Riikka Susitaival and Samuli Aalto. Adaptive load balancing with OSPF. In Proceedings of the Second International Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks (HET-NETs 2004), pages P9/1-P9/10, 2004.

Adaptive load balancing with OSPF

Riikka Susitaival and Samuli Aalto
Networking Laboratory, Helsinki University of Technology
Email: {riikka.susitaival, samuli.aalto}@hut.fi

Abstract

The objective of load balancing is to move traffic from congested links to other parts of the network. If the traffic demands are known, the load balancing can be formulated as an optimization problem. The resulting traffic allocation can be realized in the networks that use explicit routes, such as MPLS-networks. It has recently been found that a similar load balancing is possible to be implemented even in the IP networks based on OSPF-routing by adjusting the OSPF-weights of the links and the traffic splitting ratios in the routers. However, if the traffic demands are unknown or they may change rapidly, another approach is needed. In this paper we study adaptive load balancing in OSPF-networks based on measured link loads. We propose an adaptive and distributed algorithm that gradually balances the load by making small changes in the traffic splitting ratios in the routers. The algorithm is tested numerically in different networks and traffic conditions. The results show that the performance of OSPF-networks can significantly be improved as compared to the equal splitting.
Keywords: OSPF, Traffic Engineering, adaptive routing, load balancing

1 Introduction

Traditionally the traffic is routed along the minimum-hop paths since the usage of resources is then minimized. However, some links may become congested while others remain underloaded. The idea of traffic-aware routing, and specially load balancing, is to avoid congested links when traffic is routed from a router to another. There are two distinct methodologies to implement traffic-aware routing in IP networks. The first one uses current routing protocols like OSPF [1] but the link weights and the traffic splitting ratios in the routers are defined differently from the traditional approach. The second one takes advantage of some explicit routing protocol like MPLS [2] and defines the used paths beforehand.

One of the first proposals to tune OSPF-weights to achieve an optimal load distribution is presented by Fortz and Thorup in [3]. They assume that the routers are bound to split the traffic to a fixed destination equally to the admissible[1] next hops. Under this assumption, however, it is not possible to select the OSPF-weights so that the load distribution is optimal. Even finding a best possible weight setting is shown to be a NP-hard problem. Instead, Fortz and Thorup propose a local heuristic search method for setting the OSPF-weights. In [4] the same authors point out that OSPF-weight changes should be avoided as much as possible, because they confuse the active routing and the performance of TCP goes down.

The load balancing problem in IP networks based on OSPF-routing (OSPF-networks) is also investigated by Wang et al. in [5]. They show that optimal routing in terms of any objective function can be converted to a shortest-path routing with positive link weights if the traffic of each ingress-egress pair can be split arbitrarily to

[1] A next hop is defined here to be admissible if it belongs to some shortest path to the destination.

the shortest paths. However, in current IP routing with OSPF, only equal splitting is possible and, furthermore, the splitting in each router is done based on the destination address only.

Sridharan et al. [6] solve the problems appeared in [5]. The source-destination based splitting is easy to convert to destination based splitting by dividing the sums of incoming and outgoing traffic at the node. The problem of the unequal splitting ratios is solved by taking advantage of the existence of multiple prefixes to a certain destination. For a particular prefix only a part of next hops are available. As the size of the routing table increases this approach approximates well the arbitrary splitting ratios of the optimal routing.

The problem of the approaches presented above is that the traffic demands are assumed to be known, which may be an unrealistic assumption. If the traffic demands are not known, or the traffic conditions may change unexpectedly, another approach is needed. One possibility is to adaptively react to changes in the traffic detected by measurements, such as end-to-end monitoring or monitoring of each link individually.

In paper [7] we studied adaptive load balancing in MPLS-networks. In the present paper we study how similar ideas can be applied in OSPF-networks. Our assumption is that the link loads are measured periodically and the information on the measured loads is distributed to all routers. We suggest an adaptive and distributed algorithm to improve the performance of the network without knowledge of the traffic demands. The idea is that, based on the measured link loads, the routers make independently small changes in the load distribution by adjusting their own traffic splitting ratios.

The rest of the paper is organized as follows. In section 2 we first review a static load balancing problem for off-line optimization of OSPF-weights and then formulate another optimization problem for adjusting the splitting ratios when the paths are fixed. The adaptive and distributed algorithm to optimize the splitting ratios is presented in section 3, and the performance of the proposed algorithm in different test networks and under various traffic conditions is evaluated numerically in section 4. Section 5 concludes the paper.

