Benefits of limited context awareness in stateless PCE

O. Gonzalez de Dios, F.J Jimenez Chico, F. Muñoz del Nuevo
Telefónica I+D, C/EmilioVargas 6, Madrid
{ogondio, fjjc, fmn}@tid.es

Abstract: A temporary resource reservation mechanism can enhance the performance of a stateless PCE in situations when the TED might not be up-to-date. This paper examines the benefits of such a mechanism by means of simulation.

OCIS codes: (060.4251) Networks, assignment and routing algorithms; (060.4261) Networks, protection and restoration

1. Introduction

The Path Computation Element (PCE) provides path computation functions in support of traffic engineering in Multi-Protocol Label Switching (MPLS) and Generalized MPLS (GMPLS) networks. According to RFC4655 [1], a PCE can be either stateful or stateless. In the former case, there is a strict synchronization between the PCE and not only the network states (in term of topology and resource information), but also the set of computed paths and reserved resources in use in the network.

In other words, the stateful PCE utilizes information from the TED as well as information about existing LSPs in the network when processing new requests. However, the maintenance and synchronization of a stateful database can be non-trivial, not only because it should verify the actual establishment of the computed paths, but also because it might not be the unique element to compute paths. Moreover, maintaining such a stateful database does not seem to be a function of the PCE, but rather of an NMS.

On the other hand, a stateless PCE does not keep track of any computed path, and each set of request(s) is processed independently of each other. A stateless PCEs may eventually compute paths based outdated network status information if path computation request arrive before the TED has converged. RFC 4655 suggests that a limited form of statefulness might be applied within an otherwise stateless PCE. The PCE may retain some context from paths it has recently computed so that it avoids suggesting the use of the same resources for other TE LSPs.

In line with this idea, this paper proposes a procedure for PCE operation in which the PCE blocks the computed resources in a path request for subsequent requests for a certain time, and it examines the benefits of such a mechanism by means of simulation.

2. Challenges of a stateless PCE-based massive restoration

One of the most challenging scenarios for a PCE-based architecture is the one of massive restoration. In the event of a network failure affecting a high number of LSPs (e.g. a fiber cut), a PCE could potentially receive a significant amount of restoration requests in a short period of time. One of the various challenges in this scenario is the fact that the PCE needs to sequentially perform multiple independent path computations. In this scenario, a stateless PCE would rely on TED information, which could eventually be up-to-date before the first incoming request (e.g. in case the routing algorithm has disseminated the failure event), but will definitely be outdated for subsequent requests. Note that the PCE TED would only be updated after the LSP has been established and its update advertised. Moreover, the establishment of the LSPs can take longer than the set of computations, e.g. in case of lightpaths.

Thus, the main issue is that the TED information in which the stateless PCE relied is either outdated, because there has been no time for the TED update to arrive, or incomplete, because the LSPs have not yet finished to set up. As a result, it might be expected that subsequent path computations would rely on the same nodes, TE-links or even labels assigned to previous computations. Therefore, there is a significant chance at the signaling phase that multiple connection requests are contending for the same resources.

After the eventual failure in the establishment of some of the connections, subsequent re-tries of path computation requests to the PCE would be triggered. After a number of loops, the PCE-based restoration would be eventually solved, but the potential number of retries could be significantly high. In this context, the availability of a limited context aware PCE [2] could potentially solve the issue in a graceful fashion. Each of the restoration path requests could include Resource Blocking information, which will state the kind of resources and the amount of time they should be blocked. The PCE would assume that the LSP is going to be established soon, and then temporarily would 'mark' the resources as blocked, so as not to consider them in subsequent connection requests and thus avoiding the contention at the signaling phase.

3. Simulation model

In order to evaluate the potential benefits of the proposed mechanism, a simulation study in a WSON network has been performed using Omnet++[3]. The well-known 14-node NSFNet backone was used a reference topology for the simulation scenario (see Fig. 1). In the scenario, one LSP (lightpaht) is established between every pair of nodes. For routing and wavelength assignment, a typical Shortest Path plus First Fit approach was followed. The control plane communication network was emulated by introducing delays for control plane messages between neighboring nodes. For simplicity, it was considered a similar control plane delay for messages between ROADMs, and that every node had a similar delay to the PCE as well as.

