NTNU
Innovation and Creativity

Cost comparison studies of different network platforms

OpEx modeling and analysis

Øyvind Solberg Moe

Master of Science in Communication Technology

Submission date: October 2007

Supervisor: Bjarne Emil Helvik, ITEM

Co-supervisor: Carmen Mas Machuca, Technische Universität München

Norwegian University of Science and Technology
Department of Telematics

Problem Description

This thesis is a part of a series of OpEx studies aimed at comparing different platforms able to offer Ethernet services. The objective is to define and implement a generic model to evaluate the network operating costs including the provisioning cost of these services, as well as a comparison of different network platforms.

Assignment given: 01. April 2007
Supervisor: Bjarne Emil Helvik, ITEM

Abstract

The importance of Operational Expenditures (OpEx) on the overall network cost has been shown in recent studies. This master thesis is part of a series of techno economical studies focused on the evaluation of OpEx and the provision of a general tool. This thesis proposes a network model which is integrated with a recently proposed service model and service migration model, combined into an OpEx tool. This tool allows network operators comparing different network platforms with different technologies, studying the impact on the cost of new control and/or management capabilities, etc.

This tool allows the evaluation of OpEx cost of different network platforms and the identification of the most important cost factor. This thesis has focused on the most costly process by finding an optimal solution that minimizes it. Hence this master thesis forms a complete cost reduction study of an optical backbone network.

My part of the work has been the integration of the three cost models into one combined cost model and the development of the resulting tool as well as the improvement of the service model implementation. I have then developed a complete scheme for reducing the cost of failure reparations in a backbone network.

Contents

1 Introduction

1.1 Background/Motivation

Throughout history there has been an ongoing increase in standards of living, from the harsh lives in the earliest civilizations to the comforts of today's society. The society in general has always demanded that the standards of living should be rising.

In today's capitalistic society higher standards of living is accomplished when the employees receives an increase in wages. Rising wages does not lead to higher standards of living if they only lead to higher prices, but if matched by rising productivity or carved from profit margins they do lead to higher standards of living [21].

Because people who are in possession of the profit margins are not to eager on decreasing them and more importantly the fact that there is a limit to how much the profit margins can be decreased, have a wage increase traditionally been compensated by rising productivity. This is manifested in history where new technical solutions has increased productivity (e.g. irrigation that increased crops, the steam engine that provided mechanical energy for producing goods, etc).

The telecommunication industry is in this context no exception and therefore its manager's tries to deal with the society's ever present demand for higher standards of living. Network operators want to maximize the Total Benefit Ownership (TBO). TBO is the difference between profits and costs. Profits depend mainly on the money received from the services. These profits are limited due to the strong market competition in today's converging telecommunications market. Hence to maximize TBO the best approach is to minimize costs. The cost of operating a telecommunication network is known as Total Cost of network Ownership (TCO).

The cost of a business, a product, etc. can be divided into Capital Expenditures (CapEx) and Operational Expenditures (OpEx). CapEx refers to the cost of developing or providing non-consumable parts for a product or system. Its counterpart OpEx is the on-going costs for running a product, business, or system. Let us explain the difference between the two with a naive example, the investment in a printer: The cost of buying the printer is CapEx and the cost of the consumption of paper and toner/ink is OpEx.

Until recently most studies of the economics of network structures focused on the initial investment related to the planning, building and deployment of the network often called Capital Expenditures (CapEx) studies. The first techno economic studies were comparative studies of the investment that was required to build the network, i.e. considering a given network topology and traffic matrix, which were the equipment needed when using one particular technology and/or some management and control capabilities [17, 14].

Techno economic studies have normally neglected the cost of normal network operation, also known as Operational Expenditures (OpEx). But recent studies have shown that OpEx is a major and maybe the most important contributor to TCO e.g. Nokia claims that OpEx will be the 50-80 percent of TCO over 10 years and the Metro Ethernet Forum claims 60-85 percent of TCO [3].

With respect to the percentages mentioned in the previous paragraph could only a small reduction in OpEx could lead to a substantial decrease in TCO. Hence OpEx studies are a growing area of interest as the awareness of the importance of network operational cost is increasing.

1.2 OpEx

This TCO dichotomy can also be used in a telecommunication context where the Total Cost of network Ownership (TCO) for a network operator can be divided into CapEx and OpEx. OpEx can be divided up into two sub parts; the network related costs and the service related costs. CapEx can be divided up into three sub parts; the none telecom costs, the network related costs and the service related costs. An overview of the TCO's cost structure is shown in Figure 1.

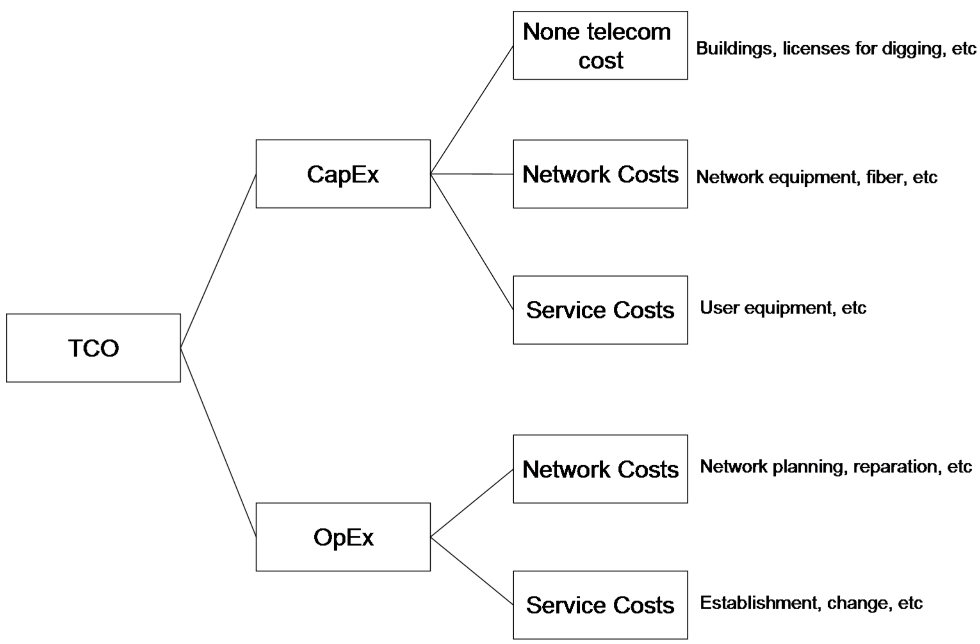

Figure 1: Cost structure of TCO

For a network operator, CapEx contributes to the fixed infrastructure belonging to the operator. This includes the purchase of land and buildings (e.g. to house everything from equipment to personnel), the network infrastructure (e.g. switches, fibers, etc.) and software (e.g. network management systems). [19]

OpEx on the other hand does not contribute to the infrastructure in itself but on how the network is operated and includes: the cost to keep the network in a failure free condition, the reparation in case of a failure, the process of provisioning services, service management, pricing, billing, ongoing network planning and marketing.

Depreciation is used to correctly calculate the present value of an investments future cash flow. CapEx are subject to depreciation and therefore the concept of present value calculation has to be used. However, the calculations done on OpEx in this master thesis are not subject to depreciation, as suggested in [16], but is proposed as a further study.

This is the definition of OpEx that is used in the series of OpEx studies that this master thesis is a part of. But it should be noted that there are no standardized way of defining OpEx, hence are there some differences of opinion in which factors should be counted as OpEx and which as CapEx. The approach used here is the one followed by most companies in the telecommunications industry.

1.3 Goal

This master thesis is part of a series of techno economical studies focused on OpEx. The importance of OpEx has been highlighted in Section 1.1 and the goal of this master thesis is to develop a generic method to evaluate the OpEx of different network platforms of any network operator.

The plan is to define, develop and make use of a generic model which enables the prediction of the network operating cost of offering services on a certain platform. With this generic model, the comparison of different platforms offering different types of services and protection schemes and the cost impact of new control functionalities can be investigated. A network operator should be able to exploit this generic model to reduce the OpEx of his network(s) and hence, reduce TCO.

This tool should integrate the pure operational network costs (such as failure reparation, network planning, etc.) with the costs of the services offered by the network. Previous work [9, 10, 13] dealt with the modeling, implementation and studies of service costs. This model should be integrated in the proposed network cost tool and its implementation should be improved to reduce computation time.

With the use of this tool we will compare different network platforms. From these comparisons the most important cost factor should be identified and as a case study, we looked into a way to minimize it.

1.4 Outline of the work

Section 2 presents the network model. This model is based on the network life cycle and for its cost calculation, the cost of the running services and the possible service implementations have been included. The proposed network cost model is based on a structure so that each network process can be as accurately defined as desired. The contribution of this thesis consist on the integration of the network, service and service migration cost model within a single tool, while improving the implementation of the service cost.

Section 3 different methods of modeling and implementing the cost structure presented in Section 2 are introduced. There is given a detailed introduction to each of the different implementation techniques that is chosen as potential candidates for implementation.

The advantages and disadvantages of each of the different methods are introduced and discussed. Thereafter are the different implementations technique compared against each other before a technique is chosen.

In Section 4 is it explained how the 3 integrated cost models introduced in Section 2 are implemented with the technique chosen as a result of the comparison of the alternative techniques in Section 3.

Section 5 gives a short description of the network scenario that is the basis of the investigations performed in this thesis. It also introduces the network scenarios which the comparative studies are performed with as well as the result of these studies.

The cost reduction proposal of the most important cost factor found in Section 5 is presented in Section 6. A complete scheme for the optimization of the cost related to the failure reparations of the network scenario introduced in Section 5 is presented before the result of the optimization is presented at the end.

Finally in Section 8 and 7 is the discussion and conclusion parts of the master thesis respectively presented.

1.5 Publications

In relations with this master thesis following articles have been published or is to be delivered for publication.

Modeling of OpEx in network and service life-cycles
Carmen Mas Machuca, Øyvind Moe and Monika Jäger
ECOC, Berlin, September 2007

Impact of the protection schemes and network platforms available on OpEx
Carmen Mas Machuca, Øyvind Moe and Monika Jäger
to be submitted at JON

2 Description of the network cost model

The following network description is formed on the basis of the German back bone network operated by T-systems.[1]

2.1 Network cost model

The network operation cost can be divided up in 4 so called core processes, which together forms the network cost model shown in Figure 2. Each of these processes contains a set of related tasks that are performed during the life of a network. In this section each of the 4 core processes are introduced and explained with their associated sub processes.

The basis of the network cost model consists of the core processes connected with each other in a chain structure as shown in Figure 2. The chain starts in "Long-term Planning", "Installation and Initial configuration" and "Network Operation" before the life of the network ends in the process known as "Network Dismantle".

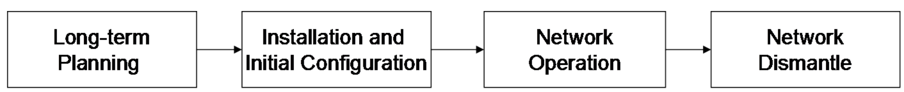

Figure 2: Basic network cost model

When one process in Figure 2 has been finished it moves on to the process at the right i.e. a queue of processes. At a later point in time it will never return to an already visited process, i.e. a process to the left of the current process.

Each of the 4 core processes has a set of underlying sub processes, which each has an associated cost. Based on this information and on the internal relationship of the sub processes the cost of the core process, which is formed by the set of sub processes, can be calculated.

Each of the core processes are presented independently in Sections 2.1.1 trough 2.1.4. The associated sub processes is also briefly explained in the following sections provided that the complexity allows it. If not, is the complete graphical description to be found in Appendix 8.

2.1.1 Long-term Planning

"Long term planning" is the process that involves the design, dimensioning, service portfolio definition, etc. of the new network. Network operators decide to take a new network platform in use when there is a new technology able to offer new services and/or reduce the cost of existing services running on other platforms.

The sequential boxes shown in Figure 3 are the sub processes of "Long-term Planning". It starts with the marketing apartment surveying the market

[1]a daughter company of Deutsche Telekom.

potential (e.g. the expected number of services needed, the bandwidth, the resilience needed, price, etc). It then moves into the planning phase where a technology which fulfills the requirements drawn up by the marketing apartment and at the same time minimizes the cost, is chosen. A plan for the configuration is then developed for the technology chosen.

There has to be a lot of key management personnel or hired experts involved in the process of "Long-term Planning". They need to have a good grip on both existing and future technologies, as well as experience from similar work. These kinds of personnel are a scarce resource and normally well paid therefore this process can be somewhat costly.

The sub processes of "Long-term Planning" are structured in a quite easy way. The method of calculating the total cost of all the sub processes that together forms the cost of the "Long-term Planning" is a simple matter where the cost of all the sub processes are added together. I.e. each of the sub processes shown in Figure 3 has an associated cost which are added together.

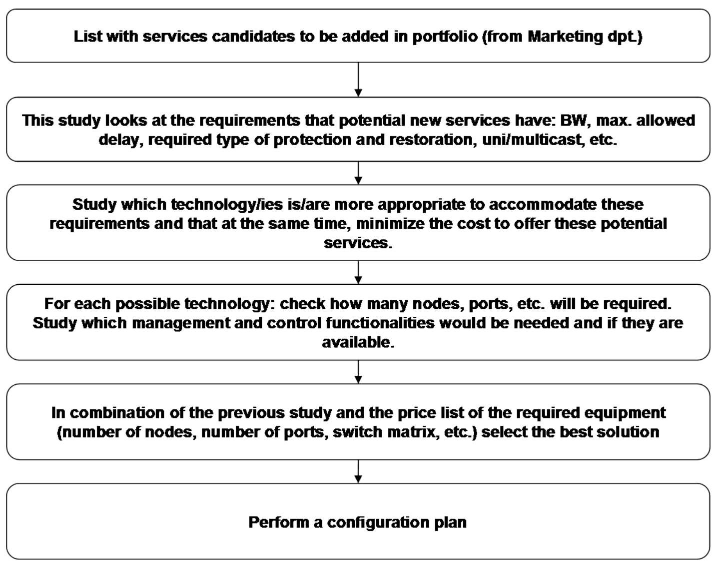

Figure 3: Sub processes of Long-term planning

2.1.2 Installation and Initial Configuration

This process stretches from placing the order of new equipment to the final testing of the system before it goes into normal operation. In-between are the deployment, configuration and testing phases, an overview of it is given in Figure 4. The process consists of 6 sub processes each of which has its own set of sub processes.

The structure of the sub processes associated with "Installation and Initial Configuration" are a bit more complex than the one of "Long-term planning" described in Section 2.1.1. Instead of just one layer of sub processes there are two layers of sub processes e.g. the sub processes that the core process comprises of have sub processes of their own. This structure of double sub layers are shown in Figure 4 where only the sub processes of the sub process "Order and receive new equipment" are shown. It should be noted that all sub processes observed on the left side in Figure 4 have sub processes themselves, except for number 5 "Perform local tests, ...".

The complete overview of the second layer sub processes is to complex to be shown here. Therefore it is to be found in Appendix 8.

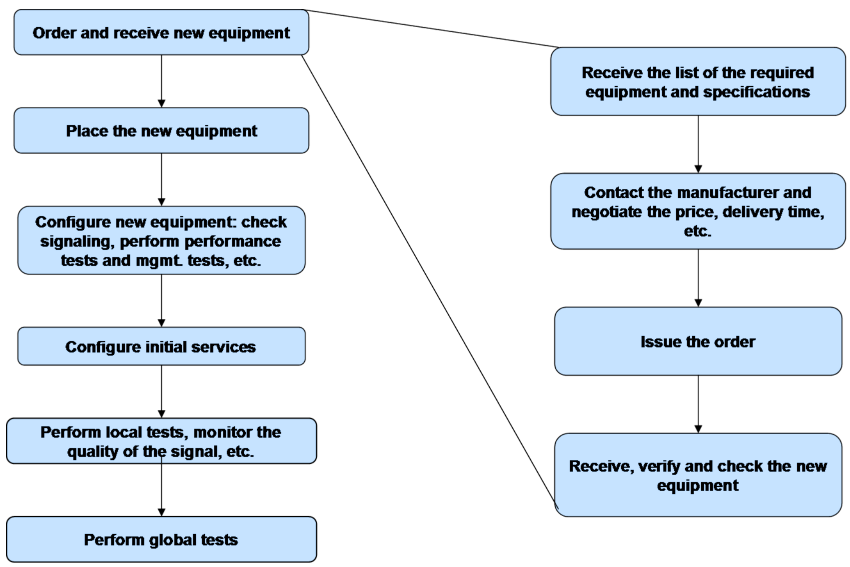

Figure 4: Sub processes of Installation and Initial Configuration

2.1.3 Network Operation

"Network Operation" is the process of normal network operation e.g. the every-day operation and maintenance performed by the network operator. It contains the following sub processes: Network maintenance, short term planning, failure reparation, software upgrade, hardware upgrade, and service migration. Their internal relationship with each other is shown in Figure 5.

Below is a short description of each of the sub processes shown in Figure 5. They also have associated sub processes which can be observed in Appendix 8.

