Part IV

Traffic and Network Engineering

The Competitive Internet Service Provider: Network Architecture, Interconnection, Traffic Engineering and Network Design
Oliver Heckmann ISBN: 0-470-01293-5

11

Traffic and Network Engineering Overview

So far in this book, the network architecture and the interconnections were analysed. Apart from choosing and tuning a network architecture and connecting its network with selected other Internet Network Service Providers (INSPs), an INSP also has to engineer its network: Nodes (Points of Presences (POPs) and routers) have to be connected by links, and these links have to be dimensioned and upgraded at regular intervals. Furthermore, with traffic engineering, the routing of traffic flows through the network can be influenced to increase the performance of the network. In this part, we investigate these engineering measures. We call the long-term engineering measures that influence the topology – for example, the link bandwidth – *network engineering*. State of the art in network engineering is discussed in Section 11.1. The medium-term engineering measures that assume the topology to be fixed and instead influence the routing of the flows through the network are called *traffic engineering* and discussed in Section 11.2. Traffic engineering and network engineering algorithms use traffic predictions between node pairs as input in the form of a traffic matrix. Because of their relevance to both topics, traffic matrices are discussed separately in Section 11.3.

11.1 Network Design and Network Engineering

In works related to network and traffic engineering, the term commodity is often found in the literature. With respect to IP networks, a *commodity* is a traffic flow between a specific pair of nodes. To be consistent in terminology with the rest of this book, the term *flow* is preferred to *commodity* here. The size of that flow is normally given as an entry in a traffic matrix.

With respect to routing, these works typically distinguish between non-bifurcated and bifurcated routing. Non-bifurcated routing (also called *singlepath routing)* implies that a flow, or commodity, is routed over a single path and cannot be split up to be routed over multiple paths. The latter is allowed if bifurcated or *multipath routing* is used.

The Competitive Internet Service Provider: Network Architecture, Interconnection, Traffic Engineering and Network Design
Oliver Heckmann ISBN: 0-470-01293-5

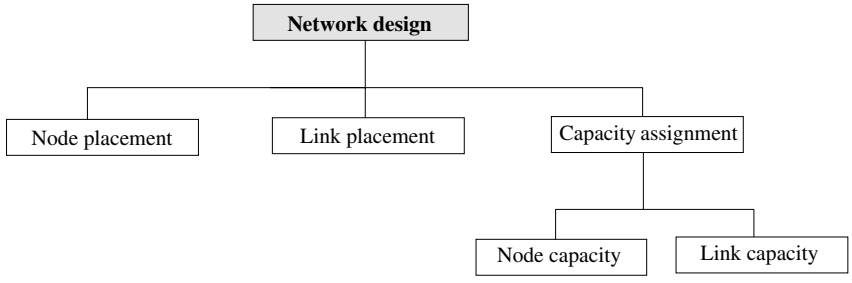

Figure 11.1 Network Design

11.1.1 Network Design

We distinguish between network design and network engineering. *Network design* (see Figure 11.1) is concerned with synthesising a new network topology. Network design consists of three parts: Node placement, link placement and capacity assignment (to nodes and links). The *node placement* sub-problem is about geographically placing the nodes of the topology that resemble the POPs of the INSP. The *link placement* sub-problem deals with connecting the nodes with each other while the *capacity assignment* problem assigns capacities (bandwidth, buffer, etc.) to the nodes and links. For designing a completely new topology, all three of these sub-problems have to be solved. Existing works often treat only a subset of these optimisation problems. The node placement especially is often assumed to be given and fixed.

Bley *et al*. (2004) describe how the GWiN backbone of the German Research Network Deutsches Forschungsnetz DFN (see Figure A.1 in the Appendix) was designed with a 2-level hierarchical approach. The set of nodes is given, so no node placement problem has to be solved. Each node becomes either an access node that is connected to a single backbone node or it becomes a backbone node that can be connected to any other node. For the capacity assignment process, a discrete list of nodes and link configurations are used.