2 Load balancing based on known traffic demands

In this section we consider a static load balancing problem, in which the traffic demands are assumed to be known. We start with an OSPF-network model. Then we consider the optimization problem in general, after which we review how the OSPF-weights can be determined so that the optimal performance is achieved using shortest path routing. Finally we consider the case where the paths are fixed and only the traffic splitting ratios in the routers may be optimized.

2.1 Network model

Consider an IP network based on OSPF-routing (OSPF-network). Let $\mathcal{N}$ denote the set of nodes (routers) n and $\mathcal{L}$ the set of links l of the network. Alternatively we use notation (i, j) for a link from node i to node j. The capacity of link l is denoted by b_l. The set of ingress-egress (IE) pairs $k = (s_k, t_k)$ is denoted by $\mathcal{K}$ with s_k referring to the ingress node and t_k referring to the egress node of IE-pair k. Let $\mathcal{P}_k$ denote the set of all possible paths p from node s_k to node t_k. We use notation $l \in p$ if link l belongs to path p. The traffic demand of IE-pair k is denoted by d_k.

In the link state based routing protocols like OSPF, each link l is associated with a fixed weight w_l and the traffic is carried along shortest paths. Let $\mathcal{P}_k^{\text{SP}}$ denote the set of shortest paths from node s_k to node t_k with respect to link weights w_l,

$$\mathcal{P}_k^{\text{SP}} = \{p \in \mathcal{P}_k \mid \sum_{l \in p} w_l = \min_{p' \in \mathcal{P}_k} \sum_{l' \in p'} w_{l'}\}.$$

The standard choice $w_l = 1$ for all l results in minimum-hop paths and, thus, minimizes the total required link bandwidth.

In each node i, the incoming traffic with the same destination t is aggregated and then splitted to links (i, j) that belong to some shortest path of the ingress-egress pair (i, t). Such adjacent nodes j are called admissible next hops. Let ϕ_{ij}^t denote the corresponding splitting ratios. Thus, ϕ_{ij}^t refers to the fraction of overall traffic passing node i and destined to node t that is forwarded on link (i, j). It is required that

$$\sum_{j:(i,j) \in p \text{ for some } p \in \mathcal{P}_{(i,t)}^{\text{SP}}} \phi_{ij}^t = 1.$$

For an illustration, see Figure 1. As, e.g., in [3], it is usually assumed that these splitting ratios ϕ_{ij}^t are equal,

$$\phi_{ij}^t = \frac{1}{|\{j' : (i,j') \in p \text{ for some } p \in \mathcal{P}_{(i,t)}^{\text{SP}}\}|}.$$

This choice is referred to as Equal Cost Multiple Path (ECMP). However, as mentioned in section 1, there is a method that allows unequal splitting ratios [6].

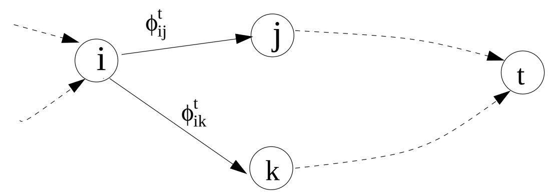

Figure 1: The network model

2.2 Static load balancing problem

Load balancing[2] can be based on minimizing the mean delay or minimizing the maximum link utilization, for example. The former one emphasizes both load balancing and short paths, whereas the latter one can route traffic along long routes also. In this paper we concentrate on the latter one.

The optimal solution to the minimization problem of the maximum link utilization is not unique in general. Among the optimal solutions, the one that minimizes the overall usage of the resources is the most reasonable. Thus it is convenient to formulate an LP-problem that minimizes the maximum link utilization with a greater weight but also takes into account the overall usage of the resources with a smaller weight as, e.g., in [5]:

$$\begin{aligned} &\text{Minimize } \alpha + r \sum_{k\in\mathcal{K}} \sum_{l\in\mathcal{L}} x_l^k && \text{subject to the constraints} \\ &\alpha \geq 0,\ x_l^k \geq 0, && \text{for each } k \in \mathcal{K} \text{ and } l \in \mathcal{L}, \\ &\sum_{k\in\mathcal{K}} x_l^k \leq \alpha b_l, && \text{for each } l \in \mathcal{L}, \\ &Ax^k = R^k, && \text{for each } k \in \mathcal{K}, \end{aligned} \tag{1}$$

