Book Reviews

Raj Jain

FDDI Handbook. High-Speed Networking Using Fiber and Other Media

Addison–Wesley Publishing Company, Reading, 1994, pp. xxvii, 528, ISBN 0–201–56376–2.

FDDI (Fiber Distributed Data Interface) is the well-known standard for an optical fiber based high-speed LAN and MAN of ring topology, primarily intended as a follow-on for existing 10Mbps (Ethernet like) or 16Mbps (Token Ring like) LANs. Operating at a transmission speed of an order of magnitude higher, it also offers itself for the role of backbone for those LANs. Originally devised as an ANSI standard (ASC X3T9.5) already in 1987, it is also being adopted at the international level since 1989 as an ISO standard (IS 9314-1 to -6). The standard has a refined structure, consisting of a series of component documents, some of which are fully accepted (e.g. the first three parts of IS 9314) while others are still under consideration.

FDDI is presently gaining broad popularity, such that respective products experience a yearly growth rate approaching 100% in number of units, as well as sinking prices. Since the topic is a (relatively) hot one, there has already been plenty of published material, but mostly as research papers or short outlines in review books. To the best of knowledge of this reviewer, no comprehensive text covering this topic in such detail has appeared up to the publication of the reviewed book.

The book consists of 26 chapters, 6 appendices, a thorough list of references and a well-compiled index, all-in-all over 500 pages of easily read text. Although quite abundant in number, the chapters could be partitioned by subject closeness into several groups. The first group of chapters, encompassing the bulk of the book – some 300 pages in 18 chapters – addresses the functional issues introduced by FDDI: the main concepts of the standard and the architecturally related matter. Follows a relatively short group of practical stuff chapters concerned with purchase and installation of FDDI products. The book ends with a group of chapters of varied character covering network performance and analysis, and standardization associated issues.

After the enlightening Chapter 1, which is offering a really nice introduction and overview of the subject matter, Chapters 2 to 9 bring the fundamental description of FDDI, mostly following the original standard structure. The specific topics addressed are Media Access Control (MAC), Medium Independent Physical Layer (PHY), and a series of Physical layer Medium Dependent (PMD) choices, starting from the standard multimode fiber, up to single-mode and low-cost fibers, as well as twisted pair copper cables. Since bulding computer nets over optical communication links is a relatively recent technology, the author provides a well-placed introduction to the area, embracing fundamentals of underlying physics up to technical issues like e.g. connectors. The last of these chapters addresses the use of SONET (Synchronous Optical NETwork) links for building FDDI networks. This being the main part of the book, some 150 pages of text, it is accompanied in the following chapters with FDDI accessory topics: management, FDDI extensions and relationship with higher communication layers.

Management at the FDDI (i.e. at MAC, PHY and PMD) level, which is within the standard documents termed station management (SMT), is described in Chapters 10 to 12. Description of FDDI- II, FDDI's extension for telecommunications supporting isochronous service, is given in Chapters 13 to 15; the basic notions of the approach are explained, and supplemented by explications of the relative (sub)layers MAC-2 and PHY-2. FFOL (FDDI Follow On LAN), the envisaged backbone for FDDI networks operating at 1 Gbps and providing packet (i.e. isochronous, synchronous and asynchronous) as well as ATM cell transport services, is outlined in Chapter 16. The higher layers according to the OSI resp. TCP/IP architecture are covered in Chapters 17 and 18. First the Link Layer Control (LLC) providing the network layer a unified service over a variety of LANs is overviewed, subsequently followed by a discussion of network related issues in TCP/IP and OSI networks built over FDDI.

Chapters 19 to 21 cover buying and installing fiber cables, design and analysis of cable plants and directions for buying specific FDDI products.

Chapters 22 to 24 discuss performance issues of FDDI networks under heavy and normal load, and provide related analytical models. In Chapter 25 effects of errors are analysed. Finally, Chapter 26 shortly elaborates on standardization issues like conformance and interoperability testing.

The appendices furnish useful information on standard organizations, sources for further information on FDDI, a list of symbols used within the book, solutions to self-test exercices, status of FDDI standards and a list of addresses of FDDI products manufacturers.

