12

Evaluation of Traffic Engineering

Traffic engineering is concerned with minimising the over-utilisation of capacity when other capacities are available in the network by re-routing traffic flows. A *traffic flow* in the context of this chapter is a macroflow consisting of all packets entering the network at the same ingress and exiting at the same egress node. The traffic flows of all ingress-egress node pairs are specified in a traffic matrix. Throughout this chapter, we assume that the traffic matrix is given. For Multi-Protocol Label Switching (MPLS) networks, the traffic matrix can be measured online and exactly by the method described in Schnitter and Horneffer (2004); for a general discussion of traffic matrix estimation techniques we refer to Section 11.3.

In this chapter, we investigate how traffic engineering influences the efficiency and the Quality of Service (QoS) of a network and explore the optimisation potential with an evaluation of different traffic engineering strategies.

We assume MPLS or an equivalent forwarding architecture (see Section 6.3) is used which allows us to explicitly establish the path on which a flow is routed through the network. Different traffic engineering strategies that differ in their objective function, their constraints and whether they can split up a macroflow to be routed along multiple paths (multipath routing) or not are investigated. Their performance with respect to different performance metrics is evaluated. In addition, the performance gain of the best traffic engineering strategies compared to a plain shortest path routing solution is evaluated. It measures the additional QoS achievable with traffic engineering and is a measure of the possible efficiency gain of traffic engineering.

As the currently dominant QoS architecture is an over-provisioned best-effort architecture, this architecture is assumed for the experiments in this work. Most of the results here, however, are also helpful for other architectures, e.g. Diffserv. One straightforward approach is to employ traffic engineering techniques sequentially for all traffic classes, starting with the highest priority traffic. The traffic of the next highest priority is then traffic engineered on the network that has capacity left that is not used by the higher priority traffic and so on. More sophisticated approaches for traffic engineering in the context of other QoS architectures, are discussed in Section 11.2.

The Competitive Internet Service Provider: Network Architecture, Interconnection, Traffic Engineering and Network Design
Oliver Heckmann © 2006 John Wiley & Sons, Ltd. ISBN: 0-470-01293-5

For an evaluation of the traffic engineering performance of a network for a given traffic matrix, the routing determined by the traffic engineering algorithm has to be measured. The *routing* in this context consists of the paths chosen for the different macroflows. Therefore, we start by discussing different performance metrics for evaluating the routing. The average path length is an obvious performance metric. Besides it, several other performance metrics are possible, for example, the bottleneck utilisation. They are discussed in Section 12.1. In Section 12.2, different strategies for solving the QoS maximising multi-commodity flow problem are introduced; they are evaluated in a series of experiments in the rest of the chapter. The experiment set-up is described in Section 12.3. After that, the experiment results are presented and discussed:

- In the first experiment (Section 12.4), we compare the path selection and explicit routing formulation of the optimisation problem.
- A general performance evaluation of a large number of traffic engineering strategies are presented in Section 12.5. We start with a detailed evaluation of the strategies and all performance metrics in a basic experiment and then vary several parameters of the basic experiment – e.g. the used topology – to evaluate their influence.
- In Section 12.6, the performance loss of singlepath algorithms compared to multipath algorithms is evaluated.
- The most successful strategies need a number of precalculated paths. In Section 12.7, the influence of the precalculated paths on the performance of these strategies is evaluated.

Finally in Section 12.8, the conclusions are drawn and we give recommendations whether, and how to use traffic engineering.

12.1 Traffic Engineering Performance Metrics

For the evaluation of traffic engineering, the performance of the traffic flows routed through the network has to be evaluated. In this section, we discuss several metrics that can be used to evaluate the performance of the routing.

12.1.1 Path Length

Minimising the average path length between two nodes is an obvious and straightforward traffic engineering goal: With respect to different length metrics, minimising the path length is the objective of most standard interior routing protocols like Open Shortest Path First (OSPF) (see Section 6.4.1). The motivation behind the path length as performance metric is that the longer the path becomes, the more network resources are consumed and the higher the propagation delay becomes. As in a congested network the queuing delay can easily exceed the propagation delay of a hop; re-routing a flow so that it takes more hops through the network can still lead to improved overall delay besides a reduced loss probability. This observation is the basic motivation for doing traffic engineering instead of plain shortest path routing. Nevertheless, the path length remains an important performance metric for traffic engineering solutions.

12.1.2 Maximal Bottleneck Utilisation

The utilisation u_l of a link l is defined as the load l_l per capacity (bandwidth) c_l

$$u_l = \frac{l_l}{c_l} \quad (12.1)$$

The utilisation of a link is an average over a certain period[1], typical utilisation metrics measured in Internet Protocol (IP) networks are based on 5 minute, 15 minute, 2 hour and 24 hour averages.

The maximum utilisation $\max_l\{u_l\}$ describes how loaded the bottleneck link of the topology is. QoS parameters such as delay and loss are a (non-linear) function of the utilisation of a link. Because of the bursty nature of network traffic (see Section 5.1), losses occur long before an average utilisation of 100% is reached. Minimising the maximum utilisation therefore indirectly improves the QoS on the bottleneck link – the most critical link – and creates a zone of security against unpredicted traffic increases. Therefore, minimising this metric is often the dominating traffic engineering goal in related works, for example, see Hasslinger and Schnitter (2002a,b); Lin and Wang (1993); Poppe *et al.* (2000); Roughan *et al.* (2003).

One disadvantage of this metric is that it focuses exclusively on the bottleneck links, while ignoring the other links.

12.1.3 Average Utilisation

Instead of evaluating the *maximum* utilisation – that is, the utilisation of the bottleneck link – one could also evaluate the *average* utilisation of the network. This has the advantage that no link is ignored. However, a network with some highly loaded and some lightly loaded links could show the same average utilisation as a network with only medium loaded links. As the QoS flow experiences – for example, the loss probability – is largely determined by the most utilised link on its path and not by the average utilisation along its path this metric can be misleading. This is shown in some of the experiments below.

12.1.4 Average Load

The average utilisation metric does not take into account that there might be large differences in the capacity c_l of the links in the topology. The average utilisation is influenced by low capacity links the same way as by high capacity links. High capacity links, however, typically carry more traffic flows and can therefore be expected to influence more flows (or users) than smaller links. If the utilisation metric is weighted with the link capacity, the average load can be calculated.

This metric has the same disadvantages as the previous one. It is used in Poppe *et al.* (2000) as a secondary objective, for example.

[1] On a very short timescale a link is either 100% utilised (data is currently being transmitted) or 0% (no data is currently being transmitted).

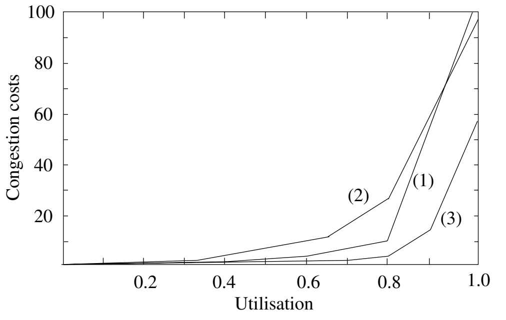


Figure 12.1 Congestion Functions

12.1.5 Congestion Costs

The high-level primary goal of traffic engineering should be to maximise the overall utility of the customers given the available network resources. This is a special form of the network efficiency we used throughout this book. The utility depends on the application, on the traffic mix and on network parameters like the loss or the queuing delay (see Chapter 8). On the timescale of traffic engineering, it is the average of the network parameters like loss and delay that can be influenced. The network parameters are a non-linear function of the utilisation or the load situation on a link. Assume for example, a M/M/1/B queue (see Section 3.1.5). The M/M/1/B queue is not the most realistic representation of an Internet link but is a commonly used one because it can be mathematically handled very well. For more realistic queuing models, see the queuing models of Appendix D. They, however, show a similar basic behaviour.

Loss and queuing delay of the M/M/1/B are depicted in Figure 3.4. As one can see there, the loss and delay are non-linear convex functions of the utilisation. In Section 13.1, we present a more detailed analysis based on the utility for various QoS systems using packet-level simulations instead of the M/M/1/B formulas. The results there also point out a convex relationship.

