Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

ANDREAS EISENBLÄTTER
ARIE M.C.A. KOSTER
RANDOLF WALLBAUM
ROLAND WESSÄLY

Load Balancing in Signaling Transfer Points

ZIB-Report 02-50 (December 2002)

Load Balancing in Signaling Transfer Points

Andreas Eisenblätter[*†] Arie M.C.A. Koster[†] Randolf Wallbaum[‡] Roland Wessäly[*†]

Abstract

Signaling is crucial to the operation of modern telecommunication networks. A breakdown in the signaling infrastructure typically causes customer service failures, incurs revenue losses, and hampers the company image. Therefore, the signaling network has to be highest reliability and survivability. This in particular holds for the routers in such a network, called *signaling transfer points* (STPs).

The robustness of an STP can be improved by equally distributing the load over the internal processing units. Several constraints have to be taken into account. The load of the links connected to a processing unit changes over time introducing an imbalance of the load.

In this paper, we show how integer linear programming can be applied to reduce the imbalance within an STP, while keeping the number of changes small. Two alternative models are presented. Computational experiments validate the integer programming approach in practice. The GSM network operator E-Plus saves substantial amounts of time and money by employing the proposed approach.

Keywords: Signaling network, STP, load-balancing, integer programming, computations
Mathematical Subject Classification (MSC 2000): 90B18, 90C90, 90C10

[*]Atesio GmbH, Rubensstraße 126, D-12157 Berlin. E-mail: {eisenblaetter, wessaely}@atesio.de

[†]Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), Takustraße 7, D-14195 Berlin. E-mail: {eisenblaetter, koster, wessaely}@zib.de

[‡]E-Plus Mobilfunk GmbH & Co. KG, E-Plus-Platz 1, D-40468 Düsseldorf. E-mail: Randolf.Wallbaum@eplus.de

1 Introduction

The signaling network is the nervous system of a telecommunication network, and it is vital to the operation of the telecommunication network itself. Signaling precedes and accompanies every communication in the telecommunication network. Without signaling, no communication is possible.

Although the required transmission capacity in the signaling network is not very large, the network has to be of highest reliability and survivability. A breakdown in the signaling network typically causes customer service failures, leads to loss in revenue, and often brings negative publicity to the network operator. Therefore, local overloads, the failure of single nodes, or the damage of a transmission line caused by earthworks should not disrupt signaling.

If, for example, the connection between a mobile service switching center (MSC) and a home location register (HLR) breaks down in a GSM network, then all customers whose service profile is maintained in that HLR cannot log into the network. If a customer is already logged in, then the customer cannot receive a call anymore. Nowadays, up to one million customers records can be maintained by one HLR. Another example is linked to the very popular short message service (SMS). In case the communication to the SMS center fails, then the reception and the transmission of SMSs becomes impossible in the entire network. Note that, SMSs are routed over the signaling network.

The design of the signaling network has, thus, to provision redundancy, and logical connections have to be realized along alternative, physically independent transport routes.

The introduction of each new service, such as high speed circuit-switched data (HSCSD) or general packet radio service (GPRS), adds to the importance of signaling. In a GPRS network the serving and gateway GPRS service nodes (SGSN/GGSN) are nodes in a signaling network.

The importance of signaling does not diminish with the introduction of third generation mobile telecommunication networks. In a UMTS network, all network components still need to exchange signaling information. Between the MSC servers and the HLR data bases, for example, this is done using a traditional SS7 signaling network. The SS7 protocol is a global standard for signaling in telecommunication networks defined by the International Telecommunication Union (ITU), the European Telecommunications Standards Institute (ETSI), and Third Generation Partnership Project (3GPP). Even if the transport technology between network components changes when ATM or IP is introduced, logical SS7 links have to be maintained.

Larger networks have routers for their SS7 signaling traffic. These routers are called *signaling transfer points* (STPs). An STP is an intelligent, high speed, reliable, special purpose packet switch for signaling messages. An arrangement of stand-alone STPs in mated pairs (at different locations) is commonly used in large GSM networks, see Figure 1. Each service switching point and service control point in the network is connected to multiple STPs, and the STPs themselves are interconnected with multiple links as well.

The availability of an SS7 network in case of local failures depends strongly on stand-by capacity. The stand-by capacity has to be available on each node and each link of the signaling network and, at least equally important, on the intermediate STPs. It is therefore customary to plan the signaling network in such a way that only up to 20% of the capacities are used in the failure free state, and within each STP, the routing load should be distributed equally over all processing units.

