Optimization and Early-Warning in DSL Access Networks based on Simulation*

Rainer Bye, Joël Chinnow, Jan Clausen
Karsten Bsufka, Sahin Albayrak
DAI-Labor, Technische Universität Berlin
{rainer.bye, joel.chinnow, jan.clausen, karsten.bsufka, sahin.albayrak}@dai-labor.de

Categories and Subject Descriptors

I.6.5 [**Simulation and modeling**]: Model Development; I.6.3 [**Simulation and modeling**]: Applications; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*

General Terms

Experimentation, Management, Verification

ABSTRACT

Network providers operate large DSL-based access networks to offer customers Broadband Internet. These networks are observed and managed by Performance Management Systems (PMS), that capture the actual situation to support network administration. In this regard, the administrator can cope with incidents such as link failures or congestion. We present an application for optimization and forecast of traffic distributions in DSL networks as an addition to an existing PMS. This application makes heavy use of simulation. In this way, we give a description of traffic models based on real network performance data reflecting: (I) individual subscribers and (II) an aggregated model for multiple subscribers. Then, we introduce the overall simulation approach based on the Network Security Simulator *NeSSi*^{2}. The evaluation takes place by a use case for simulation-based verification of applied optimization strategies and a use case for continuous forecast to predict upcoming link congestion.

1. INTRODUCTION

In contemporary times, triple play, i.e. TV, Voice and Internet, is provided to customers via a single network connection. The current VDSL (Very High Speed Digital Subscriber Line) technique provides enough bandwidth to satisfy the corresponding demands of the customers. However, the convergence of services to IP has a huge impact on user traffic characteristics and in this regard, on the underlying access network infrastructure. The network providers use *Performance Management Systems* (PMS) as passive decision support for the management of their networks. These systems gather the state of the network by means of sensors located in the infrastructure, that measure relevant performance data such as link throughput etc. The PMS enables the administrator to handle incidents, e.g. link failures or congestion, and to optimize a network configuration accordingly.

Most traffic optimization approaches focus on the IP layer, but these approaches ignore the DSL access area between the customer and its entry point to the Internet. To ensure a desired quality of service, traffic optimization needs already be applied on the link layer. The current challenges are in particular, that the traffic path assignments in the DSL access network are static. In particular, the frequent changes of the network infrastructure, caused by the ongoing upgrade to VDSL, complicate this matter of fact. In this regard, one challenge is to update and optimize traffic path assignments. Furthermore, upcoming congestion in the access network are hard to predict due to the new traffic characteristics.

In an ongoing research work, we have developed a *Decision Support Framework* (DSF) for a network provider capable of solving these problems. This framework is realized by means of a multiagent system and *NeSSi*^{2}, the Network Security Simulator[1]. The system processes network inventory as well as performance data such as link loads. Based on optimization and simulation, warnings related to the operation of networks and recommendations for future network planning are created.

1.1 Contributions

In this paper, we present the overall DSF, but focus on the realized simulation aspects. The agent-based concepts of this work are detailed in [1]. In this regard, the first contribution of this work is the provision of two realistic traffic models reflecting a number of DSL customers based on real network performance data. The second contribution is an extension to *NeSSi*^{2}, enabling the integration of new domain models and simulation logic, in particular the developed system for DSL access networks. As third contribution, we apply two case studies that show the value of simulation for two challenges in the operation of a network: (I) the verification of network optimization results and (II) the realization of a workflow for continuous forecast about the upcoming situation of a network.

*Research supported and funded by Deutsche Telekom AG.

SIMUTools March 15 - 19, 2010, Torremolinos, Malaga, Spain
Copyright 2010 ICST, ISBN 78-963-9799-87-5.

[1] Publicly available at `www.nessi2.de`

1.2 Organization of the paper

In Section 2, we give an overview of the domain of DSL Access Networks, available input data for this research and necessary terminology to follow the rest of the paper. Afterwards, we introduce in Section 3 the realized framework for performance optimization in DSL access networks and provide the rationale for simulation. The realized simulation models reflecting upstream and downstream bandwidths dependent on the number of customers are explained in detail in Section 4. Accordingly, we briefly introduce *NeSSi*^{2}, the integration of the developed simulation models in *NeSSi*^{2} and their application for forecast and verification of optimization in Section 5 and conclude in Section 6.

2. OPERATION OF DSL ACCESS NETWORKS

In the following, we refer to Figure 1 by way of illustration. Letters denote functional partitions of the network, whereas the numbers refer to types of devices.