- Network maintenance: is as the name implies monitoring of the network as it is in the state of normal operation. This is mainly the measurement of signal quality between nodes, but can also be the measurement of the individual physical connection. The motivation behind it is to detect

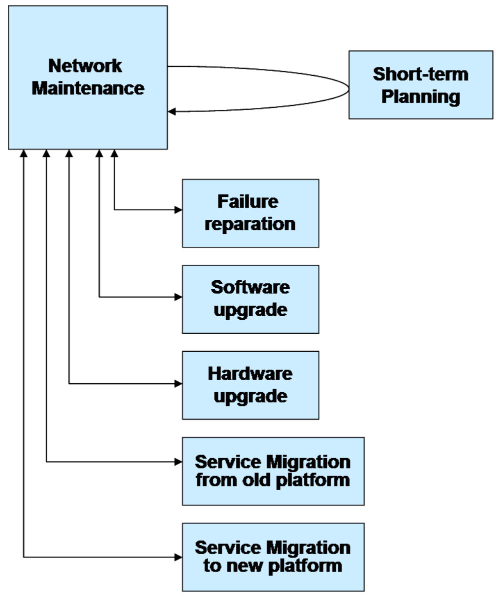

Figure 5: The sub processes of Network Operation

abnormalities in the network and take countermeasures against them (e.g. by sending a technician to the problem) before they develop into failures.

- Short-term Planning: is a sub process done on a regularly basis by the network operator itself. Network optimization is performed on the basis on existing demands, expected future demands, network status, etc. if necessary, a reconfiguration of the platform is performed on the basis of the shorth term planning result.

- Failure reparation: this is a complex structure because there are a lot of different actions that has to be taken in the process of repairing a failure. First of all the failure has to be detected and located. If it is possible the traffic may be routed around the failure under the prerequisite that the customer(s) using the affected services have this as a part of their service level agreement. The failure has to be connected before the network can return back to normal working condition. The accounting apartment has to be informed because it has to take the failure into account when billing the customer, in case of service level agreement violations. In some cases of serious failures the customer should also be informed directly about the failure.

- Software upgrade: is a sub process mainly taken care of by the company delivering the software used in the system. Depending on the agreement which the network operator has with the sofware provider he may have to pay for each upgrade or more common, it is a part of the service offered when purchasing the software. Hence the network operators ability to influence the cost associated with Software upgrade, is very little.

- Hardware upgrade: a hardware upgrade is done when the network has to be upgraded. This process includes all form of planning in relation with the upgrade, warning of customers, re-route important traffic, the installation itself, the possible removal of old hardware and finally the restoration of previous traffic.

- Service migration from old platform: this is the process of migrating services from an old platform on to the existing platform.

- Service migration to new platform: is the process of migrating the existing services on to a new platform, typically done in the process before dismantling a network. Service migration is further explained in section 2.3.

"Network Operation" has also a "hidden" set of sub processes that is not previous mentioned or shown in Figure 5. During normal "Network Operation" a set of services is also maintained and therefore contributing cost to OpEx. This set of "hidden" sub processes are further discussed in Section 2.2.

2.1.4 Network Dismantle

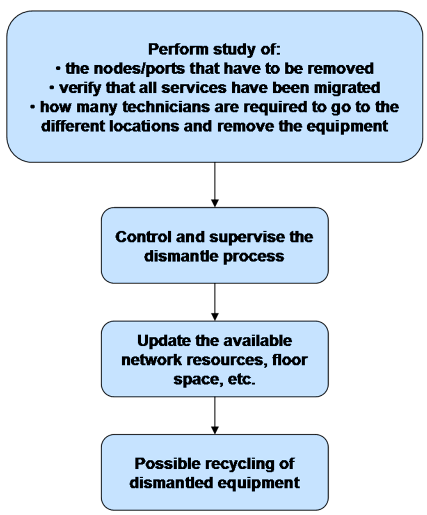

Figure 6: Network Dismantle

This is the process where the network is teared down and is the whole process from taking down the last services, to disconnecting the equipment and the possible recycling of the it. This process has only one level of sub processes and they are structured in a sequential way like in "Long-term planning" described in Section 2.1.1. How they are structured, as well as what processes they contain are shown in Figure 6.

2.2 Service cost model

2.2.1 Introduction

As mentioned in Section 2.1.3 the underlying services are also contributing to OpEx. The approach used to model services is based on the service life cycle consisting of five processes. The service life cycle is presented with its processes as square boxes shown in Figure 7. Each of the processes in Figure 7 has an associated cost. The service life cycle presented here is defined in a previous

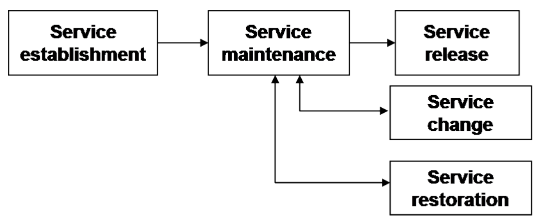

Figure 7: Service life cycle

project work [13]. There the services are modelled with the use of Markov chains. The implementation in this thesis does also use the Markov chain for modeling, but the approach of solving the Markov chain has been improved in this master thesis. Instead of simulating the each step of the Markov Chain an analytical approach called Markov reward modelling is chosen.

2.2.2 Service model

This model has been applied to Ethernet services. One service can be mapped into one or several EVCs. The mapping rate depends on the network platform, the type of connection and the protection scheme used.

In our study three platforms with different control and input capabilities have been considered:

- ”Manual” platform: First generation SDH with manual configuration capabilities which is denoted as Manual. This is the case of the first SDH networks, where every virtual connection (VC) had to be established by configuring each of the nodes of the VC. This traditional SDH platform, using a minimum of management functions, is investigated because of benchmarking reasons.
- E/SDH platform: Next Generation SDH with Ethernet over SDH (VCAT, GFP, LCAS) support and with automatic configuration capabilities which is denoted as E/SDH. However, this platform typically does not yet support multipoint EVCs. Therefore, the platform we assume for the project study is assumed to provide point-to-point EVCs only.

- E-MPLS platform: Ethernet with MPLS based control plane which is denoted as E-MPLS. There are first implementations of carrier-grade Ethernet solutions which are controlled by MPLS based L2 control functions. Practical configuration experiences with testbed implementations of the anticipated platforms accompanied the modeling. Therefore, we specified a second variant of the configuration model in an E-MPLS platform, whose configuration management allows to use some pre-defined service profiles of typical services to facilitate and speed up the establishment of new

In Table 1 the rate of mapping from service into EVC is given for all types of platforms and their types of services.

	Unprotected Service			Full Protected		
	P2P	P2MP	MP2MP	P2P	P2MP	MP2MP
"manual"	1	N	N(N-1)	2	2N	2N(N-1)
"E/SDH"	1	N	N(N-1)	2	2N	2N(N-1)
"E-MPLS"	1	1	1	2	2	2

Table 1: Rate of mapping from service into EVC

2.2.3 Integration of processes: proposed service model

The two life cycles and their respective processes are integrated into one single framework as shown in Figure 8. This integration into one single framework enables the calculation of the overall service cost.

This integration is a huge advantage because it makes the simulation model much easier to construct. Figure 8 shows how this integration can be done. Some of the processes from the Service and EVC life cycle are integrated in one new process e.g. the processes of service establishment and EVC establishment have been integrated together into "service (incl. EVCs establishment")

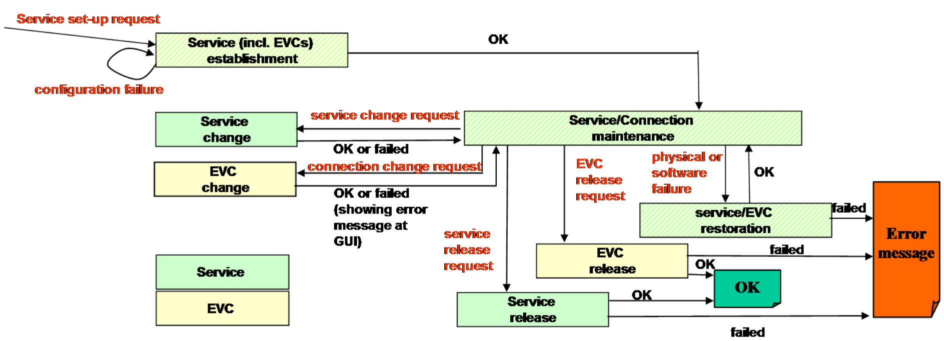

Figure 8: Proposed service model based on the integration of service and EVC life cycles.

2.2.4 The problem definition

The service cost model enables the comparison between different network and platform implementations.

Each of the processes of the new service model has an associated cost, which will differ depending on the platform, connection and type of protection that is used in the simulation.

The costs of all the processes except for maintenance are related to required actions that have to be performed. The cost of maintenance is proportional with the time spent on the process, the number of persons required for this process and their salary.

2.3 Service migration cost model

Network operators are running different platforms. "Old" platforms may not be needed since "new" platforms can offer the same services in a less costly way. Hence operators should evaluate the possibility to integrate the services to newer platforms and dismantle the older ones, which can lead to cost savings[8].The service model introduced here are the one defined and explained in detail in [8].

2.3.1 Introduction

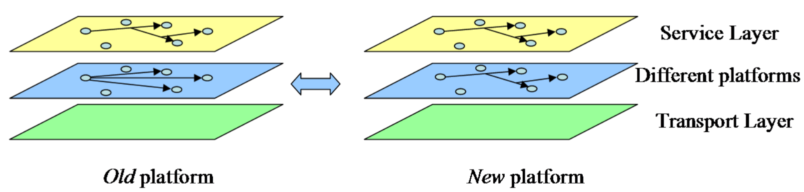

Figure 9: Service migration scenario

Service migration is the process of moving a set of services from one platform (the old platform) to another (the new platform), as shown in Figure 9. The process of service migration is a process that naturally belongs to the end phases of a network life time. In the end phase of a network life time there are still some active services in the network. One alternative would be to simply wait until all the services are teared down because their life time have expired. Instead of waiting an extensive time period on only a relative small number of services, the remaining number of services can be migrated on to a new platform.

2.3.2 Service migration model

Services can be established using different platforms using the same or different transport technologies and offering different control and management capabilities. By identifying the different processes that service migration consists of and constructing a service migration model based on that, gives network operators the ability to evaluate the cost of migrating services from one platform to another. This could be of interest when e.g. considering the possibility of dismanteling a network.

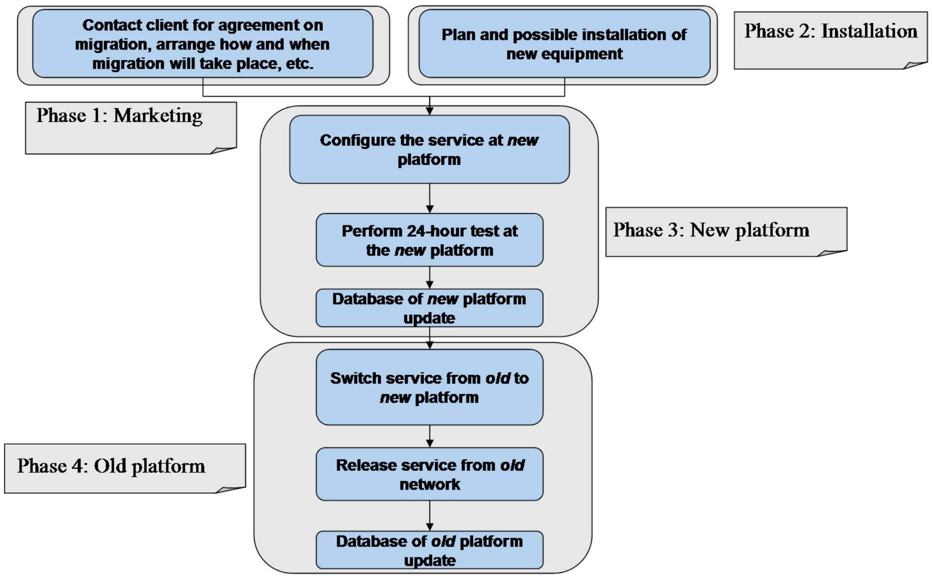

Figure 10: Service migration sub processes

Service migration consists of several sub processes they are shown in Figure 10.

1. The client of the service which is to be migrated is contacted to check if he agrees on the migration and the changes and risk it may imply. If the client agrees the specification of the migration should be drawn up.
2. In case new equipment is required by the client this should be ordered, received and installed on the client premises.
3. On the agreed date the service configured on the new platform.
4. A 24 hour test is performed to verify that the signal quality and required performance parameters.
5. Once the test has been completed successfully the service is added to the new platform database.
6. The service is then ready to be switched from the old to the new platform.
7. After the service has been successfully switched to the new platform the service is taken down on the old platform.

8. The service is then removed from the database of the old

The sub processes can be grouped into 4 groups as shown in Figure 10. Each of these phases requires different time and personnel to be performed.

- Phase 1 Marketing
- Phase 2 Installation
- Phase 3 New Platform
- Phase 4 Old Platform

2.3.3 The problem definition

With the service migration model the different migration scenarios can be compared against each other from a cost and time point of view. On one hand the slower the migration process is going the fewer services have to be migrated due that some services will be teared down while they were waiting to be migrated. The number of those incidents clearly increases when the duration of the migration is increasing.

It should also be mentioned that the faster the migration process is the less maintenance does the old platform need before it could be switched of. Hence the service migration problem can be defined as finding the optimal number of employees at each phase based on the time required in each of them and the involved personnel cost.

2.4 Integration of cost models

As mentioned previously in Section 1.3 one goal of this work is to integrate the already existing Service cost model and Service migration model introduces respectively in Section 2.2 based on previous studies in [9, 11, 10] and Section 2.3 based on previous studies [8].

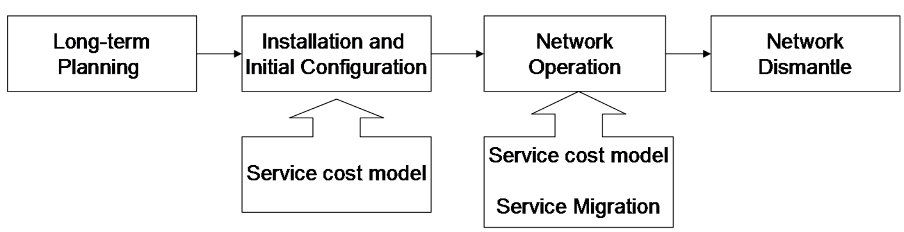

Figure 11: Integration of cost models

The conceptual model of this integration is shown in Figure 11. The top line of this figure is the 4 core processes of the network cost model. The two "arrow boxes" that comes from underneath represents the integration of the service cost model and the service migration model.

As it can be seen shown in Figure 11 service cost model merged into installation and initial integration. This is because a set of services is already established before the network gets into the phase of normal network operation.

During the normal network operation phase both cost models are incorporated into the network cost model.

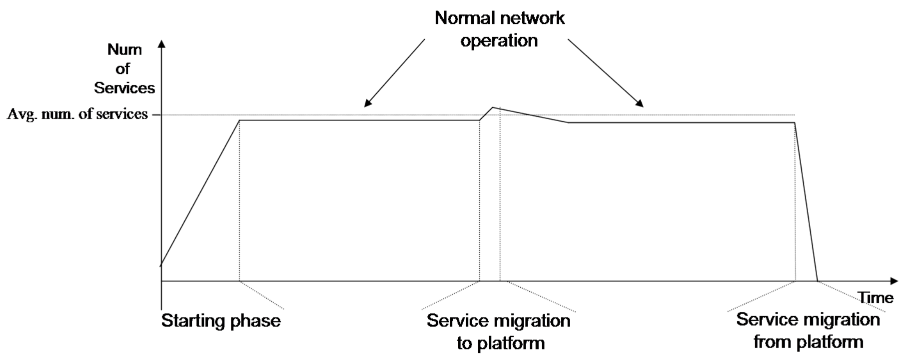

Figure 12: Number of active services in the network as a function of time

The graph in Figure 12 shows an example of how the number of active services in the network varies over time. The example consists of 4 different types of phases:

- Starting phase: the phase where there are more services established than there are services terminated due to expired life time. In other words is it in this phase a net increase in the number of active services, hence the increasing slope

- Normal network operation: the network has reached a state of equilibrium where the number of new services established in the network is the same as the number of services terminated.

- Service migration to platform: this is when services from other networks are migrated to this network. It should be noticed that the proportional increase in the number of active services shown in the graph is not realistic and in reality will a service migration only lead to a very small increase in the total number of active services.

- Service migration from platform: the last and final phase of the network life time is the before network dismantle. In this phase is the number of active services decreasing, hence the declining slope. The number of services is decreasing because of two reasons: there are no new services established in this period as the same time as services are terminated because their life time have expired and that some services also are migrated to another platform.

With hindsight to the graph shown in Figure 12 is the number of service hours needed to be analyzed dependent on the area underneath the graph i.e. the integral of the services as a function of time. This area is reliant on the life time of the network as well as the slope of the graph in the start and end phase of the network.

The service migration is present during the network operation phase. It will be called as many times as migration processes occurr in the studied network.

The only relationship between the network cost model and the two other cost models can be found in a time line context. The reason that they are presented in Figure 11 as merging is that they are active at the same point in time.

3 Comparison of different modeling and implementation techniques

Computer aided simulations consists of two main simulation approaches. A problem can be modeled and simulated in an object oriented way or in its procedural style counterpart.

In both it is possible to specify statement sequences, define variables, do branching, perform iteration and have I/O[2] or in other words everything a Turing machine can do (i.e. what a computer can do). Hence the question is not what is possible to do with each of the two modeling approaches, but where the task can be executed most easily and generic.

First the object oriented and procedural style are presented and then four possible implementation solutions are presented and discussed, before a conclusion of the implementation technique is presented at the end.

