Aalto University
School of Science
Degree Programme in Computer Science and Engineering

Betsegaw Lemma Amersho

Simulating energy-aware networks in large-scale distributed systems

Master's Thesis
Espoo, June 26, 2017

DRAFT! — March 24, 2017 — DRAFT!

Supervisors: Professor Antti Ylä-Jääski, Aalto University
Professor Pekka Perustieteilijä, University of Helsinki
Advisor: Olli Ohjaaja M.Sc. (Tech.)

Aalto University
School of Science
Degree Programme in Computer Science and Engineering

ABSTRACT OF
MASTER'S THESIS

Author: Betsegaw Lemma Amersho

Title:
Simulating energy-aware networks in large-scale distributed systems

Date:	June 26, 2017	**Pages:**	30
Major:	Data Communication Software	**Code:**	T-110
Supervisors:	Professor Martin Quinson Dr. Anne-Cécile Orgerie		
Advisor:	Olli Ohjaaja M.Sc. (Tech.)		

A dissertation or thesis is a document submitted in support of candidature for a degree or professional qualification presenting the author's research and findings. In some countries/universities, the word thesis or a cognate is used as part of a bachelor's or master's course, while dissertation is normally applied to a doctorate, whilst, in others, the reverse is true.

!FIXME **Abstract text goes here (and this is an example how to use fixme).** FIXME! Fixme is a command that helps you identify parts of your thesis that still require some work. When compiled in the custom `mydraft` mode, text parts tagged with fixmes are shown in bold and with fixme tags around them. When compiled in normal mode, the fixme-tagged text is shown normally (without special formatting). The draft mode also causes the "Draft" text to appear on the front page, alongside with the document compilation date. The custom `mydraft` mode is selected by the `mydraft` option given for the package `aalto-thesis`, near the top of the `thesis-example.tex` file.

The thesis example file (`thesis-example.tex`), all the chapter content files (`1introduction.tex` and so on), and the Aalto style file (`aalto-thesis.sty`) are commented with explanations on how the Aalto thesis works. The files also contain some examples on how to customize various details of the thesis layout, and of course the example text works as an example in itself. Please read the comments and the example text; that should get you well on your way!

Keywords: ocean, sea, marine, ocean mammal, marine mammal, whales, cetaceans, dolphins, porpoises

Language: English

Acknowledgements

I wish to thank all students who use LaTeX for formatting their theses, because theses formatted with LaTeX are just so nice.

Thank you, and keep up the good work!

Espoo, June 26, 2017

Betsegaw Lemma Amersho

Abbreviations and Acronyms

2k/4k/8k mode	COFDM operation modes
3GPP	3rd Generation Partnership Project
ESP	Encapsulating Security Payload; An IPsec security protocol
FLUTE	The File Delivery over Unidirectional Transport protocol
e.g.	for example (do not list here this kind of common acronymbs or abbreviations, but only those that are essential for understanding the content of your thesis.
note	Note also, that this list is not compulsory, and should be omitted if you have only few abbreviations

Contents

Chapter 1

Introduction

Context of the Study

Provide a brief history of the issues to date.

Situate your particular topic within the broad area of research.

Note that the field is changing, and more research is required on your topic.

Problem statement or Motivation for the Study

Identify a key point of concern (for example, increasing use or prominence, lack of research to date, response to an agenda, a new discovery, or perhaps one not yet applied to this context).

Refer to the literature only to the extent needed to demonstrate why your project is worth doing. Reserve your full review of existing theory or practice for later chapters.

Be sure that the motivation, or problem, suggests a need for further investigation.

Aim and Scope

Be sure that your aim responds logically to the problem statement.

Stick rigorously to a single aim. Do not include elements in it that describe how you intend to achieve this aim; reserve these for a later chapter.

When you have written the conclusions to your whole study; check that they respond to this aim. If they don't, change the aim or rethink your conclusions.

If you change the aim, revise the motivation for studying it.

Be sure to establish the scope of your study by identifying limitations of factors such as time, location, resources, or the established boundaries of particular fields or theories.

Significance of the Study

Explain how your thesis contributes to the field.

There are four main areas of contribution: theory development, tangible solution, innovative methods, and policy extension. One of these contributions must be identified as the basis of your primary contribution to the field.

In contrast to reports for industry, theory development is an expected and required contribution; for PhDs in particular, it must be "original".

Structure of the Thesis

You should use transition in your text, meaning that you should help the reader follow the thesis outline. Here, you tell what will be in each chapter of your thesis.