Balancing the load distribution within an STP is highly desirable from a network operator's point of view.

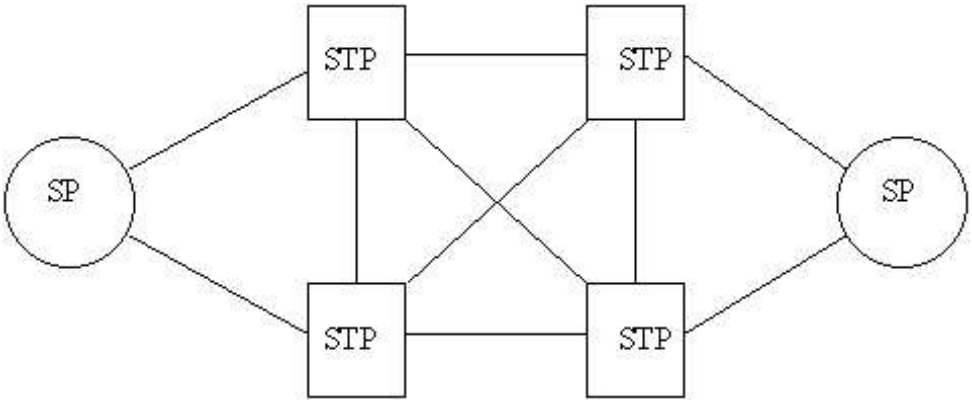


Figure 1: Signaling network with fully meshed STP core

In a balanced situation, the spare capacity is distributed evenly in order to allow for load fluctuation. Abrupt and sometimes significant load changes originate in link, linkset, or entire STP breakdowns. Smoother changes in the load come from the varying use of telecommunication services over the day, and from the steadily increasing use of such services altogether. A balanced load distribution also eases the management of an STP in case of software updates or link (-set) additions.

The small but irregular changes of the signaling load over time may cause an increasing imbalance of the STP. As a consequence, it is recommended to rebalance the load on a regular basis in order to avoid unacceptable imbalances. So far, this task is usually carried out manually. The many side constraints that have to be taken into account for a feasible STP configuration limits the effectiveness of such an approach, and makes it a very time consuming job.

In this paper, we describe how decisions to rebalance STPs can be supported by the use of discrete mathematics. We show that balanced STP configurations can be obtained from solving an integer linear program. After a first case study [3], a load balancing tool based on this approach has been implemented and is used very successfully in practice by the German GSM network operator E-Plus Mobilfunk GmbH & Co. KG. Without disclosing details, we mention the two most prominent effects. The strongly improved capability to balance the STPs directly yields a signaling network that is more robust to abrupt load shifts. This, in turn, allows E-Plus to delay planned expansions of the signaling network (and capital investments) without compromising on the established quality requirements.

This article is organized as follows. We describe the STP load balancing problem in full detail in Section 2. Sound mathematical (integer linear programming) models for (re-)optimizing the load balancing and several modeling alternatives are discussed in Section 3. In Section 4, we report on computational results for several realistic planning scenarios from the German mobile cellular network operator E-Plus.

2 Configuring an STP

An STP is a routing unit in a larger signaling network. Configuring an STP is one step in designing and maintaining a signaling network as a whole. Since the signaling network is the nervous system of a telecommunication network, a failure or breakdown in the signaling network can easily bring the entire telecommunication network to a stand-still.

Network operators spend significant amounts of money to acquire fault-tolerant soft- and hardware to operate in the signaling network. The careful configuration of this soft- and hardware adds to the robustness of the network. The configuration of each STP is certainly only a piece in the puzzle, but an important one. Our focus in the following is on the assignment of the links attached to an STP to its internal routing units.

2.1 Bits and Pieces

A STP is decomposed into clusters, which, in turn, consist of routing units and interface cards. In this paper, we denote the routing units as CCDs (for "Common Channel Distributors") and the interface cards as CCLKs (for "Common Channel Link Controllers"). Every link has to be connected to some CCD via a CCLK. A CCLK has a certain capacity, which determines the number of links that can be attached. The CCLKs within a STP are numbered consecutively. Although CCLKs and CCDs in different clusters are connected through system-bus, a link should be connected to a CCD via a CCLK within the same cluster in order to keep the internal traffic in the failure free state as small as possible. Every link is contained in some linkset. All links in a linkset have the same origin and destination, and they are all routed via the same STP. Each link carries signaling traffic, which is typically measured in milli-Erlang.

