

Technical Writing: Tips for CS Students

Dr. Kevin W. Hamlen

Outline

- covered in this talk
 - advice about paper structure
 - good technical writing style
 - quality typography
 - bibliography guidelines
- not covered
 - good research methodology
 - how to pick a fruitful research problem/direction
 - how to graduate (though good writing helps)

How Much Does Writing Quality Matter?

- Cannot be underestimated!
 - Many (maybe most) paper rejections arise from **reviewer misinterpretations** of your work.
 - can be avoided via very, very clear and concise writing
 - How often your paper is read and cited depends significantly upon how easy it is to read.
 - improve impact by making it an easy, enjoyable read
- Assume a rushed reader
 - Referees must often review 10 or more papers per week.
 - They look for quick rejections: If an early sentence gives a bad impression, they may decide to reject before reading the rest. Very hard to recover from this.
 - Authors scrambling to assemble a literature review often skim 50 papers in a day. **Clarity is everything** if you want to be cited.

Paper Structure

- A typical structure
 1. abstract
 2. introduction
 3. overview
 4. technical section(s)
 5. discussion
 6. related work
 7. conclusion
 8. bibliography
- not necessarily (but often) in that order

Writing Order for Students

- Students tend to be obsessed with low-level implementation details
 - most readers don't care (until convinced to care)
 - need to learn to see the research from a higher-level “rest of the world” perspective
- Suggested writing order for build-before-write projects:
 - Overview first
 - Technical Section(s) next
 - adjust Overview as you go to add concepts you forgot
 - make a to-do list for Discussion section as you go
 - Related Work next
 - Discussion (influenced by Related Work)
 - Go play badminton
 - Introduction (hardest section to write)
 - Conclusion (easiest section to write)
 - Abstract (compressed version of Introduction)

SECTION-SPECIFIC ADVICE

Abstract

- Purposes
 - help reader/reviewer decide whether to read/review
 - declare keywords for search engines
 - provide template for others to summarize your work
- Style guidelines
 - third person (no “I” or “we”)
 - should make sense when dislocated from paper
 - tightest, most succinct possible prose
 - no citations, avoid math
- Content
 - describe problem in 1-2 sentences
 - describe solution in 1-2 sentences
 - highlight novelties & big results in 1-2 sentences

Proactive Obfuscation

[Roeder & Schneider, TOCS'09]

Proactive obfuscation is a new method for creating server replicas that are likely to have fewer shared vulnerabilities. It uses semantics-preserving code transformations to generate diverse executables, periodically restarting servers with these fresh versions. The periodic restarts help bound the number of compromised replicas that a service ever concurrently runs, and therefore proactive obfuscation makes an adversary's job harder. Proactive obfuscation was used in implementing two prototypes: a distributed firewall based on state-machine replication and a distributed storage service based on quorum systems. Costs intrinsic to supporting proactive obfuscation in replicated systems were evaluated by measuring the performance of these prototypes. The results show that employing proactive obfuscation adds little to the cost of replica-management protocols.

Proactive Obfuscation

[Roeder & Schneider, TOCS'09]

Problem

Solution

Results

Proactive obfuscation is a new method for creating server replicas that are likely to have fewer shared vulnerabilities. It uses semantics-preserving code transformations to generate diverse executables, periodically restarting servers with these fresh versions. The periodic restarts help bound the number of compromised replicas that a service ever concurrently runs, and therefore proactive obfuscation makes an adversary's job harder. Proactive obfuscation was used in implementing two prototypes: a distributed firewall based on state-machine replication and a distributed storage service based on quorum systems. Costs intrinsic to supporting proactive obfuscation in replicated systems were evaluated by measuring the performance of these prototypes. The results show that employing proactive obfuscation adds little to the cost of replica-management protocols.

Introduction

- Most important section for acceptance!
- Content
 - describe the problem in detail (citing key works)
 - explain why past solutions are inadequate
 - vital for acceptance!
 - must cover every solution that might occur to reader
 - summarize solution
 - dazzle reader with a brilliant new insight (climax of paper)
 - list contributions in readily visible form (e.g., bullets)
 - roadmap for rest of paper
 - controversial—some think this is a waste of space

Proactive Obfuscation

[Roeder & Schneider, TOCS'09]

Independence of replica failure is crucial when using replication to implement reliable distributed services. But replicas that use the same code share many of the same vulnerabilities and, therefore, do not exhibit independence to attack. This article introduces a new method of restoring some measure of that independence, *proactive obfuscation*, whereby each replica is periodically restarted using a freshly generated, diverse executable. The diversity increases the work an adversary must undertake to compromise a service that employs replication.