where α and x_l^k are the free variables describing the minimum of the maximum link utilization and the traffic load of IE-pair k on link l, respectively, and r is some small constant. Furthermore, $A \in \mathbb{R}^{N\times L}$, where $N = |\mathcal{N}|$ and $L = |\mathcal{L}|$, denotes the matrix for which $A_{nl} = -1$ if link l directs to node n, $A_{nl} = 1$ if link l leaves from node n, and $A_{nl} = 0$ otherwise; $x^k \in \mathbb{R}^{L\times 1}$, $k \in \mathcal{K}$, refers to the link load vector with elements x_l^k; and $R^k \in \mathbb{R}^{N\times 1}$, $k \in \mathcal{K}$, denotes the vector for which $R_{s_k}^k = d_k$, $R_{t_k}^k = -d_k$, and $R_n^k = 0$ otherwise.

From the optimal traffic loads x_l^k it is possible to determine the set $\mathcal{P}_k^{\text{LB}}$ of paths p that are used to carry the traffic demand d_k from node s_k to node t_k,

$$\mathcal{P}_k^{\text{LB}} = \{p \in \mathcal{P}_k \mid x_l^k > 0 \text{ for all } l \in p\}.$$

2.3 Load balancing in OSPF-networks

Wang et al. [5] proved that there is a set of positive link weights w_l so that the optimal paths in the load balancing problem (1) are shortest paths with respect to these link weights. In other words, $\mathcal{P}_k^{\text{LB}} \subseteq \mathcal{P}_k^{\text{SP}}$ for all k. The procedure to define these link weights is given below.

Let $\tilde{y}_l = \sum_k \tilde{x}_l^k$ denote the traffic load allocated to link l in the optimal solution $\tilde{x}_l^k$ of the load balancing problem (1). Formulate then another LP-problem (primal) and its dual. In the primal LP-problem the induced

[2] Also known as optimal routing.

traffic loads $\tilde{y}_l$ serve as new capacity constraints:

$$
\begin{array}{ll}
\text{Minimize } \sum_{k\in\mathcal{K}}\sum_{l\in\mathcal{L}} x_l^k & \text{subject to the constraints} \\
x_l^k \geq 0, & \text{for each } k\in\mathcal{K} \text{ and } l\in\mathcal{L}, \\
\sum_{k\in\mathcal{K}} x_l^k \leq \tilde{y}_l, & \text{for each } l\in\mathcal{L}\ , \\
Ax^k = R^k, & \text{for each } k\in\mathcal{K},
\end{array}
\tag{2}
$$

The dual of the problem above is:

$$
\begin{array}{ll}
\text{Maximize } \sum_{k\in\mathcal{K}} d_k U_{t_k}^k - \sum_{l\in\mathcal{L}} \tilde{y}_l W_l & \text{subject to the constraints} \\
W_l \geq 0 & \text{for each } l\in\mathcal{L}\ , \\
U_{s_k}^k = 0, & \text{for each } k\in\mathcal{K} \\
U_j^k - U_i^k \leq W_{(i,j)} + 1, & \text{for each } k\in\mathcal{K} \text{ and } (i,j)\in\mathcal{L}.
\end{array}
\tag{3}
$$

The required link weights are then given by $w_l = W_l + 1$, where the variables W_l are determined as the solution to the dual problem.

In addition, optimal destination based traffic splitting ratios ϕ_{ij}^t are determined from the link loads x_l^k of the solution of the primal problem. These splitting ratios are calculated as follows [6]:

$$
\phi_{ij}^t = \frac{\sum_{k:t_k=t} x_{(i,j)}^k}{\sum_{j':(i,j')\in\mathcal{L}} \sum_{k:t_k=t} x_{(i,j')}^k}
\tag{4}
$$

2.4 Optimization of the splitting ratios

The traffic control reacting to changes in traffic demands by changing the link weights is often too time consuming or impractical. In such a case with fixed link weights, we can still affect the traffic distribution by optimizing the traffic splitting ratios used in the routers.

