12th GI/ITG CONFERENCE ON MEASURING, MODELLING AND EVALUATION OF COMPUTER AND COMMUNICATION SYSTEMS

3rd POLISH-GERMAN TELETRAFFIC SYMPOSIUM

IMPACT OF PARTIAL DEMAND INCREASE ON THE PERFORMANCE OF IP NETWORKS AND RE-OPTIMIZATION APPROACHES

Eueung Mulyana, Ulrich Killat
Department of Communication Networks, Hamburg University of Technology (TUHH)
Address: BA IIA, Denickestrasse 17, 21071 Hamburg, Germany
E-mail: {mulyana,killat}@tu-harburg.de

Abstract

An important aspect in management of IP networks is to control traffic traversing links on the network, while optimizing performance. Using offline approaches, this means that recomputation should be carried out on a periodical basis to adjust network configuration to the most actual traffic situation obtained from measurements or forecasts. In this work we investigate the effect of partial demand increase on the performance of the network and propose a simple policy scheme to decide whether re-optimization should be performed. Two re-optimization approaches based on plain local-search and simulated-annealing are presented. We apply our method for metric based traffic engineering scheme to the German scientific network (G-WiN) for which a traffic matrix and several traffic-increase patterns were randomly generated. Several computational results are provided.

Keywords
IGP routing, offline traffic engineering, metaheuristics, IP networks, re-optimization

1. INTRODUCTION

In recent years, many efforts have been invested to control and engineer IP traffic, due to rapid traffic growth and increasing requirements of service quality. An important aspect that triggers traffic engineering (TE) in IP networks is, that they were originally designed for robustness and reliability - if necessary at the cost of other performance measures. Generally, in the literature [3] TE is defined as mapping traffic flows onto the existing physical network topology in the most effective way to accomplish desired operational objectives. There are several approaches for deploying TE in current IP networks e.g. by optimizing the parameters used for routing decisions, so that a better network performance will be obtained [4-8,10-13], or by using explicit routing in an overlay model with ATM or

Frame Relay technology. In this work we limit ourselves to the metric based traffic engineering scheme for IP networks running an Interior Gateway Protocol (IGP) like OSPF (Open Shortest Path First) or IS-IS (Intermediate System to Intermediate System). In these networks, an administrative weight or metric is assigned to each link by network administrators and routing paths are defined as shortest paths according to these metric values. All demands between nodes in the network will then be routed on the corresponding shortest paths. It is obvious that in this routing scheme, the value of administrative weights plays a prominent role for controlling traffic. The objective of this work is firstly to investigate the impact of partial (non-linear) demand increase on the performance of IP networks. The second objective is to develop a policy when re-optimization should take place, since it is possibly not necessary to be performed if this partial demand increase does not result in significant performance degradation or if traffic engineering could not give better solutions due to e.g. network saturation, capacity limitation etc. Last but not least, if re-optimization is admitted, it is interesting to know, whether it is possible to obtain solutions with minimal changes compared to the original configuration, so that in this case partial demand increase will result in only partial an mainly local configuration changes. We develop two methods for re-optimization based on plain local-search and simulated-annealing approach, respectively. We apply our method to the German scientific network (G-WiN) [2] for which a traffic matrix and several traffic-increase patterns were randomly generated. The remainder of this paper is organized as follows. In Section 2, we present a mathematical model for OSPF routing and introduce some notations to describe and measure the effect of partial demand increase. A simple policy and two approaches for re-optimization are explained in Section 3. Finally, some investigations and computational results are presented.