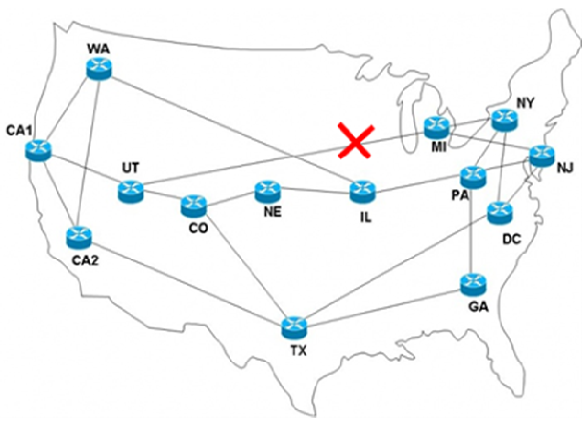

Fig 1. NSFNet Network Scenario.

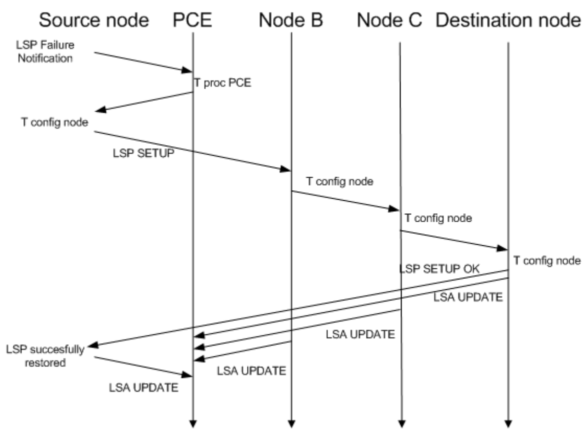

Fig 2. Control Plane Messages Workflow.

The simulation is triggered by a link cut affecting several LSPs. 6 different link failures have been considered: CA1-UT;WA-IL;CO-TX;IL-PA;UT-MI;TX-DC. After the link failure, the source node of each of the affected LSPs will independently trigger a Path Computation Request to the PCE demanding for an alternative path. The total time for path restoration will include the round trip delay to the PCE (Tpce=10ms), the time to compute the path (Tproc_PCE=50ms), the time to configure each ROADM in the path (Tsetup_roadm, which varies from 10ms to 200ms) plus the time spent for the interchange and process of control messages (Tmsg=1ms). For the sake of simplicity, the TED is automatically updated after LSP setup confirmation.

4. Analysis of the simulation results

First of all, it is analyzed the impact of the resource reservation timer (RRT). Fig. 3 shows the blocking ratio of the establishment of the restored LSPs for different values of the RRT (ranging from 0 to 1 second) without retries. It can be observed that the blocking ratio (i.e. signaling fails in the first attempt) depends on the relation between the LSP setup time and the reservation time. On one side, the higher the setup time, the higher blocking ratio. On the other side, the higher the reservation time, the lower the restoration blocking ratio. In the simulated scenario, 1 second RRT timer is enough to avoid any potential collision.

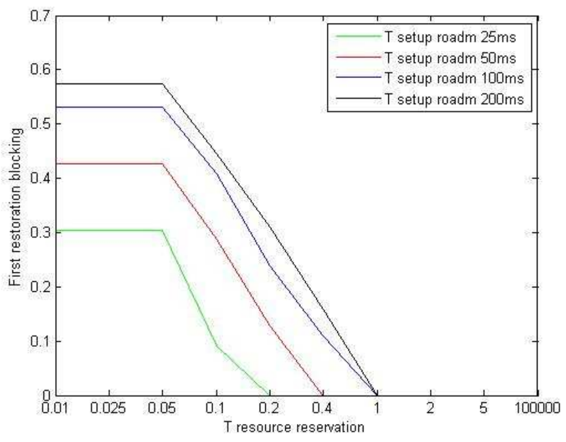

Fig 3. Blocking Ratio vs Resource Reservation Timer (without retry).

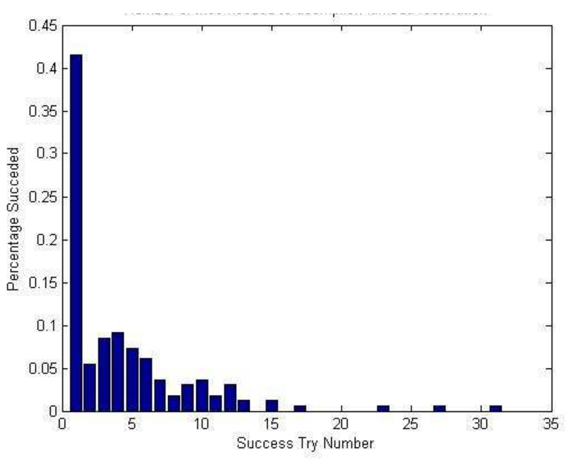

Fig. 4: Number of attempts needed to accomplish restoration for the different connections.

