

IN THE UNITED STATES DISTRICT COURT  
FOR THE DISTRICT OF COLUMBIA

UNITED STATES OF AMERICA,

Plaintiff,

v.

MICROSOFT CORPORATION,

Defendant.

Civil Action No. 98-1232 (TPJ)

**FILED UNDER SEAL**  
(Seal removed pursuant to court's  
October 14, 1998 Order)

STATE OF NEW YORK ex rel.  
Attorney General DENNIS C. VACCO, et al.,

Plaintiffs,

v.

MICROSOFT CORPORATION,

Defendant.

Civil Action No. 98-1233 (TPJ)

EXPERT TESTIMONY OF PROFESSOR DAVID J. FARBER

MY PERSONAL BACKGROUND

1. I am The Alfred Fitler Moore Professor of Telecommunication Systems at the Moore School of Engineering of the University of Pennsylvania in Philadelphia, Pennsylvania. I have held that endowed chair since 1994. From 1988 to 1994 I was a Professor of Computer Science

and Electrical Engineering at the University of Pennsylvania. From 1977 to 1988, I was a Professor of Electrical Engineering and Computer Science at the University of Delaware. From 1970 to 1977, I was an Associate Professor of Information and Computer Sciences and Electrical Engineering at the University of California at Irvine. A copy of my curriculum vitae is attached as Exhibit A to this testimony.

2. As you will see from my curriculum vitae, I have since 1988 been Director of the Distributed Systems Laboratory at the University of Pennsylvania. That Laboratory is the center for research activities in the general systems area for both computer sciences and electrical engineering at the University. In that capacity I am involved in high speed network research with emphasis on its impact on hardware and operating systems. Over the last ten years my research has generally focused on the areas of high speed networking and the implications of that networking on interconnections, protocols and computer hardware and software architecture. During that period, I have also taught graduate and undergraduate courses in the computer science field, including course work concerning software development.

3. From 1987 to 1988, I was Director of the Center for Networking Technology and Applications at the University of Delaware. As Director I was instrumental in the definition and implementation of the early stages of commercialization of the Internet. During the period I was at the University of Delaware, my research work was principally in the area of distributed systems with a focus on the hardware and software necessary to implement such systems. I also

led the creation of the University's own computer network and directed University-wide research in distributed systems.

4. In my years at the University of California at Irvine, I created and led what was then one of the largest computer research activity funded by the National Science Foundation -- the Distributed Computer Research Project. That Project created much of the software concepts for future distributed systems activities.

5. During my career, as detailed in my curriculum vitae, I have been Principal Investigator on a number of research projects (funded by government and private sources) related to computer software design and methodology. I have also been awarded two patents. I have published numerous articles in the computer science field and have lectured regularly in this country and around the world at various university and industry seminars.

6. I also have been regularly involved in activities in the business side of computers. I was a founder and Vice President of Caine, Farber and Gordon, Inc. ("CFG"), a private firm which is involved in program design methodology. CFG was an early creator of advanced software and compilers for microsystems and much of the Intel software support for the 8080 (the first pc-family microchip).

7. I have been on the technical staffs of Xerox Data Systems, the RAND Corporation and Bell Laboratories. I am currently on the technical advisory boards of a number of major corporations in the computer field, including Earthlink, Covad and Com21.

8. I have not testified in deposition or at trial as an expert witness in the last ten years, although I have consulted for parties in litigation in the past. I am being paid \$300 per hour for my work for the government in this case.

THE SUBJECT MATTER OF MY TESTIMONY ON BEHALF OF THE UNITED STATES

9. I have been asked by the Department of Justice to provide my expert opinions on software designed to function as a computer operating system and software applications that browse the World Wide Web via the Internet. The latter software is commonly referred to as an Internet or web browser (which I describe further below). In particular, I will discuss:

(A) the software development process and its implication for software products;

(B) the relationships between operating systems and application software, including Web browsers (as defined below);

(C) the significant inefficiencies in designing so-called operating systems which include inappropriate functions such as software applications (e.g. Web browsers); and

(D) the negative consequences of permitting Microsoft to add what are now applications to create an ever-larger, monolithic software package which Microsoft calls its "operating system" for personal computers.

These subjects will be addressed from the viewpoints of software developers, original equipment manufacturers (“OEMs”) and retail end users.

10. My testimony is based entirely on my lifetime of experience in the field of computer science. The matters about which I testify here concern commonly understood and accepted principles and practices in the field of computer science which are applicable to all software, whether developed by Microsoft or any other company. For that reason, I have done not, and do not believe it is necessary for me to do, extensive analytical work on the details of Windows 98 software (such as Professor Felten has done) in preparation for this testimony. My preparatory work (apart from preparing my report, this testimony and being deposed by Microsoft counsel) has been limited to a meeting with Professor Felten, review of certain memoranda filed by the parties with the Court, this Court’s opinion denying summary judgment in this case and the Court of Appeals’ decision of June 1998 concerning the Consent Decree entered against Microsoft.

#### TERMINOLOGY USED IN MY TESTIMONY

11. I and, I expect, other witnesses will use several technical terms for which I think it is useful to provide my definition for the Court. These definitions include the following:

(A) Software is the programs, routines, rules, instructions, and associated documentation of a computer system.

(B) An Operating System is software that controls the execution of programs on computer systems and may provide low-level services such as resource allocation, scheduling and input-output control in a form which is sufficiently simple and general so that these services are broadly useful to software developers.

(C) A Software Component is software that provides a specialized function or service in a form that can be incorporated into a variety of applications.

(D) An Application is a collection of software functions used to perform specific user-oriented tasks.

(E) A Browser is a software application that allows its users to examine the content of an information collection in a user friendly manner. What is commonly referred to as an Internet or Web browser permits the user to examine, display, scan, and navigate via the Internet a particular information collection on the World Wide Web.

