Traffic Re-Optimization Strategies for Dynamically Provisioned WDM Networks

Jawwad Ahmed[1], Fernando Solano[2], Paolo Monti[1], Lena Wosinska[1]
[1]Royal Institute of Technology (KTH), Stockholm, Sweden, Email: {jawwad, pmonti, wosinska}@kth.se[1]
[2]Warsaw University of Technology, Warsaw, Poland, Email: fs@tele.pw.edu.pl[2]

***Abstract*—In Wavelength Division Multiplexed (WDM) networks with dynamic lightpath provisioning, connection requests are served without any prior knowledge of their arrival and departure times. As time passes, network resources may become *fragmented* because of the network dynamism. Under these circumstances it is highly beneficial to *re-optimize* (i.e., de-fragment) the existing lightpath configuration at some specific time instances to improve the network resource utilization and reduce the risk that future connection requests will be blocked.**

Assuming that this de-fragmentation process occurs during a *re-optimization phase*, this paper presents a set of strategies which govern the time instances *when* this re-optimization phase should be triggered as well as a set of strategies to decide *which* of the currently active lightpaths should be optimized at any given re-optimization phase. These strategies are referred to as *when-to-re-optimize* (when-t-r) and *what-to-re-optimize* (what-t-r) strategies, respectively.

During the evaluation process particular attention is devoted to study the impact that when-t-r and what-t-r strategies have on the traffic disruption metrics (i.e., number of total disrupted connections, disruption time, reconfiguration time) inherent with the re-optimization process. Based on the evaluation results, it can be concluded that the choice of an optimal "when" and an optimal "what" to re-optimize strategy is dependent upon the performance objective (e.g. lower blocking probability or network disruption) in a given network scenario.

I. INTRODUCTION

In a Wavelength Division Multiplexed (WDM) network with dynamic lightpath provisioning, connection request arrivals and departures are stochastic in nature and provisioning is performed via online Routing and Wavelength Assignment (RWA) algorithms. However, after some time network resources may reach a *fragmented* state. In other words, some already provisioned lightpaths may become sub-optimal with respect to the existing state of the network. Therefore it may be beneficial to "re-think" the existing lightpath configuration so as to better utilize the available network resources. In this way blocking of the future connection requests may be reduced. This operation is referred to as network *re-optimization*.

The objective of the re-optimization process is to provide a new RWA solution for all (or a subset) of the currently active lightpaths in order to allocate network resources more efficiently. The problem of migrating traffic, from the currently-active *existing* lightpath configuration to the *new* one, is referred to as *traffic migration* or *traffic reconfiguration* problem. While solving the traffic migration problem, the performance metrics commonly considered are related to the traffic disruption while reconfiguring the lightpath topology (e.g. total number of disrupted connections, maximum number of simultaneously disrupted connections, and network reconfiguration time).

There is a substantial amount of work done on re-optimization in WDM networks. Re-optimization schemes can be divided in *reactive* and *proactive* approaches. In a reactive approach, re-optimization is initiated only when a certain performance metric (e.g., blocking probability, wavelength usage,) degrades below a specific threshold. On the other hand proactive approaches try to keep the network in an optimal state by re-optimizing the use of network resources at some predefined time instances or at periodic intervals. A proactive rerouting strategy is evaluated in [1] where re-routing is done when a lightpath connection request is due for departure. In [2] two reactive schemes are proposed: one for WDM networks with full, and the other for WDM networks with no wavelength conversion.

Re-optimization algorithms can also be classified based on their objectives during the re-optimization phase (e.g., minimum lightpath hop count, minimum wavelength usage). A detailed summary of different metrics and algorithms can be found in [3]. Most of the existing re-optimization strategies consider solely the problem of finding the lightpaths that must be reconfigured, while the work in [4] defines some re-optimization strategies which address the problem of how often to re-optimize. However, despite this extensive work, very little attention is paid to the effect that these re-optimization strategies have on the traffic disruption inherent in the traffic migration phase.

