

Masterthesis

# L<sup>A</sup>T<sub>E</sub>X Template for Thesis Documents

Vorname Nachname

Course of Studies: Angewandte Informatik  
Faculty: Computer Science and Engineering  
Esslingen University

Wintersemester 2022/23

Period: 15.05.2023 - 15.11.2023

Reviewer: Erstprüfer mit korrektem Prof. und Dr. Titel

Second Reviewer: Zweitprüfer mit korrektem Prof. und Dr. Titel

---

Company: Firmenname

Supervisor: Betreuer in der Firma

# Erweiterte Eigenständigkeitserklärung zur Verwendung generativer KI-Systeme als Hilfsmittel

Hiermit versichere ich, dass ich diese Prüfungsleistung selbständig verfasst habe. Die verwendeten Quellen und Hilfsmittel habe ich angegeben.

Sofern ich generative KI-Systeme über die Prüfung von Grammatik und Rechtschreibung von eigenständig erstelltem Text hinaus genutzt habe, bestätige ich hiermit, dass ich:

- ☐ die von generativen KI-Systemen direkt übernommenen Inhalte im inhaltlichen Teil der Arbeit entsprechend markiert bzw. zitiert habe,
- ☐ sichergestellt habe, dass die mit Hilfe der genannten generativen KI-Systemen erzeugten und sonstigen von mir verwendeten Inhalte faktisch korrekt sind,
- ☐ mir darüber im Klaren bin, dass ich als Verfasserin oder Verfasser dieser Arbeit die Verantwortung für die darin enthaltenen Angaben und Aussagen übernehme,
- ☐ sofern ich generative KI-Systeme zur Unterstützung wesentlicher Arbeitsschritte eingesetzt habe, die nicht bereits im inhaltlichen Teil der Arbeit oder den Referenzen thematisiert bzw. zitiert werden, dies tabellarisch erfasst habe. Die Tabelle habe ich mit Abgabe der Arbeit eingereicht (siehe Anhang).

Esslingen, den November 12, 2025

\_\_\_\_\_  
Unterschrift

# Sperrvermerk

Die nachfolgende **Masterthesis** enthält vertrauliche Daten der **Firmenname**. Veröffentlichungen oder Vervielfältigungen dieser Arbeit – auch nur auszugsweise – sind ohne ausdrückliche Genehmigung der **Firmenname** nicht gestattet. Diese Arbeit ist nur den Prüfern, sowie den Mitgliedern des Prüfungsausschusses, zugänglich zu machen.

**Sperrvermerk is an optional page, and depends on the company requirements. Make sure to discuss with your professor and company supervisor.**

# Acknowledgement

If you like, you can add a thanks to the profs, company and supervisor. Additionally, thanks to the family and friends :)

# Abstract

As a first step towards your successful thesis, you need to evaluate a few things. First, is the topic of value for you and do you feel motivated to work on this project for either three or six months? This does not only mean that you work in a specific field for this period, but before you start, you already have a clear vision on how to start. Of course, things change over time, and it would be boring, if you would already know everything in advance. However, prepare yourself with research and discussions with your supervisor to be on top of the topic. Second, you need to document your thesis in a way that **people can follow your thoughts and understand decisions** you had to take on the way. This includes **clear writing** in a useful structure, where your supervisor will help you with. But it also includes the choice of language: English or German? (others are not accepted) Of course, English is preferred, as it fits best to the scientific environment where you will position yourself with this thesis. But you can still choose German to create a document without losing time if your English is not fluent or too stable. In this document, you need to set the selected language at two different positions:

- A) In the files *template/title.tex* and *template/metadata.tex*, you need to adjust the information on your thesis to generate a proper front page.
- B) In *template/packages.tex* you need to set the correct option for the *babel* package by either using `\usepackage[english]{babel}` for english or `\usepackage[ngerman]{babel}` for german. If you want to use umlauts with e.g. packages like *hyphenat* you should use a font encoding with good support of accents. To do this, put `\usepackage[T1]{fontenc}` above the `\usepackage[ngerman]{babel}` command.
- C) In *thesis.tex* you need to change the strings starting with "List of" to german and add `\renewcommand{\listalgorithmname}{Algorithmusverzeichnis}` above `\addcontentsline{toc}{chapter}{\listalgorithmname}`.