3.1 The two main approaches of computer aided simulations

3.1.1 Procedural style

The procedural programming approach was for decades the one used in all programming. This was because there did not exist other programming approaches for the time being. Students introduced to programming for the first time are almost without exception introduced to a procedural programming approach, which is forming a set of procedures structured in such a way that they solve the problem at hand.

In a procedural approach the problem that is to be simulated is decomposed into different procedures. Often is one decomposition not enough and several levels of decomposition are needed before a complexity level is reached where it is possible to program code. The procedures are either represented by general components, like a queue, or represented in programming code with data structures and code. [6]

The procedural style of modeling and simulating offers some fundamental problems. The procedures the problem at hand has been decomposed into does not correspond to real life components, they corresponds only to methods and algorithms.

This leads to the maybe biggest problem with the procedural approach; the reduced ability of extension. By this it is meant e.g. when the simulation of a city needs a new type of building this has to be programmed completely new or modeled with the limited features already available.

The procedures forms together a structure and the approach is to let this structure be traversed by entities. When this structure of procedures is appropriate, as often is the case for communications systems, this way of model-

[2]Input/Output: Input are the signals received by the computing unit and Output are the signals sent from it, e.g. respectively keyboard and monitor

ing/simulating can provide us with a suitable analogue to the real system. Or in other words the provisioning of realistic results.

3.1.2 Object oriented

Object-oriented programming is a paradigm in which a software system is decomposed into sub systems based on objects. Computation is done by objects exchanging messages among themselves. The paradigm enhances software maintainability, extensibility and reusability. [23]

Object oriented simulation (OOS) consists of a set of objects that interact with each other over time. These objects represents/refers to real world objects that interact with each other to solve complex tasks. The OO approach also makes it possible for a user to access the internal functionality of a language. The behavior of objects can be altered by the user and even totally new objects can be defined and incorporated into an already existing simulation environment.

The easiest way to explain how OOS works is to explain it with the use of an example. In a fictional town there is only one mechanic that can repair carburetors. In the model the mechanic is represented as an object and this object can perform certain tasks, in this case the reparation of carburetors. On the morning we simulate, there are four individuals in the town with carburetor problems; they are also represented as objects in the model. They could for example be arriving at the garage all four at the same time, or as in this case every 10^{th} minute were the first one arrives at time zero. One reparation takes 20 min for the mechanic to perform. This completes the introduction to the two types of objects and their description. In the following paragraph are the first steps of simulating the repair shop explained.

At time zero the customer object (CO) arrives and the mechanic object (MO) seizes the first (CO) and holds it for 20 min. When the first CO is released after 20 min 2 new COs have arrived in the mean time and the MO then seizes the first one of the two to arrive. When CO number two arrived at time 10 the MO was occupied with the first CO and it was therefore forced to wait in a queue, as also was the case for CO nr. 3. The simulation continues in this pattern for 80 minutes until all 4 carburetors are repaired.

In this example the resource and entities (respectively mechanic and customers) are all objects in the simulation. And it could clearly be seen how the objects interact with each other during a simulation.

3.2 Object Oriented Simulation

The modeling and simulations of the "Network cost model" shown in Figure 2 and described in Section 2.1 and its core and sub processes would probably not be difficult to implement in an OO approach. The method of modeling will then be to construct objects for each of the core processes and for each of the different sub processes. One object will then traverse this structure where it is seized by the different resource objects. This is quite simple for the whole structure except for the sub processes of "Network Operation" where the object could be seized by more than one resource object at the same point in time.

A solution could be that the "entity" object is seized by a "resource" object and then released directly afterwards. When the seizure of the "entity" object by the "resource" object really should be taking longer time than it is actually seized an associated counter in the "entity" object is activated. This counter count downwards until the seizure really should have been ended. While the counter is active the object could be seized by one or several other "resource" object(s).

This provides the virtual ability that the "entity" object can be seized by several objects at the same time. Hence the set of counters will keep track of the "resource" objects that are currently seizing the entity object. The only thing not possible with this solution is that the "entity" object can be seized by two or more objects at the exactly same discrete time instance.

The entity object will also keep track of all the seizures that have been done of it by the resource objects. With this information and the cost of each seizure the total cost of the network life cycle is possible to calculate.

Possible implementations:

- C++ or other OO programming languages: As it is explained above is the OO approach of modeling the "Network life cycle" easy to comprehend and perform. The implementation in itself the programming does require prior experience with C++ or another OO programming language and the developing process is expected to be time consuming, but the use of C++ or a OO programming language will enable/make it less burdensome to modify or expand the model of simulation in the future.

- DEMOS or another OO based simulation language: An OO based simulation language will provide the opportunity to build the simulation model quite fast because of predefined structures. These predefined structures will also be a set back because it will probably be necessary to do some compromises because of them. Another problem is that these types of tools/languages often are proprietary software and to acquire it can be costly. It should be noted that some of them are free for research related purposes.

3.3 Procedural approach

The procedural approach so far has been based on the modeling of the life cycles, presented in Section 2, as (Markov) chains. There could be other procedural ways of approaching the problem but they have then not been identified jet. In the first section is an introduction to Markov chains, their structure, restrictions and abilities given. Then two approaches of solving the Markov chain are presented and a comparison between the two is performed.

3.3.1 Markov Chains

The basic concepts of Markov chains were introduced by the mathematician A. A. Markov in 1907. Since then Markov chain theory has been developed by a number of leading mathematicians. The use of Markov chains for research in the field of telecommunication is only one of many fields where Markov chains have shown its relevance.

The definition of a finite Markov chain is Markov process that has a countable numbers of states. A Markov process is defined as a finite stochastic process where the past has no influence on the future. The future is only dependable on the present state of the process. A finite Markov chain is a stochastic process (Markov process) which moves through a finite number of states, and for which the probability of entering a certain state depends only on the last state occupied [7].

In this master thesis a Discrete Time Markov chain (DTMC) is used. It consist of a set states S where $S = [s_1, ..., s_n]$. Each state have a set of associated probabilities, each element in this set represents the probability of the next state being the state it represents. This set of elements forms a row vector where each element is between 0 and 1, and the sum of the vector is always 1. The states comprising the Markov chain all have such row vectors and these row vectors can be represented in a matrix M. This matrix M is called a transition matrix and consists of $i * j$ elements where $i = j =$ (number of states in the Markov chain). Place M_{ij} in the matrix is the probability of going to state j when the current state is state i.

Example of a Markov chain:

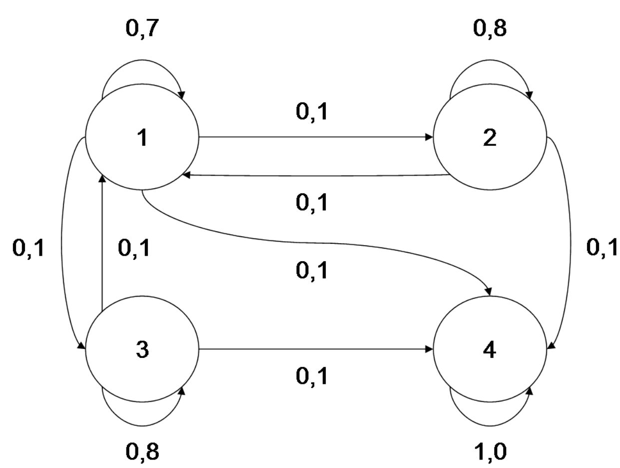

Figure 13: Example of Markov chain

Figure 13 shows a Markov chain represented as a figure. The circles shown represent the states of the chain and the arrows, with their associated probability, represent the next state probability. It should be noted that next state probabilities that has the value of 0 are not represented with an arrow because the model would otherwise be very confusing.

0,7	0,1	0,1	0,1
0,1	0,8	0,0	0,1
0,1	0,0	0,8	0,1
0,0	0,0	0,0	1,0

Table 2: Example of transition matrix

The matrix in Table 2 shows the transition matrix of the Markov chain shown in Figure 13. As it can be seen is each row in the transition matrix equal to the arrows leading out from one state, except for the elements with the value of 0 that is not represented in the figure. The states 1,2 and 3 is a transient set of states, by transient it is meant a set of states in which every state can be reached from every other state, and which can be left. State 4 has a next state probability of 1 that the next state is state 4. Hence state 4 is a so called absorbing state, which is defined as a state which once entered can never be left.

3.3.2 Markov Reward Model

Markov Reward Model (MRM) is an analytical method of solving Markov chains. A Markov reward model is made up of two main parts a behavior model and a reward structure. The behavior model is commonly known as a transition matrix. This is an $x * x$ matrix where x is the number of states/processes in the Markov chain. Each place in the matrix represents the probability to go from one state to another, this is further explained in Section 3.3.1.

The reward structure in the MRM method is a row vector containing the associated reward for each of the states in the Markov chain. In our case we want to calculate a cost not a reward, therefore the rewards are simply set to negative values.

The transition matrix is denoted as M, it should be noted that the matrix M used in the MRM method is the transposed version of the transition matrix introduced in Section 3.3.1. The cost structure is denoted by r and is a vector containing the cost of visiting each state. The number of time instances the DTMC is traversed is denoted by n. $x(0)$ is a vector with the size of one row in M, containing only zeros except for a 1 in the position where the chain starts. $x(0) = [10000....]^T$ the starting state of this example is State 1 of the chain. T means that the row vector is transposed.

The expected cost of the n^{th} iteration of the Markov chain is given by Equation 1, where $i = n$

$$cost\,of\,one\,iteration = r * M^i * x(0) \tag{1}$$

Equation 1 gives the expected cost of the n^{th} iteration of the DTMC. By summing Equation 1 from $i = 1$ to $i = n$ the expected cost of generated by all the intermediate steps from 1-n is found. This is shown in Equation 2.

$$cost\,of\,n\,iterations = \sum_{i=1}^{n} r * M^i * x(0) \tag{2}$$

According to rules that apply to mathematical sums as the one presented in Equation 2, the factors in the sum that is not containing the summation variable, in this case i, can be moved outside of the summation. The result of this is shown in Equation 3.

$$cost = \sum_{i=1}^{n} r * M^i * x(0) = r * \left(\sum_{i=1}^{n} M^i \right) * x(0) \tag{3}$$

On the basis of the time period, the behavior and reward structure the expected cost can be calculated. This is a very simple matrix operation and is not very resource demanding. It can also be very simply implemented in every type of programming and simulation language.

How the MRM method work is best explained by describing the three factors Equation 3 comprises of individually. M^i is the transition matrix M to the power of i time instances. $\sum_{i=1}^{n} M^i$ is a matrix where each element of the column gives the expected number of visits to a specific state in the Markov chain after n time instances.

x(0) is the transposed vector representing the starting state of the Markov chain. The multiplication of $x(0) * \sum_{i=1}^{n} M^i$ simply picks out the column that contains the expected number of visits to each state for the respective starting state.

The factor r is a vector containing the cost of visiting each state in the Markov chain. When r is multiplied $x(0) * \sum_{i=1}^{n} M^i$ the result is an element by element operation where the result is the total cost of the chain traversed n time instances.

One important limitation of the MRM is the inconsistency that occurs when the Markov chain contains absorbing states. This is best illustrated with a small example.

0,7	0,1	0,1	0,0
0,1	0,8	0,0	0,0
0,1	0,0	0,8	0,0
0,1	0,1	0,1	1,0

Table 3: Example of M Matrix

In Table 3 is the M matrix representation of the Markov chain shown in Figure 13. It should be noticed that the matrix representation shown in Table 3 is the transposed version of the matrix shown in Table 2 in Section 3.3.1. The

MRM method requires that each column is equal to 1 and not each row as in the normal way of transition matrix representation.

0.2487	0.1709	0.1709	0
0.1709	0.3736	0.0460	0
0.1709	0.0460	0.3736	0
0.4095	0.4095	0.4095	1.0000

Table 4: Matrix M in Table 2 to the power of 5 (M^5)

Table 4 shows the M^5 matrix. When the number of instances in Equation 3 is $n = 5$ is this the matrix that is multiplied with the two vectors, r and $x(0)$ in Equation 3. Let us disregard the cost vector for the time being and concentrate on the $M^5 * x(0)$ part of the equation. The starting state of the Markov chain was state one (i.e. $x(0) = [1000]^T$) hence the first column in the matrix shown in Table 4 represents the probabilities of the Markov chain being in the respective states after 5 time instances when the starting state is State 1. The three other columns represent the similar for the other states as starting states.

The probability of ending up in State 4 when the starting state was State 4 is 1, due to its absorbing nature. Hence the values of the fourth column are zero except for the 1 representing the probability of being in State 4.

0.0000	0.0000	0.0000	0
0.0000	0.0000	0.0000	0
0.0000	0.0000	0.0000	0
1.0000	1.0000	1.0000	1.0000

Table 5: Matrix M in Table 2 to the power of 100 (M^{100})

The M^{100} is shown in Table 5 and it can be observed that after 100 time steps the MRM method gives the result that the Markov chain is with a 100% probability is State 4, independently of the starting state. It should be noted that the 1,0000s in Table 5 only are rounded values. They can never become the value 1,0000 even tough they tend towards the value of 1, $lim_{t\to\infty} = 1,00000$. It should be noticed that the opposite is the case for the $0,0000$ values shown in Table 5.

The MRM simulation is a so called analytical approach of solving the Markov chain. The input parameters forms a transition matrix that represents the Markov chain, this together with the runtime and the starting state of the chain is all that is needed for MRM analysis of a Markov chain. The expected cost can then be calculated by combining these three factors in accordance with Equation 2.

The result provided by Equation 2 is an expected cost, hence the results of different calculations will not differ from each other. Therefore there is no need for the use of CI to give the results obtained statistical significance. Hence, only

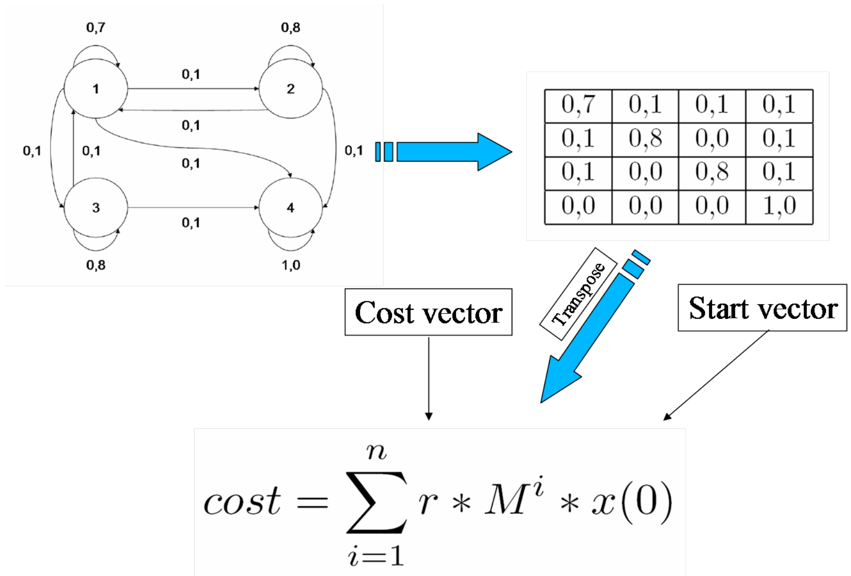

Figure 14: MRM approach

one set of analysis has to be performed to obtain statistical significant results. In Figure 14 the analytical MRM approach is shown.

3.3.3 Markov chain simulation in Matlab

The simulation of the Markov chain is a non analytical approach. Non analytical means that the Markov chain is simulated step by step. A simulation model of the Markov chain is built and for each step the in the chain a stochastical process gives the next state on the basis of the current states transition probabilities. This stochastical process can be compared with the simulation of throwing a dice that has different probabilities for landing on the different sides. Hence the result of two simulation runs with exactly the same input parameter will differ from each other.

One simulation can in some extreme cases be very different from the average simulation, hence the validity of the simulation results needs to somehow be confirmed. Therefore a simulation approach will need to incorporate the use of confidence intervals (CI).

Confidence interval: confidence estimation is a method in mathematical statistics for the construction of a set of approximate values of the unknown parameters of probability distributions. A confidence interval (CI) for a population parameter is an interval between two numbers with an associated probability p which is generated from a random sample of an underlying population, such that if the sampling was repeated n times and the confidence interval recalculated from each sample according to the same method, for a proportion p of the n samples the confidence intervals would contain the population parameter in question. [20]

The use of CI requires at least two samples to create a valid CI, but normally will more than two be needed to construct a satisfying CI. Which means that by simulating the Markov chain we do not only need one simulation, but probably several. To run several different simulations to obtain one mean result with associated CI will drastically increase the time needed for one simulation.

3.3.4 Markov chain simulation vs. MRM

The MRM method and the Markov chain simulation do both provide analogue and statistical significant result. Both methods have been proven to provide results that are realistic in nature. Therefore the only way to compare the two implementation methods against each other is to compare the runtime and the implementation simplicity.

With regards to the simplicity of implementation, is this in both cases a fairly easy task. The simulation approach is lighter to adapt to special properties that the Markov chain need to have. E.g. the ability to pause the chain in one state for an amount of time steps when the chain reaches a certain state, i.e. the time, in the model, moves on while the chain is forced to stay in this state. In the simulation approach we have the ability to force the chain to remain on one state this can be done e.g. with a simple "if sentence". The task of giving the MRM the same ability is a bit more intricate. This is while it does not simulate the states it only calculates the expected probability that the chain is in a state at time t.

