How to Use the IEEEtran LaTeX Class

Michael Shell, Member, IEEE

Abstract—This article describes how to use the IEEEtran class with LaTeX to produce high quality typeset papers that are suitable for submission to the IEEE Computer Society. IEEEtran can produce journal and technical note (correspondence, short) papers with a suitable choice of class options. Correspondence papers typically use a subset of the commands discussed here. Please note that the use of IEEE Computer Society templates is meant to assist authors in correctly formatting manuscripts for final submission and is not a guarantee on how the final paper will be formatted by IEEE Computer Society staff. This template may be used for initial submissions; however, please consult the author submission guidelines for formatting instructions as most journals prefer single column format for peer review. An abstract should be 100 to 200 words for regular papers, no more than 50 words for correspondence (short) papers and comments, and should clearly state the nature and significance of the paper. Abstracts must not include mathematical expressions or bibliographic references.

Index Terms— Keywords should be taken from the taxonomy (http://www.computer.org/mc/keywords/keywords.htm). Keywords should closely reflect the topic and should optimally characterize the paper. Class, IEEEtran, LaTeX, paper, style, template, typesetting.

1 INTRODUCTION

WITH a recent IEEEtran class file, a computer running LaTeX, and a basic understanding of the LaTeX language, an author can produce professional quality typeset research papers very quickly, inexpensively, and with minimal effort. The purpose of this article is to serve as a user guide of IEEEtran LaTeX class and to document its unique features and behavior.

This document applies to version 1.6b and later of IEEEtran. Prior versions do not have all of the features described here. IEEEtran will display the version number on the user's console when a document using it is being compiled. The latest version of IEEEtran and its support files can be obtained from the IEEE Computer Society's web site [2], or CTAN [1]. This latter site may have some additional material, such as beta test versions and files related to non-IEEE uses of IEEEtran.

Complimentary to this document is the file bare_jrnl.tex which is a "bare bones" example (template) file of a journal paper. Authors can quickly obtain a functional document by using these files as starters for their own work. It is assumed that the reader has at least a basic working knowledge of LaTeX. Those so lacking are strongly encouraged to read some of the excellent literature on the subject [3]. General support for LaTeX related questions can be obtained in the internet newsgroup comp.text.tex. There is also a searchable list of frequently asked questions for this newsgroup [4].

Please note that the appendices sections contain information on installing the IEEEtran class file as well as tips on how to avoid commonly made mistakes. Appendix F, which describes information for advanced LaTeX users, has been made supplemenatal and is also included with this document.

• *M. Shell is with the Georgia Institute of Technology, Atlanta, GA 30332.*

2 CLASS OPTIONS

There are a number of class options that can be used to control the overall mode and behavior of IEEEtran. These are specified in the traditional LaTeX way. For example, \documentclass[10pt,technote]{IEEEtran} is used with correspondence (technote) papers. The various categories of options will now be discussed. For each category, the default option is shown in bold. The user must specify an option from each category in which the default is not the one desired. The various categories are totally orthogonal to each other—changes in one will not affect the defaults in the others.

2.1 Type Size: 12 pt

There are many possible values for the normal text size when setting up a template. However, the IEEE Computer Society requires that all single column submissions are set to 12pt and double column submissions are set to 9.5pt. This includes short papers and correspondence items, hereafter referred to as technote papers. To help guarantee an article will meet submission length limit requirements, please use the correct point size when creating an article.

2.2 draft, draftcls, final

IEEEtran provides for three draft modes as well as the normal final mode. The draft modes provide a larger line spacing to allow for editing comments. The standard draft option puts every package used in the document into draft mode. With most graphics packages, this has the effect of disabling the rendering of figures. If this is not desired, one can use the draftcls option instead to yield a draft mode that will be confined within the IEEEtran class so that figures will be included as normal. draftclsnofoot is like draftcls, but does not display the word "DRAFT" along with the date at the foot of each page. When using one of the draft modes, most users will also want to select the onecolumn option. Please note, however, that final submissions of accepted papers should be set as two-column to best regulate final article length.

Digital Object Identifier 10.1109/TDSC.2017.2684801

2.3 journal, technote

IEEEtran offers five major modes to encompass journal, correspondence (technote), and peer review papers. Journal and technote modes will produce papers very similar to those that appear in many IEEE Computer Society Transactions journals. When using technote, users should also select the 12pt option. The peer review mode is much like the journal mode, but produces a single-column cover page (with the title, author names, and abstract) to facilitate anonymous peer review. The title is repeated (without the author names or abstract) on the first page after the cover page. Papers using the peer review options require an \IEEEpeerreviewmaketitle command (in addition to and after the traditional \maketitle) to be executed at the place the cover page is to end—usually just after the abstract. This command will be silently ignored with the nonpeer review modes. See the bare template files for an example of the placement of this command.

2.4 letterpaper, a4paper, finalsubmission

IEEEtran supports both US letter (8.5in X 11in) and A4 (210mm X 297mm) paper sizes for formatting your paper for peer review. However since the IEEE CS uses a page size of 7.875in X 10.75in when formatting final publication materials authors should select the "finalsubmission"option before submitting their work to IEEE. The main purpose of the a4paper option is to allow authors outside the US to print their work on A4 paper. Changing the paper size will not alter the typesetting of the document—only the margins will be affected. In particular, documents using the a4paper option will have reduced side margins (A4 is narrower than US letter) and a longer bottom margin (A4 is longer than US letter). For both cases, the top margins will be the same and the text will be horizontally centered. Note that authors should ensure that all postprocessing (.ps, .pdf, etc.) uses the same paper specification as the .tex document. Problems here are by far the number one reason for incorrect margins. See Appendix B for more details. 2.5 onecolumn, twocolumn These options allow the user to select between one and two column text formatting. Since IEEE always uses two column text, the one column option is of interest only with draft papers.

2.6 nofonttune

IEEEtran normally alters the default interword spacing to be like that used in IEEE publications. The result is text that requires less hyphenation and generally looks more pleasant, especially for two column text. The nofonttune option will disable the adjustment of these font parameters. This option should be of interest only to those who are using fonts specifically designed or modified for use with IEEE work.

3 THE TITLE PAGE

The parts of the document unique to the title area are created using the standard LaTeX command \maketitle. Before this command is called, the author must declared all of the text objects which are to appear in the title area.

3.1 Paper Title

The paper title is declared like:

```
\title{A Heuristic Coconut-Based Algorithm}
```

in the standard LaTeX manner. Line breaks (\\) may be used to equalize the length of the title lines.

3.2 Author Names

The name and associated information is declared with the \author command. \author behaves slightly differently depending on the document mode.