The convex relationship between the utilisation and network congestion indicators like loss and delay has an important implication for traffic engineering. If the load of one highly utilised link is reduced by a certain amount due to a routing change, the overall performance can improve even if the load on multiple other (but not so highly utilised) links is increased because of the routing change. The lower the utilisation becomes, the less can be gained by re-routing. This behaviour is not correctly expressed by any of the above-mentioned metrics. Therefore, we propose to use the following metric called *congestion costs*[2] that captures this non-linear behaviour. Figure 12.1 presents three different stepwise linear convex congestion cost functions $p^x(u)$ that we use throughout this chapter to model how the congestion situation of a link depends on the utilisation of the link. Fortz and Thorup (2002) use a very similar metric to evaluate OSPF-based traffic engineering. The parameters of the congestion cost functions are arbitrarily chosen but roughly oriented to

[2] The reason for calling this congestion measure "costs" becomes more visible in Chapter 13 where it has to be added to the costs for expanding the capacity of the network and therefore has to have the same unit as true monetary costs. For the experiments of this chapter, the scale and unit of this congestion measure do not matter and do not influence the results.

Figure 3.4 and the results of Section 13.1. The default congestion cost function is labelled (1) and used in the following experiments by default if nothing else is mentioned; it is varied in Section 12.5.2.

The congestion costs are calculated for every link and can be summed up for the complete topology in the following two ways:

- Weighted congestion costs: $\sum_l l_l \cdot p^x(u_l)$
  The motivation behind weighting the congestion costs with the load l_l follows the same argument as for the average load versus average utilisation metric above: Links with a high load are likely to affect more customers and can therefore be judged more important than links with a lower load.
- Unweighted congestion level: $\sum_l p^x(u_l)$
  For comparison reasons, we will also investigate the unweighted congestion costs metric in this chapter.

12.2 Traffic Engineering Strategies

We use optimisation models to describe different traffic engineering strategies mathematically, using the following notation:

A network (η, ζ) consists of a set of nodes η and a set of directed links $l_{ij} \in \zeta$ with link l_{ij} connecting node i to j. A link l_{ij} has a capacity c_{ij}.

A subset η_e of the nodes is marked as *edge nodes*. Customers and interconnection partners are connected to these nodes; therefore the edge nodes are potential sources and sinks for the traffic flows while the other nodes $n \in (\eta \backslash \eta_e)$ only forward traffic (*core nodes*).

There are F traffic flows f that have to be routed through the network. A traffic flow f is characterised by its ingress node $i_f \in \eta_e$ and egress node $e_f \in \eta_e$ and its size r_f; the size of the flow is its traffic volume or – if we assume time periods of a fixed duration as a basis – its average transmission rate.

The ingress and egress nodes (i_f, e_f) of flow f are connected by a set of different paths ρ_f. Each path $p \in \rho_f$ is an ordered set of links $\phi_p = \{l_{i_f j_1}, l_{j_1 j_2}, \ldots, l_{j_k e_f}\}$ from the ingress i_f to the egress node e_f. For our analysis, we assume that the length l_p of a path p is the number of links it contains; for a real network other factors such as path length metric could also be taken into account, for example, the propagation delay.

12.2.1 Traffic Engineering Objectives

The overall goal of traffic engineering is to optimise the routing of flows through a network of given and fixed capacity; traffic engineering is thus an optimisation problem. Several specific objectives can be formulated as an objective function of the traffic engineering problem. As several objectives can be optimised at the same time, the optimisation problem can be a multi-objective optimisation problem[3]. The different objective functions can be combined, either as prioritised objectives (multilevel programming) or as weighted summed objectives. In the first case, the problem is first optimised with the primary objective function only in mind and among all the solutions that optimise

[3] For multi-objective optimisation see Eschenauer *et al*. (1990); Statnikov and Matusov (1995).

the primary objective function the one that optimises the secondary objective function is selected. In the latter case, both objective functions are added with certain weights to a single objective function and the resulting problem is then optimised for the aggregate objective function. Prioritised objective functions can be approximated with the weighted ones by giving the primary objective function a much larger weight than the secondary one; because of that, we restrict ourselves to the second approach with weighted objective functions in the optimisation problems that we present and discuss below.

In Section 12.1, metrics for evaluating the performance of traffic engineering strategies were presented and discussed. Obviously, they can also be used as objective functions for the traffic engineering problems. We do so by integrating them into the more sophisticated traffic engineering strategies below.

12.2.2 Shortest Path Routing

The shortest path routing strategy is straightforward: Each traffic flow f is routed along its shortest path p^* with $l_{p^*} = \min_{p \in \rho_f}\{l_p\}$. The shortest path can, for example, be determined with the Dijkstra algorithm (see Dijkstra (1959)). Each flow is routed along a single path only, multipath routing is never used. The shortest path routing algorithm minimises the average path length metric only, other target functions are not considered. This strategy is used as a reference because it is the default strategy of a network with a standard routing protocol and no traffic engineering functionality.

12.2.3 Equal Cost Multipath

As another reference strategy, we include an equal cost multipath algorithm. It splits a flow evenly among a given number of paths. The equal cost multipath algorithm we use has two parameters n and Δl. n denotes the maximum number of paths considered. For a flow f, the n shortest paths are determined with a modified Dijkstra algorithm. The shortest of these paths is denoted as p^*. All paths that are more than Δl hops longer than p^* are discarded. If there are more than n shortest paths left within Δl hops, those that have the most overlapping (same links) with the shortest path are discarded until only n paths are left. The traffic is split up evenly among the remaining paths. This algorithm does not directly minimise any of the metrics of Section 12.1; it is included for reference purposes only.

12.2.4 Explicit Routing

The explicit routing strategy is based on the explicit routing form of the multi-commodity flow problem (see Section 11.2). The network's topology is modelled by the set I_n and O_n that contains the ingoing and outgoing links l of node n. The explicit routing optimisation problem is given with Model 12.1 as a singlepath model and Model 12.2 as a multipath model, both with the weighted maximum utilisation and average utilisation criteria as objective function (12.2).

Variable a_{lf} describes which proportion of flow f is routed via link l. Constraint (12.3) is the flow conservation constraint: For all nodes that are not the ingress or egress node of flow f, the amount of traffic from flow f that flows into node n also

Model 12.1 Explicit Routing (Singlepath)

Indices

$f = 1, \ldots, F$	Flow f
$n = 1, \ldots, N$	Node n
$l = 1, \ldots, L$	Link l

Parameters

r_f	Size of flow f
I_n	Set of incoming links of node n
O_n	Set of outgoing links of node n
i_f	Ingress (start) node of flow f
e_f	Egress (end) node of flow f
w^ξ	Weight for the maximum utilisation objective
w^u	Weight for the average utilisation objective
c_l	Capacity of link l

Variables

ξ	Maximal link utilisation
a_{lf}	Routing variable, flow f is routed by this proportion on link l

$$\text{Minimise } w^\xi \xi + w^u \frac{1}{L} \sum_l \sum_f \frac{r_f a_{lf}}{c_l} \tag{12.2}$$

subject to

$$\sum_{l \in O_n} a_{lf} = \sum_{l \in I_n} a_{lf} \qquad \forall f\, \forall n \neq i_f,\, e_f \tag{12.3}$$

$$\sum_{l \in O_{i_f}} a_{lf} = 1 + \sum_{l \in I_{i_f}} a_{lf} \qquad \forall f \tag{12.4}$$

$$\sum_f r_f a_{lf} \leq c_l \xi \qquad \forall l \tag{12.5}$$

$$0 \leq \xi \leq 1 \tag{12.6}$$

$$a_{lf} \in \{0, 1\} \qquad \forall l\, \forall f \tag{12.7}$$

Model 12.2 Explicit Routing (Multipath)

Constraint 12.7 is replaced with the following constraint in the otherwise unchanged Model 12.1:

$$0 \leq a_{lf} \leq 1 \qquad \forall l \, \forall f \tag{12.8}$$

has to leave node n. Constraint (12.4) specifies that 100% of a flow f is inserted into the network at the ingress node i_f. Because of (12.3) and (12.4) no extra constraint for the egress node e_f is necessary. (12.5) forces variable ξ to the maximum utilisation of all links l and in combination with (12.6) ensures that the capacity c_l of a link l is not exceeded.

The multipath explicit routing problem can be solved with the simplex algorithm, for example (see Section 3.3), the singlepath version is harder to solve because of the binary constraint (12.7). It has to be solved with Mixed Integer Programming (MIP) solving techniques like, Branch & Bound with LP relaxation as discussed in Section 3.3. Let F denote the number of flows, N the number of nodes and L the number of links. As $O(L) = O(N)$ and $O(F) = O(N^2)$ the number of constraints and therefore the complexity of the explicit routing LP/MIP models is $O(N^3)$. The number of (computationally expensive) binary variables in the singlepath model are $O(N^3)$. As this is a rather high complexity, we next present a more efficient model for traffic engineering.