Figure 2 shows a schematic diagram of an exemplary STP with four clusters, each containing two or three CCDs and twenty CCLKs.

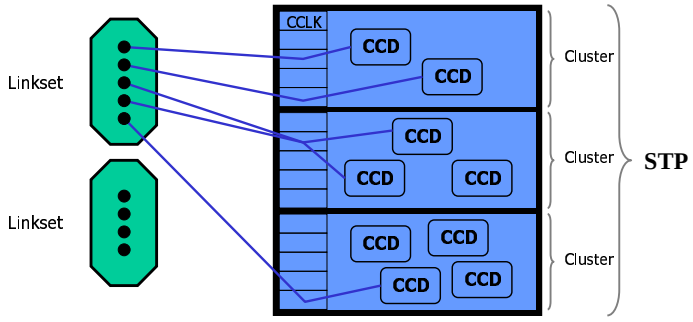


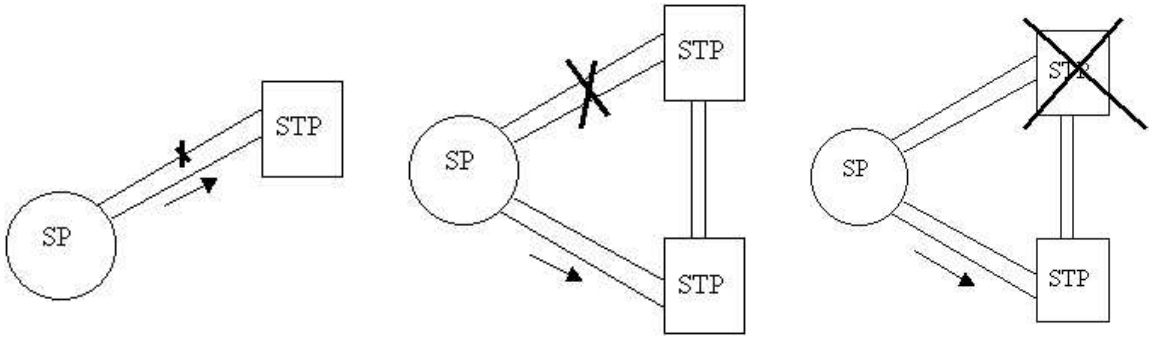
Figure 2: Schematic view of an exemplary STP

When we speak of an STP configuration, this is about which link is attached to which CCD via which CCLK. The load of a CCD is determined by adding the traffic load of all links that are routed by the CCD. Clearly, some configurations are more favorable than others. This is discussed next.

2.2 Load balancing is important

A few examples shall illustrate failures or changes we want to be prepare for and why the load balancing within an STP is important.

Single link failure. In case a SS7 link between two nodes fails (due to a physical disconnection or due to a bit error rate beyond the permissible maximum), see Figure 3(a), the remaining links from the linkset



(a) link (b) linkset (c) STP

Figure 3: Failure situations

take over the failing link's traffic. Acknowledgment procedures ensure that no message is lost during the changeover, because unacknowledged messages are repeated. Clearly, the traffic on the substituting links increases abruptly, and the load distribution at the STP serving the linksets changes abruptly as well. The load on the processors previously serving the failing link decreases, whereas the load on the processors serving the substituting links increases.

Linkset failure. In case all links from a linkset fail, then the direct route between the two endpoints is no longer available, see Figure 3(b), and an alternative route has to be established. Along this alternative route the traffic load jumps up, whereas it drops down along the original route. Again, no message should be lost due to the acknowledgment procedures.

STP failure. In case an entire STP fails despite all its build-in redundancy, this must not cause failures in the remaining network, see Figure 3(c). The breakdown of an STP causes all the linksets attached to the STP to fail, and all these linksets need to be substituted at once. On all substituting links and their STPs the traffic load increases abruptly.

Changing link loads. The traffic load on a link is not static. It varies with the daytime, but the extreme values change over time as well. New customers, for example, cause the signaling load to increase. When some threshold load value is reached, new links are added. Due to load sharing arguments, the number of links in a linkset ought to be a power of two. Hence, enlarging a linkset basically implies to double its size. After the doubling the load on each of its links is roughly cut in half. Not only new customers increase the demand for signaling. A much larger impact on the signaling load typically stems from new network services. Two prominent examples from GSM network are cost control function offered together with prepaid services or the "home zones" for mobile users.