### THE METHODOLOGY OF SOFTWARE DEVELOPMENT

12. The process of designing, writing, modifying, and testing software is one which is the subject of academic analysis and writing. Modern software is developed in small modules (i.e., routines, subroutines, methods, functions, procedures, etc.) which are aggregated into larger modules (i.e., files), which are in turn aggregated into software products.

13. The nature of the software development process, motivated in part by human limitations in managing design complexity and in part by a motivation of functional reusability,

imposes a modular structure on software. A module is the computer software equivalent of a paragraph, in which, via one or more routines, one reasonably-sized function is accomplished. Each routine is a sequence of instructions (i.e. lines of code -- often in one of a number of high level programming languages). An instruction is a call on a lower level routine requesting that the function of the lower level routine be performed using data provided by the higher level routine. For example, if a computer is asked to calculate the hypotenuse of a triangle, it will have to call a lower level routine which computes the sine or cosine of the angle. At the lowest levels, an instruction is directly interpreted by the processor after possible translation by a compiler. One example would be a command to add two integers together. Routines typically contain a few tens to a few hundreds of lines of code each.

14. These software modules are then “knitted together” into unified programs. That is, each software product is built up from simple low level routines that are then called by routines at a higher level of composition. Routines at each level are called by yet higher level routines until the desired functionality of the end product is achieved. In this manner, all software is built up layer by layer through the use of often large numbers of routines, but each with limited complexity.

15. As a result of this layering, software has an inherently malleable and modular structure which gives software developers broad freedom in combining (i.e. bundling) different functions into software products. This malleability also gives a software developer two related types of design freedom: (1) to integrate two separate cd-roms because the functions on one

particular cd-rom can be integrated by an OEM or retail end user with functions on another cd-rom and (2) to determine which functions to include within software sold as one product and which to separate and sell as a different product, whether produced by the same or a different software developer, for installation and use together by the a retail end user. Therefore, for example, software malleability provides a developer with the freedom to choose whether to incorporate a particular application (such as a Web browser) into a software product or sell that application separately for later combination by another software developer or retail end user with other software.

16. A common theme in software engineering and among software development methodologies is the advocacy for, and techniques in support of, the modular design and implementation of software. There are several efficiencies which result from development of software in this modular form. These include reduced errors in the development process, easier testing of products, reduced cost and complexity for maintenance and upgrades, and greater possible sharing of software modules among separate products. For example, the function of a computer which permits it to determine the sine of an angle should be modularized so that any application can call that function and, if there are any improvements or errors, the source can be more readily found and changed or corrected.

17. Applications may be large, often involving a very large number of routines. Examples of applications are Microsoft Word, Notepad and Speech Recorder -- as well as Internet Explorer. Software, including applications, must be stored in the form of files. The



formation of these types of files (sometimes called dynamically linked libraries or DLLs in the Microsoft Windows world) should be dictated primarily by considerations of storage/loading efficiency (time to prepare the software to run), file reusability (the ability to use all or most of file in more than one application) and execution performance. The most technically efficient size for a file is generally larger than a single routine and smaller than an entire application. Thus, the aggregation of routines into products involves two processes: the routines of the product must be aggregated into files, which must then be aggregated into products, often with connecting code.

18. All the routines that are called directly or indirectly by a program should be available when the program is being used. But whether those routines originate from one particular software program or another is irrelevant to the performance of the functions, so long as the software is written and installed such that the programs work together. Moreover, a software developer is free (subject to minimal limitations of no relevance here) to change the partitioning of routines among files at any time without changing their function or correct operation when the files are combined during execution in an end user's computer. Thus, there is generally no technical reason why a particular routine must be included in the same file with another routine so long as the routines are appropriately compiled and linked in the end user's computer.

19. There are several software tools (i.e., software analogous to machine tools in the manufacturing world) that are used by software developers to accomplish this process of

“knitting together” routines into files and, in turn, into programs. These tools include compilers, linkers and loaders. A compiler translates instructions (written in a language efficient for programmers) into the language understood by the computer hardware. A linker takes separately compiled program units (i.e., files or sets of routines) and combines them into a complete program. A loader loads programs, or parts of a program, into the memory of a computer in a form in which it can be executed. Programs are normally compiled only by the software developer and the resulting code files are delivered to the customer. Although the end user may not realize it, loading is typically done by the end user as part of his/her initiating execution of an application. Linking can be performed statically by the software developer or dynamically when needed for execution of an application.

#### APPROPRIATE METHODS TO DEVELOP APPLICATIONS TO RUN ON A PARTICULAR OPERATING SYSTEM

20. Developers of software, whether end user applications or OEM components, write their programs with the expectation that certain functions can and will be performed by the operating system of the computer on which the software will be used. The application invokes the operating system by calling routines supplied as part of the operating system. The interconnection is referred to as an application-programming interface (API). An API is the software convention that must be satisfied by a programmer when calling a function provided by the operating system. Access to and use of APIs allows: (1) an application developer to avoid what otherwise may be the expense of writing or purchasing all of the software necessary and (2) the application to run on a computer in conjunction with a specific operating system.

21. Developers of software applications depend on functions provided by the operating system. In that sense, application software developers and any operating system developer share a mutual dependency: applications developers rely on the presence of certain simple, general functions to which they interconnect their applications via APIs while an operating system developer must offer appropriate functions and APIs to which applications developers can write applications in sufficient quantity and quality that OEMs and end users will choose to buy both the operating system and applications.

