

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

**A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600**

Florida International University

Miami, Florida

**THE APPLIED COMPUTER SCIENCE OF ECONOMIC ANALYSES:
INTERNET PRIORITIZATION AND INTERNET PRICING,
AND
SOFTWARE ENGINEERING AND SOFTWARE PRICING FOR THE MASS
MARKET**

A dissertation submitted in partial satisfaction of the

requirements for the degree of

DOCTOR OF PHILOSOPHY

IN

COMPUTER SCIENCE

by

Bernard C. Parenteau

1998

UMI Number: 9821696

**Copyright 1998 by
Parenteau, Bernard C.**

All rights reserved.

**UMI Microform 9821696
Copyright 1998, by UMI Company. All rights reserved.**

**This microform edition is protected against unauthorized
copying under Title 17, United States Code.**

UMI
300 North Zeeb Road
Ann Arbor, MI 48103

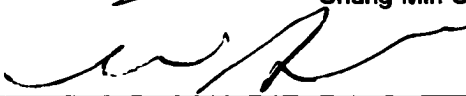
To: Dean Arthur W. Herriott
College of Arts and Sciences

This dissertation, written by Bernard C. Parenteau, and entitled THE APPLIED COMPUTER SCIENCE OF ECONOMIC ANALYSES: INTERNET PRIORITIZATION AND INTERNET PRICING, AND SOFTWARE ENGINEERING AND SOFTWARE PRICING FOR THE MASS MARKET, having been approved in respect to style and intellectual content, is referred to you for judgement.

We have read this dissertation and recommend that it be approved.

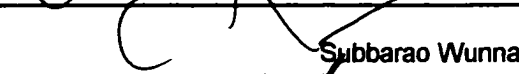


Chung-Min Chen





Wei Sun



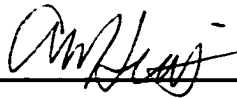
Subbarao Wunnava



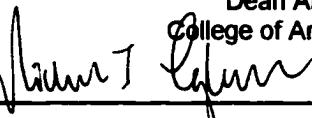
Naphtali Rische, Major Professor

Date of Defense: March 28, 1997

The dissertation of Bernard C. Parenteau is approved.



Dean Arthur W. Herriott
College of Arts and Sciences



Dr. Richard L. Campbell
Dean of Graduate Studies

Florida International University, 1998

© COPYRIGHT 1998 by Bernard C. Parenteau

All rights reserved

Dedicated to my sons; Vincent, Jean-Paul, and Jim.

ACKNOWLEDGMENTS

I would like to acknowledge and thank my wife Alyson for personally enabling me to accomplish my goals.

I would also like to acknowledge and thank my Major Professor Naphtali Rishe, as well as my committee members and other members of the computer science and economic faculties, for academically enabling me to reach this goal.

ABSTRACT OF THE DISSERTATION
THE APPLIED COMPUTER SCIENCE OF ECONOMIC ANALYSES:
INTERNET PRIORITIZATION AND INTERNET PRICING,
AND
SOFTWARE ENGINEERING AND SOFTWARE PRICING FOR THE MASS MARKET

by

Bernard C. Parenteau

Florida International University, 1998

Miami, Florida

Professor Naphtali Rishe, Major Professor

This research examines evolving issues in applied computer science and applies economic and business analyses as well. There are two main areas. The first is internetwork communications as embodied by the Internet. The goal of the research is to devise an efficient pricing, prioritization, and incentivization plan that could be realistically implemented on the existing infrastructure. Criteria include practical and economic efficiency, and proper incentives for both users and providers. Background information on the evolution and functional operation of the Internet is given, and relevant literature is surveyed and analyzed. Economic analysis is performed on the incentive implications of the current pricing structure and organization. The problems are identified, and minimally disruptive solutions are proposed for all levels of implementation to the lowest level protocol. Practical issues are considered and performance analyses are done. The second area of research is mass market software engineering, and how this differs from classical software engineering. Software life-cycle revenues are analyzed and software pricing and timing implications are derived. A profit maximizing methodology is developed to select or defer the development of software features for inclusion in a given release. An iterative model of the stages of the software development process is developed, taking into account new communications capabilities as well as profitability.

Table of Contents

I. INTRODUCTION	1
II. PRICING THE INTERNET	2
A. INTRODUCTION.....	2
B. WHAT IS THE INTERNET	5
1. <i>The History and Organization</i>	5
2. <i>The Communications</i>	11
C. ECONOMIC INCENTIVES AND CONGESTIBLE RESOURCES	16
1. <i>Economic Literature</i>	18
D. NETWORK DELAY: CAUSES, EFFECTS, AND REMEDIES.....	22
1. <i>Economic Analysis with Exogenous Network Delay</i>	23
2. <i>Endogenizing Network Delay: Economic Analysis</i>	31
3. <i>Endogenizing Network Delay: Internet Literature/RFCs</i>	32
4. <i>Endogenizing Network Delay: Implementation Issues</i>	36
5. <i>Summary</i>	38
E. QUALITY OF SERVICE AND PRIORITIZATION; BENEFIT ANALYSIS	39
1. <i>Economic Analysis</i>	40
2. <i>Performance Analysis of Multiple Priority Queues</i>	44
3. <i>Internet Literature/RFCs</i>	49
4. <i>Implementation Issues</i>	53
5. <i>Summary</i>	58
F. USAGE PRICING; BENEFITS ANALYSIS	59
1. <i>Economic Analysis</i>	60
2. <i>Internet Literature/RFCs</i>	62
3. <i>Implementation Issues</i>	63
4. <i>Comparisons with Telecommunications and Utility Pricing Schemes</i>	65
5. <i>Summary</i>	68
G. SUMMARY	69
H. EPILOGUE	72
III. SOFTWARE PRICING AND SOFTWARE ENGINEERING	73
A. INTRODUCTION.....	73
B. SOFTWARE PRICING AND TIMING	76
1. <i>Software Market Growth Model</i>	78
2. <i>Model Implications: Optimal Price and Time to Market</i>	89
3. <i>Upgrade Frequency</i>	91
C. OTHER PROFIT AND REVENUE CONSIDERATIONS AND DISCUSSION	93
D. SOFTWARE ENGINEERING IMPLICATIONS	97
1. <i>Software Engineering Development and LifeCycle Models</i>	100
E. THE SOFTWARE ARTS MODEL	104
1. <i>Rationale and Discussion</i>	104
2. <i>Notation</i>	105
3. <i>Development Steps</i>	108
F. SUMMARY AND CONCLUSIONS	112
IV. CONCLUSION	115
V. REFERENCES	118

I. Introduction

This research examines selected technology-related issues from economic and business perspectives, deriving applied computer science problems and solutions. In particular, this paper will discuss and analyze new issues that have arisen with the advances and proliferation of telecommunications, and the effects on applied computer science. The two main areas of investigation are internetwork communications services and software development.

Internetwork communications services are, in practice, the Internet. The issues raised in this area have to do with incentivization, efficient pricing and prioritization of usage. The Internet is the topic of a considerable amount of research and development in computer science, as well as business. The growth rate in recent years has been huge, and is estimated to continue for years to come. With this explosive growth have come performance problems. The analysis herein first considers the problems from an economic viewpoint, and then considers the applied computer science problems of the implementation of economically efficient systems. The topics considered include providing economic incentives to both suppliers and users to optimize welfare.

The second area of analysis is software engineering. The paradigm shift of software development for the mass market, rather than development to the specifications of a single customer, is considered. Moreover, the opportunities and exigencies of costless software distribution and software availability over the internet are discussed. The economics of standards and the maximization of revenues over the lifecycle of a software product are considered.

In both areas, new issues have been raised by the availability of widespread, cost-effective computer communications. This dissertation analyzes these issues from an economic perspective, proposes efficient methodologies, and discusses the applied computer science problems involved.

II. Pricing the Internet

A. Introduction

We would like to investigate how the internet might be priced and utilized more efficiently. The greatly increased usage of the Internet and the resultant performance degradation have focused attention on the inefficiencies of the traditional pricing and structure. This traditional pricing is flat-rate for unlimited usage of a single class of best-effort service. The traditional structure includes transparent routing through anonymous intermediate forwarding nodes and systems.

Although bandwidth has increased dramatically, traffic demand has more than kept pace. The occasional outages and routine delays are the price of growing popularity combined with inefficient incentives. There is no deterrent to a user casually downloading many megabytes in which he has only limited interest, while the user who is performing time-critical communications is placed in the same queues and suffers the effects. Clearly the latter's preferences are not being served. Nor is the interest of the service provider when one considers that the time-critical user would likely be willing to pay for increased performance but the provider is unable to offer it.

On the other side of the equation, there is limited incentive for a service provider to upgrade capacity when the service level that he provides to his customers is deleteriously affected by network delay outside of his control.

An economically efficient pricing scheme should allow users to specify the quality of service that they would like and would be willing to pay for. The scheme should also provide incentives to service providers to increase bandwidth to improve performance capabilities.

The following research will consider the economics of congestible resources, especially as applied to the case of the Internet. Three separate areas of research will be developed, and in each the economics will be analyzed, the applicable internet literature will be reviewed, and implementation details will be discussed.

The first area that will be examined is the structure of the internet, in which most of the network services are provided anonymously. The effects of this structure will be analyzed mathematically using economic theory, and a remedy will be proposed. The remedy is the reduction or elimination of

the anonymity via source specified routing. The applicable internet literature will be reviewed to determine the most efficient means of implementation of this remedy from a technical viewpoint. Practical and business implementation issues will also be discussed, and a conclusion as to the benefits and feasibility will be drawn.

The second area to be examined is the single class of service provided to all internet users. The economic benefits of a choice of multiple classes of service will be demonstrated. The rather extensive internet literature on this subject will also be reviewed and the various proposed technical solutions will be analyzed and found lacking. A simple, novel solution will be proposed. The technical details of its implementation will be discussed and analyzed in terms of required modifications to IP and additional overhead. A performance metric will be devised and the scheme will be analyzed via examples using currently common load factors. Practical and business implementation issues will also be discussed. A conclusion as to its benefits and feasibility will also be drawn.

The third area that will be examined is usage pricing. The assumptions in the economic literature will be discussed, as well as their lack of consideration of the overhead involved. The only applicable internet literature regards the methodology of metering, and it will be reviewed. Other implementation details will also be analyzed including the determination of whether to bill the sender or recipient. Comparisons to telecommunications and utility pricing are drawn and the applicable economic literature is reviewed. A conclusion will be drawn as to the benefits and feasibility in the current scenario and in the future.

Before beginning these analyses, the evolution of the Internet will be traced from its beginnings as a government-funded project of the United States Department of Defense, through its National Science Foundation funding period and expansion, to its mass-market commercial form today. The details of the communications methodology and some of the protocols involved will also be discussed.

We will finish this research by reviewing our conclusions and their benefits. We will find that Network Service Providers should be selected like long distance telephone companies are today, multiple classes of best-effort network services should be offered rather than the single class best-effort service offered today or even the proposed integrated services model, and usage charges may

be feasible when the explosive growth of the internet has slowed but are likely not feasible at this point in time.

B. What is the Internet

1. The History and Organization

The first point to note is that the Internet is not a static system. It is constantly evolving. To understand its structure today, one needs to be aware of its history. The Internet in the United States began as a project of the Department of Defense's Advanced Project Research Agency (ARPA) to link together a handful of supercomputer sites, mostly located at universities. The inaugural four universities (UCLA, Stanford, UCSB, Utah)¹ were linked in 1969, and by 1971 close to twenty nodes were connected^{2,3}. The communications protocols that are most common on the Internet today, TCP/IP (Transmission Control Protocol / Internet Protocol), were developed in the mid 1970's⁴ and evolved throughout that decade (as they continue to do). They were developed to be machine and medium independent. These protocols became standards on the ARPANET in the early 1980's. Although they are separate protocols (TCP runs on top of IP, as can other protocols such as UDP) they were developed together. They will be described in more detail below. In the early to mid 1980's, other networks communicating with TCP/IP began connecting to ARPANET as vendors such as 3COM and Sun came out with TCP/IP implementations⁵ and began using it in earnest. Many of these implementations of TCP/IP were based on the Berkeley stack developed for the BSD (Berkeley Software Distribution) version of Unix and ultimately funded by DOD ARPA grants⁶. (Recent sockets implementations, such as Microsoft's, are still based on the BSD Unix sockets.)

¹ Cerf, Vinton as told to Bernard Aboba, "How the Internet Came to Be", <[ftp://ftp.isoc.org/internet/history/cerf_Internet.txt](http://ftp.isoc.org/internet/history/cerf_Internet.txt)>, 1993

² *ibid*

³ Sterling, Bruce, "Short History of the Internet", <[ftp://ftp.isoc.org/internet/history/sterling_Internet.txt](http://ftp.isoc.org/internet/history/sterling_Internet.txt)>, February 1993

⁴ Cerf & Aboba (1993)

⁵ *ibid*

⁶ Even today, the "© Regents of University of California" from the BSD implementation can be found on commercial network software

In 1986 the National Science Foundation (NSF)⁷ began funding a network to connect supercomputer sites^{8 9}. This network, known as NSFNet, expanded rapidly to additional university and research sites. Its capacity increased from an original 56Kbps (56,000 bits per second) to T1 capacity (1.5Mbps) by 1989 and to T3 capacity (45Mbps) by 1991, keeping pace with its accelerating connectivity and usage as other educational and research networks connected to it¹⁰. NSFNet was run by ANS, a non-profit group created by Merit (a consortium of Michigan universities), IBM, and MCI¹¹. ARPANET was essentially superseded by "the Internet", of which NSFNet was a part, and whose funding agencies came to include not only NSF but also NASA and the Department of Energy. In the late 1980's, the U.S. military split off its own MILNet. ARPANET was retired in 1990 (responsibility had been transferred to the Defense Communications Agency in 1975)¹². NSFNet "acceptable usage policy" (AUP) was ostensibly restricted to educational and research endeavors and institutions. The NSF, however, did not comprehensively enforce its AUP, and furthermore envisioned its funding as temporary and advised the local networks to begin commercial preparations. With the upgrade to T3 lines in 1991, there was considerable commercial demand to join the network. Commercial providers had begun providing services such as email, and ANS spun off a for profit subsidiary, ANS CO+RE Systems¹³ to coordinate the interconnection of the educational and research nets with the commercial nets.

⁷ HREF=<http://www.nsf.gov>

⁸ Sterling (1993)

⁹ Cerf, Vinton, "A Brief History of the Internet and Related Networks", <ftp://ftp.isoc.org/internet/history/unknown_brief.txt>, undated, accessed March 1997, original <gopher://gopher.isoc.org:70/00/internet/history/_A%20Brief%20History%20of%20the%20Internet%20and%20Related%20Networks_%20by%20V.%20Cerf>, accessed March 1996

¹⁰ Sterling (1993)

¹¹ Merit Network, "NSFNET: Bringing the World of Ideas Together", <<http://www.merit.edu/nsfnet/nsfnet.overview>>, April 1992

¹² Cerf, Vinton and the Computing Research Association, "Computer Networking: Global Infrastructure for the 21st Century", <<http://www.cs.washington.edu/homes/lazowska/cra/networks.html>>, 1995

¹³ Merit Network Inc., "NSFNET: Transition to T-3", <<http://www.merit.edu/nsfnet/final.report/transition.html>>, undated, accessed March 1996,1997

The U.S. Corporation for National Research Initiatives (CNRI) participated in the commercial and educational interconnections and has continued funding up to the present¹⁴. The rapidly increasing connectivity has also continued to the present. The NSF phased out funding for the original NSFNet in 1995. Numerous vendors now provide backbone service, including BBN Planet¹⁵ which built the original ARPANet boards (and purchased SURANet in 1995, to go along with New England's NEARNet and San Francisco's BARRNet), PSINet¹⁶ which was created from the New York University Network System NYSERNET, ANSNet¹⁷ which includes ANS CO+RE mentioned above, and UUNet¹⁸ which began as a non-profit and is now for-profit. PSINet and UUNet, along with phone companies MCI and Sprint, and network and access companies Netcom and Apex, are the largest network service provider backbones¹⁹.