12.2.5 Path Selection

As mentioned above, the explicit routing model is of high complexity. The main reason for this is that it explicitly models the topology and thus the solution algorithm searches for paths through the network at the same time as assigning the flows to these paths so that the traffic engineering goals are optimised. For computing paths through the network, especially, there exist efficient specialised algorithms like the Dijkstra algorithm rather than the general LP/MIP solving algorithms.

Therefore, the optimisation problem can be simplified by precomputing the possible paths for all flows in a first step. Then in a second step, the path(s) for each flow are selected in a way that optimises the objective function. Precomputing the paths can, for example, be done with a (modified) Dijkstra algorithm in polynomial time. The optimisation models for selecting one or more paths for each flow among the precomputed ones are discussed below and called *path selection models*.

If all possible paths for all flows f are precomputed and used as input in the path selection models, the path selection models yield the same optimal solution as the explicit routing model. However, as for a large topology the number of possible paths is extremely high, only the shortest n paths for each flow can typically be considered in the path selection model, making the solution space of the path selection smaller than that of the explicit routing problem. In that case, it is possible that the path selection model does not find the globally optimal solution. We investigate this experimentally in Section 12.4. At first glance, this might seem a drawback, but, in actuality, the fact that the path selection models use precomputed paths gives the decision maker more control over the possible paths. The explicit routing models could route a flow over a path that is much longer than the shortest path. For the path selection models, the decision maker can limit the paths, for example, so that they do not have more than Δl additional hops than the shortest path between two nodes.

The basic path selection model is mathematically specified as a mixed integer programming (MIP) model in Model 12.3. It is a singlepath model. The multipath version of Model 12.3 is given by Model 12.4.

Model 12.3 accounts for four of the five traffic engineering goals discussed in Section 12.1. To account for the congestion costs, additional parameters and variables are necessary. Model 12.5 is an extension of Model 12.3 that also accounts for the congestion costs in the objective function.

The path selection models can be solved with the same methods as the explicit routing models. Their complexity is reduced to $O(N^2)$.

The objective function (12.9) of Model 12.3 minimises the maximum utilisation, the average utilisation, the average load, and the average path length. Each of these criteria is weighted with a special parameter w, if a parameter w is set to zero, the according criterion is ignored when searching for the optimal solution.

Constraint (12.10) is the routing constraint and makes sure that every flow is routed along *one* path. Please note that in the basic model variable a_{fp} is a binary variable. If the binary condition (12.15) is relaxed towards (12.16) in Model 12.3, multipath routing is allowed and a flow can be split up.

Constraint (12.11) sets the utilisation u_l of a link l in relation to the amount of traffic routed through that link and its capacity. Constraint (12.12) forces ξ to the maximum utilisation. (12.13) to (12.15) form the non-negative binary constraints of Model 12.3.

In Model 12.5, the congestion costs are additionally added to the objective function (12.17). They are measured with variable x_{sl} that is set in (12.18) to the value by which the lower threshold of step s of the congestion cost function is exceeded on link l. The congestion with added weighted high capacity links are likely to be used by more users than low capacity links; therefore, they should be weighted higher. The unweighted congestion costs (the last term in the objective function) are included for reference only.

Please note that any algorithm could be used to calculate the paths that are used as input for the path selection models. Throughout our experiments we use the same method described above in Section 12.2.3 to the n shortest paths with minimal overlappings that have no more than Δl additional hops than the shortest path. How to choose the parameters n and Δl is discussed in Section 12.7.

12.3 Experiment Setup

In the rest of the chapter, the above presented traffic engineering strategies are evaluated in a number of experiments. Each experiment is repeated N times. The average of the performance metrics of Section 12.1 and the 95% confidence intervals are derived from the results. They are presented and discussed in the following sections.

For each experiment, a topology is selected; we use the German Research Network (DFN) topology as the default topology for all the experiments. For some experiments,

Model 12.3 Path Selection (Singlepath)

Indices

$f = 1, \ldots, F$	Flow f
$p \in \rho_f$	Path p for flow f
$l = 1, \ldots, L$	Link l

Parameters

r_f	Size of flow f
ρ_f	Set of available paths for flow f
l_p	Length of path p
c_l	Capacity of link l
ϕ_p	Set of links belonging to path p
w^ξ	Weight for the maximum utilisation objective
w^u	Weight for the average utilisation objective
w^l	Weight for the average load objective
w^p	Weight for the average pathlength objective

Variables

ξ	Maximal link utilisation
u_l	Utilisation of link l
a_{fp}	Routing variable, flow f is routed via path p by the amount denoted with a_{fp}

$$\text{Minimise } w^\xi \xi + w^u \frac{1}{L} \sum_l u_l + w^l \frac{1}{L} \sum_l c_l u_l + w^p \frac{1}{F} \sum_f \sum_{p \in \rho_f} l_p a_{fp} \tag{12.9}$$

subject to

$$\sum_{p \in \rho_f} a_{fp} = 1 \qquad \forall f \tag{12.10}$$

$$\sum_f \sum_{p \mid l \in \phi_p} r_f a_{fp} = c_l u_l \qquad \forall l \tag{12.11}$$

$$u_l \leq \xi \qquad \forall l \tag{12.12}$$

$$\xi \geq 0 \tag{12.13}$$

$$0 \leq u_l \leq 1 \qquad \forall l \tag{12.14}$$

$$a_{fp} \in \{0, 1\} \qquad \forall f \, \forall p \in \rho_f \tag{12.15}$$

Model 12.4 Path Selection (Multipath)

Constraint (12.15) of the otherwise unchanged Model 12.3 is replaced with

$$0 \leq a_{fp} \leq 1 \qquad \forall f \, \forall p \in \rho_f \tag{12.16}$$

Model 12.5 Path Selection with Congestion Costs (Singlepath)

Model 12.3 is extended as follows:

Index

$s = 1, \ldots, S$	Step s of the congestion costs function, see Figure 12.1

Parameters

p_s^x	Additional congestion costs in step s of the congestion costs function
q_s	Lower threshold of step s of the congestion costs function
w^x	Weight for the congestion costs objective (weighted with capacity)
$\tilde{w}^x$	Weight for the congestion costs objective (not weighted)

Variables

x_{sl}	Congestion costs variable, denotes by how much the threshold of step s of the congestion cost function has been exceeded on link l

$$\text{Minimise } (12.9) + w^x \sum_l c_l \sum_s p_s^x x_{sl} + \tilde{w}^x \sum_l \sum_s p_s^x x_{sl} \tag{12.17}$$

subject to (12.10)-(12.15) and

$$x_{sl} \geq u_l - q_s \qquad \forall s \, \forall l \tag{12.18}$$

$$x_{sl} \geq 0 \qquad \forall s \, \forall l \tag{12.19}$$

we also vary the topology. The topology is modelled as a directed graph, the capacity of opposing links is assumed equal in all experiments of this section.

A traffic matrix[4] is necessary to evaluate the strategies. Unfortunately, measured traffic matrices are not available as providers are reluctant to reveal information about their topology and traffic characteristics or prohibit publication. Therefore, we have to generate

[4] More exactly: The structural relationship between the traffic matrix and the link capacities of the topology.

artificial traffic matrices based on the information known about the characteristics of traffic matrices. We generate multiple traffic matrices per experiment and vary the generation method during the experiments – our experiments below show that the results are stable for different traffic matrices and traffic distributions.

12.3.1 Traffic Creation

Traffic flows are created between all node pairs. Bhattacharyya *et al.* (2001) show that traffic flows differ drastically in their size (hence often named mice and elephants) and that points of presence (POPs) nodes in a POP level topology show large differences in throughput. We model this behaviour with node weights; the node weight of the source and sink node massively influence the flow size. The node weight can be imagined to represent the size of the customer base served by this node. Prior to the traffic generation, for each node n a node weight w_n is randomly selected from the list (1, 2, 3, 4) with the probabilities (60%, 20%, 10%, 10%).

Then, the size r_f of traffic flow f between ingress node i_f and egress node e_f is drawn from a uniform distribution in interval $[0.6 \cdot w_{i_f} \cdot w_{e_f}, 3.0 \cdot w_{i_f} \cdot w_{e_f}]$.