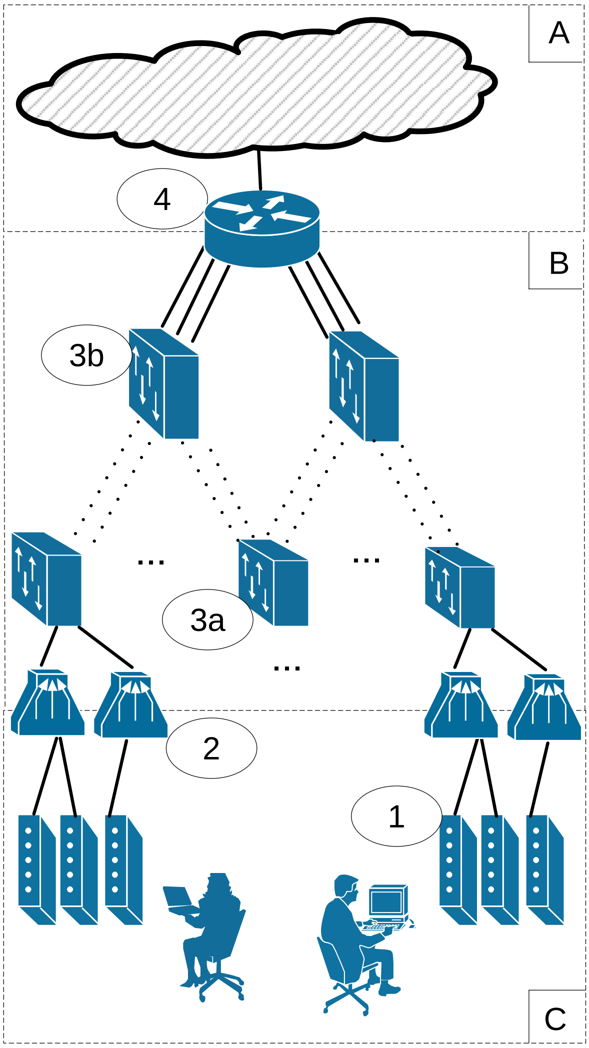

Figure 1: Schematic of DSL-Access Networks

In DSL access networks, customers are connected via their modem (1) to a DSLAM, the *Digital Subscriber Line Access Multiplexer*. These DSLAMs are usually located close to or sometimes even inside customer homes, terminate the local loop (area C in figure) and represent the crossover to a telecom provider network. In this way, the DSLAMs denote the entrance to the concentrator network (area B in figure), a layered hierarchy of switches (3a, 3b), either ATM or Ethernet-based, where the BRAS, the *Broadband Remote Access Server* is at the top (4) . The BRAS represent the entrance point to the Internet respectively the core network of the telecommunication provider (area A in figure). In addition, the BRAS terminates the PPPoA, *Point-to-Point over ATM* or PPPoE *Point-to-Point over Ethernet* connection, that is established between the BRAS and a customer DSL modem and realizes in this way the first hop for an user-initiated IP connection. The hierarchy layers are connected via redundant links where traffic switching is realized via static link assignments based on customer IDs or acquired products , i.e. TV, Voice and Internet. In the context of Ethernet-based switches, VLANs are used to distinguish between customers respectively products.

The concentrator networks are monitored permanently. In the scope of this study, we have access to an operationally used PMS. The PMS contains inventory data of DSL access networks, including the device types, and customer IDs as well as the concentrator network topologies. In addition, the PMS processes performance data, i.e. bandwidth consumption upstream/downstream. Supplementary, the number of customers connected to a DSLAM is available as well. To ensure confidentiality, user relevant data such as the ID and the infrastructure relevant data as device and link names is pseudonymized. The performance data is measured in regular intervals in a range of 10 to 30 minutes. This data has been used for the optimization as well as for the simulation model representing DSL customers. The developed system is able to compute the optimal configuration of links, i.e. the customer to link assignment. In the following, we introduce the architecture for performance optimization.

3. PERFORMANCE OPTIMIZATION IN DSL ACCESS NETWORKS

We have developed a design concept for agent-based *Decision Support* and realized the application for optimization and continuous forecast of traffic distribution accordingly.

3.1 Agent-based Design of a Decision Support Framework (DSF)

We created an abstract framework for Decision Support based on agents, using an autonomic control loop as described in [4]. Figure 2 denotes a view on the decision support framework, which initially targeted the domain of security related system states and the enforcement of security policies. In general, there are four distinct functionality classes agents are associated with:

The *Data Acquisition* agents are responsible to interface with existing applications and devices. They are mediators and their task is to collect data, store and deliver it in an intermediate format. The raw data is processed in more suitable knowledge representation formats by the *Data Processing* agents (example in next Section). In addition, they conduct consistency checks, aggregation and filter operations.

A *Data Exploitation* agent is capable of inferring knowl-

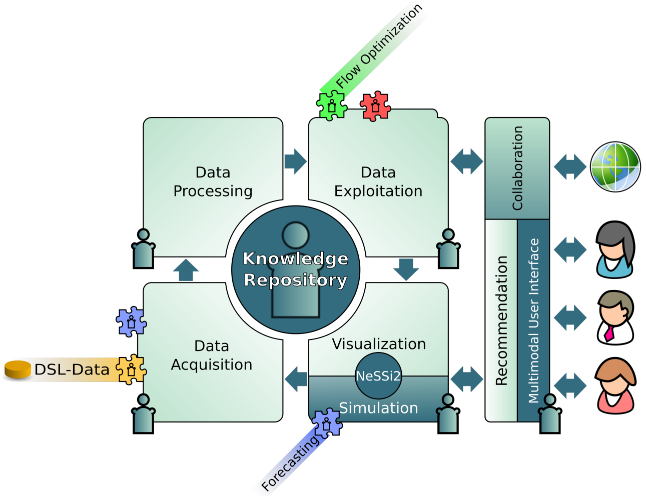

Figure 2: Agent-based Decision Support Framework- Overview

edge based on the data provided by the data processing agents. A focus is the prediction of system states and management. In addition, exploitation agents may trigger reactions, either automatic actions altering the system state or informative feedback for the operator of a system. The agents responsible for *Visualization/Configuration* are responsible for user interaction and manually driven configuration changes of the observed system. We realize the DSF with the agent framework JIAC[2] [6] and present the resulting application in the following.

