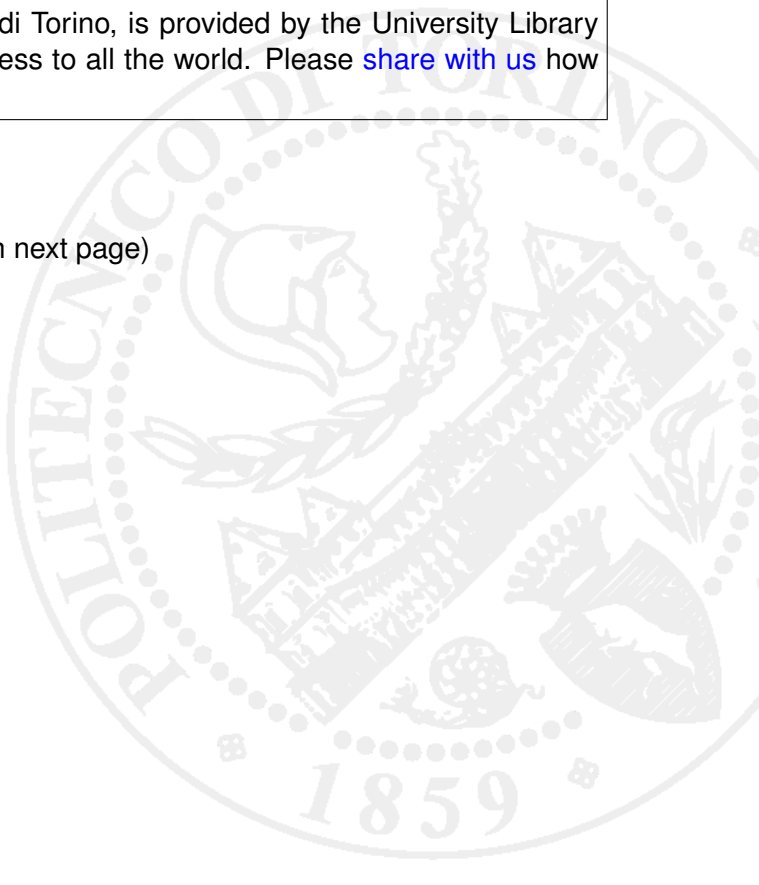

POLITECNICO DI TORINO

SCUOLA DI DOTTORATO

Dottorato in Ingegneria Elettronica e delle Communicazioni – XV ciclo

Tesi di Dottorato

From Measurements to Modeling The Tradeoff between Energy Efficiency and System Performance

Faheem Khuhawar
matricola: s169205

Tutore
prof. Marco Mellia

Coordinatore del corso di dottorato
prof. Ivo Montrosset

Dicembre 2013

Dedicated to my father and family for their everlasting support

Acknowledgements

I would like to acknowledge and thank for the ever willing help and supervision of Prof. Marco Mellia. His skill to guide and teach in minute detail to make difficult concepts look comprehensible has been a great help and a model inspiration to follow. The words are mere substitute to express my sincere gratitude to prof. Michela Meo, for her continuous guidance and valuable support throughout the completion of my research work.

I am also thankful to prof. Fabrizio Dabbene, Maurizio Munafo, Florin Bota and to all those personalities whose names are not mentioned here but have helped me.

Abstract

In this thesis, the work is based on experimental and modeling methodologies. I applied this starting from measurements of Asymmetric Digital Subscriber Line (ADSL) technology, which is the preferred high-speed access to the Internet. The work focuses on the problem of determining the maximum stable bandwidth that an Internet Service Provider (ISP) can offer. This task was accomplished by monitoring and analyzing a large set of ADSL end-users. Huge amount of collected data allowed us to get a detailed statistical analysis of the behavior of ADSL lines, such as the distribution, variable dependencies and correlation of effective bitrate with the physical measurements exposed by network devices. Analysis of collected data suggest that there is no clear trend to predict a downstream bandwidth of ADSL line based on line conditions, and thus requires an intelligent way of analyzing the ADSL lines. Thus, a Neural Network (NN) was employed, which is an intelligent machine learning tool to (i) learn the behavior of ADSL lines, (ii) extract useful information from huge set of measurements, (iii) automatically suggest maximum stable bandwidth. The results indicate that NN performs well in predicting end-users available bandwidth. However, NN is required to be properly trained, as well as needs careful selection of design parameters.

Later, the focus of the work was centered over the energy efficiency of telecommunication systems using mathematical modeling approach. Motivation of first work was to know how much energy efficient is Voice over Internet Protocol (VoIP) architecture as compared to traditional Public-Switched Telephone Network (PSTN) architecture used for voice communication. To answer this, these two architectures already implemented at our campus were extensively examined by means of measuring real power consumption. To generalize or estimate power consumption for any given number of users, a mathematical model of power consumption for both the architectures is built. The results indicate that VoIP architecture consumes a lot of power, yet VoIP system have the flexibility to be made energy efficient by adding some sort of energy-wise schemes.

The last part of the work investigates the interaction between the Transmission Control Protocol (TCP) and rate adaptation mechanism. Where rate adaptation is an approach to save energy by adapting the transmission rates or capacity according to traffic dynamics. To have deep insight of the interaction, a mathematical model based on fluid based framework is built to depict the behavior of TCP and rate adaptation scheme. The model is then tested for its accuracy and stability by conducting simulation experiments and steady state analysis. Later, the model is used to study the impact of tuning the parameters on system performance. The results suggest that the implementation of rate

adaptation scheme can be effective but a careful parameter setting is needed to avoid undesired disruptive interaction among controllers at different levels, that impair QoS.

Contents

List of Tables

List of Figures

Chapter 1

General Introduction

1.1 Problem overview

Importance of saving energy is increasing due to the rising trend in energy consumption by telecom industry, leading to increased operational cost and higher CO_2 emissions. Information and Communication Technology (ICT) being the evident energy consumer [1] due to the expansion of network devices to meet the emerging trend in the amount of video and data traffic, as well as the more and more widespread use of communication devices in general has led to the importance of deploying green networks. Hence, robust measurements are required to produce not only energy efficient networks but also maintain the standards of Quality of Service (QoS).

Guaranteing the QoS and to keep up the pace with emerging demands to provide value added services, for example, video services such as live television, video on demand, video conference, online gaming etc., that require low latency and high bitrate, has been known to have fundamental importance. Thus, ISPs are interested in knowing the maximum stable bandwidth that a specific ADSL line can support[1]. This would allow ISPs to know what services they can offer. For this reason, ISPs monitor regularly the performance of their network and this activity involves the collection of parameters such as, attenuation, signal to noise ratio, transmission rate, error count etc. Analysis of collected parameters suggest that there is no clear trend to predict a downstream bandwidth of ADSL line, and thus requires an intelligent way of analyzing the ADSL lines. Several techniques have been reported relevant to the evaluation and optimization of performance of ADSL link either at network layer or application layer. For instance, [2] reviews the previously reported bandwidth estimation metrics and tools to improve end-to-end transport performance. While [3] addresses the shortcomings of existing bandwidth estimation techniques and resolves it by developing a new measurement tool to estimate available bandwidth for ADSL lines. Similarly, [4] proposes another bandwidth estimation technique for asymmetric links by analyzing a packet's Round Trip Time (RTT) linearity with respect to its size. We

[1]Note that the ADSL equipments calculate maximum attainable rates which are often not achievable or unstable due to various reasons.

in our case address the problem of determining the ADSL maximum stable bandwidth at the physical layer.

The second part of work focuses on the comparison of energy efficiency between the PSTN and the VoIP architecture used for voice communication. Aim is to know, will it be efficient in terms of energy if we migrate from PSTN technology to VoIP technology, due to its shared data-voice infrastructure as well as using IP to transfer long-distance calls at low cost. Several works address the pros and cons of deploying VoIP architecture. For instance [5] highlights the number of considerations to take in to account before deploying VoIP infrastructure. Whereas, [6] theoretically estimate the power consumption of copper and fiber based telephone system.

The third part of work highlights the impact of implementing rate adaptation scheme on system performance, that is to know how rate adaptation scheme interacts with the protocols implemented at upper layer, in our case, TCPs congestion control mechanism. Why we are interested in the interaction is because if these two mechanisms are not properly tuned, they might negatively interact. For example, the rate reduction by rate controller might induce congestion in a network, which signals the TCP to reduce injected traffic, this TCP traffic reduction is indicated with the decrease in queue size, this triggers further rate reduction by rate controller. Eventually, this sort of interaction might lead to bad performance. Aim of this work is to know how these two control loops interplay with each other and to know what are the parameter setting that will lead to maximum energy saving without compromising the QoS.

1.2 Contribution

Regarding the part to obtain optimal stable data rate, we employed Neural Network, which is a classical feed-forward Neural Network to work as a predictor. We have showed through our simulations that designed Neural Network performs better in predicting proper data rate, given that the network is trained properly. Thus giving opportunity to ADSL users and service providers to know about the available stable bandwidth.

To address the rising questions: How much energy efficient are these two architectures? Can VoIP replace PSTN? or Will there be power saving in implementing VoIP? we compare these two architectures in terms of energy consumption using real or actual power consumption data. Later those measurements are used to build a simple mathematical model to estimate per user consumption. Given that, energy consumption for any given number of users for either PSTN and VoIP systems is estimated. We highlight that VoIP architecture is flexible and can implement possible energy saving schemes. For example, one possibility to save energy would be to put VoIP phones into sleep mode when there is no user activity using the standard Wake-on-LAN (WOL) Ethernet feature; a second possible space to save energy would be during inactivity period of ongoing call. Lastly, we propose various energy schemes describing possible ways to save energy along with their results.

In the last part of the work, we build a mathematical model based on fluid based framework to have in depth insight of the interaction between TCP and rate adaptation

scheme. After successful validation of the model through simulation experiments, we analyze the steady state as well as stability of the model to know its behavior in long run. Later, several experiments are conducted to know the impact of tuning the parameter values, that is to know the interaction between TCP and rate adaptation. Finally, to understand the impact of rate adaptation scheme on power consumption, we evaluate the amount of energy saving by comparing the energy aware solution to non-energy aware solution in transferring a given amount of data. The results suggest that there is direct relationship of QoS with power consumption.

1.3 Thesis layout

The thesis is organized as follows: Chapter 2 focuses over the problem that ISPs face in providing their customers the best line rate. We solve the problem by (i) monitoring and analyzing the huge set of ADSL lines, (ii) develop a machine learning tool and later perform various filtering techniques to train it properly, (iii) test the accuracy of tool and predict the best line rate based on line parameters. Chapter 3 investigates the power consumption of two well known voice infrastructures through real power consumption data. We take in to account the already deployed PSTN and VoIP systems at our campus. Later, we highlight how energy is wasted and build energy consumption models for both systems. Afterwards, possible ways to improve energy efficiency along with their results are shown. Chapter 4 talks about the interaction of rate adaptation scheme with TCPs congestion control mechanism. Where we describe basic TCP behavior through differential equations and then setting up of rate adaptation scheme, its validation through OMNeT simulations, steady state analysis to know the behavior of the model in long run, parameter sensitivity to know the interaction between TCP and rate adaptation, performance comparison between two proposed rate controllers, and eventually show the impact of rate adaptation scheme on power consumption. Finally, chapter 5 summarizes the overall conclusion and possible future work.

Part I

Reliable Data Transmission

Chapter 2

Predicting ADSL Lines Data Rate using Neural Network

2.1 Introduction

Broadband access to the Internet is a reality in many countries of the World. Among the different access solutions, ADSL/ADSL2/ADSL2+ based [7] technology is the most commonly adopted solution to provide end-users high-speed Internet access. Relying on the old telephone copper links that have been deployed by Telecom Operator during the past century, ADSL technology offers a capacity up to 24Mb/s and to 1.4Mb/s on the downstream and upstream channels respectively. Actual maximum bitrate is however strongly depending on the quality of the physical medium. ADSL signals are degraded by older telephone lines, by poorly designed microfilters, by radio frequency interference, by electrical noise, and, most of all, by long telephone wirings. These effects are especially significant where the customer's phone line is more than 4 km far from the DSLAM in the telephone exchange, which causes the signal levels to be lower relative to any local noise and attenuation. This will have the effect of reducing data rates or causing connection failures so that most of users' lines will not even get close to the maximum transmission rates.

Today, Internet Service Providers (ISP) are increasing their offer by proposing value-added services on the top of the simple Internet access. Many of those services pose more stringent Quality of Service (QoS) constraints than the one of simple data transfer. VoIP services require low bandwidth and latency, video services need higher bandwidth while gaming require moderate bandwidth and low latency. Guaranteeing a minimum service level for a ADSL line is becoming therefore a critical point for ISPs. In particular, ISPs are interested into a method to efficiently predict the maximum bandwidth the ADSL line can reliably support to determine which services can be activated on a specific contract. Indeed, users will be disappointed if they cannot enjoy the additional services the ISP has promised, while they would possibly accept to not use an additional service if told in advance.

In this work, we focus on the downstream channel which is more sensitive to physical

impairments. Working together with one of the main Italian ISP, we analyze the measurement data collected from a large installment of real ADSL lines. Then, we develop a machine learning technique to predict the possible optimal configuration of ADSL lines. In particular, we adopt a Neural Network (NN) machine learning technique which correctly predicts the maximum stable bandwidth a specific line can potentially support. We show that the training of the NN requires some ingenuity, i.e., the training set must be correctly defined to avoid misbehaving lines to fool the NN prediction [8]. Note that the DSL equipments calculate maximum attainable rates which are often not achievable or unstable due to various reasons.

Considering ADSL line performance, several authors addressed the problem of estimating the ADSL offered bandwidth at the network layer. See for example [4, 2, 3]. Those techniques can be used from one end-user to evaluate his ADSL link performance in terms of application layer throughput, and all assume a fully functional link. In this work, we face instead the problem of determining the ADSL maximum stable rate that could be supported at the physical layer. In other words, we answer the question "could this line be upgraded to the next physical layer data rate?". Considering supervised machine learning techniques, Neural Networks have been used for long time in classifying data, with many of the techniques summarized in [9]. More recently [10, 11] present the applicability of NNs in data mining for classification purposes, while [12] presents an overview of possible use of artificial intelligence in data mining. However, no previous work explicitly targets the design of a NN predictor considering ADSL lines.

2.2 Measurement dataset

This section presents the dataset we used to design and test the NN predictor. The first part describes the monitoring setup, while the second part presents some statistical data about the ADSL line characteristics.

2.2.1 Environment setup

ISPs monitor regularly the performance of their network to supervise the service offered to the users. This activity usually involves the collection and processing of information monitored via SNMP - Simple Network Management Protocol. For ADSL and ADSL2+, the list of Management Information Base - MIB - provided by the equipment is defined in ITU-T G.992.1 [13] and G.992.2 [14], and it covers all the aspects of the transmission system for the DSLAM concentrators, both at the customer and provider ends.

Among the several available MIBs, our partner ISP decided to focus in the collection and tracking of few specific parameters, that were considered the most reliable and representative of the current state of the ADSL line. These parameters are:

- Line Attenuation, both for the uplink channel, i.e., from the customer modem to the DSLAM, and the downlink channel, i.e., from the DSLAM to the customer modem. Line Attenuation is measured in dB, and the MIB usually provides it in tenths of

dBs, so that figures are reported with just one decimal place. Line Attenuation is linked to the length and quality of the physical line.

- Signal-to-Noise Ratio (SNR) Margin, both for the uplink and downlink channels, in dB. For this measure, depending on the DSLAM, the MIB was reported either as an integer or with one decimal figure. SNR indicates how much the channel quality is affected by noise.

- Current Transmission Rate for the uplink and downlink channels. It is reported in bps, and is the target of the optimization.

Besides there parameters, the ISP collects line error counters, Line Interleaving status, and the Attainable Transmission Rate, but these parameters were not considered in our study.

The ISP provided us with the data for a subset of their customer lines. The subset contains data for more than 9400 single ADSL lines, obtained from several DSLAM concentrators, using both ADSL and ADSL2+ technology. The measurements were collected every six hours for each active line, over a six month period, from 09/07/2009 to 03/07/2010. Each sample contained values for the parameters described above, plus some additional information, such as the line identification number, date and time of the collection, and the technology and identifier for the involved DSLAM. All collected data was inserted in a SQL database to allow the statistical analysis and data mining.

2.2.2 Dataset statistics

During the six months of data collection we obtained about 5.7 million samples, with 3.7 million samples corresponding to 5691 ADSL lines, and 2 million samples corresponding to 3750 ADSL2+ lines. This huge amount of data allows us to provide a detailed statistical analysis of the behavior of ADSL lines, studying distribution, variables dependencies and correlation.

In the following, we use the term *down* (*up*) when referring to the downlink (uplink) channel, respectively; for example, "attenuation down" will be used to indicate the Line Attenuation measured for the downlink channel.

Fig. 2.1 and Fig. 2.2 present the number of lines versus the average line attenuation measured during the six month period. The curves follow a bell-shaped distribution, with an average of 24.15dB for ADSL lines and 22.25dB for ADSL2+ lines. A similar bell-shaped distribution is obtained for the lines considering attenuation up, with an average of 12.50dB for ADSL and 8.95dB for ADSL2+ lines, as shown in Fig. 2.2. This reflects the physical impairments that copper lines pose: there are very good and bad channels with very low and high attenuation, respectively. This figure is mainly driven by the physical length of the copper wires.

Figures 2.3, 2.4, 2.5 and 2.6 show the downlink transmission rate in function of attenuation down and up, for ADSL and respectively ADSL2+ lines. The figures include the average downstream bandwidth for each attenuation value, calculated over all samples of ADSL or ADSL2+ lines with the specific attenuation up or down. Few considerations

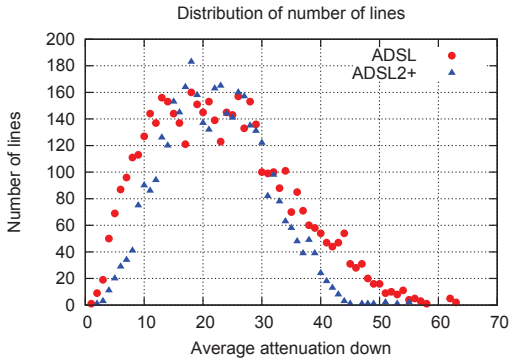

Figure 2.1. Distribution of number of lines (downlink)

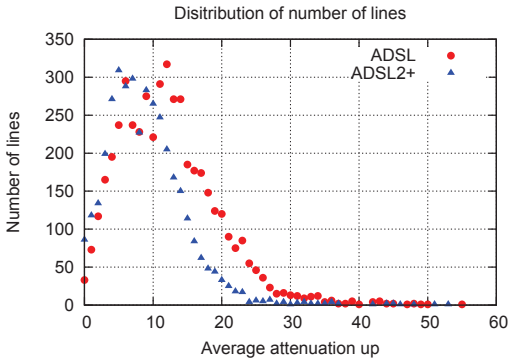

Figure 2.2. Distribution of number of lines (uplink)

hold: first, the actual downstream bandwidth decreases as the attenuation is increasing as expected. Note the small number of lines with the very highest attenuation values are statistically not significant, so that the average rate presents some artifacts. Second, downstream capacity is highly clustered to some specific values, e.g., {6,5,4,2}Mb/s for ADSL lines. This is due to the ISP preferring this discrete set of values. Third, despite this, the number of samples that deviated from the expected values is very large. For example, there are lines that can easily obtain 6Mb/s capacity despite suffering very high attenuation. Even worse, there are lines that cannot achieve rates higher than few Mb/s despite enjoying very small attenuation.