12.3.2 Capacity Assignment

Finally, the capacities of the links have to be determined. As the link capacities very strongly influence the performance (see Section 13.1), it is very important to set them to "realistic" values. Similar to a QoS system, traffic engineering has the highest impact in times when the network is highly loaded. Therefore, for our evaluation, a high-load situation is assumed as they typically occur in the late morning or early evening hours (see Roberts (2001)) .

In a real network, traffic volumes increase over time and link capacities are upgraded at regular intervals and in discrete steps by adding new or upgraded line cards to the routers. A typical approach is to double the capacity of a link once a certain utilisation threshold is exceeded. How large this threshold is strongly depends on the timescale used for the utilisation. For our evaluation of traffic engineering, we assume that the evaluation is based on a rather short timescale and a busy period.

We use the following algorithm to set the link capacities (bandwidths) in order to reflect that the network has a history and has grown to satisfy the traffic patterns:

1. Each link is assigned an arbitrary starting bandwidth of 155. This value is motivated by the bandwidth provided by Synchronous Transfer Mode-1 (STM-1)/Optical Carrier-3 (OC-3) links, see Table 4.2 in Section 4.1.4.
2. The utilisation of all links is determined based on the assumption that the flows are routed on their shortest path through the network.
3. If the utilisation of a link exceeds 80%, the bandwidth of the link is doubled successively until the utilisation is below 80%. This represents the "history" of the network and that it has grown to accommodate the traffic.
   The drawback of this approach is that the network capacities will be optimised to a certain extent for the shortest path routing algorithm which can give it a slight edge compared to the other algorithms. In Section 12.5.4, the generation method is therefore varied and different traffic distributions are analysed.

4. As the next step, each traffic flow is increased randomly by 1% to 10% to introduce more variation and to make sure that the capacities are not fully optimised for shortest path routing. One can imagine that this represents traffic growth since the network was expanded the last time.
5. If the bandwidths of two opposite links are not equal, they are set to the maximum of the two bandwidths so that the bandwidth between two nodes is symmetrical.

12.4 Explicit Routing versus Path Selection

As mentioned above, the path selection strategies offer a reduced computational complexity over the explicit routing strategies at the costs of a reduced solutions space because the choice of paths is restricted. The reduced solution space can lead to sub-optimal results with respect to the selected objective function. To evaluate how likely sub-optimal results are, we run an experiment with $N = 50$ repetitions with the singlepath and multipath strategies for the DFN topology (see Figure A.1). For the path selection algorithm we chose two different sets of paths, one with a maximum number of $n = 5$ paths between each node pair and maximal $\Delta l = 2$ additional hops and one with the shortest $n = 10$ paths and any number of additional hops allowed ($\Delta l = \infty$). The maximum utilisation was chosen as objective function with a weight of 1000 and the average utilisation with a weight of 1. As can be seen from the results in Tables 12.1 and 12.2, the $10/\infty$ path selection and the explicit routing strategy came to the same solution for all 50 different problem incarnations. However, the explicit routing strategy needed considerably more time[5]. The $5/2$ path selection strategy leads to the same results for the primary objective

Table 12.1 Explicit Routing versus Path Selection, Multipath

Strategy	Time to Solve [s]	Maximum Utilisation (%)	Average Utilisation (%)
Shortest-Path	0.289	88.906	53.7
Path Selection 5/2	1.748	82.329	52.34
Path Selection 10/∞	5.296	82.329	52.21
Explicit Routing	18.553	82.329	52.21

Table 12.2 Explicit Routing versus Path Selection, Singlepath

Strategy	Time to Solve [s]	Maximum Utilisation (%)	Average Utilisation (%)
Shortest-Path	0.289	88.906	53.7
Path Selection 5/2	9.383	85.376	53.63
Path Selection 10/∞	17.282	85.376	53.18
Explicit Routing	33.695	85.376	53.18

[5] The *time to solve* in Tables 12.1 and 12.2 was measured on a 2 GHz Mobile Pentium with 512 MB Random Access Memory (RAM) using the MIP solver CPlex (see ILOG CPLEX (2004)).

function and due to the reduced solution space to slightly worse results for the secondary objective function. It is, however, very fast to solve.

Because of their better computational performance, their increased flexibility, and the insignificant difference in the results, we focus on the path selection strategies in the rest of the chapter.

12.5 Performance Evaluation

In this section, the performance of different traffic engineering strategies is evaluated. The shortest path strategy is used as a reference and several path selection strategies with different objective functions are evaluated. Their parameters n and Δl are set to $n = 5$ and $\Delta l = 2$. The effect of changing these parameters is analysed in Section 12.7. The first experiments are based on the DFN topology, other topologies are evaluated in Section 12.5.3. We start with multipath routing. The discussion of the singlepath variant of the strategies will be the subject of Section 12.6. Table 12.3 lists the selected strategies and their abbreviations.

We evaluate the performance of the strategy based on all metrics discussed in Section 12.1. Our focus, however, will be on the congestion costs because it best captures the overall performance of a network. The absolute value of the congestion costs and the link load bears no deeper meaning, therefore these values are normalised relative to those yielded by the SP strategy.

12.5.1 Basic Experiment

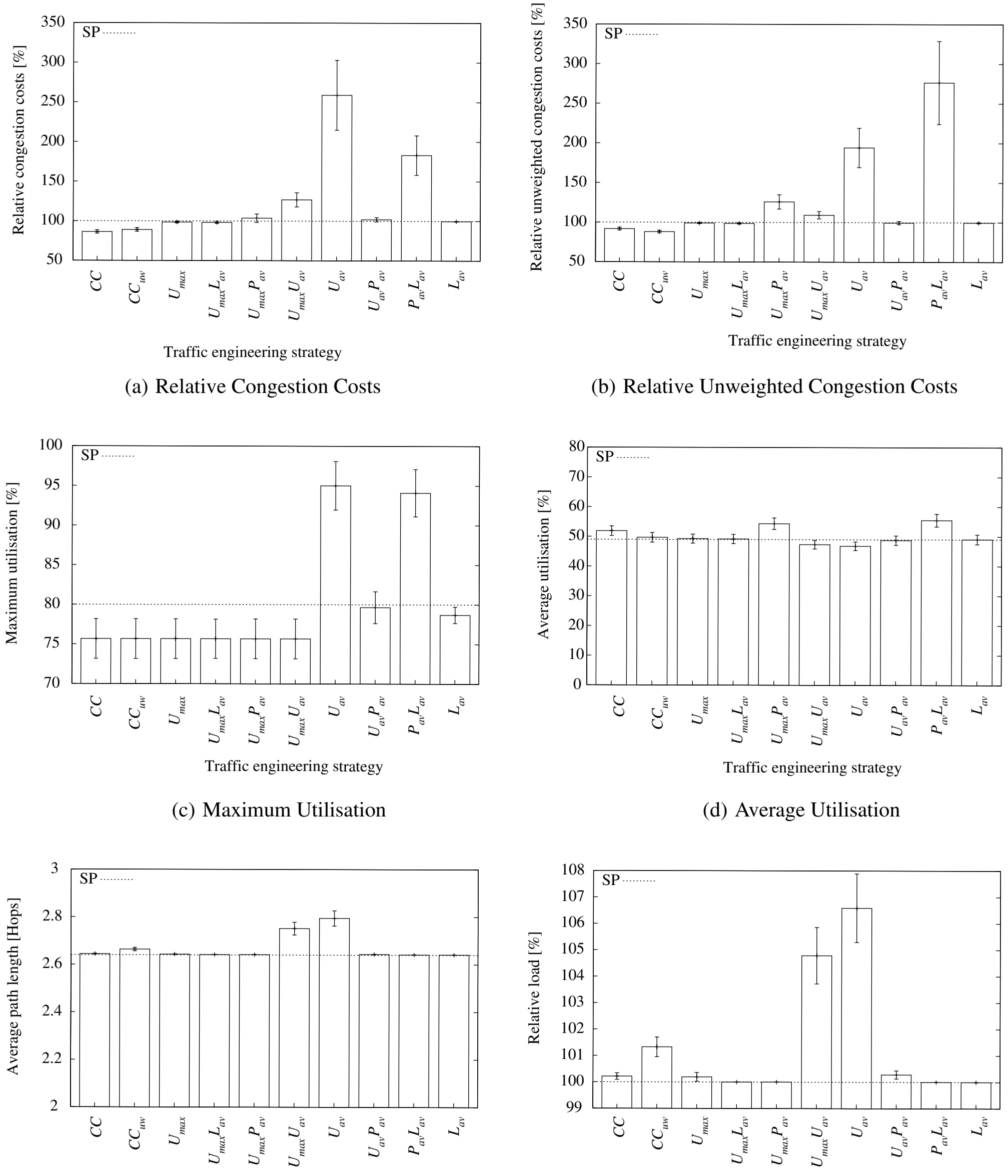
The average and 95% confidence intervals over all $N = 20$ different randomly created problem incarnations are summarised in Table 12.4 and shown in Figure 12.2.