To give the implementation using MRM the same ability the model has to be expanded with several new states, these states form a loop that takes the amount of time to traverse that the chain is supposed to pause. The expected numbers of visits to these new states are then simply not counted and the same result as with the simulation approach is obtained.

The conclusion is that the simulation approach is to some degree easier to implement than the MRM approach, but that the differences are only minor. Hence the simplicity of implementation is not a salient factor in deciding the method of implementation

The huge difference between the two methods of implementation is the run time of the implementation. The simple Matrix operations of the MRM method together with some clever solutions have been shown to be approximately 20000% faster than the simulation approach when compared under exact similar circumstances. The huge difference in run time is due to several factors;

1. the simple matrix operations of the MRM method is a less computing intensive calculation than the simulation.

2. with the MRM approach there is no need for several simulation runs to build up a confidence interval which gives the result statistical significance. Due to that every analysis with the same input values will give exactly the same output.

3. the MRM analysis of several Markov chains which all have the same properties, except for the lifetime, can be done in one analysis. This is possible while two chains with the same properties will have the same expected outcome after the same elapsed time. The solution is then to only perform analysis of one chain. The expected cost of the chain is then stored each time it reaches the lifetime of one of the other chains with a lower lifetime.

3.4 Conclusions

There are several deciding factors in choosing the implementation method of the Network life cycle and its underlying Service life cycle.

1. The ability of the implementation method to solve the problem at hand: This is the most salient factor in choosing an implementation method. This because the ability to solve the problem is a prerequisite that the implementation method has to fulfill. All implementation methods considered here could almost certainly solve the problem at hand, hence Factor 1 is not further investigated and discussed here.

2. The easy expansion of the implementation to make it fit with changes at a later point in time. For the second factor the discrepancy between the different modeling approaches are a bit more distinct than for the first factor. The implementation in an object oriented programming language is at a later point in time more adaptable to changes. E.g. when the network model needs to be changed or expanded a model implemented with an OO approach will be easier to change in such a way that it fits with the new changes. The procedural approaches will on the other hand be more complicated and difficult to adapt to changes in the future.

   Factor 2 speaks clearly for that the OO approach should be chosen as the method of implementation

3. The computing/run time of the implementation: Run time estimations and comparisons shows that the MRM is by far the least time demanding implementation method. A comparison of a Markov chain simulated in Matlab and solved in Matlab with the use of MRM shows that the MRM approach solves the problem with exactly the same degree of satisfaction but with a run time of approximately 20000% less than the simulation approach of the Markov chain.

   According to the fact that the OO approaches will need to make use of CI to give the results statistical significance and that each simulation has to be simulated independently, as the case also is for the simulation approach of Markov chains the simulation time of an OO simulation should be within the same time magnitude as a simulation of the Markov chain.

   A telecommunication system is a big and complex system to simulate. Preliminary results have shown that the simulation with the use of Markov chains simulation would take hours. Under prerequisite that this is true does 20000% faster mean that a simulation which takes 5 hours with a simulation approach will only take 1 second to analyze with the use of MRM.

For factor 1 there are none or inconsiderable discrepancies between the different implementation methods. Hence none of them are deciding factors in choosing the implementation method.

Factor 3 (run time) has shown to have huge differences between the different implementation methods namely that the MRM method has been showed to be approximately 20000% faster than the other methods. Hence the calculations of the expected cost are fare less intensive with regards to the use of computer power for the MRM method. Even tough factor 2 did speak for an OO approach is the advantages of a runtime that is several magnitudes faster more essential to the usability than easy modifications. Therefore the analytical MRM approach is chosen as the implementation method of choice in this master thesis.

Table 6 gives a systematic overview of the advantages, disadvantages and knowledge for each of the different implementation methods/approaches previous mentioned.

Method of implementation	Advantages	Disadvantages
C++ or other OO languages	-Real life like simulations with real life like objects -Easy modifications and expansion of model -Very likely possible to implement the model -Enables the possibility to build a good looking GUI	-No previous work done before therefore unknown territory -May be time consuming to build -It is hard to combine with the work already done on the service cost model and the service migration model, hence even more time consuming
Demos or other OO based simulation tools	-A lot of predefined structures that may lower the building time -Fast and proven tool	-Is also hard to combine with the work already done (service and migration cost) -Most of the software available are proprietary
Markov chain simulated in Matlab	-An easy to construct model within strict rules of the Markov chain -Has been proven previously that it works -The work already done is based on Matlab	Not that easy for outsiders to understand the inner structures of the calculations -Not that easy to modify or expand the model afterwards -The simulations are time consuming
MRM solved in Matlab	-It can be easily implemented and combined with the previous work on service and service migration -Is a result of simple a simple matrix operation -Is not time consuming to analyze	-Not that easy to modify or expand the model afterwards

Table 6: Overview over the different implementation methods

4 Model implementation

According to the description in Section 2.4 the network cost model consists of three main parts: the Network cost model itself, the service cost model and the service migration cost model. Matlab is used for all three, however is the method of implementation very different for each of the three parts.

4.1 Network cost model

The network cost model is implemented as a simply sequential addition of the cost of each sub process that the cost model comprises of. In Figure 15 are only the sub processes of "Long term planning" shown, but the other processes have the same structure of sub processes except for "Network Operation". The costs of these sub processes are only added once to the total cost of the network and are not influenced by the results from either service cost or service migration.

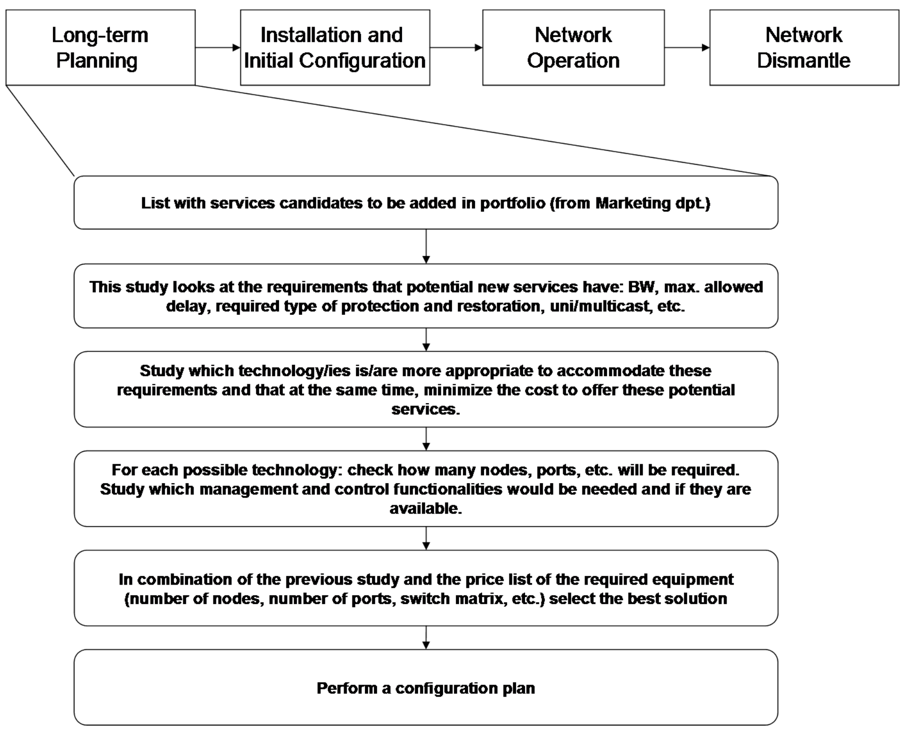

Figure 15: Basic network cost model

For the sub processes of "Network Operation" are the case different, here are the cost contribution of the sub processes influenced by other factors. The sub processes of "Network Operation" are shown in Figure 5. Three of the factors that are shown in Figure 5, short-term planning, software upgrade and hardware upgrade, is influenced by the life time of the network being simulated.

Each of the three processes has an associated rate of visit meaning that the cost of the processes can be added several times to the total network cost. The longer the time period simulated the higher is the cost generated from the three sub processes.

The cost of service migration is influenced by the number of services associated with the network. More associated services means more services to migrate to a network and more services to migrate from a network. Since each migration has a cost will the total migration cost rise when the number of services is increasing.

The longer a network is operating the more network failures will occur in it. The number of network nodes and the total length of the network links together with their respective failure rates will influence the number of failures in a network. The structure and extent of the network will together with the time decide how many failures occurring during the lifetime of it. In this thesis is the network simulated the "Germany 50" a 50 node network, it is further explained in Section 5.1.

With the above mentioned network parameters the network failures per time unit for a network scenario can be calculated. There are two main types of failures: link and node failures. Both can divided up in several classes of failures. A link failure can be a link cut, as the name implies the cable is broken or link degradation that is the when the signal quality on the link is poor but there are still contact.

There are three types of node failures: node failure, interface failure and configuration failure. A node failure is when the node itself is failing meaning that the switch is not able to perform its task for some reason. Interface failure is when one interface related to a node is failing. The last type of failure is different from the two other in the way that it does not need to be repaired on the spot but can be fixed from the NOC.

With the cost of each of the sub processes of the network cost model together with other factors (e.g. network lifetime, num of services, etc.) given as input through the Graphical User Interface (GUI) shown in Figure 16 the total network model cost can be calculated with the help of simple addition and the information given previous in this Section.

4.2 Service cost model

The service cost model integrated with the network cost model is a modified version of the one presented in [9, 11]. The cost model in itself is kept in its original Markov chain implementation the only difference is that the Markov chain is analyzed with MRM instead of purely simulated as was the original approach.

The major difference is that the MRM method is an analytical approach something that for example implies that there are no need for the use of confidence intervals. This together with the possibility to combine several analyses in one analysis, described detailed in Section 3.3.4, are the main contributors

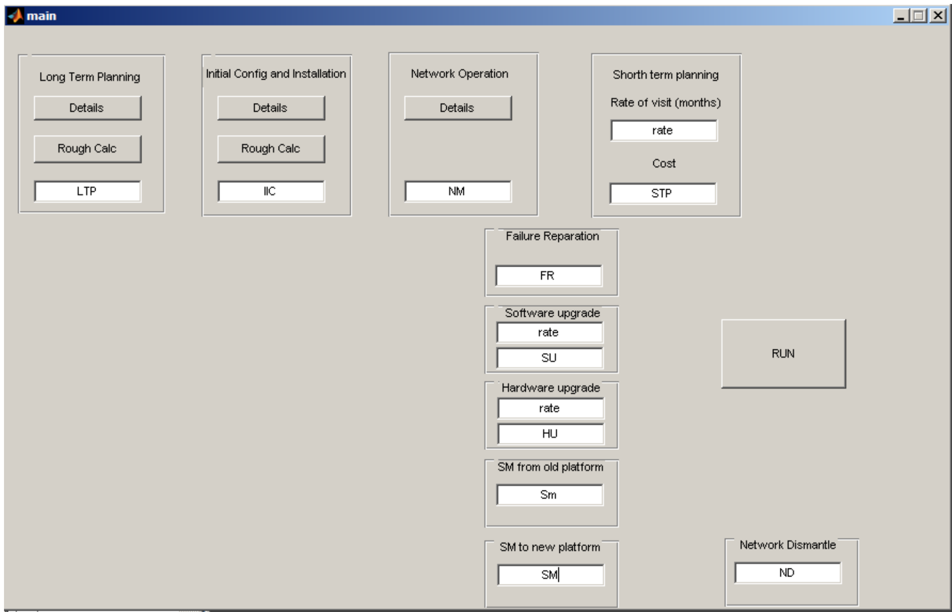

Figure 16: Graphical User Interface

to the considerable reduction in runtime achieved by changing the calculation approach.

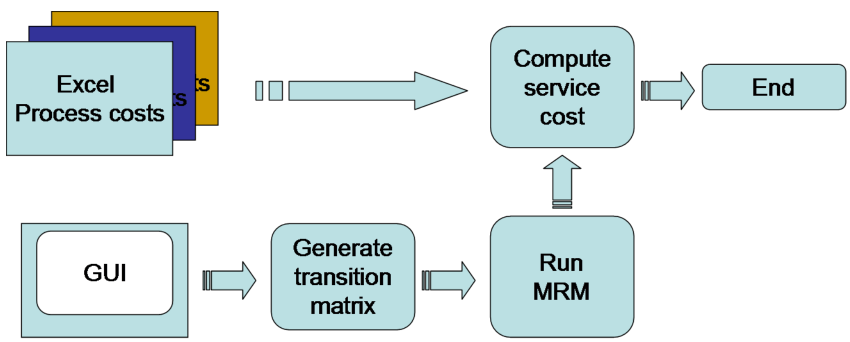

Figure 17: Approach service modeling

There are two input sources in the approach shown in Figure 17: the cost of each of the different processes in the service cost model is given as input via Excel spreadsheets. The different network parameters (probabilities, run time, type of services, etc) are given in via another part of the GUI not shown in Figure 16, but that opens when the details button in ”Network Operation” is pressed, it is shown in Figure 45. These input parameters generates a transition matrix which transposed is the M matrix used in the MRM calculations.

The MRM calculations are then performed according to Equation 4 where the expected number of visits to each process in the service cost model is calculated. In the next and final part of the approach shown is the multiplication of the expected number of visits with the cost of visiting each process done in the "Compute service cost" phase of the model.

$$expected\,number\,of\,visits = \left(\sum_{i=1}^{n} M^i\right) * x(0) \qquad (4)$$

The original approach was like the one shown in Figure 17 except for that it included the use of confidence interval, which could be seen as a box in between the "Compute service cost" and "End". If the results did not comply with the current confidence interval one more Markov chain simulation was performed, this was repeated until the results did comply with the confidence interval.

4.3 Service migration model

The service migration influences the total network operating cost in two ways:

1. The cost of the migration itself. The cost generated by the migration process itself is divided between the platform the services are migrated from and the platform the services are migrated to. Hence the service migration cost model implemented in our model can be considered to have two different processes: Process 1 where services are migrated from an existing platform and the process 2 where services are migrated to a new platform.

   The cost directly related to the service migration will be different depending on which of the two phases it is generated from as shown in Figure 10. Here the service migration process can be divided into 4 groups of sub processes. Group 1 and 4 are defined to belong to process 2 of the service migration costs while group 2 and 3 are defined to belong to the process 1.

2. Service migration involves two networks Net_1 and Net_2 and the services is migrated from the first to the second network. Hence, when our studied network is Net_2, some processes cost are included (such as the establishment of services in Net_2, updating its database, etc.). On the other hand, when our studied network is Net_1(before dismantling it), the cost associated are the cost of the processes such as the tear down of the services, update of database, etc. Service migration is also explained in section 2.4.

   The network cost is also influenced by the service migration because it affects the number of services the service cost analysis has to take into account. This can be explained with a simple comparison: at the beginning of the service migration to a new platform there are 10000 active services on the old platform. For simplicity none of the 10000 services are teared down because their life time has expired. In the first case the 10000 services all migrated over a time period of one year with an equal number of services migrated each day. Equation 5 expresses the number of active services in day x of this service migration process. Equation 6 also expresses the number of services still active, but here the service migration lasts over a period of 2 years.

$$f1(x) = 10000 - (10000/365)x \tag{5}$$

$$f2(x) = 10000 - (10000/730)x \tag{6}$$

If Equation 5 and 6 is integrated over the time period 365 and 730 respectively the results is the total number of service hours.[3]

$$\int_0^{365} f1(x) = \left[10000x - (10000/730)x^2\right]_0^{365} = 1825000 \tag{7}$$

$$\int_0^{730} f2(x) = \left[10000x - (10000/1460)x^2\right]_0^{730} = 2737500 \tag{8}$$

From the result of the integration shown in Equation 7 and 8 it can be observed that the result of the latter is bigger than the first. Hence the number of total service hours is influenced by the time period the service migration stretches itself over.

4.4 The integrated cost model

The conceptual integrated cost model already introduced in Section 2.4 has been implemented as shown in Figure 18.

The implemented cost model has two sources of input:

- The graphical user interface (GUI) which provides the analysis with input directly from the user. He is in that way able to easily change the input parameters to test different scenarios. The input provided by the GUI is the network cost, the network input parameters as well as the service input parameters. Although they could also be stored in other means of storing data e.g. database.
- Spreadsheets which contains the process costs of the service cost model.

On the basis of the input provided by the user in the GUI a transition matrix describing the service cost model is formed. This matrix is then analyzed with the help of Markov reward modeling. The result of the analysis is the expected number of visits to each state of the Markov chain. This result is combined with the cost of visiting each of those states, hence the expected cost of the maintaining the services are calculated.

[3] One service hour, is one active service for one hour.

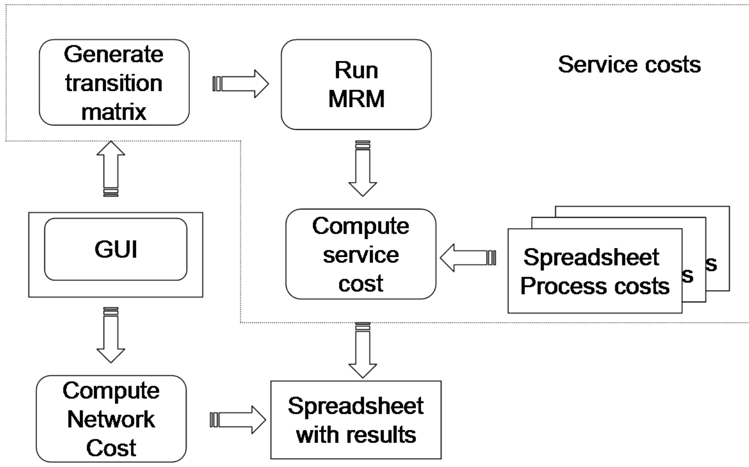

Figure 18: Implementation of the integrated cost model

The network related costs are the result of a simple calculation process where the network structure and the cost of each of the network processes are input parameters. Both set of input parameters are provided by the user via the GUI.