Many of these backbone providers have recently received investments from larger players. For example, America On Line (AOL) bought ANS in 1995. Microsoft, after acquiring a stake in UUNet, accounted for 20% of its revenue in 1995, and in February 1996 announced a joint expansion of the Microsoft Network with UUNet. AT&T acquired an equity position in BBN Planet in July 1995. UUNet, BBN Planet, and PSINet are all publicly traded companies, and all reported losses in 1995, as did AOL. Phone companies such as AT&T, MCI, and Sprint also offer backbone service. The service provider market was recently (February 1996) roiled when AT&T announced that it was entering the local access market with low prices for unlimited access. Customer response to AT&T's plan was overwhelming. MCI quickly matched the offer. In April 1996, BellSouth, GTE, and Bell Atlantic all announced that they would be offering Internet access by summer^{20 21}. In May, PacBell

¹⁴ Cerf & Aboba (1993)

¹⁵ HREF=<http://www.bbnplanet.com>

¹⁶ HREF=<http://www.psi.net>

¹⁷ HREF=<http://www.ans.net>

¹⁸ HREF=<http://www.uu.net>

¹⁹ Ubois, Jeff, "Peer Pressure", *Internet World*, vol. 7(8) August 1996, pg. 64, <<http://www.iw.com/1996/08/peer.html>>

²⁰ Whitefield, Mimi, "BellSouth will offer Internet services, too", *Miami Herald* 4/4/96, pg C1

²¹ Wingfield, Nick, "Bell Atlantic gets into Net market", *c/net*, <<http://www.news.com/News/Item/0,4,1075,00.html>>, April 10, 1996

did the same²². The service provider market, both local and national, is widely expected to consolidate. (On the day that the initial draft of this paper was first presented, April 2, 1996, the Wall Street Journal reported a merger in a related telecommunications market: two of the baby Bells, PacBell and SBC, merged. Later that month two more baby bells, Bell Atlantic and NYNEX, merged²³, and UUNet was bought²⁴)

MCI now runs the vBNS (very high speed Backbone Network Service), which is funded by the NSF, and provides connection between a handful of Network Access Points and university supercomputer sites. The vBNS bandwidth (capacity) is currently 155Mbps, and is expected to expand to 622Mbps this year (1996). Network Access Points (NAPs) provide interconnections between backbone networks, and are typically run by phone companies. There are four NAPs that include connections to the vBNS and receive funding from the NSF. These NAPs are run by PacBell in San Francisco, Ameritech in Chicago, Sprint in New York and Metropolitan Fiber Systems in Washington DC²⁵.

Interconnection points charge the backbone service providers fixed fees based on bandwidth. Backbone providers also use private connection points between themselves, as well as these NAP interconnection points. In fact, backbone providers endeavor to pass messages to other providers at the earliest possible point. The backbone service providers charge the local network providers fixed access fees based on bandwidth. Of course a backbone provider may also be a local provider, as most of them are. Finally, local service providers charge users fixed and/or hourly fees.

The NSF still funds connections for various institutions. The NSF also funds the Routing Arbiter project at Merit which involves the four NAPs mentioned above. Many networks interconnect at these NAPs, and the RA project facilitates the routing of packets between these networks. There is another

²² Lash, Alex, "Pac Bell chimes in with Net service", *c/net*, <<http://www.news.com/News/Item/0,4,1405,00.html>>, May 28, 1996

²³ Bell Atlantic, "Bell Atlantic and NYNEX Agree to Merger of Equals", <<http://www.bell-atl.com/nynex/>>, April 22, 1996 (original)

²⁴ MFS Communications Company Inc., "MFS and UUNET Announce Merger Agreement to Form Premier Internet Business Communications Company", <<http://www.mfsdatanet.com/mfs/news/Press/1996/Apr/30Apr96.html>>, April 30, 1996

²⁵ Merit Network Inc., "Merit Retires NSFNET Backbone Service", *MichNet News*, Vol 9(2), <<http://www.michnet.net/michnet/michnet.news/mnn.1995-02/nsfnet.html>>, Spring 1995

interconnection point, the Commercial Internet Exchange (CIX)²⁶ through which commercial networks are also connected. This was originally created to circumvent the NSF's acceptable usage policy. The CIX is run by a non-profit organization of the same name, which in turn is run by PSINet. NSF funded usage is still ostensibly restricted to educational and research traffic.

Although there is no single ownership of the Internet, there are a number of non-profit organizations that manage various functions. These groups and functions are coordinated, in part, by the Internet Society (ISOC)²⁷. The group that develops the technical standards for the Internet is the Internet Engineering Task Force (IETF)²⁸. The IETF is a volunteer organization that is comprised of a number of working groups which develop the standards in the various subject areas. The chairs of the working groups and the IETF chair form the Internet Engineering Steering Group (IESG)²⁹. The IESG manages the standards development process. The Internet Architecture Board (IAB)³⁰ is a body of the ISOC that adjudicates disputes and is responsible for overall architectural considerations. The Internet Assigned Numbers Authority (IANA)³¹ is chartered by the ISOC and run at University of Southern California's Information Sciences Institute (ISI)³². The Corporation for National Research Initiatives (CNRI)³³ runs the IETF Secretariat with funding from the U.S. government³⁴

European connection to the (U.S.) Internet, and the necessary adoption of the IP protocol standard, occurred in 1988³⁵. The European Internet oversight group is known as RIPE (European IP network) and has facilities in Amsterdam. The European effort has been particularly significant in the

²⁶ HREF=<http://www.cix.org>

²⁷ HREF=<http://info.isoc.org/home.html>

²⁸ HREF=<http://www.ietf.cnri.reston.va.us/home.html>

²⁹ HREF=<http://www.ietf.cnri.reston.va.us/iesg.html>

³⁰ HREF=<http://www.iab.org/iab/>

³¹ HREF=<http://www.isi.edu/div7/iana/>

³² HREF= <http://www.isi.edu>

³³ HREF=<http://www.cnri.reston.va.us/home.html>

³⁴ IETF, "IETF Home Page", <<http://www.ietf.cnri.reston.va.us/home.html>>, undated, accessed March 1996 -1997

³⁵ Segal, Ben, "A Short History of Internet Protocols at CERN", <<http://www.cern.ch/pdp/ns/ben/TCPHIST.html>>, April 1995

development of the world wide web and browser methodologies. That effort was spearheaded by researchers at CERN³⁶, the European Laboratory for Particle Physics in Geneva. The web is a set of protocols (e.g. Hyper Text Transfer Protocol or HTTP and Hyper Text Markup Language or HTML - more on these later) and conventions that allow users with a web "browser" to view documents in the appropriate format on a machine with a web "server". This work was expanded upon by NCSA in the U.S, which developed Mosaic. The primary author and co-founder of Netscape, Marc Andreessen, was the developer of the Mosaic project at NCSA as a student at Illinois. (As of March 7, 1996, outstanding shares of Netscape were worth more than \$1.5 billion³⁷). Web issues, such as transaction security, are now overseen in part by the World Wide Web Consortium (W3C) which is a non-profit organization run by MIT and INRIA (Institut National de Recherche en Informatique et en Automatique)³⁸ in collaboration with CERN. The W3C is based in Massachusetts³⁹.

As has been widely reported, the Internet is now truly global. Virtually all developed countries have connections, as do most developing nations (although their level of connectivity varies). Various sites have information on international connectivity and international service providers^{40 41 42}

43

³⁶ HREF=<http://www.cern.ch>

³⁷ Netscape Communications Corporation, US Securities and Exchange Commission Form 10-K, <<http://www.netscape.com/comprod/investor/10kpart1a.html>>, for the fiscal year ended December 31, 1995

³⁸ HREF=<http://www.inria.fr/welcome-eng.html>

³⁹ HREF=<http://www.w3.org>

⁴⁰ Internet Society, "International Connectivity", <<http://www.isoc.org/images/mapv15.gif>>, June 1996

⁴¹ Villasenor, Anthony, "Survey of International Internet Connectivity", <<http://nic.nasa.gov/ni/survey/survey.html>>

⁴² Commercial Internet Exchange Association, "CIX members", <<http://www.cix.org/CIXInfo/members.html>>

⁴³ Mecklermedia, "The List (Internet Service Providers)", <<http://thelist.iworld.com/>>

2. The Communications

Internet communications are done by a method known as "packet switching". This method breaks messages into acceptably sized small chunks, or packets, and sends each packet independently of the others. This is in contrast to what happens in a phone call where a portion of bandwidth is reserved for the call, regardless of whether either party is actively talking. With packet switching, everyone shares the connections as their flows of data dictate. Each packet goes through a series of "routers", which are computers that do just that: route packets on a first-in-first-out (FIFO) basis. These routers have routing tables that indicate where to forward packets, depending on "header" information in each packet. These routing tables may be dynamically updated to provide transparently robust service, as interconnected "neighbor" nodes notify each other of their status and which other nodes they can reach. The format of the packet header information, and what to do with it, is known as a protocol. Packets on the Internet use IP, Internet Protocol

The currently common version of IP is version 4, and it is extensively described in the IETF's RFC791⁴⁴ and RFC1812⁴⁵. A newer version, version 6, is in the Proposed Standard stage, and was first described in RFC1883⁴⁶. The RFC prefix stands for Request For Comments, and is the form of most all Internet standards as issued by the IETF. The IP header contains the source and destination addresses. These addresses are 32 bits long and are usually represented as "four octets", or four numbers separated by dots. An octet is 8 bits (a byte is also 8 bits), so it can represent a number from 0 to 255 (2^8-1). Each computer (host) on the Internet has a unique address. For example, the address of the computer at FIU known as Solix is "131.94.128.200"⁴⁷. The address of FIU's network is 131.94.0.0 meaning that the addresses of all machines on the network start with 131.94 (the third

⁴⁴ Information Sciences Institute, University of Southern California, "INTERNET PROTOCOL, DARPA INTERNET PROGRAM, PROTOCOL SPECIFICATION", <<http://ds.internic.net/rfc/rfc791.bt>>, September 1981

⁴⁵ IETF Network Working Group, F. Baker (ed.), RFC 1812: "Requirements for IP Version 4 Routers", <<http://ds.internic.net/rfc/rfc1812.bt>>, June 1995

⁴⁶ IETF Network Working Group, S. Deering, RFC 1883: "Internet Protocol, Version 6 (IPv6) Specification", <<http://ds.internic.net/rfc/rfc1883.bt>>, December 1995

⁴⁷ This example configuration is as of 1996

octet references a particular sub-net at FIU, while the fourth is a particular machine). Routing tables match destination addresses with the machine that should be used to forward the packet. In other words, they find the next hop to send the packet to. These machines, of course, have to be directly reachable from the machine sending the packet. So there needs to be a direct line to the machine, or the machine has to be on a common local area network (LAN). If it's on a common LAN, then the IP packet is sent inside the LAN protocol. The LANs at FIU run Ethernet, which is described by IEEE's 802.3 specification. Ethernet uses a local bus design, meaning all machines on the LAN listen in on a common line, and transmit on this line as well.

It's not feasible for every routing table to have all IP addresses, of course. So tables have default entries for all addresses not listed. The default address at FIU (131.94.128.100) is a router that is also connected to a node on SURANet's network⁴⁸. (Note the changes in network ownership and topology since this was originally written, as indicated in the footnotes. This is indicative of the rapidly evolving nature of the Internet.) So a packet from a machine on campus to a machine that's not on campus, goes to the default FIU router, which then sends it out on the SURANet network. SURANet is a wide area network (WAN) which is a point to point network, rather than a common bus like FIU's LAN. SURANet will route the packet to the appropriate node on its network. For example, a packet from FIU to the University of Florida⁴⁹ might go from SURANet's Miami node, to its Jacksonville node (or Atlanta node), then to UF's SURANet connection and UF's main campus router for further direction. A packet from FIU to, say Chicago, would go through SURANet's hub in Atlanta (again via the Miami and Jacksonville nodes) and then to a gateway to a national backbone, like the Washington DC NAP, MAE East. There it would be transferred to the NSP (backbone) to which the University of Chicago's service provider was connected.

The IP routing sends it to the appropriate host machine. IP is what's called a connection-less datagram protocol. That means each packet is considered an independent piece of information, unrelated to any other. So packets can arrive out of order, or they may not arrive at all, given

⁴⁸ SURANET became BBN Planet SouthEast in 1996

⁴⁹ The University of Florida is no longer on SURANET/BBN Planet SE as of 96/97. As of this time, packets to UF from FIU would go from BBN Planet SE to MAE East to BellSouth's gridnet to get to UF.

intermittent hardware or software delays or other problems. The protocol known as TCP (transmission control protocol) has the logic that disassembles the message into packets, passes these packets to IP, and receives them from IP at the other end. It also acknowledges packet receipt, initiates retransmission if necessary, and reassembles the packets into a message at the receiving end. TCP is a connection oriented protocol. It also does flow control, meaning that it won't send too many packets to the recipient before the recipient is ready for them. It also determines when a connection times out, based on either historical information for the destination, or default values. Information in the TCP header (which is inside the data portion of the IP packet) also indicates which process (port) the message is meant for. This is how a browser can communicate with a web server. The browser knows the web server process is listening for HTTP messages on a particular port (port 80 - unless explicitly specified otherwise by the URL via a ":nn" suffix, where nn is port number). Other services such as FTP (file transfer protocol), Telnet, and mail, listen for messages on their pre-defined port numbers (21, 23, and 25, respectively).

The browser and server software pass messages between themselves in HTTP format⁵⁰ (inside of TCP, inside of IP). HTTP messages contain some header data, such as date, forwarding and version number information, and a command such as get, put, head, post, delete, link or unlink. The world-wide-web refers to those sites supporting HTTP, and the data therein. (The web, then, is a logical network which is a subset of the physical Internet.) The web data is in HTML format. This may be text, pictures, audio, or anything. It also contains links to other addresses, hence the name hyper-text. The browser uses the last suffix of the page name to determine what type of data it is, and what to do with it. For example, html is a suffix meaning hyper-text markup language, as is txt for text, and gif for a particular image format. Plug-ins may be installed to automatically process the data within the browser, or helper applications can be defined to process it outside of the browser (for example, to call the sound card driver program for data with a wav suffix indicating a particular audio format).

⁵⁰ IETF Network Working Group, R. Fielding, RFC 2068: "Hypertext Transfer Protocol -- HTTP/1.1", <ftp://ds.internic.net/rfc/rfc2068.txt>, January 1997

But how does a browser get from an location name such as "http://www.mydept.myschool.edu/directory/mypage.html" to the appropriate IP address? This is done by the Domain Name Service (DNS). The DNS consists of a couple parts, one of which is the resolver. The resolver has a list of names that it knows, and a list of names that it has recently looked up (a cache). The resolver first looks for the name of the server (www.mydept.myschool.edu) in its directory or its cache. (The server name is the first text string delineated by slashes. Some browsers allow the http:// prefix to be omitted, so the server name is everything up to the first slash.) If the specific server name is not in the directory or the cache, the resolver looks for successively smaller portions of the name: mydept.myschool.edu, then myschool.edu, and failing that, edu. It must find at least a default listing of another machine whose DNS knows more details. For example, there are currently nine master DNSs in the U.S. When looking up www.mydept.myschool.edu for the first time, a local DNS will query a master edu DNS, whose IP address it has. This master DNS will return the IP address myschool.edu, and the local DNS will cache it, and then ask it the address of www.mydept.myschool.edu, and then cache that response. So now the browser knows the IP address of the page it wants to reach. The http:// indicates the type of data that can be expected, and to use an HTTP get command. It also will default to port 80, unless a port is explicitly specified as a suffix to the address (e.g. www.mydept.myschool.edu:70 would mean port 70). Port 80 is where the default web server is usually listening, although it can be configured for any port.