Table 12.3 Abbreviations of the Traffic Engineering Strategies

Strategy	Denotation
SP	Shortest path routing
CC	Path selection: Minimise (weighted) congestion costs
CC_{uw}	Path selection: Minimise unweighted congestion costs
U_{max}	Path selection: Minimise maximum utilisation
$U_{max}L_{av}$	Path selection: Minimise maximum utilisation with $w^{\xi} = 1000$ and average load with $w^l = l_{SP}(l_{SP}$ is the average load of the SP strategy)
$U_{max}P_{av}$	Path selection: Minimise maximum utilisation with $w^{\xi} = 1000$ and average path length with $w^p = 1$
$U_{max}U_{av}$	Path selection: Minimise maximum utilisation with $w^{\xi} = 1000$ and average utilisation with $w^u = 1$
U_{av}	Path selection: Minimise average utilisation
$U_{av}P_{av}$	Path selection: Minimise average utilisation with $w^u = 1000$ and average path length with $w^p = 1$
$P_{av}L_{av}$	Path selection: Minimise average path length with $w^p = 1000$ and average load with $w^l = 1$
L_{av}	Path selection: Minimise average load

Table 12.4 Results of the Basic Experiment

Strategy	Congestion Costs		Unweighted Congestion Costs		Maximum Utilisation		Average Utilisation		Average Path Length		Average Load	
	Average	Confidence Interval	Average	Confidence Interval	Average	Confidence Interval	Average	Confidence Interval	Average	Confidence Interval	Average	Confidence Interval
SP	*1.00*	–	*1.00*	–	*0.80*	0.01	*0.49*	0.02	*2.64*	0.00	1.00	–
CC	*0.86*	0.02	*0.92*	0.02	*0.76*	0.03	*0.50*	0.02	*2.66*	0.01	1.00	0.00
CC_{uw}	*0.89*	0.02	*0.88*	0.02	*0.76*	0.03	*0.50*	0.02	*2.66*	0.01	1.01	0.00
U_{max}	*0.99*	0.01	*0.99*	0.01	*0.76*	0.03	*0.49*	0.02	*2.64*	0.00	1.00	0.00
$U_{max}L_{av}$	*0.98*	0.01	*0.99*	0.01	*0.76*	0.02	*0.49*	0.02	*2.64*	0.00	1.00	0.00
$U_{max}P_{av}$	*1.04*	0.05	*1.26*	0.09	*0.76*	0.03	*0.54*	0.02	*2.64*	0.00	1.00	0.00
$U_{max}U_{av}$	*1.27*	0.09	*1.09*	0.05	*0.76*	0.03	*0.47*	0.01	*2.75*	0.03	1.05	0.01
U_{av}	2.59	0.44	*1.94*	0.25	*0.95*	0.03	*0.47*	0.01	*2.79*	0.03	1.07	0.01
$U_{av}P_{av}$	*1.02*	0.03	*1.00*	0.02	*0.80*	0.02	*0.49*	0.02	*2.64*	0.00	1.00	0.00
$P_{av}L_{av}$	*1.83*	0.26	*2.76*	0.52	*0.94*	0.03	*0.56*	0.02	*2.64*	0.00	1.00	0.00
L_{av}	*0.98*	0.01	*1.00*	0.01	*0.79*	0.01	*0.49*	0.02	*2.64*	0.00	1.00	0.00



(a) Relative Congestion Costs

(b) Relative Unweighted Congestion Costs

(c) Maximum Utilisation

(d) Average Utilisation

(e) Average Path Length

(f) Average Load

Figure 12.2 Basic Results

The *congestion costs* are evaluated first. As the CC and CC_{uw} strategies directly optimise the congestion costs, they yield the minimal weighted or unweighted congestion costs. All other strategies show a bad performance with respect to congestion. Only a few of them (U_{max}, $U_{max}L_{av}$, L_{av}) perform a little better than the shortest path (SP) reference strategy. Compared to SP, they can reduce the overall congestion of the network only by 1% or 2%. The other strategies perform worse than the shortest path strategy with respect to the congestion. Of all, the U_{av} strategy leads to the worst performance. Part of these results can be attributed to the fact that due to our method of generating the traffic and the link capacities the network capacities are relatively well adapted for the shortest path strategy. The extent of this effect is analysed in Section 12.5.4. Also, for other topologies the performance of the traffic engineering strategies compared to SP improves, see Section 12.5.3.

Comparing the results for the unweighted congestion costs with those of the weighted (default) congestion costs, some interesting effects can be observed. To explain them, one has to keep in mind that the difference between the utilisation of a link and the load of a link is the factor link capacity. The link capacity is also the difference between the weighted and unweighted congestion costs – the link capacity influences the weighted but not the unweighted congestion costs. This explains why the strategies that consider the average load ($U_{max}L_{av}$, $P_{av}L_{av}$, L_{av}) and therefore (indirectly) the link capacities, perform relatively better for the weighted congestion costs than for the unweighted ones. Vice versa, the strategies that consider the average utilisation ($U_{max}U_{av}$, U_{av}, $U_{av}P_{av}$) and therefore ignore the link capacities when calculating the average, perform relatively better for the unweighted congestion costs.

Next, the *maximum utilisation* performance metric is evaluated for all strategies. The maximum utilisation of a network shows how loaded the bottleneck links of that network are. As can be seen from Table 12.4, all U_{max} strategies lead to the lowest maximum utilisation as the maximum utilisation is their objective function. Besides these strategies, the CC and CC_{uw} strategies – despite having a different objective function – also lead to the lowest maximum utilisation. This is also not surprising, considering the convex shape of the congestion cost function that gives strong incentives to keep the utilisation low.

The U_{av} and $P_{av}L_{av}$ strategies lead to an unacceptably high maximum utilisation and thus create at least one bottleneck that is higher utilised than the bottleneck in the shortest path routed network. This behaviour should be avoided by traffic engineering strategies. These strategies cannot therefore be recommended.

Looking at the *average utilisation* as a performance metric one can notice that all strategies except $U_{max}P_{av}$ and $P_{av}L_{av}$ lead to an average utilisation very close to that of the SP reference strategy. The strategies minimising the average utilisation – especially $U_{max}U_{av}$ and U_{av} – lead to a slightly lower average utilisation. There is a trade-off between optimising average load and average utilisation. This can also be seen in the results for the *average load* performance metric. There, all strategies except $U_{max}U_{av}$ and U_{av} lead to almost the same average load[6] as the SP reference strategy while $U_{max}U_{av}$ and U_{av} lead to significantly higher average loads. There is no potential for reducing the average load compared to SP, as the average load is automatically minimised if flows are routed along their shortest path. Only if flows are routed on a path that is longer than the shortest path the average load is increased – and besides that obviously also the average path length. This is also visible for the *average path length,* only $U_{max}U_{av}$ and U_{av} show a significant increase in the average path length compared to the reference strategy, the increase of the path length for the other strategies is very small. This result shows that there is no reason

[6] Most differences are smaller than 10^{-2}.

to worry about the increase of the propagation delay for the traffic engineering strategies. Also, for all path selection strategies the maximum increase of the propagation delay is controlled by the parameter Δl.

Next, the performance for the individual strategies is summarised. CC performs very well for all criteria and can therefore be recommended without doubt. Also, it shows the best performance with respect to the congestion cost metric which we deem the most important metric. CC also performs significantly better than the SP strategy. It reduces the overall congestion by 14%.

The excellent performance of CC also reflects itself in the performance of the related CC_{uw} strategy. Here, the congestion costs are not weighted in the objective functions, congestion on a low bandwidth link is therefore treated the same as congestion on a high bandwidth link. As explained before, we do not recommend doing this, nevertheless, the performance of the CC_{uw} strategy is very good.