3.2.1 Names in Journal/Technote Mode

A typical \author command for a journal or technote paper looks something like this:

```
\author{Michael~Shell,~\IEEEmembership{Member,~I
EEE,} John~Doe,~\IEEEmembership{Fellow,~OSA,}
and~Jane~Doe,~\IEEEmembership{Life~Fellow,~IEEE}
%
\thanks{Manuscript received January 20, 2002;
revised August 13, 2002. }%
\thanks{M. Shell is with the Georgia Institute
of Technology.}}
```

The \IEEEmembership command is used to produce the italic font that indicates the authors' IEEE membership status. The \thanks command produces the "first footnotes." Because the LaTeX \thanks was not designed to contain multiple paragraphs, one will have to use a separate \thanks for each paragraph. However, if needed, regular line breaks (\\) can be used within \thanks. In order to get proper line breaks and spacing, it is important to correctly use and control the spaces within \author. Use nonbreaking spaces (~) to ensure that name/membership pairs remain together. A minor, but easy, mistake to make is to forget to prevent unwanted spaces from getting between commands which use delimited ({}) arguments. Note the two %, which serve to prevent the code line break on lines ending in a } from becoming an unwanted space. Such a space would not be ignored as an end-of-line space because, technically, the last\thanks is the final command on the line. "Phantom" spaces like these would append to the end of the last author's name, causing the otherwise centered name line to shift very slightly to the left.

3.3 Running Headings

The running headings are declared with the \markboth{}{} command. The first argument contains the journal name information and the second contains the author name and paper title. For example:

```
\markboth{Journal of Quantum Telecommunications,~
Vol.~1, No.~1,~January~2025}{Shell
\MakeLowercase{\textit{et al.}}: A Novel Tin
Can Link}
```

Note that, because the text in the running headings is automatically capitalized, the `\MakeLowercase{}` command must be used to obtain lower case text. However, IEEE Computer Society style leaves all items in the running headings capitalized, so use of this command is not necessary. Technote papers do not utilize the second argument. **Authors should not put any name information in the headings (if used) of anonymous peer reviewed papers (double blind review).**

3.4 Publication ID Marks

Publication ID marks can be placed on the title page of journal and technote papers via the `\pubid{}` command. The title page of this document (PDF form) has a publication ID that was made with:

```
\pubid{0000--0000/00\$00.00~\copyright~200X
IEEE}
```

Although authors do not yet have a valid publication ID at the time of paper submission, `\pubid{}` is useful because it provides a means to see how much of the title page text area will be unavailable in the final publication.

4 Abstract and Index Terms

The abstract is generally the first part of a paper after `\maketitle`. The abstract text is placed within the abstract environment:

```
\begin{abstract}
We propose ...
\end{abstract}
```

Journal and technote papers also have a list of key words (index terms) which can be declared with:

```
\begin{keywords}
Broad band networks, quality of service, WDM.
\end{keywords}
```

5 Sections

Sections and their headings are declared in the usual LaTeX fashion via `\section{}`, `\subsection{}`, `\subsubsection{}`, and `\paragraph{}`. The numbering for these sections is in upper case Arabic numerals, then upper case Arabic numerals, separated by periods. The `\paragraph{}` section is not allowed for technotes as they generally are not permitted to have such a deep section nesting depth. If needed, `\paragraph{}` can be restored by issuing the command `\setcounter{secnumdepth}{4}` in the document preamble.

5.1 Initial Drop Cap Letter

The first letter of a journal paper is a large, capital, oversized letter which descends one line below the baseline. Such a letter is called a "drop cap" letter. The other letters in the first word are rendered in upper case. This effect can be accurately produced using the IEEEtran command `\PARstart{}{}`. The first argument is the first letter of the first word, the second argument contains the remaining letters of the first word. The drop cap of this document was produced with:

```
\PARstart{W}{ith}
```

6 Citations

Citations are made with the `\cite{}` command as usual. IEEEtran will produce citation numbers that are individually bracketed in IEEE Computer Society style. ("[1], [5]" as opposed to the more common "[1, 5]" form.) Citation ranges should be formatted as follows: [1], [2], [3], [4] (as opposed to [1]-[4], which is not IEEE Computer Society style). Please note that references will be formatted by IEEE Computer Society production staff in the same order provided by the author. The base IEEEtran does not sort or produce "ranges" when there are three or more consecutive citation numbers. However, IEEEtran predefines some format control macros to facilitate easy use with the LaTeX cite.sty package [6]. So, all an author has to do is to call cite.sty:

```
\usepackage{cite}
```

and the citation numbers will automatically be sorted and arranged IEEE Computer Society style.

Note that, if needed, the cite.sty's `\cite{}` command will automatically add a leading space, i.e., "(`\cite{mshell01}`)" will become "([1])." If this behavior is not desired, use the cite package's noadjust option (cite.sty V3.8 and later) which will turn off the added spaces:

```
\usepackage[noadjust]{cite}
```

`\cite{}` also allows for an optional note, e.g., `\cite[Th. 7.1]{mshell01}`. If the `\cite{}` with note has more than one reference, the note will be applied to the last of the listed references. It is generally desirable that, if a note is given, only one reference should be listed in that `\cite{}`.

7 Equations

Equations are created using the traditional `equation` environment:

```
\begin{equation}
\label{eqn_example}
x = \sum\limits_{i=0}^{z} 2^{i}Q
\end{equation}
```

which yields:

$$x = \sum_{i=0}^{z} 2^{i} Q \tag{1}$$

Use the displaymath environment instead if no equation number is desired. When referring to equations, articles

TABLE 1
Math Spacings Used by LaTeX

Size	Width	Cmd.	Used for	Example
small	1/6 em	\,	symbols	$a\,b$
medium	2/9 em	\:	binary operators	$a + b$
large	5/18 em	\;	relational operators	$a = b$
negative small	−1/6 em	\!	misc. uses	$a\!b$

in IEEE Computer Society publications do not use the word "equation," but rather just enclose the equation number in parentheses, e.g.,

```
... as can be seen in (\ref{eqn_example}).
```

The IEEE Computer Society's two column format puts serious constraints on how wide an equation can be. So, a fair portion of the effort in formatting equations usually has to be devoted to properly breaking them. It is the author's responsibility to ensure that all equations fit into the given column width.

8 Multiline Equations

Perhaps the most convenient and popular way to produce multiline equations is LaTeX2e's `eqnarray` environment. However, `eqnarray` has several serious shortcomings:

1. The use of 2x`\arraycolsep` for a column separation space does not provide natural math spacing in the default configuration.
2. Column definitions cannot be altered.
3. Is limited to three alignment columns.
4. Column alignment cannot be overridden within individual cells.

