Architecture of a Network Control Server for autonomous and efficient operation of Next Generation Networks

Uwe Walter, Martina Zitterbart
Institut für Telematik, Universität Karlsruhe (TH)
Karlsruhe, Germany
{walter,zit}@tm.uka.de

Joachim Charzinski
Siemens AG
München, Germany
joachim.charzinski@siemens.com

Abstract

Due to the growing importance of Internet connectivity and the increasing pressure of competition, network providers must be able to realize and operate modern Next Generation Networks (NGN) in an economically efficient manner.

We present the architecture and function of a Network Control Server (NCS) prototype that has been developed for deployment in an NGN to keep that in an acceptable operating condition. For this task, the NCS gathers statistical data from the network and uses this information to generate a global view of the current operating condition. Using this information, the NCS follows its internal strategy and uses optimization algorithms to adapt the network to changing operating situations.

Among the benefits of the presented approach are an increased network resilience with a quick reaction towards changes in the offered traffic, a reduced QoS request blocking rate and a more efficient network operation. These have been evaluated in simulations and shown in a functional lab prototype.

1 Introduction

The current trend in the telecommunications area is leading towards the deployment of so-called Next Generation Networks (NGNs). These are converged IP-based networks, which are planned to offer the future networking platform for different services and applications. NGNs are required to provide high transmission bandwidths and quality of service guarantees in combination with first-class resilience.

Because of falling profit margins and increasing competition, these networks must offer a very efficient operation. To meet this demand, an NGN should be kept in a good working point, which enables it to transport the highest amount of traffic with in-spec service quality. This requires the network parameters to continuously get adapted to changing traffic or failure situations.

When admitting high priority traffic with strict resilience requirements into such a network, possible link failures must be efficiently taken into account. Incorporating spare capacity into the network design to withstand link failures (e.g. by redundant links or overprovisioning) is a too inefficient approach.

To accomplish the given tasks, a node called the Network Control Server (NCS) has been developed and built[1]. This NCS collects current operational statistics from the network in which it is deployed. By using these data, the NCS can construct a network-wide view of the current network situation, e.g. its link loads, traffic matrix demands and more.

Additionally, the NCS is equipped with powerful optimization algorithms, which can adjust the amount of traffic admitted to enter the network and optimize the internal routing. Using these mechanisms, the NCS continuously keeps the network in a good working point and can therefore react to changes in the traffic matrix or in the network topology (e.g. after link failures).

Besides keeping the network in a well-balanced operating condition, the NCS can relieve operators of routine maintenance tasks and aid in the more complex tasks, e.g. in traffic engineering for changing traffic matrices or in evaluating network upgrade options.

---

[1] Parts of this work were funded by the German Ministry of Education and Research (BMBF) under contract 01AK045. The authors alone are responsible for the content of this paper.

In this paper, the basic architecture of the developed Next Generation Network and Network Control Server will be described. Its internal strategy will be presented and the benefits of the re-optimization process for network operation will be shown.

2 The KING Next Generation Network Architecture

The development of the Network Control Server presented in this paper has been done during the KING research project (Key components for the Internet of the Next Generation) described in [2, 10]. The aim of this project was to develop a comprehensive architecture for a Next Generation Network. In the following, the key paradigms and design decisions of this NGN architecture will be presented in a concise manner.

In particular, the KING network architecture aims to offer support for advanced quality of service guarantees in combination with first-class resilience while still allowing a very efficient NGN deployment and operation.

To meet these requirements, the network was designed to have a stateless core (see figure 1). By using this well-known approach, it is not necessary in case of failures to immediately restore state information in other components. Additionally, this concept allows for an increased scalability of the network, e.g. by avoiding per-flow state management using expensive router ressources.

Different QoS classes are supported by scheduling policies inside the routers. To offer QoS guarantees in an efficient way (without the need for excessive overprovisioning), access to higher priorized QoS classes is restricted by a Network Admission Control (NAC) located at the network borders.

Many of these design decisions clearly reflect the principles of the Differentiated Services (DiffServ) architecture [1]. One of the main goals of the DiffServ concept is to prevent core routers from having to cope with per-flow state information. Only the network border routers, which are also responsible for the admission control, shall process and store per-flow information.

Network functions are grouped into realtime and non-realtime depending on their time constraints.

2.1 Realtime functions

To offer the user a good network performance, realtime functions must be performed according to their timing requirements. Due to their urgent and important nature, they do not allow too much communication and processing overhead. E.g. it is generally not possible to perform complicated calculations or query a central instance for information. Especially the latter would add an intolerable delay just because of the communication round-trip times to the queried system.

The network architecture is designed to execute the most important realtime functions locally and as independently as possible:

- Packet forwarding — The forwarding of data packets is handled locally by the routers (usually by their line card hardware).

- QoS reservations — The handling of reservations for higher priorized QoS classes should be carried out in near-realtime to avoid a disappointing user experience. If it would be necessary to query a central bandwidth broker for every QoS reservation request, this would bring several disadvantages. If the central node fails, all admission control instances would malfunction. Under heavy load, scalability issues might arise. Our architecture allows a decentralized approach for the request handling part of the admission control. Each NAC instance at the network border is allowed to independently manage the traffic budget that was allocated to it. This greatly enhances the resilience of the admission control due to its independence of other components during the decision about QoS requests.

- Failure handling — This means the reaction to failures of links or network components by redirecting traffic via available alternative paths. Usually, network routers utilize routing protocols to find paths (and alternative paths) to possible destinations. There is no need to change this when optimizing routing or performing traffic engineering. In addition, bidirectional forwarding detection (BFD) [3] allows for a fast reaction to link failures, so that together

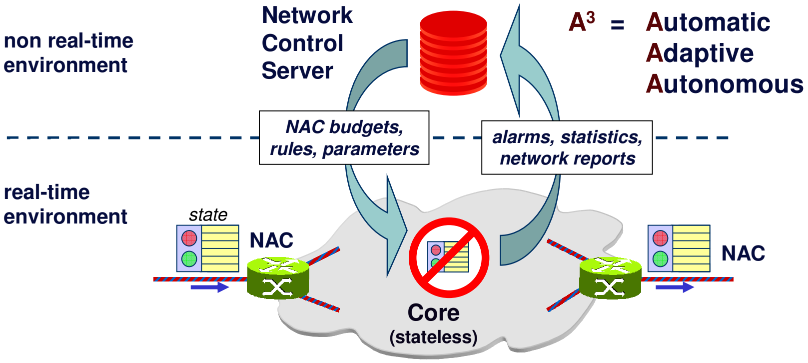

Figure 1: Deploying a Network Control Server in a Next Generation Network

with accelerated re-routing, even most real-time applications do not have to suffer from link failures if enough capacity is still available after re-routing.

2.2 Non-realtime functions

The counterpart of the tasks presented in the list above are the non-realtime functions. These comprise aspects of network operation which do not need to be carried out in realtime, but allow for more relaxed time constraints.