The U_{max} strategy and the derivatives of that strategy that minimise the average load, utilisation or path length as secondary objective obviously show the best performance for the maximum utilisation metric. Also they perform well for the average utilisation (except $U_{max}P_{av}$), path length (except $U_{max}U_{av}$) and load (except $U_{max}U_{av}$). However, for the congestion costs they do not perform well. $U_{max}P_{av}$ and $U_{max}U_{av}$ perform especially badly and cannot be recommended. If a U_{max} strategy has to be used, $U_{max}L_{av}$ should be used. However, the CC strategies perform significantly better and should be favoured.

U_{av} only minimises the average utilisation and cannot be recommended. The performance improves considerably if the objective function is combined with a second objective function as in $U_{av}P_{av}$. However, $U_{max}L_{av}$ and especially CC then still perform better. Similarly, $P_{av}L_{av}$ and L_{av} perform worse than these two mentioned strategies.

12.5.2 Variation of the Congestion Cost Function

We argued above that the congestion cost function is the best and most important traffic engineering performance metric. While it is clear that the congestion cost function is of a convex shape, the question remains how the exact shape of the function influences the results. In this section, we evaluate this influence by repeating the above experiments for the three different congestion cost functions of Figure 12.1. The resulting congestion costs are summarised in Table 12.5. The evaluation of other criteria like the *average utilisation*, the *average load* and the *average path length* was not affected more than 1%.

As one can see, the strategies that perform exceptionally badly with respect to congestion costs (U_{av}, $P_{av}L_{av}$) are influenced to a great extent by the exact shape of the congestion cost function. Nevertheless, independent of the shape, they remain the worst strategies with respect to congestion costs.

The other strategies are only slightly influenced by the congestion cost function. The exact shape of the congestion function does not influence the ranking of the strategies. However, the advantage of the CC strategies compared to the SP strategy depends on the shape of the congestion cost function. In the experiment, this advantage varies between 5% and 14%. The relatively small advantage for the congestion cost function (3) can be explained by the relatively small steepness of the function for high values of utilisation. By re-routing flows, highly utilised links are relieved by the CC strategy. The higher the steepness of the function, the higher the lowered utilisation reflects itself in the results.

Table 12.5 Congestion Cost Metric for Different Strategies and Congestion Cost Functions

	Original (1)		Function (2)		Function (3)	
	Average	Confidence Interval	Average	Confidence Interval	Average	Confidence Interval
SP	*1.00*	–	*1.00*	–	*1.00*	–
CC	*0.86*	0.02	*0.91*	0.02	*0.95*	0.01
CC_{uw}	*0.89*	0.02	*0.94*	0.02	*0.98*	0.02
U_{max}	*0.99*	0.01	*0.99*	0.01	*0.98*	0.01
$U_{max}L_{av}$	*0.98*	0.01	*0.98*	0.01	*0.98*	0.01
$U_{max}P_{av}$	*1.04*	0.05	*1.03*	0.05	*1.05*	0.04
$U_{max}U_{av}$	*1.27*	0.09	*1.23*	0.08	*1.16*	0.07
U_{av}	*2.59*	0.44	*2.03*	0.27	*3.40*	0.84
$U_{av}P_{av}$	*1.02*	0.03	*1.01*	0.02	*1.02*	0.03
$P_{av}L_{av}$	*1.83*	0.26	*1.49*	0.15	*2.52*	0.49
L_{av}	*0.98*	0.01	*1.00*	0.00	*1.00*	0.00

See Figure 12.1 for the shape of the congestion cost functions.

The results of this experiment show that for the choice of the strategy the exact shape of the congestion cost function is not important. This is important for the application of the congestion cost strategies because it cannot be expected that a single function can be specified for a network that exactly represents the influence of the link utilisation on the congestion for all traffic types and users (see also Section 13.1). Congestion cost functions will always be approximations and estimates. Due to the relatively small influence of the exact shape, however, this does not matter much.

12.5.3 Influence of the Topologies

The previous experiments were based on the DFN topology. In this section, the influence of the topology network graph on the performance of the traffic engineering strategies is evaluated. The different analysed topologies and their basic connectivity properties like the diameter and the out-degree distribution are presented in Appendix A.

Because of the little influence of the other metrics in the previous experiments, the evaluation is restricted here to the congestion cost metric (Table 12.6) and the maximum utilisation metric (Table 12.7).

As one can see from the results, the topology significantly influences the performance of all traffic engineering strategies. We first address the question of how the topology influences the ranking of the strategies and next, how the topology influences the overall benefits of traffic engineering compared to shortest path routing.

The *ranking of the strategies* depends on the topology. While most strategies show similar behaviour for all topologies, the performance and ranking of $U_{max}U_{av}$, U_{av}, and $P_{av}L_{av}$ with respect to congestion costs depend strongly on the topology. $U_{max}U_{av}$ becomes the best strategy of all U_{max} based strategies for topologies like Colt and Artificial-2/3 and the worst of them for topologies like the DFN and C&W. The different parameters of the topologies (Table A.1) offer no clear explanationfor that. U_{av} and $P_{av}L_{av}$ show the same trend for the same topologies as $U_{max}U_{av}$. Looking at the maximum utilisation,

Table 12.6 Normalised Congestion Costs for Different Topologies

	DFN		Deutsche Telekom		Colt		C&W	
	Average	Confidence Interval	Average	Confidence Interval	Average	Confidence Interval	Average	Confidence Interval
SP	*1.00*	–	*1.00*	–	*1.00*	–	*1.00*	-
CC	*0.86*	0.02	*0.94*	0.05	*0.87*	0.02	*0.85*	0.01
CC_{uw}	*0.89*	0.02	*0.94*	0.05	*0.89*	0.02	*0.88*	0.02
U_{max}	*0.99*	0.01	*0.97*	0.05	*0.99*	0.01	*1.00*	0.01
$U_{max}L_{av}$	*0.98*	0.01	*0.95*	0.05	*1.00*	0.01	*1.00*	0.02
$U_{max}P_{av}$	*1.04*	0.05	*0.94*	0.05	*0.99*	0.1	*1.02*	0.04
$U_{max}U_{av}$	*1.27*	0.09	*0.94*	0.05	*0.96*	0.02	*1.20*	0.06
U_{av}	*2.59*	0.04	*0.99*	0.06	*1.09*	0.05	*2.14*	0.19
$U_{av}P_{av}$	*1.02*	0.03	*0.99*	0.07	*1.05*	0.03	*1.36*	0.10
$P_{av}L_{av}$	*1.83*	0.03	*0.97*	0.06	*1.30*	0.07	*1.67*	0.14
L_{av}	*1.00*	0.00	*0.96*	0.06	*1.14*	0.06	*1.11*	0.07

	SWITCH		Artificial-1		Artificial-2		Artificial-3	
	Average	Confidence Interval	Average	Confidence Interval	Average	Confidence Interval	Average	Confidence Interval
SP	*1.00*	–	*1.00*	–	*1.00*	–	*1.00*	-
CC	*0.91*	0.02	*0.91*	0.02	*0.79*	0.02	*0.78*	0.02
CC_{uw}	*0.94*	0.02	*0.94*	0.02	*0.82*	0.02	*0.79*	0.02
U_{max}	*1.07*	0.07	*1.01*	0.03	*0.94*	0.08	*0.89*	0.03
$U_{max}L_{av}$	*0.99*	0.01	*0.97*	0.02	*0.94*	0.03	*0.91*	0.02
$U_{max}P_{av}$	*0.98*	0.01	*0.96*	0.02	*0.96*	0.02	*0.83*	0.02
$U_{max}U_{av}$	*1.16*	0.03	*1.00*	0.02	*0.90*	0.03	*0.87*	0.02
U_{av}	*1.67*	0.13	*1.08*	0.05	*1.00*	0.10	*0.94*	0.03
$U_{av}P_{av}$	*1.07*	0.07	*1.01*	0.03	*0.94*	0.08	*0.89*	0.03
L_{av}	*1.00*	0.00	*1.23*	0.08	*1.37*	0.10	*1.07*	0.06
$P_{av}L_{av}$	*1.05*	0.03	*1.31*	0.07	*1.47*	0.11	*1.06*	0.04

the Deutsche Telekom topology shows a very low overall utilisation because its very small size (see Table A.1)leads to sufficient bandwidth on most links in the first step of the bandwidth assignment, see Section 12.3. This stresses that – as in every experiment based on randomly generated traffic – it is important to vary the generation method. We do so in the next section.

Besides that, the maximum utilisation results also show that different topologies have different potentials for optimisations. The SP strategy has a maximum utilisation close to 80% in all topologies (except Deutsche Telekom). The U_{max} strategies can reduce the maximum utilisation by 2% to 7% depending on the topology.