HTTP and the web are not the only applications that run on the Internet. In fact, they're rather recent arrivals. Other applications that are widely used include ftp (file transfer protocol), gopher (developed at the University of Minnesota, it provides a menu system of available files), telnet and rlogin (which allow a user to log in to a remote machine), and news and mail protocols. Also, TCP is not the only protocol that can run on top of IP. UDP (User Datagram Protocol) is a connection-less protocol that also runs on top of IP. An application such as real-time video or audio may not want to use TCP because of the additional overhead, and because if a packet doesn't get there immediately, it's of no use to retransmit it anyway. Unix NFS (Network File System) also uses UDP.

Now that we've described the physical and logical connections, we can discuss what's happening on the Internet. In short, usage is increasing very rapidly. All of the NAPs listed above have

statistics, as do a number of other sites⁵¹. This increasing usage, and the accompanying occasional periods of congestion lead us to question the economics of the current setup. Although bandwidth has increased greatly, so has demand. There's a computer adage (adapted from the transportation industry) that says demand will increase to fill capacity, and it's a good bet that this is applicable to the Internet as well. So among a number of interesting economic questions that the Internet raises (e.g. consolidation and foreclosure, network externalities, information economics, economic impacts and opportunities, zero marginal costs), the issue of pricing is prominent and unresolved. (There are a number of Internet sites of economic interest, including^{52 53 54}, which in turn contain numerous links). Once a practical, economically efficient pricing scheme has been devised, there are a number of computer science implementation issues to consider. In fact, the practicality of economically efficient schemes, in terms of implementation, is paramount.

The following sections will discuss the economic issues and literature, the internet literature, and some practical implementation issues.

⁵¹ National Laboratory for Applied Network Research, "Internet Information Presentation", <<http://www.nlanr.net/INFO/>>

⁵² Varian, Hal, "The Information Economy", <<http://www.sims.berkeley.edu/resources/infoecon>>

⁵³ Mackie-Mason, Jeff, "Telecomm Information Resources On the Internet", <<http://www.spp.umich.edu/telecom/telecom-info.html>>

⁵⁴ Massachusetts Institute of Technology Research Program on Communications Policy, <<http://far.mit.edu/Workshops/dist.html>>

C. Economic Incentives and Congestible Resources

The internet is unusual in that it is a privately provided public resource. As such, there need to be proper incentives for both producers (service providers) and consumers (users).

On the consumer side, we have billing schemes that currently are essentially flat-rate or by-the-hour charges. These do not distinguish between users who place a heavy load on the resources and those who place a light load, and so make the internet susceptible to the problems of the commons described in economic literature. The form of "overgrazing" to which the Internet is prone is congestion. A number of alternative pricing schemes have been proposed. There are, however, many complications inherent in Internet pricing that not all of these schemes address. In particular, there is a huge, diverse, disparate, installed base of users and providers already up and running. Any pricing scheme has to work within the framework already in place. In addition, the Internet is made up of many separate networks. This severely complicates matters in that no node has knowledge of the entire net's condition at any point in time. Also, each session might involve a large number of individual messages, and each message might traverse a large number of links to reach its' destination. This presents both pricing and accounting difficulties. The pricing difficulty is that the unit of measure (many packets each forwarded by many nodes) is transparent to the user, and so usual pricing and allocation mechanisms (e.g. auctions) cannot be used. The accounting difficulty stems from the same fine granularity of the unit of measure. Even in telephone companies, where the billing unit is a single call, the overhead cost of accounting and billing is significant. In the case of the internet this is magnified many-fold, as an equivalent to a call would involve many messages, each likely made up of many packets, and each of these forwarded by a handful of independent entities. So incentivizing pricing is deeply problematic.

Another issue affecting both consumers and producers is the single quality of service currently offered by the internet. Consumers are unable to specify high priority traffic, or to signal their willingness to pay for it. Their options are limited. Producers are unable to offer a noticeably higher

grade of service when the majority of the delay is outside of their control due to the integrated nature of the internet.

In addition, there have to be incentives to producers to provide adequate levels of service. In the anonymously cooperative and transparent forwarding scheme of the internet, these incentives may be lacking. As described above, a typical packet traverses several nodes and networks en route to its destination. However the user, who is paying for the services directly or indirectly, is not aware of all the networks involved in the transmission, and not aware of where delay is originating. This lack of "accountability" is driving many corporate customers to develop private networks, and limiting the potential welfare (usefulness) of the internet.

1. Economic Literature

Bohn, Braun, Claffy, and Wolff addressed the problem of internet congestion in a 1994 paper⁵⁵. They suggest that the existing IP header priority field could (again) be used to depart from the single FIFO queue model used by routers today. They note that relatively new on-line applications, many non-TCP, have the potential to swamp the existing infrastructure. And as was the case in the late 1980s when traffic increases overtook bandwidth, the IP priority field could be used to mitigate the situation. They suggest a form of soft quotas and voluntary prioritizing for the allocation of priority usage. The paper is notable in its knowledgeable treatment of the computer science aspects of the problem, and its proposed solution meshes closely with the proposed IPv6 standard. The paper also reports the results of traffic analysis and a simulation.

A paper by Scott Shenker of Xerox describes pricing by Quality of Service required⁵⁶. However the paper notes that billing could be problematic as there is no accounting infrastructure in place. Also, the pricing issue is not probed beyond differentiating prices based on user application requirements. In a separate paper, with co-authors Cocchi, Estrin and Zhang⁵⁷ (November 1993), a more detailed pricing scheme is described. This pricing scheme describes desirable service qualities for various applications (e.g. delay is less important for email than for telnet, also a tight time bound is required for real-time applications but occasional packet loss is not especially bad), and assigns variable pricing parameters to these quality of service measures depending on the application. The accounting and billing issues are not addressed.

Another somewhat more recent paper (revised April 1995) by Edell, McKeown and Varaiya⁵⁸ describes pricing and billing at the TCP level. This paper involved an actual experimental

⁵⁵ Bohn, R., Braun, H., Wolff, S., Claffy, K., "Mitigating the coming Internet crunch: multiple service levels via Precedence", <<ftp://ftp.sdsc.edu/pub/sdsc/anr/papers/precedence.ps.Z>>, March 1994

⁵⁶ Shenker, Scott, "Service Models and Pricing Policies for an Integrated Services Internet", <<ftp://ftp.parc.xerox.com/pub/net-research/policy.ps.Z>>, undated, accessed March 1996

⁵⁷ Cocchi, R., Shenker, S., Estrin, E., Zhang, L., "Pricing in Computer Networks: Motivation, Formulation, and Example", <<ftp://ftp.parc.xerox.com/pub/net-research/pricing2.ps.Z>>, November 1993

⁵⁸ Edell, R., McKeown, N., Varaiya, P., "Billing Users and Pricing for TCP", <<http://paleale.eecs.berkeley.edu/~edell/papers/Billing/article.ps>>, April 1995

implementation of the scheme at UC Berkeley. The scheme addresses the issue of willingness and ability to pay. Although the message passing overhead proved to be acceptable within the Berkeley test in an intranet environment, it is questionable whether it could scale to the Internet. Also, billing at the TCP level would miss the potentially considerable amount of non-TCP traffic. This is especially important considering resource intensive real-time applications may not use TCP.

A group at the University of Texas, including economist Dale Stahl, has looked at network computing with priority classes and separately taken the Internet as a special case⁵⁹. The idea in the generalized network computing model is that there are a number of computers which can provide networked users with their desired results. The prices charged for services vary dynamically as the loads vary stochastically. The price variation is done by an iterative tatonnement process that contains a weighting factor which can dampen the price fluctuation. Welfare is shown to increase in an example of single priority charges (essentially usage charges) versus no usage charges. The welfare difference between the case with usage charges and the case without them is shown to increase as traffic increases. The Internet is taken to be a special case of the network resource problem, in which the services provided are message forwarding. A drawback in their approach is that the user's delay contra-utility function is needed if the process is to be done by a smart agent (program) and thus transparent to the user. It would be non-trivial to design and inform such a smart agent, and it would be burdensome for the user to decide manually for every packet.

An additional article by the same group shows the economics benefits of a single priority class metered for usage pricing⁶⁰.

Hal Varian and Jeffrey Mackie-Mason are among the few leading economists noticeable in proposing detailed responses to the usage pricing issue. They have a number of papers on Internet

⁵⁹ Gupta, A., Stahl, D., Whinston, A., "An Economic Approach to Networked Computing with Priority Classes", <<http://cism.bus.utexas.edu/alok/nprice/netprice.html>>, December 1994

⁶⁰ Gupta, A., Stahl, D., Whinston, A., "Managing the Internet as an Economic System", <<http://cism.bus.utexas.edu/alok/nprice/netprice.html>>, July 1994

economic topics^{61 62 63 64 65}. Their paper on the pricing of congestible resources (November 1994) develops the economic theory that should be the basis of any pricing scheme.

The theory basically states that given a congested resource, the price one pays to send a message (one's utility) should reflect the loss of utility inflicted on other users whose messages are waiting. They develop this theory further describing numerous sub-topics, including consumer and producer optimization, free entry, and capacity expansion. They show that in the competitive case, producer optimization results in social welfare optimization. Their perfectly reasonable model contains two price elements: connection charges and usage charges. It is shown that if the marginal cost of capacity is low then connection charges will dominate, and vice versa. As they note, this is related to a paper by Scotchmer⁶⁶ in which she shows that connection fees decrease with the number of competing firms. They also show that pricing may be higher or lower in the monopoly case, depending on the variation in user preferences.

Another of the Mackie-Mason and Varian papers describes a "smart market" approach to message pricing. In this scheme, users bid the maximum price they are willing to pay to send their message. The highest bid messages are sent first. In any given time interval, the lowest bid message that gets sent sets the price for all messages sent. This is an economically reasonable scheme, but the overhead in bidding each individual message/packet price would be considerable. More critically, the proposal implicitly assumes the Internet to be a monolithic entity, which it's not. When a user bids a price for a message, how is that apportioned across the multiple networks that forward the message? Even in a single owner network, the packets traverse many nodes and it's not

⁶¹ Mackie-Mason, J., Varian, H., "Economic FAQs About the Internet", <http://www.spp.umich.edu/spp/papers/jmm/Economic_FAQs.ps.Z>, June 1995

⁶² Mackie-Mason, J., Varian, H., "Some Economics of the Internet", <http://www.spp.umich.edu/spp/papers/jmm/Economics_of_Internet.ps.Z>, revised February 1994

⁶³ Mackie-Mason, J., Varian, H., "Pricing Congestible Network Resources", <http://www.spp.umich.edu/spp/papers/jmm/Pricing_Congestible/eeee.ps.Z>, revised November 1994

⁶⁴ Mackie-Mason, J., Murphy, L., Murphy, J., "The Role of Responsive Pricing in the Internet", <<http://www.spp.umich.edu/spp/papers/jmm/respons.ps.Z>>, August 1995

⁶⁵ Mackie-Mason, J., Varian, H., "Pricing the Internet", <http://www.spp.umich.edu/pps/papers/info-nets/Pricing_Internet/Pricing_the_Internet.ps.Z>

⁶⁶ Scotchmer, S. (1985) "Two-tier pricing of shared facilities in a free-entry equilibrium" *Rand Journal of Economics*, 16(4), 456-472

clear what should happen when a packet's bid price enables it to be forwarded from one or more nodes, but then cause it to be delayed indefinitely in a "higher-priced" node. These issues appear insurmountable for the "smart market" approach.

D. Network Delay, Causes, Effects, and Remedies

One of the problems with the current structure of the internet is that the source of delays are anonymous. An ISP knows that there will be occasional (or regular) significant delays beyond his control. The customers of the ISP also know that many delays are beyond local control. However the customer cannot determine when delays are due to the local ISP and when they come from elsewhere, or where they come from. This makes the determination of an ISP's quality difficult, if not impossible. The only apparent quality criteria is the availability of a line to dial into.

Furthermore, even if customers could determine the source of delays, they have no way of avoiding the majority of them. Beyond the choice of an ISP, customers have no way to specify how, or by whom, their messages are forwarded.

ISPs and NSPs have the potential, through capital investments in increased capacity, to improve their service and reduce delay. The issue is whether they have the incentive to do so. If actual and prospective customers are unable to distinguish higher ISP or NSP quality, and if such quality is expensive, it is unlikely that service providers will offer higher quality service.

In the following we will economically analyze the current situation to demonstrate mathematically these points. We can consider the difference between when network delay is beyond our control (exogenous) and within our control (endogenous). We can also look at how a greater portion of network delay can be endogenized, and the effects that this may have.

1. Economic Analysis with Exogenous Network Delay

We would like to determine what kind of incentives the anonymously cooperative structure of the internet creates for service providers. This structure introduces delays that are beyond the control of any individual service provider (exogenous). Our analysis will involve the mathematical modeling of users trying to maximize their utility within their budget constraints, while producers are simultaneously maximizing their profit. We will then be able to arrive at a level of usage and capital investment that all parties will find optimal, within the constraints. The underlying assumptions are that the level of capital investment by the service provider dictates the local delay, and that increased delay diminishes the users' utility.

In the following, we will use a framework (similar to that of Mackie-Mason & Varian) that assumes there are many service providers of various levels of quality and capacity, and that the prices that these providers are able to charge are based on their quality of service as embodied by their local delay. We will begin by introducing network delay functions from queuing theory, and choosing appropriate assumptions to allow us to determine the correct formula.

First, we will assume that the probability distribution function of inter-message arrival times is exponential (e.g. a chi-squared distribution) and independent ("memoryless"). For a large user base this is a reasonable assumption. Even if we allow for the possibility of discouraged users by say decreasing the rate of arrivals based on queue size, we can still derive a Poisson distribution⁶⁷. In any case, we can reasonably assume a Markov process where the probability of advancing to the next state (one more or less packet in the queue) is either independent of the current state (Poisson) or a function of only the current state (and not of the history) and is exponential. Furthermore, we can even assume this function on an internal leg of a route since the arrival rate is based on message size.

We will also assume that the message size is exponentially distributed, again possibly chi-squared. Since packet size is bounded and large messages are disassembled into packets, packet

⁶⁷ Kleinrock, Leonard, Queueing Systems, vol. 1, pp.99-100, (1975), John Wiley & Sons, New York

size is also exponentially distributed (with a beta distribution)⁶⁸. Service time is directly related to the packet size, thus the service time is Markov process and is exponentially distributed.

So we can use the simple M/M/1 formula (Markov process for arrivals, Markov process for service times, and 1 server) to determine average delay. This is one of the models that has been used for telephone systems since early in the century⁶⁹.

$$\text{The average waiting time for an incoming packet is } T = \frac{1}{\mu - \lambda}$$

Where μ is the mean service time and λ is the mean arrival rate.

In our case; capacity is K, average packet size is \bar{x} with a Markov distribution, and the arrival rate is X. Note that the arrival rate X should be in similar units as the capacity, say bits per second, so to get the average packets per second we divide by the average packet size.

$$\text{The local delay function is then } D_L = \frac{1}{\frac{K}{\bar{x}} - \frac{X}{\bar{x}}} = \frac{\bar{x}}{K - X} \quad (\text{equation 1.0})$$

This delay is additive at each node (router and respondent). We will denote network delay exogenous to the local provider as D_N . So the total delay for a packet is then

$$D_T = D_L + D_N.$$

We will use this delay function in the user's utility function. The user's utility is obviously diminished by delay. The Mackie Mason & Varian assumption is that the loss of utility is additive and unrelated to the original utility. We will introduce a utility function in which the loss of utility due to delay is multiplicative and related to the original utility. We would like our utility function, with respect

⁶⁸ It should be noted, however, that one could argue for an unspecified General function of service times: a M/G/1 model. The average wait time for this model is $\frac{\mu\lambda\sigma^2 / 2}{(\mu - \lambda)}$ ⁶⁸, incorporating the variance. Given the common denominator term, the analysis and results are thus similar.)

⁶⁹ *ibid*, vol. 2, p. 11

to delay, to be both decreasing and convex. The convexity is because the loss of utility in, say, the first minute of waiting is greater than the loss in say the 20th minute of waiting.

The utility function that we will use is

$$u(x, D_T) = \frac{v(x)}{D_T}$$

where $v(x)$ is the utility of sending or receiving x bits of data with no delay. We divide directly by the total delay without any additional variable or exogenous parameters, as these can be thought of as implicit in the utility function $v(x)$.