# Contents

<b>List of Figures</b>	<b>III</b>
<b>List of Tables</b>	<b>IV</b>
<b>List of Listings</b>	<b>V</b>
<b>List of Algorithms</b>	<b>VI</b>
<b>List of Acronyms</b>	<b>VII</b>
<b>List of TODOs</b>	<b>VIII</b>
<b>1. Introduction</b>	<b>1</b>
1.1. Motivation . . . . .	1
1.2. Goals . . . . .	1
1.3. Structure . . . . .	1
<b>2. Background</b>	<b>2</b>
2.1. metadata.tex and title.tex . . . . .	2
2.2. packages.tex . . . . .	2
2.3. macros.tex . . . . .	2
2.3.1. todonotes . . . . .	2
2.3.2. Text Suggestions . . . . .	3
2.3.3. Marked Text . . . . .	3
2.3.4. Thesis Goals and Numbered Elements . . . . .	3
2.4. Acronyms . . . . .	3
2.5. Finalize your Document . . . . .	4
2.5.1. Remove unused Tables of Entries . . . . .	4
2.5.2. Preparing for the Print . . . . .	4
<b>3. Related Work</b>	<b>5</b>
3.1. Structure . . . . .	5
3.2. Searching for Literature . . . . .	5
3.3. Writing Scientific Text . . . . .	6
3.3.1. In German . . . . .	6
3.3.2. In English . . . . .	6
3.4. Tools to Support Writing . . . . .	6
3.4.1. Tools for Writing . . . . .	6
3.4.2. Tools to Organize References . . . . .	7
<b>4. Design</b>	<b>8</b>
<b>5. Implementation</b>	<b>10</b>
<b>6. Evaluation</b>	<b>11</b>

## *Contents*

<b>7. Conclusion</b>	<b>13</b>
7.1. Summary . . . . .	13
7.2. Future Work . . . . .	13
<b>8. Bibliography</b>	<b>i</b>
<b>A. Appendix</b>	<b>ii</b>
A.1. A Filter and Sort Function in Python . . . . .	ii
<b>Anhang der erweiterten Eigenständigkeitserklärung</b>	<b>iii</b>

## List of Figures

3.1. An example search in Google Scholar . . . . .	5
3.2. VS Code Plugin: LaTeX Workshop . . . . .	7
3.3. VS Code Plugin: LaTeX Utils . . . . .	7
3.4. VS Code Plugin: LTeX - LanguageTool . . . . .	7
6.1. Single node queuing delay evaluation . . . . .	12



# List of Tables

4.1. Example transmission delays. . . . .	8
6.1. Single Node Evaluation Settings . . . . .	11

# List of Listings

5.1. A filter function in Python . . . . .	10
A.1. A filter and sort function in Python . . . . .	ii

# List of Algorithms

4.1. Group Sort . . . . . 9

# List of Acronyms

This document is incomplete. The external file associated with the glossary ‘acronym’ (which should be called `thesis.acr`) hasn’t been created.

Check the contents of the file `thesis.acn`. If it’s empty, that means you haven’t indexed any of your entries in this glossary (using commands like `\gls` or `\glsadd`) so this list can’t be generated. If the file isn’t empty, the document build process hasn’t been completed.

Try one of the following:

- Add `automake` to your package option list when you load `glossaries-extra.sty`.  
For example:

```
\usepackage[automake]{glossaries-extra}
```

- Run the external (Lua) application:

```
makeglossaries-lite.lua "thesis"
```

- Run the external (Perl) application:

```
makeglossaries "thesis"
```

Then rerun `LATEX` on this document.

This message will be removed once the problem has been fixed.

# Todo list

Short TODO (*Vorname*) . . . . . 2

This is a very long TODO note that might go across multiple lines. You can really  
write some meaningful text in here and still read it later on. (*Vorname*) . . . 2

REVIEW . . . . . 3

FEEDBACK Is this a correct use of a feedback mark? . . . . . 3

Add some information on German texts (*Vorname*) . . . . . 6

Add some information on English texts (*Vorname*) . . . . . 6

Include details on tool support (*Vorname*) . . . . . 6

Include details on Zotero (*Vorname*) . . . . . 7

# 1. Introduction

This L<sup>A</sup>T<sub>E</sub>Xtemplate is just a guideline to get started with your thesis, it is not a one fits all solution. For further ideas and tips, please refer to additional literature, like “How to Write a Better Thesis” from Evans et al. [1].

## 1.1. Motivation

In most cases, your introduction will start with a motivation section. This section summarizes an existing problem and presents the environment for the solution. This environment also highlights the technical boundaries of your design and implementation.

## 1.2. Goals

After the motivation, you can shortly summarize the main goals of the thesis. These goals enable an evaluation that measures your implementation against these goals. For each goal, you can always highlight the evaluation technique to verify the achievement.

The following approach is fully optional and only recommended, if you have clear and distinct goals for your thesis. You can list the goals with an explicit label as follows.