CC remains the best overall strategy for all topologies, it reduces congestion by 6% to 22%. For some of the topologies, it also leads to the optimal maximal utilisations and in that respect is always better than SP.

Table 12.7 Maximum Utilisation for Different Topologies

	DFN		Deutsche Telekom		Colt		C&W	
	Average	Confidence Interval	Average	Confidence Interval	Average	Confidence Interval	Average	Confidence Interval
SP	*0.80*	0.02	*0.53*	0.07	*0.80*	0.01	*0.80*	0.01
CC	*0.76*	0.03	*0.40*	0.05	*0.79*	0.01	*0.78*	0.02
CC_{uw}	*0.76*	0.03	*0.41*	0.05	*0.78*	0.01	*0.78*	0.02
U_{max}	*0.76*	0.02	*0.42*	0.07	*0.79*	0.01	*0.77*	0.02
$U_{max}L_{av}$	*0.76*	0.02	*0.39*	0.07	*0.77*	0.01	*0.76*	0.02
$U_{max}P_{av}$	*0.76*	0.03	*0.39*	0.05	*0.75*	0.02	*0.76*	0.02
$U_{max}U_{av}$	*0.76*	0.03	*0.39*	0.05	*0.75*	0.02	*0.76*	0.02
U_{av}	*0.95*	0.03	*0.44*	0.08	*0.89*	0.03	*0.98*	0.02
$U_{av}P_{av}$	*0.80*	0.02	*0.45*	0.08	*0.82*	0.01	*0.93*	0.03
$P_{av}L_{av}$	*0.94*	0.03	*0.44*	0.07	*0.98*	0.01	*0.97*	0.02
L_{av}	*0.94*	0.03	*0.44*	0.07	*0.98*	0.01	*0.97*	0.02

	SWITCH		Artificial-1		Artificial-2		Artificial-3	
	Average	Confidence Interval	Average	Confidence Interval	Average	Confidence Interval	Average	Confidence Interval
SP	*0.79*	0.01	*0.79*	0.01	*0.80*	0.01	*0.79*	0.01
CC	*0.78*	0.01	*0.77*	0.02	*0.76*	0.02	*0.73*	0.03
CC_{uw}	*0.78*	0.01	*0.77*	0.02	*0.76*	0.02	*0.73*	0.03
U_{max}	*0.78*	0.02	*0.76*	0.02	*0.76*	0.02	*0.73*	0.03
$U_{max}L_{av}$	*0.78*	0.02	*0.76*	0.02	*0.76*	0.02	*0.73*	0.03
$U_{max}P_{av}$	*0.78*	0.02	*0.76*	0.02	*0.76*	0.02	*0.73*	0.03
$U_{max}U_{av}$	*0.78*	0.02	*0.76*	0.02	*0.76*	0.02	*0.73*	0.03
U_{av}	*0.96*	0.02	*0.85*	0.04	*0.81*	0.04	*0.83*	0.04
$U_{av}P_{av}$	*0.82*	0.03	*0.83*	0.04	*0.80*	0.03	*0.81*	0.04
$P_{av}L_{av}$	*0.87*	0.04	*0.99*	0.01	*0.97*	0.01	*0.93*	0.03
L_{av}	*0.79*	0.01	*0.96*	0.03	*0.95*	0.02	*0.92*	0.04

12.5.4 Variation of the Traffic Distribution

As has been pointed out before, the influence of the traffic distribution also has to be evaluated. The following variations to the procedure described in Section 12.3 were evaluated for a subset of all traffic engineering strategies:

1. Assignment of equal node weights for all nodes in the network.
   If equal node weights are assigned to all nodes, the traffic is spread more evenly among the topology than in the basic set-up.
   Table 12.8 depicts the results (Experiment Setup 1). The benefit of traffic engineering improves a lot if the traffic is spread more evenly among the topology. In that case, all strategies show far better performance than shortest path routing. The maximum utilisation is now almost half of that of the shortest path routing.

Table 12.8 Variation of the Traffic Distribution

Congestion Costs								
Experiment Set-up	Default		1		2		3	
Strategy	Average	Confidence Interval	Average	Confidence Interval	Average	Confidence Interval	Average	Confidence Interval
SP	*1.00*	–	*1.00*	–	*1.00*	–	*1.00*	–
CC	*0.86*	0.02	*0.81*	0.01	*0.90*	0.04	*0.90*	0.04
CC_{uw}	*0.89*	0.02	*0.81*	0.02	*0.90*	0.04	*0.90*	0.04
U_{max}	*0.99*	0.01	*0.84*	0.01	*0.97*	0.03	*0.98*	0.01
$U_{max}L_{av}$	*0.98*	0.01	*0.81*	0.01	*0.91*	0.03	*0.92*	0.03
$U_{max}P_{av}$	*1.04*	0.05	*0.81*	0.01	*0.91*	0.04	*0.90*	0.04
$U_{max}U_{av}$	*1.27*	0.09	*0.81*	0.01	*0.92*	0.04	*0.92*	0.03
$U_{av}P_{av}$	*1.02*	0.03	*0.81*	0.01	*0.91*	0.04	*0.91*	0.04

Maximum Utilisation								
Experiment Set-up	Default		1		2		3	
Strategy	Average	Confidence Interval	Average	Confidence Interval	Average	Confidence Interval	Average	Confidence Interval
SP	*0.80*	0.01	*0.75*	0.01	*0.70*	0.05	*0.68*	0.06
CC	*0.76*	0.03	*0.42*	0.01	*0.65*	0.06	*0.62*	0.08
CC_{uw}	*0.76*	0.03	*0.42*	0.01	*0.65*	0.06	*0.62*	0.08
U_{max}	*0.76*	0.03	*0.42*	0.01	*0.65*	0.05	*0.62*	0.08
$U_{max}L_{av}$	*0.76*	0.02	*0.42*	0.01	*0.65*	0.06	*0.62*	0.08
$U_{max}P_{av}$	*0.76*	0.03	*0.42*	0.01	*0.65*	0.06	*0.62*	0.08
$U_{max}U_{av}$	*0.76*	0.03	*0.42*	0.01	*0.65*	0.06	*0.62*	0.08
$U_{av}P_{av}$	*0.80*	0.02	*0.49*	0.01	*0.65*	0.05	*0.63*	0.07

The behaviour is explained by the fact that if node weights differ, the flow between two different node pairs can differ by a great amount. If that is the case, the bandwidths of the links of the network are also likely to differ to some extent as we assumed the network to be roughly adapted to the traffic. The differing flows and link bandwidths limit the re-routing of flows as large flows can only be re-routed to a great extent on other high-bandwidth links. This limits the traffic engineering potential in the case of different node weights and explains the observed behaviour.

2. Assignment of equal bandwidth to all links.
   In a different set-up, we assign all links equal bandwidth. This removes possible advantages for the SP strategy because the bandwidth assignment process in the basic set-up used the shortest paths to derive reasonable bandwidth settings.
   The results are shown in Table 12.8 (Experiment Set-up 2). All traffic engineering strategies now show very similar performances, the congestion can be reduced by 10%, the maximum utilisation by 5%. The now smaller advantages of the CC strategies compared to the others with respect to the congestion is explained by the fact that due to the different setting of bandwidth the network is now less utilised on average. This is also visible from the maximum utilisation values of the SP strategy. Because of

the lower overall utilisation and the exponential shape of the congestion cost function CC has less advantages and the performance differences between the strategies are smaller.

3. Bandwidth assignment based on the *EQMP* (equal cost multipath) strategy instead of the SP strategy.
   A possible bias towards SP can be analysed by replacing SP in the creation process with a different strategy, in this case *EQMP*.
   The results for $EQMP$ with $n = 3$ paths are shown in Table 12.8 (Experiment Set-up 3). A behaviour similar to that in experiment set-up 1 can be observed, albeit not as extreme. The explanation is similar; flows are now assumed to be spread over the three shortest paths for the bandwidth calculation which creates a more even traffic distribution leading to the effects observed and explained above.

12.5.5 Conclusions

As a conclusion of the performance evaluation we recommend the CC strategy for traffic engineering as its overall performance is better than that of the other strategies under all evaluated circumstances. It optimises the congestion costs that we deem the most important metric. The congestion costs consider all links but – because of the convex shape – higher utilised links influence the routing decision more. Also, we recommend the use of the weighted congestion costs with the link bandwidth because high-bandwidth links are likely to be used by more users and should thus have more influence on the routing decision. Therefore, the weighted congestion costs were used in this section as default. Other strategies try minimising the maximum or average utilisation, the average load or the path length or a combination of these objectives and did not perform well in all experiments.