To see the implications we want to solve the user's and the supplier's maximization problems. We continue with a model in which there are assumed to be many providers of various quality. The quality in this case is embodied by the delay time, and the prices that the providers can charge are functions of the delay time. The firms are price takers in this perfectly competitive model. It should be noted that it doesn't matter whether there is usage pricing or not. We will consider the more general case in which there is usage pricing, and we will see all the price factors drop out.

The user's maximization problem can be reduced to⁷⁰

$$u(x, D_T) - q(D_T) - x \cdot p(D_T)$$

Where $q(D_T)$ represents the connection charges and $p(D_T)$ represents the usage charges. Note that both charges are based on the quality of service as embodied in the total delay, and that usage charges are per unit (bit), so are multiplied by the message quantity.

⁷⁰ This format stems from the Lagrange-Multiplier method of constrained maximization: the full format is

$u(x, D_T) - \lambda(B - q(D_T) - x \cdot p(D_T))$. The budget constraint B drops out upon differentiation with respect to delay, and the multiplier λ becomes a constant incorporated into the utility function.

The result of user maximization with respect to D_T is

$$\frac{du(x, D_T)}{dD_T} = q'(D_T) + x \cdot p'(D_T)$$

The supplier's maximization problem is

$$n \cdot q(D_T) + X \cdot p(D_T) - c(K)$$

Where n represents the number of users, X represents the quantity of messages sent by all users ($X=nx$), and $c(K)$ represents the cost of K capital (capacity).

The result of supplier maximization with respect to K is

$$n \cdot q'(D_T) \frac{dD_T}{dK} + X \cdot p'(D_T) \frac{dD_T}{dK} = c'(K)$$

or

$$n \left[q'(D_T) + x \cdot p'(D_T) \right] \frac{dD_T}{dK} = c'(K)$$

Here we substitute the results of the user's 1st order maximization conditions and we see that the price terms drop out, regardless of whether there is usage pricing or not.

$$n \left[\frac{du(x, D_T)}{dD_T} \right] \frac{dD_T}{dK} = c'(K)$$

And we are left with

$$n \frac{du(x, D_T(K))}{dK} = c'(K) \quad (1.1)$$

a) Linear Cost of Capacity

Now we will substitute in our specific utility function above, and we will assume that the supplier's cost function is linear. This is a valid assumption in some cases, such as in the short run, at the high-end of technology, and whenever capacity is added incrementally, for example an ISP adding additional T1 connections. (We will consider the case of economies of scale in a later example).

So we have

$$n \frac{v(x)}{D_T^2} \cdot \frac{\bar{x}}{(K-X)^2} = c'(K) = c \quad (1.2)$$

or

$$\frac{v(x) \cdot X}{c} = (K-X)^2 \cdot D_T^2 \quad (1.25)$$

so

$$\sqrt{\frac{v(x) \cdot X}{c}} = (K-X) \cdot \left[\frac{\bar{x}}{(K-X)} + D_N \right]$$

then

$$(K-X) \cdot D_N = \sqrt{\frac{v(x) \cdot X}{c}} - \bar{x} \quad (1.3)$$

finally

$$K = X + \frac{\sqrt{\frac{v(x) \cdot X}{c}} - x}{D_N}$$

Which gives us the reasonable results of

$$\frac{dK}{dX} > 0, \frac{dK}{dv(x)} > 0$$

Meaning that as the usage increases, or the utility of the usage increases, the supplier would add capacity. (He would do this of course because he could charge more.)

We also see that

$$\frac{dK}{dc} < 0$$

Meaning that if capacity is more expensive, we add less of it (or if it's cheaper, we add more of it).

Most importantly for our analysis, though, we see

$$\frac{dK}{dD_N} < 0$$

This means that as the network delay increases, the local provider will have no incentive to increase his capacity and in fact will have an incentive to decrease it, *cet. par.*

b) Economies of Scale in the Cost of Capacity

We can also consider the case where the cost of capacity is not linear but rather exhibits economies of scale, for example an ISP going from multiple T1s to a T3 connection. We will use $c = a \cdot \ln(K) + b$ (with a and b as arbitrary constants) as a representative cost-of-capacity function with a positive first derivative and a negative second derivative. We then have, from equation 1.2;

$$n \frac{v(x)}{D_T^2} \cdot \frac{\bar{x}}{(K - X)^2} = c'(K)$$

from which we can follow the same analysis to equation 1.3;

$$(K - X) \cdot D_N = \sqrt{\frac{v(x) \cdot X}{c'(K)}} - \bar{x} \quad (1.4)$$

at which point it is apparent that solving for K directly in order to gain insight on the sign of dK/dD_N will not be the path of least resistance. By solving for D_N and checking the sign of dD_N/dK (at our particular point of interest) we can obtain our desired insight.

We have

$$D_N = \frac{\sqrt{\frac{v(x) \cdot X}{c'(K)}} - \bar{x}}{K - X}$$

so

$$\frac{dD_N}{dK} = \frac{-\frac{1}{2}(K - X)(\sqrt{v(x) \cdot X}) \left[c'(K)^{-3/2} \right] c''(K) - \sqrt{\frac{v(x) \cdot X}{c'(K)}} + \bar{x}}{(K - X)^2}$$

then substituting $\sqrt{\frac{v(x) \cdot X}{c'(K)}} = \pm(K - X) \cdot D_T$ (from equation 1.25 above) into the first

numerator term, and equation 1.4 above for the remainder of the numerator, we obtain

$$\frac{\partial D_N}{\partial K} = \frac{-\frac{1}{2} \cdot \pm(K - X)^2 \cdot D_T \cdot \left(\frac{c''(K)}{c'(K)}\right) - (K - X) \cdot D_N}{(K - X)^2}$$

Then using our logarithmic cost function of $c = a \cdot \ln(K) + b$, we have

$$\left(\frac{c''(K)}{c'(K)}\right) = -\frac{1}{K}$$

and dividing through by $(K - X)^2$, we have

$$\frac{\partial D_N}{\partial K} = \pm \frac{1}{2} \frac{D_T}{K} - \frac{D_N}{(K - X)}$$

Which is unambiguously negative if the first term is negative.

And if network delay is greater than the local delay, then

$$\frac{D_N}{D_T} > \frac{1}{2} > \frac{1}{2} \frac{(K - X)}{K}$$

so

$$\frac{D_N}{(K - X)} > \frac{1}{2} \frac{D_T}{K}$$

and

$$\frac{\partial D_N}{\partial K} < 0$$

and we still infer that $\frac{\partial K}{\partial D_N} < 0$ (ceteris paribus, over our area of interest) even with economies of

scale in our cost function.

2. Endogenizing Network Delay; Economic Analysis

We can continue with the preceding analysis to determine the effects of bringing more of the exogenous network delay under local control. It's obvious that in the case of a single owner network there are no exogenous network delays ($D_N=0$), and so no potentially negative incentives. What is also clear are the effects of decreasing the amount of exogenous network delay in relation to the local delay.

From the preceding analysis, we had

$$\frac{\partial K}{\partial D_N} < 0$$

(whether the cost of capacity was linear or exhibited economies of scale)

Which means that as the network delay increases, the amount of capital that a local provider will invest will decrease. Conversely, as the exogenous network delay decreases, the optimal amount of capital that a local provider will invest in capacity will increase. So if can find a way to decrease exogenous network delay, we will give incentives to service providers to reduce their delays as well.

3. Endogenizing Network Delay; Internet Literature/RFCs

One method of reducing anonymous delay is to make fewer of the forwarding entities anonymous. If the choice of an NSP were available, much as the choice of a long-distance telephone company is, then a significant portion of the potential for anonymous delay would be eliminated.

There have been a number of IETF RFCs regarding policy routing^{71 72 73}, source demand routing⁷⁴ and IP within IP^{75 76 77}. All of these touch on the issue of route specification by the user.

The early RFCs regarding policy routing began recognizing that the internet was turning into a hybrid mesh and tree structure, rather than strictly a spanning tree, and that this presented theretofore unanticipated routing flexibility and decisions. But due to the still relatively small size, and single NSFNET backbone structure at the time, the routing policy suggested was in large part the specification of a regional network (which was connected to the NSFNET) as the designated primary forwarding network for messages to a given connected user/network⁷⁸. Another early RFC by David Clark⁷⁹ is notable for its description of routing as a marketplace with numerous potential vendors, its analogy to telephone systems and long-distance carrier selection, and its brief discussion of billing and

⁷¹ IETF Network Working Group, J. Rekhter, RFC 1092: "EGP and Policy Based Routing in the New NSRNET Backbone", <<http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1092.bt>>, February 1989

⁷² IETF Network Working Group, D. Clark, RFC 1102: "Policy Routing in Internet Protocols", <<http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1102.bt>>, May 1989

⁷³ IETF Network Working Group, H-W. Braun, RFC 1104: "Models of Policy Based Routing", <<http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1104.bt>>, June 1989

⁷⁴ IETF Network Working Group, D. Estrin, T. Li, Y. Rekhter, K. Varadhan, D. Zappala, RFC 1940: "Source Demand Routing: Packet Format and Forwarding Specification (Version 1)", <<ftp://ds.internic.net/rfc/rfc1940.bt>>, May 1996

⁷⁵ IETF Network Working Group, S. Hanks, T. Li, D. Farinacci, P. Traina, RFC1702: "Generic Routing Encapsulation over IPv4 Networks", October 1994

⁷⁶ IETF Network Working Group, C. Perkins, RFC 2003: "IP Encapsulation within IP", <<http://ds.internic.net/rfc/rfc2003.bt>>, October 1996

⁷⁷ IETF Network Working Group, C. Perkins, RFC 2004: "Minimal Encapsulation within IP", <<http://ds.internic.net/rfc/rfc2004.bt>>, October 1996

⁷⁸ IETF Network Working Group, J. Rekhter, RFC 1092 (1989)

⁷⁹ IETF Network Working Group, D. Clark, RFC 1102 (1989)

collection (at the time there was no billing or nominal charges as all traffic went over the publicly funded NSFNET backbone). There is also mention of a user class identifier, to distinguish among customers within a given source-destination route. There is also considerable detail regarding routing table specification (by source host, source domain, source forwarding domain, destination host, destination domain, destination forwarding domain, user class, and conditions) which while arguably feasible when the internet was much smaller, is clearly unsuitable for current use due to the size explosion of the internet. The proposal also called for a number of modifications to IP and additional servers to be implemented. The complexity and overhead required doomed the proposal, in spite of its insightful discussion.

The encapsulation of IP within IP has also been mentioned in the literature as a means for user-specification of routing. RFC1702⁸⁰ describes a protocol type to use within IPv4 to specify either a list of IP addresses or of Autonomous Systems (AS) to be used as forwarding nodes. Source routing options within IPv4 allowed for strict source routing: a sequence of adjacent IP addresses, or loose source routing: a sequence of IP addresses that were not necessarily adjacent⁸¹ (both required quite a bit of IP header modification in route). Such as sequence of IP addresses poses the problem of message failure if any of the specified nodes is temporarily out of service. The specification of an AS instead of an absolute IP address allows for the handling of transient problems within an AS, and allows for increased efficiency and improvements within an AS, transparent to users. The overhead of encapsulation, however, is that each specified node (AS or address) must process the encapsulating header with additional checks and must also update the pointer in the header to point to the next specified AS/address. The destination node must also strip the encapsulating header.

⁸⁰ IETF Network Working Group, S. Hanks, T. Li, D. Farinacci, P. Traina, RFC1702 (1994)

⁸¹ Postel, J. "DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION", RFC 791 <<http://ds.internic.net/rfc/rfc791.txt>>, September 1981

A recent RFC describing IP within IP⁸² discusses its use as a "tunnel" for mobile IP purposes or for security purposes, as well as source demand routing. The advantages over loose source demand routing (SDR) within IP include mainly the downsides of the latter; security problems with IP SDR, lack of uniform implementation of IP SDR across the internet, and the IP SDR breach of etiquette involved with modifying an IP header en route. A limitation on IP within IP, in terms of its use as an alternative for SDR is that only one source IP address is specified per header. So in order to specify anything more than a single intermediate node, additional header "wrappers" would be required, adding considerably to the overhead. Also, the decapsulating address is an absolute address, rather than an AS number. Again this increases the potential for non-delivery due to transient failures. Furthermore, the use of a single IP address would limit the scalability of deployment in the case where an NSP wanted to offer forwarding services to end customers. For each customer, it would have to be predetermined which IP address of the NSP was "closest", and the customer would have to be configured with that address.

A recent RFC⁸³ describes another encapsulation method within IP, referred to as Source Demand Routing Protocol (SDRP). SDRP defines an identifying IP protocol number and the format of the following SDRP header and packet, which encapsulate the original IP packet. The details are similar to methods described above, with a sequence of either IP addresses or AS numbers specified as the route, and with a flag indicating either strict source routing (the sequentially specified addresses/systems must be adjacent), or loose source routing (specified addresses/systems need not be adjacent). The benefit over IP encapsulation is the capability of specifying numerous intermediate routing hops, either IP addresses or AS numbers, rather than the single IP address specified in IP encapsulation. The downside includes security and lack of (uniform) implementation. This particular RFC was designated as informational, rather than standards track, meaning that unless the status is changed, widespread implementation is unlikely. Actually the specification is a very reasonable

⁸² IETF Network Working Group, C. Perkins, RFC 2003: "IP Encapsulation within IP", <<http://ds.internic.net/rfc/rfc2003.txt>>, October 1996

⁸³ IETF Network Working Group, D. Estrin, T.Li, Y. Rekhter, K. Varadhan, D. Zappala, RFC 1940 (1996)

solution to the problem, but its functionality is incorporated into the next version of IP, IPv6, the relevant features of which are described below.

IPv6^{64 85} incorporates the security of IP within IP encapsulation, and the ability to specify multiple (adjacent or non-adjacent) forwarding domains or addresses, all with less overhead. The source routing specification is similar to that of IPv4, but allows for AS/domain specification, and requires little header update en route. IPv6 also has security and authentication measures built in, addressing some of the problems that IP encapsulation addressed. The only question is the timing of the upgrade from IPv4 to IPv6. Considerable work has been done, and is being done, by the IETF's IP next generation Working Group on transition and implementation issues⁸⁶. The transition is expected to last well into the next decade, and is currently underway only in test environments. The logistical issues are daunting, with a huge disparate installed base running IPv4, and with no central authority to mandate whether and when the transition will take place. Only market forces and needs requirements will drive the transition. There are, however, abundant reasons for IPv6 which should turn the tide⁸⁷.

In short, IPv6 offers a relatively efficient and flexible methodology for source nodes to specify all or part of their preferred intermediate routing to any destination. And there are specified methodologies for the operation of IPv6 "islands" in a sea of IPv4⁸⁸. These enable the use of IPv6 by corresponding parties that wish to utilize its features, while not requiring that the entire internet also implement the transition. IPv6, then, answers the technical questions regarding the specification of forwarding entities.

⁶⁴ IETF Network Working Group, S. Deering, RFC 1883 (1995)

⁶⁵ IETF IPNG Working Group, S. Deering, R. Hinden, "Internet Protocol Version 6 (IPv6) Specification". <<ftp://ietf.org/internet-drafts/draft-ietf-ipngwg-ipv6-spec-v2-01.txt>>, November 1997

⁶⁶ For a list of current works in progress, see <<http://www.ietf.org/ids.by.wg/ipngwg.html>>

⁶⁷ IETF Internet Architecture Board, S. King, R. Fax, D. Haskin, W. Ling, T. Meehan, R. Fink, "The Case for IPv6", <<ftp://ietf.org/internet-drafts/draft-ietf-iab-case-for-ipv6-00.txt>>, November 1997

⁶⁸ *ibid*, pp37-38

4. Endogenizing Network Delay; Implementation Issues

Consolidation or affiliation of ISPs with particular NSPs would of course solve this issue. But given the existence of independent ISPs, customers need a way to specify their "long distance" provider.