The main tasks that fall into this category are the adaptation of bandwidth budgets and routing re-optimization. NCS-based failure reactions only re-optimize the situation after a failure and are not required for the continuous operation of QoS services through failures.

It is possible to deploy a central management instance to perform these non-realtime functions. During the KING research project such a management node has been developed. Its architecture and operation will be presented in the following sections.

3 The Network Control Server

While the realtime functions described in section 2.1 can and must be performed autonomously by the affected network components, all optimization functions that benefit from a network-wide view of the current condition can take advantage of the deployment on a central management node.

We developed such a node that is responsible for the long-term network health and named it Network Control Server (NCS). It is one of the key components of the KING architecture that will help to steadily ensure a good performance of the NGN core network. To accomplish this tasks, the NCS is able to receive operational statistics from the network routers and traffic data from the Network Admission Control. If an intervention is necessary, the NCS can re-adjust different network parameters to keep it in a good condition. Figure 1 depicts this integration into a Next Generation Network architecture.

In general, the Network Control Server tries to maintain a sensible traffic balance, which is the prerequisite for the combination of resilience and QoS support. The NCS deals with all network issues which need a network-wide overview of the network state. Using the information from the NAC instances, the Network Control Server is able to calculate the current traffic that is offered traffic to the network. In particular, the NCS is responsible for the following tasks and offers a platform for the respective optimization algorithms:

- Routing Optimization — The NCS optimizes the network routing by optimizing the link cost metrics, which are subsequently used by the routing protocol to construct the routers' routing tables. Via this mechanism, the NCS adapts traffic distribution inside the network to the current network situation and traffic matrix.
- NAC budget assignment — While each Network Admission instance is independently responsible itself for managing its allocated traffic budget, the NCS assigns these budgets. Whenever necessary, it can redistribute them to suit the current network condition and QoS usage situation.
- Preventive preparation for failure cases — The optimization algorithms of the NCS take link failures into account (if the operator chooses to do so). The network routing and NAC budgets can already be prepared for possible outages before they actually occur, so no QoS guarantees will be violated. After a failure has occurred, the network can again be prepared to be resilient to further failures.

Of course, the network operator who deploys a Network Control Server can configure the reaction of the NCS and define his own notion of an acceptable network condition and when which reaction is necessary to re-optimize it.

Another important ability of the NCS is to relieve the network management of routine tasks by automatically handling as many of these issues as possible, or at least offer an option to aid the network operators in dealing with them. This helps further to an economical operation of the network.

The functionality of the NCS can be characterized as being:

- Adaptive — The Network Control Server is be able to adapt its behavior to changing network conditions, e.g. a varying traffic matrix or an altered network topology.
- Automatic — All actions the NCS judges to be necessary can be taken automatically, without the need for operator intervention (however, the operator might want to acknowledge any actions first).
- Autonomous — The NCS is built as an autonomous system, requiring no further manual assistance or input for performing its functions.

As the NCS is operating in the non-realtime environment, the functions that must be performed in realtime do not depend on the availability of the NCS. It does not introduce a new point of failure. Even if the NCS is offline for some time, the network can still continue to operate without being affected from an NCS outage. In such a case, only the network re-optimization will be delayed for some time; a fact that can be considered to be rather uncritical.

4 NCS Architecture and Modules

The internal architecture of the Network Control Server is derived from its design as a closed-loop system and is shown in figure 2. Every depicted box represents a functional building block of the complete system, being either a dedicated algorithm, a special tool or an important function. Starting from the upper part of the figure and descending to the lower part, the NCS consists of functional building blocks serving the following purposes:

1. Sensor components — used to gather operational information about the current network condition.
2. Strategy components — evaluate the current network situation and decide if an intervention of the NCS is necessary and what actions will be taken.
3. Optimization Algorithms — generate NAC budgets and link metrics that are adapted to the current network condition to enhance the network's efficiency.
4. Actuators — used to actually change the network behavior, i.e. by configuring the optimized parameters in the network components.

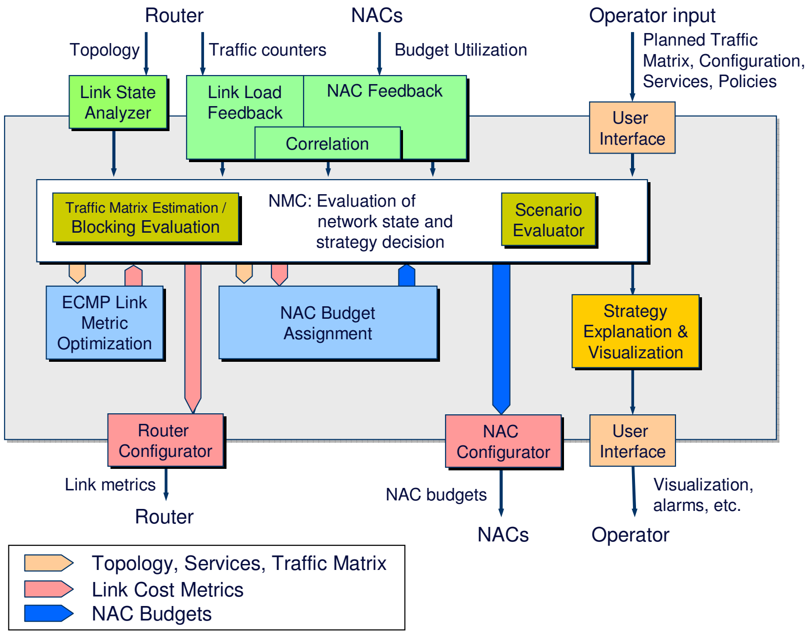

Figure 2: Internal architecture of the Network Control Server

4.1 Sensors

Mainly two building blocks provide the NCS with the necessary information it needs to build its own view of the network condition.:

- **Link State Analyzer** — The Link State Analyzer continuously listens to the routing protocol (e.g. OSPF/ECMP) messages that are exchanged between the routers in the network. Whenever one of the routers detects a change of the state of one of its attached links, it sends out corresponding Link State Announcements to its neighbor routers. These announcements spread throughout the whole network and enable each router to update its routing table.

  The Link State Analyzer (passively) takes part in this message exchange and can hereby receive all Link State Announcements. The included information enable the Link State Analyzer to build a valid view onto the current network topology (just like the routers do).

  This mechanism allows the NCS to react to changes in the network topology (e.g. link failures or reconnections).

- **Link Load and NAC Feedback Collection** — The NCS collects network statistics from two kinds of sources:

  - The *Routers* periodically report the amount of traffic that has been transmitted via each link per network service-class.
  - All *Network Admission Control* instances are queried for their budget and request status. Included in these data are, for example, the budget utilization, the amount of blocked and admitted QoS requests.

Besides collecting measurement data, the *Data Collection and Correlation component* (DCCC) of the NCS also correlates data between the different time windows they have been reported for, thus providing properly aligned data for further use in the NCS.

Based upon the statistics provided by the DCCC the other functional building blocks of the NCS can take advantage of the available information about the current network condition.