3.2 DSF for Performance Optimization

We deploy two agent nodes (in JIAC V, a node is a Java virtual machine for running several agents), one node hosting the *NeSSi²* agents for visualization and manual configuration and the other node hosting the remaining DSF agents. In Figure 3, on the left-hand side the rectangle indicates the DSF agents for data acquisition (DAA), data processing (DPA) and Optimization (OA). The OA is a realization of the introduced functionality class exploitation. On the right hand side, the agents used for running simulations with *NeSSi²* are depicted. Here, we refer for further description of *NeSSi²* to Section 5.1. The screenshot in the top area of figure 3 depicts the *NeSSi²* user interface, whereas in the bottom an extended *NeSSi²* database is shown. The functionality of the incorporated agents is as follows:

Data Acquisition and Processing: The DAA receives and stores the performance measurements exported from monitors in the DSL access network in an interval of 15 minutes. In addition, it receives inventory data in a weekly manner representing a network topology. Next, the DPA extracts information out of raw data, used for visualization, optimization and forecasting. This includes the creation of the access network topologies, consistency checks of performance as well as inventory data, and other statistical metrics such as the busy hour of a link, i.e. a sliding 60-minutes interval with the largest traffic amount. If irregularities are already detected here, they are documented via a log file and optionally reported by e-mail.

Optimization: After the DPA finished his tasks, optimization can be carried out. We focus here on local optimization, i.e. the balancing of traffic between parallel links. The local optimization refers to the redistribution of the customer traffic on parallel links. This problem can be modeled as the well-studied bin packing problem, where the links represent the bins, the capacity of the bin is the capacity of the physical link, and the items are the bandwidths consumed by the customers. The size of an item naturally corresponds to the aggregated amount of traffic of all customers connected to one specific DSLAM. We chose the worst-fit decreasing heuristics which is most suited to achieve a balanced distribution over the bins [3]. This involves sorting the DSLAM traffic amounts according to size, and subsequently putting them in the bin/on the link with the most space remaining. In the following, we introduce traffic models that can be used to verify the optimization results (c.f. Section 5.3).

4. TRAFFIC MODEL

A main ingredient of our simulation tool is a mathematical model of the flow occurring at DSLAMs. In the sequel of this section the model is presented and its plausibility elucidated. Further, it will be revealed how model parameters are extracted from the data and how this theoretical model is utilized by simulation.

4.1 Model for the Departure Traffic at Flow Level

We propose a mathematical model for traffic at flow level comprised of the aggregated bandwidth consumed by the customers connected to one specific DSLAM. The measured traffic exhibits a noisy and unpredictable nature but still having some characteristics. Therefore, we propose a stochastic model able to mimic the underlying dynamics. At each DSLAM upstream and downstream flow is modeled based on the number of assumed customers connected to this DSLAM. This flow serves as input for the simulation described in Section 5.2. Depending on the average number of customers (low or high) connected to a DSLAM, we employ model classes for individual and aggregated flows respectively.

4.1.1 Flow Generated by an Individual Subscriber

We assume each subscriber to generate a flow which is modeled as an *interrupted Poisson process* with partially time dependent parameters. This is a standard server theory model [8, 10, 12] which is motivated by the following scenario: A subscriber requires to be served when she has time, i.e. she waits an exponentially distributed time with a time depending parameter $\lambda(t)$, which models her availability, before being served. The service also lasts an exponentially distributed time, but with a time independent parameter μ representing the minimal service length she always requires. The result is an interrupted Poisson process, called $X(t)$, alternating between the states 0 (inactive/not served) and 1 (active/being served). This process possesses the Markov property and is therefore fully described by the infinitesimal transition rates μ and $\lambda(t)$ for the transition from 1 to 0 and 0 to 1 respectively. In service also la the

[2] http://www.jiac.de/

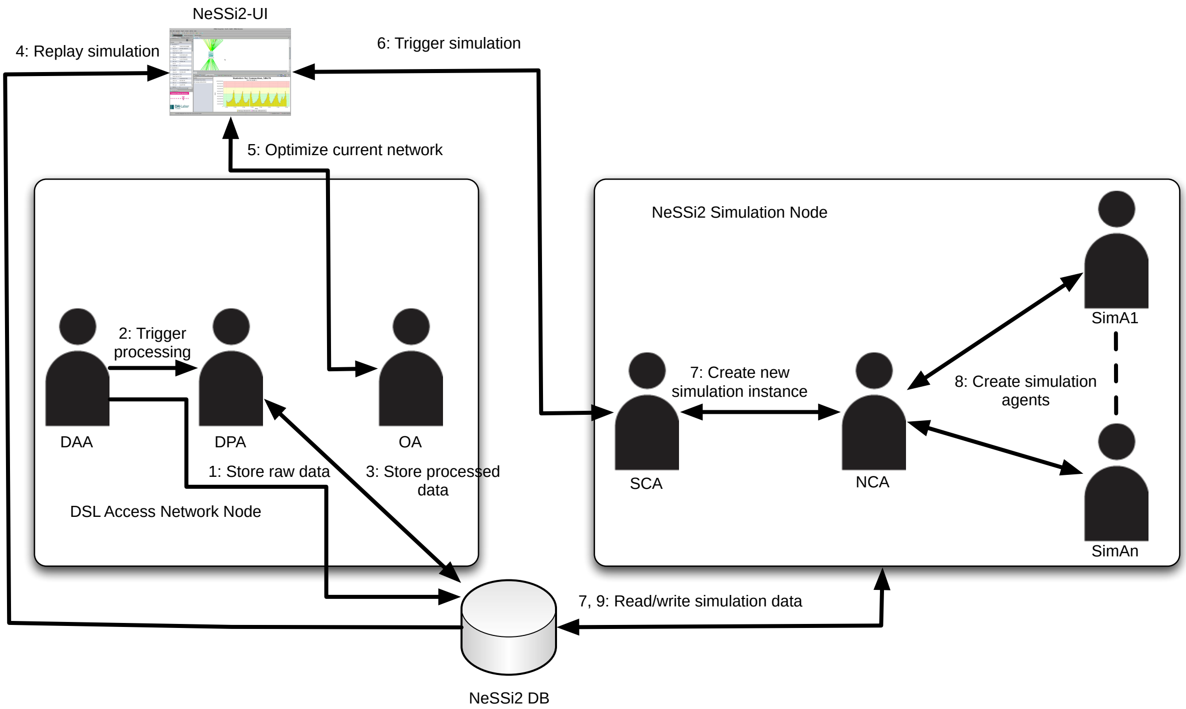

Figure 3: Realization of DSF for Performance Optimization. Agents observe the DSL access network performance measurements, carry out network optimization proposals or forecast future network behavior.