This paper presents and evaluates a set of re-optimization strategies, which govern the exact *time* and *frequency* of the re-optimization phase. These strategies are called *when-to-re-optimize (when-t-r)* strategies. The other set of policies consi-

This work was supported by the Network of Excellence "Building the Future Optical Network in Europe" (BONE), funded by the European Commission through the 7th ICT-Framework Programme.

dered in the paper is related to *how many* and *which* lightpaths, currently active in the network, should be re-optimized at any given re-optimization phase. These strategies are called *what-to-re-optimize (what-t-r)* strategies. The importance of choosing a good *what-t-r* strategy is based on the intuition that, in some circumstances, it might be sufficient to optimize only a small subset of the currently active lightpaths to achieve most of the optimization benefits. This is in contrast to the extreme case in which all active lightpaths in the network are re-optimized (i.e., *greenfield* re-optimization). Another aspect to consider is the minimization of disruptions during the traffic migration process where some connections might need to be interrupted. By making an appropriate choice of the active lightpaths to be re-optimized, an efficient *what-t-r* strategy may have a beneficial effect on the minimization of both the traffic disruptions and the network reconfiguration time.

In this paper both *when-t-r* and *what-t-r* strategies are evaluated in a variety of network scenarios to explore their pros and cons under dynamic network conditions. Particular attention is also devoted to their effect on the reconfiguration performance metrics. The paper outline is as follows: in Section II the problem of traffic migration is introduced along with the definition of two important traffic migration objectives to be considered in the performance evaluation section. Section III presents a series of different *when-t-r* and *what-t-r* strategies. Performance evaluation is shown in section IV, where the strategies presented in Section III are compared in terms of network performance and traffic disruption related metrics. Finally, conclusions are drawn in Section V.

II. Traffic Migration Problem

The traffic migration problem arises when a new lightpath configuration computed during a re-optimization phase requires resources (e.g., wavelengths or wavelength converters) that are already allocated to some other existing lightpaths. As a result, these new lightpaths cannot be setup before some of the existing lightpaths are torn down. However, in order to minimize traffic disruption, the existing lightpaths should not be terminated before the new lightpaths are up, according to the make-before-break policy. At this point, the reconfiguration process is said to be in a deadlock state. In other words there are cyclic dependencies between the existing lightpath configuration and the new one that need to be resolved. These dependencies can be represented using a dependency graph [5]. In order to assess the complexity of the induced dependency graph, the concept of complexity ratio is introduced and defined as the ratio between the number of vertices in the dominant Strongly Connected Component[1] (SCC) of the dependency graph and the total number of currently active connections. This quantity gives an approximate measure on how strong the inter-dependencies are in the considered reconfiguration problem.

An important point to note is that a migration from the existing to the new lightpath configuration can be performed in polynomial time without traffic disruptions if the induced dependency graph is acyclic [6]. If the graph is cyclic then a subset of connections must be disrupted in order to migrate to the new configuration. Two important objective functions for the traffic migration are presented next. For a more exhaustive treatment of the traffic migration problem the interested reader is referred to [5].

A. Minimizing the total number of disrupted connections (MFVS -Minimum Feedback Vertex Set)

If the dependency graph is cyclic, then the problem of finding the minimum number of disrupted connections, while migrating the lightpath topology configuration, corresponds to solving of the Minimum Feedback Vertex Set (MFVS) problem [6]. In MFVS, given a dependency graph the objective is to find a set of vertices in the graph with the smallest cardinality whose removal will result in the dependency graph to be acyclic. This problem is NP-hard and is also known to be APX-complete [7]. Connections in the solution will need to be disrupted to perform the traffic migration from the existing to the new lightpath configuration.

B. Minimizing the maximum number of simultaneously disrupted connections (MMD)

The MMD problem may be modeled using the same dependency graph used to solve the MFVS problem. The objective of the MMD problem is to minimize the total number of simultaneous connections that need to be disrupted at any given point in time during the whole migration process. The solution of the MMD problem not only provides the answer to the question of *which* lightpaths need to be disrupted, but also the *order* in which the lightpaths need to be released.