Chapter 2

Background

In this chapter ...

Electricity consumption of ICT equipments

ICT equipments consume a significant amount of electricity. A survey conducted by Heddeghem et al. [12] shows the electricity consumption and growth trends of three classes of ICT equipments: personal computers, communication networks, and data centers. Personal computers include equipments such as desktop, laptop and external monitors. Communication networks includes residential network access equipments (such as WiFi routers and modems), network equipments used in offices (such as routers and switches) and telecom-operator network equipments (such as base stations, routers and optical amplification systems). Data-centers house storage and computing servers, communication network equipments, and power provisioning and cooling facilities. In this classification there are overlaps, for instance, telcom operator can have office network equipments and data-centers. After carefully avoiding possible redundant measurements, the researchers estimated absolute electricity consumption and annual consumption growth rate of each category of equipments for the period 2007 and 2012. The results of the study show that the global electricity consumption of ICT equipments in all the three categories combined contributed 3.9% in 2007 and 4.6% in 2012. The estimated annual growth rate of the individual category is 5% for personal computers, 10% for communication networks, and 4% for data-centers. These growth rates are higher than that of the total global electricity consumption, which is 3%.

Data-center electricity consumption

In Section 2.1 we described data-center's global share in electricity consumption. In this section we describe the components involved within the data center itself.

Electricity consumption units with in a typical data-center can be classified into two broad groups [8]: The first group is IT equipments (which includes computing servers, storage servers and networking components) and the other group is infrastructure facilities (which includes power provisioning, cooling and lighting components).

Figure 2.1 [8] shows the electricity consumption proportion of the data-center components. This value differs significantly from one data-center to another [2], for instance, due to architectural difference[11] or energy efficiency of the components. The infrastructure facility components take the large proportion (65%) of the consumption.

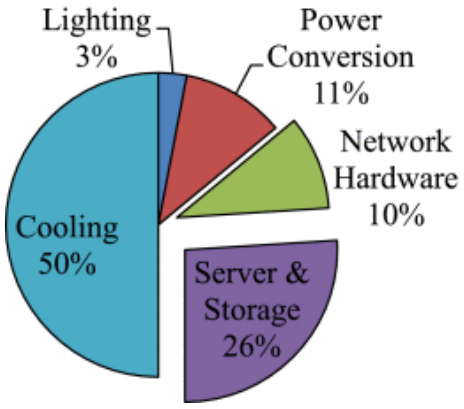


Figure 2.1: Energy consumption percentage of data-center components

Though the infrastructure facility consumes relatively larger amount of electricity, the focus of this study is on the IT equipment components, particularly on the network equipments.

If we further zoom in on the IT equipments part, we can find server, storage and network equipments. A data-center servers consist of one or more CPU cores, memory and I/O devices. The energy consumption relationship among these components is shown in Figure 2.2. Combined, Memory and CPU units consume the larger amount of energy relative to other components. The fact that CPU is the dominant electricity consuming unit is exploited by Fan et al. in [9] to model the dynamic power usage of thousands

of servers by using only CPU utilization as a parameter. The result of their study was very accurate, with error as low as 1%.

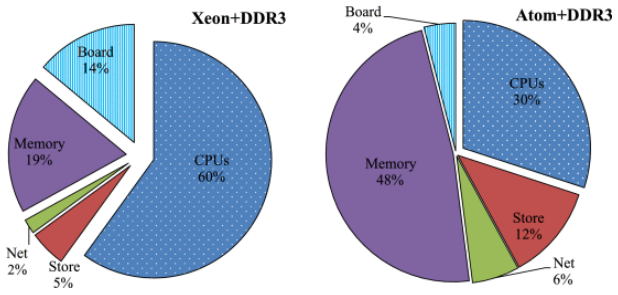


Figure 2.2: Energy consumption percentage of Xeon based (on the left) and Atom based (on the right) servers