**Goal (G1)** *This is an example goal to test the counter of the goal for correct references.*

**Goal (G2)** *Students should have a seamless start in writing the thesis, without unnecessary L<sup>A</sup>T<sub>E</sub>X troubles.*

You can reference the goals from anywhere in the document and always link decisions to original goals. For example, we write this document to fulfill Goal 2.

## 1.3. Structure

Finally, you should quickly highlight the structure of your thesis and the solution, but keep it short and leave out details. It is important that you already present a little insight into your structure, as the reader can follow easier through background and related work.

## 2. Background

The background section guides you through the use of the template and some features we included to make writing a little easier.

### 2.1. metadata.tex and title.tex

The file *template/metadata.tex* includes all information on you, the title of the thesis and supervisors. This information is used during the compilation of the template to fill out all the details. Especially in the file *thesis\_template/title.tex*, you will see the use of this metadata. Please adjust this title page and the metadata to fit to your thesis topic, thesis category, company, and supervisors.

### 2.2. packages.tex

The file *template/packages.tex* includes all used packages. Please only add your packages here! This makes debugging of conflicts between packages easier, as you have them all together.

### 2.3. macros.tex

This section introduces the use of the predefined macros. You are free to extend this list of macros in the file *template/macros.tex*.

#### 2.3.1. todonotes

*todonotes* is a very useful package to track open items within a document and not mess around with special identifies to search for your TODOs.

#### List of TODOs

In the *thesis.tex* document, you can find the `\listoftodos` line. This generates a list of all TODOs defined in this document.

#### TODOs

In your document, you have two options to mark TODOs on your text. The first option is the one for short notes: `\todonote`. This `todonote` will pop up on the side of the document and point to the location in the paragraph, where it is linked to. Do not use it for long texts. In case you have long notes that you want to remember, you can use `\todoinline`.

This is a very long TODO note that might go across multiple lines. You can really write some meaningful text in here and still read it later on. (*Vorname*)

You will notice, that after such an inline TODO, the text is indented, even if you did not create the paragraph yourself. But as you are on a work-in-progress document, this does not matter too much.

Short  
TODO  
(*Vor-  
name*)

### REVIEW

#### REVIEW

The review notice shows that a section is ready for review. Use it to show your supervisor that you would like to get feedback for this section. Just add the `\review` command after the headline and the according information will be inserted into the PDF document.

### FEEDBACK

#### FEEDBACK Is this a correct use of a feedback mark?

If you want to document a question you have for your supervisor, just use the command `\feedback`. This will generate an inline comment with the note that feedback is required.

### 2.3.2. Text Suggestions

We developed a macro to indicate suggestions for change. This macro keeps the old text striked through and prints the suggested text colored in olive next to it. For example, ~~this is the old text~~ this is the new text .

### 2.3.3. Marked Text

You do not need to create TODOs for all elements that you need to recheck. For example, you can just mark text sections with the command `\marktext` and the macros will highlight this text for you. These are not listed in the list of todos.

### 2.3.4. Thesis Goals and Numbered Elements

In the file *template/macros.tex*, we included the definition of goals for your thesis. You can use the numbering of goals through the section *goal* as used in Section 1.2. If necessary, you can also create such numbering for other elements that come up in your thesis. One student started the thesis by defining anomalies that should be detected with an Intrusion Detection System (IDS). He used this mechanism to reference the individual anomalies.

## 2.4. Acronyms

When writing your thesis, you might use a lot of acronyms. In order to have the fully written on the first use, and after that only use the short version you would need to go through the document at the end. Therefore, we recommend to use the following package: *glossaries-extra* with the module *acronym*. In the file *content/acronyms.tex*, we defined a few acronyms as example and will explain the use in the following.

In your text, you should always use the command `\gls{gPTP}` to call an acronym. With that, the first appearance will include both, short and long form (to visualize the effect, we mark the output red): Generalized Precision Time Protocol (gPTP) With the second use, you will only have the short form: gPTP

If you need to use the full term, you can use the command `\acrlong{gPTP}`: Generalized Precision Time Protocol. Alternatively, you can force the short version with the command `\acrshort{gPTP}`: gPTP.

To update the list of acronyms, you must enter the following command in your terminal: `makeglossaries thesis` Please remember to do that also before you generate your final version.



## 2.5. Finalize your Document

### 2.5.1. Remove unused Tables of Entries

In the *thesis.tex*, you find a number of tables for different kinds of entries (see `\listof.....`). Depending on if you use Tables or Code snippets, you need them, or you don't. Please remove all unused ones, such that you do not have empty ones in the final document. Also make sure, that the `\listoftodos` is empty and then remove it as well.