transition from 1 to 0 and 0 to 1 respectively. In other words,

$$\begin{aligned}\mathbb{P}\{X(t+h)=0|X(t)=1\} &= \mu h + o(h),\\ \mathbb{P}\{X(t+h)=1|X(t)=0\} &= \lambda(t)h + o(h), \quad \text{as } h \to 0,\end{aligned}$$

for all possible times t, assuming all subscribers behave identically.

4.1.2 *Aggregation of Subscribers Flows at the DSLAM*

The flow X, which arrives at a DSLAM with N subscribers, is computed as the accumulation of all single subscriber flows X_i, $i = 1, \ldots, N$, i. e.

$$X(t) = (1 + Y(t)) \sum_{i=1}^{N} X_i(t), \qquad (1)$$

where $Y(t)$ is a white noise which influences $X(t)$ proportionally to the total flow and represents unrelated random effects. Simulations of 1 are depicted in Figure 4.

When the average number N of customers per DSLAM is large, the simulation of the departure traffic by (1) is computationally infeasible. According to Donsker's theorem [5] the normalized version of (1) converges towards a *Gaussian process* $Z(t)$, which is fully described by its mean function $m(t) = \mathbb{E}[Z(t)]$ and covariance function $k(t,s) = \mathrm{COV}[Z(t), Z(s)]$ for times s, t. Simulation results for these processes are depicted in Figure 5.

4.2 Estimating Model Parameters

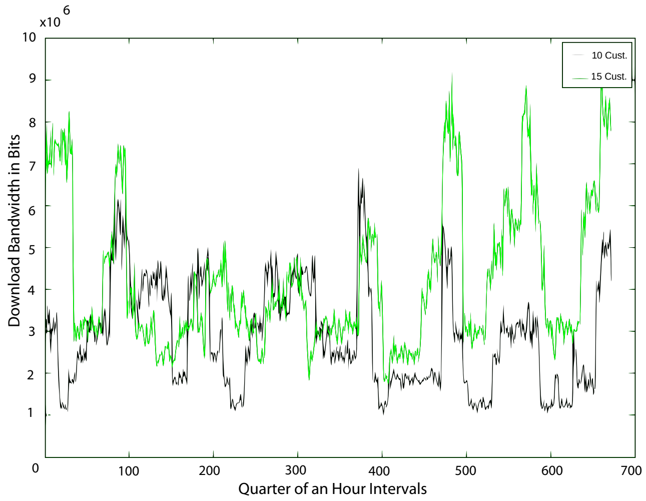

Figure 4: Simulation of the departure traffic at a DSLAM with 10 and 15 subscribers respectively.

The parameters of our model are derived using statistics of actual traffic flows such that the following is achieved:[3]

1. The first and second order statistics including the autocovariance function of the stochastic process (the model) will match those of the actual flow in our data set. We employed maximum likelihood estimators or

[3] A standard requirement, c.f. [13]