The results from both the network costs analysis and the service costs analysis is then combined into a common source of output. They are in this case stored in an Excel spreadsheet but the data could of course be stored in any other means of storing data e.g. a database system or simply given as output directly on the computer screen.

5 Network Comparative study

This section presents a cost comparative study of two different platforms that could be used in the German backbone network. The comparison identifies the less costly platform and the most costly network processes.

5.1 The German 50 node optical backbone network

The network scenario used in this master thesis is based on the 50 node optical backbone network, "Germany 50" [22]. Figure 19 shows the placement of the 50 nodes and their links shown.

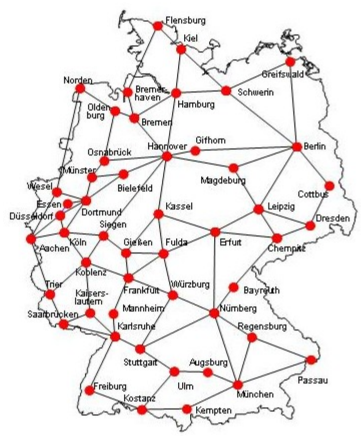

Figure 19: Map of the German 50 node backbone network, the points of presence and their links

The German state is structured as a federation and decentralization is an important political goal, which is a structure chosen by the allies after the Second World War to prevent a too strong and centralized Germany in the future. The backbone network built by Germany's old telecommunication monopolist, the now privatized Deutsche Telekom, is also structured in a decentralized way. If there are any link between the network structure and the organization of the German state is hard to say, but there are of course other compelling reasons to choose a decentralized network structure e.g. the resilience it provides.

The German network structure consist of 2 disjoint[4] backbone networks (BBN1 and BBN2) that connects 14 stand alone networks with each other. A conceptual view of this network structure is shown in Figure 20.

Each of the 14 stand alone networks has their own Network Operation Center (NOC) that means that each network is managed independently of the other networks. This decentralized solution has its advantages and disadvantages. The choice of a decentralized structure of NOCs gives the network as a unity

[4] two networks are disjoint if they do not share any link or any kind of network equipment

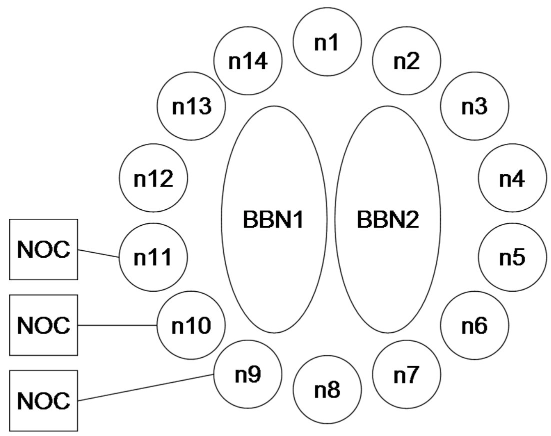

Figure 20: Conceptual model for the German network structure

a higher resiliency because if one NOC fails and is unable to perform its tasks this will only affect the connections originating and terminating in the affected network, all other connections will remain unaffected by the failure. On the other hand are there for obvious reasons higher cost of operating 14 smaller NOCs instead of one big centralized NOC which is responsible for the whole network.

5.2 The network scenarios

Two network scenarios, referred to as the "old" and the "new" platforms, are investigated in this comparative study. They are both applied onto the reference network structure introduced in Section 5.1. The cost generated by the two network scenarios over a ten year period is compared with each other. The goal of this comparison is not only to find the least costly scenarios, but to identify the most costly processes within each scenario.

In this study is an average of 500.000 active services in the network assumed referring to the average value in Figure 12.

For this purpose, the total network cost is broken down into process costs. The results of the "old" and "new" network scenario costs are expected to be correlated with each other i.e. the same processes are the most costly ones in both network scenarios. When the same processes are identified as the most costly ones in both network scenarios it can be concluded that they should be further investigated to reduce their cost as much as possible.

In Figures 21 and 22 the "old" and "new" network scenario are respectively presented. As it can be observed it varies between the two scenarios how large the cost of each process is as well as the rate the processes are visited.

It should be noticed that the costs in Figures 21 and 22 corresponds to one execution of the cost. For the platform in Figure 21 will e.g. the Failure reparation (FR) cost 30000 a year and short term planning (STP) cost $3*1920 =$

5760 a year. It should be noticed here that the numbers presented here are "fake" due to confidentiality issues.

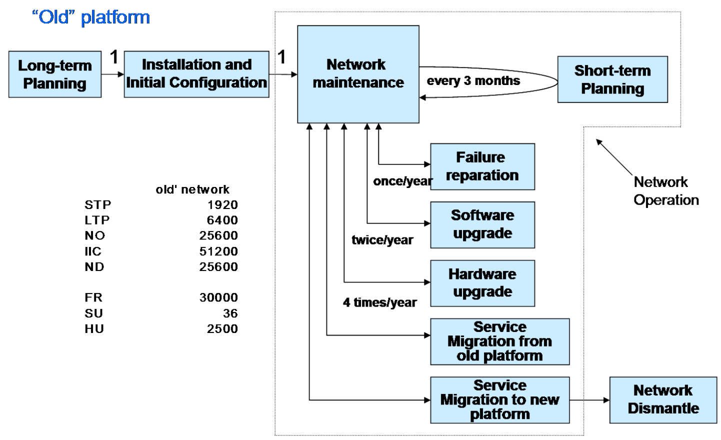

Figure 21: "Old" Platform Scenario

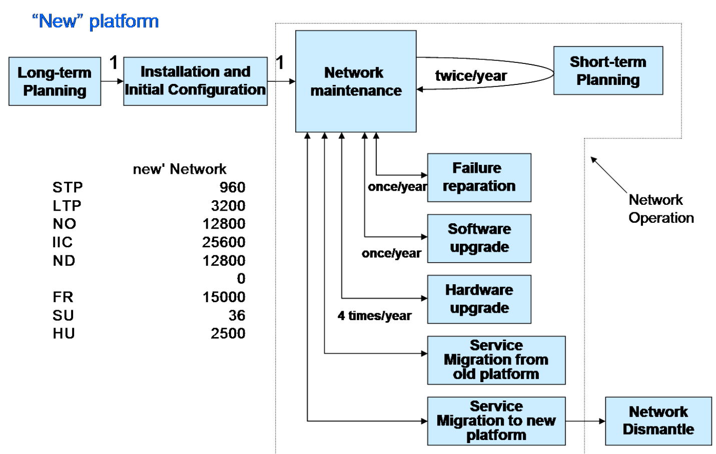

Figure 22: "New" Platform Scenario

5.3 Results of comparative studies

The goal of the comparative studies is to identify the salient cost factors of network operation. In Figure 23 is the cost of the four core processes, long term planning, installation and initial configuration, network operation and network dismantle presented. Each process has a column for both the "old" and the "new" network scenario and it should be noted that the y axis representing the cost has a logarithmical scale.

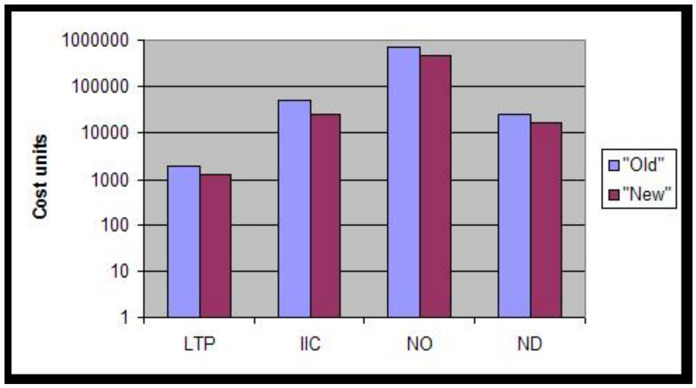

Figure 23: Cost breakdown of the network costs

According to the cost figures shown in Figures 21 and 22 should Installation and Initial Configuration (IIC) be the most costly of the four core processes. But this core processes is only done once during a network life and hence, when computing the total life time cost, IIC is not the most costly, but NO.

As it can be observed in Figure 23 the cost of network operation is the largest of the four core processes. This is valid for both the "old" and "new" network scenario, hence the internal cost distribution of network operation is given a closer look with the goal to find the most costly process. The breakdown of the "Network operation" cost into its sub processes is shown in Figure 24.

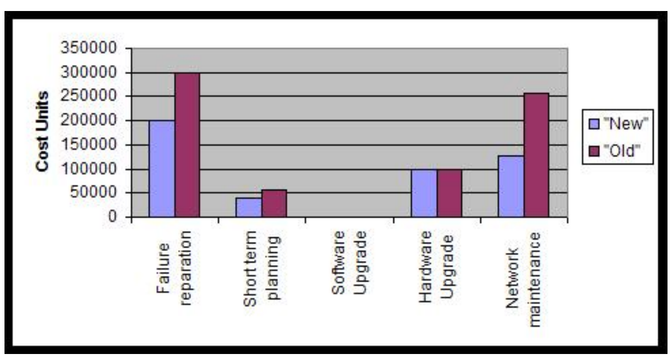

Figure 24: Cost breakdown of the core process "Network operation"

In Figure 24 it can be observed that the cost related to failure reparation is larger than the cost of the three other core processes (LTP, IIC and ND). Hence the cost of the other core processes is not further investigated. Failure reparation is identified as the most important cost factor and is the subject of further cost reduction investigations presented in Section 6.

6 Cost reduction proposal

It has been shown in Section 5.3 that the cost of repairing failures occurring in the nodes and links is the most important cost. This section proposes a solution to reduce it.

6.1 Introduction

A telecommunication network consists of a set of nodes that are connected with each other via links. Both nodes and links in a network will be subject to failures.

When a failure occurs, two mechanisms are activated:

1. Service restoration: It is a service process cost. When a failure occurs the network operator can take measures so that the affected traffic is restored as soon as possible, so that the SLAs are fulfilled, the client are satisfied and the operators don't have to pay any penalty. A Service Level Agreement (SLA) a network operator has with his clients normally contains values for the maximum service restoration time and the average reparation time. If the network operator is not able to keep his outages within these values he normally has to pay the client damages for the violations. The size of the damages depends on to which extent the SLA is breached i.e. the longer the time period the SLA is violated the more damages has to be paid to the client.

2. Failure reparation: It is a network process cost. This process includes the failure location, the checking for personnel and equipment availability, the traveling time, reparation of the failure and the testing of the equipment quality. The time constrains for failure reparation are not as hard as for the service restoration, but they are important when service restoration mechanisms fails.

Our problem is to minimize the failure reparation cost. Based on a network topology and the availability of their components, find the optimal number of locations and the optimal number of employees at each location. This optimum is a compromise between the cost of having these employees and the locations; and the cost paid as penalties for the SLA violations. The SLA violations which depend on the number of services that depends on the failure, the average time to repair the failure and the SLA penalties agreed on with the clients.

The optimal placement of personnel can be divided into two separate problems:

- The problem of optimal choice of sites where the personnel should be placed 6.2.

- The optimal number of personnel at each site 6.3.

First the optimal sites for 1,2,3,.... bases of personnel are solved. Then the optimal number of personnel at each site is identified under the restriction that a maximum number of personnel are not exceeded.

6.2 The optimal placement of reparation personnel

6.2.1 General introduction to location problem

The task of the reparation personnel is to repair the failures occurring in the network. The optimal placement of the base, where the reparation personnel are located, is the site where the sum of the distances to all other network elements is minimized. This is true under the prerequisite that all network elements of the same type have the same failure probability that is independent of their geographical placement.

In this master thesis we are looking into the concrete 50 node German backbone network "Germany 50"'. Referring to Figure 19 there is obvious that if this network was to be maintained from personnel all placed in one site Flensburg is not an optimal choice. A node that lies more in the middle will be a better choice simply because the average travel distance to the failures is less.

Network reparation personnel normally use cars as a means of transportation; hence they need to travel along roads to the site of the failure. The links in a cable based communication network does almost without exception follow dominating infrastructure as major roads and railways. Hence the physical distance between two nodes in the network can be simplified to the link distance between the two nodes.

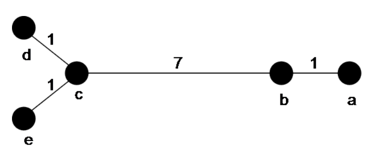

Figure 25: Example graph

In this approach the reparation personnel can only be placed in one of the already existing node locations in the network. This restriction is chosen because the network operator already has infrastructure (buildings) where the nodes of the network are placed. Another advantage of placing the personnel in the already existing nodes is that the nodes are primarily placed in major regional centers where the availability of qualified employees is better than in more rural areas.

When looking into a single location for the failure reparation personnel, the problem is reduced to finding the node that minimizes the total distances as shown in the following example: the graph matrix found in Table 7 corresponds to the network dissipated in Figure 25, where the values corresponds to the distances between the neighboring nodes. The Floyd-Warshall (FW) algorithm offers a fast solution to this problem and has a running time of impressing $\Theta(n^3)$ where n is the number of vertices in the graph [2].

	a	b	c	d	e
a	0	1	0	0	0
b	1	0	7	0	0
c	0	7	0	1	1
d	0	0	1	0	0
e	0	0	1	0	0

Table 7: Example graph matrix

Table 8 shows the result of running the FW algorithm on the matrix in Table 7. As it can be observed contains each row the shortest path from each node to every other node in the graph and by simply summing up all the elements of each row the total distance to every other node in the graph is obtained.

If each node in the network fails once during a certain time period Node c will obviously be the optimal location of the reparation personnel because the total traveling time to the failures will here be at its lowest. The node with the lowest sum after the FW algorithm is performed, in this case Node c, is the optimal node for placing the reparation personnel.

	a	b	c	d	e	SUM
a	0	1	8	9	9	27
b	1	0	7	8	8	24
c	8	7	0	1	1	17
d	9	8	1	0	2	20
e	9	8	1	2	0	20

Table 8: Result from performing the Floyd-Warshall algorithm on the matrix in Table 7

In a larger network structure than the one shown in Figure 25 there may be more efficient having more than one location of the reparation personnel. In this case the different sub graphs the original graph can be divided into have to be identified. This problem is known as the facility location problem and has been shown to be NP-Hard [15].

Brut forcing is the preferred way of solving NP-Hard problems, but if the solution space grows too big it becomes impossible to solve the problem within a realistic time perspective with the use of brutforcing.

The German backbone network shown in Figure 19 consist of 50 nodes that are connected with each other in a mesh structure. For the case of two locations of personnel in the network there are $50!/((50-2)! * 2!$ different solutions of placement. As Equation 9 shows are there $1.1259 * 10^{15}$ different solutions of where the reparation personnel can be placed in "Germany 50".

A 1.8 GHz computer is only able to process one solution i.e. finding the optimal sub-graph belonging to the location nodes within 1/15.5 of a second.

This means that it will take $2.3 * 10^6$ years to find the optimal solution for the entire set of sites [1,2,....,50]. The processing speed of an 1.8 GHz processor is roughly 8 GFlops and the fastest computer for the time being is the IBM BlueGene/L measured to 280.6 TFlops according to TOP500[5]. In other words the fastest known computer is only 5 magnitudes faster and it will still take 23 years to brut force the problem at hand i.e. the problem faced is too big to be brut forced.

$$\sum_{k=1}^{50} 50!/((50-k)! * k!) = 1.1259 * 10^{15} \quad (9)$$

Then optimality has to be sacrificed and replaced by a "good" solution obtained with an approximation algorithm, but of course we would like to sacrifice as little optimality as possible while gaining as much as possible in efficiency [5].

6.2.2 The solution approach for Germany 50

When faced with a NP-hard problem that is too big to be brut forced there is three alternative ways of solving the problem. All three will provide a "good" solution but can never guarantee that it is the optimal solution. The three alternatives is the use of: a randomized algorithm, a genetic algorithm or an intelligent removal of parts of the solution space.

Randomized algorithm: One way is to find a solution using a randomized algorithm which is an algorithm where some parts of the solution space are picked out randomly. As long as a wide enough solution space is picked is the probability of getting a good solution with the use of a randomized algorithm high. This approach also requires that the solution space not only consists of one optimal solution and the rest bad ones. In other words should the solution space not look like the one that can be observed in Figure 26 and more like the solution space shown in Figure 27. This is because the task of finding the optimal solution in this case will require the use of brut force or a large portion of luck.

Genetic Algorithm: In a genetic algorithm (GA) is a set of solutions created where each solution is given a rating how good it is based on certain criteria. A new "generation" of solutions is then formed on the basis of Darwin's principles. The "genes" from the solutions that are higher rated have a greater probability of forming the next generation of individuals. In this way a genetic algorithm for each new generation will create a population that is better adjusted to the environment or in other words solves the problem better.