IPv6 describes the protocol for source specification forwarding entities (the implementation issues for IPv6 are discussed above). In addition, users need a straightforward method of utilizing the IPv6 source routing feature to specify a particular NSP, or "long distance carrier". And the NSPs need a method of authorizing and verifying authorization of such usage.

These could be accomplished by the NSP offering downloadable software which configured the user's IP software to use the loose source route option and to set that NSP as the first (and possibly only) AS in the list. At the time of download, the NSP would gather the necessary information from the customer for billing and authorization purposes.

Authorization checks by the NSP would not be as onerous as might be expected, as only the first BR (border router) of the NSP needs to check authorization, and it needs only to check for locally authorized users. Flow IDs within IPv6 might be a method of specifying particular users, and the downloaded software from the NSPs would set these values in customers' traffic (customer IP addresses would not be suitable IDs, as these are sometimes dynamically allocated by ISPs, and so they may not be consistent for particular customers).

NSPs need only to check traffic that explicitly specifies them as a forwarding "agent", and again, they need only check it once at the point of entry into their routing cloud. Additional traffic may still pass through the NSPs, as it does now, even when source routing is used (as long as the NSP isn't specified in the source route). For example, if NSP "A" is my specified NSP and I'm sending a message to a destination that is hosted on an ISP that in turn is connected to NSP "B"'s backbone, then my message will need to traverse part of NSP "B"'s network as well (after NSP "A" has routed it).

As NSP "B" will not be explicitly specified in my source routing (but nevertheless will need to be used to get the packet to its destination), NSP "B" will not check my packets for authorization (only NSP "A" will upon entry into their network).

This may cause a subtle (and beneficial) difference in the way that packets are forwarded. If I am connected to NSP "A" and the destination of my packets is on NSP "B" and both NSPs connect at multiple points (which is common), then NSP "A" can theoretically transfer my packets to NSP "B" at any of a number of these multiple points. Currently, the incentives are for the NSPs to transfer traffic at the earliest possible common connection point (e.g. the common NAP closest to the source), which they do. However, if NSPs were selling themselves as preferred carriers then they would likely want to control the handling of the traffic for as long as possible, to insure the quality of their product. In that case, NSP "A" would not transfer traffic to NSP "B" until last possible common connection point (e.g. the NAP closest to the destination). This would clearly reduce the amount of anonymous delay even further. And if the NSP of the destination were known (e.g. publicized by the destination), then the anonymous delay would be eliminated. (Although there may still be the question of who caused the delay, the potential sources would be few and identifiable, and practice would indicate which are usually involved during problem periods.)

Moreover, the implementation of such a scheme could be undertaken unilaterally by a large NSP. They need only the internal capability to authorize and route the IPv6 source routed traffic properly, the software for users to download to configure their traffic to use the feature, and the means to gather and process user information at the time of download for billing and authorization purposes. Of course the development and implementation would be somewhat complex, but its clearly doable. Furthermore, the potential gains, both in terms of business for the NSP and in performance benefits for the user, are significant.

5. Summary

The selection of NSPs as long distance carriers provides a number of potential benefits, and is technologically feasible. An important result is that anonymous delay is reduced (or eliminated). This provides the carriers with incentives to increase their capacity. And as was shown in the economic analysis section, a service provider has incentives to adjust his investment in inverse proportion to the exogenous network delay. As the anonymous (exogenous) portion of the delay is reduced, service providers have incentives to increase capacity. And as other service providers upgrade, they have double incentives to do so (competition, and further reduction in network delay).

There are a number of methods of specifying intermediate forwarding agents within the current internet structure, but each has limitations. IPv6 addresses all the concerns and could be the most efficient, flexible, and secure method of source demand routing. IPv6 is currently in the test stage, and the transition is expected to be lengthy. Nevertheless, IPv6 can operate within IPv4, and provides the best means for source demand routing in the near future.

Finally, this capability could be unilaterally implemented by a single large NSP to the benefit of itself and its customers. When and if other providers implement the same functionality, welfare will be increased further.

E. Quality of Service and Prioritization; Benefit Analysis

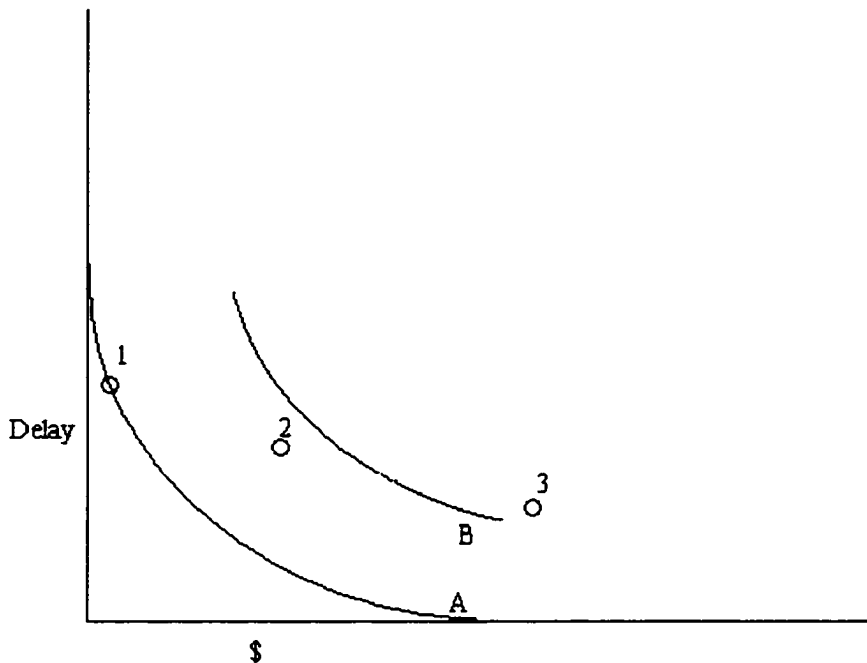
In the following section we will look at the potential benefits from supporting multiple classes of service on the internet. As described previously, at present internet traffic is routed on a first-in first-out basis. This may not serve either the users or the providers well. Many large companies with specific time-sensitive requirements choose to implement costly private networks to obtain the level of service that they need. Not only might they benefit from having this level of service available through the internet (at a greatly reduced price in comparison to building a private network), but the providers would likely also benefit greatly by the addition of large customers willing to pay a premium for higher quality service.

In the following we will demonstrate some of the economics underlying these conclusions. In addition we will devise a comparative metric for performance analysis and analyze some selected examples. We will also review the internet literature that addresses this topic, and discuss technical and practical implementation issues in some detail.

1. Economic Analysis

As was mentioned in the initial economic literature section, a group at the University of Texas has analyzed network computing with priority classes. Although their analysis included usage charges, nevertheless, economic welfare was shown to increase in with multiple priority classes⁸⁹.

To demonstrate the benefits to the user graphically we create the following examples

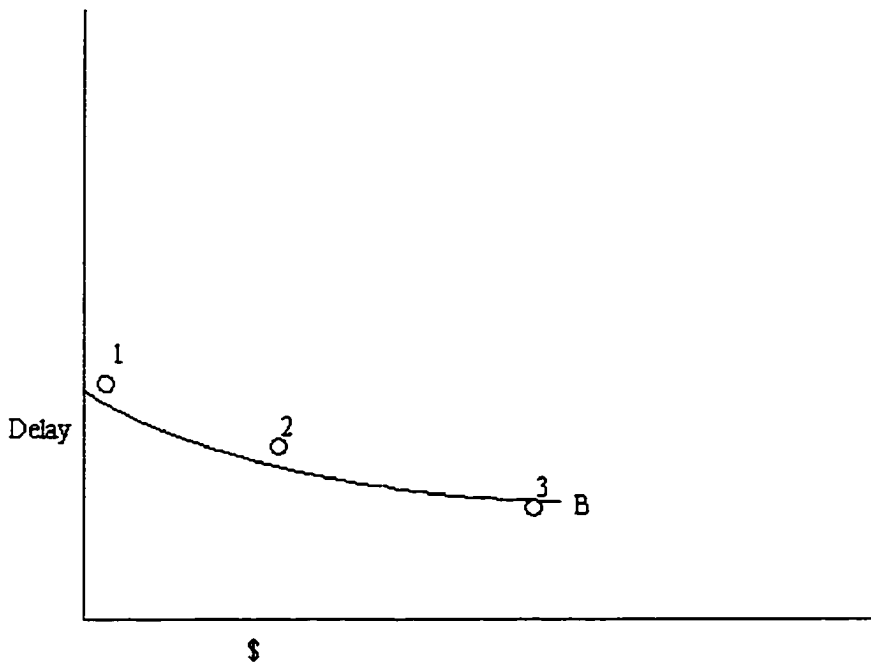


Each curve represents an “iso-util” or “indifference curve”, which is a set of Delay/Cost combinations that are of equal value to the user (he is indifferent to selection between any of the points on a given curve). Those closer to the origin are of greater value, so B may be the set of minimum performance versus price points for which the user is willing to pay, while the points on A are clearly preferable over B (maybe “twice as good”: the measurement of utility is subjective). The numbered points are service levels.

We can consider the middle service level (point 2) to be equivalent to the price/performance offered by the single first-in first-out priority class. It's better than the minimum requirements (curve B) for our representative user, so he would use it if that was all that was offered. However if he were offered the choice of three levels of service, represented by points 1,2,3 then his welfare may be increased. We can see that the highest priority level, level 3, would be unacceptable to him (it's outside of B: even though the delay is less, the price is too much for him), while priority level 2 is the same as the single service level, and is acceptable. However the lowest priority level, level 1, represents a price/performance combination that he likes even better than he is presently receiving. This user is obviously a price sensitive user, so high priority levels don't offer him any perceived benefits. However he's happy to experience more delay if the price is sufficiently less.

We can easily imagine the opposite type of user, who is more delay-sensitive, and is willing to pay more to reduce delay. His minimum acceptable "iso-utility" curve is shown in the next graph.

⁸⁹ Gupta, A., Stahl, D., Whinston, A., "Priority Pricing of Integrated Services Networks", <<http://cism.bus.utexas.edu/>>, 1995



As before, curve B represents the minimum price/performance combinations that this user is willing to purchase. We can see that he would not be using these network services if only the single service class (2) were offered. However, he would be willing to pay the premium for the highest priority level (3) to reduce his delay if this service class were offered.

So regardless of his preferences, a customer can do no worse if he is given choices. In fact he will always be better off unless his preference combinations coincidentally happen to exactly match the all the service levels on a single "iso-util", in which case he would be indifferent to the additional choices.

Producers also gain with the ability to offer additional service classes. The preceding example demonstrated that higher service levels may attract customers that currently aren't using network services (as might the addition of lower service levels). The abilities to price discriminate and attract additional customers can directly impact providers' profitability. Also, like the consumer, the technical

availability of multiple classes of service doesn't require the provider to offer them. If he finds he is better off offering only a single class of service, he can of course still do so. Moreover, if these multiple service classes represent no additional costs to the producer and do not reduce throughput, they can have no adverse effects on producers, and will almost surely provide benefits in the form of greater profits.

2. Performance Analysis of Multiple Priority Queues

We would like to compare multiple priority queue performance with that of the current single FIFO queue system. Recall that the delay function for a single queue is

$$D = \frac{1}{\mu - \lambda}$$

where μ is the service speed and λ is the arrival rate.

For our comparison examples, we want to be independent of physical speeds, so we will develop and use our own metric unit; message units M , to measure delay. M is the time it would take to transmit an average message when that message is the only message in the system. So, recalling that \bar{x} is the average message size (in say bytes per message) and K is the capacity (in say bytes per time unit) we have the time per message M as

$$M = \frac{\bar{x}}{K} \quad (\text{equation 2.0})$$

We want to express delay in these message units, so we have

$$D = \frac{1}{\mu - \lambda} = \frac{\bar{x}}{K - X} = \frac{\bar{x}}{K} \left(\frac{1}{1 - \frac{X}{K}} \right) = \left(\frac{1}{1 - \rho} \right) M \quad (\text{equation 2.1})$$

where ρ is the load factor. We will use this formula below in our comparison.

Before we do the comparison, we have to derive the delay formula for the prioritized queues. We have two components of delay. The first is the time waiting in the queue and the second is the actual service (transmission) time. The queue delay itself has three components:

- The mean residual service time of the message that is in the process of being transmitted when our message arrives. (We assume non-preemptive prioritization, meaning that a message in process will be allowed to complete even if it is of a lower priority.)
- The mean service time of all messages of higher priority that are in the queue when our message arrives.
- The mean service time of all higher priority messages that arrive while our message is waiting to be serviced.

These three components of queue time can be summarized by the following equation ⁹⁰

$$D_q = \frac{W_0}{\left(1 - \sum_{i=y}^h \rho_i\right) \left(1 - \sum_{i=y-1}^h \rho_i\right)}$$

where W_0 is the mean residual service time of a message in-process at any given point in time, y is the priority class whose delay we are determining, and ρ_i is the load factor for priority class i , where h is the highest priority.

⁹⁰ for the derivation, see for example Harrison & Patel, (1992) Performance Modelling of Communication Networks and Computer Architectures, Addison-Wesley, Reading, MA, p.282-285

If we assume an exponential distribution of service times, we can use the following equation⁹¹

$$W_0 = \lambda M^2 \quad (\text{equation 2.2})$$

where λ is the cumulative arrival rate for all priority classes. We use X_T to indicate the total arrival rate expressed in bytes per time period, while \bar{x} is the average message size in bytes per message. So we have λ in messages per time period:

$$\lambda = \frac{X_T}{\bar{x}} \quad (\text{equation 2.3})$$

combining equation 2.3 with equation 2.0 and substituting into 2.2 we have

$$W_0 = \frac{X_T}{\bar{x}} \left(\frac{\bar{x}}{K} \right)^2 = \frac{X_T}{K} \cdot \frac{\bar{x}}{K} = \rho_T M$$

where ρ_T is the total load factor. So we have the queue delay as

$$D_Q = \frac{\rho_T M}{\left(1 - \sum_{i=y}^h \rho_i\right) \left(1 - \sum_{i=y-1}^h \rho_i\right)} = \left[\frac{\sum_{i=1}^h \rho_i}{\left(1 - \sum_{i=y}^h \rho_i\right) \left(1 - \sum_{i=y+1}^h \rho_i\right)} \right] M$$

To this queue delay we have to add the service time delay of sending the message, which is M .

So the mean total delay for priority class y is

$$D_y = \left[1 + \frac{\sum_{i=1}^h \rho_i}{\left(1 - \sum_{i=y}^h \rho_i\right) \left(1 - \sum_{i=y-1}^h \rho_i\right)} \right] M \quad (\text{equation 2.4})$$

⁹¹ see for example Hammond & O'Reilly, (1986) Performance Analysis of Local Computer Networks, Addison-Wesley, Reading MA, p. 101

Now we can compare the priority queue performance (equation 2.4) to the single queue FIFO system (equation 2.1), expressing both in message units M . We will do two examples, using realistic peak load levels in the 75 to 80% range⁹². We will also implicitly assume a straight-line (or other smooth) demand function, in which case equal volume priority classes would maximize revenue.

The first example will compare a 4-priority system with equal load distribution among the priority classes, to a single queue system with the same load. We use a total load factor of 80%, so

$$\rho = 0.8, \rho_1 = 0.2, \rho_2 = 0.2, \rho_3 = 0.2, \rho_4 = 0.2$$

Substituting in, we see that in the single queue system each message waits an average of

$$D = \left(\frac{1}{1 - 0.8} \right) M = 5M$$

In the multiple priority class system, we have the following average delay times

$$D_4 = \left(1 + \frac{0.8}{(1)(0.8)} \right) M = 2M$$

$$D_3 = \left(1 + \frac{0.8}{(0.8)(0.6)} \right) M = 2\frac{2}{3}M$$

$$D_2 = \left(1 + \frac{0.8}{(0.6)(0.4)} \right) M = 4\frac{1}{3}M$$

$$D_1 = \left(1 + \frac{0.8}{(0.4)(0.2)} \right) M = 11M$$

Notice that the total waiting time for four messages, one from each priority class, would be $20M$. This is, of course, the same total waiting time that four messages in the single queue system would have, with $5M$ each. This has to be true since the overall arrival rate and transmission speed are the same.