2.3 Survivability and Maintainability

Our examples from the previous section demonstrate why a substantial amount of stand-by capacity is highly desirable for each CCD. In addition to the sliding changes accompanying the changes in signaling demand over time, failure situations cause instantaneous and drastic load shifts for STPs in general and for their CCDs in particular.

In practice, STP configurations are favored in which all CCDs have roughly the same load in the failure free state. Achieving a good load balance among the CCDs of an STP will be our optimization objective. There are four major side constraints to be taken into account.

Diversification: In case a cluster breaks down, then all links served by CCDs belonging to that cluster fail. To ensure that no linkset is affected too heavily by such a potential break down, at most half of a linkset may be connected to a single cluster.

CCLK parity: The CCLKs in an STP are numbered consecutively. They can be shut down for maintenance for each parity separately. For this reason, the CCLKs used by the links in a linkset shall be of odd and even parity in roughly equal parts.

No inter-cluster traffic: Although a bus-system within an STP allows to assign a link to a CCD via an arbitrary CCLK, it is preferred that a CCLK from the same cluster is used. The initial configuration of an STP usually has this property. Operational changes in the STP configuration may not always obey this preference, but it constitutes an added value of the re-optimization if it does so.

Configuration stability: A "greenfield" configuration of an STP is only possible at deployment. Every subsequent change of the configuration should take the existing configuration into account and keep the number of changes in the configuration small. Each configuration change results in a down-time of the corresponding link.

The STP load balance optimization has usually been performed manually. Equipped with a list of CCD load figures, the planner determines links for which the CCD assignment should be changed. With the growing size of the STPs, it becomes more and more difficult to manually meet all the side constraints just mentioned.

In the next section we present basic mathematical formulations for the load balance optimization problem of STPs and discuss alternatives to keep the number of required changes to the existing STP configuration small. These models form the mathematical core of an automatic STP load balancing tool.

3 Modeling

In this section we present a sequence of integer linear programming formulations for the load balancing within signaling transfer points.

Our first model, introduced in Section 3.2, is simple, but incomplete. We merely formalize what a feasible STP configuration is (neglecting a few practical side constraints). Sections 3.4 and 3.5 complete the basic model with two alternative objective functions. Assuming that there is already a configuration of the STP, we limit the allowable changes while searching for a configuration minimizing the load

difference in Section 3.4. In Section 3.5 we limit the maximal allowable load difference among CCDs and search for a new configuration achieving this bound and requiring the least changes in the current configuration.

In Section 3.6, the two previous models are augmented further. We incorporate the distribution of links within the same linkset between CCLKs of even and odd parity.

3.1 Notation

We use the following notation throughout this section.

l	a link
e_l	load of link l
L_s	linkset s
S	index set for linksets
$L = \bigcup_{s \in S} L_s$	set of all links
q	a cluster
Q	set of all clusters
n_q	total CCLKs capacity in cluster q
d	a CCD
D_q	set of CCDs contained in cluster q
$D = \bigcup_{q \in Q} D_q$	set of all CCDs

Recall that every link is contained in exactly one linkset and that every CCD is contained in exactly one cluster. Hence, $\bigcup_{s \in S} L_s$ and $\bigcup_{q \in Q} D_q$ are disjoint unions.

Each link $l \in L$ has to be assigned to exactly one CCD $d \in D$ in the final configuration c of an STP. Formally, c is a mapping $c\colon L \to D$, specifying for each link l the CCD $c(l)$ to use. Initially, the CCLK to use when attaching link l to CCD $d = c(l)$ is not specified at all. We merely ensure that no more than the available n_q CCLK ports are needed per cluster. Later, we will also take CCLK parity constraints into account. The details of when a configuration c is feasible are described next.