Key Question: After reading only my first paragraph, do I want to read on?

Related Work: Where?

- Early or late? (highly controversial)
- Best practice: Put early only if absolutely necessary to introduce concepts critical for presenting novel contributions
- Early
 - delays reader from reaching real contributions
 - Result: reviewers complain
- Late
 - delays full argument of novelty
 - reviewers remain unconvinced throughout majority of paper
 - Result: reviewers reject

REDACTED

Related Work: Content

- Purpose
 - convince reviewers that you know all related work
 - convince reader that your work improves upon past
 - tell story in which your work emerges as the hero
- Content
 - organize by theme / approach
 - not a laundry list!
 - needs to tell a story
 - avoid pre- or re-presenting your own work
 - avoid re-presenting prior work
 - summarize connections & differences
 - avoid insulting prior work (!!!)
 - cite ACCURATELY (requires deep understanding of papers!)

Overview Section

- Purpose
 - provide whole-system context in which reader can place lower-level innovations introduced later
 - put innovative claims into practical context
- Content
 - describe system from user/admin perspective
 - summarize main components (picture is ideal)
 - clearly define TCB and system assumptions
 - conceptual description of each component's design
 - What's the “big idea”?

Technical Section(s)

- Purpose
 - provide detailed answers to all reader questions
 - present evidence/proof of claims
- Content
 - must be well-organized and systematic
 - Some referees won't fully read it! They hunt for answers to specific questions.
 - Never present old news as if it's new.
 - Re-invention / rediscovery is an admission of ignorance!
 - Reuse of prior discoveries immunizes you (somewhat) from criticism on those points.

Discussion Section

- mainly about (apparent) limitations
 - If you have trouble thinking of limitations, you don't understand your work well enough.
- think like an attacker
 - pull all the dirtiest tricks; don't play "fair"
- put a positive spin on each limitation
 - best spin: mitigated by related work
 - okay spin: mitigated by future work
 - okay spin: out of our scope (but declare your scope early)
 - dangerous: open, unsolved problem
- reviewers DO use these admissions against you
 - but omitting a limitation entirely is much worse

Conclusion Section

- Purpose
 - reinforce what readers should remember
 - make a “final impression”
 - excite reviewers about future progress
- Content
 - restate main innovations and results, but this time you can use terms/concepts introduced throughout the paper
 - suggest future work (try to be inclusive!)

Bibliography

- Purpose
 - give credit where credit is due
 - prove that you know how to give credit accurately
 - prove your awareness of past work
- Quality standards for first submission
 - needn't be perfect but must be accurate!
 - referees are angered by miscitation
 - double-check author orderings
 - ensure each citation is most recent version
 - do not trust CiteSeer! (consult but verify)
 - DBLP is more reliable, but don't blindly copy & paste its BibTeX entries

Bibliography: Camera-ready

- Quality standards for camera-ready copy
 - journal editors may do it for you
 - proof-read their edits VERY carefully (usually wrong)
 - or they may leave it to you
 - read EVERY entry in its entirety and double-check ALL info
 - always include page numbers if they exist
 - be consistent about venue acronyms, publishers, addresses, etc. (don't include it in one but omit it from others)
 - don't just proof the source; proof the final document
 - don't just skim; actually read every (boring) line

WRITING STYLE

Paragraphing

- no monolithic paragraphs
 - max paragraph height: ~2 inches
 - larger paragraphs = poorly organized ideas
- topic sentences
 - far more prominent than buried sentences
 - make sure anything you want readers to see is a topic sentence
 - try reading only topic sentences in order—should make sense and tell the story of your paper!
 - find “most important” sentence in each paragraph—that one should be the topic sentence
 - Common mistake: Most students put all the best sentences at the *ends* of their paragraphs rather than the beginnings.

Topic Sentence Example

BACKWARDS ABSTRACT

The growing proliferation of widget technologies has led to increased demand for high performance at low cost. Most past work has focused on heat dissipation in widgets for higher efficiency, but this fails when widgets are densely packed. This dissertation therefore proposes and investigates an alternative approach based on compile-time optimization of widget software for reduced power consumption.

CORRECTED ABSTRACT

Topic Sentence Example

BACKWARDS ABSTRACT

The growing proliferation of widget technologies has led to increased demand for high performance at low cost. Most past work has focused on heat dissipation in widgets for higher efficiency, but this fails when widgets are densely packed. This dissertation therefore proposes and investigates an alternative approach based on compile-time optimization of widget software for reduced power consumption.