22. From at least a technical viewpoint, the most efficient operating system is one that includes only those functions described in Paragraph 11(b) above which will be used by large numbers of application and component software development projects or are, by their unusual and peculiar nature, required to be at the operating system level. An example of the latter is security and protection of other operating system functions, which has become more critical as personal computers interconnect in networks which could otherwise invade and damage those computers' operating system functions. Moreover, only functions that are critical to the operating system itself or functions that the operating system makes available in the form of simple, general purpose APIs exist in a very efficient operating system. By way of example, network software that regulates the transfer of information to or from an ethernet card is normally part of an operating system while the file transfer client, which is further up the chain of routines, should not be in the operating system. This distinction is based on real time performance demands and security needs.

23. Inclusion of inappropriate functions at the level of what some software developers call an “operating system” is very likely to impose inefficiencies on application and other software developers, OEMs and retail end users. If the developers must use the particular functions, inappropriately placed in what I and others in the field refer to as “the operating system environment” (to distinguish it from what I consider to be an appropriate operating system), because of the manner in which that operating system environment is written or marketed, developers who need to write applications for that so-called “operating system” will, by definition, be forced to use that particular function regardless of whether that developer would prefer to use another software routine or develop a new one. Even if the structure and marketing of the operating system environment permits the addition of similar functions in addition to the functions provided by the operating system environment, the developer who wants or needs a different method of achieving a particular function must nevertheless write or buy often expensive, additional software. This duplication will likely: (1) increase storage requirements for different versions of software separately developed and included by the operating system vendor and application developer, (2) cause performance degradation from unused functions of the operating system and (3) increase risks of “bugs” (i.e., malfunctions) in software.

THE NEGATIVE EFFECTS OF MICROSOFT'S BUNDLING  
OF ITS SO-CALLED OPERATING SYSTEM WITH ITS INTERNET BROWSER

24. Microsoft claims in its memoranda filed with this Court that certain “efficiencies” result from its “integration” of some of the files (or DLLs) that are included in its Internet Explorer (IE) product as part of Windows 98 (which I refer to as an “operating system environment plus applications” to distinguish it from my definition of a true operating system and an operating system environment). The claims that efficiencies exist from this combination of functions are misleading. While the combination may offer certain efficiencies, these same efficiencies can be achieved without bundling of the Web browser software with what Microsoft calls its Windows operating system. This is because there are no technical barriers that prevent Microsoft from developing and selling its Windows operating system as a stand alone product separate from its browser software -- or other software functions beyond the appropriate operating system functions. Windows 98 (like all other software) necessarily consists of modules which are malleable and separable. There are no technical efficiencies for users achieved by combining Microsoft’s browser software with the remainder of the software sold as Windows 98 that could not be achieved by writing two programs in a manner that later could be loaded and “integrated” either by the retail end user (i.e., just as end users install any other application that runs on Windows) or by an OEM.

25. One example of such a division of the existing Windows 98 would be that which I understand Professor Felten has developed with his team. The software which is necessary to

accomplish Web browsing can be separated from the remainder of Windows 98 and sold on a separate disk (or some other distinct method). Yet, if an OEM or end user so desires, the operating system, operating system environment and browser application software can be installed by them to create the current version of Windows 98 (with the browser software labelled for commercial purposes as a Microsoft product called Internet Explorer); if that is not desired, an alternative Web browser can be installed with the Windows 98 operating system environment. Moreover, given the completely malleable nature of software and the apparently extensive number of files included in Windows 98, there are very likely to be alternative ways to separate Microsoft's web browsing software from those files which could and should be part of a Windows 98.

THE NEGATIVE TECHNICAL CONSEQUENCES OF PERMITTING MICROSOFT  
TO BUNDLE APPLICATIONS AND OTHER SOFTWARE FUNCTIONS INTO ITS  
OPERATING SYSTEM

26. I understand that Microsoft claims that it should be permitted to include any software it chooses in its Windows product so long as some efficiency can be articulated as a result of the "integration" of that software into that product. But if that standard were adopted by this Court, then any application could be bundled into Microsoft's Windows product -- regardless of the ability, inherent in the nature of software described above, to achieve that efficiency without combining the applications into an ever larger bundle of software which includes not only operating system functions but operating environment functions and applications. Taken to its logical extreme, that standard would mean that Microsoft could bundle together all its existing and future applications with its current (already massive) product sold as Windows 98.

Windows 98 (or whatever later version of Windows) could become the one and only universal software product, and only Microsoft could develop software for Intel-based personal computers. For example, Microsoft could next claim that, because there are some functions which overlap between Windows 98 and Microsoft's suite of applications called Office, now sold separately, those applications could be bundled into one product to achieve some "efficiency". But there is no such "efficiency" which cannot be achieved by the same separate distribution of those applications and their "integration" by an OEM or retail end user when the applications are installed (as is now the case).

27. In my view, any analysis of the propriety of such actions by Microsoft should take at least the three technical factors discussed above into account:

(a) for software, "efficiencies" achieved through combination of different functions into one so-called "integrated" software program can also be achieved without that combination by a software developer by simply separating that program into two or more pieces, distributed separately and, if so desired, integrated for use by an OEM or retail end user;

(b) combining applications with an operating system into a single product available only with all functions combined imposes technical inefficiencies for OEMs, other software developers and retail end users, including redundancy, performance degradation of unused software and increased risks of "bugs"; and

(c) any function provided by an operating system (as distinct from higher level files) that does not satisfy the criteria of simplicity, general applicability and accessibility reduces the efficiency of the operating system environment and applications that use it.