Apart from the autonomous NCS tools to gather feedback from the network and decide if an intervention is necessary, there is always a third reason that can trigger a reaction of the NCS:

- **Manual Operator Input** — Network operators can continuously monitor the data which are processed by the NCS and supervise its reactions if deemed necessary. Based upon the network statistics, it is always possible to manually initiate every desired reaction of the Network Control Server. All this can be done through a provided User Interface.

4.2 Strategy

One of the key aspects of the NCS is its strategy. After its sensor components gathered actual network statistics, the NCS can construct its notion of the network condition. Using its *Traffic Matrix Estimation* and *Blocking Evaluation* tools, the NCS can calculate the current matrix of traffic offered to the network and the expected probabilities of QoS requests getting blocked.

Comparing this network state against configured parameters (as given by the operator's policies) and operational boundaries (e.g., upper limits for blocking probabilities), the Network Control Server decide about the necessary strategy with the help of its *NCS Main Control* (NMC), which represents the inner core of the NCS.

Even if the NCS decides that a re-optimization is necessary and starts one (or both) of the optimization algorithms, it does not need to actually use the new parameters. The impact of newly calculated NAC budgets or link metrics can be evaluated by the *Scenario Evaluator*. For example, if the advantages of the new values are deemed to be below a configurable threshold, the NCS may save the network the hassle of a metric reconfiguration.

During its operation, the NCS continuously keeps its operator informed about the status of the network, the actions taken and their influence via its *Strategy Explanation and Visualization* component. The operator can inspect the optimization process by using the *User Interface* module, through which the available statistics can be presented.

Due to the importance of the internal reaction strategy of the NCS, it will be described in more detail in its own dedicated section 5.

4.3 Optimization Algorithms

The NCS supplies the platform for the following two important optimization tools:

- **NAC Budget Assignment** — Each high-priority data flow that wants to cross a KING network needs the permission of the Network Admission Control (NAC). For this task, each NAC box at the network border is supplied with a dedicated traffic budget. It manages this budget autonomously and freely assigns parts of it to incoming flow requests.

  The Network Control Server calculates these traffic budgets for all NAC boxes and configures them accordingly. By influencing the algorithm that generates the NAC budgets, the NCS can control the amount of high-priority traffic that flows through the KING network.

  At this point, there is a key trade-off between resilience and admissible traffic: While more high-priority traffic means more revenue, it also results in less resilience, since fewer link failures can be survived without losing the desired quality of service. This is why a lot of effort has been put into enhancing the budget calculation [5, 6].

  The NCS provides the NAC Budget Assignment algorithm with the current topology, traffic matrix, link metrics and the failure cases to be taken into account. Using this information, the Budget Assignment calculates new traffic budgets for each NAC instance, which are optimized for the current network condition.

- **Link Metric Optimization** — The NCS can increase network efficiency by varying OSPF interface costs (link metrics) for the Equal Cost Multipath (ECMP) routing. These metrics are used in each router by its activated ECMP mechanism to distribute outgoing traffic via every available path that has the same cost towards the requested destination. Adjusting these link metrics affects the paths on which traffic is routed through the network and can thus be used to control the link loads of the whole network.

  Therefore, the NCS incorporates a building block that is specially dedicated to the optimization of link metrics. To achieve good and fast results, a highly optimized genetic algorithm is used for this task. More details about the Metric Optimization can be found in [9].

  Based upon the current network topology and traffic matrix, the Metric Optimization produces adapted link metrics. If configured, it is able to take possible link failures into account during the optimization. By using this option, the network can be made more efficient in resilient operation.

4.4 Actuators

After the optimization algorithms deployed by the NCS have produced a new set of parameters that offer an advantage for network operation, the NCS can use its actuator modules to control certain network components:

- **NAC Configurator** — This tool is used to transmit newly optimized traffic budgets to the NAC instances.

  During the configuration with new budgets, it is a design principle of the KING Network Admission Control never to disrupt flows that have already been admitted. If a NAC instance is configured with a lower NAC budget than it has already assigned to active flows, it simply stops admitting new high-priority flows to the network. This is done until enough established flows have ended and enough budget capacity has been returned to the NAC and is then available for new flows. This behavior makes the implementation of NAC budgets a very powerful NCS tool, since it is non-disruptive to network operation. Whenever the NCS needs to limit the amount of high-priority traffic flowing through the network, it can recalculate and distribute new NAC budgets without affecting customers that already have a valid reservation. Because of these reasons, changing NAC budgets is always the Network Control Server's first choice of reaction towards changing network conditions.

- **Router Configurator** — This component is able to transfer calculated link metrics to the affected routers and configure them accordingly.

  While the variation of link metrics is an effective mechanism to balance traffic flows across the network, it is unfortunately not a non-destructive one. Each change of a single link cost results in network-wide distribution of link state information, usually causing a redirection of traffic via new paths with all corresponding consequences, e.g., a short period of time with higher latencies or even connectivity loss, in any case decreasing the quality of service during that time.

  Although special measures are taken to avoid fatal problems during link metric changes, the NCS still aims to avoid them to keep disturbances of the normal network operation limited and negative influences as low as possible. This is why the internal strategy of the NCS aims to only adjust link costs if that is necessary and if they cause a considerable improvement of the network's condition.

If the optimization algorithms and automatic actuators of the NCS have done their best effort for upholding the network health, there is always a third option how to react to persisting problems:

- **Manual Operator Intervention** — If the automatic tools inside the NCS do not suffice to keep the network in the desired condition, the Network Control Server will notify network operators about the status and the detected problem. The information gathered by the NCS is presented to them and, based upon these data, network operators can decide about further measures and instruct the NCS accordingly.

  Is is important to note that although the NCS was unable in this case to keep the network condition at the desired working point, this must not be seen as a failure of the concept. There might simply be too much (high priority) traffic being offered to the network. If this happens frequently, the operator might decide to expand or upgrade its network to match the growing traffic and can generate a higher income resulting from this investment.

5 NCS Operation

Combining the available tools, modules and optimization algorithms of the NCS into a useful and configurable strategy is the main task of the core NCS Main Control.

Besides the decision, if a and what reaction is appropriate, this building block is also responsible for starting and stopping the other internal parts of the Network Control Server and handing the right parameters and input data to them.

The default steering process that has been derived for the NCS prototype follows the strict policy to only change network parameters if really necessary and useful. Unnecessary actions, which could disturb active flows, should be avoided. This is why the impact of new parameters onto the network condition is evaluated before critical changes to the network configuration are made. Besides reducing the load put onto network components due to re-configuration actions, this precautious strategy also helps to avoid oscillations between different parameter sets in case of measurement fluctuations.

In the following, the basic default strategy, which has been derived for the NCS prototype will be presented. At first, the periodic optimization process will be described, followed by the handling of link failures.