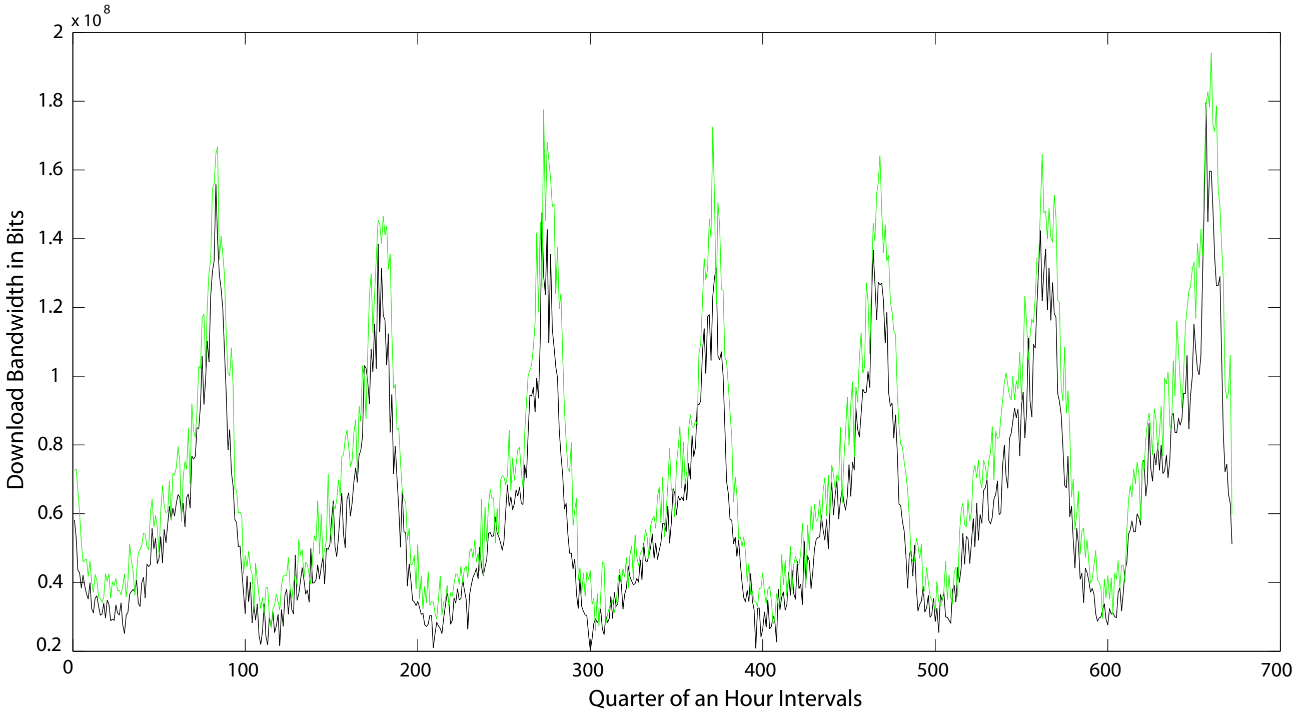

Figure 5: Simulation of the departure traffic of two DSLAMs with many subscribers, i.e. >100, modeled by Gaussian processes.

at least numerical approximations for this purpose.

2. The stochastic process should accurately reflect the nature of the traffic that is being modeled.[4]

4.3 Simulation of Departure Traffic

The applied simulation model is discrete-event based. In this regard, simulation time is following a tick-wise progression (a tick, called h, is the smallest discrete time span within the simulation). Our model guarantees that the state of a process for the next tick $X(t+h)$ is computable by the knowledge of the process history $X(s),\ s \leq t$. Since both presented process classes exhibit the Markov property, it is sufficient to take into account only the current state $X(t)$ when computing $X(t+h)$. Hence, the conditioned transition densities are of importance. Assume that our process is in state x at time t, $X(t) = x$. Then, in the *Gaussian* case, the conditioned random variable $X(t+h)|X(t) = x$ is normally distributed with mean

$$m(t,h) := m(t+h) + k(t,t+h)\sigma(t+h)\frac{x - m(t)}{\sigma(t)}$$

and variance

$$v(t,h) := \sigma(t+h)\sqrt{1 - k(t,t+h)^2},$$

where $\sigma(t)$ denotes the standard deviation of $X(t)$. Note that this discretization in time does not cause a loss of precision. The considered processes can be theoretical exactly simulated in discrete time steps since both considered process classes exhibit the Markov property. Still, the natural gap between model and computer simulation remains due to the usage of pseudo-random numbers, machine accuracy, and the numerical condition of executed operations. When we are simulating a *Markov jump process*, a random variable is drawn from an exponential distribution with a parameter depending on $\lambda(t)$ and μ. If it is smaller than h, the process performs a jump according to the embedded Markov chain and remains in the current state otherwise.

The described processes model the *downstream*. Our data analysis revealed that up- and downstream are strongly correlated: for the correlation coefficients between up- and downstream for a fixed time holds $\rho(t) \geq 0.75\ \forall t$. Furthermore, the standard deviation of the up- and downstream ratio $r(t)$ is less than ten percent of the mean for all t. Thus, when the downstream $d(t)$ for a DSLAM is already simulated for a given time t, the value for the upstream $u(t)$ for this very DLSAM can be derived from the downstream, the time dependent correlation $\rho(t)$, and the ratio $r(t)$:

$$u(t) = d(t)\mathbb{E}[r(t)]W(t),$$

where W is normal distributed with mean 0 and variance $\rho(t)$. The random variable Y can be generated by drawing from a normal-standard distribution and multiplying the outcome by $\sqrt{\rho(t)}$.

[4] Admittedly, this criterion is not rigorous. In our case it means that observed traffic paths for a DSLAM with many subscribers look like a typical path of a Gaussian Process and that for a DSLAM with few subscribers the observed paths are similar to the typical path of a Markov jump process with pertubation.

4.4 Discussion on Model Assumptions

In our model, three particular critical assumptions are made:

- Subscriber homogeneity: It is assumed that all subscribers behave according to the same laws.
- Independence of subscribers.
- Range of time parameter over one week: Let T be the amount of ticks in a week and X a process describing the flow at a DSLAM, than X_t has the same distribution as X_{t+nT} for all $n \in \mathbb{N}$.

The first two assumptions are necessary to keep the model mathematically and computational tractable. In addition, user data is measured at the DSLAM level so that separate data for users is not available.

The third assumption is a natural one as our lives tend to follow a weekly rhythm. However, further investigation and proof is expected from the following test: compare the correlation of weekdays of different weeks with correlations of days within a single week. This would lead to conclusions such as Mondays of arbitrary weeks exhibit more traffic similarity than a Monday and a Sunday of the same week.

Unfortunately, this test has not been possible so far due to two reasons: we only have received input data for two weeks, and due to pseudonymization, the flow data of a particular DSLAM from one week could not be correlated with the data from the second week. This makes it impossible to compare the correlation of weekdays of different weeks with correlations of days within a single week.

Nevertheless, our dataset gives further hints that the model assumptions are correct. The statistical characteristics of two distinct weeks are very similar. This is shown in Figure 6, where the normalized mean of the two weeks is compared.

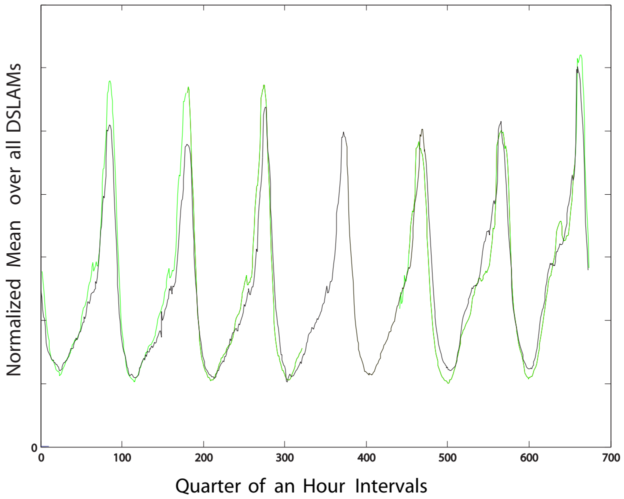

Figure 6: Down flow mean function $m(t)$ of a February week and a June week normalized by the weekly mean of each week $\frac{1}{T}\int_0^T m(t)dt$. Data of the June week is lacking one day.