Because the environment in this "world" is constant and not varying as in a real world, a "good" solution to the problem would probably be found within an

[5]TOP500 is an NPO aimed at ranking the 500 fastest public known computers in the world (www.top500.org)

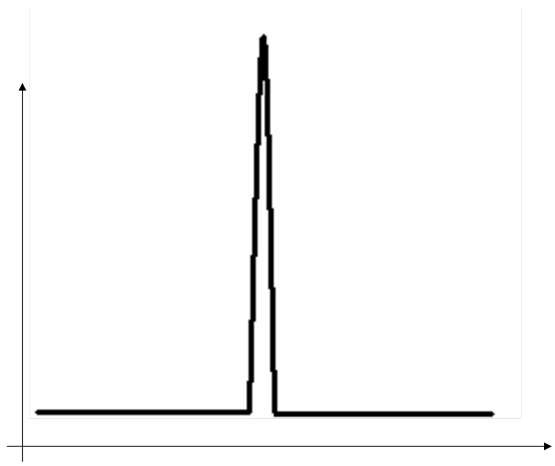

Figure 26: Unfavorable solution space

over seeable number of generations. One problem that can occur with the use of a GA can be that the algorithm gets locked in a so called "false" peak. This means that the GA will find a good solution of the false peak but the actual optimal solution is not a part of this "false" peak and is therefore missed. This problem is illustrated in Figure 27 where there are two so called "false" peaks and if the GA gets stuck in one of these two the real optimal will never be found.[12]

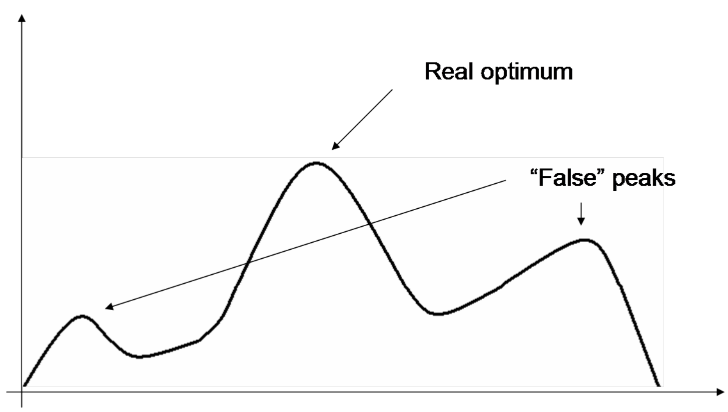

Figure 27: Example of solution space

Intelligent reduction of solution space: The other way to attack the NP-hard problem is to simply reduce the size of the problem. This approach can be used in cases where a set of the solution space can obviously not be an optimal solution. This can be a very effective way to reduce the problem and still find a satisfying "good" solution. When using this method it is important that the reduction of the possible solution space is the result of a rigorous judgment such that potential good solutions don't get lost. The parts of the solution space which are not removed in the reduction process, are then simply brut forced. Optimal solutions can then be found under the prerequisite that the intelligent reduction process did not remove the optimal solution.

The solution where parts of the solution space are removed can be an excellent approach, but requires that the selection is intelligent. For the case of the graph representing Germany 50 the task of removing some nodes that with a high probability do not belong to a "good" solution is considered a fairly easy task. For example in this case the edge nodes can be removed, since the core nodes have a more centric location in the network.

Comparison of alternatives: When comparing the different approaches against each other the most important factors in that decision making process is:

1. The time used to find the optimal locations has to be within an over seeable time period, meaning within hours.
2. The complexity to implement approach.
3. The expected "goodness" of the results, meaning how near the optimum the approximation algorithm comes.

The time used to find the "good" solution will affect the "goodness" of the result. The longer the time used the better the result obtained will be, this is valid for all three approximation approaches. The question is if one approach provides a better solution than the others within the same period i.e. the approach is more effective. It is plausible that the intelligent removal will be more effective than the randomized approximation approach. Both are simply reducing the solution space but the first approach should do a better job in the reduction process and therefore more effective. Regarding the second option, the use of a GA, this could probably do a very effective job of identifying the optimal starting nodes, and may be more effective than the two other options.

The randomized and intelligent removal approach are both simple to implement because it is a straight forward removal of parts of the solution space. The GA on the other hand is more complex to implement because a "gene structure" has to be constructed as well as a set of characteristics defining a "good" solution. To implement a GA to approximate the graph problem can also offer difficulties because the different "good genes" can be overlapping, this means that the good abilities from two individuals cannot be combined together in one new individual in the next generation.

With all three approaches there exists a risk that the provided "good" solution is in reality far away from the actual optimum. This can happen when the randomized algorithm has too few samples i.e. the optimal area of the solution space is very little. For the case of the GA the hazard of getting gridlocked in a "false" optimum as shown in Figure 27.

The approach "intelligent reduction of the solution space" is the one chosen in this study. This solution is chosen over the randomized one because it provides a solution that is likely to be better i.e. less optimum is sacrificed even though the speed is expected to be similar. The implementation of a GA could very likely produce a very good solution in a short amount of time, but the implementation is expected to be very complicated and the risk of getting stuck in a so called "false" optimum still exist.

6.2.3 The implementation of the location problem

There are two steps in the implementation of the location problem:

1. The intelligent reduction of the solution space
2. The identification of the optimal locations and their respective sub networks

The possible solution space is reduced with two methods:

- All solutions including a node with a lower nodal degree[6] than 4 are not investigated. Nodes with a high nodal degree also have a higher probability of being part of an optimal solution than nodes with a lower nodal degree. This is because nodes with a high nodal degree (usually core nodes) do not have to travel over so many intermediate links to reach the other nodes belonging to its network as nodes with low nodal degree (usually closer to the edge).
- In the second way of reducing the solution space the complete set of solutions formed on the basis of the nodes remaining after the first reduction process. The solutions where two or more nodes are neighbors are then removed from the solution space. A solution including neighboring nodes will very likely not be an optimal solution because an optimal solution is characterized by nodes in the center of sub graphs that equally divide the original mesh network between themselves.

To find the optimal locations and their respective networks a modified version of the Floyd-Warshall algorithm has been used:

The nodes defined as the starting nodes, the nodes where the reparation personnel are placed, are each the first member of a set of selected nodes. All the other nodes that are not starting nodes forms a set of nodes not yet selected.

The one node with the shortest distance to any one of the starting nodes is selected and does now belong to the set of nodes which the starting nodes it had the shortest distance to belonged to. If there are more nodes with an equally shortest distance a random one is chosen. This process is repeated until all nodes are selected.

[6]How many nodes a nodes is directly connected to with a link is the nodal degree of a node.

The links not yet belonging to a set i.e. the links dividing the original network up into disjoint sub network also has to be chosen to a set. They are distributed among the different sub networks on the basis of the same strategy, where the total distance from the starting nodes should be at a minimum.

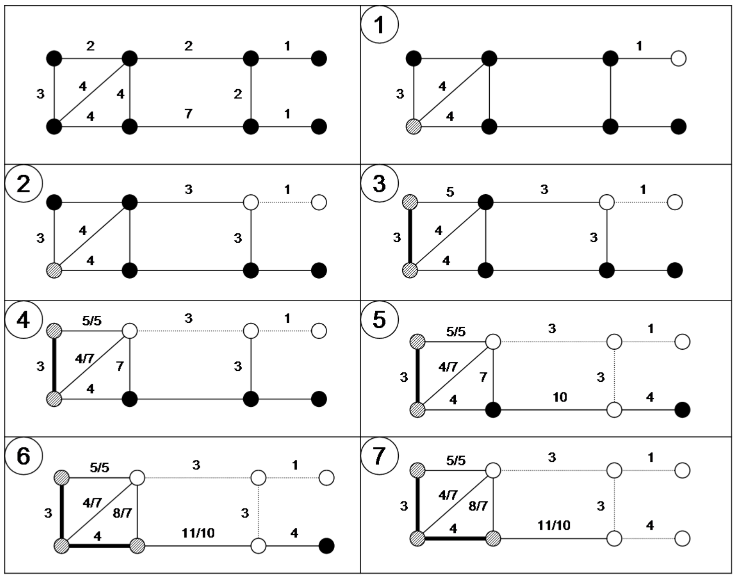

Figure 28: Animation of optimization algorithm

Figure 28 shows step by step animation of how the above described algorithm for a simple graph consisting of 8 nodes. The original graph can be observed in the upper right corner. In Step 1 it can be observed that the two starting nodes are taken and the distance to the neighboring nodes is updated. Step 2 shows that the right starting node includes the neighboring node that has the shortest distance to any one of the two starting nodes. This can be observed in Figure 28 as the node changes color to white and the link changes to a dotted line. This process is repeated until all nodes are finally selected in Step 7.

In Step 7 in Figure 28 are all the nodes selected but as it can be observed are there still links that does not belong to one of the two sub graphs. The four links not yet selected is assigned to the sub graphs based on the same algorithm as before that has the goal of minimizing the total distance from the starting nodes. Figure 29 shows the final division of the original graph into two sub graphs and as it can be seen is every link and node in the original network assigned to a starting node i.e. every network element is a assigned to the responsibility area of a reparation location.

For each case of location placement the total distance to all nodes in the network from their respective starting nodes is compared with the total distance of all other cases with the same number of starting nodes and the one with the lowest total distance is the optimal solution. This process is then repeated for

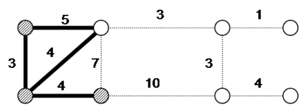

Figure 29: The final division of the original graph

the other cases with different number of starting nodes.

6.2.4 Optimal sub graphs results

The optimal sub graphs for 1-18 locations in the "Germany 50" have been identified in this study. Nodes with a lower nodal degree than three have not at all been considered as potential locations for the reparation personnel. Nodes with a nodal degree of four or higher have all been considered as potential location nodes. For the nodes with a nodal degree of three have some been included as potential locations based their central location in the network. The reduction of the potential locations has been done because the problem would otherwise be too big to brutforce.

For until ten locations have the strategy where a potential solution can not contain nodes that are neighbors been used. For the cases for ten and more locations have this restriction been loosened such that also sets containing neighboring nodes can be a part of the optimal solution.

When the network should be maintained from only one location in the network has Fulda been identified has the optimal solution. In Table 9 are the optimal nodes for the cases of five and ten reparation locations in the network. The case of 15 locations is presented in Table 14

Five locations	Ten locations
Bremen	Aachen
Dortmund	Bremen
Karlsruhe	Dortmund
Leipzig	Giessen
München	Karlsruhe
	Kempten
	Leipzig
	München
	Schwerin
	Würzburg

Table 9: The optimal nodes for five and ten locations

6.3 The optimal number of reparation personnel

6.3.1 The optimization problem

In every type of network there are failures occurring and these have to be repaired by reparation personnel before the network can return to a state of normal operation. A normal telecommunication network covers an extensive geographical area and includes numerous nodes and links which each are failing with a specific rate.

When a failure occurs in the network one or several reparation personnel has to travel to the failure site and repair the failure. The reparation time of a failure is the time from when the failure occurs until the failure is repaired. If the failure occurs at a time when there is no available reparation personnel there would be a waiting time until there are available personnel that can repair the failure. Hence the total repair time is the sum of the time used for the reparation itself (T_{rep}), the traveling time of the personnel to the failure (T_{trav}) (except for the case when the failure occurs at the site where the personnel is placed) and the eventual waiting time before there is personnel available to perform the reparation (T_{av}). Equation 10 expresses the total time from a failure occurs until it is repaired.

$$T_{tot} = T_{rep} + T_{trav} + T_{av} \tag{10}$$

Each area covered by a site of reparation personnel will have a limit to how many personnel should be placed at the site. When the number of employees at a site is increased the average waiting time before the personnel starts on the task of repairing the failure is decreased. At some point there will be enough employees such that at the moment when a failure occurs there will always be reparation personnel to fix it to fix it i.e. $T_{av} = 0$. Hence every employee added after $T_{av} = 0$ does not lead to a detriment in reparation time.

The reparation personnel are placed at different sites in the network and the more sites in the network, the higher is the cost. This is because more sites imply more energy consumed, material, support, etc. Hence each extra site where reparation personnel are placed has an associated penalty.

A scenario where all reparation personnel is placed at one site will have a very low infrastructural cost but will have a high cost related to SLA violations because the average traveling time to the failures would be very high. The opposite scenario where there are a high number of sites, would have a low cost related to SLA violations but a high cost due to the extensive infrastructure needed. The optimization problem is to find a solution that minimizes the overall cost including: the employees, infrastructure and the SLA violations.

6.3.2 The method

The optimization problem of one location can be expressed as an equation shown in Equation 11. The employees, the number of locations and the time length of the SLA violations is denoted as e, s and t respectively. The cost of each employee and the cost of each location is denoted Ce and Cs while the cost related to the SLA violations is expressed as a function $v(t)$ of the time the SLA is violated.

$$F_{cost}(e) = e * Ce + s * Cs + v(t) \tag{11}$$

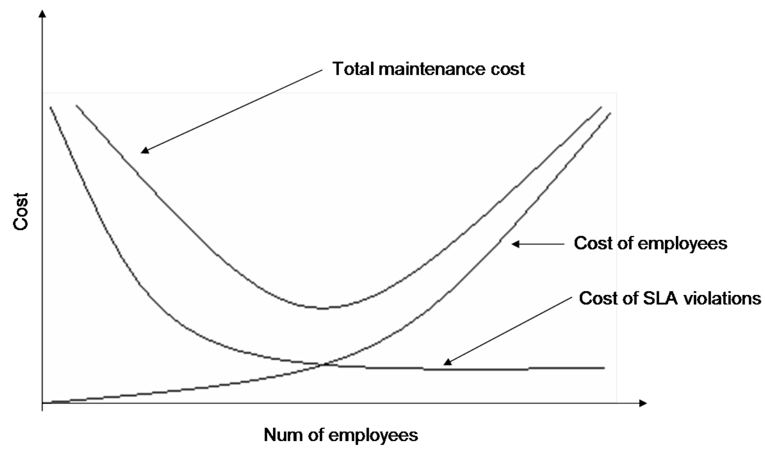

Figure 30: Graphical representation of single optimization problem

Figure 30 shows an expected cost graph of the employees for the location case. This includes direct expenditures salary, car, tools, etc. and indirect costs generated by infrastructural requirements.

The cost of the SLA violations is a result of how many and how long the SLA violations are. The graph shown in Figure 30 is shows the expected behavior of the SLA violations costs.

The graph representing the total reparation cost is the sum of the cost of the employees and the SLA violations. The optimal number of employees in this graph representing expected results is where the total reparation cost is at a minimum.

When the optimum for each of the sub networks the original graph has been divided into is found, is also the optimum for the whole network found.

6.4 Network scenarios

Two different types of network platforms are investigated with regards to optimizing the reparation cost, the "E/SDH" and the "E-MPLS" further explained in Section 2.2.2, which both are test beds available at T-Systems. Both platforms are compared when having the same topology as Germany50 (Figure 19).

Three types of failures are considered:

1. Link failures: One link consists of a bundle of several fibers, in our case 192 fibers. The occurrence of a fiber cut that is only affecting one or a few fibers is very unlikely since almost every fiber cut is caused by an excavation of some kind [4]. Since excavations normally cut the whole bundle of fibers only cable cuts (link failures) and not single fiber cuts are considered in this study.

2. Node failure: It occurs when the hardware at a node fails. This hardware failure could be caused by normal wear and tear, lightning, overheating, etc. Reparation of this type of failure normally consists of simply replacing the failed hardware components with new ones.

3. Interface failure: An interface failure is not that serious and time consuming as a normal node failure because it only affects one card.

There have been assumed that the exact location of the failure is know as it occur i.e. the failure location time is assumed to be equal to zero.

All these failures have an availability parameter which is defined as the degree to which a system, subsystem, or equipment is operable and in a committable state at the start of a mission, when the mission is called for at an unknown, i.e. random time.[1] Or in more common words the time an element is in a functional condition. The availability of an element is obtained as shown in Equation 12 where availability is denoted q, where MTBF is mean time between failures and MTTR is mean time to reparation.

$$Availability\, q = (MTBF - MTTR)/MTBF \tag{12}$$

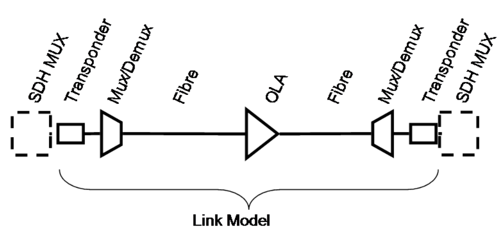

Figure 31: Link Model for E/SDH

Figures 31 and 32 show the link model used for calculating the availability of the link as function of its length. The availability of a "series" of components is the product of the availability of each component. An overview of the link availability values for "E/SDH" and the "E-MPLS" links are presented in Table 10. The values are based on the numbers given in [18].

The availability of fiber cable depends on its length. This can be expressed as a function of the fiber length shown in Equation 13

$$FC(l) = 0,9999969582^{l} \tag{13}$$

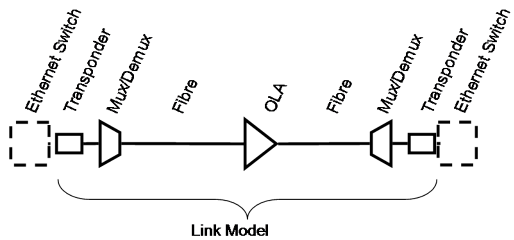

Figure 32: Link Model for E-MPLS

Component	MTBF	Availability
Transponder	4,00E+05	0,9999800000
Mux/Demux	1,67E+05	0,9999520958
OLA	2,50E+05	0,9999680000
Fiber(1 km)	2,63E+06	0,9999969582

Table 10: Link availability of both platforms [19]

The availability of the OLSs (Optical Level Attenuator) in a link also depends on the link length. There is as an average one OLA for every 80 km on a link. E.g. a link that is 160 km long will have one OLA. The availability of the OLA's on a link of length l is shown in Equation 14. In Equation 14 is a round function used to get an integer from the division in case the result is a decimal number. It is assumed that the rounding function will give a realistic result of the number of OLAs on a link.

$$O(l) = round(l/80 - 1) * 0,999968 \tag{14}$$

The total availability of a link is a product of the availabilities of the link components shown in Figure 31 and 32. The availability of a link of length l is the same for both platforms. The availability of a link as a function of the length l is shown in Equation 15 where A_{Trans} and $A_{M/D}$ is the availability of a transponder and a MUX/DEMUX respectively.

$$L_{Total} = A_{Trans}^2 + A_{M/D}^2 + FC(l) + O(l) \tag{15}$$

There is more than one interface per node and in this study is the number of interfaces assumed to be 100. Hence is each node simulated with 100 interfaces.