5.1 Normal Operation

The Network Control Server is designed to continuously monitor the network condition and take appropriate actions as soon as an undesired state is detected. For this purpose, a strategy has been designed that connects the different tools, functions and building blocks of the NCS in the necessary order. The sequence of this steering process is depicted in figure 3 and will be described in the following:

1. The NMC will *periodically* call an algorithm called *Traffic Matrix Estimation* (TME). The task of the TME is to fetch data about the current NAC budget statistics from the Feedback component and calculate a so-called *Active Traffic Matrix*. This process takes the amount of reserved budgets and blocked traffic requests into account.

   The aim of this step is to produce a traffic matrix that does not only reflect the currently admitted traffic, but also the high-priority traffic that *could have been* admitted, if the budgets had been high enough. This amount of offered traffic that has been blocked, directly translates into possible revenue that could not be generated due to budget restrictions.

   The calculation of the active traffic matrix is also supposed to help the NCS to react as quickly as possible to changing traffic behavior, even before too many requests get actually blocked. Figure 4 depicts the benefits of this approach during the reaction to a sudden rise in the offered traffic. If we had chosen to trigger a reaction only by the measured blocking rate, QoS requests would already get rejected before a re-optimization would get started (see the second last diagram). However, as we estimate the blocking rate based on the estimated traffic matrix, it is possible to react far earlier (as shown in the lower diagram). More details about this can be found in later in the evaluation section 6.2 of this paper.

2. The *Blocking Evaluation* function uses the generated traffic matrix of the TME and the *currently configured* NAC budgets to estimate the current blocking of high-priority flow requests. This can be either the maximum blocking probability, the sum of blocked traffic or the ratio of blocked to offered traffic.

   The calculated blocking is compared to a configured value that represents the operator's policy of how much blocking is acceptable. While a blocking rate of zero would be desireable, this is generally not possible to achieve with finite capacity. This is why we introduced the notion of a working point being *good enough*, which does not justify any interference with the current network operation. Therefore, if the blocking is lower than the configured

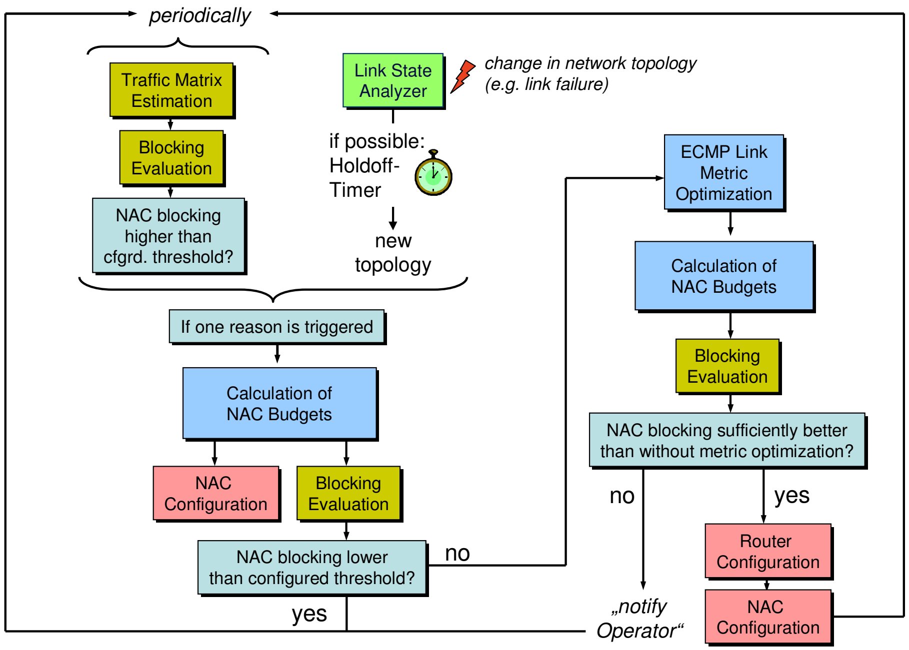

Figure 3: Operation default strategy of the Network Control Server

threshold, no action is necessary and the NMC waits until a new traffic matrix estimate is available for the next evaluation period.

If the blocking is higher than acceptable, the NMC continues with step 3.

3. Since the current blocking is deemed unacceptably high, the NMC initiates the calculation of new NAC budgets based upon the new traffic matrix.

   After the calculation has been finished, the generated budgets are transfered to the NAC instances at the network border by the *NAC Configurator* module. This is done unconditionally, since the new NAC budgets are currently the best available reaction to the current network situation and the setting of new NAC budgets is a non-destructive process (more on this has been explained in subsection 4.4).

   However, the NMC will evaluate if the adapted NAC budgets are already a sufficient reaction by re-calculating the resulting blocking. If the blocking is now lower than the acceptable maximum value, the calculation of new NAC budgets alone will successfully adapt the network to the new traffic condition and no further action is necessary.

   If the new NAC budgets still result in a blocking rate being too high, the NMC escalates the problem and proceeds to the next step.

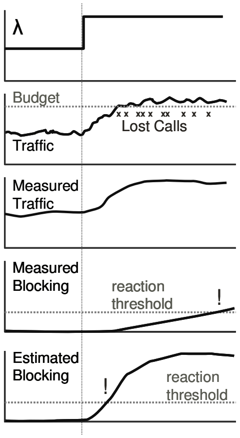

Figure 4: Measured vs. estimated traffic matrix and blocking rate

4. At this point, it is obvious that newly optimized NAC budgets alone could not sufficiently reduce the current blocking rate of high-priority flow requests at the network border.

   In this case, the NMC starts the *Metric Optimization* algorithm, whose purpose is to optimize the network link metrics that are responsible for routing and ECMP-based traffic distribution. As already mentioned in subsection 4.4, the setting of new link metrics should only be done if really necessary, since it can disturb network operation. Because of this reason, the NMC does only initiate a metric optimization if other measures (new NAC budgets) have not been sufficient.

   On the other hand, it is quite possible that optimized link cost metrics could re-balance the network and bring it back to an acceptable operating point. This is the result of simulations that have been done to evaluate this multi-stage strategy. Some details about the findings are presented in the evaluation section 6.1 below.

5. Based on the newly optimized link metrics, the NAC Budget Calculation is started again and takes the resulting new traffic distribution into account.

6. With the help of the Blocking Evaluation, the blocking probabilities with the new link metrics and NAC budgets are evaluated again.

   - If the blocking is now below the acceptable threshold, the new link metrics are transferred to the routers using the *Router Configurator*. This will implement the new routing and traffic distribution throughout the network.

     After the new metrics have been configured, the new NAC budgets are transmitted to the NAC instances to complete the process.

   - If the blocking is still higher than the configured threshold, but is significantly better (where ”‘significantly”’ is configurable) than it has been before link metric optimization, the metrics and budgets are nevertheless transferred to the routers, resp. the NAC boxes. In this case, the operator is additionally notified about the fact that the autonomous reactions of the NCS were not sufficient to reduce the blocking to an acceptable value and that other measures are necessary (e.g., increasing the capacity of certain links).