3.2 Feasible Configurations

Our first step towards an integer linear programming formulation for optimizing the configuration of an STP is to describe when a configuration is considered feasible. We do this in terms of indicator variables x_{ld}, which encode whether link l is assigned to CCD d by either being 0 or 1. Such an indicator variable is introduced for each possible link/CCD combination. A *feasible STP configuration* is encoded if the following constraints are met:

$$x_{ld} \in \{0,1\} \qquad \forall l \in L, \forall d \in D \tag{1}$$

$$\sum_{d \in D} x_{ld} = 1 \qquad \forall l \in L \tag{2}$$

$$\sum_{l \in L} \sum_{d \in D_q} x_{ld} \leq n_q \qquad \forall q \in Q \tag{3}$$

$$\sum_{l \in L_s} \sum_{d \in D_q} x_{ld} \leq \left\lceil \frac{|L_s|}{2} \right\rceil \qquad \forall s \in S, \forall q \in Q \tag{4}$$

The configuration is (y, z)-*loaded* if two further constraints are met:

$$\sum_{l \in L} e_l x_{ld} \geq y \qquad \forall d \in D \tag{5}$$

$$\sum_{l \in L} e_l x_{ld} \leq z \qquad \forall d \in D \tag{6}$$

Each of the above constraint sets is briefly explained. The constraints (1) simply impose that all x-variables are binary variables. Constraints (2) impose that each link is assigned to exactly one CCD. Constraints (3) ensure that no more links are assigned within a cluster q than the number n_q of available CCLK ports, thus, ensuring that no inter-cluster traffic is present in the failure free state. Constraints (4) reflect the diversification requirement that at most half of the links in a linkset are assigned to the same cluster. Hence, at most 50% of the links in a linkset are lost in case of a cluster failure. In case x satisfies the constraints (1)–(4) we write

$$x \in \text{FEASIBLE}.$$

Constraints (5) and (6) express that a feasible configuration is (y, z)-loaded if the load of every CCD lies between y and z. Recall that the load of a CCD is the sum over the loads of all assigned links. If x meets the constraints (5), (6) we note

$$x \in \text{LOAD}(y, z).$$

Clearly, each setting of the x-variables satisfying all the constraints (1)–(6) is in one-to-one correspondence to a (y, z)-loaded configuration c and vice versa.

3.3 Counting Changes

As indicated before, the operational burden for changing the existing configuration of a STP is strongly related to the number of links that change their CCD. We cope with this fact in different ways in the following, but in all cases we need to count those changes.

Given a configuration c^* in which each link l is assigned to one CCD $d = c^*(l)$, the number of different CCD assignments are counted by the expression $\sum_{l \in L} \sum_{d \in D: d \neq c^*(l)} x_{ld}$.

In case only changes for a subset L' of all existing links should be accounted for, then L' is used instead of L in the first sum. L' may, for example, contain all previously assigned links in the presence of new links.

Two alternative models are proposed next, which differ in how the trade-off between the required number of changes and reducing the load-imbalance is done.

3.4 Minimizing the Load Imbalance

In our first complete model for the STP reconfiguration optimization problem, we minimize the CCD load imbalance, *i. e.*, we minimize $z - y$, while bounding the number of allowable CCD changes by a parameter B. Formally, our optimization problem reads as follows.

$$\min_{x,y,z} \; z - y \tag{7a}$$

s. t.

$$\sum_{l \in L} \sum_{\substack{d \in D: \\ d \neq c^*(l)}} x_{ld} \leq B \tag{7b}$$

$$x \in \text{FEASIBLE} \cap \text{LOAD}(y, z) \tag{7c}$$

Clearly, the larger value of the parameter B, the smaller is the achievable load difference between the CCDs.

Kühn and Mayer [4] propose a model similar to (7). An underlying order of the CCDs is assumed, and it is imposed that the first CCD (according to that order) has the highest load whereas the last CCD has the smallest load. The objective is to minimize the load difference, but the number of allowable changes is not bounded. In fact, the additional restriction coupled to the order of the CCDs has the tendency to drive the number of required changes up. This is a clear drawback from a practical point of view.

3.5 Minimizing the Changes

In our second model, we reverse the priority for the STP reconfiguration optimization problem. The required number of CCD changes that is necessary to obtain a maximum prescribed load imbalance D is minimized. The second model is formally stated as:

$$\min_{x,y,z} \sum_{l \in L} \sum_{\substack{d \in D: \\ d \neq c^*(l)}} x_{ld} \tag{8a}$$

s. t.

$$z - y \leq D \tag{8b}$$

$$x \in \text{FEASIBLE} \cap \text{LOAD}(y, z) \tag{8c}$$

3.6 Keeping odd and even balanced

One important aspect concerning the ease of maintaining a STP has so far been neglected in the models. Our focus has been on balancing the loads among the CCDs in the STP and, while doing so, we ensured that for no linkset more than half of the links are served within one cluster. Also the "'no inter-cluster traffic"' constraint has been counted. In practice it is, however, also convenient that the links in each linkset are attached to CCLKs of different parity in roughly equal number. How to incorporate this additional constraint into our models from Sections 3.4 and 3.5 is sketched in the following.