III. Traffic Re-Optimization Strategies

Traffic re-optimization strategies can be divided into two main categories: 1) *when-t-r* strategies and 2) and *what-t-r* strategies. *When-t-r* strategies are used to decide *when* to trigger the re-optimization process and also to decide how often (i.e., the *frequency*) the lightpath configuration needs to be re-optimized. *What-t-r* strategies are used to decide *how many* and *which* lightpaths should be re-optimized at any given re-optimization phase. The relation between traffic migration and re-optimization strategies in the context of a dynamically provisioned network is shown in Fig. 1. In the remainder of this section a series of when-t-r and what-t-r strategies are presented.

[1]A directed graph is called strongly connected if there is a path from each vertex in the graph to every other vertex.

A. When to Re-optimize (when-t-r) Strategies

As already explained in the Introduction, when-t-r strategies can be sub-divided in to *proactive* and *reactive* approaches. The following two strategies are considered for evaluation.

EveryXRequestBlocked: this is a reactive strategy triggered when the number of blocked connection requests, after the last re-optimization phase, exceeds a certain threshold. Note that, since the threshold is based on the number of connection requests that are blocked, when the blocking probability increases (i.e., for increasing values of the network load) so does the frequency of the re-optimization phase. In case of no blocking, re-optimization is not triggered.

EveryXRequestsDepart: in this proactive strategy, the idea is to trigger the re-optimization phase only when a certain number of connections (i.e., X) have departed from the network. When a connection departs from the network some resources are released and they can be potentially used to optimize the existing lightpath configuration in the network. Ideally the value for X should be 1 so that re-optimization takes place every time a connection departs, but this is not practical because it will cause a significant overhead, i.e., the re-optimization phase will be triggered too often. As a consequence the network will experience more traffic disruptions in addition to a significantly higher computation times, due to the large number of reconfigurations.

B. What to Re-optimize (what-t-r) Strategies

The objective of all the strategies in this category is to choose the X lightpaths (LPs), from the set of currently active lightpaths $LP_{ActCons}$ in the network that will have to go through the reconfiguration process. Here $X = \lceil \alpha * LP_{ActCons} \rceil$ and α is a parameter with values between 0 and 1. Note that a high value of X will likely reduce the blocking probability and/or the network utilization, but may increase traffic disruption.

The following notations will be used to formally define the what-t-r strategies. Let $G(V,E)$ represent the directed symmetric graph corresponding to the current network topology, where V represents the set of vertices, and E represents the set of edges in the network respectively. Let R_i represents the ith connection request and $L(R_i)$ the lightpath on which Ri is established. Each $L(R_i)$ comprises n links namely $l_1, l_2 \cdots l_n$ where $li \in E$. The following two strategies are evaluated in this category.

XLPsOnMostCongestLinks: the objective of this strategy is to reduce the number of lightpaths that are on the most congested links and consequently reduce the network congestion. The strategy works iteratively, where, at each iteration, first the most *congested* link is identified (ties are broken with a random selection), then, based on a specific cost function, a lightpaths is selected for re-optimization. This process continues until X lightpaths are selected. The procedure works as follows. Let $load(li)$ represent the link congestion (i.e., the number of established lightpaths that passes through that link) for each link l_i. At each iteration, the most congested link is defined as: $l_{cong} = Max_{i \in E} load(li)$. Then for each lightpath passing through l_{cong}, $Congest(L(R_i))$ is computed as $\sum_{i=1}^{k} load(l_i)$, assuming that $L(R_i)$ comprises links $l_1, l_2 \cdots l_k$. The lightpath with the highest value of $Congest(L(R_i))$ is then selected.