This shows that predicting the downstream rate of a ADSL or ADSL2+ line is a complex problem. To highlight this, Fig. 2.7 and 2.8 show the Probability Density Function - PDF - of the attenuation down for different classes of bandwidth. For the sake of simplicity, the PDF has been approximated by a mixture of Gaussian distributions. Mean, Standard Deviation (StDev) and the number N of samples for each class are reported in the small table close to the figure. It can be seen that the overlap between classes is very large, which underlines once more that predicting the line maximum rate cannot be easily achieved using the line attenuation.

To resolve this, statistical analysis or some sort of intelligent way of analyzing the lines and extraction of knowledge to create proper threshold for each class needs to be done as same attenuation value can give false results.

The analysis of correlation between SNR and Attenuation for ADSL lines is presented in Fig. 2.9 and Fig. 2.10 which report the average SNR margin down for samples of a class versus attenuation down. For the sake of clarity, only some of the ADSL classes are shown, being the ADSL2+ results qualitatively similar.

Similarly in Fig. 2.11 and 2.12 we have the average behavior of the most common ADSL2+ classes (beside the ones of ADSL there are four other classes: 8Mb, 12Mb, 16Mb and 20Mb).

We can observe there are lines having a low attenuation and high SNR, suggesting that there is room to upgrade the downstream bandwidth. e.g., increasing the bitrate is

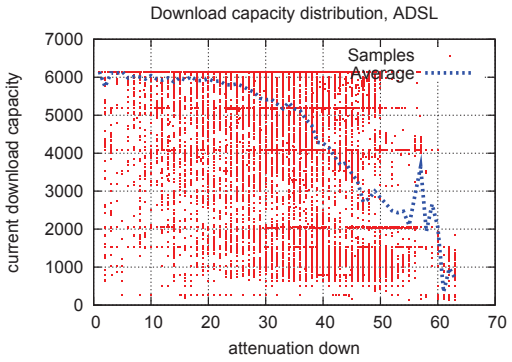

Figure 2.3. Distribution of downstream capacity for ADSL lines

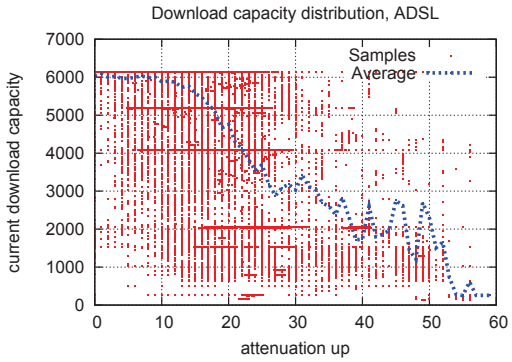

Figure 2.4. Distribution of downstream capacity for ADSL lines

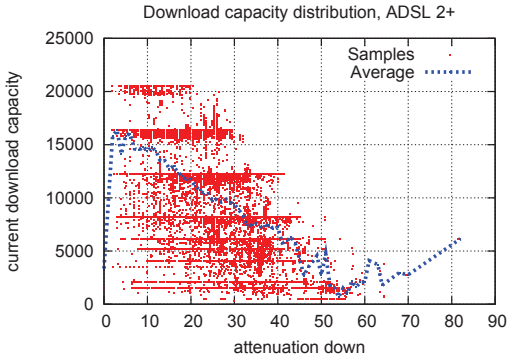

Figure 2.5. Distribution of downstream capacity for ADSL2+ lines

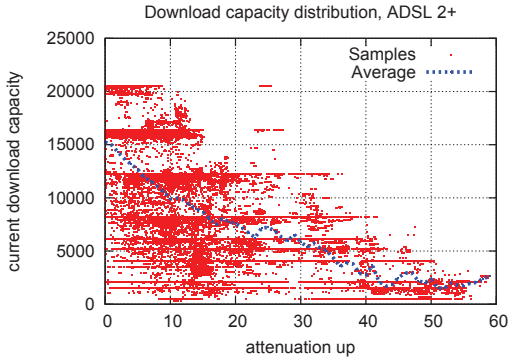

Figure 2.6. Distribution of downstream capacity for ADSL2+ lines

possible since there is still a large SNR margin. Considering larger attenuation, we observe that the SNR margin typically decreases. This reflects the fact that those lines are already reaching the physical channel limits.

The data reported so far shows that deciding the stable bandwidth supported by a specific line is a non trivial problem, and automated methods must be designed to predict each line supported bandwidth.

2.3 Neural Network Design

Artificial Neural Network is a non-linear multi-variable machine which can accept many inputs at a time and produce many outputs at a time. A training set is used to automatically build a model, which is then later used for prediction. NN has the ability to generalize a system and its prediction performance strongly depends on the training set quality. One advantage of NN compared to other supervised machine learning tools is its ability to cope with conditions that are not defined in the training. Artificial NN is the key

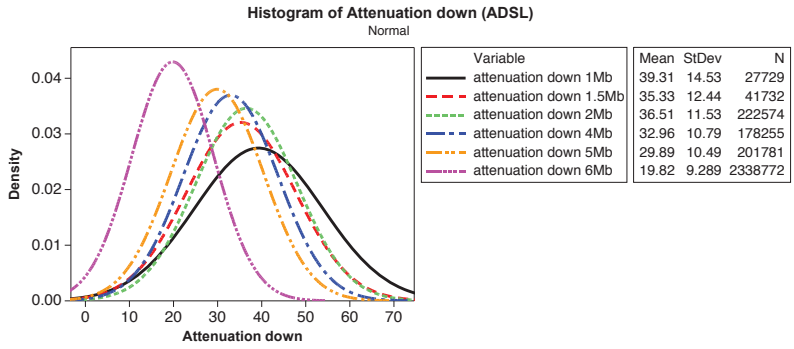

Figure 2.7. Histogram of attenuation down for ADSL classes

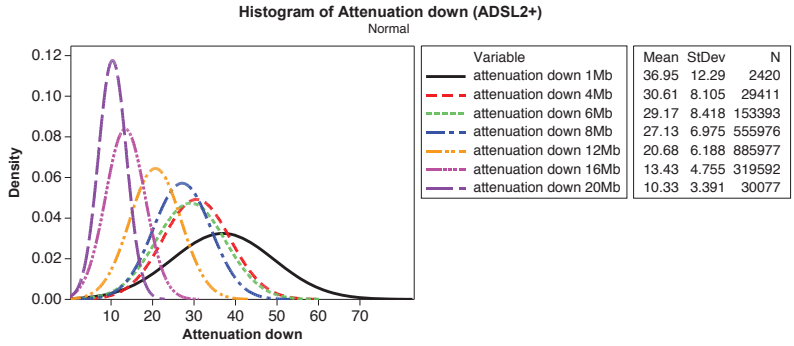

Figure 2.8. Histogram of attenuation down for ADSL2+ classes

technology to achieve classification of data i.e., the arrangement of data into predefined groups.

Multilayer Perceptron (MLP), which is a Supervised Feed-Forward Neural Network is the preferred NN used for classification. In this work, we design a MLP based NN to predict the maximum stable downstream capacity of ADSL and ADSL2+ lines. We tested also other supervised machine learning tools, such as support vector machines, or simple Bayesian classifiers, which however performed poorly compared to NN because of their inability to classify highly overlapped data. The comparison of those tools is however outside the scope of this work.

Our goal is to design a NN that, given a description of the line physical impairment (SNR and Attenuation), predicts the line maximum stable downstream bitrate, despite of what has been offered to ADSL user. To accomplish this task, we employed a NN to learn the behavior of ADSL lines, extract useful information, and automatically create thresholds for each class (data rate).

Only statistics from lines with stable current data rates have been used to train the NN to get reliable predictions and guarantee maximum stable bandwidth available for a specific ADSL line. All lines with commercial or technical limitations imposed by the ISP have been removed from the training sets in order to have accurate predictions.

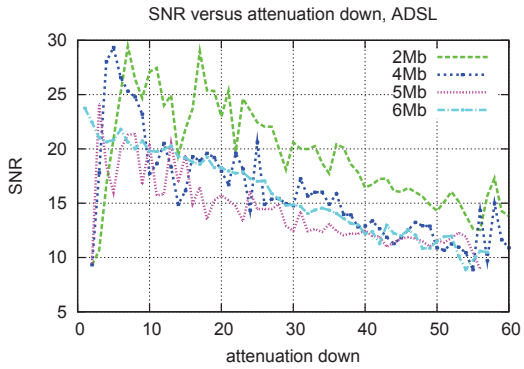

Figure 2.9. Average SNR down and attenuation down for ADSL bandwidth classes

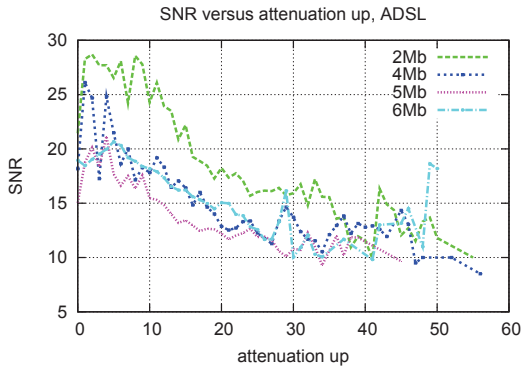

Figure 2.10. Average SNR down and attenuation up for ADSL bandwidth classes

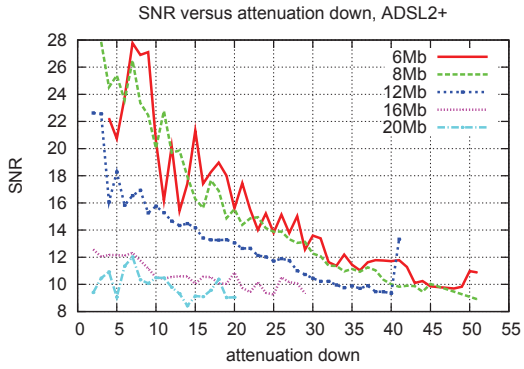

Figure 2.11. Average SNR margin down and attenuation down for ADSL2+ bandwidth classes

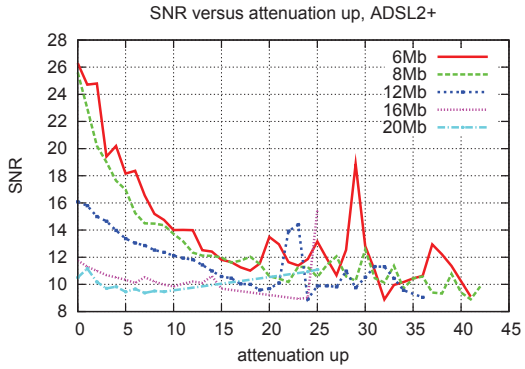

Figure 2.12. Average SNR margin down and attenuation up for ADSL2+ bandwidth classes

ISPs are interested in giving customers the "best line rate": a link with highest possible stable data rate. Today, to achieve this, they rely on some heuristic approach based on manual thresholding of line parameters to determine the best line rate. This manual approach requires long delay before achieving the highest stable bitrate.

By exploiting the NN prediction capability, we can react faster to changes in the line quality, suggesting an upgrade or downgrade based on learnings from the huge amount of measurement data of ADSL lines. Moreover, setting manually the thresholds is a complex task, so that the NN can also be seen as a mean to automatically learn and define thresholds.

2.3.1 Extraction of training dataset

Several experiments were conducted to extract a proper training set from the massive database of measured ADSL data. First, a dataset consisting of two attributes {*attenuation down, attenuation up*} corresponding to a specific class was extracted from the database. Those two attributes were used as an input to train the NN, with three possible classes as an output: 1, 4 and 6 Mb/s downstream classes as shown in Table 2.1.

Table 2.1. NN input and output relationship

Input	**Output**
Attenuation up, Attenuation down	Downstream bitrate from $\{1, 4, 6\}$Mb/s classes

The first step is to fully train the NN with the data to a satisfactory error level. A poorly trained NN will yield poor results. For this reason, various filtering techniques were used to extract a dataset that did not contain any misleading data from the database of ADSL lines and use it to train the NN. We tested several training sets which entails an increasing filtering. In the following we present the most significant tests, and discuss their performance. For training, we used two different training algorithms, namely Levenberg-Marquardt backpropagation and Gradient descent with momentum and adaptive learning rate backpropagation to update weights and biases. The performance of both the training algorithms was found to be similar. Similarly, training and results are consistent when repeated in independent runs.

The design of Neural Network was such that, it accepts a matrix of data with attributes on the columns and different ADSL lines on the rows. The result is a vector containing one value for each row, indicating the suggested class for the ADSL line. The NN model consists of two hidden layers with a number of 20 and 3 neurons respectively. These two hidden layers were set to use tangent sigmoid and linear transfer function.

To test the NN performance, the same testing dataset is used throughout. The testing dataset is extracted without any sort of filtering technique and is not a subset of the training set. It contains 300 distinct randomly chosen lines for each class to test the performance of NN in eradicating outliers present in the testing set. Figure 2.13 shows with different markers and color those samples in the plane {*attenuation down, attenuation up*}. As it can be seen, it reflects the noisy behavior we have observed before: some lines with excellent physical channel properties have however limited downstream bitrate, and vice versa. Figure 2.14 represents the same samples shown in Fig. 2.13 as a contour plot. Contour lines delimit areas where samples have similar current rate values, as reported by the labels. As it can be seen, there is no clear trend in the contours. For example, we observe some contours of low bitrate samples having low attenuation, while high bitrate samples are present for high attenuation lines. This shows that misconfigurations are present in the testing set.

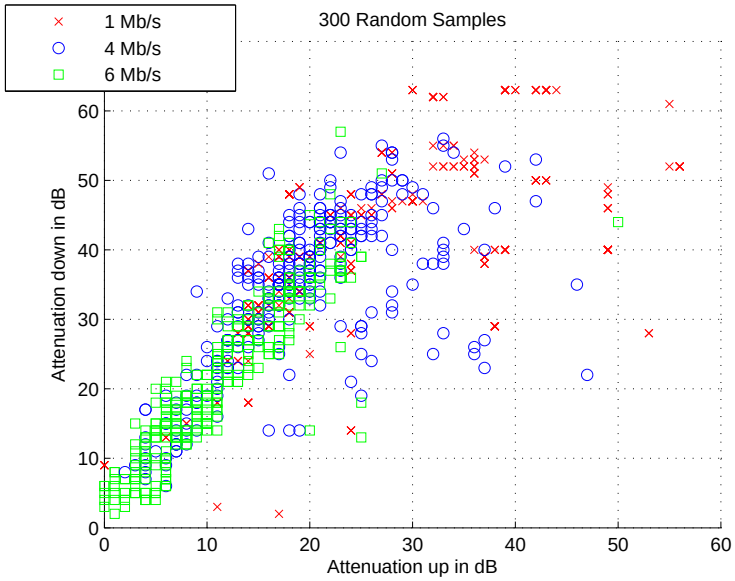

Figure 2.13. Scatter plot of testing dataset used to test NN prediction ability

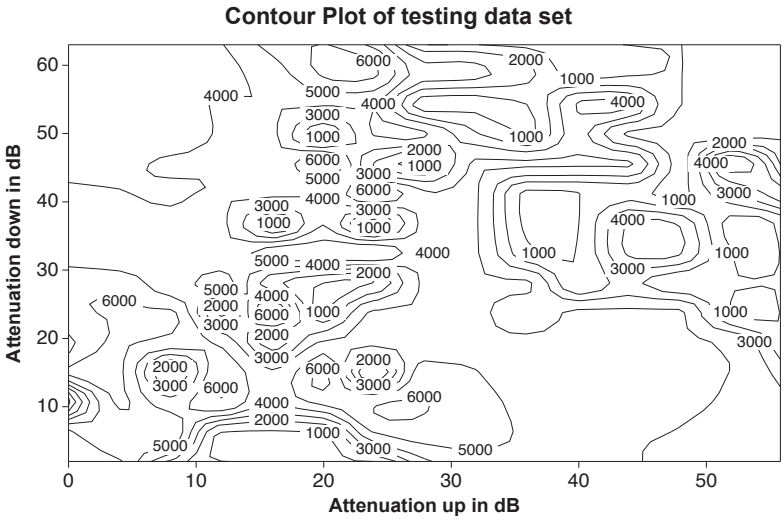

Figure 2.14. Contour plot of the testing dataset, 300 samples for each class

2.3.2 Training using stable lines

Intuitively, one could consider a line to be representative, if its downstream bitrate remains stable over time. Indeed, lines that often change their bitrate suggest that they suffer from physical impairments. We can therefore choose training samples among those that are distinctive and select only from ADSL lines that remained stable at the corresponding data rate for the whole six months long collection period. This leads to a dataset of 529,

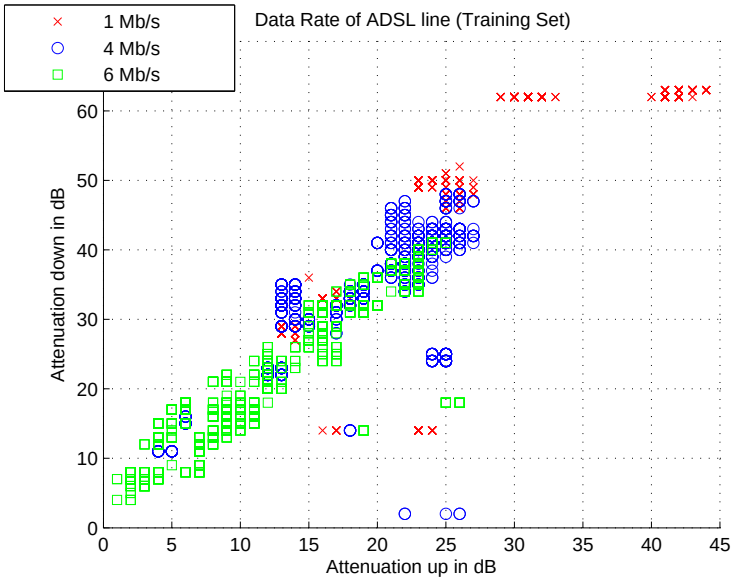

Figure 2.15. Scatter plot of initial dataset used to train the NN

1246, 1831 samples for 1, 4, 6 Mb/s classes respectively. The characteristics of this first training dataset is shown in Fig. 2.15 and 2.16. In particular, Fig. 2.15 shows with different markers and color the samples in the plane { *attenuation down, attenuation up* }. Fig. 2.16 instead reports the marginal distribution of number of samples versus attenuation down (left plot) and attenuation up (right plot). Intuitively, the selected dataset is not fully representative of the possible states, since points are concentrated on a small fraction of the state space.