5. SIMULATION-BASED VERIFICATION AND FORECAST

The two main objectives of conducting simulation are to verify the correctness of optimization results described in Section 3.2 and to predict upcoming congestions. In this section, we present our simulation environment *NeSSi²* and describe how the DSL access network simulation is realized by means of a new extension mechanism to incorporate different domain models and simulation logic. Subsequently, we give a detailed description of the use cases associated with the aforementioned objectives.

5.1 The Simulation Environment *NeSSi²*

In prior work, *NeSSi²* (Network Security Simulator), an agent-based[5] network simulation environment has been introduced [11, 2], allowing discrete event-based, packet-level simulation. Each simulated device contains a network layer enabling IPv4 or IPv6 packet transmission. On top of the network layer, end devices additionally contain a transport layer offering TCP and UDP as well as application layer protocols such as SMTP, HTTP and IRC. The discrete time steps in *NeSSi²* are referred to as "ticks". *NeSSi²* uses *profiles* to model user- or system-inherent node behavior. Thus, the profiles allow customization of traffic by specifying port ranges, mean packet sending rates and other distribution function-dependent parameters.

NeSSi² is comprised of three main components: the graphical user interface provides an *editor* for creating and editing networks or profiles as well as the combination of both elements for the preparation of a simulation configuration. A *simulation result perspective* allows the visualization of conducted simulation runs by means of a chart view and visual effects such as highlighted edges in the network editor indicating traffic distribution. Independent of the GUI, the back end uses the agent-based environment for the execution of simulation runs. Third, the simulation perspective is backed by a database where network model and simulation event information is stored and retrieved for graphical display. An EMF[6]-based common data model facilitates the interoperability of the components.

5.2 DSL Access Network Simulation

A challenge is that *NeSSi²* has been intended for packet-level simulations, starting at the IP layer. In order to model the network traffic as flows, we utilized the extension capabilities of *NeSSi²* to introduce a new domain model as well as simulation logic. An alternative traffic model is presented in Liu et al. [9] where the efficiency of fluid against packet-level simulation is compared based on the simulation event rate, i.e. number of events to be processed each tick. Generally, fluid simulation turns out to be more efficient the higher a generated packet rate is. However, the ripple effect, i.e. propagation of fluids rate changes, has severe impact on the number of events to be processed.

An alternative for the sole use of fluid simulations is presented by Kiddle et al. [7]. The authors propose a hybrid approach for concurrent packet- and flow-based network simulation. The incentive is to simulate background traffic using

[5] *NeSSi²* used also the previously mentioned agent framework JIAC.

[6] Eclipse Modeling Framework, `http://www.eclipse.org/modeling/emf/`

a flow model but foreground traffic via packet-level simulation, where hybrid buffers that operate in packet, fluid or hybrid mode change their state according to the incoming type of traffic. Their evaluation results indicate that a speed up of a factor 20 if feasible compared to purely packet-based simulation with slightly decreasing accuracy for jitter and latency. The applicability of the hybrid model is limited in highly congested networks or when strong ripple effects are present.

Upon closer examination, we found that none of the described approaches was suitable to the objectives of this research and the available performance data. A packet-level/hybrid simulation was not necessary because the modeling of individual protocols was not an important factor. In addition, the measurement interval for the performance data is in the range of minutes, so that individual peaks in the traffic flows are not visible. In summary, not the individual traffic peaks but the overall utilization of the network is the focus of our attention.

We developed an extensible network model based on generic nodes and edges based on the *Java Universal Network/Graph Framework*[7] providing out-of-the-box support for a wide range of graph operations and algorithms, e.g. shortest path computations, importance measures etc. The *Java Generics* concept enables the creation of specialized models such as the IP model or the newly created DSL access network model. Here, we encapsulate the domain specific entities incorporating DSLAMs, Aggregation Switches and BRAS. We do not consider each DSL modem but the aggregated user traffic on a DSLAM basis. This represents also the atomic unit for traffic-to-link assignments.

NeSSi^2 also provides a plug-in mechanism to exchange the simulation logic based on the *Mode* concept. We have realized the simulation logic by implementing a custom mode for DSL access network simulation. In this fashion, we are still able to use basic functionality *NeSSi*^{2} provides, such as GUI, logging framework, simulation execution runtime. In summary, the course of action for the simulation of a specific BRAS tree is as follows: In the initialization phase, for each DSLAM an instance of the traffic model (c.f. Section 4) is created with the number of connected customers as the input parameter. Then, we perform a tick-wise computation of the DSLAM bandwidth demands and derive the bandwidth consumptions in the network tree according to the known traffic-to-link assignments. Bandwidth consumption as well as traffic demands that exceed link capacity and can not be satisfied are logged.

5.3 Simulation-based Verification of Optimization

The increasing number of customers as well as consumed bandwidth necessitates capital expenditures for DSL access networks. Here, the outlined *Optimization* (c.f. Section 3.2) returns optimal traffic-to-link assignments and hence capital expenditures can be reduced to the minimum required for frictionless operation. Additionally, the network provider profits from an increased customer satisfaction as throughput is improved by eliminating bottlenecks.