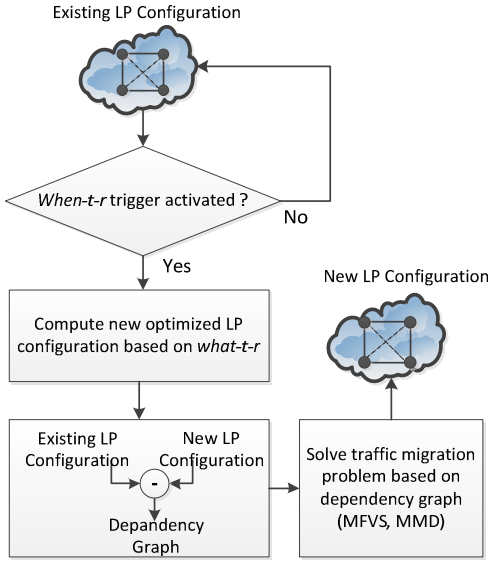

Fig. 1. Re-optimization process in a dynamically provisioned network.

XLPsCausingMostCongestion: this strategy selects and re-optimizes the lightpaths responsible for congestion in the network. The procedure works as follows. At each iteration, a set of links with the maximum congestion is identified (i.e., this is possible if more than one link experiences the same maximum congestion defined as $l_{cong} = Max_{i \in E} load(li)$. Then, the lightpath comprising the highest number of most congested links is selected. The same procedure is repeated until X lightpaths are identified. In a given iteration let's assume that S is a set with the most congested links belonging to E. Then, for each lightpath in the network, $Congest(L(R_i))$ is computed as:

$$Congest(L(R_i)) = \sum_{l_j \in S} Cont(L(R_i), l_j)$$
$$Cont(L(R_i), l_j) = \begin{cases} 1, & \text{if } L(R_i) \text{ contains link } l_j \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

A lightpath with the highest value of $Congest(L(R_i))$ is then selected.

IV. Performance Evaluation

For the evaluation part a custom-built Java based event-driven simulator was used. Results are collected for the European Core network topology with 14 nodes and 21 bidirectional (fiber) links. Each fiber is assumed to carry 10 wavelengths. Wavelength Continuity Constraint (WCC) is enforced in the network. The arrivals of the connection requests are assumed

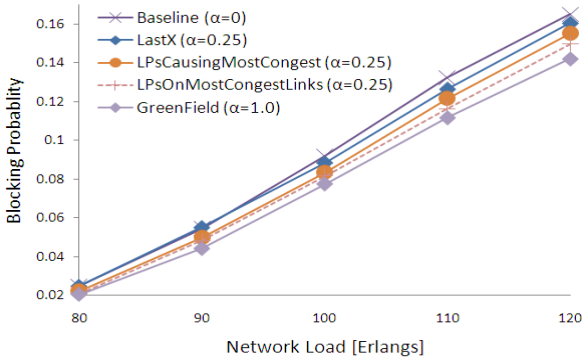

Fig. 2. Blocking probability performance for what-t-r strategies.

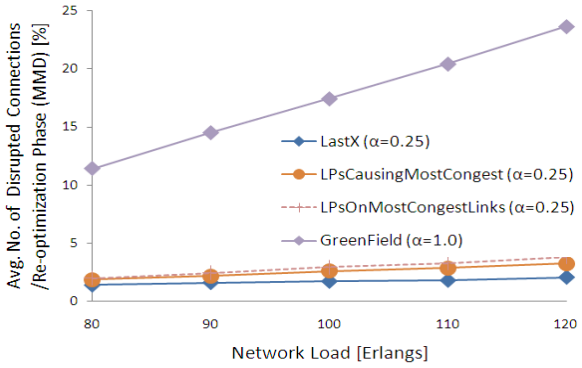

Fig. 3. Avg. disrupted connections / re-optimization phase (MMD).

to follow a Poisson distribution, while the service time is exponentially distributed.

The mean service time for the connection requests varies from 160 seconds to 240 seconds, to represent different load values. For online path provisioning, a constraint based routing (CBR) heuristic is used [8]. For the re-optimization phase a Simulated Allocation (SAL) [9] based meta-heuristic is used. The objective function of SAL is set to minimize the wavelength resource usage in the network. During each execution, SAL algorithm is set to terminate if no improvement is found in the objective function after 500 iterations.