   - If the blocking is still higher than the configured threshold and is not much better than without the new link metrics, the operator is notified about this problem (as described above). In this case, the NMC spares the network the configuration effort of setting the new link metrics, since the result is not deemed to be worth the possible disruption of the quality of service.

   In any case, the NMC will return to sleep until the steering cycle is called again.

5.2 Reaction to Link State Changes

Whenever a link state change occurs (either a link failure or a link reconnection), the NMC is notified about this fact by the Link State Analyzer, which collects topology information updates for the NCS.

The main idea behind the reaction to a link state change is to adapt the Network Admission Control budgets to the new situation. If the change was a link reconnection, this strategy will quickly allow to permit new high-priority flow requests and generate the corresponding revenue. On the other hand, if the trigger was a link outage, the NAC budgets should be reduced to prepare for the next failure without unnecessary delay.

However, as a key paradigm of KING already is resilience, the guaranteed quality of service is not disturbed by a single failure. This concept is integrated into the NAC Budget Assignment and ECMP Link Metric Optimization, which can both take link failures into account. Therefore, the parameters that are active in a KING network (link metrics and NAC budgets) will already be adapted to anticipated failure conditions in advance, so both ensure the desired quality of service even in case an outage actually occurs.

Hence, it is not really critical for a link state change to directly trigger an NCS reaction. As statistics show, there are many short-term link outages, which would all trigger budget changes if the NCS reacted immediately. Instead, the NMC offers a configurable period of waiting time, a *Holdoff-Timer*, during which it hides the new network topology from the budget calculation algorithm, as depicted in figure 3.

If the Holdoff-Timer expires without the failed link(s) getting reconnected, the new network topology will be handed to the calculation algorithms (starting with the NAC Budget Assignment).

As already mentioned, the implementation of new NAC budgets at the network border is an uncritical process, since active flows are not interrupted. At the same time, it takes some time for reduced NAC budgets (in case of a link outage) to actually develop an impact, since only new requests can be blocked and currently active flows must finish before the amount of admitted high-priority traffic decreases.

Together with the default behavior to prepare the network for possible failures in advance, this enables the network operator to set the Holdoff-Timer to a desired value. Its configuration decides how fast the NCS will re-adapt the network parameters to current outages to protect them against the next possible failure. If this is done quickly, following outages will have less negative impact. On the other hand, it is very well possible to wait a little longer for broken links to come back online, as it might not even be necessary to react at all if the problem can be solved fast enough.

6 Evaluation

The efficiency and benefits of the reaction strategy of the NCS have been demonstrated in several simulations. In the following, some simulation results will be presented that show different aspects of advantages that can be gained from an NCS deployment for network operation.

6.1 Reduction of the blocking rate after Traffic Matrix changes

A static simulation set-up has been used to validate the reaction strategy of the NCS towards changes in the observed traffic as described in section 5.

The simulated network was assumed to represent a backbone network across the U.S. with points of presence (nodes) in 20 cities. Its topology is depicted in figure 5.

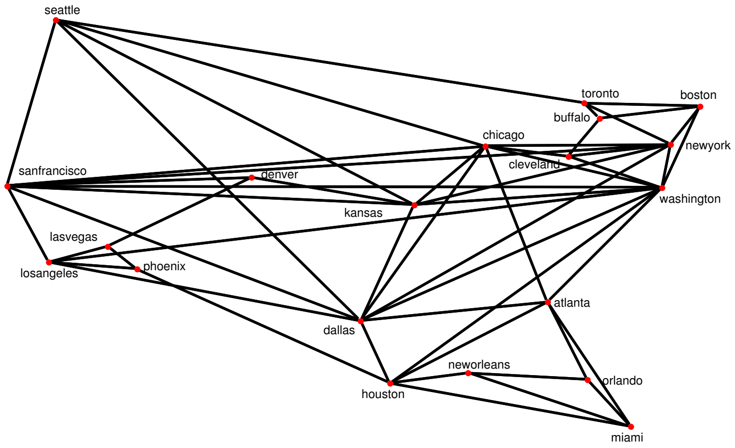

Figure 5: Topology of the simulated network

Starting from an initial homogeneous traffic matrix for which the network is optimized, a number of traffic matrix changes is simulated and the blocked traffic ratio is evaluated in the following three steps:

- *Initial blocking*: The blocked traffic ratio after a change in the traffic matrix but before any reaction by the NCS, i.e. with budget settings and link metrics optimized for the previous traffic matrix.
- *New budgets*: The blocked traffic ratio after re-computation of budgets but with the old link cost metrics (default reaction).
- *New metrics*: The blocked traffic ratio after re-optimizing link cost metrics and subsequently re-computing NAC budgets (escalated reaction).

The following list explains this investigation in a slightly more formal way. The traffic matrices used are shortly described in Tab. 1.

- Create homogeneous traffic matrix TM_1
- Optimize routing for TM_1 using the metric optimization algorithm.
- Compute NAC budgets for TM_1 using the budget calculation algorithm.
- Evaluate blocking for TM_1 using the Blocking Evaluation.
- Loop over $i \in 2, 3, \ldots, 15$
  - Create new traffic matrix TM_i
  - Evaluate blocking for TM_i. These are the *initial blocking* entries in Fig. 6.
  - Compute new budgets for TM_i without changing the link metrics
  - Evaluate blocking. These are the *new budgets* entries in Fig. 6.
  - Optimize routing (link metrics) for TM_i
  - Compute new budgets for TM_i using the new link metrics
  - Evaluate blocking. These are the *new metrics* entries in Fig. 6.

Fig. 6 gives the resulting blocked traffic ratios for two versions of the simulated network, one with only 10 nodes working as ingress and egress nodes for traffic and one with all 20 nodes working as ingress and egress nodes for traffic.

The bars in Fig. 6 show that both the re-calculation of budgets and the re-optimization of link cost metrics each significantly reduce the blocked traffic ratio. Hence, the approach of first trying to re-distribute the NAC budgets to reduce blocking and then optionally re-optimizing the routing is successful in the scenarios shown here.

It has to be noted, however, that the viability and success of this approach depend on a few parameters: If the offered traffic is very low, there will also be very little blocked traffic even after a change in the traffic matrix and an NCS reaction will not be required. If, on the other hand, more traffic is offered to the network than the network can cope with, none of the algorithms will be capable of reducing the blocking

Table 1: Traffic matrices used for the investigation.

i	Description of TM_i
1	Initial homogeneous traffic matrix: The same amount of traffic is offered at all border-to-border relations.
2	Traffic matrix with random entries uniformly distributed between 0 and twice the mean value.
3	Like TM_2, but with symmetric entries, i.e. the traffic from node j to node k is equal to the traffic from node k to node j.
4	A heuristic traffic matrix following the argument that nodes that have a large number of links should send and receive a large amount of traffic[a].
5	The *Zipf* heuristic. Like TM_4, this traffic matrix allocates more traffic to nodes with a high number of links, but instead of using random numbers, the initial factors are chosen as $t[m] = 1/m, m = 1, 2, \ldots, N_{nodes}$.
6–15	Further random traffic matrices similar to TM_2.