The developed simulation model allows the evaluation of what-if scenarios for network planning. In this regard, a network with a non-optimized routing strategy can be directly compared against one with an optimized strategy. The amount of customers can be increased to pinpoint bottlenecks. For illustration, Figure 7 contrasts simulations of the non-optimized network (left) to simulations of the optimized network (right). In this particular example, we doubled the number of customers in the simulation. The Figure is taken from the *NeSSi*^{2} graphical user interface, where the bandwidth consumption is indicated by the color of the links ranging from green (low) to red (high).

A set of twelve redundant links connect the BRAS (denoted as BBRAR_15052) and an aggregation switch (icon on the bottom). The circles mark three links for comparison. In the non-optimized version, the middle link is nearly congested, the right one almost orange and the left one completely green. In the optimized version, the color of the links is closer, indicating traffic is distributed more evenly over the available links. This graphical presentation is supposed to give an overview for the administrator of the system. When necessary, a closer inspection of the simulation results is possible by examining the relevant charts or accessing the raw data. Multiple simulation runs to satisfy a required statistical relevance can also be conducted.

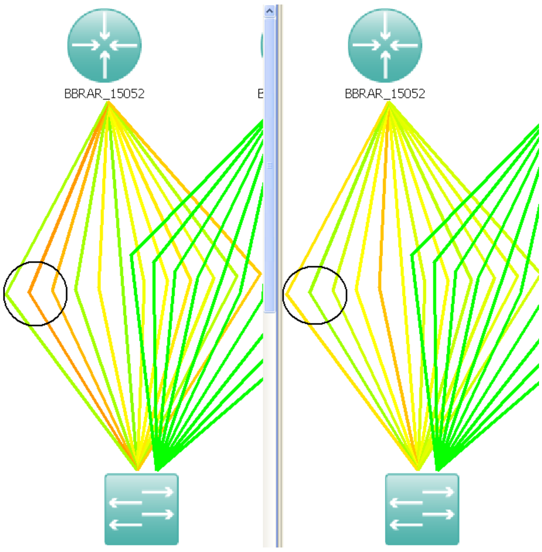

Figure 7: Comparison of two simulation runs: non-optimized double customers vs. optimized double customers.

5.4 Simulation-based Forecast

In contrast to the outlined network planning aspects, the applied agent platform supports a configuration of *NeSSi*^{2} as a real-time decision support system. Here, based on the architecture described in Section 3.2, the network is continuously observed and upcoming network states are predicted. In this regard, one big challenge is to acquire, process and exploit input data continuously in a limited time frame.

Figure 8 depicts the workflow of the real-time decision support system. It processes two different types of input: the network inventory data is provided to the system infrequently in irregular intervals. This includes the topology such as the devices and links as well as the traffic configuration. In contrast, in frequent regular intervals (e.g. 15

[7] `http://jung.sourceforge.net/`

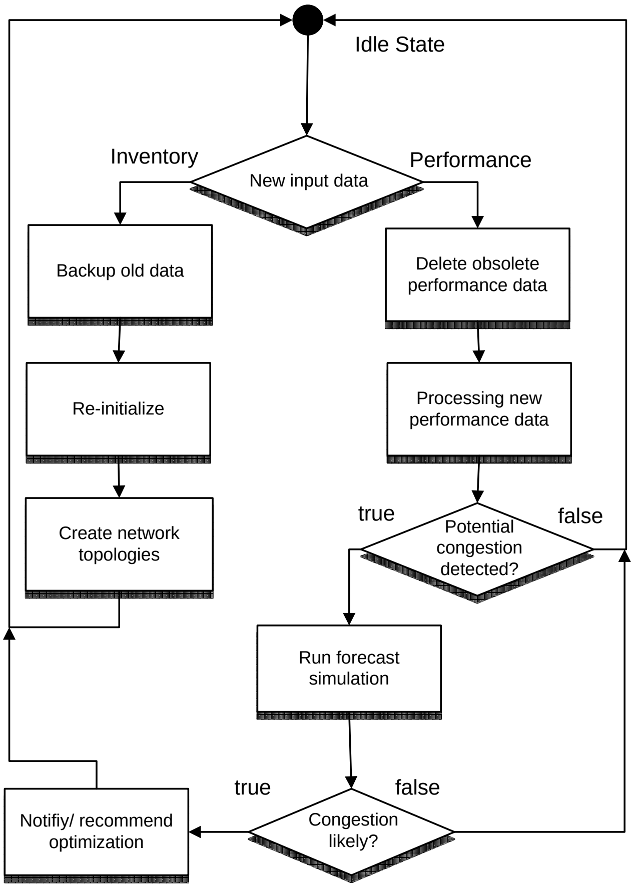
Idle State
New input data
Inventory
Performance
Backup old data
Delete obsolete performance data
Re-initialize
Processing new performance data
Create network topologies
true
false
Potential congestion detected?
Run forecast simulation
true
false
Notifiy/ recommend optimization
Congestion likely?

Figure 8: Process flow for triggering forecast simulations and optimizations.

minutes), performance data is measured and fed into the system.

Each snapshot of the inventory data represents the reference frame for a specific set of associated performance measurements, simulations and optimizations. In case new inventory data is acquired, old measurements, simulation and optimization results are archived and the system is re-initialized, including the reset of statistics or the creation of new database schemes. Afterwards, the network topologies are created and stored. The access networks are subject to frequent changes so that new snapshots are necessary. However, the whole course of actions to process updated inventory data is time consuming, preventing short intervals: we consider a weekly interval as an appropriate tradeoff.