For benchmarking purposes a simple *when-t-r* strategy is adopted from [4], which we refer to as *EveryXRequestsEnter*. This is a simple proactive strategy where the re-optimization phase is triggered after X numbers of connection requests have entered the system. Similarly a *what-t-r* strategy referred to as *LastX* [4] is also adopted for performance comparison. LastX works as follows. Let's assume that the last re-optimization phase occurred at time Tr and that the next re-optimization phase is scheduled to occur at time $Tr + 1$. Then the X candidate lightpaths will be selected that are established after time Tr. Lightpaths established before Tr will only be chosen if the number of lightpaths established in the time window $[Tr, Tr + 1]$ is smaller than X. This scheme is based on the intuition that at each re-optimization phase it is necessary to re-optimize only the lighpaths that arrived latest, since all other active lightpaths have already been re-optimized during some earlier phase.

Both *when-t-r* and *what-t-r* strategies are evaluated. For the performance evaluation we proceed as follows: first we investigate the *what-t-r* strategies (using a simple *when-t-r* strategy, i.e., EveryXRequestsEnter). Then utilizing the best performing *what-t-r* strategy we evaluate the *when-t-r* strategies. Results are provided for blocking probability and traffic migration related performance parameters: disruption time, reconfiguration time, number of total disrupted connections per re-optimization phase, complexity ratio for the generated dependency graph and an average total time spent in re-optimization phases (i.e. computation time for the re-optimization phase). In order to solve the MFVS problem an exact algorithm based on *branch and bound* [10] is used, while for the same set of input data the MMD problem is solved utilizing the algorithm presented in [5]. To compute the network disruption and reconfiguration times we assume that all the optical switches in the network have a reconfiguration time of 50 ms. The presented results are averaged over eight replications per experiment to ensure stable results. The simulation platform is a Red Hat Enterprise Linux workstation with 12 GB of memory and dual Intel Xeon CPUs (4 cores per CPU) clocked at 2.0 GHz.

A. What-to-re-optimize (what-t-r) Strategies

First we discuss the results for the *what-t-r* strategies. Initially their performance is evaluated with α = 0.25. Note that EveryXRequestsEnter is used as the when-t-r strategy with X =100 for all the results presented in this section. From the blocking probability results (Fig. 2) it is evident that for this value of α LPsOnMostCongestLinks is the best performing strategy. It can improve the blocking performance varying from 15% (at a load of 80 Erlangs) to 9% (at a load of 120 Erlangs) when compared against the *baseline* (no re-optimization) case. An important point to note is that blocking probability for the LPsOnMostCongestLinks is already very close to the greenfield case, so there is not much incentive to increase α beyond 0.25, which may also un-necessarily increase the network disruptions.

The average number of disrupted connections (expressed in percentage) for the MMD solution in each re-optimization phase can be seen in Fig.3. LastX shows the lowest number of total disrupted connections. This can be explained by observing that the later a connection request arrives in the network (before a specific re-optimization phase is triggered) the lower are the chances for the re-optimization algorithm to find a different and better path as compared to the existing one. Hence, the chances of causing any disruption by this connection request are also much lower. It is also clear that the disruption and reconfiguration times for the MMD case are much lower when compared to the MFVS case, as shown in the Fig. 4. Note that disruption and reconfiguration times for the MFVS case are more sensitive to higher loads and grow more rapidly with increasing loads than MMD.

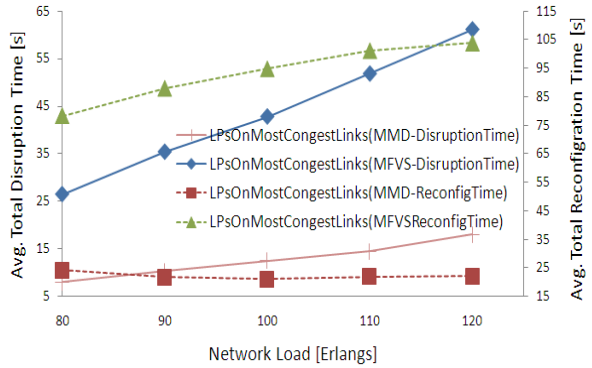

Fig. 4. Avg. total disruption/reconfiguration time for what-t-r strategies.