### 2.5.2. Preparing for the Print

If you want to print your document with a binding, you should adjust the document, depending on the type of print. Most likely, you will have a binding on the left. Therefore, you need to adjust the use of the package *geometry* in *template/packages.tex* to include the *bindingoffset=15mm*. Otherwise, you will squeeze your text into the binding on the left. Additionally, you need to choose, if you will print on both sides of the paper or only on the front side. For the front side only, you need to set the option *oneside* in the first row of the *thesis.tex* file. Make sure, that you configure the printer to only use the front side. Alternatively, you can tell the document to start chapters on the right-hand side and also print on the back side of the paper. For this, you need to set the option *twoside* in the first row of the *thesis.tex* file instead of the *oneside*.

## 3. Related Work

The related work section is often underrated. Make sure that you focus on this topic BEFORE your thesis, but do not leave it out of sight during and after the thesis.

### 3.1. Structure

The related work section can have different structures:

**Grouped by Topic** The easiest way for you is to group the related work by topic. This means that you group the literature by common attributes and discuss them together.

**Grouped by Relevance** You can also group the work by relevance. This means, you start with work impacting your thesis the most. Towards the end of the section, you introduce related work least relevant.

We will give a small example on the grouping by topic in the following.

### 3.2. Searching for Literature

A related work chapter starts with the literature research. There are two main sources, and many others, which you should use in parallel! In general, you should not only use the keywords that you come up with first, but continue your search with synonyms or other phrasings for the same topic.

Google Scholar [2] is the largest search engine for scientific literature. Sometimes, you do not find the PDF version of a document directly, but below the documents, you can find the *All XYZ versions* link, as shown in Figure 3.1. Often, this helps to get the PDF version without a paywall.

Another good source for scientific literature is ResearchGate [3]. Many authors upload the full text versions of their papers here. Alternatively, you can request the full text by the authors (sadly, they often do not respond...).

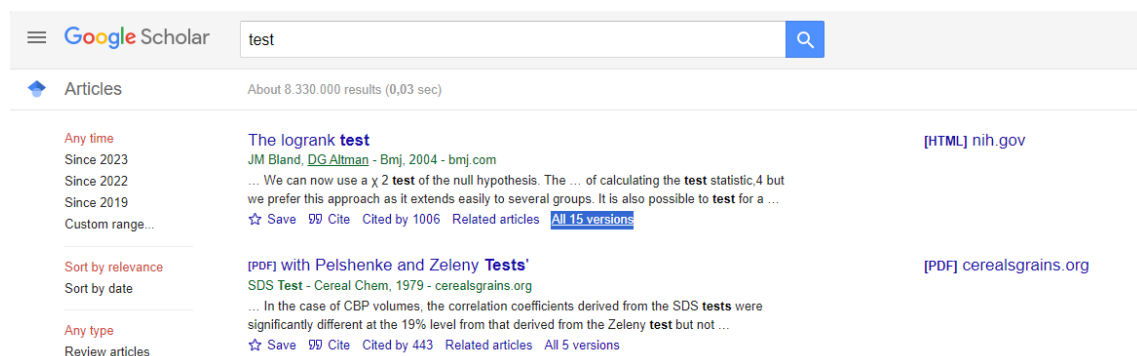


Figure 3.1.: An example search in Google Scholar

### 3. Related Work

You need to enter all references that you want to use in the *content/thesis.bib* file. Both, Google Scholar and ResearchGate give you export formats in BibTeX style, such that you can copy and paste the entry. Additionally, some tools presented in Section 3.4 do the same thing. However, please make sure to double-check the BibTeX entry yourself!! **One common mistake is that you assume the capitalization stays the same, but it does not.** Make sure to use double brackets `{{SOME TEXT}}` around the text you want to keep the capitalization. Finally, you can also use the library DBPL [4] to retrieve well maintained BibTeX entries for a very large number of scientific literature.

With scientific literature, it is important that you state the authors name, if you refer to their work in detail. For example, Müller et al. [5] present a scanning pipeline for the analysis of software versions on the internet. This pipeline includes banner grabbing to retrieve version information of services. It is important that you put the `~` between the name (always use the first author's last name and add *et al.* if there are multiple authors) and the `\cite`. To have a good understanding of the differences to your work, you **need** to put a relation between your work and the related work. Compared to Müller et al., we also include the probes for cryptographic configurations. With that, we are able to analyze the landscape of offered cypher suites in, e.g., TLS or OPC UA.

Alternatively, you can also group literature that is very similar. The modeling of TSN networks is well researched for numerous use cases [6, 7], including detailed forwarding latency, jitter, and interference models. However, none of these components models load dependent delays, such as firewalls. Therefore, we present the first model of a firewall to be included in TSN network simulation. We recommend grouping literature only, if they are far away from the own research, and you want to show that a certain field of research (not too important for you) is well covered.