In contrast, the right branch of Figure 8 can be traversed more quickly. Here, old performance data is cleaned up and the new data set processed. This includes the steps outlined in Section 3.2. In case the active input indicates suspicious link states, a simulation is triggered with the active consumed bandwidths as input for that specific time step. We provide a *Decider* interface to decide whether conducting a simulation is warranted. The rationale not to conduct a simulation is to save processing time as the system is working in a real-time environment. A sample implementation for the *Decider* is to compare the active state with previous states or to define a bandwidth consumption threshold.

When a simulation is triggered, it predicts the link utilization for the next couple of hours. If a second critical threshold is surpassed in the simulation, a notification is sent to the responsible entity, e.g. the control room of the part of DSL access network. This happens by the means of an e-mail service incorporated in the system. The e-mail includes information about the simulated network, the input interval and possibly congested links. This information prompts the notified person to look up the simulation results in the *NeSSi*^{2}graphical user interface.

In a first test with enabled debug settings, the system needed 40 minutes to process a set of performance data for all German DSL access networks were the monitoring equipment was deployed. This test was conducted on a MacBook Pro (2.4 GHz, Core2 Duo, 4Gb RAM). In this context, 117 simulations have been triggered. We found that the write access to the database, not the number of simulations, is the limiting factor when processing the real performance data. In this context, there is room for improvement to accommodate shorter intervals. To this end, it needs to be analyzed if all performance data must necessarily be written to the database. Basically, this is only important in case of an incident, so the notified admin can have a look at the system. However, the processing, analysis and simulation may be separated from storage and later inspection of performance data. Additionally, a machine specialized for this task, e.g. a dedicated database server, may be deployed to improve performance.

6. CONCLUSION AND OUTLOOK

We have presented an agent-based Decision Support Framework for the enhancement of existing network performance management systems in the scope of DSL access networks. In particular, we have focused on the simulation aspects, including provision of two realistic traffic models; one model is based on individual subscribers and the other is based on an aggregated view of multiple subscribers. These models have been created on the basis of real network performance data. In addition, we have presented new extension mechanisms for *NeSSi*^{2} and integrated, as an example, the domain model as well as the simulation logic for the DSL access networks. Finally, we outlined the application of simulation to verify optimization results as well the continuous forecast of potential future congestions.

The customer products we encountered in the examined network were quite homogeneous; but, because of the dynamics in the growth of Broadband Internet and the emergence of new services, a differentiation between the customers might be useful. This can be interesting for upcoming scenarios or the transfer of the traffic models to other domains such as cable connections. However, this requires additional performance data not yet available in the current DSF. This also included device characteristics, e.g. the backplane of incorporated switches. Although *NeSSi*^{2} provides a graphical editor for the creation of networks, generated DSL access networks can not be altered so far. This functionality will be added soon to enhance the measures for network planning.

Acknowledgements

The authors thank the project team on Deutsche Telekom side and their fellow colleagues at Competence Center Secu-

rity, DAI-Labor.

7. REFERENCES

[1] K. Bsufka, R. Bye, J. Chinnow, S. Schmidt, and L. Batyuk;. Agent-based decision support: A case-study on dsl access networks. In *Proceedings of the Seventeenth Innovative Applications of Artificial Intelligence Conference, Atlanta, Georgia.*, 2010. Under Review.

[2] R. Bye, S. Schmidt, K. Luther, and S. Albayrak. Application-level simulation for network security. In *Proceedings of the First International Conference on Simulation Tools and Techniques for Communications, Networks and Systems*, 2008.

[3] E. G. Coffman, Jr., M. R. Garey, and D. S. Johnson. Approximation algorithms for bin packing: a survey. pages 46–93, 1997.

[4] S. Dobson, S. Denazis, A. Fernández, D. Gaïti, E. Gelenbe, F. Massacci, P. Nixon, F. Saffre, N. Schmidt, and F. Zambonelli. A survey of autonomic communications. *ACM Transactions on Autonomous and Adaptive Systems*, 1(2):223–259, Dec. 2006.

[5] R. Dudley. *Uniform Central Limit Theorems.* Cambridge University Press, 1999.

[6] B. Hirsch, T. Konnerth, and A. Heßler. Merging agents and services — the JIAC agent platform. In R. H. Bordini, M. Dastani, J. Dix, and A. El Fallah Seghrouchni, editors, *Multi-Agent Programming: Languages, Tools and Applications*, pages 159–185. Springer, 2009.

[7] C. Kiddle, R. Simmonds, C. Williamson, and B. Unger. Hybrid packet/fluid flow network simulation. In *Proceedings Seventeenth Workshop on Parallel and Distributed Simulation, 2003. (PADS 2003)*, pages 143–152, 2003.

[8] L. Kleinrock. *Queueing Systems, Volume II: Computer Applications.* Wiley Interscience, 1976.

[9] B. Liu, D. Figueiredo, Y. Guo, J. Kurose, and D. Towsley. A study of networks simulation efficiency: Fluid simulation vs. packet-level simulation. In *Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 3, pages 1244–1253, 2001.

[10] L. Muscariello, M. Mellia, M. Meo, M. A. Marsan, and R. L. Cigno. Markov models of internet traffic and a new hierarchical MMPP mode. *Computer Communications*, 28:1835 – 1851, 2005.

[11] S. Schmidt, R. Bye, J. Chinnow, K. Bsufka, A. Camtepe, and S. Albayrak. Application-level simulation for network security. *SIMULATION*, page 0037549709340730, Aug 2009. `http://sim.sagepub.com/cgi/content/abstract/0037549709340730v1`.

[12] K. Xiong, H. Perros, and S. Blake. Bandwidth provisioning in adsl access networks. *International Journal of Network Management*, 19(5):427–444, 2009.

[13] M. Zukerman, T. Neame, and R. Addie. Internet traffic modeling and future technology implications. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 1, pages 587–596 vol.1, March-3 April 2003.