We present a procedure to determine the splitting ratios that minimize the maximum link utilization with the given link weights. As before, let $\mathcal{P}_k^{\text{SP}}$ denote the set of shortest paths for IE-pair k with respect to these link weights. Let ϕ_p denote the fraction of traffic demand d_k that uses path $p\in\mathcal{P}_k^{\text{SP}}$. We start by solving these splitting ratios for each IE-pair k from the following LP-problem:

$$
\begin{array}{ll}
\text{Minimize } \alpha + r\sum_{k\in\mathcal{K}}\sum_{l\in\mathcal{L}}\sum_{p\in\mathcal{P}_k^{\text{SP}}:l\in p} d_k\phi_p & \text{subject to the constraints} \\
\alpha \geq 0,\ \phi_p \geq 0, & \text{for each } p\in\bigcup_{k\in\mathcal{K}}\mathcal{P}_k^{\text{SP}}, \\
\sum_{k\in\mathcal{K}}\sum_{p\in\mathcal{P}_k^{\text{SP}}:l\in p} d_k\phi_p \leq \alpha b_l, & \text{for each } l\in\mathcal{L}, \\
\sum_{p\in\mathcal{P}_k} \phi_p = 1, & \text{for each } k\in\mathcal{K}.
\end{array}
\tag{5}
$$

Let ϕ_p be the optimal traffic share on path p. This induces the following link loads:

$$
x_l^k = \sum_{p\in\mathcal{P}_k^{\text{SP}}:l\in p} d_k\phi_p.
$$

The destination based splitting ratios for each node i can then be calculated as in (4).

3 Adaptive load balancing

The static load balancing problem presented in the previous section is possible to be formulated and solved only if the traffic demands d_k are known. It may well be the case that such information is either imprecise, outdated, or totally missing. In such a case, another approach is needed. In this section we first formulate the corresponding dynamic load balancing problem for OSPF-networks and then describe an adaptive and distributed algorithm to solve the dynamic problem.

3.1 Dynamic load balancing problem

Our assumptions are as follows. The traffic demands d_k are fixed but unknown. The link loads are periodically measured at times t_n. Let $\hat{y}_l(n)$ denote the measured link load of link l from the measurement period (t_{n-1}, t_n). The information on the measured loads is distributed to all nodes in the network.[3] The time needed to distribute the information is negligible in comparison to the length of the measurement period.

In general, the objective of our dynamic load balancing problem is as follows. Based on the measured link loads, the link weights w_l and the traffic splitting ratios ϕ^t_{ij} should be adjusted so that they converge, as soon as possible, to the (unknown) optimal values of the corresponding static load balancing problem presented in subsection 2.3.

However, as mentioned in section 1, it is not desirable to modify the link weights too frequently. Therefore, we consider the dynamic problem in two time-scales. In the shorter time-scale, only the traffic splitting ratios are adjusted but the link weights are kept fixed. The objective in this case is to adjust the traffic splitting ratios so that they converge to the (unknown) optimal values of the corresponding restricted optimization problem presented in subsection 2.4. In the longer time-scale, also the link weights should be adjusted so that the optimal load distribution is finally achieved. One way to do it is to estimate the traffic demands from the measurement data and then determine the link weights as a solution to the dual problem presented in subsection 2.3.

In this paper we focus on dynamic load balancing in the shorter time-scale. An adaptive and distributed algorithm to solve this problem is described in the following subsection.

3.2 Adaptive and distributed algorithm for load balancing

We assume that the link weights w_l are fixed. For each IE-pair k, let $\mathcal{P}^{\mathrm{SP}}_k$ denote the set of shortest paths from node s_k to node t_k with respect to these link weights w_l.

Let $\phi^t_{ij}(n)$ denote the traffic splitting ratios that are based on the measured link loads $\hat{y}(n) = (\hat{y}_l(n); l \in \mathcal{L})$. We note that, since the measured link loads are distributed to all nodes, the decisions concerning the traffic splitting ratios can be done in a distributed way. Thus, in our adaptive and distributed algorithm, each node i independently determines the traffic splitting ratios ϕ^t_{ij} for all destination nodes t and admissible next hops j.

The decisions in the algorithm are based on a cost function $D_p(y)$ defined for each path $p \in \mathcal{P}^{\mathrm{SP}}_{(i,t)}$ by

$$D_p(y) = \max_{l \in p} \frac{y_l}{b_l},$$

where $y = (y_l; l \in \mathcal{L})$. This is a natural choice as the objective is to minimize the maximum link utilization. The idea in the algorithm is simply to alleviate the congestion on the most costly path by reducing the corresponding traffic splitting ratio. This should, of course, be compensated by increasing the splitting ratio related to some other path. A problem in adaptive adjustment of the splitting ratios in a short time-scale is the possible disorder of the packets. However, this can be solved by changing only a part of the splitting ratios at a time, for example.