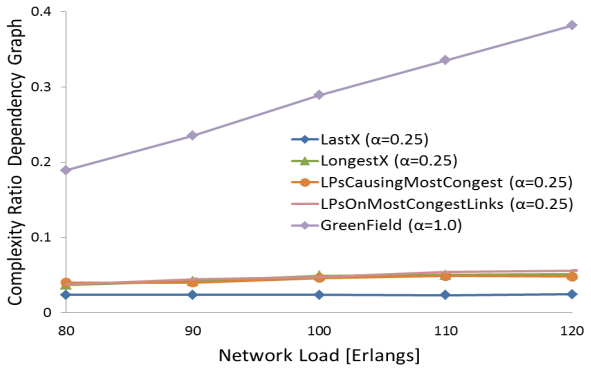

Fig. 5. Complexity ratio of dependency graph for what-t-r strategies.

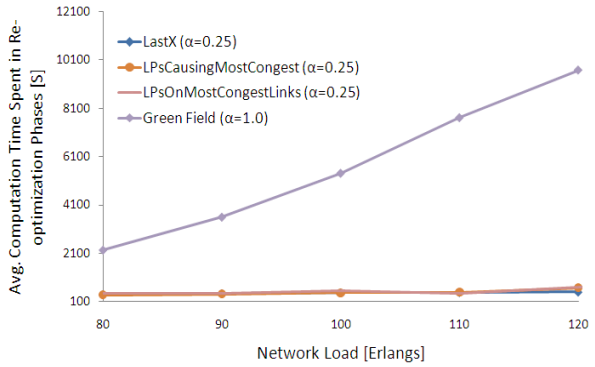

Fig. 6 Avg. computation time spent in re-optimization phases.

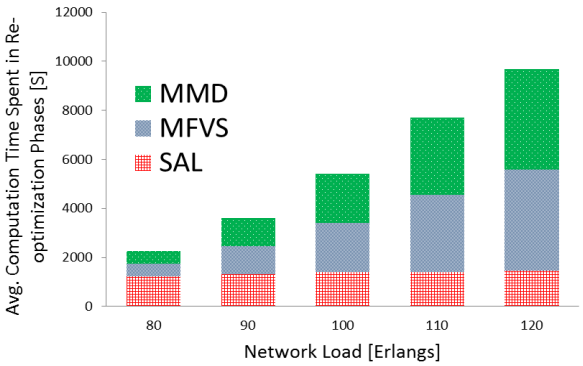

Fig. 7. Avg. computation time breakdown (Greenfield).

Fig. 5 shows results for the complexity ratio of the dependency graph for different *what-t-r* strategies. As expected, the dependency graphs generated for the greenfield re-optimization case are substantially more complex than the one generated for other *what-t-r* strategies. Furthermore, LastX leads to slightly less complex dependency graphs than other strategies here. Fig. 6 shows the computational overhead of re-optimization phases by the different strategies, which is almost the same for all the presented strategies and is not sensitive to the varying load values. However, greenfield case is significantly more computationally intensive, and its computation times grows rapidly with increasing load values. Fig.7 presents a breakdown of the time consumed by different modules of the re-optimization process namely: MMD, MFVS and SAL in the greenfield case. It is noteworthy that at lower loads the time consumed by SAL to perform re-optimization is dominant, while at the higher loads a large bulk of the time is consumed to solve the MMD and MFVS problems.

We also considered a scenario where $\alpha = 0.5$ (not shown), but we didn't observe any tangible performance gain. This confirms the earlier statement that a small value of α (i.e., 0.25) is sufficient to gain most of the re-optimization benefits using the LPsOnMostCongestLinks strategy. Such a small value of α will also have the added benefits of low traffic disruption and low computation time for the re-optimization phases.