28. None of the above denies the possible convenience or preference of some users for “one stop shopping” for bundled products such as the current version of Windows 98 sold as one product by Microsoft. Those OEMs and retail end users who may find this convenience outweighs any technical inefficiencies described here can certainly still choose to buy Windows 98 in the form it now exists. But only the availability of an unbundled version of Windows 98 will cure the difficulties which arise for many OEMs, application developers and retail end users who may find too burdensome the problems arising from their inability to substitute different functions and applications (such as the Web browser) for use with only parts of what is now sold as Windows 98. The OEMs, developers and retail end users who do not want all the software that Microsoft sells as Windows 98 (including, for example, what Microsoft labels Internet Explorer) suffer the inefficiencies described above. Given the ease with which a properly structured browser application can be unbundled from Windows 98, I know of no technical reason why any OEMs, software developers or retail end users must suffer these negative consequences.

I declare under penalty of perjury that the foregoing is true and correct.

---

David J. Farber

Executed on October 8, 1998.



Exhibit A

**DAVID J. FARBER**

**PERSONAL INFORMATION**

Home Address:



Business Address:



**EMPLOYMENT EXPERIENCE**

The Alfred Filer Moore Professor of Telecommunication Systems, Moore School, University of Pennsylvania (1988 - present)

Research work has concentrated in ultra high speed networking and the implications of that on processor interconnect, protocols and software. This has created several joint study agreements with industrial research laboratories such as Bellcore and the RBOCS (Project Dawn - with MIT), IBM and Bellcore (Project Aurora - with MIT), and to becoming one of the principals of the NSF/Darpa research project in Gigabit Networking and Chairman of the Coordination Committee..

Director of the Distributed Systems Laboratory, University of Pennsylvania (1988 - present)

The DSL is the focus of the research activities in the general systems area of both the Computer Sciences and the Electrical Engineering Departments. The past year has seen extensive physical plant improvements as well as a major revamping of the educational and research programs.

Director of the Center for Networking Technology and Applications, University of Delaware (1987 - 1988)

Professor of Electrical Engineering and Professor of Computer Science, University of Delaware (1977 - 1988)

Research work concentrated in distributed systems with particular emphasis on the integration of software and hardware leading to efficient implementations of such systems. Had been the leader in the creation of a campus network and had spearheaded the formation of and was the Director of the Center for Networking and Distributed Systems Applications devoted to research in such systems. It was at Delaware that the creation of SODS was undertaken and where the CSNET mail system - MMDF was conceptualized and implemented.

Associate Professor of Information and Computer Sciences and of Electrical Engineering (with Tenure), University of California at Irvine (1970 - 1977)

Created and lead the Distributed Computer System Research Project ( 1971)- at the time the largest computer research activity funded by the National

## Exhibit A

Science Foundation. It created the software architecture that has formed the basis for much of the Distributed Systems activities that followed. It had a number of ideas such as Client/Servers, micro-kernal, process migration, message based IPC, contract resource allocation etc. Also conceived and directed the implementation of the first distributed token ring -- a forerunner of the IBM Token Ring. The activity transferred its technology into the Darpa work via collaborative efforts with IPTO and MIT.

Founder and Vice President of Research and Planning for Caine, Farber and Gordon Inc. (1970 -)

CFG is a key player in the Program Design Methodology area. Its products -- PDL (tm) is widely used in the Aerospace and Intelligence community. CFG was one of the very early creators of advanced software and compiler for micro systems and was the creator of much of the Intel software support for the 8080. It recently has created a state of the art compiler systems for the N Cube Inc systems.

Principal Member of the Technical Staff, Xerox Data Systems (1969 - 1970)

Was responsible for the design and development of a PL/I implementation for the Sigma computer series. After was Chief Technical manager for Xerox Computer Marketing.

Member of the Technical Staff, the RAND Corporation (1967 - 1969)

Was a principal researcher in several computer graphics projects and created an advanced language extension to the PL/I family for use in real time control. Was an advisor to the Air Force in several communications based activities as well as software design methodologies.

Supervisor; Systems Programming Department, Bell Telephone Laboratories (1965 - 1966)

Was responsible for the operation of the Holmdel Computer Centers system and applications staff. Was also a key player in Bell Labs activities which lead to the design and implementation of the Multics Operating system. In addition lead a group doing advanced graphics research.

Member of the Technical Staff, Programming Research Department, Bell Telephone Laboratories (1962 - 1965)

Was a co-author of the SNOBOL (I, II and III) programming language. SNOBOL is a major language in the field of string manipulation and several areas in expert systems. Was responsible for the compiler/interpreter for the language.

Member of the Technical Staff, Electronic Switching Engineering Systems Department, Bell Telephone Laboratories (1956 - 1962)

Was one of the system engineers responsible for the design of the world's first

## Exhibit A

Electronic Switching System. Particular emphasis was toward the software structure and software support infrastructure.

### ACADEMIC RESEARCH MANAGEMENT

Principal Investigator, Information and Society Project of Annenberg School, University of Pennsylvania on Electronic Commerce (\$200,000)

Co-Principal Investigator and conceptualized, - TeleMentoring: A Novel Approach to Undergraduate Computer Science Education, National Science Foundation 1992-1995 (\$400,000)

Principal Investigator, Aurora Project - A Gigabit Networking Testbed - effort in collaboration with Bellcore Incorporated, IBM Research Laboratories and MIT's Laboratory for Computer Science, National Science Foundation and Darpa (1989-present) (\$1,000,000)

Principal Investigator, Very High Speed Switching Studies - Project DAWN - Bellcore and the Bell Regional Companies (1988- present) (effort in collaboration with Bellcore Incorporated and MIT's Laboratory for Computer Science) (\$450,000)

Principal Investigator, Networking studies, AT&T Bell Labs 1990-1992 (\$150,000)

Principal Investigator, Project Mirage Darpa (1990-1991) studies in the formulation of high latency networking problems and models (\$300,000)

Principal Investigator ( Joint with Robert Kahn - CNRJ), Study in Very High Speed Networking, National Science Foundation (1988 - 1989 ) (\$50,000)

Director, Distributed Systems Laboratory, University of Pennsylvania (1988 - )

Director, Center for Networking Technology and Applications, University of Delaware (1987 - 1988)

Principal Investigator, Bitnet Modernization, National Science Foundation (1986-1988) [\$100,000]

Principal Investigator, Memnet, Northrop Corp. (1986 - 1988) [\$200,000]

Conceived and developing the MEMNET local network which includes complete software support (a NRTC cooperative research effort)

Co-Director, Educational Technologies Laboratory, University of Delaware (1985 - 1988)

A Laboratory devoted to the application of computers to the university functions with the main emphasis on ways of improving the productivity of the faculty.