It has been already noticed elsewhere [1] that the book is a *handbook*, and not a textbook. Therefore it is linearly structured, emphasizing the standard topics, that way enabling the readers to get a fast information on desired subjects. As the author states in the preface, "*the* FDDI Handbook *has been designed to be a comprehensive reference book providing an easy-to-understand explanation of key aspects of FDDI.*" It is the opinion of this reviewer that this goal is fully achieved, even more than that. Although written with other intentions, because of its good educational approach in subject matter exposition it can well be used as a textbook, too. Its style is systematical and methodical, the chapters and sections possessing an almost scrupolously enforced structure of the overview-detail-summary type. Additionally the chapters offer information on further readings and historical notes, as well as self-test exercices. Mostly welcome are the sidebars providing an entertaining way to exposition, and boxes supplying short reviews of important facts.

In conclusion, the book *FDDI Handbook. High-Speed Networking Using Fiber and Other Media* by Raj Jain is to be recommended as a valuable handbook for "*users, buyers, managers and designers of computer networking products*" interested in practically mastering FDDI technology, as claimed in the preface. Its additional value consists in furnishing comprehensive, detailed and up-to-date material for computer networks courses both on undergraduate and graduate level.

Vlado Glavinić
Faculty of Electrical Engineering and Computing
University of Zagreb
Zagreb, Croatia

References

[1] R. J. Clark and A. Mukherjee: "Book Review – FDDI Handbook", *ACM Comp. Comm. Rev.*, Vol. 24, No. 2, April 1994, 44.

Dave Jewel

Polishing Windows

Addison–Wesley Publishing Company Inc., Wokingham, England, 1994, pp xiii, 491, ISBN 0–201–62437–0.

If you have ever tried to write even a simplest program for Microsoft Windows (e.g. "Hello world") from a scratch, you experienced how frustrating job it could be. Writing more complex programs would be a real nightmare. Fortunately, there are software tools like Windows SDK (Software Development Kit) or Object Windows which make life easier, preventing you from writing hundreds of lines of code just to open a menu or display a warning message. These tools give support for almost all Windows actions. I stress almost, because there are some actions that are not supported. An example would be floating boxes utilized by most of the contemporary Windows graphical applications. Floating toolboxes (or tool palettes) are those floating windows that can be moved around the screen and positioned at user's convenience. They are particularly used in painting and drawing software (AutoCad, Corel PhotoPaint, PicturePublisher), but can also be found in some development systems either (Visual Basic, Borland C++). *Polishing Windows* demonstrates how to build these and many more professional-quality user interfaces which will improve the appearance of your software.

The author Dave Jewell is a freelance journalist, consultant and Windows developer. He writes programming and technical columns for many magazines in the UK. The book consists of eight chapters, extensively illustrated by a source code in C language and a lot of figures. Eight appendices at the end provide a complete set of Pascal units that duplicate the functionality of the C code for readers more familiar with this language. A disk with a complete C and Pascal source code is included. Although there is a general perception that C++ is the 'in' language to be used for Windows programming because of its object-oriented nature, Jewell has chosen the C language, since, according to his experience, at present no more than 20% of professional programmers use C++ for developing commercial applications.

The warm-up chapter *Laying some foundations* is intended mainly for users not familiar with Windows software development. The author introduces the principles of code writing he is going to use throughout the book. This is done through the development of a simple set of general purpose library routines. The author suggests how to create and handle string resources, how to use variables and how to create and use profile (.INI) files.

Chapter 2 *Toolboxes made easy* demonstrates how to create floating toolboxes efficiently, providing the maximum flexibility in positioning and arranging. Since each tool in the toolbox is represented by a small picture (toolbox bitmap), these bitmaps have to be designed to give the user an idea about what that particular tool is supposed to do.

This is followed by the chapter entitled *Classy subclassing*, in which the author demonstrates the power of window subclassing. This is done through the development of a simple set of subclassing routines used to add 3D effects to the dialog boxes. Subclassing is a concept taken from object-oriented programming which provides a high code reusability. It is a popular Windows technique because it allows the programmer to add custom features to a vast majority of existing Windows classes.