There are a number of vastly superior packages for formatting multiline mathematics. Perhaps the most popular is the amsmath package [7]. Amsmath is a comprehensive work which contains many helpful tools besides enhanced multiline alignment environments. So, all authors should give serious consideration to its use—regardless of what they use to generate aligned equations. One thing to be aware of is that, upon loading, amsmath will configure LaTeX to disallow page breaks within multiline equations (even within nonamsmath defined environments). The philosophy here is that author should manually insert breaks where desired so as to ensure that breaks occur only at acceptable points. To restore IEEEtran's ability to automatically break within multiline equations, load amsmath like:

```
\usepackage{amsmath}
\interdisplaylinepenalty=2500
```

Another extremely powerful set of alignment tools, one of which is a totally rewritten eqnarray environment, is provided by mathenv.sty which is part of Mark Wooding's MDW Tools [8].

Finally, IEEEtran provides a fully integrated custom IEEEeqnarray family of commands (see Appendix F) that are designed to have almost universal applicability for many different types of alignment situations.

Nevertheless, it is instructive to show a simple example using the standard eqnarray in order to explain some of the fine points of math spacing under LaTeX. As shown in Table 1, TeX normally draws from four different spacings when typesetting mathematics. In order to produce precision (and correct) mathematical alignments, it is crucial to understand how to control such spacing. Consider a multiline equation

$$\begin{aligned} Z = {} & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \\ & +a+b \qquad (1) \\ & +a+b \qquad (2) \\ & \ +a+b \qquad (3) \\ & +a+b \qquad (4) \end{aligned}$$

(in typical IEEE Computer Society style) which was produced by:

```
\setlength{\arraycolsep}{0.0em}
\begin{eqnarray}
Z&{}={}&x_1 + x_2 + x_3 + x_4 + x_5 +
x_6\nonumber\\
&&+a + b\\% line 1
&&+{}a + b\\% line 2
&&{}+a + b\\% line 3
&&{+}\:a + b% line 4
\end{eqnarray}
\setlength{\arraycolsep}{5pt}
```

Lines one through four show some possible ways the $+ a + b$ line could be implemented.[1] Only number four is the correct way for most IEEE purposes. In TeX's math mode, spacing around operators can be inhibited by enclosing them within braces (e.g., `{=}`) or forced by surrounding them with "empty ords" (e.g., `{}={}`). It is important to understand that the empty ords do not have width themselves. However, their presence causes TeX to place space around the operators as if they were "next to something." With this in mind, the first step in the example is to set `\arraycolsep` to zero to prevent eqnarray from putting in the unwanted, artificial, intercolumn spacing. Placing empty ords around the equal sign then forces the correct natural spacing. Alternatively, `\arraycolsep` could have been set to 0.14em and the empty ords around the equal sign eliminated.[2] It is important to remember to restore `\arraycolsep` to its default value of 5pt after the `eqnarray` is complete as other environments (such as `array`) depend on it.

The first line is incorrect because "a" is being indicated as a positive quantity rather than something that must be

[1] In this absurd example, the equation numbering system is used to identify lines.
[2] This assumes that 1em in the text font has the same width as 1em in the math font. For the standard fonts, this is indeed the case.

added to the previous line. (i.e., the "+" is being treated as a unary, rather than a binary, operator.) In line two, adding an empty ord to the right side of the plus sign does nothing, except to demonstrate that empty ords have zero width. Adding an empty ord to the left side of the plus sign (line three) does engage binary spacing, but causes an unwanted[3] right shift of the line. Finally, manually adding a medium space to the right side only of the plus sign in line four does the trick. The suppression of automatic spacing around the plus sign ({+}) is unneeded in this case, but may be required in other alignment environments that "expand" such operators by default.

Another way around the spacing problem is to use only two alignment columns (as is done by amsmath.sty's \align). e.g., in the previous example, "Z =" would be contained in the first column.

8.1 Cases Structures

Incidentally, the `numcases` (or `subnumcases`) environments in Donald Arseneau's cases.sty package [9] should be used when "cases" structures in which each branch can be referenced with a different equation (or subequation) number are needed:

$$|x| = \begin{cases} x, & \text{for } x \geq 0 \quad (2a) \\ -x, & \text{for } x < 0 \quad (2b) \end{cases}$$

```
\begin{subnumcases}{|x|=}
x, & for $x \geq 0$\\
-x, & for $x < 0$
\end{subnumcases}
```

because those built from the `array` or amsmath `cases` environments will have a single equation number that encompasses both branches.

9. Floating Structures

9.1 Figures

Figures are handled in the standard LaTeX manner. For example:

```
\begin{figure}
\centering
\includegraphics[width=2.5in]{myfigure}
\caption{Simulation Results}
\label{fig_sim}
\end{figure}
```

Note that 1) figures should be centered via the LaTeX `\centering` command. This is a better approach than using the `center` environment which adds unwanted vertical spacing; 2) the caption follows the graphic; and 3) any labels must be declared *after* (or within) the caption command.

When referencing figure numbers in the main text (via `\ref{}`), the the IEEE Computer Society uses the abbreviation "Fig." rather than "Figure."

[3] The IEEE Computer Society normally wants all of the lines left aligned, but there are cases when such an indention may be desirable.

The `\includegraphics` command is the modern, preferred, way of including images and provides a flexible interface that makes it easy to scale graphics to size. To use it, the graphics or graphicx (recommended) must first be loaded.

```
\usepackage{graphicx}
```

It is strongly recommended that authors be familiar with the graphics package documentation [10] as well as Keith Reckdahl's excellent *Using Imported Graphics in LaTeX2e* [11]. The reader is reminded that the "draftcls" or "draftclsnofoot," not "draft," class option must be selected in order to get draft papers with visible figures.

As explained in Appendix D, Encapsulated PostScript (EPS) is the preferred graphics format for LaTeX work. Furthermore, the user's drawing/graphing application should be capable of outputting directly in EPS vector form (which will not degrade or pixelize when magnified)—although photos will likely have to be in (EPS) bitmap form.

The psfrag package [12] might also be of interest. Psfrag allows the user to "go into" an EPS graphic and replace text strings contained in it with real LaTeX code! In this manner, LaTeX's extensive support of mathematical symbols and fonts can be extended to figures made with applications with more modest glyph support. Using psfrag does require the use of the dvips DVI to PostScript conversion step (not pdfLaTeX[4]) as some of the features of the PostScript language have to be utilized. There is additional usage information on psfrag in the *Using Imported Graphics in LaTeX2e* guide [11].