2. OSPF ROUTING AND PARTIAL DEMAND INCREASE

OSPF Routing. In OSPF networks, each link is assigned a dimensionless metric value, also called cost or weight. Demands are routed along paths, which are selected using Dijkstra’s shortest path algorithm with respect to these link metric values. In the case of multiple shortest paths, some vendors have implemented a so-called ECMP rule (Equal Cost Multi-Path) [8, 9], so that the traffic flow will be split over those paths roughly evenly. This enhances routers’ capability for balancing the flows in the network e.g. to avoid congestion. However, some operators might want to avoid such a situation for management or other reasons [5, 13]. In this case, one might want to disable the splitting capability of the routers or to find a set of metric values that result in a networkwide *unique* shortest path routing. For illustration, consider two network settings in Figure 1. In the first configuration (Figure 1 (a)) each flow will be routed uniquely, fully independent of whether the ECMP feature is enabled or disabled. The second configuration (Figure 1 (b)) results in several ties, so that by enabling ECMP the flow from the node 1 to the node 6 will be split to the paths (1-2-4-6), (1-3-4-6) and (1-3-5-6) with the composition of traffic fraction of 50%, 25% and 25% respectively. In this work and for the following discussion we always assume that the ECMP is enabled. The methods for obtaining metric values for unique shortest path routing could be found e.g. in [4, 5,13].

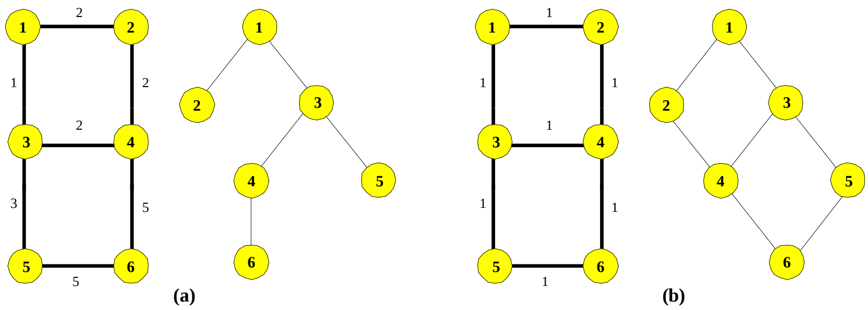

Figure 1 Shortest path structures seen from node 1 for the case: (a) unique, and (b)not-unique shortest path metric values

We will now formulate the problem. Given is a directed network $G=(N,A)$, where N is the set of nodes representing the network's routers and A is the set of arcs representing the network's links. Each link $(i,j)\in A$ has a capacity $c_{i,j}$. Furthermore, we have a demand $f^{u,v}$ for each pair $(u,v)\in N\times N$, giving the demand to be carried from source u to destination v. A real variable $l_{i,j}^{u,v}$ is associated with the load on link (i,j) resulting from flow demand $f^{u,v}$. Let $A^{u,v}=\{A_1^{u,v},\cdots,A_k^{u,v},\cdots,A_K^{u,v}\}$ be defined as the set of shortest paths for the flow $f^{u,v}$, $A_k^{u,v}=\{(n_1^k=u,n_2^k),\cdots,(n_{s-1}^k,n_s^k=v)\}$ as the set of links that belong to the shortest path k for the flow $f^{u,v}$ and $x_k^{u,v}$ as a fraction of $f^{u,v}$ that is routed through $A_k^{u,v}$ (calculated using the ECMP rule). The total load on the link (i,j) can be computed as follows:

$$l_{i,j}=\sum_{uv} l_{i,j}^{u,v} \tag{1}$$

where

$$l_{i,j}^{u,v}=\sum_k \sum_{l\in A_k^{u,v}} d_{i,j}^{l} x_k^{u,v} \tag{2}$$

$$d_{i,j}^{l}=\begin{cases}1 & \text{if } l=(i,j)\\ 0 & \text{otherwise}\end{cases} \tag{3}$$

$$\sum_k x_k^{u,v}=f^{u,v} \tag{4}$$

Note that in the case of *unique* shortest path routing i.e. K=1 , (2) becomes $l_{i,j}^{u,v} = \sum_{l\in A_1^{u,v}} d_{i,j}^{l} f^{u,v}$. For a given traffic $F = (f^{u,v}), \forall (u,v) \in N \times N$, the problem is to find a set of metric values $W = (w_{i,j}), \forall (i,j) \in A$ to increase the network performance which can be formulated as :

$$\min \{ r_{\max} \} \quad (5)$$

$$r_{i,j} \le r_{\max}, \forall (i,j) \in A \quad (6)$$

where $r_{i,j} = l_{i,j} / c_{i,j}$ is the utilization of the link (i,j) .