B. *When-to-re-optimize (when-t-r) Strategies*

For evaluating *when-t-r* strategies we used LPSOnMostCongestLinks as the *what-t-r* strategy of choice with $\alpha = 0.25$ for all the presented results in this section. We observed a negligible blocking performance difference among all the strategies (Fig. 8). However, if we look at the network disruptions (Fig. 9) it is evident that disruption levels are somewhat higher for the EveryXRequestsBlocked. Furthermore, lowest levels of disruptions are experienced when the strategy EveryXRequestsDepart is employed. This result is motivated by the fact that when a request departs from the network, some resource are released and can be used during the re-optimization phase, leading to a lower probability of disrupting the lightpaths that are still in the network. In addition, the difference between EveryXRequestsDepart and EveryXRequestsEnter is not that significant. The complexity ratio results presented in the Fig. 10 show a similar behavior. The use of the EveryXRequestsBlocked strategy leads to the generation of more complex dependency graphs. From Fig. 11 it can be observed that with the EveryXRequestsBlocked strategy the computation time is dependent on the current load in the network. It increases rapidly with increasing load values. An explanation for this sensitivity to the load can be found in the reactive nature of this strategy. At higher loads, with an increasing blocking probability there is also an increase in the frequency of re-optimization phases, which in turn increases the computation time (time spent in re-optimization phases).

In summary if the objective is to have lower disruptions in the network, then EveryXRequestsDepart is a good strategy to deploy. Similar results were also observed in the MMD case.

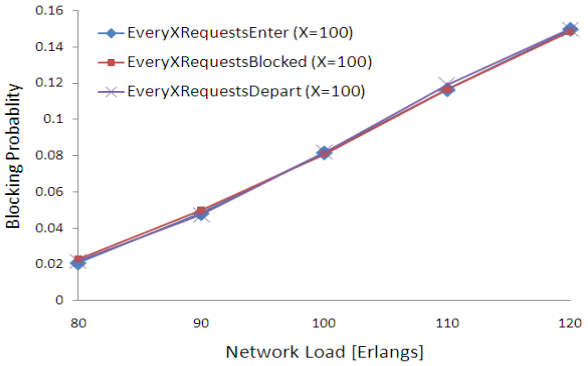

Fig. 8. Blocking probability performance for when-t-r strategies.

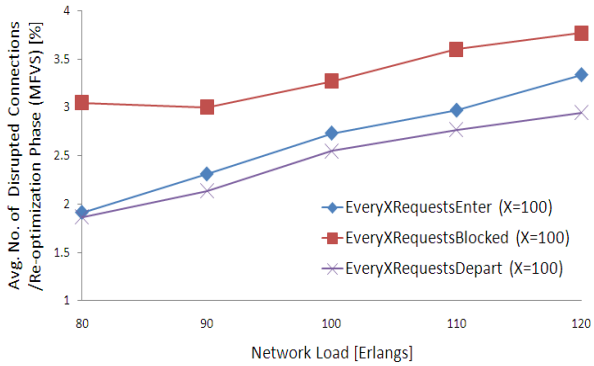

Fig. 9. Avg. disrupted connections / re-optimization phase (MFVS).

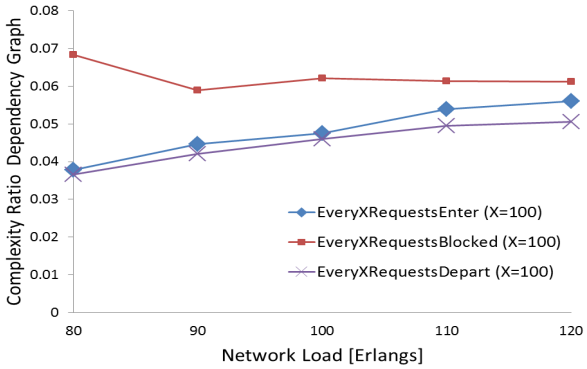

Fig. 10. Complexity ratio of dependency graph for when-t-r strategies.