Also note that these comparisons are valid over a multiple node network as the delays are additive, so the factors remain the same. (The only difference would be that in an N node network, M would be replaced by NM .)

⁹² see for example <<http://www.caida.org/INFO/>> for links to NAP and NSP statistics

Our second example is a 3-priority system, again with equal load distribution among the priority classes. This time we use a 75% total load.

For the single queue system we have

$$D = \left(\frac{1}{1 - 0.75} \right) M = 4M$$

For the priority queue system we have

$$D_3 = \left(1 + \frac{0.75}{(1)(0.75)} \right) M = 2M$$

$$D_2 = \left(1 + \frac{0.75}{(0.75)(0.5)} \right) M = 3M$$

$$D_1 = \left(1 + \frac{0.75}{(0.5)(0.25)} \right) M = 7M$$

(again verify that total delay is unchanged: $3 \cdot 4M = 2M + 3M + 7M$)

These examples demonstrate the very attractive nature of the priority queue system. In our examples, only messages in the lowest priority class experience increased delays over the single queue system. All other classes see improvements, most of them substantial. Even the performance degradation in the lowest priority class is not particularly onerous, averaging just double the unprioritized delay over the two examples. And many applications, such as email, are not particularly sensitive to this level of delay.

So we see a tremendous opportunity with prioritized queuing. We can have greatly increased performance for applications that need it and users that are willing to pay for it, without imposing too much of a burden on less time-sensitive applications or users.

3. Internet Literature/RFCs

Multiple classes of service on the internet have long been discussed. An early RFC on policy routing⁹³ mentions multiple prioritized queues as a potential means of efficient dynamic allocation of resources. Also, experimental protocols have been used from early on to accommodate real-time streams⁹⁴. These worked at the IP layer alongside IP, reserving resources for their traffic flows. Work has continued until very recently on these protocols^{95 96}, but they have remained experimental, and have largely been passed by in favor of other protocol methodologies that work within IP, as IP has become the single de facto standard (and as these require the additional overhead and complexity of deployment alongside IP). (As a point of interest in trivia, ST protocol uses IP version number 5, and is the reason that IP is going directly from IPv4 to IPv6.)

As alluded to previously, the priority field in IPv4 was used to give interactive traffic priority in the late 1980s as bandwidth lagged behind demand and delays increased. This issue has resurfaced with the larger demand increase in the 1990s, including real-time audio and video applications. The currently popular approach to the problem is described as "Integrated Services", which means a concurrent combination of dedicated resource reservation and best effort forwarding, and dynamic allocation of resources between these two service classes, all within (or on top of) IP. One of the assumptions that was made to support this model is that simple prioritized service will not always be adequate for those applications that need a defined quality of service⁹⁷. While this may be true tautologically (if the service is not guaranteed, then the service is not guaranteed), the coordination,

⁹³ IETF Network Working Group, H-W. Braun, RFC 1104 (1989)

⁹⁴ Forgie, J., "ST - A Proposed Internet Stream Protocol", IEN119, MIT Lincoln Lab, September 1979

⁹⁵ IETF Network Working Group, D. Topolcic, "Experimental Internet Stream Protocol, Version 2 (ST-II)", <<http://ds.internic.net/rfc/rfc1190.bt>>, October 1990

⁹⁶ IETF Network Working Group, L. Delgrossi, L. Berger, "Internet Stream Protocol Version 2 (ST2)", <<http://ds.internic.net/rfc/rfc1819.bt>>, August 1995

⁹⁷ IETF Network Working Group, R. Braden, D. Clark, S. Shenker, "Integrated Services in the Internet Architecture: an Overview", <<http://ds.internic.net/rfc/rfc1633.bt>>, June 1994

overhead, and complexity required are likely to limit implementation (or at least implementation speed). Nevertheless, the IETF is moving forward on standards for Integrated Services and Resource Reservation Protocol.

The Integrated Services model describes two types of "reserved" services; Controlled Load⁹⁸ and Guaranteed Quality⁹⁹. The specifications do not comprise a protocol, but rather a set of parameters that a protocol should use to invoke these types of services. The Controlled Load Service is closer to prioritized traffic in that the nodes along the route are asked to agree to forward a specified traffic flow (within parameters) as if they were not otherwise loaded. This makes no explicit guarantee of service quality or delay times, but these can be inferred within bounds. On the other hand, the Guaranteed Quality of Service incorporates the additional overhead of calculating and agreeing to actual delay maximums, within bounds. Both these services of course require agreement from and setup at each link in a route before they can be used. As such, they depart from the IP datagram mode and incur significant additional overhead. While the Controlled Load Service is more flexible, it does so at cost; either the resources are reserved or the node must have statistical reason to believe that it can oversubscribe and/or temporarily under allocate. Some of the flow management complexity traditionally found in TCP is then being shifted down to the IP level. Both of these service specifications represent proposed standards that are currently under review.

While the Integrated Service protocols describe performance characteristics and the parameters that should be used to specify them, they do not describe a specific format for these parameters, or a specific methodology to use them. Resource Reservation Protocol (RSVP) does just that. In fact

⁹⁸ IETF Network Working Group, J. Wroclawski, "Specification of the Controlled-Load Network Element Service", <<http://ds.internic.net/rfc/rfc2211.txt>>, September 1997

⁹⁹ IETF Network Working Group, S. Shenker, C. Partridge, R. Guerin, "Specification of Guaranteed Quality of Service", <<http://ds.internic.net/rfc/rfc2212.txt>>, September 1997

RSVP may invoke Controlled Load services, Guaranteed QoS, or other yet to be specified service types¹⁰⁰.

RSVP¹⁰¹ has been getting a lot of attention, but carries a lot of baggage. Although claimed to be at the transport layer level¹⁰² due to the fact that it operates on top of IP, it must be invoked and operational at each participating router along a path, unlike a true transport layer protocol. It actually works below the transport layer, interfacing with TCP or UDP. It is comprised of a number of modules including a packet classifier and a packet scheduler. The former must check and classify all incoming packets, and the latter must work with (or within) the link layer to provide the agreed upon QoS (obviously requiring significant ad hoc integration with each link layer version specifics). Other modules invoked during setup include policy control and admission control. The former determines if a given user has authority to use the resources that he is requesting, and the latter determines if these resources are available. The admission control parameters include those to describe services such as guaranteed QoS. The policy parameters are still under development.

RSVP uses transport layer port numbers for packet classification, raising additional complications: transport layer headers may not have a fixed offset from the IP header, and in fact may be encrypted. Extensions and use of the IPv6 flow id have been suggested as remedies (but of course they are not costless).

Other problems with RSVP include security vulnerability unless keys are deployed across the system, and more importantly, the fact that it does not scale well (as one might guess by its required overhead)¹⁰³. An example is a high bandwidth backbone which could easily be swamped by requests

¹⁰⁰ IETF Network Working Group, J. Wroclawski, "The Use of RSVP with IETF Integrated Services", <<http://ds.internic.net/rfc/rfc2210.txt>>, September 1997

¹⁰¹ IETF Network Working Group, R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource Reservation Protocol (RSVP) – Version 1 Functional Specifications", <<http://ds.internic.net/rfc/rfc2205.txt>>, September 1997

¹⁰² *ibid*, pg. 3

¹⁰³ IETF Network Working Group, A. Mankin, F. Baker, B. Braden, S. Bradner, M. O'Dell, A. Romanow, A. Weinrib, L. Zhang, "Resource Reservation Protocol (RSVP), Version 1 Applicability Statement, Some Guidelines on Deployment", <<http://ds.internic.net/rfc/rfc2208.txt>>, September 1997

for relatively small amounts of its bandwidth, when in fact it might easily be able to accommodate all those flows (just not the overhead in keeping track of them). There may also be problems with the required link level integration on certain routing and interface products, particularly those high speed products with traditional software functions incorporated into hardware.

And there are still more unresolved questions concerning variable routing based on requested QoS (RSVP is not a routing protocol), and on the definition of policy objects (not to mention the accounting and billing for them), and on other issues.

IPv6¹⁰⁴, as previously described, is the new version of IP that is expected to supercede the currently deployed IPv4. IPv6 contains a Traffic Class field that corresponds roughly to the IPv4 type of service and precedence fields. In the current version of the IPv6 spec, the Traffic Class field is 8 bits, and is unspecified. Interestingly, in the previous version of the IPv6 spec, the field was referred to as "Priority" and was 4 bits with predefined recommended values based on traffic type (news=1, mail=2, ftp=4, interactive=6, non-TCP=8 through 15)¹⁰⁵. It is important that the IPv6 Traffic Class field have an explicit priority component (as in IPv4) which is not tied to traffic type. Grouping traffic by type may be worthwhile, but grouping by priority (i.e. willingness to pay) could be of greater immediate benefit.

The IPv6 header also contains a 20 bit Flow Label field which could be used to identify customers and their priority access privileges in our multiple priority scheme. Like the source demand routing verification, this need only be done at the point of entry to an explicitly specified domain. Other routers within the domain need not do a verification check or a resource availability check. They merely queue the packet according to its priority. Other border routers downstream or those of NSP peers would be expected to provide reciprocal service in forwarding packets according to their priorities.

¹⁰⁴ IETF IPNG Working Group, S. Deering, R. Hinden, (1997)

¹⁰⁵ IETF Network Working Group, S. Deering, RFC 1883: "Internet Protocol, Version 6 (IPv6) Specification", <<http://ds.internic.net/rfc/rfc1883.txt>>, December 1995

4. Implementation Issues

a) Service Provider Cooperation

A single service provider could not implement prioritized message service within the current structure of the internet. Cooperation is required for it to be effective. However if NSPs can be specified like long-distance telephone companies are today (as described in preceding sections), then the consolidation that is likely will make the task of cooperation less difficult. Moreover, the cooperation required on a technical level is less than that required for RSVP.

When a service provider gives prioritized service to traffic from another service provider (or customer), he will want to be sure to be compensated for it. Conversely, when a service provider (or customer) pays for prioritized service, he will want to be sure that is what he receives. Measurement and verifiability are both important issues.

The simplest and crudest form of measurement is the one that is in use today; users and downstream service providers are charged on "pipe-size", or the bandwidth of their connections. This is essentially a single-level-of-service usage-limiting hardware solution. A similar measure could be implemented through software for multiple priority levels of service. Each incoming physical (or logical) connection would have a maximum queue size for each high-priority level. When that is met, an additional incoming priority traffic on that connection is queued at a lower priority. The IP overhead is minimal and is discussed below.

Fairly detailed coordination is required for this scheme to work well. A given priority must mean the same thing across service providers. For example, an ISP selling high priority service to its customers must have that priority recognized and handled appropriately by all upstream providers. For this reason, standardization of priority levels must begin with the large NSPs. As there are only a handful of these, the coordination can be done on an individual basis. This issue raises numerous

other interesting questions of coordinating dissimilar pricing/priority plans, which are beyond the scope of this research. However an agreed-on small number of defined service levels would be straightforward to implement and coordinate.

As for verifiability, the issue would be ameliorated by the explicit selection of NSPs and ISP/NSP affiliation discussed in preceding sections. From a customer's viewpoint, inadequate levels of service could be remedied by switching ISP/NSPs. From an ISP's point of view, inadequate service could be remedied by switching affiliation to a different NSP. For inter-NSP coordination, each is sophisticated enough to determine the level of service provided by the others, and there are forums for discussing it. And in an arrangement where NSPs are explicitly selected, potential negative publicity would be a powerful force.

b) IP

At the lowest level, IP must forward messages according to priorities. IP implementations typically keep packets in a sorted queue. Including priority as the primary sort criterion would be a simple modification, and one which added little or no additional overhead. In fact, a number of IP implementations have parameterized the sort criteria that are applied to the queue. As previously mentioned, the priority field in the IP header was used for this purpose in the late 1980s when bandwidth lagged well behind demand. At that time priorities were assigned by application type (e.g. interactive terminal sessions received higher priority than email). This was done without publicity to minimize unscrupulous manipulation of the priority field.

An example of an IP implementation (with available source code) is Dave Mischler's "IPRoute", a PC-based Router¹⁰⁶. In this package there is a parameter that determines the queuing method. The possible values are "fifo", "priority", and "fair". "Fifo", of course, queues packets on a first-in-first-out basis, which is the norm today. "Priority" uses the precedence and type of service fields in the IP header to determine queuing order, with the former being the primary criterion. "Fair" is similar to "Priority" in that messages are queued and sent in priority order. The difference is that a packet's

¹⁰⁶ Mischler, David, (©1995,1996), "IPRoute", V0.9, <<http://www.mischler.com/iproute/>>, 2/14/97

internally stored priority is incremented when a higher priority packet is inserted in the queue ahead of it. The results, according to the author of the code, is to provide better response time than FIFO queuing, but fairer service distribution than strict priority queuing.

Prioritized queuing is also used in TCP/IP implementations such as the KA9Q package from Phil Kam¹⁰⁷. In this package, the routine that enqueues the packets (`q_pkt`), automatically uses the IP priority/tos field as the primary criterion to sort outgoing packets. It also uses the TCP port number to try to determine whether the packet is part of an interactive session, and if so, to give it a higher secondary sort priority. The routine contains the following comments:

```
/* Add an IP datagram to an interface output queue, sorting first by
 * the precedence field in the IP header, and secondarily by an
 * "interactive" flag set by peeking at the transport layer to see
 * if the packet belongs to what appears to be an interactive session.
 * A layer violation, yes, but a useful one...
 */108
```

The overhead to use this queuing method is not onerous. The queues are kept as sorted linked lists, so insertion is a simple matter of updating the "next" entry pointers. The only overhead is checking the TCP port, and bouncing through the linked list to find the proper point for insertion.

The overhead of implementing bandwidth limits for multiple priorities is also minimal. It would require only a few more operations in the same IP loop described above: When looping through the linked list queue of outgoing messages, in addition to checking the priority, the incoming-connection field of each queued message is also checked. If the incoming connection and the priority of the queued message are the same as those of the message to be queued, then the size of the queued message is added to a total. When this total reaches or exceeds the maximum set for the given incoming line and priority, then the priority of the message to be queued is decremented (and the

¹⁰⁷ Kam, Phil, (©1992) KA9Q package, <<http://www.qualcomm.com/people/pkarn/tcpip.html>>

¹⁰⁸ *ibid*, `lproute.c` file

running total is zeroed). The loop through the linked list then continues as is with the priority check. An example of what the additional code might look like (in C) is given in the following.

```
if (inmsg.priority)
{if (inmsg.lineid=qmsg->lineid and inmsg.priority=qmsg->priority)
    {qtot+=qmsg->msgsize;
if (qtot>=ConnPriMax)
    {inmsg.priority--; qtot=0}
    }
}
/*continue with checking priority to find proper insertion point in list*/
```

The variable "ConnPriMax" is assumed to have been previously set to the maximum queue size for the given incoming connection and priority (it could also be a dimensioned variable, with priority and/or connection number as indices). The "inmsg" structure is the packet to be queued, and the qmsg structure is the packet in the queue that is being examined. The "qtot" variable is zeroed at the beginning of the loop and is used to total the lengths of the packets waiting in the queue with the same incoming connection and priority as the packet to be queued.

As can be seen, the additional overhead is minimal. As for processing overhead, only a few extra statements are required in the queuing loop. As for storage overhead, a table of incoming connections and (non-zero) priorities and their associated maximum queue sizes would need to be kept. As the number of priorities would be small, and the number of connections limited by hardware, this storage overhead should not be significant.

c) Marketing, and the Number of Priorities

The question of the number of priorities offered will be a marketing issue, rather than a technical issue. In any case, it's likely that only a small number of different priorities would be needed, and so only a few bits of data, so there are no additional technical details that need to be addressed. However the marketing issue would most definitely need to be addressed. Providers would want to do market research and analysis experimenting with various product plans. Premium service could be bundled with access to proprietary content and possibly other services. Providers may want to

offer just a few well defined, brand-named priority plans to create brand awareness. So while more priorities may be technically available, only a few might be used by a given provider.