Principal Investigator; Internet Mail Relays, ARPA IPTO (1983- 1984)

Principal Investigator; CSNET Phonenet and CSNET Relay, National Science

## Exhibit A

Foundation (1981-1985) [\$700,000]

Conceived and developed the Phonet system and Relay for CSNET and designed and implemented the MMDF system that implemented it.

Principal Investigator; Computer Message Services, U.S. Army DARCOM (1979-1984) [\$60,000]

Principal Investigator; Oversight of Distributed Processing Systems, National Science Foundation (1977-1980) [\$65,000]

Applying software design methodology to the monitoring of real time distributed systems. Developed the idea of the Overseer -- a monitoring environment.

Principal Investigator; Research in Distributed Processing and Office Systems, General Business Systems Division of IBM (1977- 1980) [\$450,000]

Developed SODS -- a capability based distributed software system for a new processor architecture. SODS is currently in heavy use at Bellcore.

Principal Investigator; Local Network Architecture, Advanced Research Projects Agency, Department of Defense (1976-1978) [\$80,000]

Developed the prototype of the LNI -- the r&d version of the Proteon Token Ring and the for-runner of the IBM ring

Principal Investigator; Audio Conferencing, The Institute for the Future (1974-1977) (\$100,000)

Principal Investigator; Network Security and Secure Protocols, Advanced Research Projects Agency, Department of Defense (1974- 1977) (\$300,000)

Principal Investigator; Distributed Computer Project, National Science Foundation (1971 - 1975) (\$1,200,000 -- the first such large award in computer science at the NSF)

A pioneering effort in the design of a fully distributed operating system with the first example of message passing and the first fully distributed token ring.

### **International Activities**

Active in collaborative activities to establish an international testbed involving the CEC, the USA and Japan

Involved in coordination activities involving networking in Japan and have been credited by key people in Japan as being the Grandfather of Japanese networking.

Actively involved in a multimedia experiment involving Fokus Germany and the Univ of Pennsylvania

### **EDUCATION**

## Exhibit A

University of Pennsylvania MA (honorary), 1988

Stevens Institute of Technology BSEE, 1956

Stevens Institute of Technology, MS in Math, 1962

Bell Telephone Laboratories Communication Development Program, 1963 (Equivalent to MS in EE)

### **HONORARY APPOINTMENTS**

Fellow of the IEEE

Fellow of the Glocom Insitute of Japan

Visiting Lecturer of the ACM

Distinguished Visitor of the IEEE

Traveling Lecturer of the International Computer Communications Council and the International Telecommunications Union

Distinguished Visitor of the IEEE Computer Society

Appointed to the Philadelphia Academy of Sciences

### **APPOINTMENTS (partial)**

Program Committee INET 93 and 94 and IFIPS 94 and ICC 95

Study Committee of the OECD on International Aspects of the HPCC

Scientific Advisory Board of the Swedish Institute of Computer Science and the Royal Institute of Technology (1994-present)

Chairman of the Advisory Board - the First Internet Society Conference - INet '92

Member of the Board of Directors of the ISODE Corporation (1992 - 1993)

Member of the Board of Directors of the Electric Frontier Foundation (1991 - )

Chairman of the Selection Committee for the Kobayashi Award of the IEEE (1990)

Member of the Board of Governors of the Academy of Sciences in Philadelphia (1989 - )

Board on Computer Science and Telecommunications of the National Research Council (1991 - 1995 )

Member of the Board of Trustees of the Corporation for Research and Educational Networking (1989 - 1991)

## Exhibit A

Policy Advisory Board, Chairman of the Networking Subcommittee, National Science Foundation, Office of Advanced Scientific Computing and Division of Network Research (1987 - 1989)

Board on Telecommunications & Computer Applications, National Research Council (1986 - 1990)

Founding Chairman of the Network Program Advisory Group (NPAG), Network Research and Infrastructure, NSF (1985 - 1987)

Active as a founder and technical manager of CSNET. On the CSNET Management Committee since the beginning.

Chairman; CSNET Executive Board, UCAR (1986 - 1988)

Past activities have included the SHARE Executive Board; The Fortran Standards Board; PL/I Standards Board; etc.

### **HONORARY SOCIETIES**

Fellow of the IEEE

Sigma Xi

### **Editorial Boards**

Editorial Board, Computer Networks (1980-1988)

Editorial Board, IFIPS Compact Journal (1983-1988)

Editor Series in Innovative Computing, Prentice-Hall (1987 - )

### **Recent Invited Addresses (selected) (last three years)**

Distinguished Visitor University of British Columbia Communications Series (1991)

External Opponent - Helsinki University of Technology, Esbo Finland (1991)

Distinguished Visitor of the University of California at San Diego (1992)

Keynote Speaker International Workshop on Advanced Communications and Applications for High Speed Networks in Munich Germany (1992)

Banquet Speaker IFIPS WG6 Vancouver Canada (1992)

Distinguished Lecturer, University of California at San Diego [1992]

Keynote Speaker, Technical University of Finland, [1993]

Invited Speaker Ministry of Post and Telecommunications Annual Conference Tokyo [1993]