---

[a] The principle in more detail: Generate random numbers $t[m]$ on [0.1,1.0] and sort them. Associate the largest value with the node that has the largest number of links attached, the second largest with the second largest node, etc. Each traffic matrix entry (j, k) is the product t[j]*t[k]. Rescale afterwards to reach the target value of mean offered traffic.

probability to the desired value. Of course, the action of blocking excess traffic is exactly what an admission control is used for.

In addition, the topology of the network must allow for a change in routing for the metric optimization step to be of any value. However, this seems to be a valid assumption for a carefully engineered backbone network offering a good level of resilience.

6.2 Traffic Matrix Estimation for quick reaction

The NCS Traffic Matrix Estimation (TME) module is responsible for providing estimates of a current traffic matrix to several modules within the NCS. This section documents the data flow leading to decisions in the NCS scenario evaluation strategy and motivates some design choices taken.

6.2.1 NCS Data Flow for Budget Adaptation

Two central algorithms within the NCS, metric optimization and budget assignment, require a demand matrix as input. This demand matrix has to reflect current traffic patterns at all times. This task is handled by the Traffic Matrix Estimation (TME) module within the NCS, which obtains correlated measurement data from the Data Collection and Correlation Component (DCCC).

An overview of the data flow between the relevant modules of the NCS is shown in Fig. 7. Along with the current topology from the link state analyzer, a traffic matrix is used as input by the metric optimization as well as the budget assignment components. In order to realize a reaction to current traffic patterns with these components, they require a current estimate of demand, as provided by the "active traffic matrix" computed by the TME.

The ultimate target of the metric optimization and budget assignment algorithms is to minimize the amount of traffic blocked by admission control. Therefore, the NCS' decision whether to exchange currently used budgets for a new set of budgets is based on a comparison of current blocking probabilities to the amount of acceptable blocking configured by the network operator as described in section 5.1.

6.2.2 Simulation Model

The measurement algorithms were tested on simulated flow level traffic for a single border-to-border budget. Currently, the NCS operates on the *reserved* bit rates only and does not regard the actually transported packets. Correspondingly, it is sufficient to simulate the utilization of budget reservations without simulating traffic on the links.

Fig. 8 depicts our flow level simulation model along with the relevant parameters and measurement points. An infinite number of sources described by the distribution of request inter-arrival time T_A, holding time T_H and reserved bit rate R reserve capacity at an admission control. The inter-arrival and holding times are described by negative exponential distributions with mean values t_A and t_H, respectively. At measurement point ①, the admitted bit rate R_a as well as the amount of blocked reservations b_m can be observed. The admission control component is parameterized by the budget B and the statistics sampling interval T_N. Mean values for the admitted traffic as well as the number of blocked reservations and the sum of the bit rates of all blocked requests are delivered to the data collection component periodically with a period of T_N (measurement point ②). The traffic matrix estimation component periodically requests admitted and blocked traffic values from the data collection component (measurement point ③) for the previous interval of duration T_W at offset T_O, i.e. at point in time $t_k = k \cdot T_E$ mean values are requested for the interval $[t_k - T_O - T_W, t_k - T_O]$. Using the exponentially weighted moving average (EWMA) smoothing parameter α and the blocked traffic weighting parameter β, the traffic estimation component produces samples of estimated traffic demand R_e every T_E (measurement point ④). The blocking calculation component uses R_e together with the configured budget B and a given traffic mix to compute the estimated ratio of blocked traffic b_e (which is available at measurement point ⑤) using the Kaufman-Roberts formula [7]. A decision unit compares b_e to a threshold value b_{thr} and switches the configured budget from B_{old} to B_{new} once b_e is greater or equal to b_{thr} for the first time. Note that, as only a single budget is considered, no real budget re-allocation can take place in the simulation, but it is modelled by the two values B_{old} and B_{new}. The currently configured budget B is available at measurement point ⑥ for trace visualization purposes.

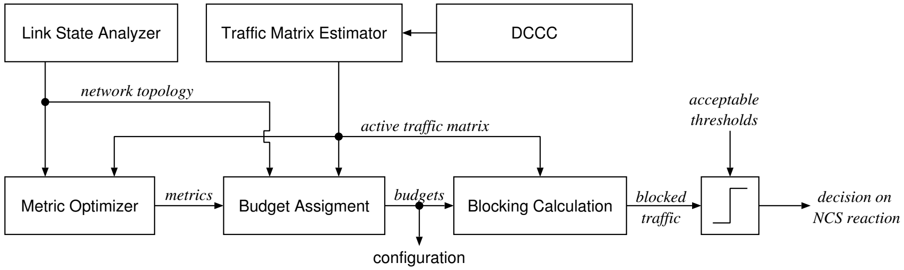

Figure 7: Data flow within the NCS for Budget Adaptation.

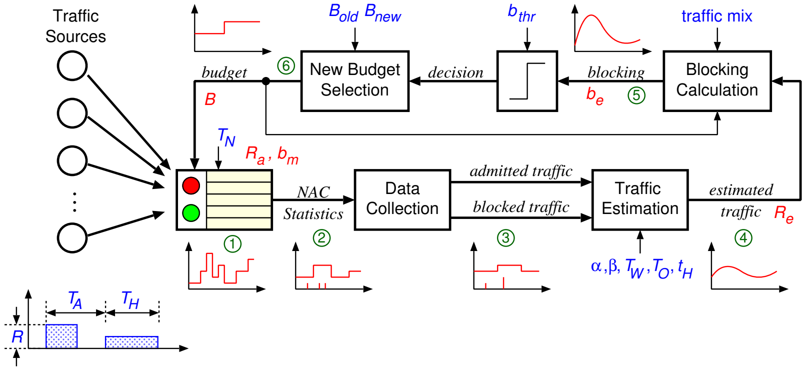

Figure 8: Simulation model with parameters and measurement points.

6.2.3 Basic Simulation Scenarios

Three basic simulation scenarios have been chosen to evaluate the traffic matrix estimation module. They have been parameterized such that they can also be realized in the prototype lab network where there are certain restrictions on the complexity and dynamics of traffic models:

- No more than 100 reservations can be active at the same ingress node at the same time.
- There must be a minimum gap time of two seconds between two requests at the same ingress node.

Given a default budget of 1.28 Mbit/s and a bit rate per reservation of 64kbit/s, three scenarios with blocking probabilities of around 10 %, 1 % and 0.1 % were created with the parameters as given in Tab. 2. Interarrival times and holding times are exponentially distributed with a lower truncation bound of 2 s.

The NCS does not only measure traffic in the network, but it also reacts on changes in traffic patterns. Such a change is often caused by a sudden increase in demand on a border-to-border relation. In the simulation, this is modeled by changing the interarrival distribution at a certain point of time.