Creating brand distinction serves a number of purposes. First, differentiating the product moves the firm from zero-profit perfect competition to profitable monopolistic competition with prices above marginal costs. Second, effective marketing of well-positioned bundled products can increase demand. Third, brand awareness creates opportunities for the introduction of related products and services under the brand aegis.

In addition, separately priced qualities of service allow the firm to effectively price discriminate and thereby capture additional surplus in the form of increased profits.

5. Summary

Prioritized queuing can provide both users and service providers with significant benefits while requiring little in the form of additional overhead. One of the reasons that this has approach has been neglected in favor of services such as RSVP, is the assumption that prioritized best effort will not necessarily be adequate for some applications at times of congestion. Strictly speaking this may be true. But providers could adjust prices to minimize these periods for the highest priority level(s), and users can clearly see when their priority level is inadequate for their requirements. They would then have the choice to pay for a higher level of service, to try their application later in the hopes of finding less congestion, or to put up with the intermittent delays. The users' willingness to pay should make the decision, as it does in most all economic welfare maximizing schemes.

As mentioned, RSVP is widely considered as the coming solution to congestion problems. However it has many (possibly insurmountable) problems including complexity and overhead (as described in the internet RFC section above). Also, even given its eventual (and unlikely) successful deployment, it is suited to relatively consistent data flows such as real-time audio or video. It's inherent setup overhead does not lend itself efficiently to connections of short duration. It will not solve the problem of minimizing delay for time-critical customers with bursty traffic, who are willing to pay for the reduction in delay. This problem is especially significant as most current internet users have exactly that; bursty traffic as they search for data, download the occasional file, and communicate with others.

Prioritized queuing is an alternative that should be given close scrutiny by the IETF, and should be considered unilaterally by large NSPs.

F. Usage Pricing; Benefits Analysis

Free market pricing is the mechanism that is typically used to efficiently allocate resources (except in special cases like monopolies). Supply and Demand are balanced through the market price, and changes in either are reflected through price changes. Scarce resources, those of limited availability, are usually best allocated through some form of an "auction" system. Such a pseudo-auction system has been proposed by economists for the internet as well (e.g. Mackie-Mason & Varian's smart market system described in the economics of congestible resources section above) .

However, unit pricing brings a unit overhead. When the unit of measure is sufficiently small that the overhead in bidding and pricing individual units outweighs the benefits, some form of aggregation and/or approximation must be done. Clearly, the transparent decomposition of messages into packets forwarded by many nodes introduces an atomic unit of measure whose cost is likely less than the cost of measurement and billing.

In the following we will look at how internet services should be priced in theory and how this might be done in practice. We will also discuss whether there is sufficient benefit to warrant usage pricing, given the overhead and cooperation required to implement it.

1. Economic Analysis

The ideal usage incentive is price. Single priority class usage pricing has been shown to be economically superior to free access¹⁰⁹ (when not considering the overhead of accounting and billing). Virtually all economic schemes for addressing congestion assume that usage pricing will be done.

The ideal price of sending a message should reflect the loss of utility inflicted on those users whose messages are waiting. Mackie-Mason & Varian mathematically describe this optimal price as;

$$p = -\frac{1}{K} \sum_{j=1}^n \frac{du_j(x_j, Y)}{dY}$$

where, as in the previous section:

X = usage

K = capacity

Y = utilization = X/K

n = the number of users

A usage charge is frequently combined with an access charge to form what's known as a two-part tariff (examples include country-club membership and greens fees, amusement park admission and ride charges, and monthly phone access and itemized call charges). Frequently, the access portion of the charges are used for capacity expansion while the usage fees support operations and maintenance (as cited below in the telecommunications and utility peak-load pricing literature).

A two-part tariff that included a monthly access charge might also be used for the internet. The theory that the access part of the charge should be used for capacity is interesting when applied to this medium. The internet represents a new and rapidly expanding technology, and as such capacity expansion costs are likely to dominate. According to the theory then, access charges should

¹⁰⁹ Gupta, A., Stahl, D., Whinston, A. (1994)

dominate. If optimal usage charges were minimal, then the issue of the costs of metering and billing for usage in relation to the amount billed would loom large: it is likely not worth the huge overhead for minimal charges. Possibly this is the optimal structure.

2. Internet Literature/RFCs

There are no internet RFCs that deal explicitly with usage pricing, as pricing is outside of the technical realm of the IETF. The only applicable internet RFCs are those that deal with the issue of metering.

Notable work done on metering by Nevil Brownlee of New Zealand^{110 111}. His package is called NeTraMet. He has modified a freeware version of IP for DOS PCs to log data flows. He has a meter reading package as well. His implementation puts meters at any desired points in the network, and uses the IP headers to measure the cumulative flows between pairs of endpoints. There are numerous options such as settable traffic limits by node by time period, and alert conditions such as queue size.

A similar architecture would be applicable to the prioritized metering problem as well. What is needed is a cost-effective means to log traffic by endpoint by priority. A PC that logs and accumulates data by IP headers could perform this function. It is questionable, however, whether this logger should be an actual IP forwarding node, or merely the recipient of replicated data lines. In the latter case, there would be no additional overhead to the network, only additional dummy nodes that receive duplicates of all messages at meter points and discard the messages after logging the header information.

¹¹⁰ Brownlee, N., IETF Internet Draft, "Traffic Flow Measurement: Experiences with NeTraMet", <<ftp://ietf.org/internet-drafts/draft-ietf-rtfm-acct-experiences-01.txt>>, August 1996

¹¹¹ Handelman, S., Brownlee, N., Ruth, G., IETF Real Time Flow Measurement Working Group, <<ftp://ietf.org/internet-drafts/draft-ietf-rtfm-new-traffic-flow-00.txt>>, November 1996

3. Implementation Issues

Due to the nature of the internet, in which a message is broken into multiple packets which independently traverse many nodes, a typical message may interact with thousands of other messages; both in contributing to their delay and being delayed by others. This makes strict application of the economic pricing principle clearly infeasible. So we are best served by using an approximation.

The first modification to the (theoretically) economically optimal price formula above that we will introduce will use our previously introduced multiple priority queues to aggregate messages. The price formula then becomes

$$p = b \cdot \alpha \sum_{q=1}^k (\beta^{q-1} \cdot s_q)$$

where:

p = price

b = the number of bits or bytes of the given packet

α = a cost factor (specific to the provider)

k = the priority of the packet

β = the priority differentiation factor

s_q = the size of (waiting packets in) priority queue q

And we can also average queue sizes (possibly over time periods), to alleviate the need to dynamically capture them, and still use the same formula.

The actual logging of packets might be done similarly to the method done by NeTraMet, described above in the internet/RFC section.

Once logged information is captured at endpoints, these endpoints can be queried for cumulative data by a central bill processor. To minimize additional overhead, this should likely be done at off-peak times, and only cumulative data collected. Billing would most efficiently be done on-line, based on this cumulative data. Detailed data would be available for audit, but would usually remain distributed at the meter collection points. The online billing and distributed detailed data should serve to keep the cost of billing to a reasonable level. In telecommunications, bill processing is a significant cost factor. Processing bills and payments on-line should greatly reduce the cost.

One of the related issues in metering that could be problematic in this setting is the issue of who pays, sender or receiver. There are a couple of ways to address this question. One is the definition of a bit in the IP header as a "collect" bit. But a considerable support structure would be needed in the higher level protocols to properly set and monitor this bit. A message could authorize a response of a certain maximum size, and these limits could again be settable as preference parameters. But dispute resolution procedures could also prove costly and problematic.

An alternative would be to use port number. Global services such as HTTP and FTP have predefined low port numbers (most common services have port numbers under 100, while all numbers under 1024 are generally reserved for defined global services). When a reserved port number appears as either source or destination, the other port is the one that would be charged. This would handle the cases where users download files from web pages. By requesting the transfer, the user obligates himself to pay for it. This is certainly desirable since charging the owner of the web page would inhibit the amount of information that was made available. In the case when both source and destination port numbers are predefined low port numbers (such as mail) or both are ad hoc high port numbers, sender would pay. This is also the desired scenario for email and other symmetric processes.

4. Comparisons with Telecommunications and Utility Pricing Schemes

If we categorize dynamic load into periods of different demand, we obtain the peak-load pricing problem that has been discussed for some time in the utility pricing literature^{112 113 114}. There have been two basic welfare maximizing pricing models proposed in this literature: One is the two-part tariff model which uses the fixed access cost to cover the fixed capacity costs, while the usage charges cover the operating costs¹¹⁵. The second type is the peak-load pricing model, with has usage pricing only, but prices the off-peak period to cover operating costs only, while peak period charges include fixed capacity costs as well as operating costs¹¹⁶.

As mentioned previously, the newness of internet technology presents additional pricing considerations. High access charges would likely deter some potential new users, slowing revenue growth. On the other hand, usage metering and accounting requires tremendous overhead. Rapidly expanding usage means that capacity expansion costs clearly dominate the costs of operations. So it may be that an ideal two-part tariff would contain only a minimal usage charge corresponding to the relatively minor cost of operations. As such, the overhead costs of usage metering and accounting would likely outweigh the benefits of a minimal usage charge. It may be that usages charges will not be feasible until the internet becomes a more mature industry in which capacity expansion costs do not dominate.

In any case, a number of peak-load pricing variants have appeared in the literature and in practice. One scheme, proposed by Panzar and Sibley in 1978, allows customer selection of

¹¹² Bye, R., (1926) "The Nature of Fundamental Elements of Costs", *Quarterly Journal of Economics* 41(November), 30-63.

¹¹³ Bye, R. (1929) "Composite Demand and Joint Supply in Relation to Public Utility Rates", *Quarterly Journal of Economics*, 44(November), 40-62.

¹¹⁴ Crew, M. & Kleindorfer, P. (1986) *The Economics of Public Utility Regulation*, MIT Press, Cambridge, MA

¹¹⁵ Oi, W. (1971) "A Disneyland Dilemma: Two-Part Tariffs for a Mickey Mouse Monopoly", *Quarterly Journal of Economics*, 85(1: February), 77-96.

¹¹⁶ Crew, M., Fernando, C., & Kleindorfer, P. (1995) "The Theory of Peak Load Pricing: A Survey", *Journal of Regulatory Economics*, 8(0) 215-248.

individual time-invariant capacity¹¹⁷. Prioritization based on a similar capacity selection was described by Spulber in 1992^{118 119}. The real-time dynamic pricing approach was put forth by Vickrey in 1971¹²⁰. Although not adopted by telecommunications firms, a modified version of real-time pricing, with one-day forward price announcements, was adopted by some utilities¹²¹. Such a scheme may be applicable to the internet at some point in the future.

A considerable amount of work has been done on telephone systems pricing, and much is applicable. Most studies have found that the demand is extremely inelastic with regards to access charges, with ranges from -0.05 to -0.17¹²². This is at least partially due to a regulatory environment in which long distance charges were meant to subsidize access, making access more widely available¹²³. This coincided with monopoly regulation which included regulation on rate of return¹²⁴. On the other hand, long distance charges were priced such that demand elasticity was closer to -1 (revenue maximization), although survey results varied widely¹²⁵. It was also found that there was a great deal of inertia, with short run demand considerably less price elastic than long run demand¹²⁶. Coincidentally, studies have shown that consumers are generally unaware of specific item pricing on their telephone bills, but rather are aware of only of general price trends¹²⁷. More interestingly, some

¹¹⁷ Panzar, J., & Sibley, D. (1978) "Public Utility Pricing under Risk: The Case of Self-Rationing", *American Economic Review*, 68(5) 888-95.

¹¹⁸ Spulber, D. (1992) "Optimal Nonlinear Pricing and Contingent Contracts", *International Economic Review*, 33(4) 747-72.

¹¹⁹ Spulber, D. (1992) "Capacity-Contingent Nonlinear Pricing by Regulated Firms", *Journal of Regulatory Economics*, 4(4) 299-320.

¹²⁰ Vickrey, W. (1971) "Responsive Pricing of Public Utility Services", *Bell Journal of Economics*, 2(1:Spring) 337-46.

¹²¹ Crew, M., Fernando, C., & Kleindorfer, P. (1995) "The Theory of Peak-Load Pricing: A Survey", *Journal of Regulatory Economics*, 8(0) 215-48.

¹²² Taylor, Lester (1980) Telecommunications Demand: A Survey and Critique, Ballinger Publishing Company, Cambridge, MA, p.80

¹²³ Mitchell & Vogelsang (1991) Telecommunications Pricing: Theory & Practice Cambridge University Press, Cambridge

¹²⁴ *ibid*

¹²⁵ Taylor (1980), p. 99

¹²⁶ *ibid*

¹²⁷ Bidwell, M., Wang, B., & Zona, D., "Analysis of Asymmetric Demand Response to Price Changes", *Journal of Regulatory Economics*, 8:285-298 (1995).

recent studies have shown the effectiveness marketing in altering elasticities^{128 129}. This is not surprising as telephone companies enter a new period of increased competition. They are discovering the effectiveness of marketing (witness the ubiquitous marketing efforts of U.S. long distance phone companies), which they didn't need when they were monopolies. In addition to price promotions¹³⁰, telecomms are using various forms of non-linear pricing to allow customers to select the price plan that best suits them¹³¹. This self selection gives the companies information about customers, allowing them to price discriminate and to tailor price plans. This type of customer selected pricing could also be beneficial in the internet.

¹²⁸ Cracknell, D. & Knott, M. "The Measurement of Price Elasticities - the BT Experience", *The International Journal of Forecasting* 11 (1995) 321-329.

¹²⁹ Hanssens, D. & Parsons, L. (1994) "Econometric and Time Series Market Response Models" in OR/MS in Marketing Handbook Eliashberg & Lilien (eds.) Elsevier, New York

¹³⁰ Cracknell & Knott (1995)

¹³¹ Mitchell & Vogelsang (1991)

5. Summary

Usage pricing would be the ideal mechanism to alleviate congestion, were it relatively costless to implement. It is not. It would be extremely burdensome to implement. When one combines this with the two-part tariff model in which the fixed fee pays for capacity expansion, one can justify flat rates for a rapidly expanding medium such as the internet.

However, usage pricing may still prove ideal when the internet is less dynamic and more mature. It will also be less problematic if the consolidation and affiliation of ISPs and NSPs occurs (as suggested in the source routing sections above). Some of the experiments done in metering have demonstrated potential means of implementation. Other issues, such as billing sender or recipient will also arise, but they too can be solved by various mechanisms and conventions.

G. Summary

We have considered and analyzed three problems in the current structure and pricing of the internet.

The first issue that we addressed was the anonymous nature of the internet, in the sense that users typically do not know the entities that are involved in forwarding their messages. We constructed an economic model that brought user utility maximization into equilibrium with producer profit maximization (via constrained maximization and the LaGrange method). Into this model we incorporated algorithmic detail of network delay, which is a novelty in economic analysis. We demonstrated that the anonymous (exogenous) network delay that is common in the current internet, has perverse incentives for network service providers. Service providers actually have incentives to decrease their investment in capacity as exogenous network delay increases.

We then considered the case of reducing network delay by endogenizing it. This can be accomplished through the explicit selection by users of the entities that will forward their messages. This selection of NSPs as "long distance carriers" provides a number of potential benefits and is technologically feasible. As the anonymous (exogenous) portion of the delay is reduced, service providers have incentives to increase capacity. And as other service providers upgrade, they have double incentives to do so (competition, and further reduction in network delay).

We saw that there are a number of methods of specifying intermediate forwarding agents within the current internet structure, but each has limitations. For example, IP within IP encapsulation ("tunnelling") specifies a single IP address as the decapsulation point which provides less flexibility than the specification of an AS, and requires multiple encapsulations to specify multiple intermediate points. On the other hand, SDRP has security risks and its functionality is incorporated into IPv6. IPv6 addresses all the source routing concerns and should be the most efficient, flexible, and secure method of source demand routing.