Energy proportionality

The only reason the study of energy consumption management of network equipment becomes so important is that, in general, ICT equipments do not consume energy proportional to their workload. An ideal ICT equipment is the one which consume zero electricity when it is idle, and it consumes electricity proportional to its workload when it is active. However, the reality is, even power efficient servers consume about 50% of their peak power [3], even when they are doing nothing. This percentage can even reach 85% for network switches [10]. Figure 2.3 in [15] shows the energy proportionality of a typical network equipment. From the graph we can observe that the dynamic power consumption range is narrow. Three approaches are in common use to deal with this situation. The first one is re-engineering network devices so as to make them more energy proportional, device vendors are the prime role player in this aspect. The second approach is related to the operating rate of a network equipment port. A typical switch can operate on different transmission rate (100Mbps, 1 Gbps or 10 Gbps). An active port transmitting at 10 Gbps can consume more energy than if it transmit at 100 Mbps. Rate adaptation is the approach devised to take advantage of this situation. Instead of transmitting at the maximum rate all time, the network port can be made to adapt to the actual traffic load. This energy saving approach is known as Adaptive Link Rate (ALR). The third approach, which is known as Low Power Idle (LPI), allows a network device to send data as fast as possible and then enter low power mode between transfers. The low power

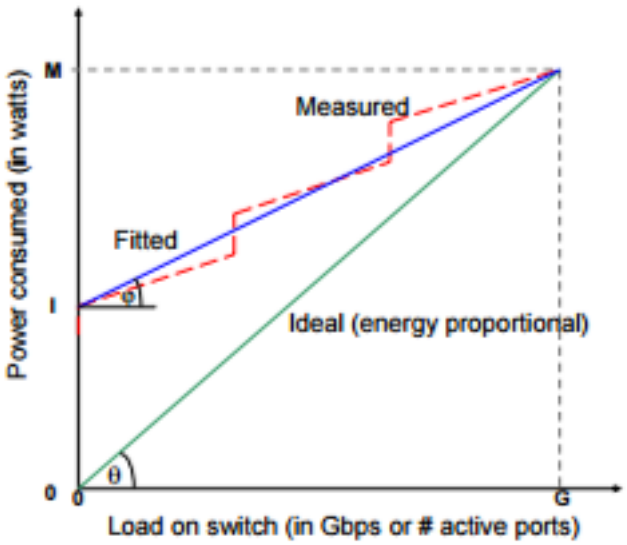


Figure 2.3: Ideal and measured energy proportionality of a network equipment

mode can further be extended by a technique called packet coalescing, which allows more energy saving [4].

Packet-level and flow-level Simulators

Packet-level simulators strives to model a given network phenomenon at the granularity level of packets, thus in general they are accepted by the research community to be more accurate compared to flow-level simulators [6]. One of the most popular packet-level simulator is NS-3, which is categorized under discrete-event simulator with events corresponding to sending and receiving packets [14]. Though packet-level simulators tend to be more accurate, they fail to scale well in the area of large-scale networks.

In the area of large-scale networks, flow-level simulators are the preferred alternative. Rather than modeling a given network phenomenon at a packet level, flow-level simulators treat a set of packets as a single unit [6]. The most commonly used definition for flow in the context of computer networking is coined by Claffy et al. in [7]:

"...a flow ...a unidirectional traffic stream with a unique [source-IP-address, source-port, destination-IP-address, destination-port, IP-protocol] tuple ..."

In addition to the five tuple mentioned in the definition, a flow also has

a limited time duration. Claffy et al. used a time limit of 64 seconds as a flow duration in their study. Researchers such as Carneiro et al. [5], adopted this same definition to develop flow monitoring module for NS-3, a module that can generate information such as amount of packets or bytes transferred, packets dropped or transmission start and end time for each flow. Barakat et al. in [1] also used the same definition to model traffic at the flow-level for the Internet backbone link. By abstracting away fine details, flow-level models provides easy way to instantiate experiments and they also scale very well for conducting large-scale network simulations [1, 6].

Simulating and modeling energy consumption of large-scale networks

One way of conducting energy consumption or any other experiment is to use real production environment or test-bed environment, both are referred to as *in vivo* in [6]. In the former case, handling transient and varying conditions would make the data collection and prediction very difficult and often times, a production environment is not available for experimentation. In the later case, it requires setting-up a separate testing environment designed solely for the purpose of conducting the desired experiment. This approach apart from being expensive, it requires significant amount of time for experiment setup and, it is also non-repeatable as experimenting with different scenario demands a modified or new configuration.

The other alternative for experimenting is simulation, also referred to as *in silico* in [6]. Simulation, unlike real environment, allows great flexibility in terms of experiment configuration, control and repetition. In addition it can also be less time consuming and less expensive.That is why virtually in all computer network related researches simulations are widely used.

In this study we simulate energy-aware large scale distributed networks using SimGrid (Detail description about SimGrid follows in the next section). When we say large-scale distributed network, we are referring to a set of networks residing inside in the distributed data centers and also the networks that are used to connect them.