CORRECTED ABSTRACT

This dissertation proposes and investigates compile-time optimizations of widget software for reduced power consumption. Compile-time optimization is shown to be a promising alternative to heat dissipation approaches to widget optimization, which fail for densely packed widgets. The compile-time alternative therefore addresses the increasing demand for higher widget performance at low cost, which has been driven by the rapidly growing proliferation of widget technologies.

“Tight” Prose

- each sentence conveys its meaning in the fewest words/clauses possible
- each paragraph makes a self-contained point in a clear, logical, concise manner
- Requires MANY EDITING PASSES
 - I rewrite each paragraph at least 3 times, removing/reordering words and sentences each time.
 - When in paper-shrinking mode, I additionally remove non-essential, redundant, or ancillary points.
- My mantra: “This paper will compete with other papers in which every inch of space is devoted to something interesting and novel.”

Strategies for Tightening

- reorder and restructure thoughts to prioritize main subjects/objects
 - “First it compiles the module, which can fail, but if it succeeds it results in object code, and then all the object codes are linked together to create the final executable.”
 - “Final executables are generated by compiling all modules and linking the resulting object codes. Compilation failure results in ... ”
- relocate dangling prepositions/participles
 - “The first solution we thought **of** didn’t work.”
 - “The first solution **of which** we thought didn’t work.”
 - “Our first solution failed.”
- shorten clumsy, overly verbose phrases
 - “the fact that”
 - “in order to”
 - “all of”
 - “so called”
 - “the reason is that”

Tense

- Use present tense for almost everything
 - past tense = less practical
 - “Our checker found all the vulnerabilities in X.”
 - “Our checker finds all the vulnerabilities in X.”
 - future tense = unproved/untested
 - “The analysis will check both the positive and negative branch.”
 - “The analysis checks both the positive and negative branch.”
- Cross-references should **ALWAYS** be in present tense
 - **INCORRECT**: In Section 4 we will discuss this further.
 - **CORRECT**: Section 4 discusses this further.
- Some past tense is unavoidable
 - “One experiment failed, but we were unable to duplicate it.”
- Future tense should usually be reserved for future work
 - “Once version 3 is publicly available, we will support it.”

Voice

- Avoid passive voice
 - BAD: Results were tabulated in parallel.
 - BETTER: Our system tabulated results in parallel.
 - BEST: Our system tabulates results in parallel.
- Objections to passive voice
 - imprecise (Who did the tabulating?)
 - weak
 - unnecessarily verbose
- Possible exception: abstracts (controversial)
 - OPTION 1: “This paper presents X, Y, and Z.”
 - OPTION 2: “X, Y, and Z are presented.”

Proper Word Choices

- “proven” (adjective) vs. “proved” (past tense verb)
 - INCORRECT: These results have been proven.
 - CORRECT: These are proven results.
- “which” (descriptor) vs. “that” (qualifier)
 - I appreciate info flow papers, which are written well.
 - I appreciate info flow papers that are written well.
- “whether” (alternatives) vs. “if” (implication)
 - INCORRECT: Module X decides if P is true or false.
 - CORRECT: Module X decides whether P is true or false.
- “like” (similarity) vs. “such as” (example)
 - INCORRECT: We use many tools, like SAT-solvers and disassemblers.
 - CORRECT: We use many tools, such as SAT-solvers and disassemblers.
- “dissertation” (a document) vs. “thesis” (an assertion)
 - My thesis is that widgets are useful. My dissertation proves the veracity of my thesis.
- “may” (permission) vs. “might” (possibility) vs. “can” (capability)
 - The user may delete all the files. [User is allowed to delete them.]
 - The user might delete all the files. [Possible that user deletes them.]
 - The user can delete all the files. [User is capable of deleting them.]

“Besides”

- Never begin a sentence with “Besides, ...”
 - It means, “Even if you ignore everything I already said, ...”
 - You are admitting that everything prior is inconsequential!
 - Example: “Smoking causes lung cancer. Besides, I dislike the taste.”
 - Translation: The real reason I don’t smoke is because I dislike the taste. That argument alone explains my behavior, so the other argument is unnecessary and can be safely disregarded by the reader.
- To build upon what you’ve previously said, use...
 - “Additionally, ...”
 - “Furthermore, ...”
 - “Moreover, ...”
- To provide examples of previous points, use...
 - “For example, ...”
 - “For instance, ...”
- To summarize previous points, use...
 - “In summary, ...”
 - “In conclusion, ...”