9.2 Tables

Tables are handled in a similar fashion, but with a few notable differences. For example, the code

```
\begin{table}
\renewcommand{\arraystretch}{1.3}
\caption{A Simple Example Table}
\label{table_example}
\centering
\begin{tabular}{c||c}
\hline
\bfseries First & \bfseries Next\\
\hline\hline
1.0 & 2.0\\
\hline
\end{tabular}
\end{table}
```

results in Table 2. Note that the IEEE Computer Society places table captions *before* the tables. Within the table environment, the default text size is footnotesize which is what theIEEE Computer Society typically uses for tables. When using the tabular environment to construct tables, it is usually a good idea to increase the value of `\arraystretch` above unity to "open up" the table rows a tad.

[4] PdfLaTEX users currently have to "preprocess" their figures by importing them into a dummy document using psfrag, running LaTEX followed by dvips, then converting the PostScript output to a PDF graphic for imporation into the main document which is then processed by pdfLaTEX.

TABLE 2
A Simple Example Table

First	Next
1.0	2.0

Unfortunately, the standard LaTeX2e tabular environment has a number of shortcomings. Two notable problems are 1) the corners where lines meet are improperly formed and 2) it is not very flexible in terms of user control. For these reasons, authors are urged to look into some of the other packages for making tables. A good one that provides revised "drop-in replacements" for both the tabular and array environments is Frank Mittelbach's and David Carlisle's array package [13]. Even more powerful (and complex) is the tabular and array environments provided by the mdwtab.sty package which is part of Mark Wooding's MDW Tools [8].

As an alternative, IEEEtran offers the IEEEeqnarraybox command which can also be used to produce tables[5] of high quality. See Appendix F for more details.

9.3.1 Footnotes within Tables

Footnotes normally cannot be placed directly within some commands and environments such as `\parbox` and `tabular`, etc., because they become "trapped" inside. One way around this is to split the place the footnote marker (`\footnotemark`) is located (within the table) from where the footnote text itself is declared (outside of the table using `\footnotetext`).

Another approach is to use the footnote.sty package (which is part of Mark Wooding's MDW Tools [8]) which allows environments to be configured so as not to trap footnotes:

```
\usepackage{footnote}
\makesavenoteenv{tabular}
```

Note that is probably not a good idea to use footnotes in floating structures (like `table`) because the position of each can move relative to one another. To put the footnote at the end of a table instead of at the bottom of the page, just enclose `tabular`, etc., inside a minipage (no footnote package needed). A very good approach for handling footnotes within tables (including those that float) is to use Donald Arseneau's threeparttable package [14] which was used to generate Table 3 (the code of which is an example in the threeparttable.sty file).

9.3 Double Column Floats

LaTeX's `figure*` and `table*` environments produce figures and tables that span both columns. This capability can be used with the Steven Douglas Cochran's subfigure package [15] to format a set of related figures:

```
\begin{figure*}
\centerline{\subfigure[Case
I]{\includegraphics[width=2.5in]{subfigcase1}
```

[5] Table 1 was made using this command.

TABLE 3
The Skewing Angles (β) for Mu(H) + X^2 and Mu(H) + HX [a]

	$\mathrm{H(Mu)+F_2}$	$\mathrm{H(Mu)+Cl_2}$
β(H)	80.9° [b]	83.2°
β(Mu)	86.7°	87.7°

[a] for the abstraction reaction, $\mathrm{Mu + HX \rightarrow MuH + X}$.
[b] 1 degree $= \pi/180$ radians.

```
\label{fig_first_case}}
\hfil
\subfigure[Case
II]{\includegraphics[width=2.5in]{subfigcase2}
\label{fig_second_case}}}
\caption{Simulation results}
\label{fig_sim}
\end{figure*}
```

Note how labels can be tagged to each of the subfigures as well as to the overall figure. `\hfil` is used as a subfigure separator to achieve equal spacing around the graphics. More complex implementations are possible. See the subfigure documentation as well as the *Using Imported Graphics in LaTeX2e* guide [11] for more details.

It is a limitation of the LaTeX2e kernel that double column floats cannot be placed at the bottom of pages. That is to say "`\begin{figure*}[!b]`" will not normally work as intended. Authors that need this capability should obtain and load Sigitas Tolusis' stfloats package [16] which patches the LaTeX2e output routine to allow it to handle double column floats at the bottom of pages. Please note that stfloats is a very invasive package which may not work with versions of LaTeX other than the standard LaTeX2e release and may cause problems with other packages that modify the output routines (such as those that balance columns, alter the placement of floating figures, etc.). IEEE Computer Society authors are warned *not* to use packages that allow material to be placed across the middle of the two text columns (such as cuted.sty, midfloat.sty, etc.) as the IEEE Computer Society does not do this.

Another LaTeX2e limitation (patched with stfloats or not) is that double column floats will not appear on the same page where they are defined. So, the user will have to define such things prior to the page on which they are to (possibly) appear.

LaTeX2e (patched with stfloats or not) does not attempt to keep double and single column floats in sequence with each other. This can be fixed by loading David Carlisle's fix2col package (already installed on most LaTeX systems) [17]. However, fix2col should not be used with the stfloats package as they both modify some of the same output routines in different ways.

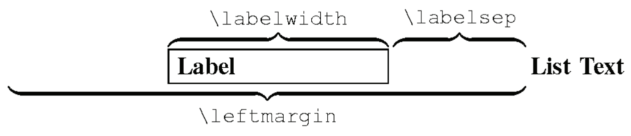
Finally, authors should also be aware that the LaTeX2e kernel (patched with stfloats or not) has a long standing limitation in that it will not allow rubber space that spans both columns to stretch or shrink as needed for each of the two main text columns. Therefore, it is possible for double column floats to cause underfull vbox errors because the remaining text height may not be equal to an integer number of normal size lines. The problem can occur in main text columns (on pages with double column floats) that do not have vertical rubber spacing (such as that around section headings, equations, etc.) and results in underfull vbox warnings coupled with paragraphs that are "pulled apart" from each other. To correct this, users can manually tweak the amount of space between the double column structure and main text by inserting a command like

```
\vspace*{-3pt}
```

(adjusted as needed) within the double column structure. Incidentally, IEEEtran automatically compensates for this problem when forming the paper title.

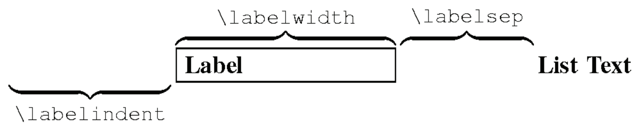
10 LISTS

The traditional LaTeX itemize, enumerate, and description (IED) list environments are ill-suited for producing the style of lists used in IEEE Computer Society publications. The main problem is that they do not provide the user a means for controlling the parameters of the resultant list. Furthermore, making global changes to the parameters of the underlying `\list` will result (often unexpectedly to a user) in the improper behavior of other commands that depend on it, such as `\quote`. Finally, LaTeX's `\list` considers the left margin of the list text to be the reference point that determines how the list is positioned relative to the left margin of the main text:



This contrasts with IEEE Computer Society lists which use the label box as the reference point for the list structure, i.e., for a given circumstance, the list labels will be indented by a certain amount, the list text block will be indented from the label boxes by a given amount, and these spacings will determine the position of the list text.