The energy consumption E of an equipment depends on the operating power P at time t. The total energy consumption for a time period T is given by equation (2.1) [16].

$$E(T) = \int_0^T P(t)dt \tag{2.1}$$

Due to the energy proportionality characteristic described in Section 2.3, the common approach used to compute the energy consumption is to divide the power component into two parts: static/idle power (P_{static}) and dynamic power ($P_{dynamic}$) as shown in equation (2.2). Then the total energy is obtained by multiplying the total power, P_{total} by the time duration [8, 13, 15, 16].

$$P_{total} = P_{static} + P_{dynamic} \tag{2.2}$$

For a typical network equipment such as a switch, the static part constitutes the power consumption of the chassis and the line-cards (when all the ports on the line-cards are switched off). The dynamic part, on the other hand, constitutes the power consumption of the switch ports running at a given rate multiplied by the utilization factor [15]. Equation (2.3) shows how to compute the total power for a switch, where P_{switch}, is the total power consumption of a switch, $P_{chassis}$ and $P_{linecard}$ is the power consumption of the chassis and the line card, respectively. P_{rate}, is the power consumption of a given port at a given rate and $numports_{rate}$ is the number of ports running at a given rate. The rate can take any value such as 10 Mbps, 100 Mbps, 1 Gbps or 10 Gbps.

$$\begin{aligned} P_{switch} = P_{chassis} + (numlinecards \times P_{linecard}) + \\ \sum_{rate=min}^{max} (numports_{rate} \times P_{rate} \times utilizationFactor) \end{aligned} \tag{2.3}$$

SimGrid

Figure 2.4 shows the structure of SimGrid and how its core works. The top three components are the APIs that users can use to develop their simulation. Both MSG and SMPI are used to specify simulated applications as concurrent processes. The difference is that using MSG, users can simulate any arbitrary application, whereas, using SMPI users can simulating existing MPI applications, the MPI processes are created automatically from C or Fortran MPI programs. SIMDAG, on the other hand, does not use concurrent processes. It allows users to describe their application as communicating task graph. The next layer, SIMIX, implements the mechanisms that are required to simulate the concurrent process of MSG and SMPI applications. It also provides process control and synchronization functionalities. The bottom layer, SURF, is the simulation core, it simulates the execution of activities on computing or communication resources [6].

In SimGrid for each simulated activity, such as computation or data transfer, there is a corresponding condition variable, in Figure 2.4 it is shown in