By (3), it is already guaranteed that each link can be connected the CCD it is assigned to via a CCLK from the same cluster. It remains to control the parity of the CCLKs. We therefore replace the variables x_{ld}, indicating that some link l is assigned to some CCD d, by pairs x^e_{ld}, x^o_{ld} of variables, which, in addition, indicate whether the link is attached to a CCLK of even or odd parity. The changes in the constraints (1)–(6) as well as in the models (7) and (8) are straightforward. Merely two of them are mentioned here.

The constraints (2) requiring each link to be assigned to some CCD now read as

$$\sum_{d \in D} \left(x^e_{ld} + x^o_{ld} \right) = 1 \quad \forall l \in L \tag{9}$$

The constraints (3), bounding the number of links per cluster, are split into two constraints sets; one each for CCLKs with even and odd parity, and $\left\lfloor \frac{n_q}{2} \right\rfloor$, $\left\lceil \frac{n_q}{2} \right\rceil$ are used as the right-hand sides in the new constraints.

New are constraints balancing the links within each linkset between CCLKs of even and odd parity:

$$a_s \leq \sum_{l \in L_s} \sum_{d \in D} x^o_{ld} \leq b_s \quad \forall s \in S \tag{10}$$

where the parameters a_s and b_s can be specified independently for each linkset s. If, for example, half of the links in each linkset shall be assigned to a CCLK of odd parity, then we let $a_s = \left\lfloor \frac{|L_s|}{2} \right\rfloor$ and $b_s = \left\lceil \frac{|L_s|}{2} \right\rceil$ for each $s \in S$. In order to demonstrate the flexibility of this definition, let $a_s = \min\{ \left\lceil 0.75 \cdot \frac{|L_s|}{2} \right\rceil, \left\lfloor \frac{|L_s|}{2} \right\rfloor \}$ and $b_s = \max\{ \left\lfloor 1.25 \cdot \frac{|L_s|}{2} \right\rfloor, \left\lceil \frac{|L_s|}{2} \right\rceil \}$. Then small linksets have to be essentially balanced between odd and even, whereas larger linksets may have an imbalance of up to 25% in this example.

4 Computational Results

Integer linear programs such as (7) and (8) can be solved by linear programming based branch-and-bound algorithms. Recent advances in the development of commercial mixed integer programming solvers (see [1] for examples of such developments) have resulted in solvers that are capable in solving quite large integer linear programs. Our models contain many well-studied structures. We have therefore chosen to apply such a solver in order to find the optimal (or good) configuration. For the computations presented here, we used the callable library of CPLEX, version 7.1 [2]. A major advantage of integer programming solvers in comparison to heuristics is that they provide a lower bound on the imbalance or the number of changes respectively, in addition to a solution.

We present computational results for three real-life instances provided by E-Plus. We refer to these instances as SMALL, MEDIUM, and LARGE. Table 1 shows some statistics for these instances: the number of links ($|L|$), the number of linksets ($|S|$), the number of clusters ($|Q|$), the number of CCDs ($|D|$), and the average number of CCLK ports per cluster ($\overline{n}_q$). In addition, Table 2 lists some statistics on the size of model (7), where we minimize the load imbalance . Similar values apply for model (8), where we minimize the number of changes. The build-in preprocessing algorithm of CPLEX hardly reduces these sizes.

instance	$\|L\|$	$\|S\|$	$\|Q\|$	$\|D\|$	$\overline{n}_q$
SMALL	173	8	3	15	64
MEDIUM	331	8	6	30	64
LARGE	602	16	11	55	64

Table 1: Statistics test instances

instance	# variables	# constraints	# non-zeros
SMALL	5146	248	35717
MEDIUM	19486	468	135909
LARGE	62742	839	435090

Table 2: Statistics integer linear programs for minimizing the load imbalance after CPLEX preprocessing

First of all, the given configurations for SMALL, MEDIUM, and LARGE are infeasible with respect to the diversification constraints (4) and/or the CCLK parity constraints (10). Thus, a few changes, namely 6, 1, and 13, are certainly necessary for all instances in order to obtain a feasible solution.

We first consider model (7), including the extensions proposed in Section 3.6. Hence, we want to minimize the load difference subject to a maximum number of changes B to the current configuration. Notice that for small values of B the resulting integer program may be infeasible.