After training, the NN performance is tested by classifying the testing samples described above. Results are illustrated in Fig. 2.17, which qualitatively shows the assigned class for each sample. Comparing Fig. 2.13 with Fig. 2.17, we observe that the picture is now more clear, but still there are samples that are classified as high bitrate despite the poor channel properties, e.g., there are green square dots in the top right part of the plot. It shows that the NN is often suggesting to upgrade the bitrate. Checking those lines, we observe that most of them are still characterized by very large attenuation, suggesting that the upgrade is unwilling to be practical for specific value of attenuation. Similarly, some red dots are still visible for lines with low attenuation, which seems a wrong configuration.

Since the training set suffers from overlapping of data as shown in Fig. 2.16, the NN suggested output is not reliable. Practically for some input samples, the class predicted by the NN was not consistent, calling for a better training set.

Indeed, the criterion used to select the training samples is not representing the typical conditions: a line can be stable at a low bitrate even if it could be stable at a higher bitrate. Practically for some input samples, the class predicted by the NN is not consistent, calling for a better training set which does not contain misleading data.

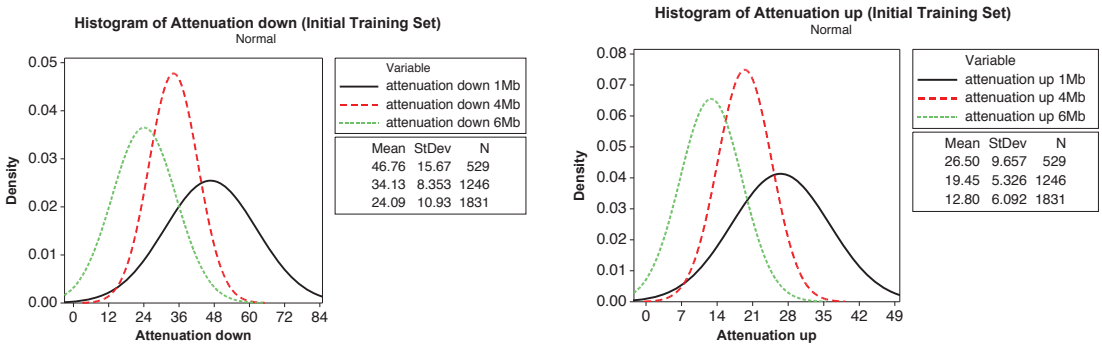

Figure 2.16. Histogram of the first data set used for training

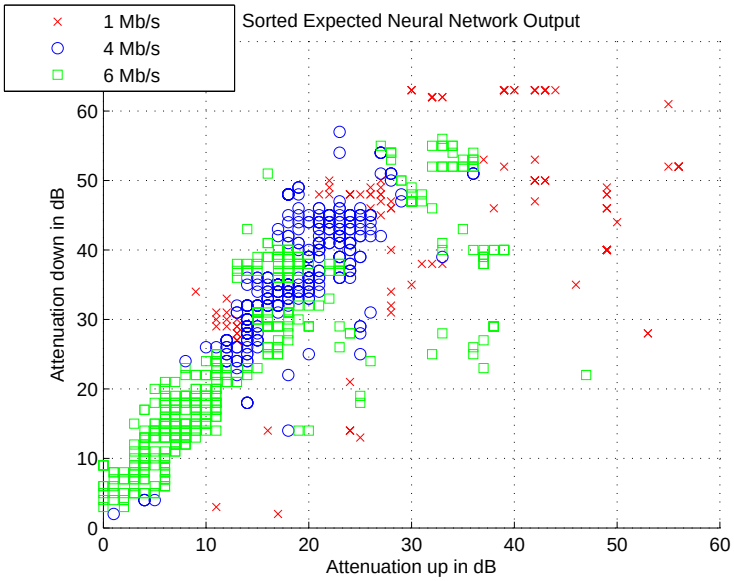

Figure 2.17. NN prediction based on initial training dataset against the testing dataset

2.3.3 Improving the training dataset

To improve the NN performance, it is clear that the training set plays a key role. We tested therefore several different training sets, obtained by filtering the set of lines according to some criteria. Due to lack of space, here we present our final criterion and evaluate its performance. The new training dataset is obtained by considering a line to be "representative" if all samples of that specific line take values inside the central 80% of class distribution, i.e., we discard samples in the bottom 10-percentile and the top 90-percentile of the distribution for the considered class. The scatter plot of the final training set using percentile restriction is shown in Fig. 2.18 and its contour plot is shown in Fig. 2.20. This time, we observe a much more clear distribution of samples in the three classes, so that the lower the attenuation is, the higher is the probability of achieving high downstream

Table 2.2. NN prediction of expected data rate (final training dataset)

		Expected Data Rate		
		1 Mb/s	4Mb/s	6Mb/s
Current Data Rate	1Mb/s	166	116	18
	4Mb/s	74	167	59
	6Mb/s	6	112	182

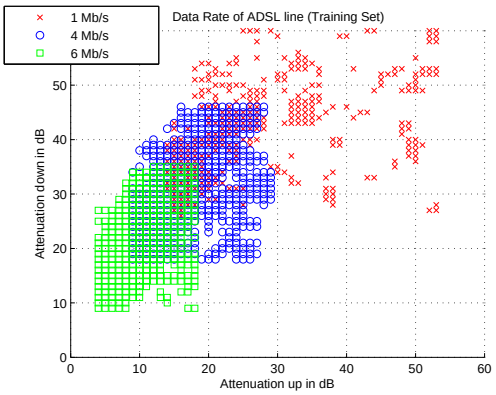

Figure 2.18. Scatter plot of final training dataset for the NN using percentile restrictions

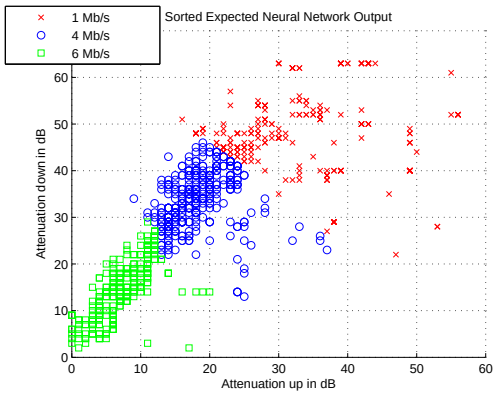

Figure 2.19. NN prediction based on the final training dataset

bitrate.

After training the NN, its performance is tested using the same testing samples as shown in Fig. 2.13. Results are reported in Fig. 2.19, while the summary of the upgraded/downgraded samples for each class is presented in Table 2.2. For example, out of 300 samples for 1 Mb data, 18 and 116 samples were upgraded to 6 Mb/s respectively 4 Mb/s class, and 166 samples were neither upgraded nor downgraded. Graphical illustration of this example is shown in Fig. 2.21 and 2.22 which shows NNs ability in correctly classifying 1 Mb samples into its respective classes, that is, lower the attenuation, higher the bitrate.

Qualitatively, it can be observed that the NN is able to correctly suggest the class of a given sample, so that it reflects the performance expected from the training set. Looking at the number of upgrades/downgrades and correlating this information with the actual line attenuation, we observe that the suggested operation is sound.

Comparison of Fig. 2.19 and Fig. 2.17 shows the higher ability of NN in successfully classifying the ADSL data rates into their respective classes, which reflects the better training set properties.

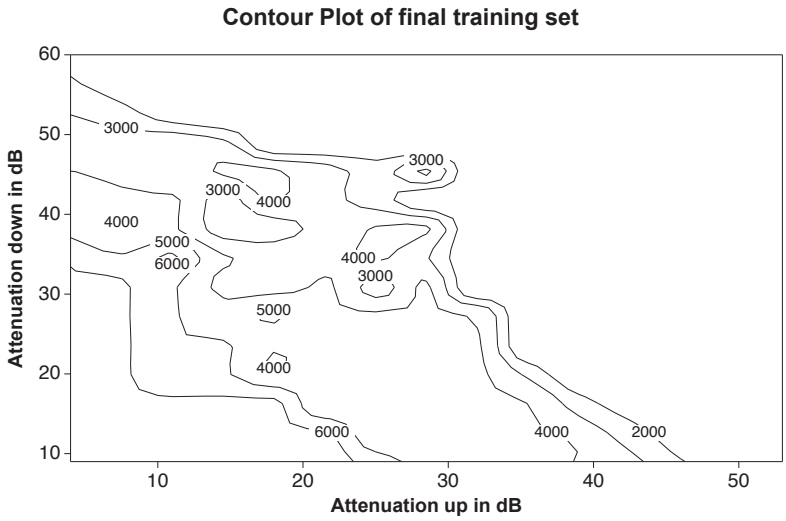

Figure 2.20. Contour plot of the final training set for the NN using percentile restriction

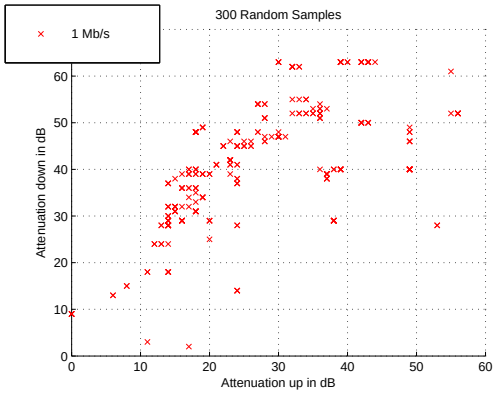

Figure 2.21. Testing set containing samples of 1Mb/s class

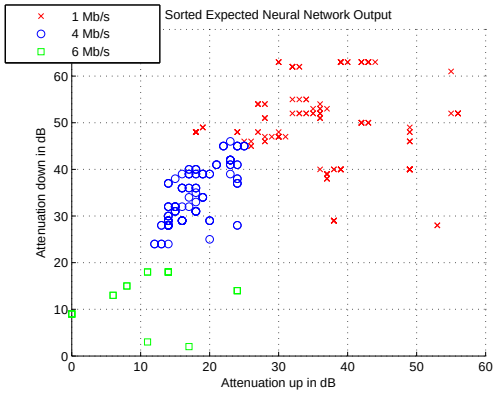

Figure 2.22. NN prediction for 1Mb/s class

2.3.4 Increasing the Number of Input Attributes

In this section we evaluate the benefit of increasing the number of attributes that characterize samples, i.e., by feeding the NN the SNR information as well, as shown in Table 2.3.

Figure 2.23 depicts the information carried by the SNR margin information. Different dots/colors represent the sample class in the {*SNR up, SNR down*} plane. The same samples as in the previous experiment are used for training. Note the coarse set of values exposed by the SNMP information. The intuition is that the overlap among samples of different classes is large, so that it is hard to predict the benefit of the information carried by the SNR margin attributes.

After training, the NN performance are evaluated using the testing dataset. Result is shown in Fig. 2.24. Compare it against Fig. 2.19 which shows the prediction of a

Table 2.3. NN input and output relationship

Input	Output
Attenuation up, Attenuation down, SNR margin up, SNR margin down	Downstream bitrate from $\{1, 4, 6\}$Mb/s classes

Figure 2.23. Characteristics of SNR down parameter of the training set

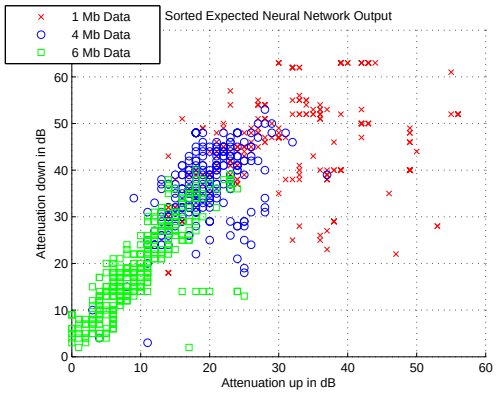

Figure 2.24. NN performance using 4 parameters against the testing set

NN that considers only attenuation attributes. Results show that the NN that considers attenuation only attributes performs slightly better than the NN with 4 parameters. The reason for this inefficient performance is due to the fact that NN trained with 4 parameters contains more samples of false or misleading data then in the 2 parameter training.

2.3.5 Considering more classes

So far we considered only 3 possible classes. What happens if the number of classes is increased? Is the NN still able to correctly classify samples? To answer this question, we considered ADSL2+ lines, and 5 data rate classes in {1,4,6,12,16}Mb/s set. The training set considers the same two input attributes, and includes samples that have been filtered discarding the bottom 10-percentile and top 90-percentile similar to the ones in Fig. 2.18.

Similarly, the testing set includes 300 randomly chosen samples for each of the 5 classes, whose characteristics are shown in Fig. 2.25, which shows the noisy distribution of samples among classes. Fig. 2.26 reports instead the NN prediction. Results suggest that the NN performs well even for an increased number of classes.

The summary of the upgraded/downgraded samples for each class is detailed in Table 2.4. In summary, the NN performance results show the best accuracy when using attenuation as a attributes for the training. Moreover, it is very important to consider a training set that does no contains any biased samples.

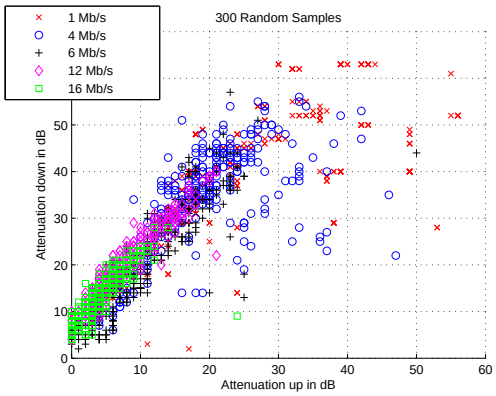

Figure 2.25. Testing set used to test NN prediction ability

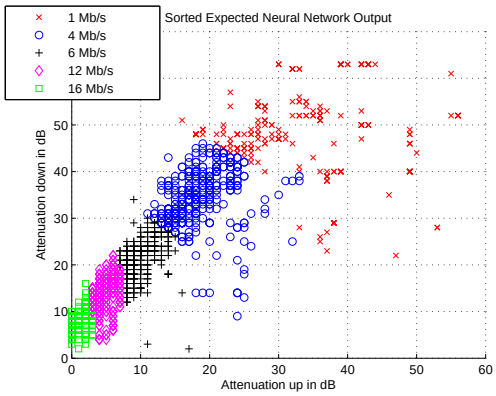

Figure 2.26. NN prediction in classifying five classes

Table 2.4. NN prediction of expected data rate (5 classes)

		Expected Data Rate				
		1 Mb/s	4Mb/s	6Mb/s	12Mb/s	16Mb/s
Current	1Mb/s	166	102	25	3	4
Data Rate	4Mb/s	71	163	40	20	6
	6Mb/s	6	96	94	62	42
	12Mb/s	0	47	98	109	46
	16Mb/s	0	1	44	118	137

2.4 Conclusion

For the first time to the best of our knowledge, we faced the problem of predicting the ADSL maximum stable downstream data rate. We have analyzed a large dataset of ADSL lines which shows that predicting the actual ADSL or ADSL2+ data rate is a complex problem. By exploiting a Neural Network predictor, we showed the importance of carefully selecting the training set and attributes to be adopted. Training set plays an important role in the performance of Neural Network, since the presence of outliers can easily poison the result. Other factors that influence performance are training time, number of hidden layers, number of neurons etc. However, in the considered scenario all these factors play a minor role compared to selection of the training set. The results suggest that, NN can be successfully employed to predict the expected ADSL/ADSL2+ data rate.

Part II

Energy Efficient Networks

Chapter 3

Comparison of Energy Efficiency in PSTN and VoIP systems

3.1 Introduction

There has been a wave of keen interest in saving energy due to increase in consumption by ICT industry [1], which leads to increased operational costs and higher CO_2 emissions. Continuous rising trend in energy consumption essentially depends on new services being offered, as well as on an increase of the number of devices and traffic.

Previous works dealt with power consumption of copper and fiber-based systems where different states of telephone systems are analyzed theoretically [6]. In our work, we focus on the comparison of two common architectures deployed in large voice communication infrastructures, that are: traditional circuit-switched technology used in PSTN and the VoIP technology. Many companies are moving away from the PSTN technology to obtain more functionalities for their voice systems while lowering the operational costs due to shared data-voice infrastructure as well as using IP to transfer long-distance calls. Switching from PSTN technology to VoIP technology could lead to an increase of power consumption, so the question is which architecture out of these two alternatives is more energy efficient? The two technologies follow two opposite design choices: on the one hand, complex structured circuit-switched technology utilizes centralized switching and group switching among Private Branch Exchanges (PBX) that implements all intelligence interconnecting very simple phone devices; on the other hand, the VoIP technology uses simple Ethernet switches and a gateway to interconnect intelligent phones that implement all advanced features.

Studies have shown a swift increase of the number of VoIP users over the past few years [15, 16]. With this constant growth of VoIP users and power consumption being a critical issue, the question would be, how much energy efficient are these VoIP systems as they require continuous consumption of energy [5]. And, can VoIP replace the traditional circuit switch technology without increasing operative cost due to higher energy consumption?

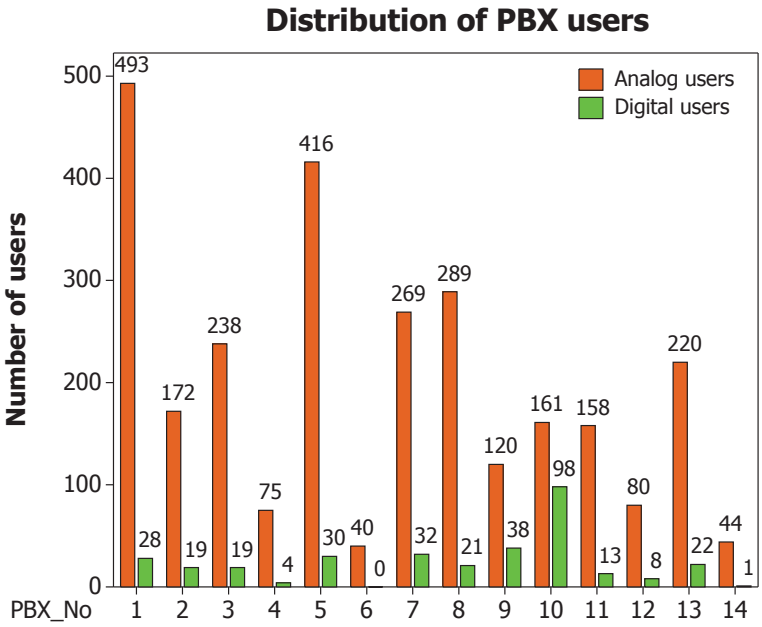

Figure 3.1. Number of users at Politecnico's PBX

3.2 Private Voice Systems

3.2.1 Traditional Circuit Switched System

We consider the telephone system of our university campus as reference to compare VoIP and PSTN architectures. The configuration implemented at Politecnico PSTN consists of 14 PBXs, serving 3120 phones. These phone lines are divided into 2787 analog and 333 digital lines. The already known distribution of number of lines at all 14 PBXs is shown in Fig. 3.1 and the basic view of Politecnico's telephone network is shown in Fig. 3.2. All 14 PBXs are connected to the Group Switch (GS) located at Central Node. The Group Switch provides connection between all 14 PBXs and the external telecom network.