### 3.3. Writing Scientific Text

In general, scientific text differs from what you learned in school. The text should not be entertaining or exciting, but present technical information. Therefore, do not be afraid to repeat technical terms, without finding synonyms. Also, keep your sentences simple and short (especially in English).

#### 3.3.1. In German

German texts are typically written in a passive form.

#### 3.3.2. In English

Writing in English is different from German. In English texts, you use active voice. A good source for detailed information is the book "Scientific Writing" from Justin Zobel [8]. Students from HS Esslingen can read it for free.

Add some information on German texts (**Vorname**)

### 3.4. Tools to Support Writing

Include details on tool support (**Vorname**)

Add some information on English texts (**Vorname**)

#### 3.4.1. Tools for Writing

A very useful tool for your thesis is Visual Studio Code (VS Code) [9]. With this tool, all programming, but also writing of the thesis is made easier! To write your thesis with

### 3. Related Work

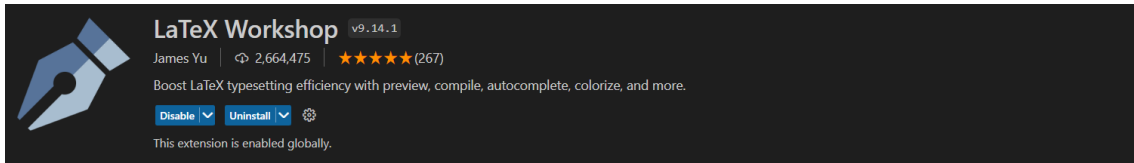


Figure 3.2.: VS Code Plugin: LaTeX Workshop

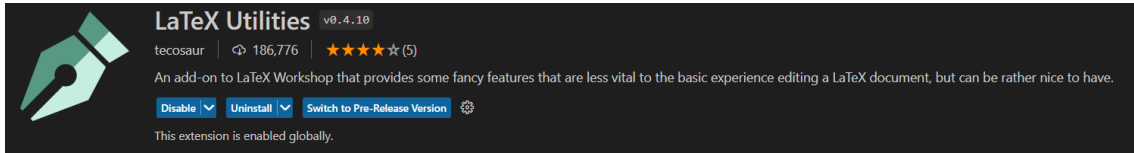


Figure 3.3.: VS Code Plugin: LaTeX Utils

VS-Code, we recommend the following extensions:

- VS Code Extension: LaTeX Workshop (c.f. Figure 3.2)
- VS Code Extension: LaTeX Utils (c.f. Figure 3.3)
- VS Code Extension: LTeX - LanguageTool (c.f. Figure 3.4)

With these extensions, VS Code will automatically compile the LaTeX code of this thesis template and the previewed PDFs are updated automatically in the background. You can view the compiled PDFs in VS-Code using the shortcut *cmd + alt + v*. Other useful shortcuts include:

- jumping from the code to the PDF:
  - **mac:** *cmd + option + j*
  - **windows/linux:** *ctrl + alt + j*
- jumping from the PDF to the code:
  - **mac:** *cmd + click*
  - **windows/linux:** *ctrl + click*

Additionally, *LTeX* will highlight grammar and spelling mistakes. Please follow the documentation of this plugin to change the language to something else than English if necessary.

#### 3.4.2. Tools to Organize References

We recommend the use of the tool Zotero [10] to manage your references.

Include details on Zotero (*Vorname*)

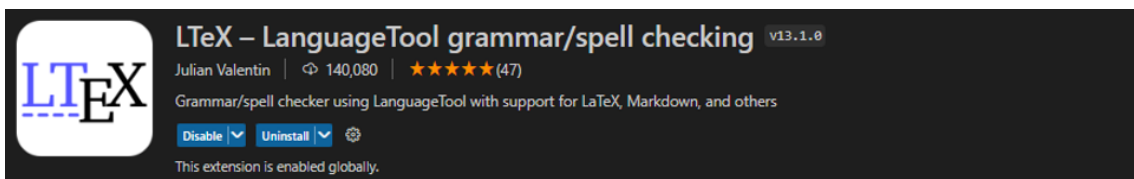


Figure 3.4.: VS Code Plugin: LTeX - LanguageTool

## 4. Design

The structure for the design chapter strongly depends on the topic of your thesis. Throughout the process, you will refine this structure with your supervisor.