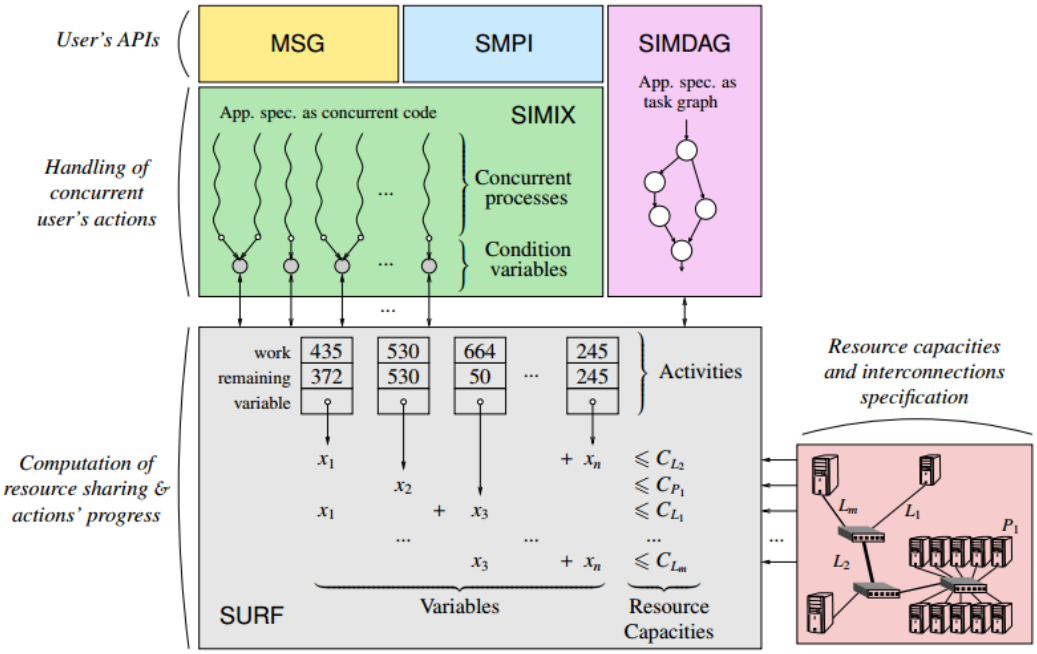


Figure 2.4: Architecture of SimGrid

SIMIX box. This condition variable synchronizes the concurrent processes of the simulated applications. The computing (P_x) and the communication (L_x) resources are shown on the bottom-right side of the figure. Computing resources are defined in terms of computing power, whereas, communication resources are defined in terms of bandwidth and latency. As shown in the SURF box, multiple activities can share the same resource (e.g. (x_1, x_n), (x_1, x_3) or (x_3, x_n)) or one activity can use multiple resources (e.g. x_1 or x_3 or x_n). Activities that share the same resource are limited by the capacity of that resource. Each activity is defined by the total and remaining work to be executed. When the work associated with the activity completes, the corresponding upper layer components receive a notification signal [6].

As we have already pointed out in Section 2.4, the primary advantage of flow-level simulation is its scalability in terms of speed and memory usage. SimGrid uses flow-level analytical model for simulating TCP network phenomenon [6]. To see the scalability of the flow-level model, the SimGrid team compared it with other widely used simulators such as GridSim and OverSim. After simulating 500,000 tasks both on GridSim and SimGrid, the results demonstrate that SimGrid is 257 times faster and 26 times more memory efficient. Similarly, the comparison result with OverSim shows that

SimGrid is 15 times faster and it can also simulate scenarios 10 times larger. Concerning the accuracy, though the simulator gives very good accuracy in most case studies, there are situations where it fails to give accurate result. As an example, the comparison study of SimGrid with packet-level simulator GTNetS show that for data size less than 100 KiB there is a significant difference in prediction.

Currently SimGrid is used to simulate different network phenomenon in the area of large-scale distributed systems such as grid, cloud, volunteer and HPC [1]. Concerning energy consumption models, it houses energy models for CPU but there is no model for network equipments.Therefore, the focus of this study is to propose and implement network energy consumption model for SimGrid. The implementation of this model, together with the existing CPU energy model, allows us to estimate the energy consumption of large-scale networks that reside within or outside a data-center that are discussed in Section 2.2 and Section 2.1.

Related Simulators

[1] http://simgrid.gforge.inria.fr/

Chapter 3

Environment

A problem instance is rarely totally independent of its environment. Most often you need to describe the environment you work in, what limits there are and so on. This is a good place to do that. First we tell you about the LaTeX working environments and then is an example from an thesis written some years ago.

LaTeX working environments

To create LaTeX documents you need two things: a LaTeX environment for compiling your documents and a text editor for writing them.

Environment

Fortunately LaTeX can nowadays be found for any (modern) computer environment, be it Linux, Windows, or Macintosh. For Linuxes (and other Unix clones) and Macs, I'd recommend *TeX Live* [**?**], which is the current default LaTeX distribution for many Linux flavors such as Fedora, Debian, Ubuntu, and Gentoo. TeX Live is the replacement for the older *teTeX*, which is no longer developed.

TeX Live works also for Windows machines (at least according to their web site); however, I have used *MiKTeX* [**?**] and can recommend it for Windows. MiKTeX has a nice package manager and automatically fetches missing packages for you.

Editor

You can write LaTeX documents with any text editor you like, but having syntax coloring options and such really helps a lot. My personal favourite

for editing LaTeX is the *TeXlipse* [**?**] plugin for the Eclipse IDE [**?**]. Eclipse is an open-source integrated development environment (IDE) initially created for writing Java code, but it currently has support for editing languages such as C, C++, JavaScript, XML, HTML, and many more. The TeXlipse plugin allows you to edit and compile LaTeX documents directly in Eclipse, and compilation errors and warnings are shown in the Eclipse *Problems* dialog so that you can locate and fix the issues easily. The plugin also supports reference traversal so that you can locate the source line where a label or a citation is defined.

Eclipse is an entire development environment, so it may feel a bit heavy-weight for editing a document. If you are looking for a more light-weight option, check out TeXworks. TeXworks is a LaTeX editor that is packaged with the newer MiKTeX distributions, and it can be acquired from `http://www.tug.org/texworks/`.