Terminology

- Never invent your own unless it's a genuinely new concept.
 - If you think it's a new concept, you better be right.
 - Reusing terminology accurately is good:
 - convinces reader of your competence
 - connects concepts to those already known by readers
 - shifts criticism away from you (and toward inventors of those terms)
- Important terms should be italicized at first use
 - never use “scare quotes”
 - Possible exception: referring to the word rather than its meaning
 - Example: The word “foobar” was first coined by Allied soldiers in World War II.
 - don't italicize twice
 - use `\emph` (not `\it` or `\textit`)

Citations

- Avoid using citations as nouns
 - BAD: The results were published in [1].
 - GOOD: The published results [1] are quite promising.
- Never begin a sentence with a citation
 - INCORRECT: [2] is an excellent paper.
 - CORRECT: The most recent study [2] is excellent.
- Avoid putting author names in text
 - BAD: Hamlen [3] rejects this approach.
 - GOOD: Related work rejects this approach [3].
 - System names are okay
 - GOOD: MOBILE [4] adopts a different approach.
 - Super-famous researchers (e.g., Turing winners) are special
 - GOOD: Hoare's axiomization [5] is the standard approach.
- Never cite Wikipedia as an authoritative source
 - BAD: "Fixed points [6] are widely used in denotational semantics."
 - GOOD: "This highly questionable programming practice is nevertheless widely advocated in the JavaScript community (e.g., [7])."

Figure/Table Captions

- Check style guide for journal/conference
- If not specified...
 - begin with a *tag*: a descriptive phrase, not a complete sentence, but capitalized and punctuated like a sentence
 - optionally follow with complete sentences
- Example:

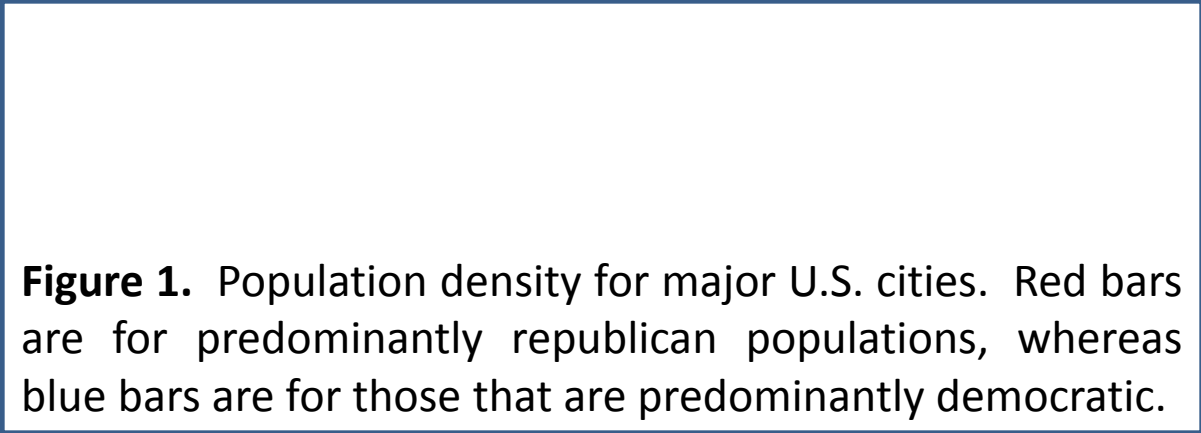


Figure 1. Population density for major U.S. cities. Red bars are for predominantly republican populations, whereas blue bars are for those that are predominantly democratic.

Punctuation: Commas

- series
 - INCORRECT: Writing can be good, bad or ugly.
 - CORRECT: Writing can be good, bad, or ugly.
- intro words
 - INCORRECT: In conclusion we solved the problem.
 - CORRECT: In conclusion, we solved the problem.
- sub-clauses
 - INCORRECT: We applied many tools such as X and Y to the problem.
 - CORRECT: We applied many tools, such as X and Y, to the problem.
- separate thoughts, introduce pauses, etc.
 - use to lucidly structure ideas within each sentence
 - too many commas = overly convoluted sentences
 - too few commas = reader must work too hard to disentangle ideas