If you have a mathematical content in your thesis, you can use equations as follows. Each equation uses its own space and should not be longer than one line, otherwise it is difficult to read. For this document, the content of the equation is not important. Still, we reference and explain it as an example. Equation 4.1 defines the set of streams  $\mathcal{S}_{\text{ifr}}^{e,s}$  that will interfere with stream  $s$  if no Time-Aware Shaper (TAS) is configured. For the purpose of consistent formulas, we introduce commands for complex terms in the file *content/definitions.tex*

$$\mathcal{S}_{\text{ifr}}^{e,s} = \{g | g \in \mathcal{S}^e \wedge g \neq s \wedge p_g \geq p_s\} \quad (4.1)$$

In your thesis, you can also use more complex equation structures. This structure still has one equation per line, but aligns them to be grouped together and indent the same way. The alignment happens through the  $\&$  across the lines. This complex structure still has references to each of the individual formulas. Therefore, we can easily explain each of the Equations 4.2, 4.3, and 4.4 in the text.

$$\text{case C1: } t_{\text{enqCT}}^{e,s} < t_{\text{open}}^{e,s} \quad (4.2)$$

$$\text{case C2: } t_{\text{enqCT}}^{e,s} + d_{\text{dwell}}^{e,s} + d_{\text{ifrcross}}^{e,s} < t_{\text{close}}^{e,s} \quad (4.3)$$

$$d_{\text{gate}}^{e,s} = \begin{cases} t_{\text{open}}^{e,s} - t_{\text{enqCT}}^{e,s} + d_{\text{ifpath}}^{e,s} & \text{case C1} \\ 0 & \text{case C2} \\ CT_{\text{GCL}}^{e,s} - t_{\text{enqCT}}^{e,s} + t_{\text{open}}^{e,s} + d_{\text{ifpath}}^{e,s} & \text{otherwise} \end{cases} \quad (4.4)$$

For your thesis, we recommend the use of tables to summarize series of data. Sometimes, tables are easier to read than a lot of text with data. In Table 4.1, we present the transmission delay of frames with different sizes at different link speeds. Instead of writing all these numbers in a long paragraph, we can now use the time to highlight key insights. For example, you can see that a frame with 1,522 B only consumes about twice the time at 1 Gbit/s, compared to a frame with size 64 B at 100 Mbit/s.

Table 4.1.: Example transmission delays.

Size	100 Mbit/s	1 Gbit/s	2.5 Gbit/s	10 Gbit/s
<b>64 B</b>	6.7 $\mu\text{s}$	672 ns	269 ns	67 ns
<b>123 B</b>	11.8 $\mu\text{s}$	1.2 $\mu\text{s}$	470 ns	118 ns
<b>1,522 B</b>	123.4 $\mu\text{s}$	12.3 $\mu\text{s}$	4.9 $\mu\text{s}$	1.2 $\mu\text{s}$

#### 4. Design

In your thesis, you might want to explain your solution with Pseudocode algorithms. For example, Algorithm 4.1 presents a sorting algorithm for firewall rules. In Line 4, we compare if a rule depends on the current group.

---

**Algorithm 4.1** Group Sort

---

```
1: for  $x \in RuleSet$  do
2:    $group \leftarrow [x]$ 
3:   for  $y \in RuleSet[x.idx :]$  do
4:     if  $dependency(group, y)$  then ▷ Extend group
5:        $group \leftarrow group \# y$ 
6:     else
7:       if  $hit\_count([y]) < hit\_count(group)$  then ▷ Move complete group
8:          $y.idx \leftarrow x.idx$ 
9:         for  $z \in group$  do
10:           $z.idx \leftarrow z.idx + 1$ 
11:        end for
12:      end if
13:       $group \leftarrow [y]$ 
14:    end if
15:  end for
16: end for
```

---

Finally, you can also use theorems, and lemmas in your thesis. Please make sure (as always) to reference each of them in the text and explain their meaning. For example, Theorem 4.1, or Lemma 4.2.

**Theorem 4.1.** *Let  $f$  be a function whose derivative exists in every point, then  $f$  is a continuous function.*

**Lemma 4.2.** *Given two line segments whose lengths are  $a$  and  $b$  respectively there is a real number  $r$  such that  $b = ra$ .*

For formal procedures and algorithms, use the unified notation of proofs to state your conclusions. An example for that follows here:

*Proof.* To prove it by contradiction try and assume that the statement is false, proceed from there and at some point you will arrive to a contradiction. □

## 5. Implementation

In your implementation section, you can also use code listings to explain the details of your work. To include the code you have two options: A) use the inline method for short one liner: `[elem for elem in my\_list if elem.variable==filter\_value]`, or B) use multiline listings to visualize more complex examples. Similar with figures, make sure to always reference your multiline listings and explain what they do. In Listing 5.1, you find the function `my_filter` with two parameters. This function uses the list in the first parameter for all elements with the variable equal to the second parameter.

```
1 def my_filter(my_list, filter_value):
2     new_list = []
3     for elem in my_list:
4         if elem.variable == filter_value:
5             new_list.append(elem)
6     return new_list
```

Listing 5.1: A filter function in Python

If you have even more detailed code listings, they belong into the appendix. For example, you can find a filter function with an additional sorting method in the Appendix A.1.