Guest Speaker, Glocom Conference Oita Japan [1993]

Invited keynoter, Distributed Cooperative Systems Conference Tokyo Japan 1993

Featured Speaker at the GMD Annual Conference [1994]

### **Panelist and Speaker**

INet '92 Panelist - Future of the Internet

## Exhibit A

The EFF/ACLU Roundtable on Privacy and Ethics on the Electronic Frontier  
The NRC Roundtable on System Integration  
The First Conference on Computers, Privacy and Freedom Burlingame Ca  
panlist  
The IEEE Optical Switching Conference Monterey Ca - Speaker  
Keynote Speaker, Student Pugwash, Philadelphia PA  
Harvard School of Government Conference on the NREN

### **Invited Speaker**

Intel Corporation - internal technology series (twice)  
HP Laboratories  
IBM Research Hawthorne  
Finland Telecom  
HaL Computing  
Stanford University  
University of British Columbia  
University of Newcastle upon Tyme  
University of Sydney  
University of Melbourne  
University of New Zealand  
Keio University (Science Campus)  
CEC Annual Conference Brussels  
University of Tokyo  
Technical University of Helsinki

### **Patents Awarded**

Patent No. 5,329,623 awarded July 12th, 1994. "Apparatus for Providing Cryptographic Support in a Network," Jonathan M. Smith, C. Brendan S. Traw, and David J. Farber

Patent No. 5,353,419 awarded 10/4/94 "An Active Instruction Decoding Processor-Memory Interface" J. Touch and D. Farber

### **SELECTED PUBLICATIONS ( \* - student co-author(s))**

#### **Books**

The Office of the Future: Communication and Computers, R.P. Uhlig, D.J. Farber and J.H. Bair, North Holland Press, 1979.

#### **NATIONAL REPORTS**

Realizing the Information Future, National Research Council, 1994.

Toward a National Research Network, National Research Council, 1988

Transport Protocols for Department of Defense Data Networks, National Research Council, 1984.

Report on the Evolution of a National Supercomputer Access Network - Sciencenet,

## Exhibit A

National Science Foundation, 1984.

### Journal Articles

SNOBOL, A String Manipulation Language, Co-authored with R.E. Griswold and I.P. Polonsky, *Journal of the ACM*, 1964.

SNOBOL 3, Co-authored with R.E. Griswold and I.P. Polonsky, *Bell System Technical Journal*, 1966.

APAREL - A Parse Request Language, Co-authored with R. Balzer, *Communications of the ACM*, 1969.

Software Considerations in Distributed Architectures, D.J. Farber, *IEEE COMPUTER Magazine*, vol. 7, pp.31-35, 1974.

A Parallel Mechanism for Detecting Curves in Pictures, P.M. Merlin \* and D.J. Farber, *IEEE Transactions on Computers*, vol.24, pp.96-98, 1975.

Recoverability of Communication Protocols - Implications of a Theoretical Study, P.M. Merlin \* and D.J. Farber, *IEEE Transactions on Communications*, vol.24, pp. 1036-1043, 1976

The Convergence of Computing and Telecommunications Systems, D.J. Farber and P. Baran, *SCIENCE*, Special issue on Electronics, vol. 195, pp.1166-1170, 1977. Invited Article. (Also published in #5 of the AAAS Science Compendia, 1978.)

The National Research Network, D.Jennings, L. Landweber, I. Fuchs, R. Adrion, D. Farber, *SCIENCE* Feb 28, 1986. Invited article.

The World of Computer Networking in the 1990's, International Congress of Radio Sciences, Israel 1987

### Conference and other papers

Farber, D.J. "A Survey of Computer Networks." *Datamation* 18, 4 (April 1972), 36-39.

Farber, D.J. and F.R. Heinrich. "The Structure of a Distributed Computer System -- The Distributed File System." *Proc. International Conference on Computer Communications*, (Oct. 1972), 364-370.

Farber, D.J., M.D. Hopwood, and L.A. Rowe. "Fail-Soft Behavior of the Distributed Computer System." Technical Report #24, Department of Information and Computer Science, University of California, Irvine, California, (November 1972).

Farber, D.J. and K. Larson. "The Structure of a Distributed Computer System -- The Communications System." *Proc. Symposium on Computer-Communications Networks and Teletraffic*, Microwave Research Institute of Polytechnic Institute of Brooklyn, (April 1972).

Loomis, D.C. "Ring Communication Protocols." UC Irvine Distributed Computer Project, Memo 46-A, (May 1972).



## Exhibit A

Farber, D.J., J. Feldman, F.R. Heinrich, M.D. Hopwood, K.C. Larson, D.C. Loomis, and L.A. Rowe. "The Distributed Computing System." Proc. Seventh Annual IEEE Computer Society International Conference, (Feb. 1973), pp. 31-34.

Rowe, L.A., M.D. Hopwood, and D.J. Farber. "Software Methods for Achieving Fail-Soft Behavior in the Distributed Computing System." 1973 IEEE Symposium on Computer Software Reliability, (April 30, May 1-2, 1973), pp. 7-11.

Mockateiris, P., Lyle, M. and Farber, D. "On the Design of Local Network Interfaces", IFIPS 1977

Sincoskie, W. and Farber, D. "The Series/1 Distributed Operating System", Local Networks Conference 1981

Farber, D. "An Overview of Distributed Processing Aims." 1974 COMPCON.

Merlin, P., Farber, D. "Recoverability of Communications Protocols - Implications of a Theoretical Study" IEEE Transactions on Communications 1976

Farber, D. "Software Considerations in Distributed Architecture." COMPUTER 1974 (March).