And if you are attached to your *emacs* or *vim* editor, you can of course edit your LaTeX documents with them. Emacs at least has syntax coloring and you can compile your document with a key binding, so this may be a good option if you prefer working with the standard Linux text editors.

Graphics

When you use `pdflatex` to render your thesis, you can include PDF images directly, as shown by Figure 3.1 below.

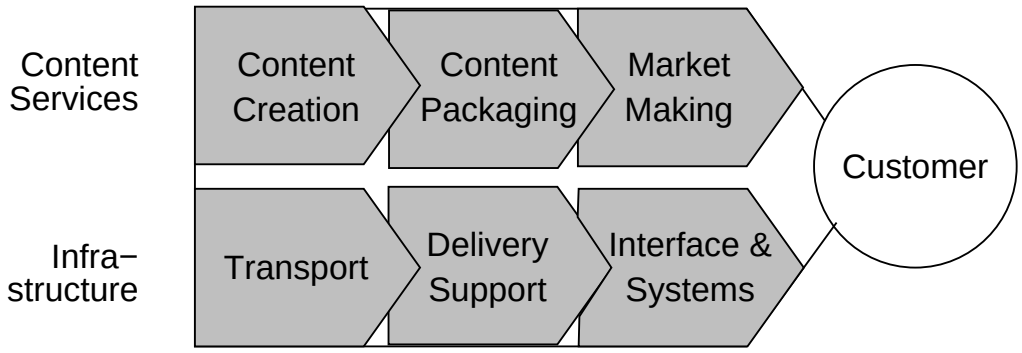


Figure 3.1: The INDICA two-layered value chain model.

You can also include JPEG or PNG files, as shown by Figure 3.2.

You can create PDF files out of practically anything. In Windows, you can download PrimoPDF or CutePDF (or some such) and install a printing

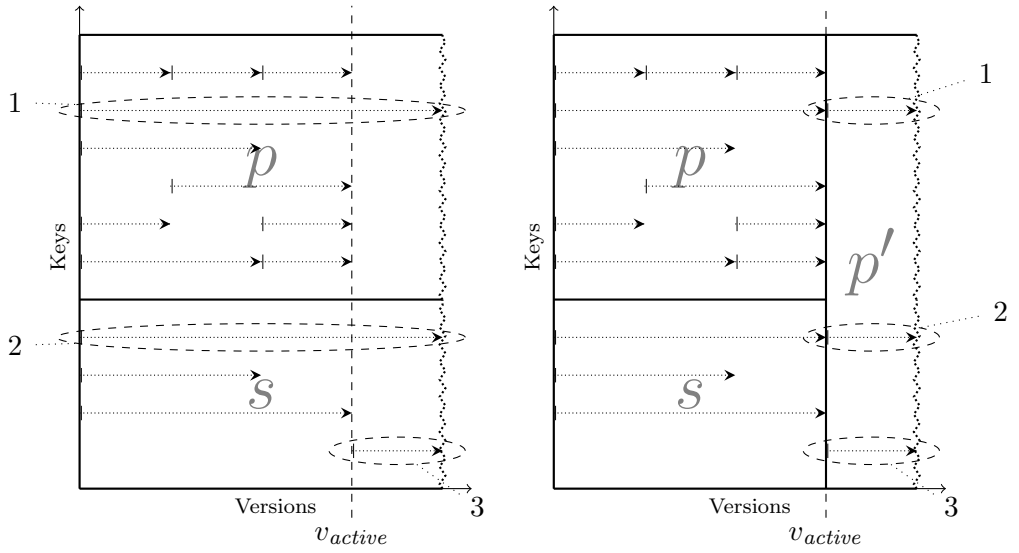
Figure 3.2: Eeyore, or Ihaa, a very sad donkey.

driver so that you can print directly to PDF files from any application. There are also tools that allow you to upload documents in common file formats and convert them to the PDF format. If you have PS or EPS files, you can use the tools `ps2pdf` or `epspdf` to convert your PS and EPS files to PDF.

Furthermore, most newer editor programs allow you to save directly to the PDF format. For vector editing, you could try Inkscape, which is a new open source WYSIWYG vector editor that allows you to save directly to PDF. For graphs, either export/print your graphs from OpenOffice Calc/Microsoft Excel to PDF format, and then add them; or use `gnuplot`, which can create PDF files directly (at least the new versions can). The terminal type is *pdf*, so the first line of your plot file should be something like `set term pdf ...`.