Table 3 shows the results for several values of B. In each case, we set B to the minimum number of changes required to obtain a feasible solution, as well as to this figure plus 5, 10, 15, 20, and 25. For every value of B a lower bound (LB) and an upper bound (UB) on the load imbalance are shown. In case both values are equal, the optimal imbalance, given the number of changes, could be found. In case, no upper bound is displayed, the integer programming could not find a solution within the time and memory restrictions. we allowed CPLEX to run for at most 24 hours using at most 1.5 GB of memory on a PC operating GNU/LINUX with a 850 MHz Pentium II processor and 1 GB of main memory. To increase the readability of the results the initial load imbalance has been normalized to 1. Hence, a value of 0.17, say, means that the imbalance has been reduced by 83%.

The results in Table 3 show that for the instances SMALL and MEDIUM an optimal solution could be found, when B is set to the number of required changes. Moreover, with this number of required changes, reductions of the load imbalance from 6% to 67% could be achieved. For larger values of B further reductions in the load imbalance are obtained. Although not for all values a solution could be found for all instances, several solutions are available to support the decision of the STP planner.

We also tested model (8) in which the number of changes is minimized subject to a maximum permissable imbalance of D. Again using normalized data, the initial imbalance equals $D = 1$. From the previous computation we know that even in this case some changes are necessary due to the infeasibility of the

instance	min	min $B+0$		min $B+5$		min $B+10$		min $B+15$		min $B+20$		min $B+25$	
	B	LB	UB	LB	UB	LB	UB	LB	UB	LB	UB	LB	UB
SMALL	6	0.36	0.36	0.00	0.17	0.00	–	0.00	0.14	0.00	–	0.00	0.14
MEDIUM	1	0.94	0.94	0.18	0.46	0.00	0.32	0.00	–	0.00	–	0.00	0.23
LARGE	13	0.26	0.33	0.05	0.28	0.00	0.28	0.00	–	0.00	0.24	0.00	–

Table 3: Computational results minimizing the load imbalance

current configuration. Table 4 shows the results for 5 other values of D. Again a lower bound (LB) and an upper bound (UB) are given. The table shows that a reduction of the imbalance by 85% is possible with 10 changes only for instance SMALL, whereas it can be guaranteed that 8 changes are necessary to achieve this goal. The gap between upper and lower bound increases rapidly for larger reductions as well as for larger instances. For instance LARGE, we cannot report any upper bounds, i.e., no solution could be found by CPLEX within the available time and memory as mentioned above.

instance	$D=1.00$		$D=0.25$		$D=0.20$		$D=0.15$		$D=0.10$		$D=0.05$	
	LB	UB	LB	UB	LB	UB	LB	UB	LB	UB	LB	UB
SMALL	6	6	7	9	7	9	8	10	8	12	8	27
MEDIUM	1	1	5	16	6	29	7	36	7	–	7	–
LARGE	13	13	14	–	15	–	16	–	17	–	19	–

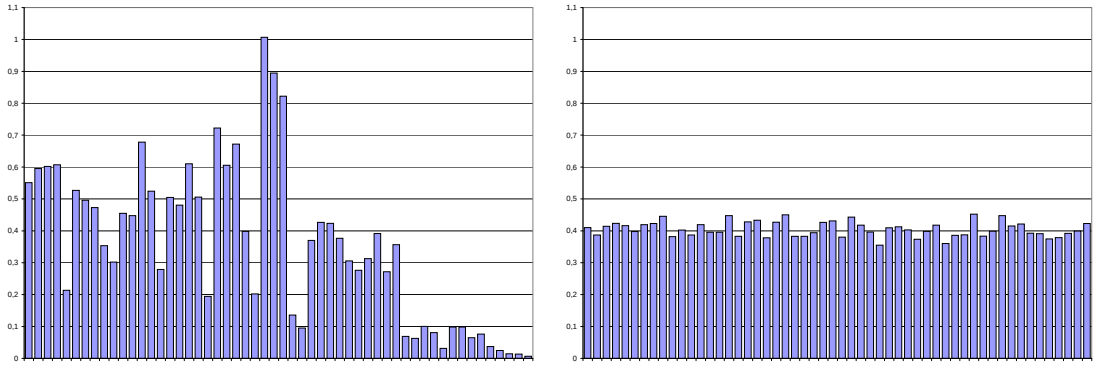
Table 4: Computational results minimizing number of changes

Combining the results for both models, some additional conclusions can be drawn. For instance MEDIUM no solution could be found with the first model, when allowing at most $B = 16$ changes. From the second model, however, we have a solution that takes 16 changes to reduce the imbalance with 75%. This value fits perfectly in between the values 0.32 and 0.23 of the first model. On the other hand, from the value 0.24 for the LARGE instance at $B = 33$ we know that a reduction of 75% can be achieved with less than 33 changes, thereby extending the results of the second model.