Fig. 4 shows the number of attempts needed to accomplish the restoration for the different connections, fixing Tsetup_roadm to 400ms and assuming and immediate retry. It can be perceived that there is a significant amount of colliding path computation requests (around 60%), and that there are a significant number of requests with a very high number of retries. In other words, the picture measures the number of retries needed for different connections, and illustrates the ineffectiveness of a non context aware massive recovery. The number of retries could be reduced by means of setting up the proper retry timers.

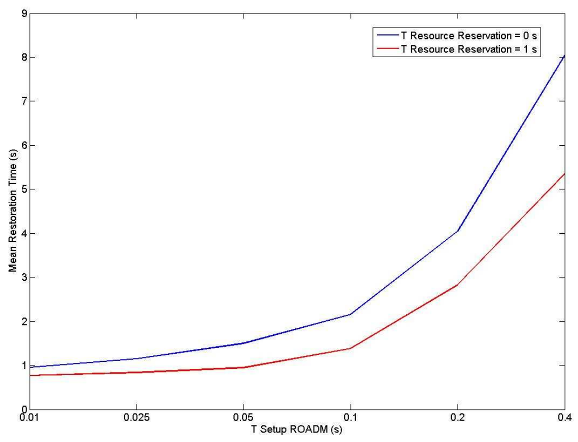

Fig. 5: Average Restoration Time needed to restore an LSPs with and without the "resource reservation period".

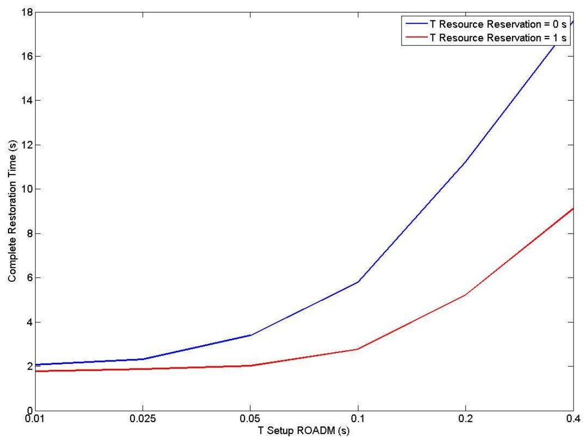

Fig. 6: Total time required to accomplish the recovery of each single connection affected by the failure, with and without the "resource reservation period" mechanism.

Fig. 5 compares the mean restoration time of and LSP needed with a RRT of 1 second with the reference case in which there is no such a mechanism. It can be observed that keeping context information for just one second may reduce by a 30% the average recovery time per connection. Moreover, Fig. 6 shows that the use of the mentioned resource reservation period can have an even more significant impact in the total recovery time. In this case, the global restoration time (i.e. the time it takes for the last LSP to be recovered) can be reduced by a 50% when the ROADM setup time is higher than 100ms. On the other side, it must be noted that this time reduction is not so significant or even negligible if the connection setup time is very low in comparison to the LSP computing time.

5. Conclusions

During the last years, there has been significant debate about the amount of information that the PCE must retain, i.e. the stateful vs. stateless PCE debate. While the stateless PCE seems to be preferred in IETF standards, a limited form of statefulness could be useful to improve PCE functionality in situations in which the local Traffic Engineering Database (TED) might not be up to date, such as a massive restoration due to a node or link failure. Alternatively, the PCE can retain some context from the resources assigned to Path Requests during a certain period of time, so that it avoids suggesting the use of the same resources for subsequent LSPs.

The simulation results show that keeping context information for just one second can reduce by a 30% the average recovery time per connection, as well as a 50% of reduction in the total recovery time, when the ROADM setup time is higher than 100ms. The benefit comes from the fact that it avoids the computation of a significant amount of colliding path requests. It must be noted that these benefits would be negligible if the path setup time is low enough.

6. Acknowledgements

This work has been partly funded by the European Union through the project STRONGEST (FP7-ICT-247674).

7. References

[1] Farrel, A., Vasseur, J.P., and Ash, G., "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.
[2] Gonzalez de Dios, O., Jimenez Chico, J., "PCEP Extensions to support limited context awareness in stateless PCE", draft-gonzalezdedios-pce-context-awareness-00, October 2010.
[3] Omnet ++ 4.0 www.omentpp.org