For these reasons, IEEEtran provides enhanced IED list environments that make it much easier to produce IEEE style lists. The underlying `\list` remains the same as in traditional LaTeX so as not to break code that depends upon it. IEEEtran uses a new length variable, `\labelindent`, so that users can specify IED list structures directly in IEEE fashion:



The IEEEtran IED lists ignore all "external" changes to the list length parameters. Instead, IED lists are controlled exclusively via two interfaces:

1. "global" control via the `\iedlistdecl` command and
2. "local" control via an *optional* argument that can be provided to `\itemize`, `\enumerate`, and `\description`.

For example, declaring

```
\renewcommand{\iedlistdecl}{\settowidth{\label
width}{Hello}}
```

in an IEEEtran document will set the default width of the label boxes in *all* later IED lists to be equal to the width of "Hello." Note: Because setting a `\labelwidth` is so commonly performed, IEEEtran provides a command: `\setlabelwidth{X}` which is a shorter form of: `\settowidth{\labelwidth}{X}`. The local control is used if the parameters are to apply only to an individual IED list:

```
\begin{itemize}[\setlabelwidth{$\gamma$}]
```

Within an IED list, the local control is executed just after the global control and, therefore, the commands in the local control can both augment and countermand those in the global control. Please note that the code in the local and global controls are executed in the same manner as normal LaTeX code. Therefore, the user should ensure that unwanted blank spaces do not appear in the controls. If a control definition is too long to fit on one line, shield the end of lines with "%" to prevent them from being interpreted as blanks. (Section 3.2.1 has some information on this topic.) Also, note that the LaTeX parser requires that braces be placed around commands with optional arguments that are placed directly within the optional arguments of other commands:

```
\begin{itemize}[{\mycmd[1]{example}}]
```

This IEEEtran IED implementation makes it easy to control IED lists, even when they are deeply nested.

The default spacings the IED lists use are stored in lengths which begin with "IEEE," i.e., `\IEEEilabel indent`. Changes to these "master" defaults are rarely needed and should be done only at the beginning of the document, *not in the IED list controls*. These constants will now be briefly explained.

`\IEEEilabelindent`: This length is the default amount the itemized list label boxes are indented from the left margin. IEEE seems to use at least two different values. For example, in *The Journal of Lightwave Technology* and *The Journal on Selected Areas in Communications*, they tend to use an indention equal to `\parindent`, while for *Transactions on Communications* they tend to indent itemized lists a little more (`1.3\parindent`). The shorter length is stored as `\IEEEilabelindentA` and the longer as `\IEEEilabelindentB`. The default is to use the shorter version. To use the longer version do a

```
\setlength{\IEEEilabelindent}{\IEEEilabelinden
tB}
```

at the beginning of the document.

\IEEEelabelindent: This length is the default amount the enumerated list label boxes are indented from the left margin. Normally, the same as `\parindent`.

\IEEEdlabelindent: Ditto for description list labels. Normally, the same as `\parindent`.

\IEEEiednormlabelsep: This length is the normal default spacing between the IED list label boxes and the list text.

\IEEEiedmathlabelsep: For nomenclature description lists (a list of math symbols and their explanations), IEEE usually increases the separation between the terms and the definitions. This length is set to the longer than normal length. To invoke its use, just issue the command `\usemathlabelsep` in a list control.

\IEEEiedtopsep: This length is the extra vertical separation put above and below each IED list. IEEE usually puts a little extra spacing around each list. However, this extra spacing is barely noticeable.

\IEEElabelindentfactori through **\IEEElabelindentfactorvi**: These contain the factors by which the effective `\labelindent` is reduced as the list nesting depth increases. The IEEE Computer Society normally decreases the amount of indention as the list nesting level increases because there isn't much room to indent with two column text. IEEEtran has an "automatic indention cutback" feature that provides this behavior. The actual amount the label boxes will be indented is `\labelindent` multiplied by the `\IEEElabelindentfactorX` corresponding to the level of nesting depth (where "X" is the nesting depth in roman numerals). This provides a means by which the user can alter the effective `\labelindent` for deeper levels. There may not be such a thing as correct "standard IEEE" values. What the IEEE Computer Society actually does may depend on the specific circumstances. The first list level almost always has full indention. The second levels usually have only 75 percent of the normal indentation. Third level and greater nestings are very rare, and probably don't use any indentation. These factors are not lengths, but rather constant macros like `\baselinestretch` so \renewcommand should be used if they need to be changed. The default values are

```
\IEEElabelindentfactori   1.0
\IEEElabelindentfactorii  0.75
\IEEElabelindentfactoriii 0.0
\IEEElabelindentfactoriv  0.0
\IEEElabelindentfactorv   0.0
\IEEElabelindentfactorvi  0.0
```

The use of these factors in IED lists may be suspended by issuing the command \nolabelindentfactortrue in a list control (which has the same effect as setting all the indent factors to 1.0).

Normally, IEEEtran automatically calculates `\leftmargin` based upon the current values of `\labelindent`, `\labelwidth`, and `\labelsep`. To stop this autocalculation so that a manually specified value of `\leftmargin` is used instead, just use `\nocalcleftmargintrue` in a list control. This feature should not be needed during the course of normal IEEE Computer Society-related work.

IEEEtran provides a means to manually specify the justification within the IED list label boxes. The commands `\iedlabeljustifyl`, `\iedlabeljustifyc`, and `\iedlabeljustifyr` can be used in a list control to justify the list labels to the left, center, and right sides, respectively. Itemize and enumerate lists automatically default to right justification, while description defaults to left justification. The justification commands should not be needed during the course of normal IEEE Computer Society-related work.

In addition to modifying the behavior of `itemize`, `enumerate`, and `description`, IEEEtran also provides the respective aliases `IEEEitemize`, `IEEEenumerate`, and `IEEEdescription`, which provides a way for the user to access the IEEE Computer Society style list environments even in the event another package is loaded that overrides the IED list environments.

10.1 Itemize

The itemized lists will normally automatically calculate the width of whatever symbol the current list level is using so that a user can just call `\begin{itemize}... \end{itemize}` without doing anything special. Furthermore, the auto-label-width feature will work properly even if `\labelitemX` has been redefined (where "X" indicates "i,ii, … iv," whichever is appropriate) *before* before the list begins. However, if any item symbols are to be specified via `\item[X]` (this is rare and may well be nonstandard as far as IEEE Computer Society related work is concerned), then the following form can be used:

```
\begin{itemize}[\setlabelwidth{Z}]
\item[X] blah
\item[Y] blah
.
.
\end{itemize}
```

where "Z" is the longest label in the list.

