Reconfigurable Optical Networks: a Cross Layer Approach

Reza Roshani, Paolo Monti, Marco Tacca, and Andrea Fumagalli

OpNeAR Lab, Erik Jonsson School of Engineering and Computer Science

The University of Texas at Dallas, Richardson, TX 75080, USA

{rxr058100, paolo, mtacca, andreaf }@utdallas.edu

Abstract

This paper presents the Pruning with Memory (PWM) algorithm, which computes a cross layer optimal reconfiguration sequence in reconfigurable IP/MPLS over optical networks.

.

Introduction

Reconfigurable optical networks [1] may yield tangible advantages, in terms of network utilization, when the offered traffic at the upper layers (e.g., IP/MPLS) fluctuates over time. Reconfiguration of the optical layer requires intermediate steps, as it is possible to set up/tear down only a limited number of lightpaths concurrently. The work in [3] presents an algorithm to identify the minimum number of lightpath moves to transition from one configuration step to the next, but no sequence is provided, i.e., all sequences are equivalent. However, in an IP/MPLS over optical network, different sequences of intermediate steps impact differently the IP/MPLS layer. During reconfiguration, LSP's must adapt to the underlying logical topology changes, i.e., LSP's traffic may be rerouted, buffered or, in some cases, (partially or totally) *disrupted.* Identifying the optimal sequence of intermediate steps and LSP's rerouting to transition the network from one state to the next, becomes a cross layer key objective in order to determine the success of reconfigurable optical networks.

This paper presents the Pruning with Memory (PWM) algorithm, a cross layer solution that identifies the optimal reconfiguration sequence while minimizing disruption at the IP/MPLS layer. The set of possible reconfiguration sequences S_R with the best number of moves is obtained from the permutations of the elements in the set identified in [2]. PWM identifies the optimal reconfiguration sequence by intelligently pruning set S_R based on two factors: (*a*) the minimization of metrics of interest, e.g., average hop count and loss at the current reconfiguration step, and (*b*) the effect of previous reconfiguration steps.

The proposed PWM algorithm is then compared to the MAPF algorithm [2]. The MAPF algorithm is a greedy algorithm that determines the optimal reconfiguration sequence irrespective of the loss of traffic at the IP/MPLS layer. Numerical results show that the presented approach reports gains, in terms of bandwidth loss, as high as 50%.

Network Model

Consider a network where each node comprises both an IP/MPLS router and an optical cross-connect (OXC). OXC's are used to establish lightpaths, i.e., provide a virtual topology to the routers. The following assumptions are made: (*i*) the number of wavelengths is unbounded, (*ii*) traffic is symmetric, i.e, for every LSP from node *s* to node *d* there is another LSP with the same bandwidth requirement routed in the reverse direction, (*iii*) LSP's may be split and carried over multiple paths on the virtual topology. Two LSP traffic matrices are considered, Λ_{in} and Λ_{fin}. The *initial* virtual topology $G_{in}(N,A_{in})$ is the virtual topology resulting from the traffic matrix Λ_{in}, while the *final* virtual topology $G_{fin}(N,A_{fin})$ is the virtual topology resulting from the traffic matrix Λ_{fin}. N is the set of routers in the network, while A_{in} and A_{fin} are the set of lightpaths connecting the routers in the *initial* and *final* virtual topology, respectively. $G_{in}(N,A_{in})$ and $G_{fin}(N,A_{fin})$ are computed using the procedure presented in [1].

Let $S=\{St_0,St_1,\ldots,St_M\}$ be a reconfiguration sequence able to transform $G_{in}(N,A_{in})$ into $G_{fin}(N,A_{fin})$ in M reconfiguration steps. Let $G_0 = G_{in}(N,A_{in})$ and $G_M = G_{fin}(N,Af_{in})$. During a reconfiguration step (St_n), virtual topology G_{n-1} is transformed into virtual topology G_n by tearing down and/or setting up a limited number of lightpaths. The execution of each step requires a time T. The execution of the reconfiguration sequence up to step St_n requires a time $T_n = nT$. Routing of LSP's in G_n is computed using the model presented in [1] modified to minimize the LSP traffic loss (if any) as follows. First a cost function is introduced. The cost function is defined as:

$$Loss = \min BL, \quad BL = \sum_{i,j} \left(\lambda_{(i,j)}^{(n)} - k_{(i,j)}^{(n)} \cdot \lambda_{(i,j)}^{(n)} \right)$$

where the *bandwidth loss* (BL) measures the total traffic disruption in the network due to lack of resources. $\lambda_{(i,j)}^{(n)}$ is the bandwidth requirement of an LSP from node *i* and node *j* at time instance (*n*), and $k_{(i,j)}^{(n)} \cdot \lambda_{(i,j)}^{(n)}$ (with $k_{(i,j)}^{(n)} \leq 1$) is the portion of the bandwidth of the LSP that is actually routed. In all the constraints of the flow model defined in [1] $\lambda_{(i,j)}^{(n)}$ is replaced by $k_{(i,j)}^{(n)} \cdot \lambda_{(i,j)}^{(n)}$.