In some cases, it might happen that your inline code does not fit into the current line anymore. For example with this code here: `[elem for elem in my\_list if elem.variable==filter\_value]` But don't worry, the arrow in the beginning of the line will highlight that!

If you require further details on the use of listings in LaTeX, you can find additional information on listings at the following two locations:

1. [https://en.wikibooks.org/wiki/LaTeX/Source\\_Code\\_Listings](https://en.wikibooks.org/wiki/LaTeX/Source_Code_Listings)
2. [https://www.overleaf.com/learn/latex/Code\\_listing](https://www.overleaf.com/learn/latex/Code_listing)

## 6. Evaluation

The content, e.g., measurements, survey, or simulations, of the evaluation depends on your thesis. What will not depend on your thesis that you have results that you will visualize in figures. In general, use one figure after each other and explain how to read it and directly discuss the content and **meaning**. Sometimes, you want to visualize a change through your design and implementation, or want to compare two systems. Then, you should place the figures either above each other or next to each other. The LaTeX package *subcaption* helps with that, and it looks as follows.

Figure 6.1 shows two figures with different configurations during the measurement. Table 6.1 presents six different settings for the evaluation of a single node. We use these settings throughout the evaluation with the labels S1 to S6. Figure 6.1a shows the measured forwarding delay for the six different settings without interference on a single node. Each measurement contains at least 1,000 packets and is visualized as box plot. Figure 6.1b shows measurements for the same six settings, but this time with interference by other traffic. One can clearly see that the majority of the results (the box represents 50% of all measurements) is similarly distributed for the settings S4 to S6. However, we also see a lot of outliers, raising the worst-case latency to  $80\mu\text{s}$ . For the settings S1 to S3, we see the same outliers, caused by the interference. Across all settings, we observe the settings S2 and S3 having the best performance, without and with interference.

Table 6.1.: Single Node Evaluation Settings

Setting	Description
S1	Strict Priority
S2	FP with priority 7 in the express category
S3	synchronized TAS with prio. 7 in the TAS window
S4	unsynchronized TAS with prio. 7; $CT_{app}^s = 100\mu\text{s}$
S5	unsynchronized TAS with prio. 7; $CT_{app}^s = 45\mu\text{s}$
S6	unsynchronized TAS with prio. 7; $CT_{app}^s = 196\mu\text{s}$



## 6. Evaluation

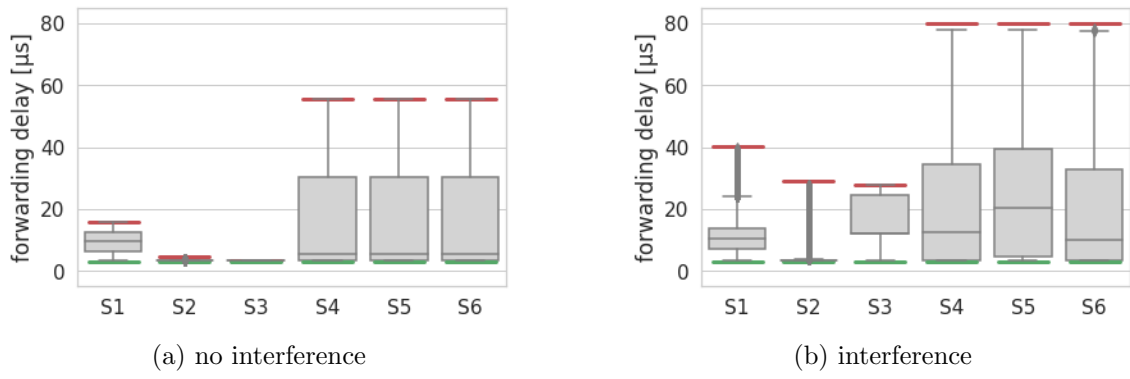


Figure 6.1.: Single node queuing delay evaluation

## **7. Conclusion**

The conclusion is the last chapter of your thesis. Make sure to match its content to your design, implementation, and evaluation. Also, that you recap your introduction and motivation to ensure you solved the actual problem.

### **7.1. Summary**

The summary recaps the complete thesis. Make sure to highlight the most important pieces of your work with a reference to the introduction. Additionally, present the applicability of your design and implementation.

### **7.2. Future Work**

Which tasks, problems, or ideas are still open? In the future work section, you should highlight these open points and give reasons for their relevance.