10.2 Enumerate

The important thing to note about enumerated lists is that the `\labelwidth` will default to the length of "9)" in the normal size and style. Therefore, the width of the longest label will have to be manually specified if *any* of the following conditions are true:

1. A top level list has more than nine items.
2. A relevant `\labelenumX` or `\theenumX` has been redefined.
3. `\item[X]` has been used to manually specify labels.
4. The labels are using a font that is not the normal size and style.
5. The enumerated list is nested (i.e., not at the top level) and is therefore not using Arabic digits as labels.

For example:

```
\begin{enumerate}[\setlabelwidth{12)}]
\item blah
\item blah
.
.
\end{enumerate}
```

10.3 Description

Generally speaking, the longest label width will always have to be specified for description lists. Furthermore, the author may wish to use `\IEEEmathlabelsep` for `\labelsep` when building a math symbol list. For example:

```
\begin{description}[\setlabelwidth{$\alpha\ome
ga\pi\theta\mu$}
\usemathlabelsep]
\item[$\gamma\delta\beta$] Is the index of..
\item[$\alpha\omega\pi\theta\mu$] Gives the..
.
.
\end{description}
```

Sometimes it can be difficult to ascertain from inspection which of the labels is the longest. For such cases, a little diagnostic code may be helpful to measure a length and then to display the result on the console:

```
\newlength{\mydiaglen} % put in preamble
.
.
\settowidth{\mydiaglen}{$\alpha\beta\gamma$}
\showthe\mydiaglen
```

11 Theorems and Proofs

Theorems and related structures, such as axioms, corollaries and lemmas, are handled in the traditional LaTeX fashion. The user must first declare the structure name via the

```
\newtheorem{struct_type}{struct_title}[in_counter]
```

command, where *struct_type* is the user chosen identifier for the structure, *struct_title* is the heading that is used for the structure, and *in_counter* is an optional name of a counter whose number will be displayed with the structure number and whose update will reset the structure counter. Most IEEE Computer Society papers use sequential theorem numbering throughout the entire work, so an *in_counter* is usually not specified. However, those papers that do use *in_counter* usually use "section" such that the section number is the first part of each theorem number. After the structure is defined, it can be used via

```
\begin{struct_type}[extra_title]
.
.
\end{struct_type}
```

where *extra_title* is an optional name that is displayed with the structure.

For example, the most common way to do theorems would be to use

```
\newtheorem{theorem}{Theorem}
```

followed as needed by environments like

```
\begin{theorem}[Einstein-Podolsky-Rosenberg]
```

Sometimes it is desirable that a structure share its counter with another structure. This can be accomplished by using the alternate form of `\newtheorem`:

```
\newtheorem{struct_type}[num_like]{struct_title}
```

where *num_like* is the name of an existing structure.

11.1 Proofs

Proofs are easily handled by the predefined proof environment. The same environment should be used to format related structures such as remarks, examples, and solutions (though these should not have a Q.E.D. box at the end).

```
\begin{proof}
.
.
\end{proof}
```

The Q.E.D. symbol, "□," is automatically placed at the end of each proof. If needed, the symbol can be manually accessed via the `\QEDopen` command.

12 End Sections

12.1 Appendices

The `\appendix` command is used to start a single appendix. An optional argument can be used to specify a title:

```
\appendix[Proof of the Zonklar Equations]
```

After issuing `\appendix`, the `\section` command will be disabled and any attempt to use `\section` will be ignored and will cause a warning message to be generated. (The single appendix marks the end of the enumerated sections and the section counter is fixed at zero—one does not state "see Appendix A" when there is only one appendix, instead "see the Appendix" is used.) However, all lower `\subsection` commands and the `\section*` form will work as normal as these may still be needed for things like acknowledgments.

`\appendices` is used when there is more than one appendix section. `\section` is then used to declare each appendix:

```
\section{Proof of the First Zonklar Equation}
```

The mandatory argument to section can be left blank (`\section{}`) if no title is desired. It is important to remember to declare a section before any additional subsections or labels that refer to section (or subsection, etc.) numbers. As with `\appendix`, the `\section*` command and the lower `\subsection` commands will still work as usual.

There is one appendix numbering convention used by the IEEE Computer Society: capital letters (e.g., "Appendix A"). However, subsection numbering conventions are preserved in the appendices, i.e., appendix subsections are labeled "A.1."

Some authors prefer to have the appendix label to be part of equation numbers for equations that appear in an appendix. This can be accomplished by redefining the equation numbers as

```
\renewcommand{\theequation}{\thesection.\arabi
c{equation}}
```

before the first appendix equation. For a single appendix, the constant "A" should be used in place of \thesection.

12.2 Acknowledgments

Acknowledgments and other unnumbered sections are created using the `\section*` command:

```
\section*{Acknowledgment}
\addcontentsline{toc}{section}{Acknowledgment}
```

Acknowledgments should be placed at the end of the paper, before the bibliography.

12.3 Bibliographies

Bibliographies are most easily (and correctly) generated using the IEEEtran BibTeX package [18], which is easily invoked via

```
\bibliographystyle{IEEEtran}
\bibliography{IEEEabrv,mybibfile}
```

See the IEEEtran BibTeX package documentation for more information.

When submitting the document source (.tex) file to external parties (such as IEEE), it is strongly recommended that the BibTeX .bbl file be manually copied into the document (within the traditional LaTeX bibliography environment) so as not to depend on external files to generate the bibliography and to prevent the possibility of changes occurring therein. The IEEE Computer Society cannot accept .bib files. For additional reference format examples, please see our Computer Society style guide at: http://computer.org/author/style/.

12.4 Biographies

Biographies for journal articles are created using the biography environment which supports an optional argument for the inclusion of a photo:

```
\begin{biography}[{\includegraphics[width=1in,
height=1.25in,clip,
keepaspectratio]{./shell.eps}}]{Michael Shell}
.
.
\end{biography}
```

Note the extra set of braces that are required to prevent the LaTeX parser from becoming confused when commands with optional arguments are used within an optional argument of another command. Alternatively, a LaTeX macro (command) could be defined to facilitate a shorthand notation for the author photos. If the optional argument is not used, space will be reserved for a photo and the message "PLACE PHOTO HERE" will be displayed in place of a photo.

IEEE Computer Society's algorithm for spacing around biographies can be a tad complex because esthetics must be considered. IEEEtran places `\vfil` above biographies. This allows the user to shove biographies down or up as desired by placing the infinitely more stretchable `\vfill` before or after the biographies.