The maximum capacity of each PBX is around 500 phones with flexibility to increase the number of phones by inserting new line cards. Each line card can handle up to 16 phones. There is a different line card for analog or digital phone lines with different power consumption values.

3.2.2 VoIP system

We consider as a reference the VoIP system deployed at Istituto Superiore Mario Boella (ISMB), a laboratory close to Politecnico campus. Its architectural view is shown in Fig. 3.3. Interestingly, the architecture is simple and does not need large scale infrastructure like in the case of the traditional phone system. We can observe 4 Power over Ethernet (PoE) switches (3 switches with 24 ports and one with 48 ports) and a PC with

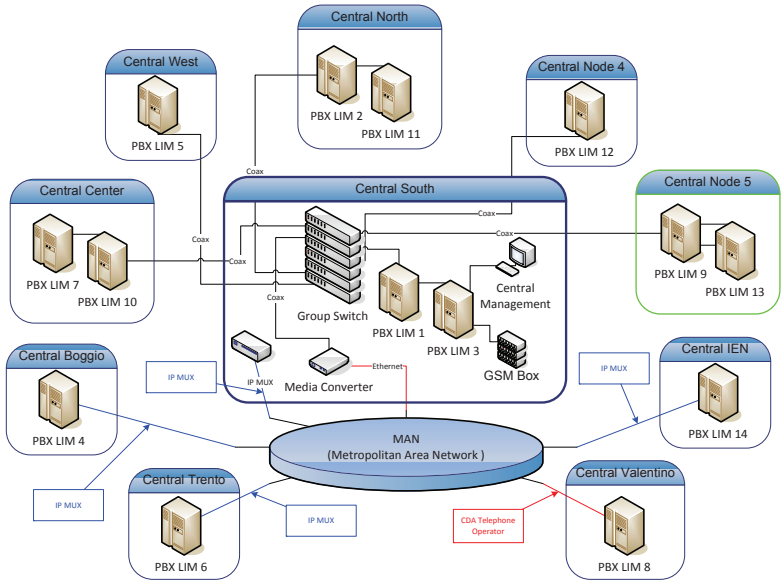

Figure 3.2. Interconnection of Politecnico's PBXs

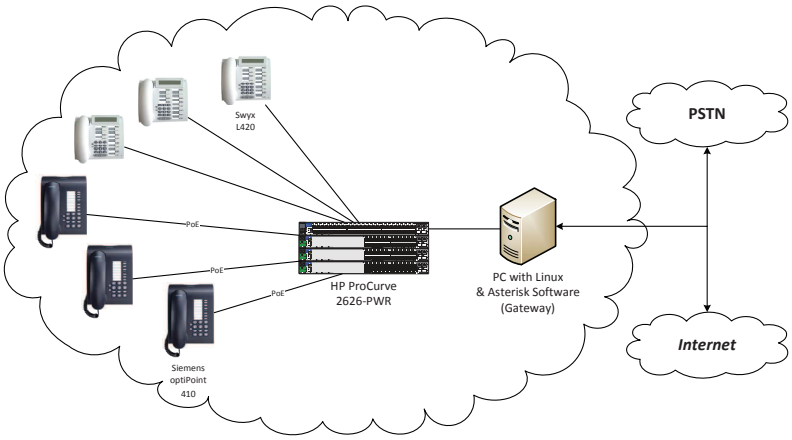

Figure 3.3. VoIP architecture at ISMB

Asterisk software[1] to act as a communication server. The referenced architecture serves around 120 users with the ability to make phone calls, and also provide data services. This means that the VoIP architecture utilizes shared infrastructure of Ethernet that provides data services, which seems to be a good option in terms of energy saving.

[1] http://www.asterisk.org/

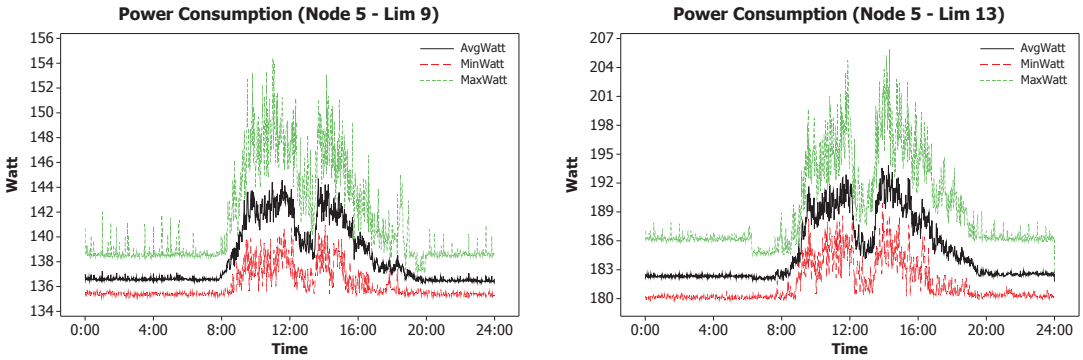

Figure 3.4. PSTN: Power consumption - Weekdays

3.3 Measurement and Modeling

The power consumption values of PBX or VoIP switches are measured locally at regular intervals by local devices. Later, the measurement data is transmitted to a central server (youMeter) through a wireless transmitter connected to the measurement device. The collected data is then stored in a database for statistical analysis. This measurement activity involves the collection of certain parameters like Real/Reactive Power, RMS Voltage/Current.

3.3.1 PBX measurements

Two PBXs (LIM 9 and LIM 13) are under power measurement[2] present at central node 5. The monitoring of power consumption of these two PBXs collected over the span of two weeks ($4-15/04/2011$) is shown in Fig. 3.4; Fig. 3.5 shows the power consumption during weekends. The three curves with green, black and red color represent, respectively, the maximum, average and minimum power consumption samples for 24 hours collected over a span of working days and weekends. Power consumption is constant during weekends and its around 137 Watts in case of LIM 9, even if there is no user activity. During weekdays, and when the system is carrying phone calls, the system consumes around 10% more of the overall consumption. Note that the measurements are vendor specific (Ericson model in our case), so the results we provide may not be representation for every scenario, but rather could be generalized to provide good estimates. For example, similar experiments were conducted over *AASTRA*'s new generation PBX, where empty PBX (no phones or line cards connected) resulted with 66 Watts of consumption.

[2]Power measurement of PBX includes power consumed by UPS/transformer that transforms 220 Volt to 48 Volt DC (which is part of PBX).

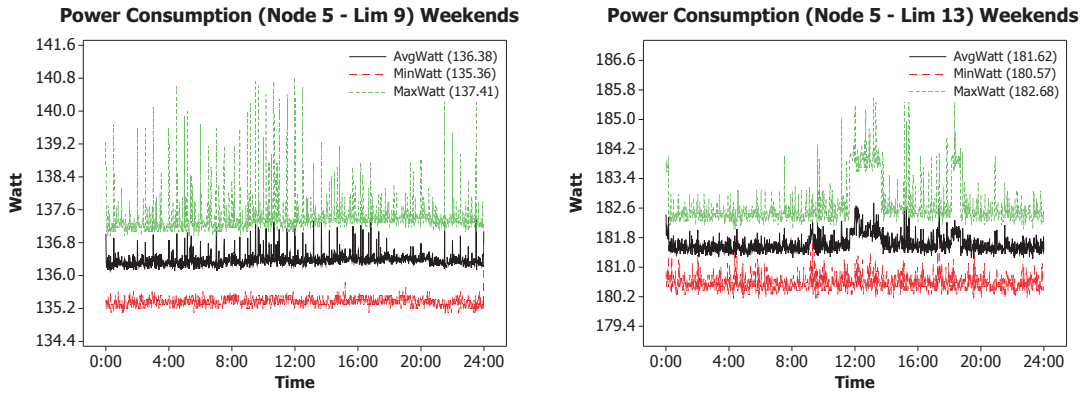

Figure 3.5. PSTN: Power consumption - Weekends

3.3.2 PSTN power modeling

We extrapolate a simple mathematical model from the acquired real measurements to estimate the per user consumption. Using this estimate, one can approximate an overall consumption for a given number of users in any campus or corporate scenario. Considering minimum power consumption when the system is in idle state, we assume that the power consumed by analog/digital line cards, as well as phones, is approximately the same;

$$P_{PBX} + X \cdot P_{lc} + K \cdot P_{ph} = P_{TOT_{PBX}} \tag{3.1}$$

The above expression divides total power consumption as the sum of minimum constant PBX power consumption, P_{PBX}, plus the power consumed by line cards P_{lc} and phone lines P_{ph}. X and K represent respectively the number of line cards and phone lines connected to specific PBX. Based on the number of interfaces and phone lines connected to each PBX, we obtained the following two equations,

$$\begin{cases} LIM9, \quad P_{PBX} + 11P_{lc} + 158P_{ph} = 135.10\,\mathrm{W} \\ LIM13, \quad P_{PBX} + 18P_{lc} + 242P_{ph} = 180.06\,\mathrm{W} \end{cases} \tag{3.2}$$

In order to solve the equations we need to get the value of one of the variables. For that purpose, we decided to directly measure the power consumption of phones lines by adding ten new phones to the PBX and observe their effect on the PBX power consumption. Figure 3.6 shows the results of such experiment during night time. We observe the average power consumption by single phone user (ON-hook) to be $P_{ph} = 0.53\,\mathrm{W}$.

Later on, the same ten new phone lines were made to call each other during the weekend in order to measure the on-call power consumption. while the power consumption during calls is 2.28W

By solving (3.2), we get power consumption of line card and PBX;

$$\begin{cases} LIM9, \quad P_{PBX} + (11)\,P_{lc} = (135.10 - 158 * 0.53) \\ LIM13, \quad P_{PBX} + (18)\,P_{lc} = (180.06 - 242 * 0.53) \end{cases} \tag{3.3}$$

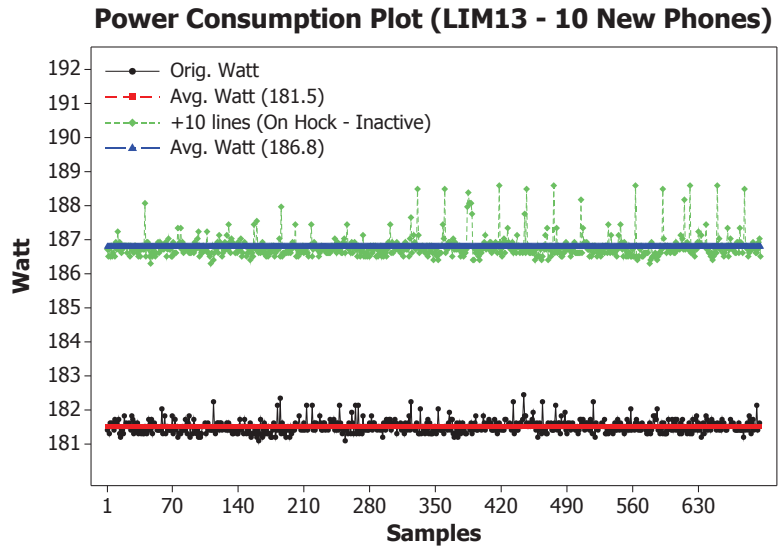

Figure 3.6. PSTN: Effect of adding 10 new lines

Therefore, we obtain;

$$P_{PBX} = 50.6686\,\mathrm{W}, \quad P_{lc} = 0.0629\,\mathrm{W} \tag{3.4}$$

3.3.3 VoIP measurements

Two switches (namely SW-2 and SW-3 which are used exclusively for connecting VoIP phones) with model HP Pro-Curve are kept under power measurement in the VoIP system. The distribution of the number of phones on SW-2 consists of 12 non-PoE[3] phones and 5 PoE phones. SW-3 has 9 non PoE phones and 12 PoE phones. The monitoring and collection of power consumption data is done as mentioned above. The 24 port switch under measurement can provide up to 370 watts of power to PoE devices, which in turn means all 24 ports deliver at an average of $370/24 = 15.4$ watts per port. Figure 3.7 shows the actual power consumption behavior of PoE switches observed over the same duration of two weeks ($4 - 15/04/2011$). Differently from the PSTN system, we observe a constant power consumption, which is independent on user activity (notice the range of the y-axis of the plot).

3.3.4 Formulation

A similar criterion is used to extrapolate a simple mathematical model to estimate per user consumption.

[3]Power is drawn from an adapter locally connected to the VoIP phones.

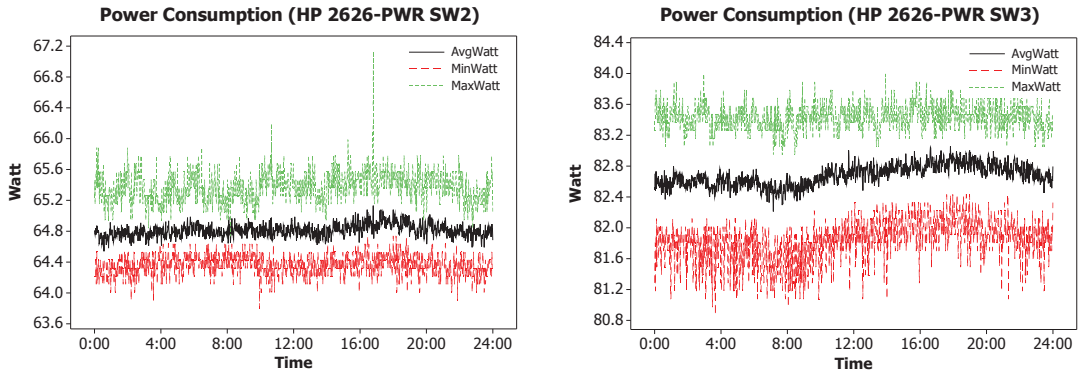

Figure 3.7. VoIPs Power consumption

$$P_{SW} + K_1 P_{PoE} + K_2 P_{\overline{PoE}} = P_{TOT_{VoIP}} \tag{3.5}$$

P_{SW} is the minimum constant power consumption of the Ethernet switch, K_1 and K_2 represent the number of PoE and non-PoE phones. Lastly, P_{PoE} and $P_{\overline{PoE}}$ represent the power consumption figure of PoE and non-PoE phones. Assuming non-PoE phones do not require power from the switch, we have $P_{\overline{PoE}} \cong 0$

$$\begin{cases} SW-2, & P_{SW} + 5P_{PoE} = 64.81\,\text{W} \\ SW-3, & P_{SW} + 12P_{PoE} = 82.67\,\text{W} \end{cases} \tag{3.6}$$

Solving the equations, we have $P_{SW} = 51.88\,\text{W}$ and $P_{PoE} = 2.6\,\text{W}$. In order to verify the results, we conducted another experiment in which SW-3 offered connectivity to only 16 PoE phones and SW-2 to only 19 non-PoE phones. The results showed that SW-2 and SW-3 gave 53.2 W and 90.7 W respectively. These numbers complied with the results derived from previous calculations.

3.4 Analysis

Using the formulation mentioned above for both the architectures, rough estimate of the power consumption values for a generic PSTN system is computed from,

$$P_{PSTN} = \left\lceil \frac{N_{T_{lines}}}{N_{users/PBX}} \right\rceil P_{PBX} + N_{T_{lines}} \cdot P_{ph} \tag{3.7}$$

where $N_{T_{lines}}$ is the number of phone lines and $N_{users/PBX}$ is the number of phones a PBX can host (500 in our case). Similarly, P'_{VoIP} is the equivalent estimated power consumption of the VoIP system:

$$P'_{VoIP} = \left\lceil \frac{N_{T_{lines}}}{N_{ports/switch}} \right\rceil P_{SW} + N_{T_{lines}} \cdot P_{PoE} \tag{3.8}$$

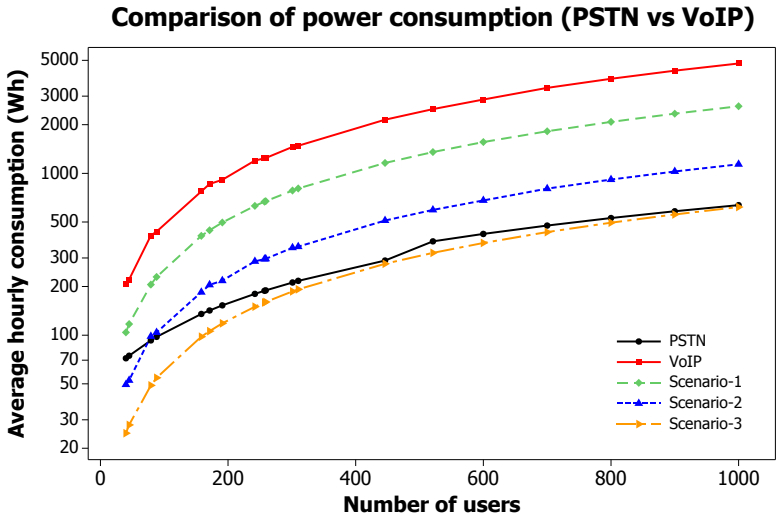

Figure 3.8. PSTN and VoIP energy consumption

where $N_{ports/switch}$ is the number of ports available on a specific switch (24 in our case). Note that (3.7) and (3.8) represent a lower bound on power consumption since they assume to fill completely a switch before adding a new one.

3.4.1 Energy saving schemes

Current Scenario - The power consumption of PBX and equivalent VoIP architecture for certain number of users is computed using (3.7) and (3.8). The results consider the total average consumption of the system during 24h period of time. They are reported with the solid black and red lines in Fig. 3.8 (*notice the ylog-scale*); They clearly indicate that VoIP solution consumes more power for the considered architectures, but this is strongly dependent on the devices used for implementing the phone infrastructure. Yet, VoIP systems have the flexibility to be made energy efficient by adding some sort of energy-wise schemes. For example, one possibility to save energy would be to put VoIP phones into sleep mode when there is no user activity using the standard Wake-on-LAN (WOL) Ethernet feature; a second possible space to save energy would be during inactivity period of ongoing call [17]. In the following text, we describe various scenarios to save energy along with their results.