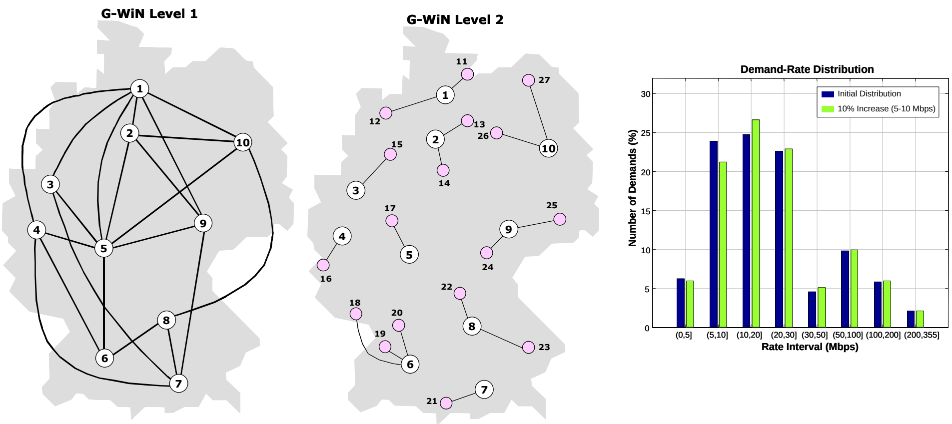

Figure 2 The G-WiN network taken from [13] and its demand distributions

With (5) we prefer solutions with a low $r_{\max}$, which implies that the network is better utilized. Using the simple objective $r_{\max}$ in some cases may need special treatment. In the network shown in Figure 2 for instance, there are no possibilities to reroute traffic traversing the level-2 links (Figure 2 middle). Thus, in those cases it would be better to exclude all such links for computing $r_{\max}$ in (6). Having the traffic matrix and the metric values, we can compute the load distribution on the network. Every solution has a quality measure according to (5). Although a solution is feasible if $r_{i,j} \le 1$ or correspondingly $r_{\max} \le 1$, the optimization is performed with no constraints to force this condition, but we simply minimize the objective function. Note that the formulation presented here is intended for the heuristic solving method to be presented in Section 3.

Partial Demand Increase. Traffic in IP networks is very dynamic and tends to increase over time. By using a simple scaling method we could easily investigate the effect of the traffic increase in the network, because in this case the utilization scales linearly. But this does not necessarily reflect reality, if we assume that traffic and its growth are stochastic quantities. It was our interest to investigate the effect of non-linear traffic growth on network performance. Let $F_o = (f_o^{u,v}), \forall(u,v) \in N \times N$ be defined as the original traffic matrix, and $\Delta F_a = (\Delta f_a^{u,v}), \forall(u,v) \in N \times N$ as a traffic-increase matrix where a denotes the number of non-zero elements of ΔF_a i.e. the number of source destination node pairs with increasing demand. Our new traffic matrix, denoted by $F_a = (f_a^{u,v}), \forall(u,v) \in N \times N$, can be written as:

$$F_a = F_o + \Delta F_a \tag{7}$$

Note that the linear increase $F = lF_o$ is a special case in (7) for $a = 100\%$ and $\Delta F_a = (l - 1)F_o$.Increasing partially the traffic matrix could change the original traffic distribution (see Figure 2 right) and correspondingly the original network utilization. If $r_{\max}^o$ denotes the original maximum utilization caused by distribution of the demands F_o, and $r_{\max}^a$ the maximum utilization caused by F_a using the same routing pattern i.e. without changing routing configuration, we define the increase of the maximum utilization introduced by demand increase ΔF_a as:

$$\Delta r_{\max}^a = r_{\max}^a - r_{\max}^o \tag{8}$$

Furthermore Δr_{diff}^a denotes the difference between maximum and average utilization in the network, resulting from the new demand F_a, i.e. :