Since the algorithm is adaptive, we have a closed-loop control problem: the splitting ratios that depend on measured loads have a major effect on the upcoming load measurements. It is well-known that feedback control systems are prone to instability if the gain in the loop is too large. Thus, to avoid harmful oscillations, we let the splitting ratios change only with minor steps. The step size is determined by the granularity parameter g. A finer granularity is achieved by increasing the value of g. The measurement period (5 minutes if SNMP is used) should be short enough to obtain reasonably fast convergence.

[3] This can done in a similar way as the link states are distributed to all routers within an AS in OSPF.

Algorithm At time t_n, after receiving the information $\hat{y}(n)$ concerning all the measured loads, node i adjusts the traffic splitting ratios for all destination nodes t as follows:

1. Calculate the cost $D_p(\hat{y}(n))$ for each path $p \in \mathcal{P}^{\text{SP}}_{(i,t)}$.
2. Find the path $q \in \mathcal{P}^{\text{SP}}_{(i,t)}$ with maximum cost, i.e. $D_q(\hat{y}(n)) = \max_{p \in \mathcal{P}^{\text{SP}}_{(i,t)}} D_p(\hat{y}(n))$, and decrease the splitting ratio of the first link (i,j) of that path as follows:
$$\phi^t_{ij}(n) = \phi^t_{ij}(n-1) - \frac{1}{g}\phi^t_{ij}(n-1).$$
3. Choose another path $r \in \mathcal{P}^{\text{SP}}_{(i,t)}$ randomly and increase the splitting ratio of its first link (i,k) as follows:
$$\phi^t_{ik}(n) = \phi^t_{ik}(n-1) + \frac{1}{g}\phi^t_{ij}(n-1).$$
4. For all other admissible next hops j', keep the old splitting ratio,
$$\phi^t_{ij'}(n) = \phi^t_{ij'}(n-1).$$

4 Numerical performance evaluation

In this section we evaluate numerically the performance of the proposed adaptive load balancing algorithm. First, in subsection 4.1, a simple but efficient numerical evaluation method is described. The method is similar to that developed in [7]. Thereafter, in subsection 4.2, the results of this evaluation method applied to two different test networks are presented.

4.1 Evaluation method

The evaluation method is iterative and runs as follows. The test network (including the nodes n, links l, IE-pairs k, and paths p), the link weights w_l and the traffic demands d_k are first fixed.[4] Traffic of each IE-pair is initially allocated to the shortest paths with respect to the fixed link weights w_l. If multiple shortest paths exist, traffic is initially split equally in each node (ECMP).

At each iteration n, the measured link loads $\hat{y}_l(n)$ induced by the splitting ratios $\phi^t_{ij}(n-1)$ are calculated as follows. First we calculate, for each IE-pair k, the induced traffic splitting ratios $\phi_p(n-1)$ for each path $p \in \mathcal{P}^{\text{SP}}_k$ by
$$\phi_p(n-1) = \prod_{(i,j)\in p} \phi^{t_k}_{ij}(n-1).$$
Then the measured link loads $\hat{y}_l(n)$ are determined by
$$\hat{y}_l(n) = \sum_{k\in\mathcal{K}} (d_k + \epsilon_k(n)) \sum_{p\in\mathcal{P}^{\text{SP}}_k : l\in p} \phi_p(n-1),$$
where the $\epsilon_k(n)$ are independent Gaussian random variables with mean 0 and variance $\delta^2 d_k^2$ describing the random fluctuations of traffic during measurement period n around the fixed demands d_k. The coefficient of variation, δ, for this random variable is assumed to be the same for all IE-pairs k. After this, the new traffic splitting ratios $\phi^t_{ij}(n)$ are determined from the measured loads $\hat{y}_l(n)$ as presented in subsection 3.2.

4.2 Numerical results

Two different test networks (see Figure 2) are used with the following characteristics:

1. 10 nodes, 52 links, and 72 IE-pairs;
2. 20 nodes, 102 links, and 380 IE-pairs.