12.6 Singlepath versus Multipath

So far, the evaluation was focused on the multipath strategies that were allowed to split a flow in order to be routed on multiple different paths through the network. Contrary to that, singlepath strategies route one traffic flow on a single path through the network. As the solution space of the singlepath strategies is a subset of the multipath solution space, singlepath solution strategies can never show a better performance with respect to the objective function than the corresponding multipath strategy. In this section, we evaluate the performance loss for the traffic engineering strategies. We focus on the congestion costs and maximum utilisation, as the other metrics did not show a significant difference.

The relative difference in congestion costs and maximum utilisation of the singlepath variants of the previously discussed traffic engineering compared to the multipath solution is presented in Table 12.9 for different topologies.

The singlepath CC strategy shows a very small and almost negligible performance loss compared to the multipath CC strategy. The largest performance loss is 0.46%, occurring at the relatively small Telekom topology. For the larger topologies, the performance loss is below 0.06%.

The performance loss of CC_{uw} is of the same order of magnitude. For the U_{max} strategies, the maximum utilisation only increases by less than 0.01%, that performance

Table 12.9 Relative Difference in *Congestion Costs* and *Maximum Utilisation* of the Singlepath Strategy Compared to the Multipath Strategies for Different Topologies

Strategy	DFN (%)	Deutsche Telekom	Colt Telekom (%)	Cable & Wireless (%)	Artificial-2 (%)
		Congestion Costs			
CC	0.03	0.46	0.01	0.06	0.01
CC_{uw}	0.34	0.52	0.11	0.36	0.14
U_{max}	7.61	7.19	2.03	9.76	−12.39
$U_{max}L_{av}$	1.03	0.27	−0.62	−0.31	−3.51
$U_{max}P_{av}$	−6.88	2.22	−0.84	−6.14	9.80
$U_{max}U_{av}$	−1.23	1.69	−0.16	−1.39	−0.66

Strategy	DFN (%)	Deutsche Telekom (%)	Colt Telekom (%)	Cable & Wireless (%)	Artificial-2 (%)
		Maximum Utilisation			
CC	0.01	0.03	0.00	0.00	0.00
CC_{uw}	0.01	0.03	0.00	0.00	0.00
U_{max}	0.00	0.00	0.00	0.00	0.00
$U_{max}L_{av}$	0.00	0.00	0.00	0.00	0.00
$U_{max}P_{av}$	0.00	0.00	0.00	0.00	0.00
$U_{max}U_{av}$	0.00	0.00	0.00	0.00	0.00

loss is negligible. However, if the congestion costs are evaluated for these strategies, it becomes obvious that the singlepath and multipath solutions differ in their routing. The congestion costs are influenced randomly by the singlepath routing variant, because they are not optimised by the U_{max} strategies directly. Depending on the strategy and topology, they can significantly improve the congestion situation. Despite this effect, the CC strategies still always perform significantly better than the U_{max} strategies.

To summarise, the performance loss of singlepath strategies compared to multipath strategies is negligible. The only drawback of the singlepath strategies is therefore the fact that they need more time to solve (see Table 12.1 and 12.2), as the singlepath MIP models use binary variables.

12.7 Influence of the Set of Paths

The path selection strategies use a precomputed set of paths for their optimisation. In this section, the influence of this set of paths on the performance of the path selection strategies is evaluated.

Two parameters (n and Δl) are used to precompute the paths for each node pair. Parameter n is the upper bound on the number of paths that are taken into account. Parameter Δl denotes the maximum number of additional hops compared to the shortest path that are allowed for paths in the set.

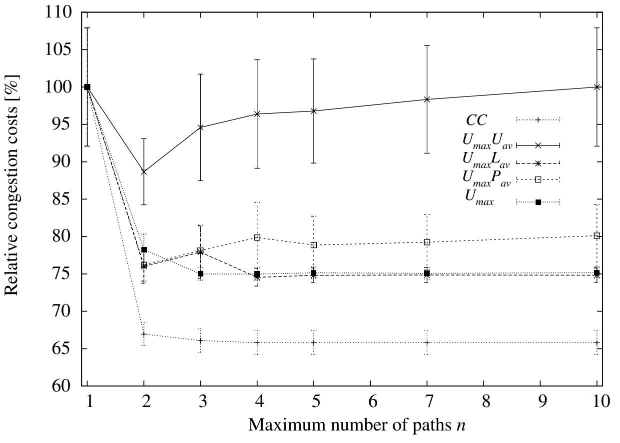


Figure 12.3 Influence of n on the Performance

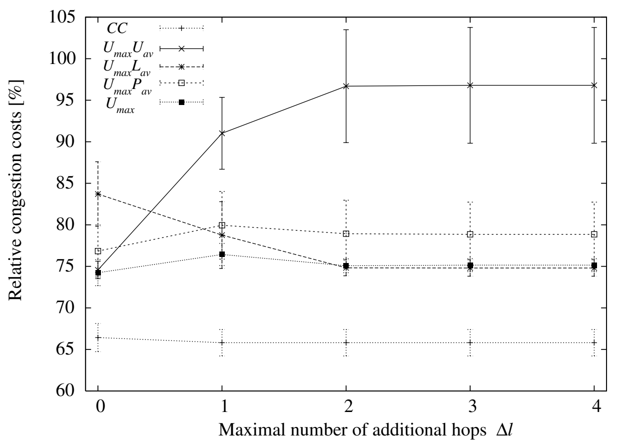


Figure 12.4 Influence of Δl on the Performance

The results are depicted in Figures 12.3 and 12.4. Figure 12.3 shows the congestion costs for the DFN topology and for different traffic engineering strategy. The maximum number of paths n is shown on the x-axis. It can be seen that the CC strategy clearly improves if n is increased. This can be expected. The largest performance increase occurs if n is increased from 1 – where all path selection strategies are effectively identical to the SP strategy – to 2. After that, the performance increase is significantly smaller.

Figure 12.4 shows the performance change if Δl is increased (for $n = 5$). The performance gain of CC is very small. This can be expected, as the previous experiments have already shown that the CC strategy does not tend to increase the average path length very much – therefore it does not make much use of the additional (longer) paths.

An important question to answer is what the optimal settings are for n and Δl. As the performance increase for values of $n > 5$ and $\Delta l > 2$ is negligible for CC, we recommend 5 and 2 for n and Δl. Higher values only lead to more computational complexity.

12.8 Summary and Conclusions

In this chapter, different traffic engineering strategies were discussed. They can be distinguished as path selection and explicit routing models. Explicit routing models show a very small performance advantage at the cost of computational complexity that prohibits their use for large networks. Path selection strategies can be computed much faster, are also more flexible, and offer the decision maker more control as they use a set of precomputed paths.

Traffic engineering strategies can also be distinguished into singlepath and multipath strategies, depending on whether they can split a flow into subflows and route them over different paths through the network. Multipath strategies have a theoretical performance advantage. In our experiments, it turned out that this advantage is extremely small for realistic topologies.

We introduced different metrics for measuring the performance of traffic engineering. Naturally, it makes sense to use these metrics as objective functions for the traffic engineering strategies. We did so for the path selection strategies. We argued that the congestion costs are the best performance metric. The strategies were evaluated in extensive simulations during which we investigated different topologies, different congestion cost functions, and traffic distributions. Throughout all these experiments, the CC strategy showed the best overall performance. Contrary to most other strategies, it performed well for practically all performance metrics. It can therefore be recommended without doubt. The other strategies showed flaws and bad performance in some or many cases and cannot therefore be recommended.

Using the correct strategy, traffic engineering can reduce the congestion of a highly loaded network and therefore directly improve the QoS. This advantage can also be used to increase the efficiency because more traffic can be served with the same capacity; correspondingly capacity expansions can be delayed and costs saved. This effect is also visible in the next chapter where capacity expansion is discussed. However, for several topologies and traffic distributions the advantages were rather small compared to the much simpler (and expectedly cheaper) solution of simply using shortest path routing. Therefore, traffic engineering cannot be recommended generally; an Internet Network Service Provider (INSP) has to carefully weigh the benefit of the increased QoS against the additional costs for the traffic management equipment, costs for staff and training, etc.