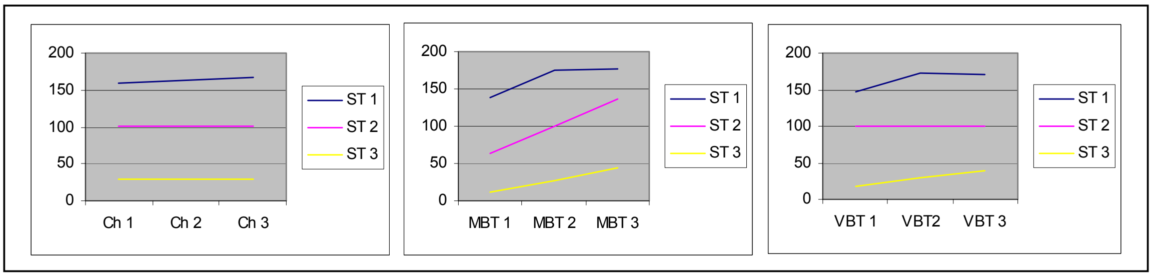

Figure 7 – Average delay interactions plots

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

- *Switching threshold* has the largest effect and it is the dominant factor (and also the most critical controllable factor).
- There are light interactions between *Switching threshold* and *Mean Background traffic* for both *Transitions* and *Average delay* metrics: as *Switching threshold* decreases, the interaction becomes more pronounced.
- There is also a light interaction between *Switching threshold* and *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 respectively the transitions response surface and isometric lines as a function of *Switching threshold* and *Mean of Background traffic*.

These plots show the effect of *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 respectively the *Average delay* response surface and isometric lines as a function of *Switching threshold* and *Mean background traffic*. By visual inspection of both *Transitions* and *Average delay* isometrics lines is possible to confirm the conclusion of early interaction plot analysis whereas interaction between *Switching threshold* and *Mean background traffic* is very light.

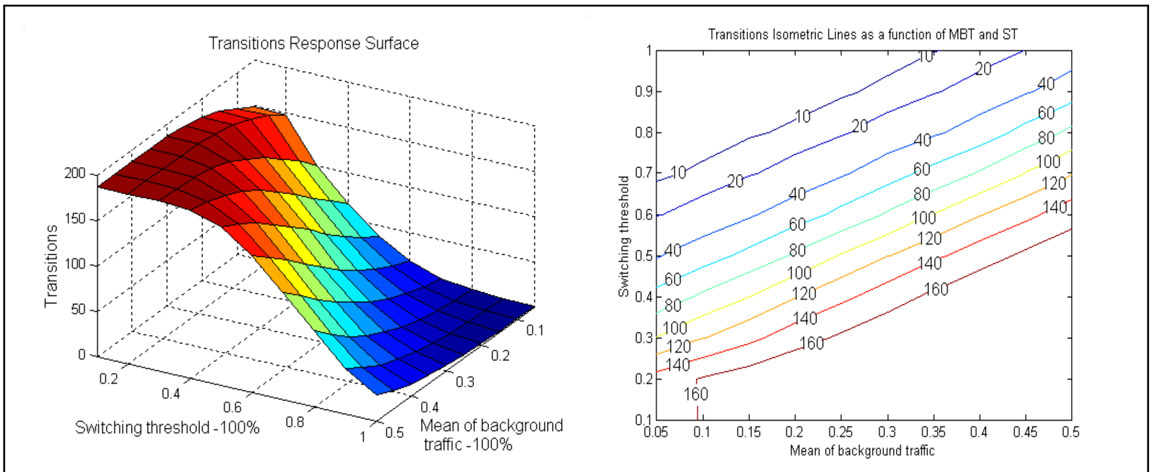

Figure 8 - Response surface and Isometric lines for Transitions

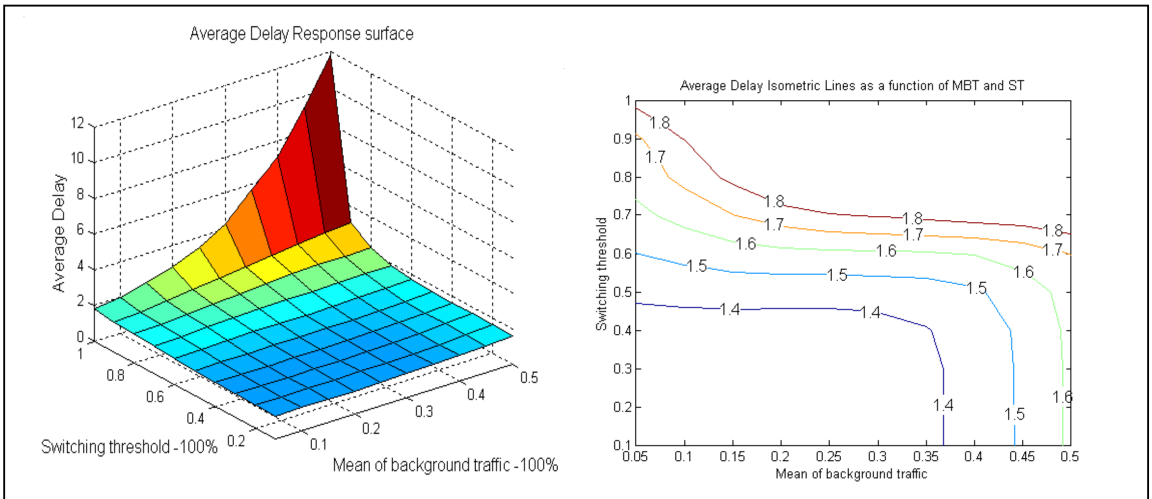

Figure 9 - Response surface and Isometric lines for Average delay

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 posterior use of OAL9 in the field evaluation phase. The use of response surface methods allow to precisely understand the behavior of those influential parameters identified previously.

The most influential parameter is the switching threshold. *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, i.e. whenever the system is possible to generate a higher amount of transitions than application can support is already enough to classify the mechanism as instable. By using the isometric lines described in this paper, it is possible to derive requirements for limiting the amount of transitions (and also *Average delay*) in function of *switching threshold* and *Mean of Background traffic.*

Future works include: (i) to perform a real traffic estimation as the first step of a field evaluation, and use it to perform a more accurate simulation, before applying DOE test, and (ii) to perform a model upgrade to take into account new alternatives for routing algorithms which also consider the delay introduced whenever a rerouting operation is fired.

7 References

[1] Reinaldo de B. Correia, Edmundo L.Cecílio, Luci Pirmez, Luiz F. Rust Carmo and Luiz F. H. Bacellar, "A Proactive Management and Rerouting Framework for QoS Critical Distributed Applications Using Active Technology", In Proc. WORDS 2003, Italy, 2003F.

[2] Kawamura, R. and Stadler, R., "Active Distributed Management for IP Networks", IEEE Communications Magazine, 2000.

[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.