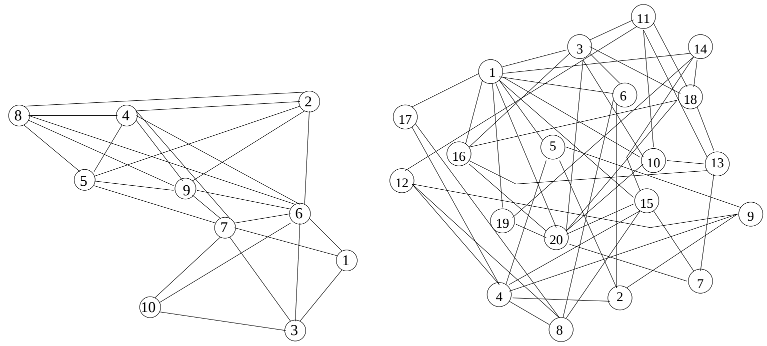

Figure 2: Left-hand-side: 10-node network. Right-hand-side: 20-node network.

The test networks are random networks generated by the mechanism described in [8].

Three different traffic scenarios are used. In the first one the random traffic fluctuations in the time-scale of measurements are ignored by setting the fluctuation parameter δ to 0. In the second one these random fluctuations are taken into account by setting the fluctuation parameter δ to 0.1. In the third scenario, we consider the traffic fluctuations in a longer time-scales by letting the traffic demands to change drastically three times during the evaluation period (after 500, 1000 and 1500 iterations, correspondingly).

The results of the adaptive algorithm are compared with

1. "ECMP": the standard policy where the traffic is splitted equally to the shortest paths with the unit link weights,
2. "Sub-optimal": the optimal value of the restricted optimization problem (5) with link weights fixed to 1, and
3. "Optimal": the optimal value of the static load balancing problem (1).

No traffic fluctuations In this scenario $\delta = 0$. The shortest paths needed for the adaptive algorithm are calculated using link weights $w_l = 1$ for all links l. Figure 3 shows the resulting maximum link utilization for the 10-node and 20-node networks as a function of the number of iterations for granularity parameters $g = 20$ and $g = 50$. We can see that the performance of the adaptive algorithm approaches the sub-optimal value and improves the performance remarkably as compared to the standard equal splitting. A small step size in the algorithm ensures that oscillations are insignificant. The convergence times are only two times greater in the 20-node network (approx. 200 iterations) than in the 10-node network (approx. 100 iterations) in spite of the huge growth in the complexity of the network.

Random traffic fluctuations in a shorter time-scale In this scenario $\delta = 0.1$. The shortest paths needed for the adaptive algorithm are again calculated using link weights $w_l = 1$ for all links l. Figure 4 shows the resulting maximum link utilization for the 10-node and 20-node networks as a function of the number of iterations for granularity parameters $g = 20$ and $g = 50$. We find that the random traffic fluctuations in a time-scale of the measurement period induce oscillations to the maximum link utilization. However, even with granularity parameter $g = 20$, oscillations are tolerable and the algorithm converges close to the sub-optimal value.

Traffic fluctuations in a longer time-scale In this scenario the traffic demands change drastically three times. The shortest paths needed for the adaptive algorithm are first calculated using link weights $w_l = 1$ for all links l. Figure 5 shows the resulting maximum link utilization for the 10-node and 20-node networks as a function of the number of iterations for granularity parameters $g = 20$ and $g = 50$. The adaptive algorithm reacts to the changes and provides a result close to the sub-optimal value in the 10-node network and close to

[4]Note that the traffic demands are used only for the *evaluation* purposes. The algorithm itself does *not* use any information on these demands.