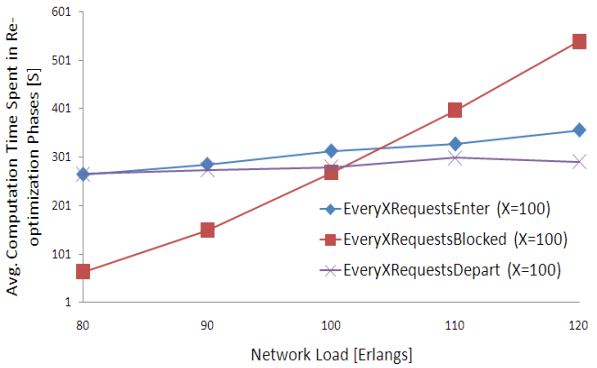

Fig. 11. Avg. computation time spent in re-optimization phases.

V. Conclusion

Network re-optimization is important for maintaining an efficient use of resources in a WDM network with dynamic lightpath provisioning. Online re-optimization is time consuming and may introduce network disruptions. Therefore, a careful decision should be made on *how often* and *when* to trigger the re-optimization phase (*when-t-r*), and on *how many* and *what* lightpaths should be re-optimized (*what-t-r*).

In this paper, we present a number of strategies and we evaluate them in a dynamic network traffic environment in terms of network and traffic disruption related performance parameters. Based on our results we conclude that in most of the scenarios it is sufficient to re-optimize only a small set of lightpaths instead of a complete greenfield re-optimization. This approach leads to significant reduction of both computational time and traffic disruptions. Based on our results, LPsOnMostCongestLinks is the most promising *what-t-r* strategy in terms of reducing blocking probability. Furthermore, in the considered cases it was sufficient to optimize only 25% of the currently active lightpaths in the network to achieve results very close to the *greenfield* re-optimization using this strategy.

For the *when-t-r* strategies, we observed similar levels of blocking probability. However EveryXRequestsDepart strategy introduces the minimum network disruptions. The choice of which *when-t-r* strategy clearly depends upon the objective for the network re-optimization phase. Interdependency between different *when-t-r* and *what-t-r* strategies, effect of different network topologies as well as the interplay between the connection service times and network arrival rate will be further explored in our future work.

References

[1] J. Zheng, B. Zhang, and H. T. Mouftah, “A departure-triggered rerouting mechanism for wavelength-routed WDM networks,” in Proceedings of International Symposium on Communication Systems, Networks and Digital Signal Processing, July 2002.

[2] X.-W. Chu, T. Bu, and X.-Y. Li, "A study of lightpath rerouting schemes in wavelength-routed WDM networks," in Proceedings of IEEE ICC 2007, June 2007.

[3] W.Golab and R.Boutaba, “Policy-driven automated reconfiguration for performance management in WDM optical networks,” IEEE Commun. Mag., Special Issue on Management of Optical Networks vol. 42, pp. 44–51, 2004.

[4] T. Kárász, "Consolidation strategies of provisioning-oriented optical networks," J. Opt. Netw. vol. 5, pp. 445-462, 2006

[5] F. Solano, “Analyzing two conflicting objectives of the WDM lightpath network reconfiguration problem”, in Proceedings of IEEE Globecom 2009, December 2009.

[6] N. Jose and A. Somani, “Connection rerouting/ reconfiguration”. In IEEE Design of Reliable Communication Networks (DRCN), pp. 23–30, October 2003.

[7] M.R. Garey and D.S. Johnson. “Computers and intractability: A guide to the theory of NP-Completeness (Series of Books in the Mathematical Sciences)”, W. H. Freeman, 1979.

[8] N. Boland, J. Dethridge, and I. Dumitrescu, “Accelerated label setting algorithms for the elementary resource constrained shortest path problem”, Oper.Res.Lett., vol. 34 no. 1, pp. 58-68, 2006.

[9] M. Pioro, “Simulation approach to the optimization of multicommodity integral flow networks”, In Proc. Of International Conference on Optimization and Simulation, 1997.

[10] H-M. Lin and J-Y. Jou, “Computing minimum feedback vertex sets by contraction operations and its applications on cad”, In IEEE Int. Conf. on Computer Design (ICCD), pp. 364–369, October1999.