The photo area is 1in wide and 1.25in long. The IEEE Computer Society recommends that author photo images should be of 300 dpi (dots per inch) resolution and in gray scale with 8 bits/sample. For more information on how to submit electronic materials, please visit: http://computer.org/author/transguide/electronicsub.htm.

If no photo is available, the `\biographynophoto` environment, which does not support an optional argument or reserve space for a photo, can be used instead.

Appendix A

Installing IEEEtran

LaTeX .cls files can be accessed system-wide when they are placed in the

```
<texmf>/tex/latex
```

directory, where `<texmf>` is the root directory of the user's TeX installation. It is recommended that the user create a subdirectory

```
<texmf>/tex/latex/IEEE
```

for all IEEE related LaTeX class and package files. On some LaTeX systems, the directory look-up tables will need to be refreshed after making additions or deletions to the system files. For teTeX and fpTeX systems, this is accomplished via executing

```
texhash
```

as root. MiKTeX users can run

```
initexmf -u
```

to accomplish the same thing.

Users not willing or able to install the files system-wide can make the copies local, but will then have to provide the path (full or relative) as well as the filename when referring to them in LaTeX.

APPENDIX B

Postscript/PDF Output

Unfortunately, many LaTeX systems are not properly configured to produce quality PostScript and/or PDF output. This is even more of a problem with IEEE Computer Society-related work because the font combination the IEEE Computer Society uses is known to cause problems with some LaTeX setups—especially those that were installed prior to mid-2002. This has been a chronic and very aggravating issue for many organizations that accept work created using LaTeX.

To assist IEEE Computer Society authors in detecting and correcting problems with LaTeX PostScript/PDF generation, the "Testflow" diagnostic suite was developed [21]. Authors are *strongly* encouraged to take the time to go through the testflow diagnostic and identify and correct potential problems *before* their LaTeX systems have to be relied on for production work. Papers with problems such as incorrect margins, font types, PDF format errors and/or improper font embedding can incur delays during the manuscript acceptance process.

APPENDIX C (OTHER USEFUL EXTERNAL PACKAGES)

C.1 The url.sty Package

Papers that contain URLs, email address, etc., can likely benefit from the use of the url.sty LaTeX package [22] which provides for more intelligent line breaking within such structures.

C.2 The IEEEtrantools Package

Some of the unique commands provided by the IEEEtran LaTeX class may be of use in non-IEEE related work using other class files (e.g., dissertations, technical reports, etc.). The IEEEtrantools.sty package [23] provides several popular IEEEtran commands including `\PARstart`, the IEEE Computer Society style IED list environments, and the IEEEeqnarray family of commands. The IEEEtrantools package is not needed under, and should not be loaded with, the IEEEtran class. See the IEEEtrantools documentation for more details.

APPENDIX D

Common User Mistakes

Most user mistakes with IEEEtran involve doing too much rather than too little. Older class files may have required hacks in order to get the formatting closer to that of the IEEE Computer Society. These tweaks are no longer needed. Users should carefully check all the loaded packages to ensure that they are still useful under the latest version of IEEEtran. Don't load packages just because "this is the way it always has been done." The same is true for manually adjusted spacing, margins, paper sizes, etc.

Another common category of error is to do things that compromise the portability of the LaTeX code. If the .tex source code will ever have to be shared with another party (IEEE Computer Society, etc.) it is important to code in traditional LaTeX fashion and not to utilize system dependent features, interfaces, graphics formats, drivers, or fonts—and pdfLaTeX users should ensure that their .tex code will also compile successfully without modification on traditional LaTeX/DVIPS systems.

Below are a few of the more commonly encountered mistakes to avoid.

Altering the default fonts. Authors should allow IEEEtran to manage the fonts. Do not attempt to use packages that override the default fonts such as pslatex, mathptm, etc. Likewise, authors should not attempt to alter the default font encoding (fontenc.sty) or input encoding (inputenc.sty).

Altering the default spacings, section heading styles, margins, or column style. Authors should not attempt to manually alter the margins, paper size (except as provided in IEEEtran class options), or use packages that do so (geometry.sty, etc.) There should be no need to add spacing around figures, equations, etc.

Using incorrect graphics file formats. LaTeX has always favored the use of Encapsulated PostScript (EPS) for graphics—and for good reason. EPS supports both vector (that is, containing objects such as lines, circles, etc., that are mathematically described) and bitmap (that is containing only samples in the form of pixels) images. The former should always be used for drawings, graphs, charts, etc., while the latter usually has to be employed with photos (because their contents usually cannot be easily described mathematically). The drawing and graphing tools used by the author should be capable of outputing *directly*[6] in vector (EPS) format. (Once an image in EPS vector form is converted to a bitmap form (GIF, TIFF, JPEG, etc.) it will almost always be irretrievably locked into bitmap form even if it is later converted back into EPS.) Vector EPS images can be scaled, rotated, and magnified without undergoing degradation such as pixelization or becoming gray or "jaggy." For photos, the IEEE Computer Society recommends the use of EPS (which is easy to directly import into LaTeX in a portable manner) or TIFF. The use of other graphic formats such as BMP, EMF, etc., is currently unacceptable for IEEE Computer Society journals. For more information on submitting electronic files, please visit: http://computer.org/author/transguide/electronicsub.htm.

Using older graphics packages. Authors should not use anything other than the graphics and/or graphicx (preferred) package for figures. Older interfaces such as psfig, epsf, etc., have been obsolete for many years.

Specifying graphics drivers. When loading the graphics or graphicx package, authors should not specify an optional graphics driver, e.g.,

```
\usepackage[pctex32]{graphicx}
```

because such code will not work properly on other systems that use a different graphics driver. If the default driver

[6] Once an image in EPS vector is converted to a bitmap form (GIF, TIFF, JPEG, etc.) it will almost always be irretrievable locked into bitmap form even if it is later converted back into EPS.

(which is loaded when none is specified) is incorrect for a LaTeX system, it is best to alter the `<texmf>/tex/latex/config/graphics.cfg` file than to have to manually specify a graphics driver in every .tex file.

Failing to properly divide long equations. It is the author's responsibility to ensure that all equations fit within the width of their columns. Admittedly, breaking an equation is not always easy to do and two column formatting places serious constraints on allowed equation width. However, only the author can divide his/her equation without unintentionally altering its' meaning or affecting readability. Using subfunctions is a valid way to reduce to width of an equation, but altering the math font size is not.

Manually formatting references. Not only is this error prone, but requires a lot of work as well. It is better to use the IEEEtran BibTeX style [18].

Appendix E

Known Issues

The small caps font used in the free LaTeX systems have about 80 percent the height of normal sized letters. However, the small caps font the IEEE Computer Society uses in the journals is slightly smaller with a ratio of around 75 percent. So, the widths of the section headings produced under the free LaTeX systems will be slightly wider than that used in actual journals. The small caps font used in many commercial LaTeX systems (such as those from YandY) has a ratio of about 65 percent. So, those systems will produce section headings that are narrower than those in some IEEE publications. Such variations should not be cause for concern.

hyperref.sty versions prior to 6.72u will interfere with the optional argument to `\appendix`.