Scenario-1 - We use the already deployed LAN connection infrastructure along with VoIP phones that are powered through Ethernet, thus eliminating the need to add and power up extra Ethernet switches ($P_{SW} = 0$). The power consumption is shown with a green line in Fig. 3.8.

$$P_{VoIP_1} = N_{T_{lines}} P_{PoE} \tag{3.9}$$

Scenario-2 - Smart VoIP is the option where we put both the VoIP enabled Ethernet switches and phones into sleep mode when there is no user presence. Assuming 8 hours of

working time, 5 days per week. P_{VoIP_2} is calculated using the values of P'_{VoIP} from (3.8) as:

$$P_{VoIP_2} = P'_{VoIP}\,(8/24)\,(5/7) \tag{3.10}$$

Scenario-3 - The combination of VoIP phones and already deployed Internet connection infrastructure with power saving scheme defined in (3.10). The power consumption P_{VoIP_3} is computed.

$$P_{VoIP_3} = P_{VoIP_1}\,(8/24)\,(5/7) \tag{3.11}$$

Scenario-4 - Use of softphones, i.e., devices connected to the user's PC that consume virtually no power, and existing Internet connection to set up voice communication. We assume softphones do not consume any power consumption. In this case, power consumption is zero ($P_{SW} = 0$, $P_{PoE} = 0$).

VoIP at Politecnico

In practice, it may not be possible to offer all user a softphone. For example, some users may be reluctant to have a softphone to be connected to their laptop. Given certain number of users, an interesting question is then to find how many VoIP phones can be accommodated so as to not exceed the power consumed by the equivalent PBX system deployed at our campus. The rest of the users are accommodated with softphones that consume no power. We assume to use the same LAN switches as in (3.9). The results of such distribution shown in Fig. 3.9 show that about 1/3 of lines could be real VoIP phones at each switch.

$$P_{VoIP_{ph}}\,(x) \leqslant P_{PSTN}$$
$$P_{VoIP_{ph}} = \begin{cases} 2.6\,\mathrm{W} & VoIPphone : (x) \\ 0\,\mathrm{W} & Softphone : (N_{T_{lines}} - x) \end{cases} \tag{3.12}$$

Comparison

Figure 3.8 illustrates graphically the power consumption of PBX and the equivalent VoIP architectures along with the comparison between energy saving VoIP scenarios. Results show that VoIP solutions are more energy expensive, unless aggressive power saving schemes are in place. For example, noticeable difference in power consumption is due to the fact that scenario 3 uses power consumption of VoIP phones only (not the switches) and sleep mode is aggressively exploited.

3.5 Conclusion

We have identified the main lines of intervention to introduce energy efficiency in two architectures for telephony, namely a traditional architecture and a VoIP solution. The analyzed results suggest that the VoIP architecture is energy hungry and thus requires,

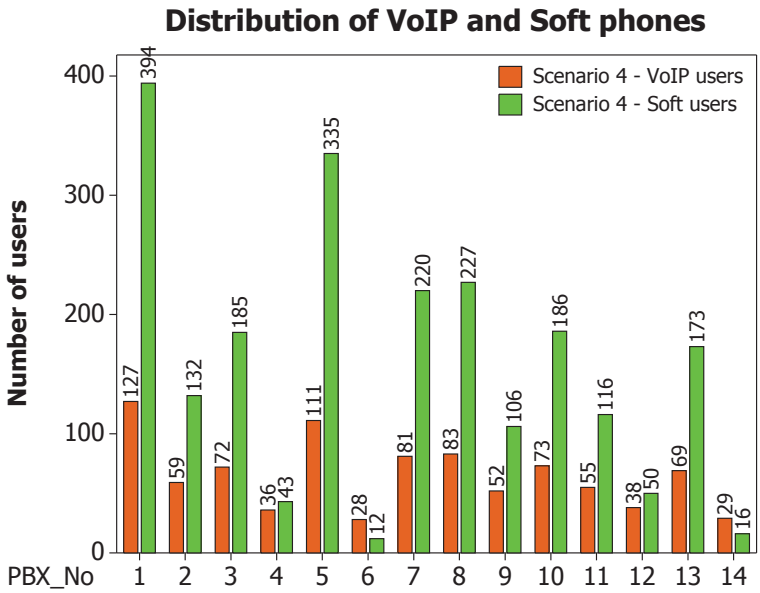

Figure 3.9. PBX equivalent no. of VoIP/Soft phones

to be competitive in terms of energy efficiency, the implementation of aggressive energy saving schemes. Newer generation of PBX and switches might lead to completely different results. To have a deeper analysis, we intend to differentiate the power consumption of analog and digital phones and compare the results with newer generation of PBXs, switches and VoIP phones. For this purpose we are currently measuring power consumption of *AASTRA*'s new generation PBX, which are replacing Ericson's PBXs installed in our university campus.

Chapter 4

Modeling The Interaction between TCP and Rate Adaptation

4.1 Introduction

The increase of energy cost and the environmental concerns related to the intensive use of energy have pushed, in recent years, the research on new solutions and technologies for reducing energy consumption in all the fields of Information and Communication Technology (ICT). Indeed, the increase of the traffic generated by a larger population of users, often running bandwidth intensive applications, combined with the massive diffusion of smart phones, has enlightened that ICT [1] is consuming a huge amount of energy, and drastic actions are needed to both reduce operational costs of the networks and make sustainable future communications.

Among the segments of the network, the access networks have shown to be the main responsible for energy consumption; this is due to the combination of two factors. On the one side, the increase of link rates needed to sustain the traffic has raised the energy consumption of links; on the other hand, at the periphery of the network, the number of networking devices is huge. Most of the traditional networking devices present a consumption profile that is very little load proportional, meaning that the devices consume, at low load or when idle, a large fraction of what is consumed at full load. In other terms, the energy cost of running a network depends on the capacity deployment rather than on the actual capacity usage. This has brought to the proposal of a number of solutions based on the use of some low consuming "sleep mode" that is entered by the devices when there is no traffic, [18]. For instance, the IEEE Energy Efficient Ethernet 802.3az standard aims at improving efficiency of the traditional Ethernet protocol, that requires that the transmitter and the receiver remain continuously active even if there is no data to send, leading to waste of energy (about 0.5W for 1GBase-T and 5W for 10GBase-T), by entering low consuming sleep modes when there is no traffic. The adoption of the new standard allows to save an amount of energy that, in the best case, i.e., when the transmission consists of large data bursts and long silence gaps; in these case, the consumption becomes roughly proportional to average link utilization [19].

Sleep mode support provides quite significant reduction of energy consumption in scenarios in which the traffic sources are very bursty with long idle periods. However, if the traffic offered to a link is not too bursty, even if far below the link capacity, the use of sleep mode becomes mostly ineffective [20]. In such scenarios, a quite promising approach to save energy is "rate adaptation". Based on the fact that less energy is needed when lower transmission rates are used instead of high rates, the idea consists in adjusting the data transmission rate (or capacity) according to traffic dynamics: when traffic increases, the capacity is increased as well, but when traffic is low, low transmission rates, that consume less energy, are used.

There is a possibility, however, that the rate adaptation mechanisms implemented in the router negatively interact with other control mechanisms implemented at upper layers of the protocol stack. In particular, we consider here the possible interaction between the rate adaptation control loop and the loop implemented by TCP for congestion control [21]. The scenario is quite important, being TCP the transport protocol that carries most of the Internet traffic. The control loop defined in TCP, i) linearly increases the amount of traffic injected in the network when there seems to be enough bandwidth and, ii) performs multiplicative decrease upon congestion indications, e.g., packet loss. On the opposite side, the rate adaptation scheme at links has a control loop that is driven by the amount of traffic injected in the network, as estimated by the queue size at links. Hence, the data rates are adapted based on the offered load. If not properly designed and tuned, the two control loops can negatively interact. For example, a rate reduction caused by the rate adaptation algorithm might induce TCP to reduce injected traffic; in its turn, the TCP traffic reduction results in a reduced offered traffic and might trigger further rate reductions at the links. This interaction might lead, finally, to bad performance and end-users Quality of Experience (QoE) deterioration.

Aim of this work is to investigate how these two control loops (rate adaptation and TCP congestion control loops) interplay with each other and to look for parameter settings that lead to high energy saving without compromising QoS. To understand the interplay of the two control loops, we propose a fluid model analysis. We use a classical model of the TCP Additive Increase-Multiplicative Decrease (AIMD) congestion control and combine it with the model of two different queue-length driven rate adaptation schemes. Several experiments are conducted through simulations to validate the results obtained from the model. A steady state analysis and a deep investigation of the impact of parameter setting are then performed through the proposed model.

Results show that, if not properly tuned and configured, rate adaptation controllers can lead the system state to a configuration where links go to the minimum offered capacity, severely impairing throughput. We derive the conditions to avoid this unfortunate scenario, and, furthermore, we show that accurate parameter tuning is required to properly minimize transients during which the system moves from low bit rate states to the maximum bit rate. Transients can last orders of seconds, thus sensibly impairing the download time of short lived flows. We, finally, show that, when traffic is present, savings are possible only if the relation between energy consumption and current rate follows a super-linear law.

4.2 Related work

This section describes the main approaches proposed for energy efficient transmission, followed by an overview of related work done to evaluate the impact of rate adaptation schemes on the network performance and its interaction with TCP.

Among the first works that appeared in the literature to introduce rate adaptation at links, [19] proposes to use a queue length based control scheme with a predefined set of discrete transmission rates along. Similar approaches are investigated through simulation in [22] and [23] that suggest that, to improve TCP throughput, a rate adaptation scheme should work as traffic pacer given that the network utilizes small buffers. Whereas [24] reports real-time measurements obtained from an hardware based Adaptive Link Rate (ALR) system and suggests that the link rate switching time to vary from 10 ms to 100 ms with link rate switching power consumption in the range of 1 to 2 Watts. Furthermore, the paper proposes a step-down technique to switch link rates in discrete small steps rather than one large step to ultimate target rate; this results to be efficient in terms of switching time and power consumption. None of the previous works includes sources that perform congestion control.

From the perspective of modeling of rate adaptation scheme and evaluation of its performance, in [25], a Markov model is used to describe the ALR scheme. System performance is evaluated through the computation of packet delay and time spent in low data rates. Traffic is assumed to follow a Poisson process, and, thus, congestion control is not considered.

There is a considerable literature see, for example, [26], [27], [28] regarding the modeling of TCP's congestion control. However, to the best of our knowledge, no work addresses the detailed interaction of rate adaptation with TCP congestion control mechanism from a mathematical modeling perspective. Only [29] presents some preliminary investigation of the interplay between rate adaptation scheme and TCP congestion control mechanism, through simulation. The results pinpoint the negative impact on QoS in the presence of capacity scaling algorithms that reduce power consumption.

Our analysis is based on a fluid model approach, and it gives us deep insight into the interaction between the two control mechanisms, in terms of both the steady state as well as the transient behavior of the system in various parameter settings. A preliminary version of our work has been presented in [30]. In this work, we extend the model to consider different scenarios, complete the parameter sensitivity and steady-state analysis, and we also estimate achievable energy savings.

4.3 Fluid model

The setting up of a rate adaptation scheme requires the modeling of TCP behavior. Hence, a set of coupled Delay Differential Equations (DDEs) are derived to describe the basic TCP evolution. We build upon the differential equations proposed in [28] that depict the well known TCP congestion control mechanism by modeling it as a Additive Increase-Multiplicative Decrease (AIMD) algorithm. The solution of the model provides the time

Table 4.1. Parameters and variables

Description	**Symbol**	**Value**	**Unit**
TCP congestion window	W	$1:150$	packets
Queue length	Q	$0:100$	packets
Estimated queue length	$\bar{Q}$	$0:100$	packets
Capacity	C	$100:1000$	packets/sec
Initial capacity	C_0	200	packets/sec
Capacity controller threshold	Q_{oc}	$5:25$	packets
Loss controller threshold	Q_{ol}	20	packets
Propagation delay	R_o	0.02	seconds
Delay of loss indication	τ	RTT	seconds
Round Trip Time	RTT	$R_o + Q/C$	seconds
$\bar{Q}$ sampling interval (model)	δ_m	$1/1000$	seconds
Control interval (simulator)	δ_s	$1/10000$	seconds
Loss scaling factor	K_l	$1/80$	–
Capacity scaling factor	K_c	$1:5000$	–
Smoothing factor	α	$0:1$	–
Loss rate	L	$0:1$	–

evolution of TCP as well as the evolution of network parameters such as window size, queue length and capacity. The framework uses the fluid based model, wherein the parameters evolve as a continuous process.

We consider a generic scenario which consists of a TCP sender and a receiver connected through a router as shown in Fig. 4.1. The TCP sender, namely the Server, sends a file as a bulk data transfer. Since on the backward path only acknowledgements are being sent, we model the rate adaptation mechanism on the forward path queue only. Table 4.1 reports the list of variables used.

4.3.1 Rate adaptation

We describe here the model of the rate adaptation algorithm, i.e., of the strategy with which the transmission rate is adapted to the offered traffic. We consider schemes where the estimation of the current offered load is obtained from the measurement of the actual queue size as typically proposed in the literature. In particular, we focus on two slightly different controllers: in the first one, the rate is adapted to the *instantaneous queue length*, in the second one a *weighted average of the queue length* is used as control parameter.

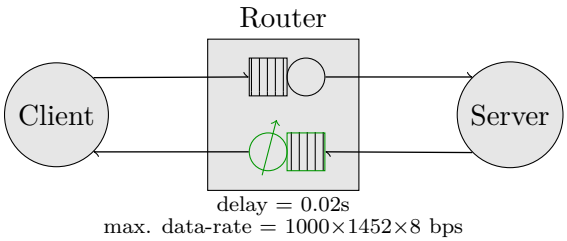

Figure 4.1. The considered topology.

Rate controller I

The idea is to design a simple rate controller that adapts the link rate according to the *instantaneous queue length* evolution. In other words, the service rate or capacity, $C(t)$, increases (decreases) if the instantaneous queue length grows above (below) a threshold Q_{oc}. Based on the value of threshold, the controller can quickly move and operate at maximum capacity and achieve minimal energy saving while maintaining high QoS (low values of Q_{oc}); or it can reduce energy consumption at the expense of possibly compromising QoS (high values of Q_{oc}). In control theory, this results in a simple proportional controller. The rate controller can be modeled by:

$$\frac{dC(t)}{dt} = K_c(Q(t) - Q_{oc}) \quad for \quad C_{min} \leqslant C(t) \leqslant C_{max} \tag{4.1}$$

where K_c represents the capacity scaling factor and Q_{oc} the queue length threshold, with the rate controller being bounded between minimum and maximum capacity constraints.

Rate controller II

In this case, we consider a second type of rate controller which adapts the rate to the *average queue length*, and, in particular to the Exponentially Weighted Moving Average (EWMA) queue length. The purpose is to possibly make the controller more stable by adapting the rate to queue length variations more slowly. In control theory, this results in an proportional integral controller.

The EWMA smooths the variations of the queue length through a smoothing factor, α, with $0 \leqslant \alpha \leqslant 1$. A queue length sampling interval equal to δ_m is used. The value of δ_m is fixed and it is set to $1/C_{max}$, which means the value is updated at almost every packet transmission. Hence, as shown in [28], the equation describing the evolution of EWMA queue length is:

$$\frac{d\bar{Q}(t)}{dt} = \frac{\log(1-\alpha)}{\delta_m}\bar{Q}(t) - \frac{\log(1-\alpha)}{\delta_m}Q(t) \tag{4.2}$$

where $\bar{Q}(t)$ represents the average queue length.

Therefore, the rate controller based on estimated queue length can be described by:

$$\frac{dC(t)}{dt} = K_c(\bar{Q}(t) - Q_{oc}) \quad for \quad C_{min} \leqslant C(t) \leqslant C_{max} \tag{4.3}$$

4.3.2 TCP model

The TCP model focuses on the evolution of the congestion window size, that we denote by $W(t)$. We explicitly build on the work presented in [28] to which we refer the reader for more details. We consider the congestion avoidance mechanism of TCP, since this is what mainly determines the performance of long-lived TCP flows. Furthermore, since we consider, as typical, a low loss regime, the time-out behavior is neglected. However, the model could be extended to include both the initial slow start and time-out loss recovery.

The evolution of the congestion window size in congestion avoidance consists of two parts: the additive increase of the congestion window by one MSS (Maximum Segment Size) every RTT, and the multiplicative decrease in case of loss indication. In our case, only losses detected by Triple Duplicate (TD) ACKs are considered. The TCP sender receives the loss indication after τ seconds, that is roughly one RTT. Thus, a loss indication received at time t means that a loss actually occurred in the router at time $t - \tau$.

Similar to [28], we consider a scenario in which a Random Early Detection (RED) queue management algorithm is used in the router. Thus, losses occur according to the RED mechanism that depends on the queue length. Further description about RED queue management algorithm can be found in [31]. By setting RED parameter appropriately, it is possible to approximate also drop-tail queue management.

Let the loss rate at time t be denoted by $L(t)$. Given TCP delay in detecting losses, at time t, TCP perceives a loss rate equal to $L(t-\tau)$. The number of losses is thus given by the product of the loss rate and the number of packets on the fly, that, in its turn, is given by ratio between the congestion window size and the round trip time; the number of losses is thus equal to $L(t-\tau)\frac{W(t-\tau)}{RTT(t-\tau)}$.

Under the fluid model based framework, the RED queuing policy can be modeled by making losses proportional to the average queue length, according to the following,

$$L(t-\tau) = K_l(\bar{Q}(t-\tau) - Q_{ol}) \quad for \;\; 0 \leqslant L(t) \leqslant 1 \tag{4.4}$$

where $L(t-\tau)$ represents the loss rate, and K_l is the loss rate scaling factor that defines the growth of the drop probability function in the RED queue. Q_{ol} is the loss threshold thereafter the RED queue starts to drop packets. Similarly, $\bar{Q}(t-\tau)$ represents the current average queue length at time $t - \tau$.

The evolution of the round trip time takes the following form,

$$RTT(t) = R_o + \frac{Q(t)}{C(t)} \tag{4.5}$$

where R_o is the total propagation delay and the fraction $\frac{Q(t)}{C(t)}$ corresponds to the queuing delay measured in seconds. Similarly, $RTT(t-\tau) = R_o + \frac{Q(t-\tau)}{C(t-\tau)}$.

Combining the above equations, the TCP congestion avoidance behavior can be modeled by the following DDE,

$$\frac{dW(t)}{dt} = \frac{1}{RTT(t)} - \frac{1}{2}W(t)\left(L(t-\tau)\frac{W(t-\tau)}{RTT(t-\tau)}\right) \tag{4.6}$$

where the derivative $\frac{dW(t)}{dt}$ represents the evolution with time of the window size measured in packets.

The differential equation that describes the queue length evolution can be written as the difference between the arrival rate of packets at the queue, represented by the term $\frac{W(t)}{RTT(t)}$, and the rate at which the packets are served, represented by the capacity $C(t)$,

$$\frac{dQ(t)}{dt} = \frac{W(t)}{RTT(t)} - C(t) \quad for \quad 0 \leqslant Q(t) \leqslant Q_{max} \tag{4.7}$$

The differential equation is bounded between 0 and Q_{max}, so that the serving of packet occurs when the queue length is greater than zero.