Plain (non-bifurcated) shortest-path routing is used in that network. This fact can be exploited to simplify the mathematical programming model that results from link placement and capacity assignment. Using a Lagrangian relaxation, the optimisation problem can be split into two sub-problems: One is finding a valid network structure and hardware installation; it can be formulated as a mixed integer programming problem and solved with standard methods. The second one is the routing problem that can be solved efficiently by any shortest-path algorithm. With this approach, the design optimisation problem was solved for the size of the GWiN topology in 15 minutes on a standard PC with an optimality gap of less than 0.6%.

A classic network design paper is that of Gavish (1992). That work contains a general overview of network design. In addition, an approach to simultaneously solve all three network design sub-problems for given end-user locations and a given traffic matrix is presented. For node placement, a set of possible candidates for the backbone node locations is assumed to be given; the chosen nodes are connected by links that lead to fixed and traffic dependent costs. Besides that, a static singlepath routing scheme for the flows is derived. Quality of Service (QoS) is accounted for by including the

delay between end-user nodes into the objective function. This, however, makes the resulting combinatorial optimisation problem non-linear. With a Lagrangian approach, the optimisation problem can be split into sub-problems that can be solved more easily and lead to a lower bound of the overall optimisation problem. This bound can be used to calculate the optimality gap for feasible solutions, which can be obtained by the simple heuristics described in the paper.

There is a vast amount of other works regarding network design. Han *et al.* (2000) present an approach with a more realistic cost function for the links; that function is sequence of steps as function of the link capacity (similar to the Internet Exchange Point (IXP) cost function used for the interconnection optimisation problem in Chapter 10; see Figure 10.1). The authors show that the optimisation problem can be reformulated into a simpler optimisation problem where each link is replaced by multiple links with constant costs and a capacity limit. Genetic algorithms have been successfully used to solve network design problems for example, in Berry *et al.* (1998) and Palmer and Kershenbaum (1995); a combined genetic algorithm and linear programming approach is presented in Berry *et al.* (1999). Simulated annealing is used in Randall *et al.* (2000) and tabu search in Glover and Laguna (1998). For more works, we refer to the related work cited in the references above and to the standard network design book Kershenbaum (1993).

11.1.2 Network Engineering

Contrary to network design problems that are about the synthesis of a *new* topology, network engineering is about improving an *existing* network topology either by changing nodes and/or links (*structural engineering*) or by expanding the capacity of an existing and otherwise unchanged network (*capacity expansion*).

New networks have to be designed only rarely as practically all INSPs already have existing networks. Therefore, network engineering is a more frequent and important challenge for INSPs. Traffic volumes are growing by 70–150% per year; see Odlyzko (2003). The bandwidth of a network has to be doubled roughly every year to keep pace with these rates. This leads to the conclusion that capacity expansion – especially link capacity expansion – is the most important of all network engineering challenges. Later in this part, we will therefore place the focus on link capacity expansion.

Hasslinger and Schnitter (2004) investigate link capacity expansion and traffic engineering for IP networks. On the basis of their experience with the IP backbone to Deutsche Telekom, they report capacity increase factors ranging to beyond a factor of 2 per year. They present a capacity expansion heuristic that takes into account the influence of traffic engineering on the network utilisation. Their work is similar to our experiments in Section 13.2 and discussed in that context.

Optimally expanding telecommunication network facilities have been studied in a number of works; for an overview see Chang and Gavish (1993, 1995) and Dutta and Lim (1992). Chang and Gavish (1993, 1995) present a Lagrangian decomposition approach for a rather complex network engineering problem for telecommunication networks. The approach is well suited to derive a development plan towards a given target network in a certain number of periods. The solved optimisation problem is a combined structural engineering and capacity expansion problem; nodes are considered to be given and fixed but links can be placed and upgraded. The objective is to minimise the net present worth