In the chapter *Getting some status* Jewell explains how to create status bars in the applications. Status bars at the bottom of the application's main window are used to show the current status of the application. For example, in a text processor they would usually display the current page number, line number and column position, which is obviously easy to obtain. A real problem arises when the status bar should display whether the CapsLock and NumLock keys are pressed, or when a help information has to be displayed on a menu pull-down. Ways of obtaining these features as well as some others are described in the examples of this chapter.

The ways of improving the look of dialog boxes are described in the fifth chapter *Better-looking dialog boxes*. The aim of this chapter is to show how to obtain Excel-style dialogs with minimum programming effort and without a need for large external libraries (such as Borland's BWCC.DLL). A special attention is paid to 3D-look of the dialog boxes.

Chapter 6 *Tempting toolbars* explains how to produce toolbars, a very important user interface element for most Windows applications. Again, the problem of button bitmaps is dealt with.

Chapter 7 *Super spin buttons* teaches us how to create spin buttons easily. Spin buttons are those tiny windows with two arrows on the right side that appear in some applications and are used to increment/decrement the value of a numeric variable (e.g. date and time in Windows Control Panel).

The final chapter describes the facilities and the use of CTL3D library and WiDE program included in the companion disk. CTL3D.DLL (originally written for Microsoft's Excel) is a library intended to improve the appearance of your Windows dialogs. It is not a part of the retail version of Windows 3.1, but is available as a part of Microsoft's support for Windows developers. WiDE (Windows Dialog Extension) is a shareware program that gives Windows 3.1 the look and feel of Windows 4.

The book follows a step-by-step approach to software development, which makes it easy to read even to a Windows programming newcomer. It includes a lot of highly reusable source code giving a valuable collection of facilities otherwise not supported in Windows SDK package. So, if you have tried to create toolbox palettes, spin buttons or toolbar facilities, but have given up, this is the book for you!

Julijan Šribar
Faculty of Electrical Engineering and Computing
University of Zagreb
Zagreb, Croatia

John H. Reif, Ed.

Synthesis of Parallel Algorithms

Morgan Kaufmann Publishers, San Mateo, California, 1993, pp.vi, 1011, ISBN 1–55860–135–X

This book deals with the development of algorithms for parallel machines from the point of view of complexity theoreticians. Their primary goal is to devise algorithms with guaranteed computational complexity in terms of measures such as time and space. In this process, a crucial role is played by the theoretical machine, or *computational model*. While the machine models for sequential machines served well in developing efficient sequential algorithms, the situation with parallel models is much more difficult. A variety of parallel computers developed by now showed to be too complex and too diverse to be approximated with a simple model. However, few models, such as parallel RAM (PRAM) machine that consists of synchronous processors communicating via shared memory, are well established. Many algorithms have been developed using these models and more and more authors, like Michael Quinn, suggest to investigate the appropriate PRAM algorithms as good starting points in developing algorithms for the existing machines.

The book gives a timely and encyclopedic review of results in complexity theory related to parallel computation models. The book consists of 22 chapters (plus an excellent introduction) written by top experts in each subject area. Each chapter is written as a self-contained overview of results for a particular area; usually the authors present the best known algorithms for the problems considered.

This book is centered around the *synthesis approach* to the development of parallel algorithms. The synthesis, or derivational, approach is based on the derivation of algorithms by combining several basic algorithmic techniques that are well understood and have the known complexity. This way, the more complex algorithms are developed by utilizing the simpler ones. The fundamental techniques spring from algorithmic graph theory, sorting techniques, computational geometry and linear algebra. Realistic problems usually employ techniques from several of these areas. All the methods that require logarithmic time (or some power of it) and polynomial number of processors versus the input size are considered parallelizable, and researchers are primarily looking for such algorithms. This class of algorithms is termed NC, from "Nick's class", which was named after Nicholas Pippenger.

The most important basic techniques arise from parallel graph algorithms. The methods for prefix sum calculations, which can be used for evaluating any associative operations over arbitrary inputs are described in the first chapter written by Guy Blelloch. Simple methods can easily calculate prefix sums in logarithmic time, hence there are simple NC algorithms. The author also gives numerous applications of these algorithms. Methods for list ranking, graph connectivity and Euler tour techniques are given in the chapter by Sara Baase. These NC algorithms are used in all subsequent graph algorithms. The optimal logarithmic algorithms for list ranking and parallel tree contraction, useful in the expression evaluations are given in a closing chapter of Part I of the book. Here, a parallel algorithm is optimal if its total "work" does not exceed the work needed by the best sequential algorithm for the same problem.