4.4 Model validation

In this section we first describe the kind of results that can be obtained by the model and provide a first analysis of them; we then validate the analytical model by comparison against simulation results.

4.4.1 Time evaluation

The system of differential equations, (4.1), (4.2), (4.6), (4.7), is solved numerically to get the evolution in time of the mentioned parameters. To derive a first set of results, we choose the model parameters as follows: the initial starting capacity is $C_0 = 200$ packets per second (pps); $C_{min} = 100$ pps to $C_{max} = 1000$ pps. The capacity controller threshold parameter is set to $Q_{oc} = 5$ packets, while, the RED loss threshold is $Q_{ol} = 20$ packets. The queue length is bounded in the region of $Q_{min} = 0$ to $Q_{max} = 100$ packets. The loss scaling factor $K_l = 1/80$, which is equivalent to RED dropping starting at $Q_{ol} = 20$ packets up to $Q_{max} = 100$ packets with the packet discard probability function taking the value of 1, when the queue length is equal to Q_{max}. The propagation delay is fixed to $R_o = 0.02$ s and the loss delay indication factor to $\tau = RTT(t)$.

Figure 4.2 illustrates an example of the evolution of the parameters based on rate controller I. From top to bottom, we report the capacity, window size, instantaneous queue length, average queue length, and the loss rate. Observe that the capacity, after some transient period, achieves, as desired, the maximum possible value. However, about 60 s are required to complete the initial transient. The congestion window size, as well as the queue length, continuously vary in time and it oscillates, as is expected from long-lived TCP flows mimicking the classic congestion avoidance behavior. Last two figures represent the smoothed version of queue length (EWMA queue length) and RED loss rate evolution. As expected, the queue length oscillates, while losses occur in bursts.

4.4.2 Validation

To validate the rate adaptation models, a simple network topology is simulated through a discrete event simulator developed in OMNeT++. We consider the same topology as the

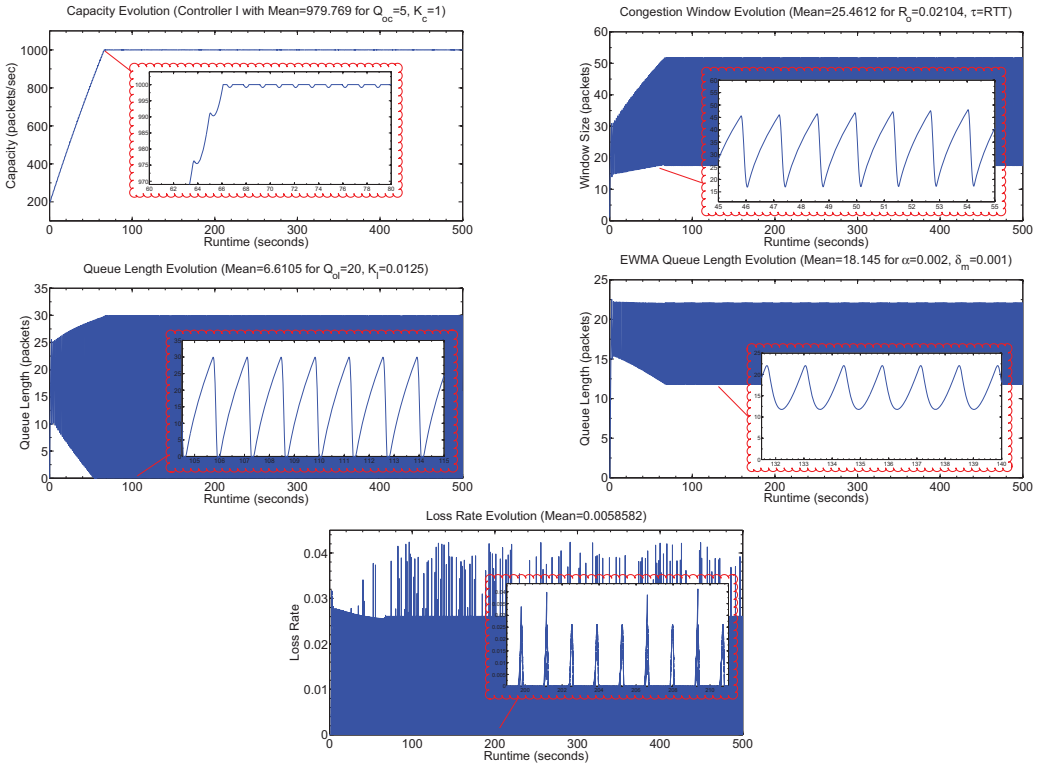

Figure 4.2. Example of evolution of the parameters based on the fluid model.

one in Fig. 4.1. The two nodes, namely the client and the server, communicate with each other using a TCP Reno connection with default TCP parameters. The communication happens in sessions and, during a session, the client sends a FTP-like request for a file of a given size to the server and waits for the reply to complete before sending a new request. The simulation ends if either simulation time limit is reached, or if the number of requests gets complete. In our case, we chose the file size to be long enough so that only one transfer is considered, and the simulation ends due to time limit constraint, which is set to 500 s. Simulation parameters are chosen as those used earlier for the analytical model. Thus, nodes are connected with link capacity of 1000 pps (packet size of $MSS = 1452$ bytes) and propagation delay equal to 0.02 s. The downlink, represented with the path from server to client, is heavily loaded, whereas the uplink path is lightly loaded as it transports only ACKs. A RED queue management algorithm is used with lower and upper bound limits of 20 and 100 packets and linear growth of the packet discard probability from 0 to 1. The router along the path implements the rate adaptation algorithm at queues as detailed below.

Rate Controller I

To validate the fluid model for rate controller I presented in Section 4.3.1, we have implemented the equivalent rate control mechanism on the router using the OMNeT++

simulator. The implementation of rate control mechanism on link layer is described by the following discrete time equation:

$$C(n+1) = C(n) + K_c^{'}(Q(n) - Q_{oc})\delta_s \tag{4.8}$$

The parameters are chosen similar to the mathematical model described earlier. Furthermore, the capacity is bounded in the region of $C_{min} = 100$ to $C_{max} = 1000$ pps with the initial starting capacity set to $C_0 = 200$ pps. The δ_s in the simulator is the time interval wherein the controller checks the queue condition and updates the rate, i.e., the discrete time used to emulate the continuous time in the fluid model. We have chosen $\delta_s = 0.1$ ms to be small enough to react adequately towards the changes in queue length. This action can be compensated if the value of $K_c^{'}$ is chosen to be high enough. Indeed, the product of $K_c^{'}\delta_s$ is equivalent to the K_c parameter used in the fluid model.

Simulation results - long flow Since the differential equation solution neglects a number of TCP mechanisms (slow start phase, time-outs, and so on), while the simulator has a detailed model of the TCP protocol, we expect to see some different behavior, especially during the initial transient phase due to TCP slow start. Furthermore, the differential equation model provides solution as continuous increments, while the simulator works in discrete steps.

Figure 4.3 illustrates the behavior of rate controller evolution implemented using OMNeT++ simulator for different values of $K_c^{'}\delta_s$ along with its comparison with fluid model based rate controller with proper value of K_c. Overall, the comparison of results shows that the fluid model is very accurate, and follows closely the trail of capacity evolution obtained through simulations. For $K_c = K_c^{'}\delta_s = 100$, we observe the impact of slow start in the simulator allows to quickly terminate the initial transient.

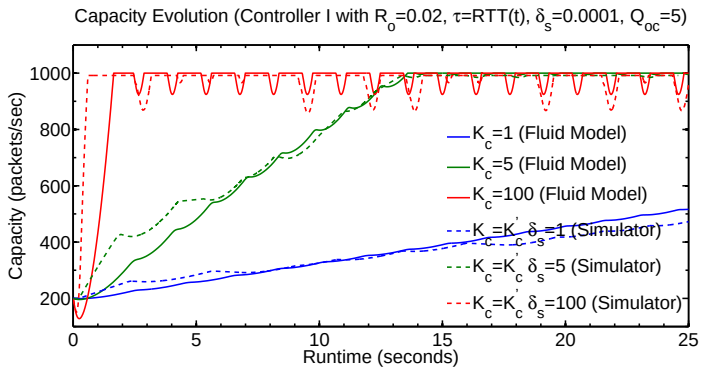

Figure 4.3. Capacity controller based on instantaneous queue length.

Simulation results - short flow We now choose the file size to be small, such that the client sends only a single request for a file and waits for a reply to complete. Thus, unlike the previous case the request gets complete before the end of simulation time limit. However, like before the connection gets closed due to simulation time limit constraint.

The Fig. 4.4 illustrates the behavior of rate controller evolution implemented using OMNeT simulator for different values of K_c along with its comparison with fluid model based rate controller for short flow, more specifically a 10 MByte of data transfer. Observe that the rate controller pushes its capacity to C_{min}, once the file transfer completes. Again, the results obtained through simulations validate the model quite well.

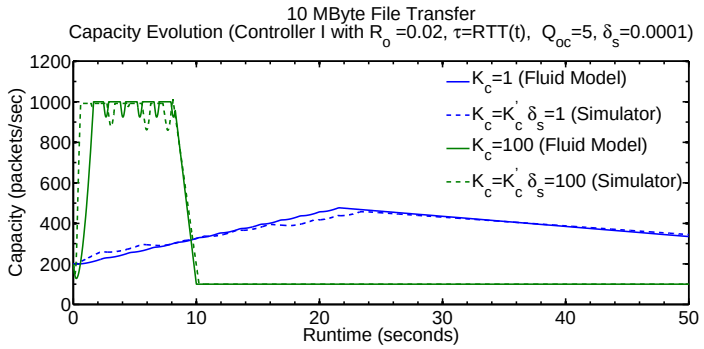

Figure 4.4. Comparison between simulator and fluid model based rate controller I for short flow

Rate Controller II

We repeat the experiments described in previous section with the same parameters but for the rate controller that evolves based on the EWMA queue length. The evolution of estimated queue length is computed using the following expression,

$$\bar{Q}(n+1) = (1-\alpha)\bar{Q}(n) + \alpha Q(n) \tag{4.9}$$

The parameter α represents the smoothing factor ranging between 0 and 1, where the value $\alpha = 1$ represents the case of no averaging, i.e., of instantaneous queue length. Hence, (4.8) becomes,

$$C(n+1) = C(n) + K_c^{'}(\bar{Q}(n) - Q_{oc})\delta_s \tag{4.10}$$

Simulation results Figure 4.5 shows the behavior of rate controller based on EWMA queue length and its comparison with the fluid model. Also in this case the simulation results show that the model captures quite well the dynamics of capacity evolution for different values of K_c. The usage of the average queue length has two impacts on the evolution: i) the capacity controller changes are smoother, and ii) the initial transient the system takes to start increasing the offered capacity is longer than when using a simpler proportional controller. We discuss difference further in the next section.

Overall, the simulations show that the model is quite accurate. As such, in the remaining part of the work, we use only the model to study the properties of the system.

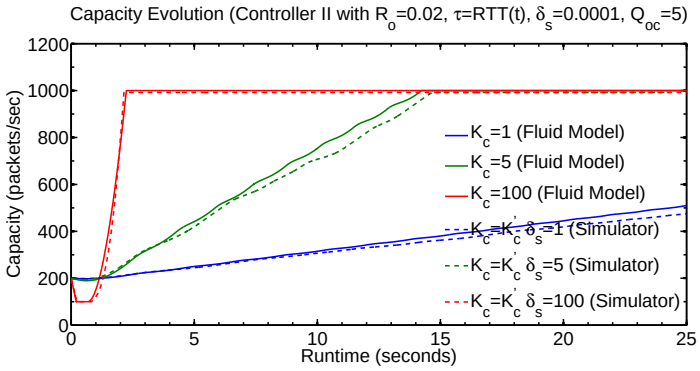

Figure 4.5. Capacity controller based on average estimated queue length.

4.5 Comparison between the controllers

In this section we make a direct comparison between the performance of the two rate adaptation controllers. To do so, we run the analytical models for 200 s and we take the average value of the capacity in the whole period; the computed average includes the transient phase as well as the steady state phase. Figure 4.6 shows the comparison of the results for the two controllers, for a wide range of values of K_c.

The results show that the rate controller based on estimated queue length yields to smoother and more stable results with respect to the rate controller based on instantaneous queue length. While this is quite intuitive, it is important to emphasize that this smoother behavior is achieved at the cost of larger amount of losses (see Fig. 4.7) and slower response to queue length variations, that might, in general, in their turn deteriorate the QoS (higher delay variance, possibility of multiple losses and therefore difficulties in reacting to losses, and so forth). However, the differences are not that large in absolute terms. Finally, for very large values of K_c, the controller I suffers from much larger oscillations than controller II (see also Fig. 4.9). This turns into a decrease of the average capacity. However, looking at Fig. 4.7, we observe a decreasing trend of the average loss rate in case of controller I with respect to controller II.

To have better insight, we evaluate the performance of the two controllers for short flows, i.e., for small file sizes. In this case, we evaluate the file completion time ratio, which we define as the ratio of the file download time in a system which enables rate adaptation to the minimum file download time that would be required in a system that runs always at maximum capacity C_{max}. Thus, a ratio greater than 1 represents a performance degradation. We let the parameter K_c vary between 50 to 150 while keeping the other parameters constant. The rate controller in this case evolves with initial capacity equal to $C_0 = C_{min} = 100$, (i.e., emulating a new download that starts when the system is in idle state). $C_{max} = 1000$, as usual.

Figure 4.8 shows the comparison of controllers performance for small file sizes. First, notice the possible large degradation that the adoption of rate controllers can have on short file download times, i.e., the initial transient of the capacity controller plays a key

role here. This slow-down reduces with the increase in file size, since the initial transient weights less in the latter case. To reduce performance degradation for short files, larger values of K_c are required. Looking at Fig. 4.6, we see that a choice of K_c close to 100 guarantees good performance for both short and long lived files. Second, we see that the rate controller I (solid lines) performs better than controller II (dotted lines) for any given value of K_c.

While, in general, the performance of the two controllers are roughly similar, the comparison shows that controller II is slightly better for long file sizes, whereas, one would prefer controller I for short file sizes. From now on, we use rate controller I, since rate controller II is shown to have more performance impairments for short files.

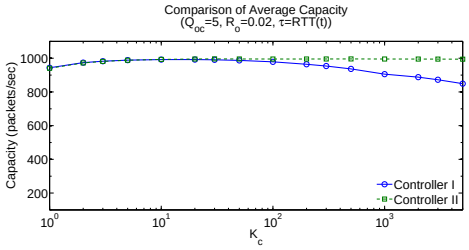

Figure 4.6. Comparison of average capacity for varying K_c.

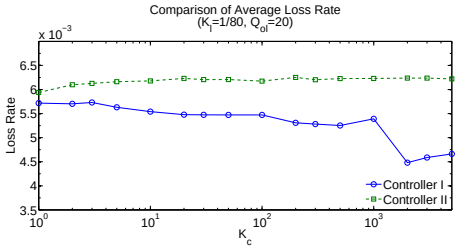

Figure 4.7. Comparison of average loss rate for varying K_c.

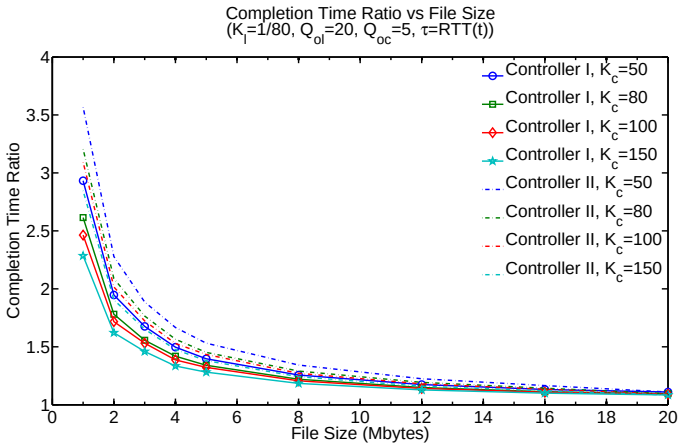

Figure 4.8. Performance comparison between the two controllers.

4.6 Steady state and stability analysis

The steady state analysis of the fluid model allows us to analyze the fundamental relationships between the parameters of the model. To investigate the steady state of the rate adaptation scheme described in section 4.3.1, we derive the equilibrium condition of the set of differential equations, (4.1), (4.2), (4.6), (4.7).

At the steady state, assuming $\bar{Q}_e - Q_{ol} > 0$, when the derivatives are null, we have the following steady states, with subscript 'e' denoting the equilibrium,

$$\begin{aligned} W_e &= \sqrt{\frac{2}{K_l(\bar{Q}_e - Q_{ol})}}, \quad C_e = \frac{W_e}{RTT_e}, \\ \bar{Q}_e &= Q_e = Q_{oc} \end{aligned} \tag{4.11}$$

From these expressions, we observe that, as expected and as typical of TCP behavior, the window size is loss dependent and the mean queue length settles to the value of Q_{oc}, that is the threshold parameter of capacity. Lastly, the capacity is scaled according to the instantaneous offered load.

Considering that the capacity is bounded within C_{min} and C_{max}, we have the following steady state equation for capacity, which basically shows two regimes:

$$C_e = \begin{cases} min\left(\frac{W_e}{RTT_e}, C_{max}\right), & for\ Q_{oc} < Q_{ol} \\ max\left(\frac{W_e}{RTT_e}, C_{min}\right), & for\ Q_{oc} > Q_{ol} \end{cases} \tag{4.12}$$

These expressions suggest that if the control of capacity kicks in before the control of TCP, that occurs through losses (i.e., $Q_{oc} < Q_{ol}$), the capacity reaches C_{max} if the steady state capacity $\frac{W_e}{RTT_e}$ is larger than C_{max}. However, capacity is scaled according to the instantaneous offered load. If, instead, losses occur when capacity controller has not yet entered into play ($Q_{oc} > Q_{ol}$), the TCP control loop shrinks the window, without actually letting the capacity grow. This eventually means that the ratio $\frac{W}{RTT}$ becomes smaller and, therefore, the capacity regime is forced to stabilize at the minimum bound C_{min}, given, of course, that $\frac{W_e}{RTT_e}$ is smaller than C_{min}.

A fundamental guideline to the design of the interaction between the rate adaptation and the TCP control loop is thus to make TCP control loop act when the capacity control is already effective. In terms of parameter settings, this means ensuring that Q_{oc} is smaller than Q_{ol}.

Extending the steady state analysis to rate controller II is simple, and the conclusions are similar to those discussed above. The only difference is that instead of the queue length, Q settling around the Q_{oc}, the rate controller adjusts its capacity in such a way that the estimated queue length $\bar{Q}$ settles around the value of Q_{oc}.

To get further insights, we evaluate the stability of our non-linear model by approximating it with a linear system near the equilibrium points. The model is thus linearized by evaluating its Jacobian matrix at steady states. We evaluate the characteristic polynomial of the Jacobian matrix and utilize Routh-Hurwitz stability criterion to determine the stability of the system. The outcome is a 4^{th} order polynomial, which requires four conditions to be satisfied for the system to be stable (see Appendix A for details).