## 8. Bibliography

- [1] D. Evans, P. Gruba, and J. Zobel, *How to Write a Better Thesis*. Springer, 2014. [Online]. Available: <https://doi.org/10.1007/978-3-319-04286-2>
- [2] Google, “Google Scholar,” 2023. [Online]. Available: <https://scholar.google.com/>
- [3] ResearchGate GmbH, “Research Gate,” 2023. [Online]. Available: <https://www.researchgate.net/>
- [4] Schloss Dagstuhl - Leibniz Center for Informatics, “International Workshop/Symposium on Database Programming Languages (DBPL),” 2023. [Online]. Available: <https://dblp.org/db/conf/dbpl/index.html>
- [5] R. Müller, J. Ruppert, K. Will, L. Wüsteney, and T. Heer, “Analyzing the software patch discipline across different industries and countries,” in *Sicherheit, Schutz und Zuverlässigkeit: Konferenzband der 11. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI), Sicherheit 2022, Karlsruhe, Germany, April 5-8, 2022*, ser. LNI, C. Wressnegger, D. Reinhardt, T. Barber, B. C. Witt, D. Arp, and Z. Á. Mann, Eds., vol. P-323. Gesellschaft für Informatik, Bonn, 2022, pp. 159–170. [Online]. Available: [https://doi.org/10.18420/sicherheit2022\\_10](https://doi.org/10.18420/sicherheit2022_10)
- [6] D. Hellmanns, L. Haug, M. Hildebrand, F. Dürr, S. Kehrer, and R. Hummen, “How to optimize joint routing and scheduling models for TSN using integer linear programming,” in *RTNS’2021: 29th International Conference on Real-Time Networks and Systems, Nantes, France, April 7-9, 2021*, A. Queudet, I. Bate, and G. Lipari, Eds. ACM, 2021, pp. 100–111. [Online]. Available: <https://doi.org/10.1145/3453417.3453421>
- [7] L. Wüsteney, D. Hellmanns, M. Schramm, L. Osswald, R. Hummen, M. Menth, and T. Heer, “Analyzing and modeling the latency and jitter behavior of mixed industrial TSN and detnet networks,” in *Proceedings of the 18th International Conference on emerging Networking EXperiments and Technologies, CoNEXT 2022, Roma, Italy, December 6-9, 2022*, G. Bianchi and A. Mei, Eds. ACM, 2022, pp. 91–109. [Online]. Available: <https://doi.org/10.1145/3555050.3569138>
- [8] J. Zobel, *Writing for Computer Science*. Springer, 2014. [Online]. Available: <https://doi.org/10.1007/978-1-4471-6639-9>
- [9] Microsoft, “Visual Studio Code,” 2023. [Online]. Available: <https://code.visualstudio.com/>
- [10] Corporation for Digital Scholarship, “zotero,” 2023. [Online]. Available: <https://www.zotero.org/>

# A. Appendix

## A.1. A Filter and Sort Function in Python

```
1 def my_filter_with_sort(my_list, filter_value, reverse=False):
2     new_list = []
3     for elem in my_list:
4         if elem.variable_a == filter_value:
5             new_list.append(elem)
6     new_list.sort(key=lambda x : x.variable_b, reverse=reverse)
7     return new_list
```

Listing A.1: A filter and sort function in Python

Even in the appendix, never just paste code without referencing it from the text. Listing A.1 presents a function that filters a list for all elements containing a specific value in *variable\_a*. Finally, the function sorts the remaining elements based on the variable *variable\_b*. With the parameter *reverse*, you can specify if the sorting should be ascending or descending.

# Anhang der erweiterten Eigenständigkeitserklärung

Arbeitsschritt <sup>1</sup>	Eingesetzte(s) Gen-KI- System(e)	Verwendung in Kapitel	Beschreibung der Verwen- dungsweise (bspw. Zweck, Prompt, Weit- erverarbeitung KI-generierter Output)
Sprachliche Op- timierung oder Strukturierung von Text			
Generierung von Formeln, Grafiken, Tabellen und Diagrammen			
Ideenentwicklung und Konzeption von Aspekten der Arbeit			
Einordnung von Literatur			
Datensammlung und -analyse			
...			

Esslingen, den November 12, 2025 \_\_\_\_\_  
Unterschrift

---

<sup>1</sup>Bitte entfernen Sie nichtzutreffende Zeilen der Tabelle. Ergänzen Sie Zeilen, wo nötig. Weitere relevante Arbeitsschritte könnten z.B. sein: Generierung von Programmcode, Interpretation von Daten, Auswahl von Modellen und Methoden, Übersetzung von Text...