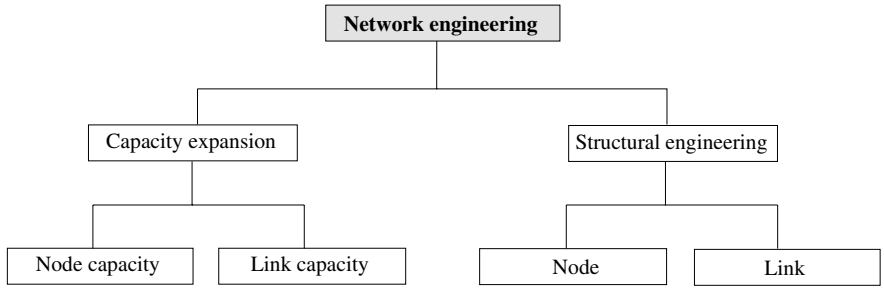

Figure 11.2 Network Engineering

of total invested costs for the given number of periods. This is contrary to our approach in Section 13.2, where the interest costs for the capacity expansion and fictive congestion costs are evaluated.

Chang and Gavish (1993, 1995) considered the fixed costs for installing a conduit for a bidirectional link between two nodes, fixed costs for upgrading the capacity of a link, and the capacity costs themselves. This leads to possible cost savings by installing excess capacity in a current period to avoid the fixed costs of later periods. Capacity is modelled by a continuous variable. The cost model and the continuous capacities are tailored for telecommunication providers and carriers but they are not suited for INSPs that typically lease the lines for their links from carriers at discrete capacities.

Dutta and Lim (1992) studied the installation of transmission capacity over time in a communication network where the nodes and the possible links are given; new nodes can be added over time but the decision of which nodes to add is not modelled. The optimisation problem is thus a combined link structural engineering and link capacity expansion problem (see Figure 11.2). Considered costs are the one-time installation costs and the per-period operation costs for links. The latter cost terms are assumed to exhibit economies of scale. The objective is to minimise the net present worth of total costs. Discrete capacities are modelled. A performance constraint based on the delay of a M/M/1 queue is also included in the model. The model is finally solved with a Lagrangian approach.

An interesting comparison of the bandwidth market and the financial market is made in d' Halluin *et al*. (2002). In that paper, capacity expansion under demand uncertainty is studied with modern financial option pricing methods. The perspective is that of a carrier that faces extremely volatile future revenues. The paper can help in explaining the current overcapacity in available bandwidth but cannot directly be transferred to the capacity expansion of INSPs that typically go to satisfy a relative constant increase in traffic volumes.

11.2 Traffic Engineering

The IETF Traffic Engineering Working Group gives the following definition of traffic engineering[1]: *Internet traffic engineering is defined as that aspect of Internet network*

[1] See http://www.ietf.org/html.charters/tewg-charter.html.

engineering concerned with the performance optimisation of traffic handling in operational networks, with the focus of the optimisation being minimised over-utilisation of capacity when the other capacity is available in the network.

Traffic engineering influences the forwarding decision of the routers with a specific goal in mind; it could for example, re-route flows so that they avoid a known bottleneck. Traffic engineering is basically an optimisation problem; the traffic engineering goal reflects itself in the objective function. In Chapter 12 of this book, different traffic engineering algorithms and different objective functions are discussed and evaluated.

If a plain IP forwarding architecture (see Section 6.3.1) is used, traffic engineering can be done by influencing the link weights of the routing protocol (see Section 6.4.1). Multiprotocol Label Switching (MPLS) as forwarding architecture (see Section 6.3.2) directly supports traffic engineering because it allows the creation of label switched paths independent of the routing protocol. It is therefore the preferred choice for traffic engineering.

The most straightforward *online algorithms* for routing traffic flows are based on the shortest-path algorithms such as Dijkstra (1959). The routing of flows is determined sequentially for all flows; a flow is routed on its shortest path where only links that have sufficient residual (remaining) capacity are considered. This type of algorithm can create bottlenecks and lead to underutilisation; see Suri *et al*. (2003).

A variant of the shortest-path algorithm called widest-shortest path is presented in Guérin *et al*. (1997). Here, the smallest residual link capacity of a path is maximised. The impact on other flows is neglected and still, bottlenecks can occur as shown in Suri *et al*. (2003).

The minimum interference routing algorithm of Kodialam and Lakshman (2000) takes the impact that the decision to route a flow on a certain path has on the maximum flow routable between other node pairs into account; this is called *interference*. An advanced version of this algorithm is presented in Suri *et al*. (2003); they use the solution of an off-line multicommodity flow problem (see the following text) based on an estimated traffic matrix as guidance for the on-line routing algorithm.

