

# PERFORMANCE OF RESTORATION PROTOCOLS IN OPTICAL MESH NETWORKS

Subir Biswas, Samir Datta, and Sudipta Sengupta

Network Architecture Group, Tellium, Inc.

2 Crescent Place PO Box 901, Oceanport, NJ 07757-0901.

## Introduction

Lightpaths in an optical mesh network are restored at two levels – (i) at local span or optical link level, and (ii) at end-to-end or path level. Whenever a failure is detected, the optical switch/cross-connect (OXC) closest to the failure first attempts to restore the connections by performing local span switching. If span switching fails, end-to-end restoration is attempted by the OXCs at the end-points of the path. In this paper, we discuss performance analysis results for Tellium’s span switching and end-to-end restoration protocols, called *StarNet-L* and *StarNet-P* respectively. Tellium’s *StarNet Modeler*<sup>TM</sup>, an OPNET based restoration simulation engine, has been used for evaluating the restoration protocols. The tool has an easy-to-use GUI, provides visual animation of restoration events as they occur, and uses advanced algorithms for primary and backup route computation.

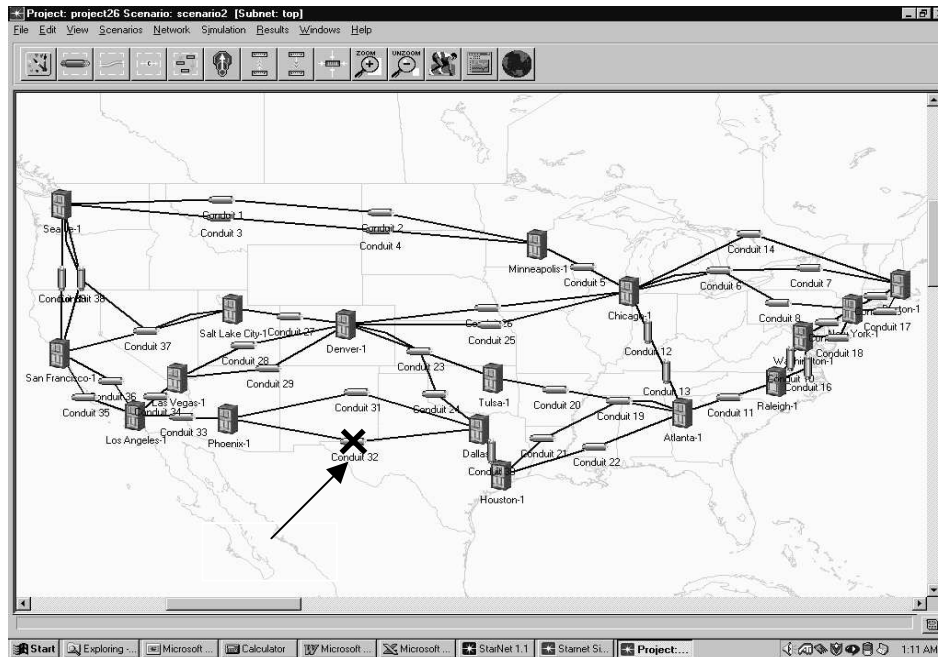


Figure 1: Experimental network topology: a US backbone

The experimental US backbone network topology is shown in Figure 1. Note that a wide range of physical route diversity is present in the network. For example, between the cities Phoenix and Dallas, there are two conduits (conduit-31 and conduit-32) that are physically route diverse. Route diversity can be useful for link-level restoration where, in case of one conduit cut (e.g., conduit-32), the paths can be locally restored by routing them through any available channels of another physically disjoint conduit (e.g., conduit-31).

The first set of restoration results (Figures 2, 3, 4) is obtained from the following simulation experiments. First, the network is loaded with 450 mesh restorable lightpaths. For each mesh restorable primary lightpath, a backup lightpath is also created. After all 450 primary and backup paths are routed, we fail conduit-32 (between Phoenix and Dallas) which carries 22 primary lightpaths through it. Average number of hops for these 22 primaries and their backups are 3.5 and 4.0 respectively. After the failure, StarNet restoration protocols for all 22 lightpaths kick in. StarNet Modeler generates performance reports in a real-time fashion as the restoration process for failed lightpaths progresses.

### ***StarNet-P Performance***

In the first experiment, we run only path-level restoration (StarNet-P) where, when a primary path fails, the corresponding backup path is chosen for migrating traffic from the broken primary path. Since StarNet allows capacity sharing among multiple backup paths, the StarNet-P restoration protocol requires end-to-end SONET-based signaling over the backup path before the traffic can be migrated. This signaling delay appears as the StarNet-P restoration latency to the lightpath end-users.