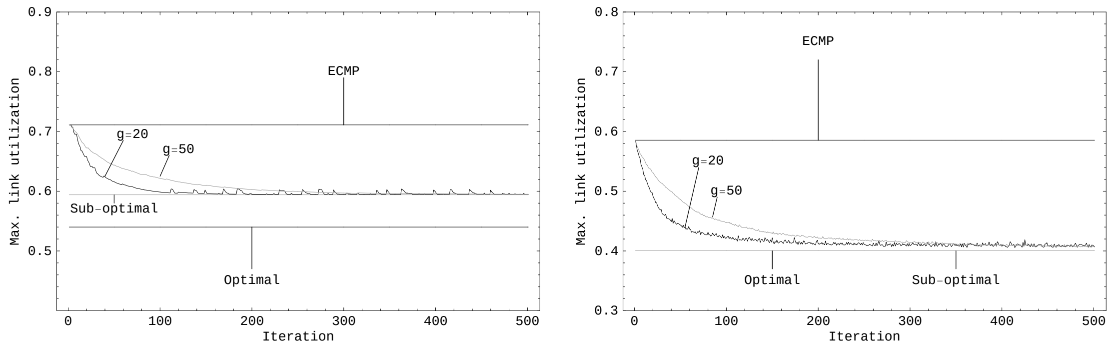

Figure 3: The maximum link utilization as a function of the number of iterations, when there are no traffic fluctuations, $\delta = 0$. Left-hand-side: 10-node network. Right-hand-side: 20-node network.

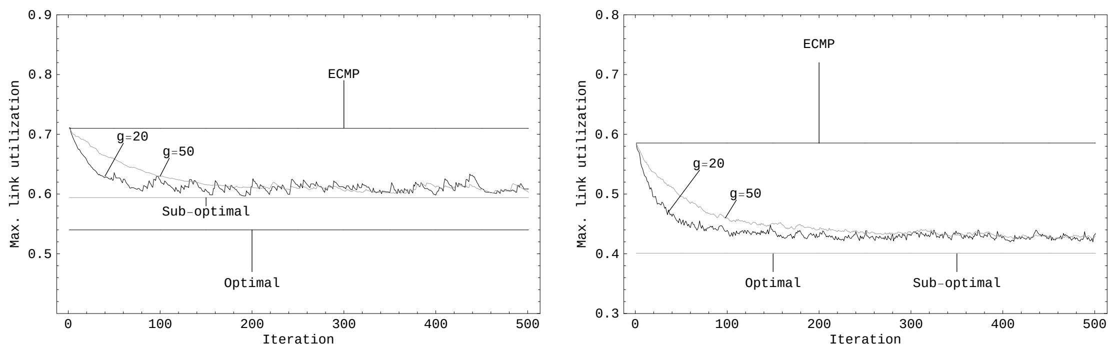

Figure 4: The maximum link utilization as a function of the number of iterations, when there are random traffic fluctuations in a shorter time-scale, $\delta = 0.1$. Left-hand-side: 10-node network. Right-hand-side: 20-node network.

the optimal value in the 20-node network. Note that in the 10-node network in the iteration rounds from 500 to 1000 the result of heuristics and the sub-optimal value are equal.

Then we assumed that the link weights and the splitting ratios of each router can be optimized in a longer time scale (hours to days). The shortest paths needed for the adaptive algorithm are thus first calculated using the optimal link weights corresponding to the original traffic demands d_k and determined from the dual problem (3) and the splitting ratios from the solution of the primal problem (2) and formula (4). After that the link weights remain unchanged. Figure 6 shows the resulting maximum link utilization for the 10-node and 20-node networks as a function of the number of iterations for granularity parameters $g = 20$ and $g = 50$. Now the sub-optimal values in the 10-node network are close to the optimal values and thus heuristics can yield to the results close to the optimal solution. In the 20-node network the results are similar to the previous case in Figure 5, except the first iterations where the results are now immediately close to the optimal one.

As a conclusion, the optimization of the link weights in the longer time-scale improves the performance of the network. In the 20-node network also the unit weights provide a good result. An explanation is that, in the 10-node network, the number of shortest paths related to the unit link weights is 116 whereas it is 153 in the case of the optimal link weights. Thus the number of ϕ-parameters is greater in the latter case and also the results are better. In the 20-node network, the corresponding numbers of the shortest paths are 782 and 785. Also the results are quite similar.