$$\Delta r_{\text{diff}}^a = r_{\max}^a - \bar{r}^a \tag{9}$$

3. POLICY AND APPROACHES FOR RE-OPTIMIZATION

Policy for Re-optimization. After recalculating network parameters with the current configuration, a decision should be made whether the network has to be re-optimized. One possibility is to check the value of the increase of the maximum utilization, that is whether $\Delta r_{\max}^a > e_1$ For illustration consider Figure 3, which shows the values of $\Delta r_{\max}^a$ for the 500 samples of traffic-increase matrices ΔF_a for $a = 2\%$ with $\Delta f_a^{u,v}$ randomly distributed in the interval [5,10] Mbps (Figure 3 left) and $\Delta f_a^{u,v} \in [5,50]$ Mbps (Figure 3 right) respectively. The investigation environment will be explained in detail in Section 4. More than 99% of the ΔF_a cause increase in maximum utilization less than 1% in the first case and lower than 5% in the second case. This means, if we set $e_1 = 5\%$ re-optimization should be performed with probability less than 1% for the second case and it is absolutely not necessary for the first case. Using the single parameter $\Delta r_{\max}^a$ sometimes is not adequate, since there are cases where traffic rerouting will not yield better situations. In those cases the network has to be expanded and new hardware capacities should be installed, as well. A further indication could be given by the parameter $\Delta r^{a}_{\text{diff}}$ as given by (9) to roughly measure balance of the traffic distribution. The higher the value of $\Delta r^{a}_{\text{diff}}$, the higher the probability that traffic is distributed in an unbalanced manner. A significant increase in the value of the parameter $\Delta r^{a}_{\max}$ max without the corresponding significant increase in the value of the parameter $\Delta r^{a}_{\text{diff}}$ may indicate that traffic engineering would not be sufficient and network upgrade would probably be necessary. Putting it all together, re-optimization to compensate the impact of demand increase ΔF_a should first be performed when :

$$\Delta r^{a}_{\max} > e_1 \text{ and } \Delta r^{a}_{\text{diff}} > e_2 \tag{10}$$