The amsthm.sty package will cause a name clash error as it will try to define a proof environment which is already provided by IEEEtran.cls. It is generally not a good idea to load packages that attempt to provide replacements for any of the custom commands provided by IEEEtran.cls.

Appendix F

The IEEEeqnarray Commands

(Optional---for advanced users)

Appendix F has been made a supplemental document to this article. It can be found included with this file.

Authors should consider making supporting materials in articles supplemental to help meet page submission guidelines for both initial and final submissions.

Supplemental materials are peer reviwed and made available for free to IEEE Computer Society Digital Library visitors. For more information on supplemental materials, please visit: http://computer.org/author/transguide/SuppMat.htm.

Acknowledgment

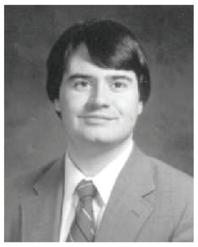
The author would like to thank Laura Hyslop, Cathy Cardon and Ken Rawson of the IEEE for their help and support in making this work possible. The knowledge and prior work of TeX gurus such as Donald Arseneau, Fred Bartlett, David Carlisle, Tony Liu, Frank Mittelbach, Piet van Oostrum, Roland Winkler, and Mark Wooding were instrumental in developing the complex IEEEeqnarray family of commands. The author is also grateful to Peter Wilson and Donald Arseneau for allowing the inclusion of their \@ifmtarg command. Finally, this work might not have been possible had it not been for the efforts of the prior IEEEtran developers: Gerry Murray, Silvano Balemi, Jon Dixion, Peter Nuchter and Juergen von Hagen. Their work still lives on to some degree within IEEEtran.

The IEEE Computer Society gratefully acknowledges the use of this edited version of Michael Shell's work for dissemination to its authors.

References

[1] M. Shell, "IEEEtran Homepage on CTAN," available at http://www.ctan.org/tex-archive/macros/latex/contrib/supported/IEEEtran, 2002.
[2] The IEEE website, available at http://www.ieee.org/, 2002.
[3] H. Kopka and P.W. Daly, *A Guide to LaTeX*, third ed. Addison-Wesley, 1999.
[4] R. Fairbairns, "The TeX FAQ," available at http://www.tex.ac.uk/cgi-bin/texfaq2html, 2002.
[5] A. Gefen, "Simulations of Foot Stability During Gait Characteristic of Ankle Dorsiflexor Weakness in the Elderly," *IEEE Trans. Neural Systems Rehabilitation Eng.*, vol. 9, no. 4, pp. 333-337, Dec. 2001.
[6] D. Arseneau, "The cite.sty Package," available at http://www.ctan.org/tex-archive/macros/latex/contrib/supported/ cite/, Nov. 2001.
[7] "The amsmath.sty Package," The Amer. Math. Soc., available at http://www.ctan.org/tex-archive/macros/latex/required/amslatex/math/, July 2000.
[8] M.D. Wooding, "The MDW Tools Package, available at http://www.ctan.org/tex-archive/macros/latex/contrib/supported/mdwtools/, Mar. 1999.
[9] D. Arseneau, "The cases.sty Package," available at http://www.ctan.org/tex-archive/macros/latex/contrib/other/misc/, May 2002.
[10] D. Carlisle, "Packages in the 'graphics' Bundle, grfguide.ps," available at http://www.ctan.org/tex-archive/ macros/latex/required/graphics/, Sept. 2001.
[11] 11] K. Reckdahl, "Using Imported Graphics in LaTeX2e. esplatex.ps or epslatex.pdf," available at http://www.ctan.org/ tex-archive/info/, Dec. 1997.
[12] C. Barratt, M.C. Grant, and D. Carlisle, "The psfrag.sty Package," available at http://www.ctan.org/tex-archive/macros/ latex/contrib/supported/psfrag/, May 1998.
[13] F. Mittelbach and D. Carlisle, "The array.sty Package," available at http://www.ctan.org/tex-archive/macros/latex/required /tools/, Sept. 2001.
[14] D. Arseneau, "The threeparttable.sty Package," available at http://www.ctan.org/tex-archive/macros/latex/contrib/other/misc/, May 1999.
[15] S.D. Cochran, "The subfigure.sty Package," available at http://www.ctan.org/tex-archive/macros/latex/contrib/supported/subfigure/, Apr. 2002
[16] S. Tolusis, "The stfloats.sty Package," documentation is in the stfloats.sty comments in addition to the presfull.pdf file, available at http://www.ctan.org/tex-archive/macros/latex/contrib/supported/sttools/, Oct. 1999.
[17] D. Carlisle, "The fix2col.sty Package," available at http://www.ctan.org/tex-archive/macros/latex/contrib/supported/carlisle/, Aug. 1998.

[18] M. Shell, “The IEEEtran BibTeX Style,” available at http://www.ctan.org/tex-archive/macros/latex/contrib/supported /IEEEtran/bibtex, 2002.
[19] P.W. Daly, “The balance.sty Package,” available at http://www.ctan.org/tex-archive/macros/latex/contrib/other/preprint/, Feb. 1999.
[20] S. Tolusis, “The flushend.sty Package,” documentation is in the flushend.sty comments in addition to the presfull.pdf file, available at http://www.ctan.org/tex-archive/macros/latex/ contrib/supported/sttools/, Oct. 1997.
[21] M. Shell, “The Testflow Diagnostic Suite,” available at http://www.ctan.org/tex-archive/macros/latex/contrib/supported/ IEEEtran/testflow, 2002.
[22] D. Arseneau, “The url.sty Package,” available at http://www.ctan.org/tex-archive/macros/latex/contrib/other/misc/, Mar. 1999.
[23] M. Shell, “The IEEEtrantools.sty Package,” available at http://www.ctan.org/tex-archive/macros/latex/contrib/supported /IEEEtran/tools, 2002.



Michael Shell (M'87) received the BEE and MSEE degrees in electrical engineering from the Georgia Institute of Technology, Atlanta, in 1991 and 1993, respectively, and is currently working toward the PhD degree. He has developed several all-optical packet-switched network subsystems and node demonstrations and is currently working on new analysis tools and techniques for the optical shared memory architecture class of packet switches. His research interests include all-optical packet-switched networks, high speed opto-electronic interface design, discrete simulation and exact Markov models for buffered packet switches. Mr. Shell is also the author of the most recent versions of the IEEEtran LaTeX class and BibTeX style packages and is the current maintainer of both.