StarNet-P latency numbers for conduit-32 failure are shown in Figure 2. Average latency of 61 ms is well within the 200 ms design-target of Tellium's StarNet restoration protocols. The wide variance in StarNet-P restoration latencies are due to the backup hop-count variation and asymmetric processor loading at different nodes on the backup paths. Since StarNet-P requires end-to-end signaling over backup paths, its latency is strongly correlated with the backup path hop-counts. It turns out that for the failed 22 primary paths in conduit-32, the backup hop-count ranges from 1 to 7. This hop-count variation is partially responsible for the wide variance of StarNet-P latency numbers. In addition, we also observed that depending on the topology of all 22 backup paths, few cross-connects carry more backup paths than the others. As a result, these nodes see a higher number of restoration messages being processed within their restoration processors. This asymmetric processor loading also contributes to the latency variance.

### ***StarNet-L Performance***

In the next experiment, we turn StarNet-L on so that the network first attempts to restore the paths by finding locally available channels within the same span but on physically diverge conduits. Paths for which StarNet-L fails, StarNet-P is attempted as described in the previous case. After conduit-32 is failed it turns out that 10 out of 22 lightpaths are restored locally by using available channels within conduit-31. The corresponding StarNet-L latency numbers are shown in Figure 3. Since for StarNet-L all the restoration signaling is restricted within the affected link, the latencies are generally smaller than the end-to-end StarNet-P. However, the restoration processing load on the link end-nodes (Phoenix and Dallas in this case) is quite bursty. Processor queuing because of this burstiness contributes to the latency numbers. However, unlike in the StarNet-P scenario, the restoration processing load for all lightpaths is concentrated to only two nodes. This explains the smaller variance of the StarNet-L restoration numbers.

### ***StarNet-P triggered by StarNet-L failure***

In the last experiment, StarNet-L succeeds only for 10 lightpaths. For other 12 failed lightpaths, StarNet-L fails because of spare channel unavailability within conduit-31. For these 12 paths, after StarNet-L fails, StarNet-P is invoked and their restoration eventually succeeds. StarNet-P restoration latencies for this situation are shown in Figure 4. Note that these numbers are substantially larger than those in Figure 2. The reason for this is that in addition to StarNet-P signaling latencies, these numbers also include the StarNet-L failure time for individual lightpaths. Note that if sufficient link level spare capacity is not available, StarNet-L can reduce the restoration latency for some of the paths but it may also introduce additional latency to the restoration for paths for which StarNet-L fails.

Based on the simulation experiments, we observe that the restoration latencies are not only sensitive to backup path hop-counts but also to the restoration processing load within the optical cross-connects. We also observed that the restoration limits are well within our target design upper bound of 200 ms for the path-level mesh restoration protocols.

### ***Sensitivity Analysis***

Two important parameters inside the OXC that affect the restoration latency are: (i) restoration protocol processing time and (ii) inter-processor communication time. We study the effect of increasing each of these parameters on the restoration latency. We consider the failure of 25 lightpaths, each of which have identical (at link level) 4-hop primary and 6-hop backup paths. The behavior is plotted in Figures 5 and 6 respectively. When one of the above two parameters is varied, the other is kept fixed at 1. With a restoration latency upper bound target of 200 ms, the corresponding cut-off values for protocol processing time and inter-processor communication time are 4ms and 16ms respectively

### Summary and Conclusions

This paper presents the performance of a span and an end-to-end restoration protocol framework. Based on the simulation experiments, we observe that the restoration latencies are not only sensitive to backup path hop-counts but also to the restoration processing load and the inter-processor communication delay within the OXC. With higher number of simultaneously restored paths, the protocol processing time for individual lighpaths increases and that, in turn, increases the overall restoration latency. For the experimental US backbone network, we observed that the end-to-end mesh restoration latencies are well within the target design upper bound of 200 ms.