Farber, D. "Information Engineering Perspectives". The NSF Conference on Information Technology, 1978

Farber, D. , Caine, S. "A Modular Office System", MICRODELCOM 1978

Von Glahn, P., Farber, D. and Walker, S. "The Trusted Office of the Future", Compcon '84

Many additional conference and symposium papers.

### **Current Papers**

"CapNet - An Alternate Approach To Ultra-high Speed Networks", Ming-Chit Tam, David J. Farber International Communications Conference, April 90, Atlanta Georgia.

"A Taxonomy Comparison of Several Distributed Shared Memory Systems" Ming-Chit Tam, Jonathan Smith, David J. Farber. ACM Operating Systems Review, June 1990.

"Mirage: A Model for Ultra High-Speed Protocol Analysis and Design" Joseph D. Touch and David J. Farber Proceedings of the IFIP WG 6.1/WG 6.4 Workshop on Protocols for High-Speed Networks, Zurich, Switzerland, 9-11 May 1989 also avail as: UPenn Dept of CIS Tech report MS-CIS-89-79, DSL-1, Dec.1989. This is under revision for IEEE Computer:

"The Mether System: A Distributed Shared Memory for SunOS 4.0" Ronald G. Minnich and Dave Farber Userix- Summer 89

"Reducing Host Load, Network Load, and Latency in a Distributed Shared

## Exhibit A

Memory Ronald G. Minnich and David J. Farber Proceedings of the Tenth (IEEE)  
Distributed Computing Systems Conference 1990

"The Gigabit Network -- does it fill a much needed gap?" presented as a Keynote  
and published in the proceedings of the International Workshop on Advanced  
Communications and Applications for High Speed Networks March 16 - 19 1992  
in Munich Germany

### SELECTED CONSULTING

Advisory Boards of Metricom, Com21, AOL and RadioMail.

Institute for Defense Analysis, Networks and Distributed processing (1991 - present)  
(continuing)

INTEL Corporation, Future LSI microprocessor organization and future business  
strategy (continuing) (1976 - present)

Hewlett Packard Research Labs and Corporate Engineering, Communications  
technology and office systems ( 1977 - present)

Ballistic Missile Defense Advanced Technology Center, High availability distributed  
systems (1975)

Bell Northern Research Laboratories (Canada), Office systems and high level protocols  
(1979)

The Federal Communications Commission, Office systems (1980)

T.J. Watson Research Labs of IBM, Communications, computer architecture and office  
systems (several periods from 1976 to 1990)

Northrop Research and Technology Centers - communication systems (1985-1988)

The Rand Corporation - communications (1967-1984)

## ADDDITIONAL PUBLICATIONS OF DAVID J. FARBER

Security for Virtual Private Intranets

William A. Arbaugh

James R. Davin

David J. Farber

Jonathan M. Smith

Cover Feature

IEEE Computer (Special Issue on Broadband Networking Security)

September 1998

Extensions to the PL/I Language for Interactive Computer Graphics

R. H. Anderson

D. J. Farber

RAND Corporation

Santa Monica, CA

RAND-RM-6028

Jan. 1970

A Secure and Reliable Bootstrap Architecture

William A. Arbaugh

David J. Farber

Jonathan M. Smith

IEEE Security and Privacy Conference

(An early version available as Technical Report MS-CIS-96-35, CIS Dept., University of Pennsylvania, December 2nd, 1996)

May, 1997

Oakland, CA

Automated Recovery in a Secure Bootstrap Process

W. A. Arbaugh

A. D. Keromytis

D. J. Farber

J. M. Smith

Internet Society 1998 Symposium on Network and Distributed System Security

March 11-13 1998

San Diego, CA

1998

State Caching in the EROS Kernel: Implementing Orthogonal Persistence  
in a Pure Capability System

Jonathan S. Shapiro

David J. Farber

Jonathan M. Smith

Persistent Object Systems: Principles and Practice

Morgan Kaufmann

Richard Connor

Scott Nettles

1997

(Presented at 7th Workshop on Persistent Object Systems May, 1996)

The Measured Performance of a Fast Local IPC

Jonathan S. Shapiro

David J. Farber

Jonathan M. Smith

Proceedings of the 5th International Workshop on Object Orientation in Operating Systems  
Seattle, WA

November 1996

Gigabit Object Network

Hyogon Kim

David J. Farber

Proceedings of IEEE Military Communication Conference (MILCOM '92)

San Diego

September 1992

The Failure of Conservative Congestion Control in Large Bandwidth-Delay Product Networks

Hyogon Kim

David J. Farber

Proceedings of International Networking Conference (INET '95)

Waikiki, Hawaii

June, 1995

A New Congestion Control: Framework for Large Bandwidth-Delay Product Networks

Hyogon Kim

David J. Farber

Proceedings of IFIP 6th International Conference on High  
Performance Networking (HPN)  
Palma, Spain  
Sept., 1995.

The Convergence of Computers and Communications - Part 2  
David J. Farber

ACM SIGCOMM Award Lecture  
August 30, 1995  
Cambridge, MA

Recoverability of Modular Systems  
P. M. Merlin  
D. J. Farber

Proceedings of the ACM SIGCOMM/SIGOPS Interprocess Communications Workshop  
Santa Monica, CA  
March 24-25, 1975

A Note on Recoverability of Modular Systems  
P. M. Merlin  
D. J. Farber

AFIPS Conference Proc. of the Nat. Comp. Conference  
1975

Recoverability of Communication Protocols: Implications of a Theoretical Study  
P. M. Merlin  
D. J. Farber

IEEE Trans. Comm.  
COM-24  
Sept. 1976

On the Recovery of Communication Protocols  
P. M. Merlin  
D. J. Farber  
1976

IEEE International Conference on Communications  
Graph Modeling of Computer Communications Protocols  
J. Postel  
D. J. Farber