Other Punctuation

- dashes
 - em-dash (“---”): for clause separation
 - “Every writer has bad days—often followed by sleepless nights.”
 - “Certifying compilers—particularly those developed within the past few years—provide much higher assurance.”
 - en-dash (“--”): for ranges: “See chapters 2-7.”
 - dashes: for hyphenated words: “The code-producer is trusted.”
- semicolon
 - separate two complete, short sentences to be kept together
 - “Our efforts failed; however, we subsequently found an alternative approach.”
 - lists in which some items contain commas
- plural acronyms: VMs (no apostrophe), OSes (no apostrophe)
 - Possible exception: math variables (all x 's in the equation)
 - Definite exception: possessive acronyms (“the API's header file”)

Useful Latin Abbreviations

- Examples vs. clarification vs. exhaustive itemization
 - “Access modifiers (**e.g.**, *private*) are helpful.”
 - “Access modifiers (**i.e.**, field qualifiers that specify security properties) are helpful.”
 - “Access modifiers (**viz.**, *public*, *private*, and *protected*) are helpful.”
- Different types of external citation
 - “AOP [1] is useful.” (The cited work defines AOP.)
 - “AOP (**e.g.**, [1,2]) is useful.” (The works are examples of AOP, but not necessarily definitional or exhaustive.)
 - “AOP is useful (**cf.**, [1]).” (The work has something to say about this assertion, but might not agree with it. The reader is advised to compare this statement with those cited.)
 - “AOP is a huge field (**cf.**, [1]).” (Cross-reference the cited work to find all the related works.)

Typography

- LaTeX is non-negotiable (no Word docs, please)
 - use `\label`, `\ref`, `\cite` for all referencing
 - put each sentence on a separate line of your source file
 - useful for version diffs when collaborating with other authors
 - break up sentences further as convenient
- common LaTeX mistakes
 - missing ties: “Dr.~Hamlen wrote Figure~\ref{fig:system}.”
 - variables not in math mode: $\$n\$$ th-order
 - multi-letter names in math: $\$\mathit{foo}_1\$$
 - inconsistent multiplication notation
 - $\$k*m\$$ vs. $\$km\$$ vs. $\$k\cdot m\$$
 - space after macro: `\mytool\` is great

Tables

- Environment: tabular
- Packages
 - booktabs: for proper line widths and spacing
 - dcolumn: for aligning columns on a decimal with the column header still centered
 - multirow: for row-spanning cells
- Style guidelines
 - no vertical lines, no double lines
 - use blank cells instead of repeating entries
 - no “ditto marks”
 - units in the headings, not the cells
 - put digits before decimals (0.1 instead of .1)

Graphics

- always use vector graphics, never bitmap
 - pdf and eps are okay
 - jpg, png, tiff, bmp, gif, etc., are NOT okay
 - ideal: use MetaPost (and my MetaFlow package)
 - MetaPost is to graphics as LaTeX is to text
 - comes with all LaTeX distributions (mpost)
 - documentation readily available online
- size graphics relative to enclosing environment
 - `\includegraphics[width=.9\hsize]{...}`
 - preserve the aspect ratio (no “height=”)

Fine-tuning

- Shortening by rewording
 - Look for paragraphs ending with a short line
 - Delete some words to shift that line up, saving a line
- Eliminating unnecessary vertical space
 - various techniques (e.g., `\vspace{-10pt}`)
 - try to get sections to start atop columns (eliminates separation space)
- Overfull hboxes
 - **microtype** is a wonderful package (use kerning option but not spacing option)
 - Introduce hyphenation points via `\-`
 - Tweak inter-word spacing (e.g., `\kern-1pt`)
 - Permit LaTeX to add space (`\emergencystretch`)
 - Introduce line break points (`\penalty0`, `\break`)
- Underfull vbox while output is active
 - Add stretchability to inter-line spacing:
 - `\advance\baselineskip 0pt minus 0pt plus 0.5pt`
 - Be careful with this; most publishers have strict rules about line spacing. Their rules trump underfull vboxes.

BibTeX

- Use the correct entry type
 - @PROCEEDINGS is not for conference publications! Use @INPROCEEDINGS.
 - @INBOOK = an *untitled* part of a book
 - @INCOLLECTION = a *titled* part of a book
 - Use @MISC only as a last resort. Nothing else may apply!
- Enclose capitalization-sensitive title words and macros in braces
 - title = “Navigating the {Java} standard {API}”
 - address = “Jyv{\”a}skyl{\”a}, Finland”
- Use descriptive identifiers
 - GOOD: @ARTICLE{hamlen12tacas, ... }
 - BAD: @ARTICLE{12, ...}
- No commas in author lines, and enclose companies in braces
 - GOOD: author = “Alan M. Turing and Alonzo Church and {Microsoft Corporation}”