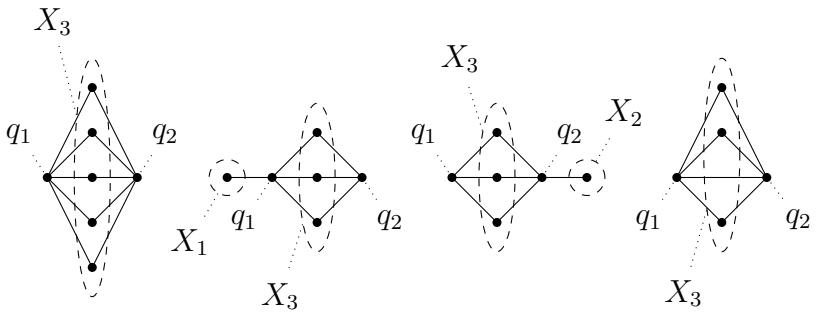
To get the most professional-looking graphics, you can encode them using the TikZ package (TikZ is a frontend for the PGF graphics formatting system). You can create practically any kind of technical images with TikZ, but it has a rather steep learning curve. Locate the manual (`pgfmanual.pdf`) from your LaTeX distribution and check it out. An example of TikZ-generated graphics is shown in Figure 3.3.

Another example of graphics created with TikZ is shown in Figure 3.4. These show how graphs can be drawn and labeled. You can consult the example images and the PGF manual for more examples of what kinds figures you can draw with TikZ.

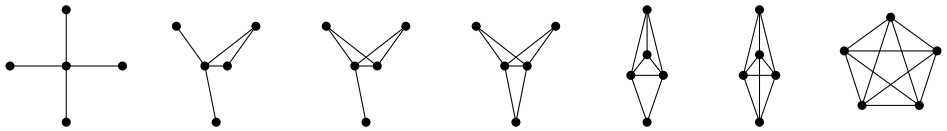


(a) Before (b) After

Figure 3.3: Example of a multiversion database page merge. This figure has been taken from the PhD thesis of Haapasalo [**?**].



(a) Examples of obstruction graphs for the Ferry Problem



(b) Examples of star graphs

Figure 3.4: Examples of graphs draw with TikZ. These figures have been taken from a course report for the graph theory course [**?**].

Chapter 4

Methods

You have now stated your problem, and you are ready to do something about it! *How* are you going to do that? What methods do you use? You also need to review existing literature to justify your choices, meaning that why you have chosen the method to be applied in your work.

If you have not yet done any (real) metholodogical courses (but chosen introduction courses of different areas that are listed in the methodological courses list), now is the time to do so or at least check through material of suitable methodological courses. Good methodologial courses that consentrates especially to methods are presented in Table 4.1. Remember to explain the content of the tables (as with figures). In the table, the last column gives the research area where the methods are often used. Here we used table to give an example of tables. Abbreviations and Acronyms is also a long table. The difference is that longtables can continue to next page.

Code	**Name**	**Methods**	**Area**
T-110.6130	Systems Engineering for Data Communications Software	Computer simulations, mathematical modeling, experimental research, data analysis, and network service business research methods, (agile method)	T-110
Mat-2.3170 Simulation (here is an example of multicolumn for tables)		Details of how to build simulations	T-110
S-38.3184	Network Traffic Measurements and Analysis	How to measure and analyse network traffic	T-110

Table 4.1: Research methodology courses

Chapter 5

Implementation

You have now explained how you are going to tackle your problem. Go do that now! Come back when the problem is solved!

Now, how did you solve the problem? Explain how you implemented your solution, be it a software component, a custom-made FPGA, a fried jelly bean, or whatever. Describe the problems you encountered with your implementation work.

Chapter 6

Evaluation

You have done your work, but that's[1] not enough.

You also need to evaluate how well your implementation works. The nature of the evaluation depends on your problem, your method, and your implementation that are all described in the thesis before this chapter. If you have created a program for exact-text matching, then you measure how long it takes for your implementation to search for different patterns, and compare it against the implementation that was used before. If you have designed a process for managing software projects, you perhaps interview people working with a waterfall-style management process, have them adapt your management process, and interview them again after they have worked with your process for some time. See what's changed.

The important thing is that you can evaluate your success somehow. Remember that you do not have to succeed in making something spectacular; a total implementation failure may still give grounds for a very good master's thesis—if you can analyze what went wrong and what should have been done.

[1] By the way, do *not* use shorthands like this in your text! It is not professional! Always write out all the words: "that is".