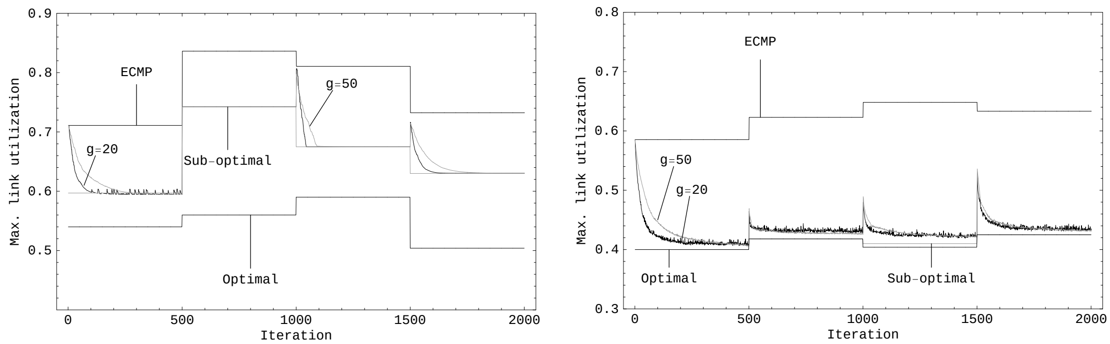

Figure 5: The maximum link utilization as a function of the number of iterations, when there are traffic fluctuations in a longer time-scale. Left-hand-side: 10-node network. Right-hand-side: 20-node network.

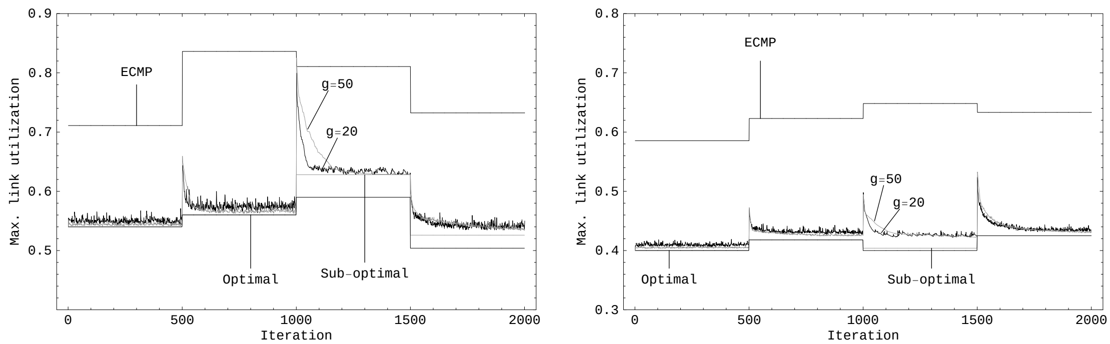

Figure 6: The maximum link utilization as a function of the number of iterations. Left-hand-side: 10-node network. Right-hand-side: 20-node network.

5 Conclusion

In this paper we studied how load can be balanced adaptively in OSPF-networks using a distributed approach. We also considered the procedure of optimizing the OSPF-weights by primal-dual methods and how this can be combined to adaptive heuristics. The results show that the optimization of traffic splitting ratios improves the performance of the network when compared to equal splitting. When the set of shortest paths is small also the changing of OSPF-weights is worthwhile.

In the future the approach which combines the shorter and longer time-scale optimization has to be developed further. Also the actual converge time of the adaptive algorithm has to be studied carefully in realistic traffic scenarios. In addition, we have to study if the disorder of packets is really a problem and how this problem can be solved.

Acknowledgements. This work was financially supported by the Academy of Finland (grant n:o 74524).

References

[1] J. MOY, OSPF Version 2, IETF RFC2328, April 1998.

[2] E. ROSEN, A. VISWANATHAN AND R. CALLON, Multiprotocol Label Switching Architecture IETF RFC3031, January 2001.

[3] B. FORTZ, M. THORUP, Internet Traffic Engineering by Optimizing OSPF Weights, in Proc. of IEEE Infocom 2000.

[4] B. FORTZ AND M. THORUP Optimizing OSPF/IS-IS Weights in a Changing World, IEEE Journal on Selected Areas in Communications, Vol. 20, No. 4, pp. 756-767, 2002.

[5] Y. WANG, Z. WANG AND L. ZHANG Internet Traffic Engineering without Full Mesh Overlaying, Proc. of IEEE Infocom 2001.

[6] A. SRIDHARAN, R. GUÉRIN AND C. DIOT, Achieving Near-Optimal Traffic Engineering Solutions for Current OSPF-IS-IS Networks, in Proc. of IEEE Infocom 2003.

[7] R. SUSITAIVAL, S. AALTO AND J. VIRTAMO, Adaptive load balancing using MPLS, accepted for publication in Proc. of MMB&PGTS, 2004.

[8] C. VILLAMIZAR, OSPF Optimized Multipath (OSPF-OMP), IETF Internet-draft, draft-ietf-ospf-omp-02, February 1999.