Table 2: Parameters of three simulation scenarios with different blocking probabilities

Scenario	A	B	C
Target blocking	0.001	0.01	0.1
Interarrival time	20 s	20 s	20 s
Bit rate per source	64 kbit/s	64 kbit/s	64 kbit/s
Mean total bit rate	0.603 Mbit/s	0.770 Mbit/s	1.128 Mbit/s
Mean number of active sources	9.41	12.03	17.63
Mean holding time per source	188.3 s	240.7 s	352.6 s

6.2.4 Measurement versus Computation of Blocking

This section discusses whether to measure blocking rates directly in the NACs or to compute blocking probabilities from budgets and estimated traffic. There are some reasons for computing blocking probabilities from estimated demand instead of measuring the blocking probability, although one might expect the latter solution to be more precise or at least faster.

- It takes very long to measure rare events at a sufficient level of precision whereas very low blocking probabilities can be easily computed analytically using the multi-rate Erlang formula [4, 8] – provided the traffic mix description and estimated demand are sufficiently precise.

- For the reasons outlined in Sec. 6.2.1 the main measure of "network health" is the amount of blocked traffic due to the AC budgets. A reaction of the NCS, i.e. an increase in budgets for a border-to-border relation with high blocking, should be visible in the quantity that is used to evaluate this "network health". Measured blocking values will only reflect the effect of budget adaptations after a certain amount of time which depends on the measurement periods T_N and T_E as well as any smoothing methods applied to the data before they are compared against the threshold b_{thr}. A blocking value b_e estimated as described in Fig. 8 on the other hand is able to immediately quantify the effect that a certain budget change will have on the blocking probability.

- Measuring blocked traffic directly would not save the NCS from estimating current traffic demand, as this is the basis for all budget calculation and metric optimization operations.

The investigation of measured blocking b_m and computed blocking b_e in Fig. 9 indicates that the blocking values computed from the traffic estimate are faster in indicating an overload situation than the measured blocking. Note that measured blocking b_m also needs to include some smoothing or time averaging in order to deliver a quantitative value.

The non-linear (multi-rate) Erlang blocking function transforms estimated offered traffic rates R_e into blocking estimates b_e, allowing a good warning indicator for likely blocking. Note that in scenario B as simulated here, a mean blocking of 1 % is expected. However, due to the statistical nature of the Poisson sources, the estimated blocking naturally varies around this value, but using blocking estimates instead of measured blocking still allows for a more timely detection of high load cases.

In the scenario depicted in Fig. 10 it can be seen how the estimation of blocking from a traffic estimate and a budget indeed leads to an immediate change in estimated blocking b_e whereas in any kind of smoothed blocking measurement b_m, it takes some time until the effect of an increased (or reduced) budget can be seen. In addition, this measure allows for *hypothetical* changes of budgets that can be evaluated in terms of the corresponding expected blocking without actually being activated in the network. This would not be possible if measured blocking probabilities were used to judge the operational state of the network.

6.3 Implementation

The presented architecture of the Network Control Server and the underlying Next Generation Network have also been implemented as a prototype in an extensive lab environment.

Several scenarios have been developed for this lab network to demonstrate the described benefits of the Next Generation Network Architecture for the provided quality of service. During presentations, links were disconnected to demonstrate the influence of link failures onto the service quality.

The NCS prototype was deployed to re-optimize bandwidth budgets and routing metrics after changes in the traffic matrix or network topology. In different demonstration scenarios, the Network Control Server fully met the expectations on its benefits for an efficient network operation.

7 Outlook

Already providing a platform for network optimization, equipped with information collection modules, optimization algorithms, strategy, visualization and control functions, the Network Control Server offers a tremendous opportunity for extensions.

Despite the proven usefulness of the basic default strategy explained in this paper, it seems desireable to offer a more flexible configuration. The operator should be enabled to conveniently chose and configure its desired policy, when and how exactly the NCS should react. The development of such a flexible strategy configuration is currently a topic of further work. For example, it would be possible to provide a graphical user interface, enabling an operator to define its desired network management strategy via intuitive drag and drop actions.

Although, in its current version, the NCS does only intervene when the network drifts out of its optimal working state, this could easily be enhanced in future version, e.g. by collecting a history of traffic matrices. Using this knowledge and interpolating the traffic demand, the NCS would be able to react in anticipation of the regular happening traffic changes and proactively adapt the network accordingly.

Because of its knowledge about the network topology and traffic demands, the NCS can help its operator to find the best opportunities for network upgrades. Using its optimization and evaluation tools, the NCS can automatically search for most effective topology changes and present its findings to the operator for further inspection.

Among other ideas for the usage of the Network Control Server's network-wide statistics is the aim to enhance network security. For example, the NCS could scan its data for traffic anomalies, as they usually occur during denial of service attacks, and alert operators or counter threats by configuring border routers accordingly.

8 Conclusion

In this paper an architecture has been presented for a Network Control Server (NCS) to be deployed in a Next Generation Network. Following its internal strategy, the NCS combines network information gathering modules, powerful optimization algorithms and network control tools to keep the network in a desired working point. The benefits of the approach have been verified in simulations and even demonstrated in a working lab prototype.

By continuously monitoring the network and adapting its parameters to changing traffic demands if necessary, the NCS helps to keep the blocking rate of QoS request low, to efficiently use the available network resources and provides an very useful tool for operators during network deployment and optimization. The NCS is especially aimed to meet the rising demands for an economically efficient operation of modern Next Generation Networks.

Acknowledgements

Siemens Information and Communication Networks (ICN) has been running the KING research project since October 2001, together with a number of leading German universities and research institutes. The project has been partially funded by the German Ministry for Education and Research (BMBF) for a period of over 3 years (01AK045).

The presented work is part of the solutions developed by the Work Package 1.1 team and, as such, is highly coupled into the overall research performed during the KING project. Therefore, acknowledgements go to the other project partners and especially the members of the WP 1.1 development team.

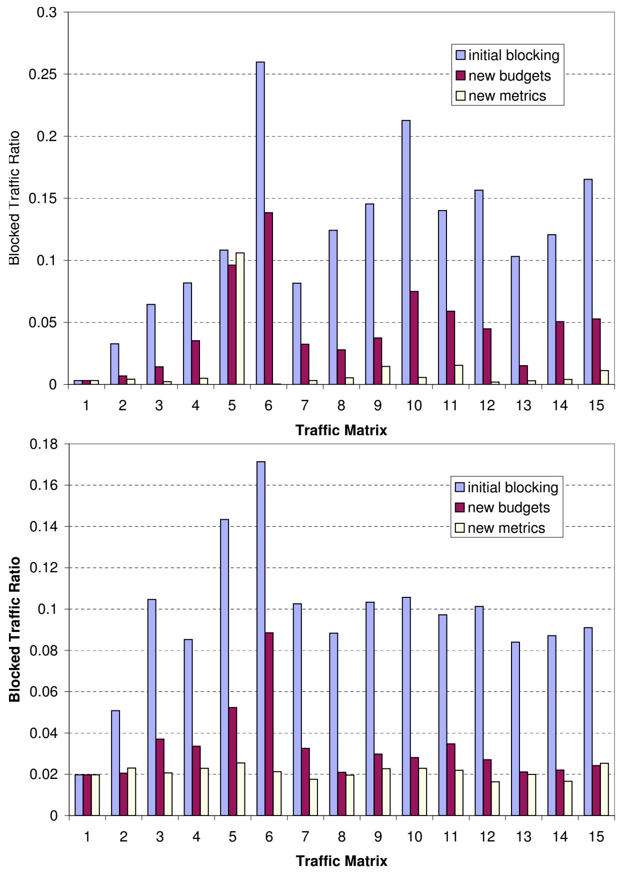

Figure 6: Blocking probabilities before budget re-allocation, after budget re-allocation and after re-optimization of link cost metrics. Above: 10 ingress and egress nodes; Below: 20 ingress and egress nodes.