Proc. of the Fifth Texas Conference on Computing Systems  
Austin, TX  
University of Texas  
October, 1976

Apparatus for Providing Cryptographic Support in a Network  
Jonathan M. Smith  
C. Brendan S. Traw  
David J. Farber  
U.S. Patent No. 5,329,623  
July 12th, 1994  
Filing Date: June 12th, 1992

A Tale of Two Major Networking Problems - One Organizational and One Technical  
David J. Farber  
The Harvard Information Quarterly  
Fall 1989

Gigabit Telerobotics: Applying Advanced Information Infrastructure  
Ruzena Bajcsy  
David J. Farber  
Richard P. Paul  
Jonathan M. Smith  
August 1994

1994 International Symposium on Robotics and Manufacturing  
Maui, HI

AURORA: An experiment in Gigabit Network Technologies  
Bruce S. Davie  
Jonathan M. Smith  
David D. Clark  
David J. Farber  
Inder S. Gopal  
Roch Guerin  
W. David Sincoskie  
David L. Tennenhouse  
Ahmed N. Tantawy

High Performance Communications  
Kluwer Academic Publishers  
January 1993

Cryptographic Support for a Gigabit Network  
Jonathan M. Smith  
C. Brendan S. Traw  
David J. Farber

Proceedings, INET '92  
June 15-18, 1992  
Kobe, JAPAN  
(Inaugural Conference of the Internet Society)

Traffic Characteristics of a Distributed Memory System  
Jonathan M. Smith  
David J. Farber

Computer Networks and ISDN Systems  
September 1991

Memory as a Network Abstraction  
Gary Delp  
David Farber  
Ronald Minnich  
Jonathan M. Smith  
Ming-Chit Tam  
July, 1991

Memory as a Network Abstraction  
Gary Delp  
David Farber  
Ronald Minnich  
Jonathan M. Smith  
Ivan Ming-Chit Tam  
Thomas L. Casavant  
Mukesh Singhal  
1994  
IEEE CS Press, Order Number 3032  
(revised version of IEEE Network paper)  
Piscataway, NJ

Readings in Distributed Computing Systems  
The AURORA Gigabit Testbed  
David D. Clark  
Bruce S. Davie  
David J. Farber  
Inder S. Gopal  
Bharath K. Kadaba  
W. David Sincoskie  
Jonathan M. Smith  
David L. Tennenhouse  
Computer Networks and ISDN Systems  
North-Holland  
January 1993

An Overview of the AURORA Gigabit Testbed  
D. D. Clark  
B. S. Davie  
D. J. Farber  
I. S. Gopal  
B. K. Kadaba  
W. D. Sincoskie  
J. M. Smith  
D. L. Tennenhouse  
Proceedings, INFOCOM 1992  
Florence, ITALY  
1992

The Series/1 Distributed Operating System: Description and Comments  
W. David Sincoskie  
David J. Farber

Proceedings, 21st IEEE Computer Society International Conference  
September 23-25, 1980  
Fall COMPCON (Distributed Computing)

The Trusted Office of the Future  
Peter G. Von Glahn  
David J. Farber  
Stephen T. Walker



Proceedings of the COMPCON '84  
Twenty-Eighth IEEE Computer Society  
International Conference  
February 27 - March 1 1984

CapNet - An Alternative Approach to Ultra High Speed Networks  
Ivan Ming-Chit Tam  
David J. Farber .

Proceedings, International Communication Conference '90  
April 1990  
Atlanta, GA

Mirage: A Model for Ultra High-Speed Protocol Analysis and Design  
Joseph D. Touch  
David J. Farber

Proceedings, Workshop on Protocols for High-Speed Networks  
Zurich  
May 1989

Reducing Host Load, Network Load, and Latency in a Distributed Shared Memory  
Ronald G. Minnich  
David J. Farber

Proceedings, 10th International Conference on Distributed Computing Systems  
Paris, France  
June 1990

A Taxonomy-Based Comparison of Several Distributed Shared Memory Systems  
Ming-Chit Tam  
Jonathan M. Smith  
David J. Farber

ACM Operating Systems Review  
July, 1990

An Analysis of Memnet: An Experiment in High-Speed Shared-Memory Local Networking  
Gary Delp  
Adarshpal Sethi  
David Farber

Proceedings, SIGCOMM'88 Symposium  
Stanford  
August, 1988

The Distributed Computer System (DCS): Its Final Structure  
Paul V. Mockapetris  
David J. Farber  
University of California, Irvine  
Technical Report  
1977  
Mockapetris\ &\ Farber\ 1977

Experience with the Distributed Computer System (DCS)  
Paul V. Mockapetris  
David J. Farber  
University of California, Irvine  
Technical Report 116  
1977  
Mockapetris\ &\ Farber\ 1977

The Distributed Computing System  
David J. Farber  
Proceedings, 1973 COMPCON IEEE  
1973

Networks: An Introduction  
David J. Farber  
Datamation  
April 1972  
Reprinted in IEEE CS tutorial on Distributed Processing (3rd ed.)

The Architecture of a Distributed Computer System- An Informal Description  
David J. Farber  
K. Larson  
Technical Report Number 11  
Department of Information and Computer Science,  
University of California, Irvine  
1970  
Cited in R. Kahn's Nov. 1972 Proc. IEEE article "Resource-Sharing Networks"  
Ronald G. Minnich  
David J. Farber

Mether: A Distributed Shared Memory for SunOS 4.0  
Proc. 1989 Summer USENIX Conference  
San Francisco, California  
June 1989  
J. R. Pickens  
D. J. Farber

The Overseer: A Powerful Communications Attribute for Debugging and Security in Thin-Wire  
Connected Control Structures  
Proceedings, International Computer Communications Conference  
also TR #75, UC Irvine, 1975  
August, 1976