Considering arbitrary values of parameters (as in the case previously evaluated with $Q_e = 15$, $C_e = 430$, $R_o = 0.02$, $RTT_e = \tau = 0.0367$, $K_c = 1$, $K_l = 1/80$, $\alpha = 0.002$, $\delta_m = 1/1000$), all the coefficients should be greater than zero as well as satisfying all the necessary and sufficient conditions for the system to be stable. The results demonstrate

that the model is stable given the system has sufficient capacity to eliminate packets in queue as described by condition I in Appendix A. Whereas condition II is always true for any positive value of the parameters. Condition III and IV are complex to simplify and draw a straightforward conclusion. However, in nutshell, the linearized model suggests that the small values of smoothing factor α, K_l, and K_c contribute to achieving stability of the system; whereas, increasing capacity way far more than the ratio $\frac{Q}{RTT}$ leads to oscillatory response and system instability.

4.7 Parameter sensitivity

We report in this section the impact of parameters on the capacity evolution considering the case where the controller is based on instantaneous queue length, namely rate controller I. Sensitivity analysis based on rate controller II is omitted, as the conclusions are similar to the ones obtained from rate controller I. In the following, we vary one parameter at a time, while keeping the other ones constant. For simplicity, the values of parameters for each experiment is indicated in the title of the corresponding figure.

Impact of K_c The capacity scaling parameter K_c symbolizes how aggressive the capacity controller is towards the changes of the queue length. Figure 4.9 illustrates the impact of K_c on the evolution of the capacity, i.e., on the performance of the controller. For low values of K_c, the rate controller is slow in reacting and reaches maximum capacity in long times (order of several tens of seconds). Large values of K_c are therefore needed to avoid that the TCP performance deteriorates. However, for high values of K_c, the controller reacts much faster, but it risks to follow too closely the variations of the queue length, as indicated by the large oscillations that can be seen in the zoomed small picture in the same figure.

To have better understanding of the impact of K_c on QoS, we evaluate file completion time ratio for increasing values of K_c and for varying file sizes. The results of such experiment are shown in Fig. 4.10, which shows that the file completion time ratio decreases with the increase in the value of K_c. Moreover, small file sizes pay more in terms of completion time ratio as opposed to large file sizes due to rate controller's inability to complete the initial transient before the download terminates.

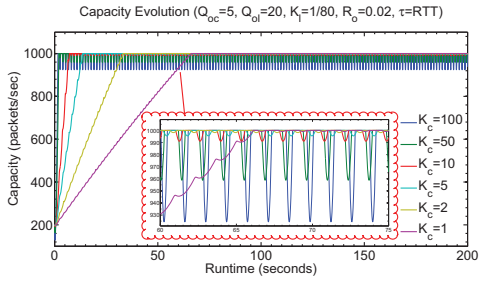

Figure 4.9. Influence of K_c.

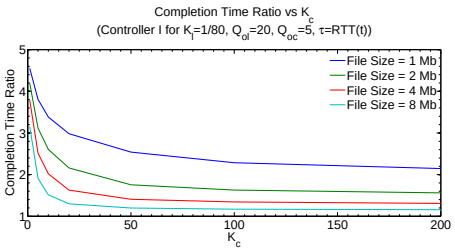

Figure 4.10. Distribution of completion time versus K_c for different file sizes.

Impact of Q_{oc} The parameter Q_{oc} is the threshold for the rate adaptation capacity controller to scale the capacity. The value of Q_{oc} should be chosen smaller than Q_{ol}, so that, before losses occur, capacity is allowed to increase and possibly avoid losses; a choice of Q_{oc} larger than Q_{ol} would make losses trigger the congestion control of TCP before capacity is increased and would, thus, end up forcing the capacity to its minimum C_{min}, as seen in Section 4.6. The steady state analysis suggests that the capacity controller tries to adjust capacity such that the mean queue length is kept around the value of Q_{oc}. The smaller the value of Q_{oc} is, the more aggressive the capacity controller is in pushing capacity to its maximum, C_{max}.

This is confirmed by the results of the experiments that are reported in Fig. 4.11, representing the impact of Q_{oc} on capacity evolution. Notice from Fig. 4.11 that the capacity is pushed to its minimum bound for about the first two seconds. This behavior is due to the fact that, at the beginning, some time is needed to start building up the queue and reach the threshold Q_{oc}. Clearly, small values of Q_{oc} are required to avoid performance deterioration and lead to a more reactive control that makes the capacity grow to the maximum in a short time. Small values allow also to avoid large oscillations in $C(t)$. Observe also the case previously mentioned with $Q_{oc} > Q_{ol}$ that converges to the case of the capacity to its minimum value.

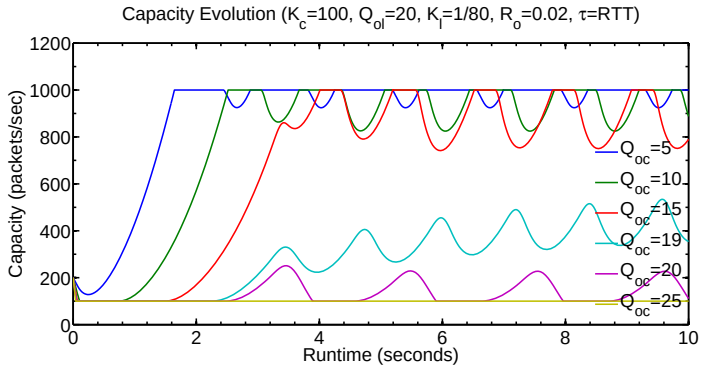

Figure 4.11. Influence of Q_{oc}.

Impact of Q_{ol} The parameter Q_{ol} represents the value of the queue size at which losses start to occur. In terms of control, since losses act as input to the congestion control loop of TCP, Q_{ol} represents the value of the queue at which the congestion control kicks in.

The impact of the parameter Q_{ol} is shown in Fig. 4.12, that reports the capacity evolution for different values of Q_{ol}. High values of Q_{ol} results in high values of the TCP window size, which eventually result in long queues. As previously discussed, when the value of Q_{ol} is large, the capacity controller can act on the capacity and reach the maximum before losses occur, i.e., before TCP control loop enters into play. Thus, the capacity controller becomes more effective for higher value of Q_{ol}, as the difference between $Q(t)$ and Q_{oc} in (4.1) gets higher. Large values of Q_{ol} means a more reactive controller, as shown in the figure. Notice again the case in which $Q_{ol} < Q_{oc}$, that is TCP congestion

control mechanism comes into play before the capacity controller is effective, causing the congestion window to shrink and, eventually, the capacity to be pushed towards C_{min}. Overall, the results are similar to those shown in Fig. 4.11.

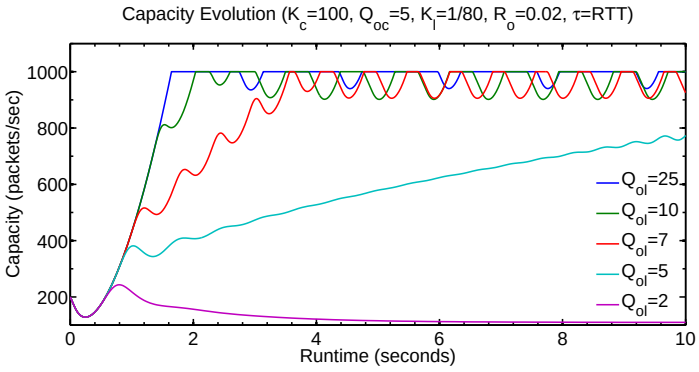

Figure 4.12. Influence of Q_{ol}.

Relationship between Q_{oc} and Q_{ol} To further investigate the interplay between rate adaptation and TCP control loop, we focus on the average capacity measured from each simulation during the initial periods of 200 s. The measure is accounting the initial transient growth, as well as the steady state capacity. We consider the rate controller I and let the parameter Q_{oc} vary between 5 to 25 while keeping $Q_{ol} = 20$ and $K_l = 1/80$. Again, the rate controller evolves based on the same values of $C_{min} = 100$, $C_{max} = 1000$, and $C_0 = 200$. Results are reported in Fig. 4.13 for different values of K_c; to emphasize the relationship between Q_{ol} and Q_{oc}, we report on the x-axis the values of $Q_{ol} - Q_{oc}$.

We observe that the capacity reaches C_{min} for $Q_{ol} - Q_{oc} < 0$, whereas the region with $Q_{ol} - Q_{oc} > 0$ represents those cases in which the capacity controller reacts to the changes of the queue length before the congestion controller kicks in. It is interesting to observe that the value $Q_{ol} - Q_{oc}$ has a stronger impact on results than the value of K_c, suggesting that the understanding of interplay between TCP and rate adaptation control loops is extremely important; carefully controlling the interaction between the two controls allows to achieve larger capacity, while acting on the rate adaptation alone might be useless.

Impact of R_o The propagation delay R_o makes the control loop of TCP be more or less closely coupled with the queue and with the rate adaptation control loop. Large values of R_o make the TCP congestion control react slowly to the queue state and this ends up inducing oscillations to the queue length and transmission rate. We, thus, perform some experiments in which we let R_o vary and collect the results in Fig. 4.14.

When R_o is large, the congestion window grows slowly and less aggressively, and reacts with a large delay to losses. This implies that many losses occur in the meanwhile and TCP ends up reducing significantly the window size. This, in turn, causes less traffic to be injected, allowing the capacity controller to reduce the rate. Notice, indeed, that for large values of R_o, the system is slower in reaching C_{max} with increase in oscillatory behavior.

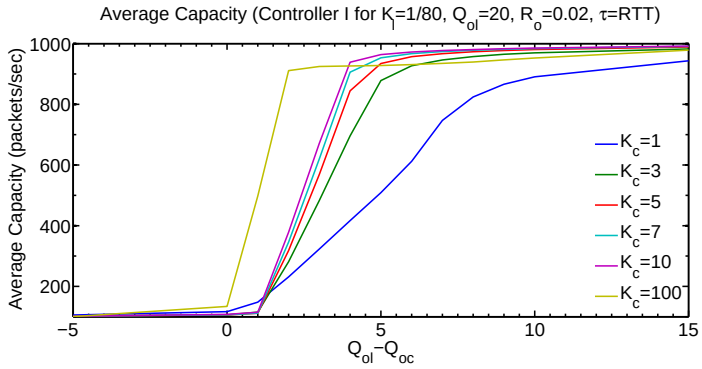

Figure 4.13. Average capacity considering transient and steady state.

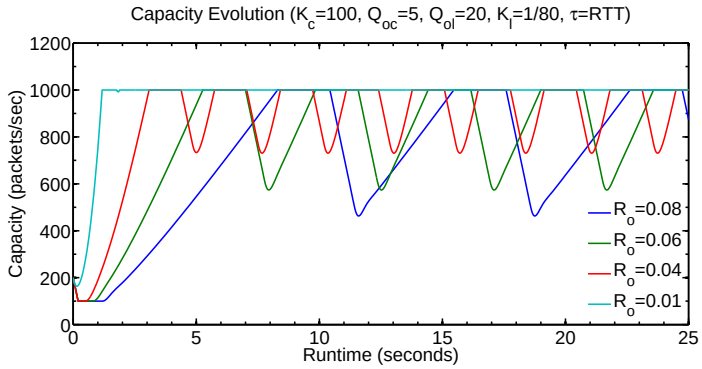

Figure 4.14. Influence of R_o.

Impact of K_l Analogous to K_c, the parameter K_l is the loss scaling factor that represents the reactivity of the loss control. Higher values of K_l result in higher losses, and lower TCP window sizes. A small window size, in its turn, implies less traffic and smaller capacity. The figure has been omitted due to lack of space.

4.8 Power consumption

We now focus on understanding the relationship of rate adaptation and its effect on power consumption with QoS. The well known power management techniques already deployed in mobile and computing devices are Dynamic Frequency Scaling (DFS) and Dynamic Voltage Scaling (DVS). The first one offers linear proportionality between power consumption and performance states, while the latter offers cubical relationship. While it is obvious that rate adaptation provides savings in case of no traffic, the case of low, but continuous, traffic is more complex to understand. To investigate this, we consider the above mentioned two common technologies that either offer a simple linear or cubic proportionality between capacity and power consumption.

4.8.1 Linear power model

In this section we consider linear power consumption profile $P \propto C(t)$, i.e., an increase of capacity corresponds to a proportional increase of power consumption. The total power consumption is the sum of minimum constant power consumption used by the network device to keep it running plus the capacity dependent power consumption. Thus, we take into account only the relative scalable proportion of power consumption as a function of capacity to represent the energy savings and ignore the minimum constant power consumption used by the network device.

The power consumption of a system based on rate adaptation scheme is evaluated using the following expression,

$$P_l = \frac{P_{max}}{C_{max}} \int_0^t C(t)dt \tag{4.13}$$

where P_{max} represents the maximum value of power consumption and the integration limit from 0 to t represents the time to transfer a given amount of data with a given rate controller. To evaluate the amount of saved energy, we compare it with a traditional system with constant maximum capacity to transfer the same given amount of data in t' seconds, where $t' \leqslant t$. This can be expressed as,

$$P_1 = \frac{\dfrac{P_{max}}{C_{max}} \displaystyle\int_0^t C(t)dt}{\displaystyle\int_0^{t'} P_{max}\, dt} = \frac{\dfrac{P_{max}}{C_{max}} \displaystyle\int_0^t C(t)dt}{P_{max}\left(\dfrac{\int_0^t C(t)\, dt}{C_{max}}\right)} = 1 \tag{4.14}$$

This result suggests that under a linear power consumption profile rate adaptation is ineffective in saving energy while it degrades the performance in terms of QoS. Indeed, the same amount of data is transferred at a lower rate (and hence at a lower consumption) but takes longer to be transferred.

4.8.2 Cubic power model

We now consider a power consumption as a cubic function of capacity, $P \propto C^3(t)$, considering again only the scalable portion of power consumption, in percentage so that a hundred percent of power consumption corresponds to maximum utilization of capacity for a specified duration. In case of idle state, where the controller's capacity is pushed to C_{min}, this accounts to about 1% of power consumption using cubic profile, besides the independent minimum constant power consumption used by the device.

The power consumption with a cubic profile can be represented as,

$$P_c = \frac{P_{max}}{C_{max}^3} \int_0^t C^3(t)dt \tag{4.15}$$

Similarly to what done above, we compare the power consumption using cubic profile for a given amount of data to transfer to the power consumption obtained using a constant

capacity C_{max} to transfer the same amount of data. We have,

$$P_2 = \frac{\frac{P_{max}}{C_{max}^3}\int_0^t C^3(t)dt}{P_{max}\left(\frac{\int_0^t C(t)\,dt}{C_{max}}\right)} \times 100 \tag{4.16}$$

The denominator as in (4.14) represents the amount of energy needed by the non energy aware solution to transfer the data.

To better visualize the amount of power consumption, we consider the model based on rate controller I and let the parameter K_c vary between 1 to 100 while keeping $Q_{ol} = 20$, $K_l = 1/80$. We again evaluate the file completion time ratio and its equivalent power consumption in the form of percentage using cubic profile for varying file sizes and increasing values of K_c. Again, the rate controller evolves based on the same values of $C_{min} = 100$, $C_{max} = 1000$, $C_0 = 200$ and $Q_{oc} = 5$.

Figure 4.15 shows the relationship between the file completion time ratio and power consumption in percentage using cubic profile. The results show the direct relationship of QoS with power consumption for increasing values of K_c, and as a result show better performance but less energy savings. Whereas the rate controller is less reactive for lower values of K_c, and thus shows higher savings at the expense of performance degradation.

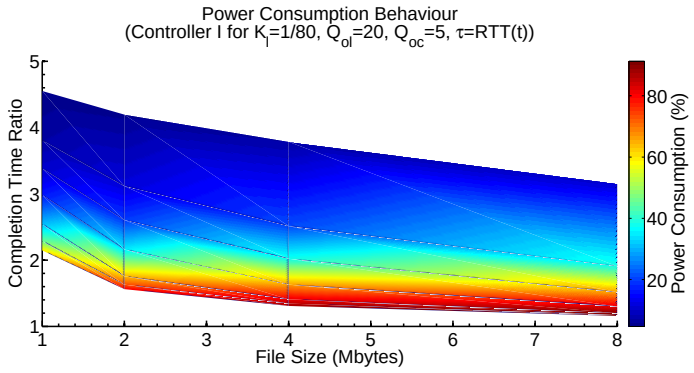

Figure 4.15. Power consumption behaviour versus completion time.

4.9 Conclusion

We have proposed a mathematical model to analyze the interaction between TCP congestion control and rate adaptation schemes implemented at the routers with the objective of saving energy. The model is based on differential equations and simulation experiments have validated the models as being accurate and suitable to predict the system performance. The model accurately approximates TCP behavior despite the fact that a number of TCP mechanisms, like timeouts and slow start, were disregarded.

Results indicate that the interaction between the rate controller and TCP is quite critical, and improper setting of the parameters might lead to bad performance. It might

indeed happen that low rates are used in the router, with corresponding low throughput for the connection. Fine tuning of the rate controller parameters is needed. When properly tuned, the controller can be stable, leading to high capacity when needed with good resulting performance for TCP, confirming that rate adaptation can reduce energy consumption while maintaining high QoS, but at the cost of careful mechanism design.

Chapter 5

General Conclusion

In this thesis, we have successfully solved the problem that ISPs face by designing a machine learning tool to, (i) automatically create proper threshold for each class using real time measurement data, and (ii) instantly suggest the expected stable bandwidth. However, careful selection of proper training set and attributes are required. Training time, number of hidden layers and neurons, architecture etc. are some of the other factors that impact the performance of NN. In future, we intend to work with the ISPs to observe how accurate is the prediction of the NN when applied to the actual users lines. Later, compare its accuracy with the ones obtained by other methods.

In the second part of the work, we have shown that the plain VoIP solution consumes more power as compared to equivalent PSTN solution through real power consumption data. However deploying VoIP is simplest, same switches are used for data and voice services. Moreover, we highlight that VoIP solution has the flexibility to be made energy efficient by adding some sort of energy-wise schemes. Through various energy saving schemes we have highlighted the possibility to reduce the power consumption of VoIP system to a level approximately equal to PSTN solution. Furthermore, optimized VoIP solution provides an additional capability of data services besides voice services at the same cost of power consumption. Newer generation of cisco VoIP switches and PBXs are yet to be deployed at our campus and we intend to compare the results with newer generation of PBXs, VoIP switches and phones.

In the third part of the work, we have shown that our designed fluid model based rate adaptation model validates through simulation experiments can be used to predict the behavior of system. The steady state analysis provide some sort of good insight about the interaction between TCP and rate adaptation scheme and results indicate that the interaction between the rate controller and TCP is quite critical. Thus, improper setting of the parameters might lead to bad performance. Finally, the controller is flexible and based on parameter settings can operate to provide aggressive energy saving or minimal energy saving while maintaining high QoS. As a future work, we plan to extend the model to support other queue management schemes as well as capture the dynamics of stochastic behaviour.