Tables 11 and 12 gives the availability, MTTR and the number of persons required for the reparation, for each of the three failure scenarios for "E/SDH" and the "E-MPLS" platform respectively.

As it can be seen in Table 11 and 12 is the number of employees needed to repair a link failure three persons. The cases of one and two persons placed at reparation location not investigated and the each location is assumed to have at least three employees working there.

Type of failure	Availability	MTTR(hours)	persons/failure	Cost
Link	See Table 10	7	3	1,62BS
Node	$0,999984$	5	1	1,62BS
Interface	$0,999994$	2	1	1,62BS

Table 11: The failure data for the "E/SDH" platform

Type of failure	Availability	MTTR(hours)	persons/failure	Cost
Link	See Table 10	7	3	1,62BS
Node	$0,99996$	5	1	1,62BS
Interface	$0,999985$	2	1	1,62BS

Table 12: The failure data for the "E-MPLS" platform

The final assumtion made in this study is the traveling speed of the reparation personnel. The German department of transportation in Berlin operate with the average speeds of $110km/h$, $80km/h$ and $70km/h$ for the "Autobahn", " Bundesstrae", and "Landesstrae" respectively according to "Bundesministerium für Verkehr, Bau und Stadtentwicklung". These are the values used for calculations regarding road planning in Germany. The reparation personnel will travel along all three types of roads on their way to reach the failures occurring in the network.

In addition to these road types are the reparation personnel also traveling on roads within cities where the average speed is even lower. Other factors affecting the traveling time can be: not all failures are reachable by road, the link failures that are not due to excavations have to be exactly located before they can be repaired, coffee breaks, etc. With all these factors taken into account we have assumed an average traveling speed of $70km/h$.

6.5 Network costs

The cost unit used in this study is Basic Salary (BS) per month. BS is the cost from the operator point of view, of a basic technician. In our case the cost is higher (1,62BS) since the employee requires extra material, car, etc which increases the cost for the operator.

$$cost\,employees(CE(e)) = 1,62BS * e \tag{16}$$

Concerning the cost of the location itself is the cost of having the location 2BS per month for every 10 employees working at the location i.e. a location having 1-10 employees has an cost of 2BS and a location with 11-20 costs 4BS and so on for the each group of ten employees. Equation 17 shows the cost of each location as a function of the number of employees e. Ceil is a rounding function that rounds the number up to the nearest integer.

$$cost\,location(CL(e)) = ceil(e/10) * 2BS \tag{17}$$

The cost of the SLA violations is based on the duration D of the SLA violation. The longer the SLA is broken the more costly are the damages that the network operator has to pay to the client. Each time the reparation time of a failure exceeds the SLA specifications the cost of this SLA violation is calculated on the basis of $F_{prot}(D)$, $F_{prot}(D)$ and $F_{prot}(D)$ shown in Equation 18,19 and 20.

$$F_{prot}(D) = S_{prot} * ceil(D/0,25 - 1) * 9,375 \tag{18}$$

$$F_{rest}(D) = S_{rest} * ceil(D/3 - 1) * 18,75 \tag{19}$$

$$F_{unprot}(D) = S_{unprot} * ceil(D/12 - 1) * 12,5 \tag{20}$$

The S_{prot},S_{rest} and S_{unprot} are the number of protected, restorable and unprotected services affected by the failure. The routing and actual traffic on the links in the network is not a part of this study, to calculate the number of services affected by a failure is average values used. There are an average number of 23800 services on a link where 80% is protected, 14% restorable and 6% unprotected. For the node and interface failures are the number of services obtained by multiplying the number of links the failure node has with 23800 e.g. a node with three neighbors has $23800 * 3 = 71400$ services it. The same percentages of protected, restorable and unprotected are also applied to the node and interface failures.

If a service is affected or not by a failure depends on the protection scheme the service is implemented with. It is assumed that 100% of the unprotected services in a link will be affected by the failure. For the case of the restorable and the protected services are the numbers respectively 5% and 0,1%. Meaning that even some of the protected services are affected even though they should be protected against failures. This occurs because multiple failures can occur in the network and the services are never 100% protected against those.

An overview of the how the different services are affected by a failure is given in Table 13.

	% of services	% Affected	Services affected by failure
Protected	80%	0,1%	19,04
Restorable	14%	5%	166,6
Unrestorable	6%	100%	1428

Table 13: The number of services (23800) affected by a failure

Until now, in the cost model presented here, have only one employee been considered. It is for obvious reasons unrealistic that one employee is working 24 hours seven days a week. Most failures occurs during normal working hours this is due to the general higher activity in the society (e.g. more digging = more cable cuts, higher network load = higher number of failures, etc.). A network operator will therefore need a higher number of employees working during day time than during the evening and night. On the other hand does the employees' working night get better paid than the colleagues working day time. This together with other factors as vacations, sickness, etc is factors in our assumption that the number of employees covering 24 hours is three employees.

The total monthly cost of a sub network where three employees covering 24 hours is then given in Equation 21 and the goal is to minimize this function i.e. find the optimal number of employees for the actual sub network.

$$Total\,monthly\,cost = CE(e) + CS(e) + F_{prot}(D) + F_{rest}(D) + F_{unprot}(D) \quad (21)$$

6.6 Comparison of different number of base stations

The different cases from 1 trough 18 locations of reparation personnel are individually investigated and optimized. The optimal number of employees for each location is found with the help of Equation 21. By optimizing all the sub networks in one case the optimal cost for the whole case is found and thereby the optimal number of locations can be found.

The results presented here are performed with the E/SDH platform as basis, but the E-MPLS is expected to give similar result.

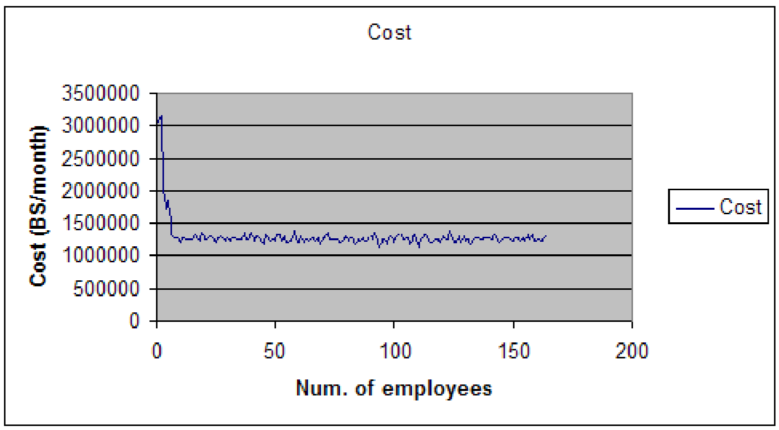

Figure 33: Cost graph of one location

Figure 33 shows the monthly cost of repairing the network, with only one location of the reparation personnel, as a function of the number of employees. As it can be observed is the cost of repairing the network very high as the number of employees is low. This is due to the serious SLA violations occurring because there simply are not enough available personnel to cope with the failures.

For a certain number of employees the reparation cost seams to stabilize. It is stabilizing because when a certain number of personnel are reached they can cope with the failure at the moment it occurs. Meaning that adding another employee at this point does not lead to a decrease in reparation time (i.e. SLA violation costs) only an increase in personnel costs. The reparation cost will therefore start to increase at this point, referring to Figure 30 explaining the expected results. This is impossible to observe here because the cost related to the personnel is microscopically compared to the cost of the SLA violations. The increase is drowning in the cost variations of the SLA violations due to the randomized simulation.

This does not mean that the total monthly reparation cost cannot be further decreased. The total monthly reparation cost can be further decreased when the network is maintained from two or more locations. Each location in these cases will behave in the same way as the graph in Figure 33. The cost of each location will also stabilize after a certain number of locations is reached, but the cost of the locations all together will stabilize at a lower level that. E.g. the cost of reparation when the network is maintained from two locations is lower than the cost of maintaining it from one, even though the number of employees is even higher. This is due to the more effective way of maintaining the network i.e. the SLA violations are lower.

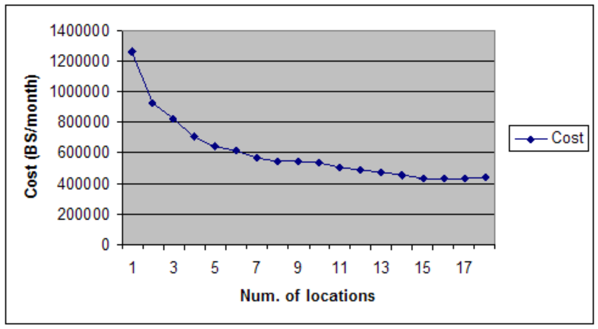

Figure 34: Cost graph of one location

The total monthly reparation cost will continue to decrease as the number of locations increases until a certain point where the decrease in SLA violations is lower than the increase in cost due to the extra location. This is due to the higher penalty of having more locations than fewer, which means that each new location added will increase the total monthly personnel cost. The graph in Figure 34 shows that the total cost decreases as the number of locations increase. It can be observed that an optimum number of locations are reached at 15 reparation locations in the network and thereafter the total cost starts to slowly increase.

The cost difference between the different locations scenarios can clearly be seen, but the result is not accurate enough, at this level, to identify the optimal number of employees for each location. Hence the optimum, 15 reparation locations, is then further studied with a larger number of simulations such that the optimum for each location can be found.

In Table 14 is the 15 reparation locations in "Germany 50" and their respective optimal number of employees presented. The geographical position of the nodes can be observed in Figure 19.

6.7 Conclusions

This section has presented a method that finds an optimum number of locations and employees, which allow finding a compromise between their cost and the penalties that should be paid for long failure reparation time. We have applied this method to the "Germany 50" network given some proposed SLA violations values. Each operator can apply this method with their own values and combine it with other parameters such as available floor space, personnel costs, etc. which may vary from operator to operator.

Name of node	Opt. num. of employees
Dresden	6
Erfurth	5
Giessen	7
Hannover	7
Kaiserslautern	3
Karlsruhe	7
Köln	7
Magdenburg	6
München	6
Münster	6
Oldenburg	4
Regensburg	7
Schwerin	7
Ulm	5
Würtzburg	5

Table 14: The optimal nodes and their optimal number of employees

7 Discussion

This master thesis consists of two parts, the modeling and implementation of the cost models that enables the identification of the most important factors and the development of a scheme for reducing the most important cost factor identified in the first part. Although they could be viewed as two separate parts they are tightly connected with each other and together they form a complete scheme of cost reduction in a realistic network scenario.

In the first part, the model and implementation of network costs includes existing service cost models. However, the implementation of the service cost model has been drastically improved from the computational time point of view.

As it is shown in Section 6.2.2 the identification of the optimal placement of the maintenance personnel is a NP-Hard problem that is too big to be brutforced. The method for solving the problem is the intelligent reduction of the solution space. First of all, this approach requires a well considered method where the probability of remove very "good" solutions should be very small. The benefits of using the approach of intelligent removal is that the set of solutions remaining after reduction process is brutforced, which means that the result will be the optimal solution if this is not removed under the reduction process.

The reduction method used in this study is a combination of two different methods: the removal of nodes with a low nodal degree and the removal of neighbors. Of course a manual removal of nodes from the solution space could be chosen instead, but a consistent approach like the one chosen could be used on several different network structures.

The removal of nodes with a low nodal degree from the solution space can be justified because of two reasons: first the nodes with a low nodal degree are almost without exception node near the edge or edge nodes. The optimal division of a network into sub networks will be a division where the sub networks are similar in size. The optimal location of the maintenance personnel within such a sub network will be a node lying towards the center of the sub network, therefore edge nodes in the original network will with a very low probability be a part of an optimal solution. The second reason is that travels originating in a node with low nodal degree will be longer in average than the ones originating in nodes with high nodal degree.

The removal of neighbors is a very effective way of reducing the solution space before performing the optimization. Although it is very unlikely that close neighbors in a network both will be a part of an optimal solution will the can the neighbor removal remove optimal solutions even tough they are neighbors. This can be observed in Figure 35 where node a and b would obvious be the optimal nodes when there should be two nodes maintaining the network shown in the figure, but they are at the same time also neighbors. This scenario is considered to be very unlikely to occur in "Germany 50" at the same time as the benefits of using the neighbor removal is greater than the detriments of using it.

The results given of the simulations may suffer from the approximation algorithm used to find the optimal graphs. I.e. there may exist more optimal

Figure 35: Example graph

graphs than the ones found with the use of the approximation algorithm.

Another uncertainty factor is introduced when only 1-18 maintenance locations in the network are investigated. The optimum identified in section 6.6 may therefore only be a local optimum and not a global one as assumed. An optimal solution would when all cases of 1-50 maintenance locations in the network would have been investigated.

There are some simplifications also done regarding the simulation of the network maintenance that is that routing is not considered to be a part of the model. In reality the effects of link cuts and node failures will be met with countermeasures to some degree something that is not taken into account in this work. The services that are protected will be rerouted in case of a failure and those that are not protected or restorable will on the other hand be affected by the failure. This is only taken into account in this master thesis with the use of average number of services affected by the failure. The kind of traffic that is using the failed network element will affect the cost of the failure i.e. the routing will affect the cost of failures. The number of services affected by a network failure will also vary between the different types of failures and the geographical position of the failure. A more realistic simulation would be provided if the routing and actual traffic was a part of the simulation.

In the simulation performed in this master thesis an average cost of a SLA violation is assumed. The cost of an SLA violation does not depend on the routing or the geographical placement of the network element failing it only depend on the length of the SLA violations. Hence a longer SLA violation is more costly than a shorter SLA violation. This simplification works under the requirement that the failures taking place in the real network are uniformly distributed, hence the average values of SLA violations can be used in the network maintenance simulation.

Another simplification made in the simulations is that it is assumed that all cable ducts goes along a road thus can the maintenance personnel always travel along the links. The "Germany 50" has the same characteristics as the as other cable based communication networks, the cable does almost always follow other major infrastructure which already have the right of way. This justifies the second simplification done in the simulation of the network maintenance.

A third simplification done in the simulation is that the node where the maintenance personnel are placed only is responsible for repairing the BBN. In reality are these locations also responsible for repairing regional networks in their domain. The study performed here does only take the BBN into account and performs a study aimed at finding the optimal placement in it.

A fourth simplification is that the locations of the failures i.e. finding the exact location of where the failure has occurred is done automatically and takes no time. Every link failure occurring in the network simplified to happen at exactly the middle of the link. This is done because the failure will as an average happen at the middle point of the links provided that the link failures happen uniformly.

8 Conclusion

In a telecommunication market that is under the constant scrutiny of reducing cost end increasing profit margins is the ability to identify and reduce important cost factors vital for a network operator. This master thesis is a part of series of studies related to OpEx studies of networks and the cost factors identified and reduced in this work is part of the OpEx of a network.

The work presented in this Master thesis a complete scheme for cost reduction in a real network scenario. First the most important OpEx cost factor is identified by using a combination of previous and new work. A scheme for the cost reduction of the most important cost factor has been developed. Due to the importance of this cost factor the proposed reduction scheme will lead to a substantial over all cost reduction for the network operator.

In the first phases of the study is new work combined together with existing work to form a tool that enables the identification of the most important cost factors of OpEx. The previous work used in this study is solved with an analytical approach in this master thesis, which offers a substantial improvement of the computational time.

A model for computing the network layer cost of OpEx was conceptualized and modeled in such a way that it could be combined with the previous work of done on the service maintenance and service migration. A common integrated cost model of the OpEx consisting of all three areas was the constructed.

Different network implementations of the German backbone network "Germany 50" were then investigated with the goal to identify the most prominent cost factors of OpEx. The performed studies showed that the failure reparation cost, a network layer cost, is the most important single cost that can be found when breaking down into the results of the studies performed.

As a network operator would do while trying to minimize OpEx we focused on how to reduce the failure reparation process cost. In every type of telecommunication network there are failures occurring and these have to be repaired by maintenance personnel before the network can return to a state of normal operation. The cost of failure reparation consist of the maintenance personnel traveling out to the failure and repairing it i.e. the reparation process, the cost penalty of possible SLA violations and the cost related to the infrastructure required by the employees stationed at a base.

The problem can be divided into two parts:

- The optimal choice of sites where the personnel should be placed. This was done for each case between 1 and 18 locations in the network. The cases where there are 19 or more locations in the network was not investigated because it was computational too intense.