The term *multicommodity flow problem* designates a class of optimisation problems that is used as the basis for many off-line traffic engineering algorithms; see for example, Ahuja *et al*. (1993); Gondran and Minoux (1984); Leighton *et al*. (1995); McBride (1998) and Stein (1992). In a capacitated graph, multiple commodities (demand, traffic flows) have to be routed. A commodity is defined by a source and destination node, a size and in some cases a revenue. The multicommodity flow problem is therefore a generalisation of the well-known maximum flow problem as described by Ford and Fulkerson (1956).

For the typical type of multicommodity flow problems, the objective is to find the routing for a subset of all commodities that conforms to the capacity of the network and maximises the revenue obtained from the routed commodities. Solution algorithms thus route traffic and impose admission control on the flows. We call this class of problems the *revenue maximising multicommodity flow problems* or traditional multicommodity flow problems. However, INSPs will often have a network of sufficient capacity or will not have the possibility of admission control, for example, because they use an overprovisioned best-effort network. In that case, the optimisation problem is different: Route the traffic flows through the network so that the general QoS

is maximised. We name this type of problem the *QoS maximising multicommodity flow problem*.

In the *path selection formulation* of multicommodity flow problems, the possible paths for a commodity/flow are given; typically they are determined in a preprocessing step before the actual optimisation. In a multiservice network, for flows with strict delay requirements typically only very short paths are considered while for flows with less strict delay more and longer paths can also be considered. In the *explicit routing formulation*, sometimes also called link-based formulation, no paths are precalculated, they are calculated during the optimisation process. We present an experimental evaluation of these two formulations in Section 12.4.

In the *singlepath formulation* of a multicommodity flow problem, a commodity/flow must be routed along a single path (non-bifurcated singlepath routing); this formulation is also often called integer formulation because as a combinatorial optimisation problem it can be modelled as a MIP (mixed integer program). Another formulation is the *multipath formulation*; here, a commodity can be split up along multiple paths (bifurcated multipath routing). This problem can be formulated as a linear programming model and can thus be solved in polynomial time.

In Chapter 12, the range of QoS maximising multicommodity flow problems are modelled and evaluated as LP/MIP optimisation problems in path selection and explicit routing formulation as well as singlepath and multipath formulation.

The work of Mitra and Ramakrishnan (2001) is based on the *revenue maximising multicommodity flow problem*. Two service classes are considered (QoS and best-effort). The revenue is modelled linear to the amount of carried data. For the QoS traffic, possible paths are precalculated while the best-effort flows can be routed freely through the network. The QoS traffic is routed through the network first, the best-effort traffic is then routed based on the remaining bandwidth. This complex, combined, optimisation problem is decomposed into three layered sub-problems and a scalable solution algorithm is presented in Mitra and Ramakrishnan (2001).

Bessler (2002) extends the multicommodity flow problem to multiple periods by considering the changes to the existing LSPs of the previous period. The idea is to reduce the number of changes by penalising them in the objective function as they lead to signalling overhead and a risk of service disruptions. Another work considering the trade-off between the network utilisation and the signalling/processing overhead is by Scoglio *et al*. (2001).

The multicommodity flow problem is extended with an auction-based mechanism in Bessler and Reichl (2003). Here, a bid is associated with each commodity/flow and bandwidth is distributed according to the bid order.

For the *QoS maximising multicommodity flow problems*, there are different approaches to formulate the objective function. A typical approach is to minimise the bottleneck utilisation of the network, see for example, Hasslinger and Schnitter (2002a,b); Lin and Wang (1993); Poppe *et al*. (2000) and Roughan *et al*. (2003). The motivation behind that is the fact that the QoS a flow receives is mostly influenced by the bottleneck it passes through; the utilisation of the bottleneck of a network thus determines the worst performance that flows can receive. In the next chapter, we evaluate different objective functions for the traffic routing problem and show that a congestion cost function should be preferred as objective function; for OSPF routing such a congestion cost function is used in Fortz and Thorup (2002).