Advanced algorithms are given in Part II for randomized connectivity (H. Gazit), prefix-sums, list ranking and connectivity (U. Vishkin), lowest common ancestor computation (B. Schieber), open-ear decomposition (V. Ramachandran) and for recognition of the important class of chordal graphs (P. Klein) and applications. Here we mention especially algorithms for open-ear decomposition. The method is important since it replaces in a way standard algorithms for graph searching, for which no NC algorithms are known.

Part III of the book contains overviews of parallel sorting and computational geometry algorithms. Important techniques of random sampling are given in the chapter by S. Rajasekaran and S. Sen. The authors explain the significance of randomization starting with the familiar sequential quicksort algorithm and finishing with parallel computational geometry algorithms. The next chapter, by Richard Cole presents in contrast several deterministic algorithms for parallel sorting. M. Atallah and M. Goodrich present many deterministic computational geometry algorithms in the closing chapter of that part of the book.

Part IV contains two chapters on parallel algebraic algorithms. An overview written by J. von zur Gathen contains an illuminating view to several important algebraic problems and a connection between various problems in linear algebra and operations over polynomials. A paper on Newton iteration techniques by S. Tate shows how iterative methods can be used primarily in the basic algebraic operations.

Part V gives more advanced results: a chapter by V.Pan gives an algorithm for sparse linear systems, while D. Ierardy and D. Kozen give parallel algorithms for resultant computation problems that occur in systems of polynomial equations.

Part VI contains two extensions of the parallel tree contraction algorithm. E. Kaltofen describes approaches to general parallel evaluation of computational graphs, while J. D. Ullman gives an account on parallel logical inference and problems encountered in its parallelization.

Part VII deals with parallel graph matching and network flow problems, two well known combinatorial optimization problems. The chapter by V. Vazirani exploits the fascinating connection between the matching in graphs and linear algebra, while A. Goldberg describes the best known algorithms for network flows. The problem is known to be P-complete (which is an analogon to NP completeness in sequential computation) and it is unlikely that there is an NC algorithm for this problem. With the graph matching problem the situation is not so clear, and the question of the status of this problem with respect to parallel models is still open.

Parts VIII and IX deal with the basic issues in parallel computation: Faith Fitch compares various PRAM models, while limits of parallel computation and the notion of P-completeness is elaborated by R. Greenlaw. Even though the models considered are rich and expressive, many practitioners would find these models inappropriate for real-life problems and solutions with current parallel computers. In that sense, the book concludes with a look (by P.Gibbons) to more realistic asynchronous models of parallel computation that would account for the synchronization overhead incurred in real parallel programs.

As a conclusion, the book gives an invaluable first-hand account on the current state of the art in the area of parallel algorithms for problems in mentioned areas. It offers a rich source for a researcher in any of the fields touched. Each chapter is written as a self-contained overview of results for a particular problem; usually the authors present the best known algorithms for the problems considered. The overlap between chapters is minimal, except for the few basic algorithms in graph theory that are explained several times. The book is organized in such a way that the basic level algorithms precede the more advanced and efficient ones. Since the book is well organized, it can serve as a textbook. To that goal, exercises are included in most of the chapters.

For an implementer of parallel algorithms, this book cannot offer instant solutions, but it can serve as a good guide and starting point towards developing algorithms matched to the particular machine. All the algorithms are developed for abstract parallel machine models that very often are not implementable or appropriate for a particular machine. Additionally, many of the algorithms, especially in graph theory, can only be appreciated by a theoretician in the area since they are often derived by in depth investigating the structure of the problem. However, the purpose of the book, i.e. to explain fundamental methods in the development of parallel algorithms, is fulfilled in the best possible way. Although some more concise texts exist in the area, its depth and clarity makes this book irreplaceable as a timely reference material for a broad class of computer scientists.

Željko Žilić
Department of Electrical and Computer Engineering
University of Toronto
Toronto, Canada