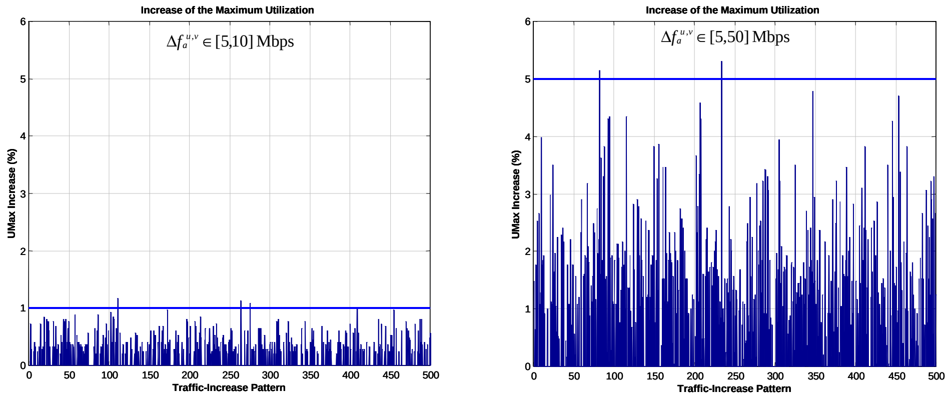

Figure 3 Increase of $r_{\max}$ *caused by* $\Delta F_{2\%}$

Re-optimization Approaches. Re-optimization could then be applied, once requirements (10) are satisfied. A method based on plain local-search (LS) could be an appropriate choice since it gives exact control over the number of changes to be performed to the original weight configuration by exploring its neighbourhood space using all predefined *move* operators. For a comprehensive review of local-search based methods (including simulated-annealing) we refer to [1]. The main layout of the used LS algorithm is shown in Figure 4 left. The first step is to define several move operators satisfying requirements e.g. maximum allowed number of weight changes. Beginning with the first type of move that corresponds to the smallest neighbourhood space, we try to find a valid solution around the original weight configuration. The algorithm will terminate, once a valid solution is found or there are no valid solutions found for all types of move. A validity condition could be set in various ways depending on trade-off factors concerning network performance, time limitation etc. One possibility is again to use the parameters $\Delta r^{a}_{\max}$ and $\Delta r^{a}_{\text{diff}}$ e.g. : a solution is said to be valid when $\Delta r^{a}_{\max} < e_1$ and $\Delta r^{a}_{\text{diff}} < e_2$. The second approach is based on simulated annealing (SA) that belongs to the oldest the metaheuristics and is one of the first algorithms which had an explicit strategy to avoid local optima by allowing moves towards less performing solutions with a certain probability, which is a function of the parameter called temperature. The probability of doing such a move is decreasing during the search. The pseudocode of the SA is displayed in Figure 4 right. In this approach it is not possible to upper bound changes using move operators since the search agent can move everywhere in the solution space. Partially changes could then be achieved by integrating a changes function in the objective function and giving an *importance-factor* between components in the objective [10]. The important part of this method is to guide moves to prefer the original weight values whenever possible.

```
Plain Local Search
solutionFound ← false;
x* ← x_o; β ∈ {1, ..., βmax}
for each type of neighbourhood N_β do
  while (not stopCriterion) do
    x' ← move(x_o, N_β);
    evaluate(x');
    if(x' better than x*)    x* ← x';
    if(isValid(x*)) then
     solutionFound ← true;
     break;
    end if
  end while
  if(solutionFound) break;
end for
```

```
Simulated Annealing
solutionFound ← false;
x* ← x_o; x ← x_o; T ← T_o;
while (not stopCriterion) do
  while (not equilibriumAt(T)) do
    x' ← move(x);
    evaluate(x');
    p ← computeProbability(T);
    if(accept(x', p)) x ← x';
    if(x' better than x*) x* ← x';
    if(isValid(x*)) then
     solutionFound ← true;
     break;
    end if
  end while
  T ← update(T);
  if(solutionFound) break;
end while
```

Figure 4 Pseudocodes for plain local-search and simulated-annealing

4. ANALYSIS AND DISCUSSION

For the following results we used the German research and scientific network G-WiN shown in Figure 2. It consists of 27 nodes (10 level-1 nodes and 17 level-2 nodes) and 76 links. Each level-1 link has a transmission capacity of either 2.5 Gbps or 10 Gbps while each level-2 link has either 2×622 Mbps or 2×2.5 Gbps [2]. The original traffic matrix F_o is composed of 702 flows and each of them was generated randomly in the interval [4,355] Mbps. The mean demand value is 34.6 Mbps and around 75% of the demands are below the value of 30 Mbps (Figure 2 left). For each traffic-increase investigation, 500 increase-patterns are generated randomly with a simple rule, that only node pairs which do not share a common level-1 node are allowed to contribute to demand increase. This is obvious, because demand increase of node pairs sharing a common level-1 node will not affect utilization of the links in level-1, which are our concern in this case. The values of a used in the investigations are 2%, 5%, 10%, 25% and 50%. The metric value was originally set inversely proportional to the link's capacity. Setting the weights in this way caused that 156 (around 22.2%) flows were split because of two or more ties. The average utilization was 23.4% and the most utilized link (of 76.6%) was the link (5,6) which carried 70 different flows. The value $\Delta r_{\text{diff}} = r_{\max} - \bar{r} = 53.2\%$ was a strong indication that the network was not configured appropriately and that traffic engineering should be performed. If we look at the network in Figure 2, we see that traffic on level-2 links is not reroutable, thus for optimization they were marked as *unconsidered* and $r_{\max}$ in (5) was substituted with $r_{\max}^{\text{cons}}$ i.e. the maximum utilization on the level-1 network. Using a slightly different version of the SA described in Section 3, after optimization we obtain: $r_{\max}^{\text{cons}} = 39.4\%$ for the link (7,9), $r_{\max} = 48.4\%$ for the level-2 link (1,12) and $\bar{r} = 24.1\%$. Looking at these values, it is obvious that the optimization saves significantly the network resources: $\Delta r_{\max}^{\text{cons}} = 37.2\%$ (about 930 Mbps capacity in a 2.5 Gbps link), $\Delta r_{\text{diff}}^{\text{cons}} = 15.3\%$ within an acceptable increase in the average number of hops for routing of 0.3 and the increase of 0.7% in average utilization. With the optimized weights' configuration, there are no split flows, since the optimization was set to prefer unique shortest path routing e.g. by using the method in [13].