Hasslinger and Schnitter (2002a,b) investigate the QoS maximising multicommodity flow problem, minimising the bottleneck utilisation. Besides solving the optimisation problem with LP/MIP methods or applying the max-flow-min-cut principle, the authors present a heuristic for the singlepath formulation of the problem based on simulated annealing. In simulations, the authors show that the maximum utilisation can be decreased by up to 42.4% compared to the utilisation with the shortest path routing. Also, the simulations indicate that the benefit of multipath routing over singlepath routing is rather low; this can also be observed in our experiments in the next chapter.

In Poppe *et al.* (2000), traffic engineering for a network with Diffserv Expedited Forwarding (EF) traffic and best-effort traffic is studied. Two traffic engineering optimisation problems are solved for the different traffic classes. For the EF traffic, the maximum utilisation is minimised as primary objective and the average load as secondary. An equivalent traffic model is also included in our experiments in the next chapter. For the best-effort traffic, fairness is maximised as the primary objective and the throughput as the secondary one. It can be argued how important fairness is for a profit-maximising INSP.

The results of the paper are that traffic engineering can significantly improve the traffic handling capabilities of a network. The findings of that paper also show that the results improve only a little if multipath routing is used instead of singlepath routing and that in the multipath case only very few flows are actually split up and routed among multiple paths. These results are consistent with our results that will be presented in the next chapter.

The off-line traffic engineering methods use a traffic matrix as input to determine the routing. In some cases, the traffic matrix can be known exactly; for example, in Diffserv networks, when only flows are considered for which SLAs exists (see Section 6.2.4.2) or in networks with reservation in advance. Normally, however, the traffic matrix is not known exactly and has to be estimated based on measurements. Traffic matrix estimation is a challenge for INSPs and discussed in detail in the next section. Roughan *et al.* (2003) asks the important question: *If traffic engineering is done based on the estimated traffic matrix, how well does it perform on the real traffic matrix?* They use the maximum utilisation as objective function for the traffic engineering algorithm, optimise the routing based on an estimated traffic matrix and verify the performance based on the real traffic matrix. The results indicate that OSPF weight optimisation combined with tomographic traffic matrix estimation (see below) performs very well, mainly because OSPF optimisation was robust to the errors found in the traffic matrix estimation. The MPLS style optimisation can determine better routing schemes but is also less robust according to Roughan *et al.* (2003).

11.3 Traffic Matrix Estimation

A traffic matrix M describes the average rate r_{ij} for a given time interval between the ingress nodes i and egress nodes j of a network.

$$M = \begin{bmatrix} \cdots & \cdots & \cdots & \cdots \\ \cdots & r_{i\,j-1} & \cdots & \cdots \\ \cdots & r_{ij} & r_{i+1\,j} & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{bmatrix}$$

Traffic matrices form the input for network design and traffic engineering optimisation problems. Therefore, it is important to determine traffic matrices in real networks. However, measuring a traffic matrix is not a trivial task. Benameur and Roberts (2002) give an overview over the two distinct approaches to measure a traffic matrix: The *direct measurement* approach as advocated by Feldmann *et al*. (2000) uses NetFlow (2004) to collect flow information. This information is evaluated off-line to derive the traffic matrix using the routing tables active at the measurement time that also have to be recorded. This approach is storage space and router-Central Processing Unit (CPU) intensive and requires all routers to support NetFlow (or a similar product) but contrary to other approaches allows them to derive the point-to-multipoint traffic matrix. A point-to-point traffic matrix M models the traffic between ingress node i and egress j while the point-to-multipoint traffic matrix $\tilde{M}$ models the traffic and ingress node i and captures the fact that this traffic can exit at more than one egress j.

Another direct measurement approach that is less resource intensive is described by Schnitter and Horneffer (2004); it works for networks that employ label switching (MPLS). Every LSP has a byte-counter measuring the traffic using this LSP. Thus, if an MPLS network is built as a full mesh of LSPs, the traffic matrix can be measured directly. However, due to scalability and load balancing reasons, a full mesh of LSPs is not often used. The technique introduced in Schnitter and Horneffer (2004) can measure the traffic matrix directly if the router has a byte counter for each Forwarding Equivalence Class FEC[2]. It does not depend on the routing method (explicit LSPs with traffic engineering or plain shortest-path routing).