However dissatisfactory these results may look at first glance (from a theoretical point of view), they are of great practical help. We illustrate the practical relevance of the described approach with two examples, see Figures 4 and 5. Each one compares the imbalance before and after implementation of the changes proposed by the optimization tool for an STP from E-Plus' signaling network.

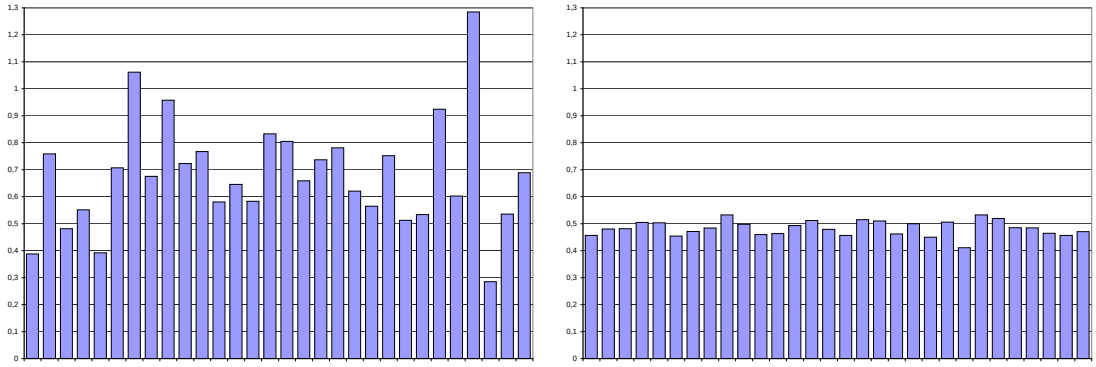
5 Conclusions

The STP load balance optimization has usually been performed manually. This approach is hitting its limits with the increasing size of STPs. We have presented essentially two integer linear programming formulations for the STP load balance optimization. In the one model, the CCD load difference is minimized subject to a limit on the number of permissible changes. In the other model, the number of changes is minimized in order to reach a pre-specified maximal STP imbalance. These models form the mathematical core of an automatic STP load balancing tool, which is nowadays in use at the German GSM network operator E-Plus.



(a) before optimization (b) after optimization

Figure 4: Reducing the CCD load imbalance by 90%



(a) before optimization (b) after optimization

Figure 5: Reducing the CCD load imbalance by 88%

From a mathematical point of view many questions still have to be clarified. The computational results show that there is still room for improvements. We see several ways to increase the performance of the algorithms. For example, it is well-known that the generation of so-called problem-specific cutting planes can improve the quality of both lower and upper bound. First experiments have shown that substantial improvements can be achieved by the application of these and other ideas.

From a practical point of view, however, the relevance of the presented approach is beyond dispute. The software for the STP optimization allows E-Plus to save about 95% of the (significant) time usually spent on optimizing an STP and, at the same time, to improve the results considerably. In a typical example, the STP optimization increased the back-up capacity of the critical, highly loaded CCDs by 41%. (No additional hardware was deployed.) The ability to limit the number of permissible link changes is essential from a practical viewpoint. It allows to routinely perform STP reoptimizations and, thus, to maintain the required CCD back-up capacities (for link failures) on the long run.

References

[1] R. E. Bixby, M. Fenelon, Z. Gu, E. Rothberg, and R. Wunderling. MIP: Theory and practice – closing the gap. Technical report, ILOG CPLEX Division, 2001.

[2] CPLEX division of ILOG. CPLEX callable library, version 7.1, 2001.

[3] A. M. C. A. Koster. Re-optimization of signaling transfer points. ZIB-report 00–18, Konrad-Zuse-Zentrum für Informationstechnik Berlin, Berlin, Germany, 2000.

[4] P. J. Kühn and S. Mayer. STP-Optimierungsproblem. Technical report, IND, Universität Stuttgart, October 1999. Forschungsbericht im Rahmen des Kooperationsprojekts von E-Plus und der Universität Stuttgart, IND. In German.