  The problem of finding the optimal placement for each of the different numbers of locations in the network was identified as a NP-Hard problem. While the solution space of this problem was to big to be brutforced other means of solving the problem had to be found. A method where parts of the solution space were intelligent removed and where the remaining part of the solution space was simply brutforced was chosen. With this method the "good" solutions for the cases of 1-10 locations of the maintenance personnel in the network was found. Brutforcing is the only guaranteed way to find an optimal solution of an NP-Hard problem, hence the word "good" is used for the result instead of optimal.

- For each number of locations, find the optimal number of personnel for each location so that the total reparation cost is minimized. This total reparation cost includes the cost of the personnel (i.e. salary, equipment,etc.), location cost and the SLA violation cost, which depends on the reparation time.

  The second part of the optimal location problem is to find which number of locations of maintenance personnel is the optimal one. We have then taken the result provided to us by the first part of the location problem and used it as input together with cost figures related to failures. The monthly failure cost of each location and its associated sub network is calculated on the basis of a simulation of the failures occurring in the real network case. An optimal number of employees for the location and its associated sub network are then found.

  The different locations belonging to each case is then summed together and a monthly reparation cost for each case is calculated. The most efficient solution of placing the maintenance personnel is the identified by simply finding the case where the monthly reparation cost is at a minimum.

Further research should be done on the subject of using a Genetic Algorithm (GA) to find the optimal placement of the maintenance personnel. This GA has to cope with the problems of getting gridlocked in a false optimum and the problem of overlapping optimal solutions. The simulation model used to find the optimal number of employees maintaining the network should in the future be expanded so that the actual routing and traffic in the network is taken into account.

The implementation of a GA will provide us with optimal locations for 1-50 locations in the network hence verification if the optimum found is global optimum or can be given. The GA will also offer a verification of the optimal sub graphs found with the approximation algorithm used here.

In this master thesis a complete cost reduction scheme has been presented. From the identification of the salient cost factors of the network and the creation of a cost reduction scheme for the factors identified. Although the work preformed here is adapted and specialized for the cost reduction of the backbone network "Germany 50" is the main parts of the principles used here adaptable to other network with only small changes in the structure.

Appendix A

The figures in this appendix gives the complete overview of the sub processes of the network life cycle.

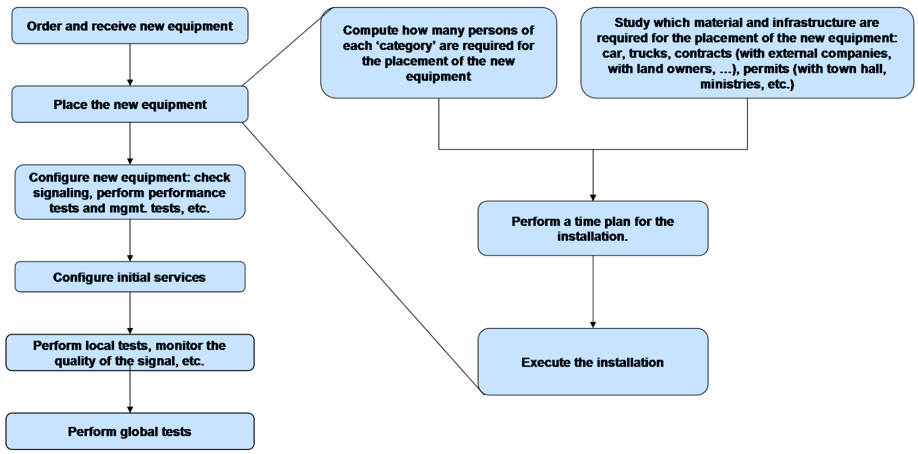

Figure 36: Place the new equipment

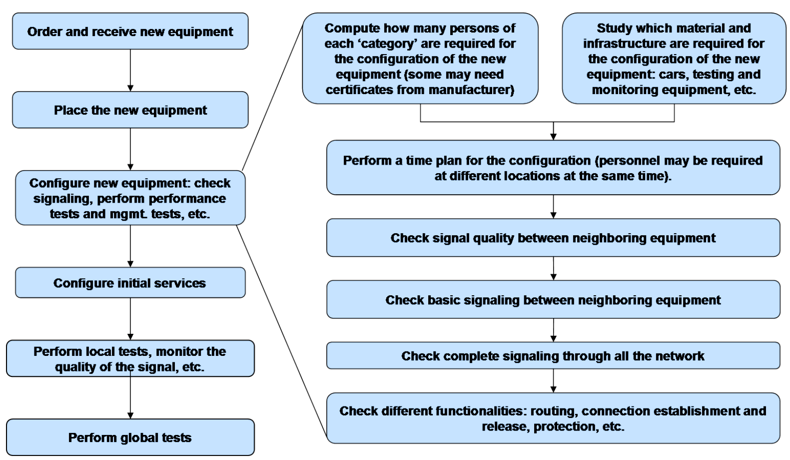

Figure 37: Configuring the new equipment

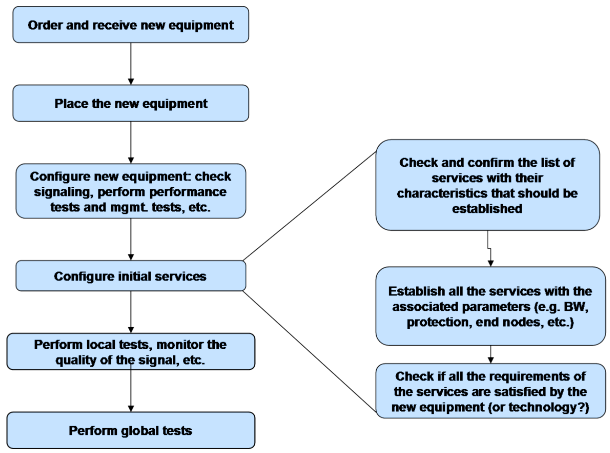

Figure 38: Configuring initial services

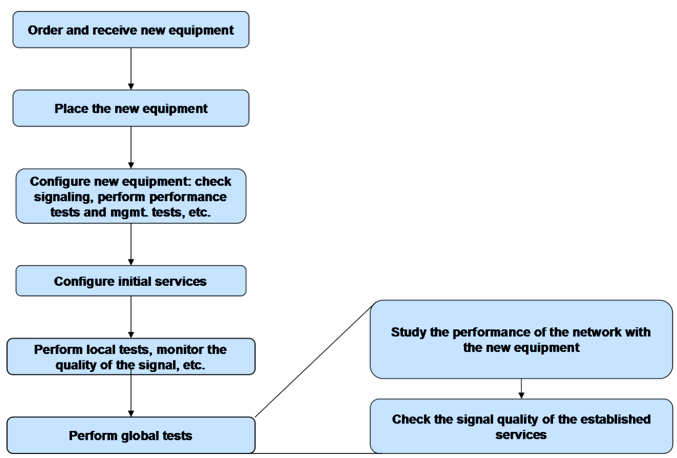

Figure 39: Perform global tests

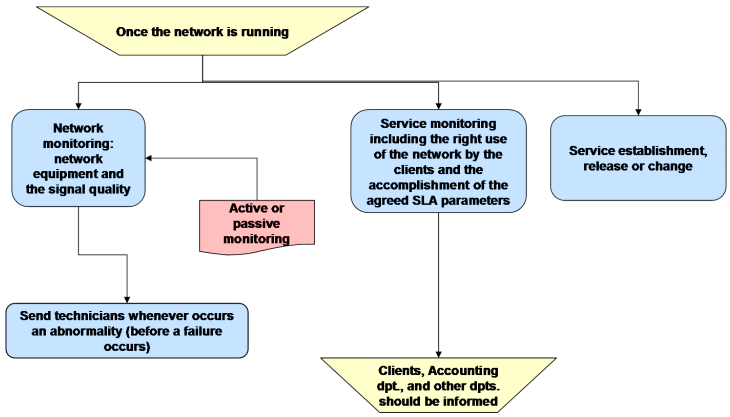

Figure 40: Network maintenance

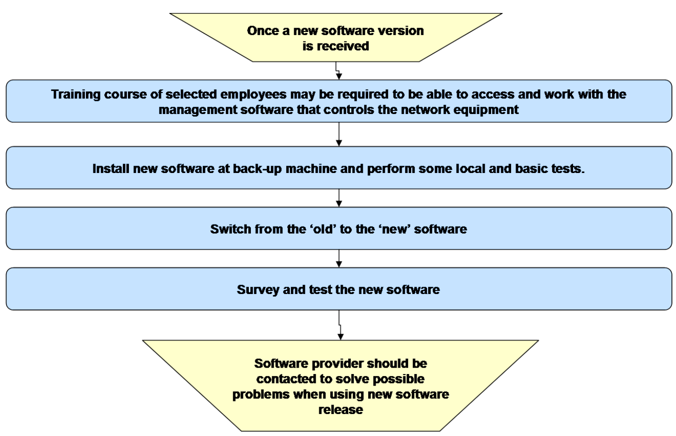

Figure 41: Software upgrade

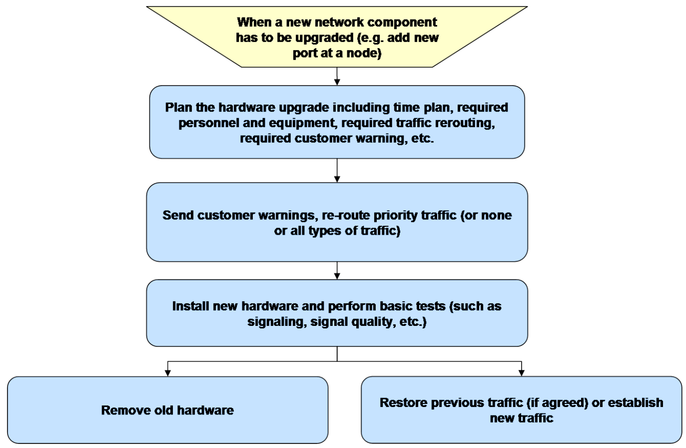

Figure 42: Hardware upgrade

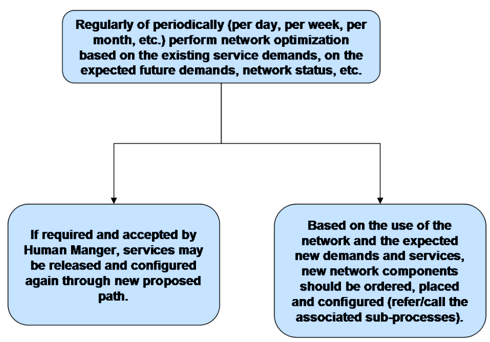

Figure 43: Short term planning

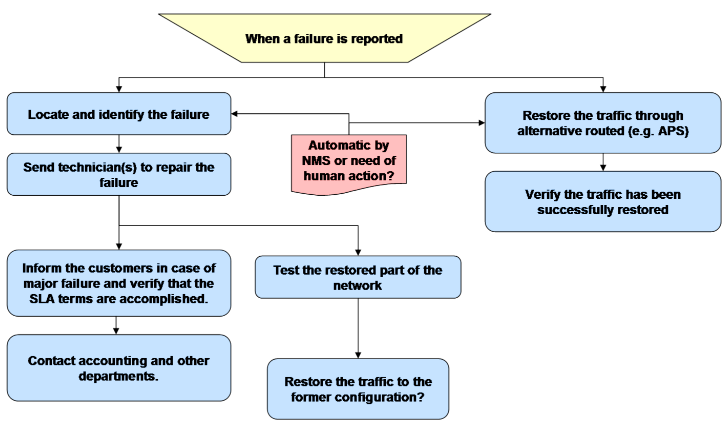

Figure 44: Failure reparation

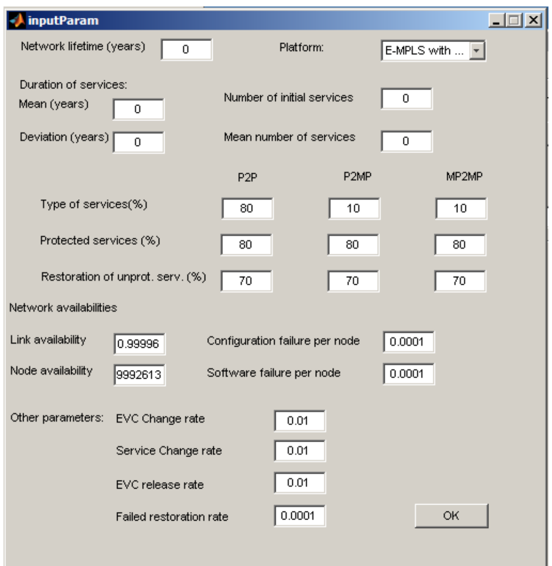

Figure 45: Grapical User Interface

List of Figures

List of Tables

Abbreviations

APS	Automatic Protection Switching
BS	Basic Salary
BBN	Backbone Network
CapEx	Capital Expenditures
CI	Confidence Interval
CO	Customer Object
DEMUX	De-Multiplexer
DTMC	Discrete Time Markov Chain
E-MPLS	Ethernet Multiprotocol Label Switching
E/SDH	Ethernet Synchronous Digital Hierarchy
EVA	Ethernet Virtual Connection
FR	Failure Reparation
FW	Floyd War shall
GA	Genetic Algorithm
GUI	Graphical User Interface
HUN	Hardware Upgrade
IRC	Installation and Initial Configuration
I/O	Input/Output
LIP	Long Term Planning
MO	Mechanic Object
MOM	Markov Reward Model
MOTIF	Mean Time Between Failures
MTTR	Mean Time To Reparation
MUX	Multiplexer
ND	Network Dismantle
NO	Network operation
NOC	Network Operations Center
NP	Non Polynomial
OLA	Optical Level Attenuator
OO	Object Oriented
OOS	Object Oriented Simulation
OpEx	Operational Expenditures
P-Router	Provider Router
PE-Router	Provider Edge Router
SLA	Service Level Agreement
STP	Short Term Planning
SU	Software Upgrade
TBO	Total Benefit Ownership
TCO	Total Cost of network Ownership

References

[1] *Glossary of Telecommunication Terms (FS-1037C).* General Services Administration Information Technology Service, 1996.

[2] Thomas H Cormen, Charles E Leiserson, and Ronald L Rivest. *Introduction to Algorithms.* MIT Press and McGraw-Hill, 1990.

[3] C. Evans. Cost justifying the carriers migration to strategy to ethernet. *WDM & Metro optical networking*, July, 2004.

[4] Wayne D. Grover. *Mesh-Based Survivable Networks.* Prentice Hall PTR, 2003.

[5] Dorit S. Hochbaum, editor. *Approximation Algorithms for NP-Hard problems.* PWS Publishing Company, 1997.

[6] Jeffrey A. Joines and Stephen D. Roberts. Fundamentals of object-oriented simulation. In *Fundamentals of Object-Oriented Simulation*, 1998.

[7] John G. Kemeny and J. Laurie Snell. *Finite Markov Chains.* Springer-Verlag, 1976.

[8] Carmen Mas Machuca, Jörg Eberspächer, Monika Jäger, and Andreas Gladisch. Service migration cost modeling. In *In ITG-Fachtagung "Photonische Netze"*, 2007.

[9] Carmen Mas Machuca and Ø yvind Moe. Impact of protection on the cost of ethernet services on different optical network platforms. In *IEEE International Conference on Transparent Optical Networks (ICTON)*, 2007.

[10] Carmen Mas Machuca and Ø yvind Moe. Modeling of opex in network and service life-cycles. In *ECOC-TUM*, 2007.

[11] Carmen Mas Machuca, Ø yvind Moe, Jörg Eberspächer, Monika Jäger, and Andreas Gladisch. Cost comparative study of ethernet services for different platforms. In *In 6th Conference on Telecommunication Techno-Economics*, 2007.

[12] Melanie Mitchell. *An Introduction to Genetic Algorithms.* The MIT Press, 1996.

[13] Ø yvind Moe. Study of the integration of the different nature opex costs of a network. *Project assignment*, 2007.

[14] S. Sengupta, V. Kumar, and D. Saha. Switched optical backbone for cost-effective scalable core ip networks. *IEEE Communications Magazine*, June, 2003.

[15] David B. Shmoys, va Tardos, and Karen Aardal. Approximation algorithms for facility location problems. *Annual ACM Symposium on Theory of Computing*, 1998.

[16] O. Soto. Network planning - business planning. In *ITU/BDT - COE workshop*, 2002.

[17] S. Verbrugge, D. Colle, Pickavet M., and P. Demeester. Capex comparison between link-by-link and end-to-end grooming in a european backbone network. In *Telecommunications Network Strategy and Planning Symposium: NETWORKS 2004*, 2004.

[18] Sofie Verbrugge, C. Colle, Piet Demeester, R. Hulsermann, and Monika Jäger. General availability model for multilayer transport networks. *Design of Reliable Communication Networks*, 2005.

[19] Sofie Verbrugge, Sandrine Pasqualini, Fritz-Joachim Westphal, Monika Jger, Andreas Iselt, Andreas Kirsstdter, Rayane Chahine, Mario Pickavet, and Piet Demeester. Modeling operational expenditures for telecom operators. In *Optical Network Design and Modelling (ONDM)*, 2005.

[20] I. M. Vinograd, editor. *Encyclopaedia of mathematics*. Kluwer Academic Publishers, 1988.

[21] Markus Walker, Greg Ip, and Andrew Batson. The high cost of growth. *The wall street journal*, June 6, 2007.

[22] Zuse-Institute Berlin (ZIB). Sndlib. *http://sndlib.zib.de/home.action*, 2006.

[23] Bernhard P. Ziegler. *Object-Oriented Simulation with Hierarchical, Modular Models*. University of Arizona, 1995.