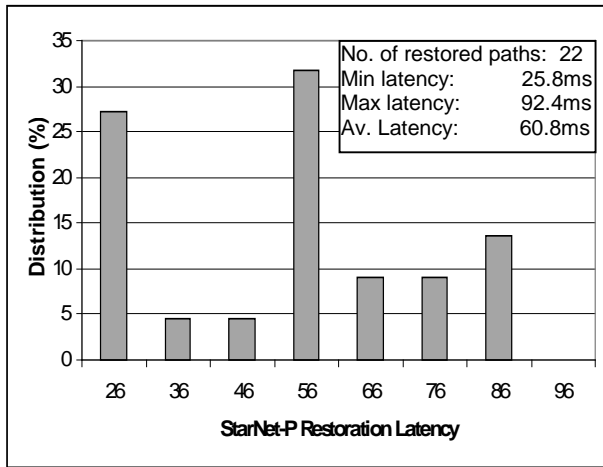


Figure 2: StarNet-P restoration latency

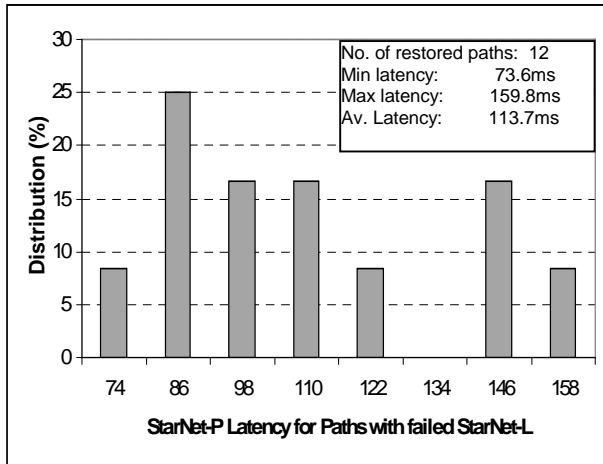


Figure 3: StarNet-L restoration latency

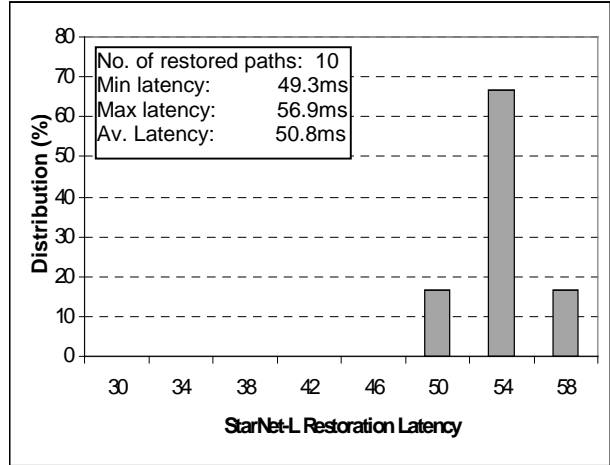


Figure 4: StarNet-P restoration latency for paths with StarNet-L failed

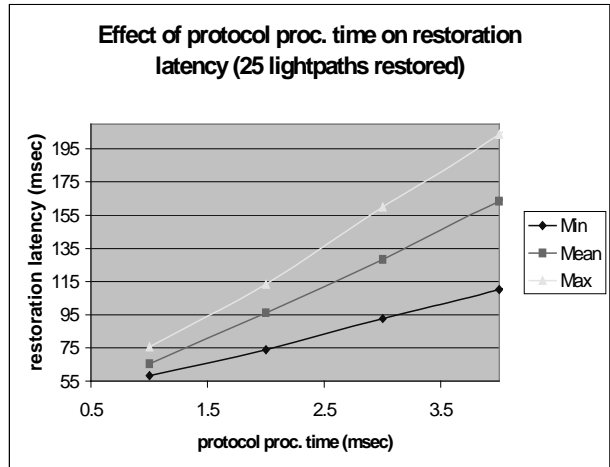


Figure 5: Sensitivity analysis for protocol processing time

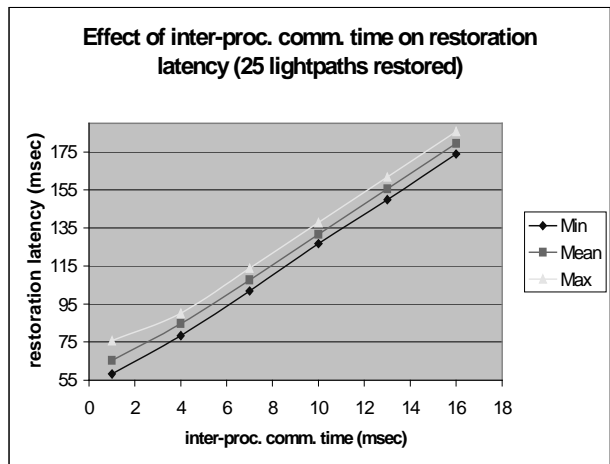


Figure 6: Sensitivity analysis for inter-processor communication time