Performance by Increase Traffic. Figure 5 shows the distributions of the values of $\Delta r_{\max}^{a}$ and $\Delta r_{\text{diff}}^{a}$ for all increase patterns, for different a and increase-intervals. The values of $\Delta f_a^{u,v}$ are randomly distributed in the intervals [2,5], [5,10], [10,20], [20,30], [5,50], [5,100], [50,50] (constant) and [100,100] (constant) Mbps. Totally we have 40 different ΔF_a's each with 500 different patterns. Looking at the first two graphs at the top, partial demand increase ΔF_a whose elements $\Delta f_a^{u,v}$ are randomly distributed below the mean value of the original demands in worst case will increase the maximum utilization up to 29% and the difference $\Delta r_{\text{diff}}^{a}$ up to 33%. In other words, the probability to obtain a value of $r_{\max}$ below 68.4% is quite high, for the case of partial demand increase with $a \le 50\%$ and $\Delta f_a^{u,v}$ randomly distributed below the value of 30 Mbps. For $a \le 10\%$ (equivalent to 36 symmetrical pairs) the corresponding value of $r_{\max}$ is around 49.4%. The last two graphs present the distribution of $\Delta r_{\max}^{a}$ and $\Delta r_{\text{diff}}^{a}$ using wider and overlapping increase-intervals as well as using constant values. In general the bigger $\Delta f_a^{u,v}$ and/or the value of a, the larger are the resulting values of $\Delta r_{\max}^{a}$, $\Delta r_{\text{diff}}^{a}$ as well as the variance of them. And the bigger the value of these two parameters, the higher is the probability that unbalanced traffic distribution occurs in the network and requires re-optimization.

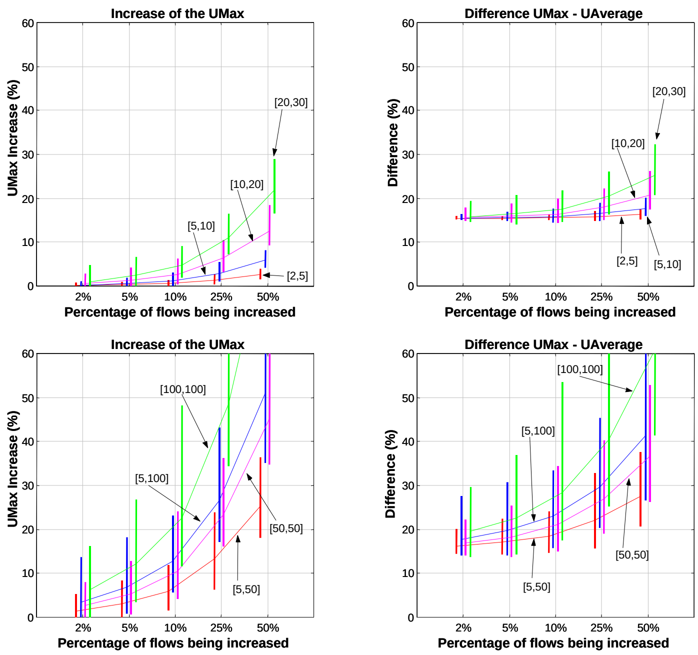

Figure 5 The values of $\Delta r^{a}_{\max}$ and $\Delta r^{a}_{\text{diff}}$ for all a, increase intervals and patterns (ΔF_a)