The Pruning with Memory (PWM) Algorithm

$G_{in}(N,A_{in})$ and $G_{fin}(N,Af_{in})$ are computed using the method explained in the previous section. The set of all possible reconfiguration sequences (S_R) able to transform G_{in} into G_{fin} with the minimum number of reconfiguration steps (M) is precomputed using the algorithm presented in [3].

The objective of the PWM algorithm is to find one reconfiguration sequence S_{min} in set S_R which is able to minimize the value of the bandwidth loss ($avgBL_{min}$). S_{min} is found by pruning S_R as explained by the pseudo code in Figure 1. Notice how the PWM algorithm, using the *margin* parameter, takes into account the effect of previous reconfiguration steps.

```
begin PWM{
    rsc = 0   //reconfiguration steps counter
    for j = 1 to M {
        for each Sᵢ ∈ S_R {
            avgAHCᵢ,ⱼ = 1/Tⱼ ∫₀^Tⱼ AHCᵢ(t) dt
            avgBLᵢ,ⱼ   = 1/Tⱼ ∫₀^Tⱼ BLᵢ(t) dt
            rcs + +
        } end for
        best_avgAHCⱼ = min_{Sᵢ∈S_R} ( avgAHCᵢ,ⱼ )
        i_min = i ; avgAHCᵢ,ⱼ = best_avgAHCⱼ
        best_BLⱼ = BLᵢ_min,ⱼ
        for each Sᵢ ∈ S_R {
            if(|avgAHCᵢ,ⱼ − best_avgAHCⱼ| <
            margin · best_avgAHCⱼ && BLᵢ,ⱼ > best_BLⱼ)
                S_R = S_R − {Sᵢ}
        } end for
    } end for
    select S_min such that avgBL_min = min_{Sᵢ∈S_R}(avgBLᵢ,M)
} end PWM
```

Figure 1: Pseudo code of the PWM algorithm

Simulation Results

This section compares the PWM and the MAPF algorithm in terms of *avgBL_{min}*. Numerical results are obtained for a network of 10 nodes. Each node has three transmitters and three receivers. The lightpath capacity is C = 10 Gbps. Transmission impairments and traffic fluctuations are modelled using set V_i and parameters f_p and f_t as explained in [1]. Λ_{in} and Λ_{fin} are assigned the same values used in [1]. The *margin* parameter is set equal to 1. Presented results are averages of 100 experiments. For each experiment Λ_{in}, Λ_{fin}, f_p and f_t are left unchanged. Only the neighboring set V_i is randomly generated for each node *i*.

Figure 2 shows the value of *avgBL_{min}* as a function of the LSP's traffic fluctuations. Results confirm the

earlier claim that considering the effect of the previous configurations steps has a beneficial effect on the traffic disruption at upper layers. The additional complexity required by the PWM algorithm is shown in Figure 3, where the value of the reconfiguration steps counter (*rsc*) (Figure 1) is presented as a function of the LSP's traffic fluctuations.
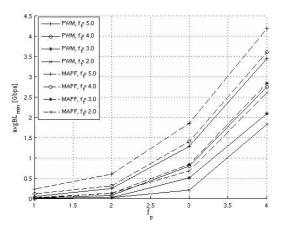

Figure 2: average bandwidth loss (avgBLmin) as a function of the LSP's traffic fluctuations
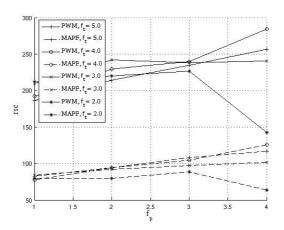

Figure 3: reconfiguration steps counter (rsc) as a function of the LPS's traffic fluctuations

Conclusions

This paper presented the PWM algorithm, a cross layer solution that reconfigures the optical layer while minimizing traffic disruption at the IP/MPLS layer. Numerical results show that the presented approach is able to reduce the traffic disruptions by up to 50%.

References

1. R. Roshani, et al. OFC2008, OTuA2

2. Y. Zhang, et al. Journal of Lightwave Technology, Volume 23 (2005), 2854-2867

3. J. F. P. Labourette, et al. IEEE Trans. Commun., Volume 42, no.10 (1994), 2822-2832