Chapter 7

Discussion

At this point, you will have some insightful thoughts on your implementation and you may have ideas on what could be done in the future. This chapter is a good place to discuss your thesis as a whole and to show your professor that you have really understood some non-trivial aspects of the methods you used...

Chapter 8

Conclusions

Time to wrap it up! Write down the most important findings from your work. Like the introduction, this chapter is not very long. Two to four pages might be a good limit.

Bibliography

[1] Barakat, C., Thiran, P., Iannaccone, G., Diot, C., and Owezarski, P. Modeling internet backbone traffic at the flow level. *IEEE Trans. Signal Processing 51*, 8 (2003), 2111–2124.

[2] Barroso, L. A., Clidaras, J., and Hölzle, U. *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines, Second Edition*. Synthesis Lectures on Computer Architecture. Morgan & Claypool Publishers, 2013.

[3] Barroso, L. A., and Hölzle, U. The case for energy-proportional computing. *IEEE Computer 40*, 12 (2007), 33–37.

[4] Bolla, R., Bruschi, R., Davoli, F., and Cucchietti, F. Energy efficiency in the future internet: A survey of existing approaches and trends in energy-aware fixed network infrastructures. *IEEE Communications Surveys and Tutorials 13*, 2 (2011), 223–244.

[5] Carneiro, G., Fortuna, P., and Ricardo, M. Flowmonitor: a network monitoring framework for the network simulator 3 (NS-3). In *4th International Conference on Performance Evaluation Methodologies and Tools, VALUETOOLS '09, Pisa, Italy, October 20-22, 2009* (2009), p. 1.

[6] Casanova, H., Giersch, A., Legrand, A., Quinson, M., and Suter, F. Versatile, scalable, and accurate simulation of distributed applications and platforms. *J. Parallel Distrib. Comput. 74*, 10 (2014), 2899–2917.

[7] Claffy, K., Miller, G., and Thompson, K. The nature of the beast: Recent traffic measurements from an internet backbone. In *Proceedings of INET* (1998), vol. 98, pp. 21–24.

[8] Dayarathna, M., Wen, Y., and Fan, R. Data center energy consumption modeling: A survey. *IEEE Communications Surveys and Tutorials 18*, 1 (2016), 732–794.

[9] Fan, X., Weber, W., and Barroso, L. A. Power provisioning for a warehouse-sized computer. In *34th International Symposium on Computer Architecture (ISCA 2007), June 9-13, 2007, San Diego, California, USA* (2007), pp. 13–23.

[10] Fiandrino, C., Kliazovich, D., Bouvry, P., and Zomaya, A. Y. Performance metrics for data center communication systems. In *8th IEEE International Conference on Cloud Computing, CLOUD 2015, New York City, NY, USA, June 27 - July 2, 2015* (2015), pp. 98–105.

[11] Gyarmati, L., and Trinh, T. A. How can architecture help to reduce energy consumption in data center networking? In *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking, e-Energy 2010, Passau, Germany, April 13-15, 2010* (2010), pp. 183–186.

[12] Heddeghem, W. V., Lambert, S., Lannoo, B., Colle, D., Pickavet, M., and Demeester, P. Trends in worldwide ICT electricity consumption from 2007 to 2012. *Computer Communications 50* (2014), 64–76.

[13] Kliazovich, D., Bouvry, P., and Khan, S. U. Greencloud: a packet-level simulator of energy-aware cloud computing data centers. *The Journal of Supercomputing 62*, 3 (2012), 1263–1283.

[14] LLC, M. NS-3 simulator, 2017.

[15] Mahadevan, P., Sharma, P., Banerjee, S., and Ranganathan, P. A power benchmarking framework for network devices. In *NETWORKING 2009, 8th International IFIP-TC 6 Networking Conference, Aachen, Germany, May 11-15, 2009. Proceedings* (2009), pp. 795–808.

[16] Orgerie, A., Lefèvre, L., Lassous, I. G., and López-Pacheco, D. M. ECOFEN: an end-to-end energy cost model and simulator for evaluating power consumption in large-scale networks. In *12th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WOWMOM 2011, Lucca, Italy, 20-24 June, 2011* (2011), pp. 1–6.

Appendix A

First appendix

This is the first appendix. You could put some test images or verbose data in an appendix, if there is too much data to fit in the actual text nicely.

For now, the Aalto logo variants are shown in Figure A.1.

(a) In English

(b) Suomeksi

(c) På svenska

Figure A.1: Aalto logo variants