Most of the other works favour *deriving the traffic matrix from link measurements,* as they are more readily available for all router interfaces via SNMP (simple network management protocol) in production networks. The problem with this approach is that estimating the traffic matrix is an ill-posed inverse linear problem: In a network with N ingress/egress nodes, the traffic matrix size is $O(N^2)$ as it contains entries for each node pair. However, there are only $O(N)$ link measurements as the number of links is the average node degree times the number of nodes. Therefore, the problem becomes massively under-constrained for large N as the number of variables then exceeds the number of equations (if the problem is formulated as a linear equation system). To solve this problem, additional assumptions for example, about the traffic and the routing have to be made. Approaches to this problem can be classified into statistical tomographic methods, optimisation-based tomographic methods and other methods:

- The *Statistical tomographic methods* use higher order statistics of the link load data like the covariance between two loads to create additional constraints. Examples are Cao *et al*. (2000); Tebaldi and West (1998) and Vardi (1996). Vardi (1996) and Tebaldi and West (1998) assume a Poisson traffic model; Cao *et al*. (2000) assume a Gaussian traffic model.

[2] The exact requirements are that the statistics include each LSP through a router, incoming and outgoing labels, the FEC, the outgoing interface, and the byte counter. These requirements are fulfilled by the most common router operation systems like Cisco's Internet Operating System (IOS) and Juniper Network Operating System (JUNOS) (see Schnitter and Horneffer (2004)).

- The *Optimisation-based tomographic methods* select a solution out of the solution space of the under-constrained problem that optimises a certain objective function using methods like linear or quadratic programming. Goldschmidt (2000) is a simple example for this approach.
- Classified as *other methods* are approaches that combine the tomographic methods with other methods like gravity or choice models. Medina *et al.* (2002) use a logit choice model that captures the choices of users (where to download from) and network designers (how to interconnect the POPs). The decision process is modelled as a utility maximisation problem.
  Zhang *et al.* (2003a) combine a optimisation-based tomographic methods with a generalised gravity model. A gravity model can, for example, be used to estimate the traffic between edge links by assuming that the traffic between i and j is proportional to the total traffic entering at i multiplied with the total traffic exiting at j.
  Zhang *et al.* (2003b) uses an information theoretic approach that chooses the traffic matrix consistent with the measured data so that it is as close as possible to a model in which the source and destination pairs are independent and therefore the conditional probability $p(j|i)$ that source i sends traffic to j is equal to the probability $p(j)$ that the whole network sends traffic to j.

11.4 Summary and Conclusions

In this chapter, network design and network engineering were introduced. Network design is concerned with building and network engineering with upgrading a network. Then, traffic engineering was presented. It influences the routing of traffic flows through the network to increase the performance of the network. Traffic engineering and network engineering algorithms use traffic predictions between node pairs as input; we call this a traffic matrix. Ways for measuring and predicting the traffic matrix were discussed in the previous part of this chapter.

The rest of this part is structured as follows. In Chapter 12, the influence of traffic engineering on the QoS of a network and the costs and efficiency of that network are analysed. Traffic engineering strategies and performance metrics are discussed and then evaluated in a series of simulation experiments.

In Chapter 13, network engineering is discussed. The focus of that chapter lies on capacity expansion because providers have to expand the capacity of their network regularly – the Internet traffic has been growing exponentially in the last years (see Odlyzko (2003)) and there is no indication why this should not continue for the future. The capacity expansion problem therefore has to be solved much more often than the general design of a new network topology. According to the system-oriented approach of this book, we start by showing how network engineering and the network architecture in the form of QoS systems interact. Then we present and evaluate different strategies for capacity expansion. After that, we investigate the interaction between traffic engineering and capacity expansion strategies in further experiments. Finally, we investigate the elasticity of traffic matrices resulting from the elastic behaviour of TCP and the impact on capacity expansion in an analytical study.