Appendix A

Linearization of the model

We re-write the set of differential equations (4.1), (4.2), (4.6), (4.7) in the following form,

$$\dot{W} = f(W, Q, \bar{Q}, C) = \frac{1}{RTT(t)} - \frac{1}{2}W(t)\left(L(t-\tau)\frac{W(t-\tau)}{RTT(t-\tau)}\right)$$

$$\dot{Q} = g(W, Q, \bar{Q}, C) = \frac{W(t)}{RTT(t)} - C(t)$$

$$\dot{\bar{Q}} = h(W, Q, \bar{Q}, C) = \frac{\log(1-\alpha)}{\delta_m}\bar{Q}(t) - \frac{\log(1-\alpha)}{\delta_m}Q(t)$$

$$\dot{C} = i(W, Q, \bar{Q}, C) = K_c(Q(t) - Q_{oc})$$

Note that for $W(t) \gg 1$, as in [32], the delay term can be ignored, thus $W(t-\tau) \approx W(t)$, similarly assuming $RTT(t-\tau) \approx RTT(t)$ and $\bar{Q}(t-\tau) \approx \bar{Q}(t)$. Finding the Jacobian matrix of the system,

$$J_{(W,Q,\bar{Q},C)} = \begin{bmatrix} \frac{\partial f}{\partial W} & \frac{\partial f}{\partial Q} & \frac{\partial f}{\partial \bar{Q}} & \frac{\partial f}{\partial C} \\ \frac{\partial g}{\partial W} & \frac{\partial g}{\partial Q} & \frac{\partial g}{\partial \bar{Q}} & \frac{\partial g}{\partial C} \\ \frac{\partial h}{\partial W} & \frac{\partial h}{\partial Q} & \frac{\partial h}{\partial \bar{Q}} & \frac{\partial h}{\partial C} \\ \frac{\partial i}{\partial W} & \frac{\partial i}{\partial Q} & \frac{\partial i}{\partial \bar{Q}} & \frac{\partial i}{\partial C} \end{bmatrix}$$

Evaluating at operating points $(W_e, Q_e, \bar{Q}_e, C_e)$ and substituting, $K_l(\bar{Q}_e - Q_{ol}) = \frac{2}{{W_e}^2}$, we have,

$$\begin{bmatrix} -\frac{2}{W_e\, RTT_e} & 0 & -\frac{W_e^2\, K_l}{2\, RTT_e} & 0 \\ \frac{1}{RTT_e} & -\frac{W_e}{C_e\,(RTT_e)^2} & 0 & \frac{Q_e\, W_e}{C_e^2\,(RTT_e)^2} - 1 \\ 0 & -\frac{\log(1-\alpha)}{\delta_m} & \frac{\log(1-\alpha)}{\delta_m} & 0 \\ 0 & K_c & 0 & 0 \end{bmatrix}$$

From equation (4.11), substituting $W_e = C_e\, RTT_e$,

$$\begin{bmatrix} -\frac{2}{C_e\,(RTT_e)^2} & 0 & -\frac{{C_e}^2\, RTT_e\, K_l}{2} & 0 \\ \frac{1}{RTT_e} & -\frac{1}{RTT_e} & 0 & \frac{Q_e}{C_e\, RTT_e} - 1 \\ 0 & -\frac{\log(1-\alpha)}{\delta_m} & \frac{\log(1-\alpha)}{\delta_m} & 0 \\ 0 & K_c & 0 & 0 \end{bmatrix}$$

Evaluating the characteristic polynomial of the Jacobian matrix,

$$\begin{vmatrix} -\frac{2}{C_e\,(RTT_e)^2} - \lambda & 0 & -\frac{{C_e}^2\, RTT_e\, K_l}{2} & 0 \\ \frac{1}{RTT_e} & -\frac{1}{RTT_e} - \lambda & 0 & \frac{Q_e}{C_e\, RTT_e} - 1 \\ 0 & -\frac{\log(1-\alpha)}{\delta_m} & \frac{\log(1-\alpha)}{\delta_m} - \lambda & 0 \\ 0 & K_c & 0 & -\lambda \end{vmatrix} = 0$$

Simplifying the above, we get,

$$\lambda^4 + \left(-\frac{\log(1-\alpha)}{\delta_m} + \frac{1}{\mathrm{RTT}_e} + \frac{2}{C_e\,{\mathrm{RTT}_e}^2}\right)\lambda^3 + \left(K_c + \frac{2}{C_e\,{\mathrm{RTT}_e}^3} - \frac{\log(1-\alpha)}{\mathrm{RTT}_e\,\delta_m} - \frac{2\log(1-\alpha)}{C_e\,{\mathrm{RTT}_e}^2\,\delta_m} - \frac{K_c\,Q_e}{C_e\,\mathrm{RTT}_e}\right)\lambda^2 + \left(-\frac{K_c\log(1-\alpha)}{\delta_m} + \frac{2\,K_c}{C_e\,{\mathrm{RTT}_e}^2} - \frac{2\log(1-\alpha)}{C_e\,{\mathrm{RTT}_e}^3\,\delta_m} - \frac{2\,K_c\,Q_e}{{C_e}^2\,{\mathrm{RTT}_e}^3} - \frac{{C_e}^2\,K_l\log(1-\alpha)}{2\,\delta_m} + \frac{K_c\,Q_e\log(1-\alpha)}{C_e\,\mathrm{RTT}_e\,\delta_m}\right)\lambda + \left(-\frac{2\,K_c\log(1-\alpha)}{C_e\,{\mathrm{RTT}_e}^2\,\delta_m} + \frac{2\,K_c\,Q_e\log(1-\alpha)}{{C_e}^2\,{\mathrm{RTT}_e}^3\,\delta_m}\right) = 0$$

Thus we have the form,

$$\lambda^4 + a_3\lambda^3 + a_2\lambda^2 + a_1\lambda + a_0 = 0$$

According to Routh-Hurwitz Stability Criterion, the following conditions must be satisfied for the 4^{th} order characteristic polynomial to be stable. To simplify the approximation, we arbitrarily choose to fix $\alpha = 0.002$ and $\delta_m = 1/1000$ as constant, so that the factor $\frac{\log(1-\alpha)}{\delta_m} = -2$. While assuming $C_e,\ Q_e,\ RTT_e,\ K_c,\ K_l$ take positive values. We have,

Condition I

$$a_0 > 0$$

$$\left(-\frac{2\,K_c\,\log\left(1-\alpha\right)}{C_e\,{\mathrm{RTT}_e}^2\,\delta_m}+\frac{2\,K_c\,Q_e\,\log\left(1-\alpha\right)}{{C_e}^2\,{\mathrm{RTT}_e}^3\,\delta_m}\right)>0$$

$$\left(\frac{4\,K_c}{C_e\,{\mathrm{RTT}_e}^2}-\frac{4\,K_c\,Q_e}{{C_e}^2\,{\mathrm{RTT}_e}^3}\right)>0$$

$$C_e>\frac{Q_e}{RTT_e}$$

Condition II

$$a_3>0$$

$$\left(-\frac{\log\left(1-\alpha\right)}{\delta_m}+\frac{1}{\mathrm{RTT}_e}+\frac{2}{C_e\,{\mathrm{RTT}_e}^2}\right)>0$$

$$\left(2+\frac{1}{\mathrm{RTT}_e}+\frac{2}{C_e\,{\mathrm{RTT}_e}^2}\right)>0$$

Condition III

$$a_2a_3-a_1>0$$

$$\frac{K_c}{\mathrm{RTT}_e}+\frac{4}{\mathrm{RTT}_e}+\frac{2}{{\mathrm{RTT}_e}^2}+\frac{8}{C_e\,{\mathrm{RTT}_e}^2}+\frac{8}{C_e\,{\mathrm{RTT}_e}^3}+$$
$$\frac{2}{C_e\,{\mathrm{RTT}_e}^4}+\frac{8}{{C_e}^2\,{\mathrm{RTT}_e}^4}+\frac{4}{{C_e}^2\,{\mathrm{RTT}_e}^5}>{C_e}^2\,K_l+\frac{K_c\,Q_e}{C_e\,{\mathrm{RTT}_e}^2}$$

Condition IV

$$a_1a_2a_3-{a_1}^2-a_0{a_3}^2>0$$

$$\frac{8\,K_c}{\mathrm{RTT}_e} + \frac{4\,K_c}{{\mathrm{RTT}_e}^2} + \frac{8\,K_l}{{\mathrm{RTT}_e}^4} + \frac{4\,K_l}{{\mathrm{RTT}_e}^5} + \frac{16}{C_e\,{\mathrm{RTT}_e}^4} +$$

$$\frac{8}{C_e\,{\mathrm{RTT}_e}^5} + \frac{32}{{C_e}^2\,{\mathrm{RTT}_e}^5} + \frac{32}{{C_e}^2\,{\mathrm{RTT}_e}^6} + \frac{8}{{C_e}^2\,{\mathrm{RTT}_e}^7} +$$

$$\frac{32}{{C_e}^3\,{\mathrm{RTT}_e}^7} + \frac{16}{{C_e}^3\,{\mathrm{RTT}_e}^8} + \frac{2\,{K_c}^2}{\mathrm{RTT}_e} + \frac{2\,{K_c}^2}{C_e\,{\mathrm{RTT}_e}^3} + \frac{8\,C_e\,K_l}{{\mathrm{RTT}_e}^2} +$$

$$\frac{4\,C_e\,K_l}{{\mathrm{RTT}_e}^3} + \frac{2\,C_e\,K_l}{{\mathrm{RTT}_e}^4} + \frac{8\,K_c}{C_e\,{\mathrm{RTT}_e}^3} + \frac{8\,K_c}{C_e\,{\mathrm{RTT}_e}^4} + \frac{8\,K_c}{{C_e}^2\,{\mathrm{RTT}_e}^5} +$$

$$\frac{4\,K_c}{{C_e}^2\,{\mathrm{RTT}_e}^6} + \frac{8\,K_c}{{C_e}^3\,{\mathrm{RTT}_e}^7} + \frac{4\,{C_e}^2\,K_l}{\mathrm{RTT}_e} + \frac{2\,{C_e}^2\,K_l}{{\mathrm{RTT}_e}^2} + \frac{2\,{K_c}^2\,{Q_e}^2}{{C_e}^2\,{\mathrm{RTT}_e}^3} +$$

$$\frac{2\,{K_c}^2\,{Q_e}^2}{{C_e}^3\,{\mathrm{RTT}_e}^5} + \frac{{C_e}^2\,K_c\,K_l}{\mathrm{RTT}_e} + \frac{2\,K_c\,K_l\,Q_e}{{\mathrm{RTT}_e}^3} + \frac{2\,C_e\,K_c\,K_l\,Q_e}{\mathrm{RTT}_e} >$$

$${C_e}^4\,{K_l}^2 + 2\,{C_e}^2\,K_c\,K_l + \frac{8\,K_c\,Q_e}{C_e\,{\mathrm{RTT}_e}^2} + \frac{4\,K_c\,Q_e}{C_e\,{\mathrm{RTT}_e}^3} +$$

$$\frac{8\,K_c\,Q_e}{{C_e}^2\,{\mathrm{RTT}_e}^4} + \frac{8\,K_c\,Q_e}{{C_e}^2\,{\mathrm{RTT}_e}^5} + \frac{8\,K_c\,Q_e}{{C_e}^3\,{\mathrm{RTT}_e}^6} + \frac{4\,K_c\,Q_e}{{C_e}^3\,{\mathrm{RTT}_e}^7} +$$

$$\frac{8\,K_c\,Q_e}{{C_e}^4\,{\mathrm{RTT}_e}^8} + \frac{4\,{K_c}^2\,Q_e}{C_e\,{\mathrm{RTT}_e}^2} + \frac{4\,{K_c}^2\,Q_e}{{C_e}^2\,{\mathrm{RTT}_e}^4} + \frac{2\,C_e\,K_c\,K_l}{{\mathrm{RTT}_e}^2} +$$

$$\frac{C_e\,K_c\,K_l\,Q_e}{{\mathrm{RTT}_e}^2}$$

References

[1] R. Bolla, R. Bruschi, F. Davoli, and F. Cucchietti. Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-Aware Fixed Network Infrastructures. *Communications Surveys Tutorials, IEEE*, 13(2):223–244, 2011.

[2] R. Prasad, C. Dovrolis, M. Murray, and K. Claffy. Bandwidth Estimation: Metrics, Measurement Techniques, and Tools. *IEEE Network*, 17(6):27–35, November 2003.

[3] D. Croce, T. En-Najjary, G. Urvoy-Keller, and E.W. Biersack. Non-cooperative Available Bandwidth Estimation towards ADSL Links. In *IEEE INFOCOM Workshops 2008*, pages 1–6, April 2008.

[4] W. Jiang and T.F. Williams. Detecting and Measuring Asymmetric Links in an IP Network. In *Global Telecommunications Conference, 1999 (GLOBECOM '99)*, volume 3, pages 1727–1731, 1999.

[5] Hui Min Chong and H.S. Matthews. Comparative Analysis of Traditional Telephone and Voice-over-Internet Protocol (VoIP) Systems. In *IEEE International Symposium on Electronics and the Environment*, pages 106–111, 2004.

[6] S. Al-Chalabi. Powering the Telephone over Optical Links for High Availability, Low Cost, and Small Carbon Footprint. *Communications Magazine, IEEE*, 49(9):48–55, 2011.

[7] John A. C. Bingham. *ADSL, VDSL, and Multicarrier Modulation.* Wiley-Interscience, 2000.

[8] K. Yale. Preparng the right Data Diet for Training Neural Networks. *IEEE Spectrum*, 34(3):64–66, March 1997.

[9] Xiao-Feng Gu, Lin Liu, Jian-Ping Li, Yuan-Yuan Huang, and Jie Lin. Data Classification based on Artificial Neural Networks. In *International Conference on Apperceiving Computing and Intelligence Analysis, 2008 (ICACIA 2008)*, pages 223–226, December 2008.

[10] H.H. Dam, H.A. Abbass, C. Lokan, and Xin Yao. Neural-Based Learning Classifier Systems. *IEEE Transactions on Knowledge and Data Engineering*, 20(1):26–39, January 2008.

[11] B.B. Misra, B.N. Biswal, P.K. Dasb, and G. Panda. Simplified Polynomial Neural Network for Classification Task in Data Mining. In *IEEE Congress on Evolutionary Computation, 2007 (CEC 2007)*, pages 721–728, September 2007.

[12] S. Nirkhi. Potential use of Artificial Neural Network in Data Mining. In *The 2nd International Conference on Computer and Automation Engineering, 2010 (ICCAE)*, volume 2, pages 339–343, February 2010.

[13] ITU-T. G.992.1: Asymmetric Digital Subscriber Line (ADSL) Transceivers. Technical report, International Telecommunication Union (ITU-T), July 1999.

[14] ITU-T. G.992.2: Splitterless Asymmetric Digital Subscriber Line (ADSL) Transceivers. Technical report, International Telecommunication Union (ITU-T), July 1999.

[15] ITU-T. The Status of Voice over Internet Protocol (VoIP) Worldwide. Technical report, 2007.

[16] Salman A. Baset and H. Schulzrinne. Energy Efficiency of Voice-over-IP Systems. Technical report, Department of Computer Science, Columbia University, NY, 2009.

[17] V. Namboodiri and Lixin Gao. Energy-Efficient VoIP over Wireless LANs. *IEEE Transactions on Mobile Computing*, 9(4):566–581, 2010.

[18] IEEE. P802.3az Energy Efficient Ethernet Task Force. 2010.

[19] Sergiu Nedevschi, Lucian Popa, Gianluca Iannaccone, Sylvia Ratnasamy, and David Wetherall. Reducing Network Energy Consumption via Sleeping and Rate-Adaptation. In *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, NSDI'08, pages 323–336, Berkeley, CA, USA, 2008. USENIX Association.

[20] K. Christensen, P. Reviriego, B. Nordman, M. Bennett, M. Mostowfi, and J.A. Maestro. IEEE 802.3az: The Road to Energy Efficient Ethernet. *Communications Magazine, IEEE*, 48(11):50–56, november 2010.

[21] M. Allman, V. Paxson, and E. Blanton. TCP Congestion Control. Technical report, IETF RFC 5681, Sept. 2009.

[22] Y.S. Hanay, Wei Li, R. Tessier, and T. Wolf. Saving Energy and Improving TCP Throughput with Rate Adaptation in Ethernet. In *Communications (ICC), 2012 IEEE International Conference on*, pages 1249 –1254, june 2012.

[23] Yan Cai, Y.S. Hanay, and T. Wolf. Practical Packet Pacing in Small-Buffer Networks. In *IEEE International Conference on Communications, 2009. ICC '09.*, pages 1–6, 2009.

[24] Baoke Zhang, K. Sabhanatarajan, A. Gordon-Ross, and A. George. Real-Time Performance Analysis of Adaptive Link Rate. In *33rd IEEE Conference on Local Computer Networks, 2008. LCN 2008*, pages 282–288, 2008.

[25] C. Gunaratne, K. Christensen, B. Nordman, and S. Suen. Reducing the Energy Consumption of Ethernet with Adaptive Link Rate (ALR). *Computers, IEEE Transactions on*, 57(4):448–461, April 2008.

[26] Matthew Mathis, Jeffrey Semke, Jamshid Mahdavi, and Teunis Ott. The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm. *SIGCOMM Comput. Commun. Rev.*, 27(3):67–82, July 1997.

[27] C.V. Hollot, V. Misra, D. Towsley, and W. Gong. Analysis and Design of Controllers for AQM Routers Supporting TCP Flows. *Automatic Control, IEEE Transactions on*, 47(6):945–959, 2002.

[28] Vishal Misra, Wei-Bo Gong, and Don Towsley. Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED. In *IN PROCEEDINGS OF ACM SIGCOMM*, pages 151–160, 2000.

[29] C. Panarello, M.A. Marsan, A. Lombardo, M. Mellia, M. Meo, and G. Schembra. On the Intertwining between Capacity Scaling and TCP Congestion Control. In *Third International Conference on Future Energy Systems: Where Energy, Computing and Communication Meet (e-Energy), 2012*, pages 1–4, may 2012.

[30] Faheem Khuhawar, Marco Mellia, and Michela Meo. Modeling The Interaction between TCP and Rate Adaptation at Links. In *Teletraffic Congress (ITC), 2013 25th International*, pages 1–8, 2013.

[31] Sally Floyd and Van Jacobson. Random Early Detection Gateways for Congestion Avoidance. *Networking, IEEE/ACM Transactions on*, 1(4):397–413, August 1993.

[32] C.V. Hollot, V. Misra, D. Towsley, and Wei-Bo Gong. A Control Theoretic Analysis of RED. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1510–1519 vol.3, 2001.