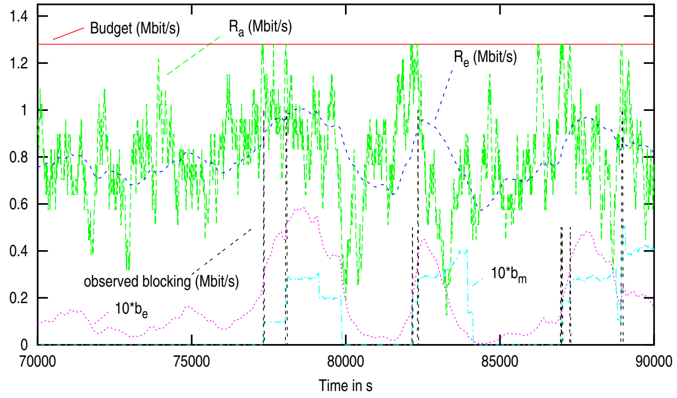

Figure 9: Simulation trace for scenario B, comparing estimated blocking b_e (measurement point (5) in Fig. 8) and measured blocking b_m (measurement point (1), smoothed).

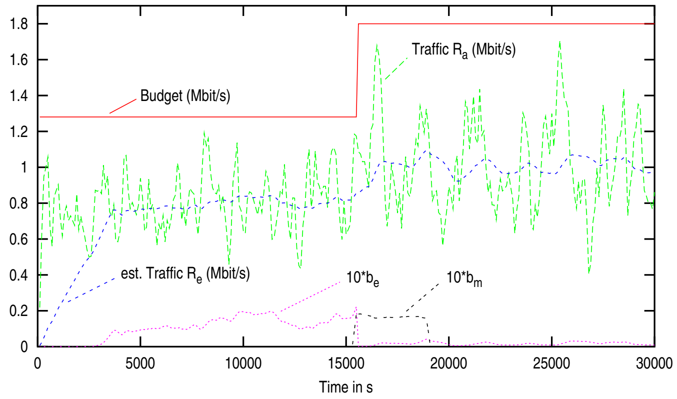

Figure 10: Simulation trace for scenario B, comparing estimated blocking b_e (measurement point (5) in Fig. 8) and measured blocking b_m (measurement point (1), smoothed) under the effects of a slight traffic change at $t = 13000$ s and a resulting budget change at $t = 15600$ s.

References

[1] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services. IETF, RFC 2475, December 1998.

[2] C. Hoogendoorn, J. Charzinski, K. Schrodi, N. Heldt, M. Huber, C. Winkler, and J. Riedl. Towards the Next Generation Network. In *12th IEEE International Conference on Network Protocols (ICNP 2004), Berlin, Germany*, October 2004.

[3] D. Katz and D. Ward. Bidirectional Forwarding Detection. IETF draft: draft-ietf-bfd-base-03.txt, July 2005.

[4] J.-F.P. Labourdette and G.W. Hart. Blocking Probabilities in Multitraffic Loss Systems: Insensitivity, Asymptotic Behaviour and Approximations. *IEEE Transactions on Communications*, 40(8), August 1992.

[5] Michael Menth. *Efficient Admission Control and Routing in Resilient Communication Networks*. PhD thesis, University of Würzburg, Faculty of Computer Science, Am Hubland, July 2004.

[6] Michael Menth, Stefan Kopf, and Joachim Charzinski. Network Admission Control for Fault-Tolerant QoS Provisioning. In *7th IEEE International Conference on High Speed Networks and Multimedia Communications (HSNMC)*, pages 1 – 13, Toulouse, France, June 2004.

[7] Ugo Mocci, Jorma Virtamo, and James Roberts. *Broadband Network Traffic: Performance Evaluation and Design of Broadband Multiservice Networks: Final Report of Action Cost 242*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1999.

[8] A. A. Nilson, M. Perry, A. Gersht, and V.B. Iversen. On multi-rate Erlang-B Computations. In *Proceedings of 16th International Teletraffic Congress (ITC 16), Edinburgh, Scotland*, June 1999.

[9] C. Reichert and T. Magedanz. A Fast Heuristic for Genetic Algorithms in Link Weight Optimization. In *5th International Workshop on Quality of Future Internet Services (QoFIS), Barcelona, Spain*, September 2004.

[10] K. Schrodi. High Speed Networks for Carriers. In *7th IFIP/IEEE International Workshop, Protocols for High Speed Networks (PfHSN 2002), Berlin, Germany*, April 2002.

Biographical sketches

Uwe Walter began his Computer Science study at the University of Karlsruhe in 1996. After his intermediate diploma, he concentrated on various topics in the telecommunications area and received his diploma degree in 2001. Since the beginning of 2002, he works as a research assistant at the Institute of Telematics at the University of Karlsruhe in the research group of Prof. Dr. Martina Zitterbart. His main areas of research include quality of service (QoS), routing and network security topics. Recently, he was deployed in a research project about packet-switched networks of the next generation, dealing with topics like network admission control, optimization and management and the impact of all these on the provided quality of service.

Martina Zitterbart is full professor in computer science at the University of Karlsruhe, Germany. She received her doctoral degree from the University of Karlsruhe in 1990. She was a Visiting Scientist at the IBM T.J. Watson Research Center from 1991–1992. She was full professor at the Technical University of Braunschweig from 1995–2001. Her primary research interests are in the areas of high performance networking, mobile communication, next generation networking, ambient technologies and eLearning. She is member of IEEE, ACM and the German GI.

Joachim Charzinski received his Dipl.-Ing. and Dr.-Ing. degrees from University of Stuttgart, Germany, in 1991 and 1999, respectively. From 1992 until 1997 he was with the Institute of Communication Networks and Computer Engineering at University of Stuttgart, Germany, working on error detection capabilities of the CAN protocol, on media access control in ATM access networks and on other teletraffic topics. He also assisted in the lectures on teletraffic theory and on coding theory. 1994 through 1996 he gave lectures on communication technology at the University of Cooperative Education in Stuttgart (Berufsakademie Stuttgart) and gave courses on OSI and Internet principles for telecommunication companies. In 1997 he joined Siemens AG, Munich, Germany, where he was a project leader for Internet traffic engineering until 2001 in the Information and Communication Networks division. Since then he has been a consultant for Quality of Service and security innovations in IP networks.