e_1 (%)	e_2 (%)	[50,50] Mbps					[5, 100] Mbps				
		a (%)	PLS		SA		**a** (%)	PLS		SA	
			b (%)	**c** (%)	**b** (%)	**c** (%)		**b** (%)	**c** (%)	**b** (%)	**c** (%)
15	25	5.6	75	3.38	39.29	25.12	25.8	48.84	3.63	25.58	24.16
15	30	4.2	57.14	3.07	61.9	24.09	23.6	54.24	3	31.36	20.63
20	25	5.6	82.14	2.52	53.57	22.63	24.4	53.28	3.2	37.7	20.25
20	30	0.6	100	1.75	100	42.98	2.6	92.31	3.29	84.62	20.57

Table 1 Re-optimization results for $a = 10\%$ with different values of e_1 and e_2

Re-optimization. Table 1 shows some computational results for different values of e_1 and e_2, for the case of $a = 10\%$ and increase intervals of [5, 100] Mbps and [50,50] Mbps (constant). Column **a** indicates the number of different increase patterns ΔF_a, which trigger the re-optimization procedure, column **b** the number of successful re-optimizations and column **c** the average number of weight changes yielded by all successful re-optimizations. Looking at the values in the columns **b** and **c**, almost in all cases PLS performs better than SA in terms both of the number of successful re-optimizations and the average value of the number of necessary weight changes. However, under identical termination condition (cf. Figure 4) more computation time was needed for re-optimization using PLS : SA was 50 % to 60 % faster.

5. SUMMARY AND CONCLUSION

In this paper we have investigated the impact of partial demand increase on the performance of IP networks. We propose a simple policy for deciding whether to re-optimize the configuration as well as two approaches for implementing the re-optimization. Our experiments show that depending on the policy parameters and demand increase patterns, it is possible to perform minimal re-configuration (in terms of weight changes) in order to keep network performance within an acceptable range. Although we considered in this work only the metric-based traffic engineering scheme, the same principle can be applied to other TE schemes involving other routing mechanisms (e.g. MPLS or hybrid IGP/MPLS).

BIBLIOGRAPHY

[1] Aarts E., Lenstra J.K., *Local Search in Combinatorial Optimization*, John Wiley & Sons Ltd., 1997.

[2] Adler H. M., *Neues im G-WiN*, 37. DFN-Betriebstagung, November 2002.

[3] Awduche D., Chiu A., Elwalid A., Widjaja I., Xiao X., *Overview and Principles of Internet Traffic Engineering*, RFC 3272, May 2002.

[4] Ben-Ameur W., Gourdin E. , Liau B. , Michel N., *Determining Administrative Weigths for Efficient Operational Single-Path Routing Management*, Proceedings of 1st PGTS, 2000.

[5] Bley A., Koch T., *Integer Programming Approaches to Access and Backbone IP Network Planning,* Preprint ZIB ZR-02-41, 2002.

[6] Fortz B., Thorup M., *Internet Traffic Engineering by Optimizing OSPF Weights,* Proceedings of IEEE Infocom, March 2000.

[7] Gajowniczek P., Pioro M., Szentesi A., Harmatos J., *Solving an OSPF Routing Problem with Simulated Allocation*, Proceedings of 1st PGTS, 2000.

[8] Karas P., Pioro M., *Optimisation Problems Related to the Assignment of Administrative Weights in the IP Networks' Routing Protocols*, Proceedings of 1st PGTS, 2000.

[9] Moy J., *OSPF Version 2*, RFC 2328, April 1998.

[10] Mulyana E., Killat U., *A Hybrid Genetic Algorithm Approach for OSPF Weight Setting Problem*, Proceedings of 2nd PGTS, 2002.

[11] Riedl A., *Optimized Routing Adaptation in IP Networks Utilizing OSPF and MPLS,* Proceedings of IEEE ICC, May 2003.

[12] Staehle D., Koehler S., Kohlhaas U., *Optimization of IP Routing by Link Cost Specification*, Tech. Report, University of Wuerzburg, 2000.

[13] Thorup M., Roughan M., *Avoiding Ties in Shortest Path First Routing,* ___________.