And importantly, this source routing capability could be unilaterally implemented by a single large NSP to the benefit of itself and its customers. When and if other providers implement the same functionality, welfare will be increased further.

In the next section we analyzed multiple priority queues. We demonstrated how prioritized queuing can provide both users and service providers with significant benefits while requiring little in the form of additional overhead.

We reviewed the considerable shortcomings of RSVP, which nevertheless is widely (and incorrectly) considered as the coming solution to congestion problems. Its problems include complexity and especially overhead, making its deployment impossible unless it is modified, and burdensome even if modified. Also, RSVP does not address the problem of minimizing delay for time-critical customers with bursty traffic, which is precisely the profile of most internet users today.

To evaluate priority queue performance we devised a comparative metric and applied it to situations with a limited fixed number of priorities and load factors that are common today. We saw tremendous potential benefits in priority queue performance.

We also analyzed in detail the modifications and overhead that would be required in IP to implement priority queues with software-enforced bandwidth limits on incoming flows. We found this to be implementable with minimal processing and storage overhead. And we looked at the interoperation and coordination details that would be required and found that these could be handled with little complexity. Moreover, if source demand routing is implemented by an NSP, multiple priority queues could also be implemented unilaterally with tangible benefits.

We found prioritized queuing to be a relatively simple solution to a problem besieged by overly complex solutions, and one which provides benefits to both users and providers.

Finally we looked at usage pricing. Clearly it would be the ideal mechanism to alleviate congestion if it were relatively costless to implement. But this is not the case as the overhead would

be staggering. In our review of telecommunications and utility pricing schemes we considered the two-part tariff, incorporating both access and usage charges. We saw that in this model the fixed fee typically pays for capacity expansion while the usage charges cover the operations. In a rapidly expanding medium such as the internet, capacity expansion costs clearly dominate. So when the cost of gathering the minimal usage costs is overwhelming, it makes sense to eliminate the usage charges and use access fees only.

However, usage pricing should not be dismissed out of hand. It may very well be useful and feasible when the internet is more static and more consolidated. We also looked at implementation details if usage pricing were to be implemented, and found that many of the questions could be overcome without great complexity.

So while Internet usage and delays are increasing rapidly, and while there are presently inefficient economic incentives for both users and providers, there are ways to address these problems within the current framework. We strongly suggest an approach based on economic incentives that takes into account issues like the deleterious effects of exogenous network delay, and the variation in user preferences and willingness to pay for reduced delay.

We have outlined what we believe are practical ways to incorporate these principles into the existing structure with as little disruption as possible. Moreover we suggest that a large NSP could unilaterally implement solutions to these problems via source demand routing and multiple priority queues, to its great benefit as well as the benefit of its customers. Participation by additional NSPs and ISPs would only serve to increase welfare further.

H. Epilogue

As an mini-epilogue to the original version of this essay, it's interesting to note that in early 1997, Cascade Communications Corp. introduced a product called "Priority Frame" that will let carriers offer four levels of prioritization and quality of service over frame relay switches.¹³² In addition, MCI and Cisco Systems announced a software-based technology that will enable customers to receive a premium grade of Internet service¹³³. Other trials using the Traffic Class field of IPv6 are also reported to be underway.

¹³² Cascade Communications Corp., "Cascade Reaffirms Frame Relay Leadership Position With Priority Frame™; Industry's First Quality of Service Technology For Frame Relay", <<http://www.casc.com/company/press/pfrel4.html>>, 1/20/97

¹³³ MCI Corp., "CISCO AND MCI LEAD THE WAY IN BRINGING "PREMIUM" GRADE SERVICE TO THE INTERNET", <<http://www.mci.com/aboutmci/news/indexnewsrefresh.shtml>>, 3/25/97

III. Software Pricing and Software Engineering

A. Introduction

The production of software is a unique phenomenon for many reasons. Once developed, it can be costlessly reproduced and distributed. And once a user has taken the time to learn and implement a software package, the effective cost of converting to a different package becomes steep and likely dissuasive. And most importantly, there are significant network externalities which make a software package more valuable with the size of its user base. So market share, which is important in any type of market, is even more important in the software market. And timing is critical in the growth of market share. Moreover, given the difficulty of thorough independent product evaluation, software purchase patterns may be likely to show evidence of information cascades. For these reasons and others, there are significant market advantages for software products that are first to market. The development process will be considered with this time constraint in mind.

Software is also unique in that a given package usually goes through a number of releases or versions as new features are added and as it is upgraded to work with other new or upgraded products. This provides an ongoing revenue stream to the producer, which in fact can dwarf the original purchase price. This poses interesting questions of optimal pricing and timing of releases. The latter should have direct impact on the development process.

The following sections first discuss, build and analyze economic models of software markets. The lifecycle of a software product is considered in the context of the market segment lifecycle. We document and consider the clustering of agents' decisions over time in the selection of our model function. This cycle includes market growth, plateau and decline. The model that we will build incorporates this market segment lifecycle, and reflects the impact of the timing of a particular product's introduction. The beneficial effects of an increased installed base of users, the network externalities, will also be incorporated into the model. Finally this must be combined with a demand function that is affected over time by these forces.

In light of all these issues, we will build a unique framework for the analysis of software pricing and releases. The resultant problem is closely related to optimal control theory and allows insights to be derived as to optimal pricing and timing decisions. The timing issues will be of critical importance to us in our subsequent analysis of the software engineering function. A number of variations on the model are also considered, including explicit recognition of upgrade sales versus new sales, and the eventual plateau and atrophy of market share. The optimal timing of releases is analyzed in the context of the product lifecycle and the resultant mix of customer types.

In addition to the insights gained from our model, we will also discuss other issues impacting software firms' profitability. These will include an expansion on issues behind our model, such as the advantages to being first to market. We will also discuss information cascades in general, which are somewhat implicit in our model, and the likelihood and the impacts of their occurrence in software markets. Further, we will discuss the issue of the pre-announcement of software product features, which is not only a common practice but also one which fits into our optimal release model. And we will also mention other issues impacting software firms' profitability including the necessity and opportunities of technical support, and how that fits into practice and into our optimal release model.

Subsequently, the ramifications on the software engineering process are considered, and suggestions are developed to tailor the development process to the economic benefit of the software firm. We will first review a number of software engineering process models. Among these will be both historic models (e.g. waterfall) and current models (e.g. OO and RAD). We will note those that provide the best fit with the exigencies of the market as we have described it.

We will then make the case for the promulgation of a new or modified development paradigm, and discuss much of the rationale for the design of our novel software development process model. The new market exigencies will be discussed and their effects on the ideal development process considered.

We will then present in set notation format, the goals and mechanics of our proposed development process. In particular we will look at the potential features (or functions) of a software product as having both a benefit value (a priority) and a cost (in terms of development time). We will look at the maximization of benefit within the fixed cost constraints of optimal release cycle timing and

budgetary considerations. The addition of the prioritization of features and the explicit recognition of schedule and budgetary considerations is rather novel in terms of development methodology.

Finally, we will present a series of steps that incorporates all of our preceding analysis. These steps represent our view of the optimal software development process, taking into account the exigencies and opportunities of the current software market environment. This optimality is both in terms of economic profitability of the firm, and also in terms of maximum effectiveness of the project and those involved. Our model departs in subtle but important ways from existing models. Most importantly is the expansion of the analysis phase, and the prioritization and schedule considerations mentioned above.

B. Software Pricing and Timing

The software pricing issue is unusual for a number reasons. The marginal cost of a software package is near zero (and when distributed over the internet it may actually be zero). Many software markets exhibit rapid coalescence around a standard. Software products are frequently modular and go through many releases and upgrades.

The marginal cost is zero because given complete documentation including tutorials, the package can be downloaded and installed by the user without cost to the firm. Although technical support may seem to imply a small non-zero cost, this can be charged for separately and in fact become a source of revenue. So we can reasonably assume that there is a zero marginal cost. If we were to consider the software market perfectly competitive we would expect that the firm will set its price at zero, the marginal cost. And in fact many software packages have zero or near zero prices. But profits are by no means driven to zero in this market. For example, the market valuation of Microsoft is in excess of \$150 billion, indicating profits and profit potential considerably above zero.

Like most markets, the software market is not perfectly competitive. Rather, it is monopolistically competitive tending towards oligopoly and monopoly. To wit, we certainly have differentiated products. We have sloping demand curves as evidenced by software firms price-setting behavior rather than price taking. And we frequently have prices above marginal costs. We also have some characteristics of a natural monopoly. There are high costs of entry to produce and advertise a software product that can be downloaded, learned, and operated independently by many users. And the average cost decreases constantly as the number of users increases. Furthermore, when one considers the network externalities that result from additional users there is even more evidence of a natural monopoly.

Network externalities are the "side effects" of a large network of users. These are almost invariably positive (although by no means are all externalities positive in other economic situations). These network externalities include the establishment and control of de facto standards and their

franchise value. These include ease of interoperability with other users of the same product and with other products made explicitly to be compatible. The number of (new) products that are made compatible is an extremely important externality (it's usually proportionate to the number of users of the existing product), and is one that is very evident in the market today. Another positive network externality of a large user base is the resultant large pool of trained users. From the users' viewpoint, the value of the product increases with this available pool of trained users, with the number of compatible ancillary products, with the de facto confirmation of other users of the product's suitability, and with the decreased in the risk of the technology being "orphaned". Only the rapid evolution of the software market slows down the convergence to monopoly and disrupts existing monopolies¹³⁴.

(It is interesting to note the basic difference with the economics of another type of zero marginal cost item: information. Information ownership increases in value with exclusivity. Software ownership is the opposite.)

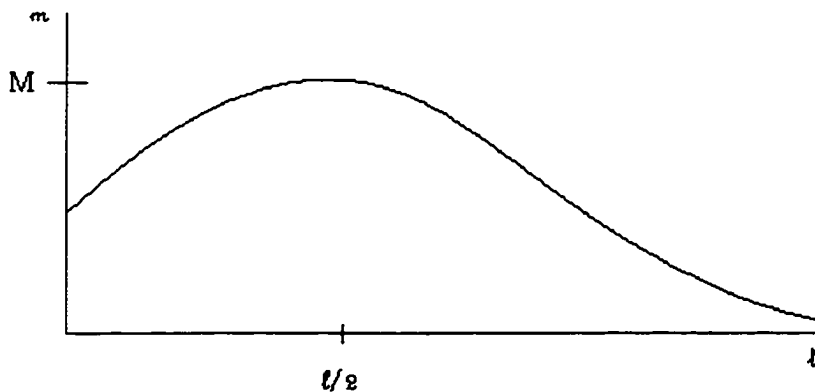
So we would like to model the market that we have described to gain insights into pricing and release patterns that can maximize the software firm's profit. From these we would like to glean insights into optimizing software engineering methodology.

Our model should include the series of many releases of a software product rather than simply modeling one-period supply and demand. It should also encompass the life cycle of market growth and maturation.

¹³⁴ The evolution of the technology market is not entirely unlike the "punctuated equilibrium" theory of the evolution of life: we have recently seen a mass extinction and are now seeing a proliferation of new life forms.

1. Software Market Growth Model

We will begin with a model of market segment life cycle. The following graph plots a market segment growth factor against a time line. We will use this market segment growth factor as a dynamic element that will be applied to our standard (static) quantity and price demand schedule.



m is the market segment growth factor, with M being its maximum value. This maximum value occurs at $t/2$ where t is the length of the market segment life cycle under investigation.

The formula that we will use for our market segment growth factor g is:

$$g(t) = \frac{\eta}{\sqrt{2\pi}} \cdot e^{-\frac{1}{2}(t-l/2)^2}$$

which is a bell curve that peaks at M at time $l/2$. The formula is in the format of the standard distribution for sake of convenience, and because this may be a good approximation over time of the clustering of new customers, contributed to by an information cascade. The area under our function is then given by η . The starting point at time zero is an arbitrary point on the upslope and represents a demand for the product (before or) at its initial release¹³⁵. In our graph, we have shown a point perhaps midway up the slope. This function can be thought of as applicable to the product category, rather than the particular product.

In addition to a product-category market-growth factor, we will also have a specific-product market-size factor to capture network externalities. These factors, indicated above, make a product more valuable with the number of users of the product. This should be strictly increasing with the number of users. For simplicity sake we may want to consider a linear function. A function with a positive first derivative and negative second derivative (such as a log function) has also been used¹³⁶. But network externalities may also be modeled as quadratics, corresponding to the number of potential relations between nodes on a network.

¹³⁵ A statistical function with a fixed beginning point, such as an F-statistic, might also be used. However the standard normal function is used to indicate that the market segment growth will follow its course regardless of the timing of our particular product.

¹³⁶ Economides, N. & Himmelberg, C. (1995) "Critical Mass and Network Size with Application to the US FAX Market", Working Paper no. EC-95-11, Stern School of Business, N.Y.U. <<http://raven.stern.nyu.edu/networks/95-11.ps>>

We may consider an equation such as the following (where c is an arbitrary constant)

$$m(y) = cy^2$$

We will use y to indicate the number of users of the product (which will be the state variable in our optimal control problem).

We will also need a demand function, matching quantity to price at points in time. We will use a linear function. Linear functions have been used in other models of software demand^{137 138}, due both to their simplicity and the fact that they fit the data¹³⁹. So we have

$$q(p) = a - bp$$

We can specify the function for revenue at any given point in time t as a function of the quantity sold and the price at that point in time. We use y to indicate the number of users of our product, and \dot{y} as the change in y over time (we are only considering the possibility of new users at this point). So we have revenue at a given time t equal to the number of new customers at that time, multiplied by the price at that time.

$$R_t = \dot{y}_t p_t$$

where

$$\dot{y} = \frac{dy}{dt} = g(t) \cdot q(p) \cdot m(y)$$

¹³⁷ Katz, M. & Shapiro, C. (1985) "Network Externalities, Competition, and Compatibility", *American Economic Review* 75(3) p. 424-40

¹³⁸ Swann, P. & Gill, J (1993)

¹³⁹ *ibid*

With zero or near zero replication and distribution costs, our cost function is fixed, and we can maximize profit by maximizing revenue. A software firm can then maximize its profit by maximizing the following function over the product's lifetime.

$$R = \int_0^l \dot{y}_t p_t e^{-rt} dt$$

subject to the constraint

$$\dot{y} = g(t) \cdot q(p) \cdot m(y)$$

where the discount factor (interest rate) r is applied to include the time value of money (in the first equation).

To maximize these, we can construct a Hamiltonian and use the Pontryagin principle for constrained maximization within an integral. We begin with

$$R = \int_0^l \left[\dot{y}_t p_t e^{-rt} - \lambda (\dot{y} - g(t) \cdot q(p) \cdot m(y)) \right] dt$$

with lambda as in the Lagrangean constrained maximization

We continue by substituting $\int (\lambda \cdot \dot{y}) dt = \lambda \cdot y - \int (\dot{\lambda} \cdot y) dt$

giving us

$$R = \int_0^l \left[\dot{y}_t p_t e^{-rt} + \lambda (g(t) \cdot q(p) \cdot m(y)) + \dot{\lambda} y \right] dt - \lambda(t) y(t) + \lambda(0) y(0)$$

where the last two terms are the transversality conditions regarding the endpoints. We would need to include these in our analysis to solve for a detailed optimal price function. However, such a detailed function will not be necessary for us to derive the implications that we are interested in, so we will omit these calculations.

Pontryagin's optimal control theory principle states that if a maximizing solution exists, then it maximizes the terms in the integral.

The first order conditions for maximization are then given by

$$1) \frac{dH}{dp} = 0$$

$$2) \dot{\lambda} = -\frac{dH}{dy}$$

$$3) \dot{y} = \frac{dH}{d\lambda} \quad (\text{which is the constraint})$$

where

$$H = \dot{y} \cdot p \cdot e^{-rt} + \lambda (g(t) \cdot q(p) \cdot m(y))$$

and substituting for \dot{y} , we have

$$H = (p \cdot e^{-rt} + \lambda) (g(t) \cdot q(p) \cdot m(y))$$

Then using our linear demand function, and solving F.O.C. 1 above, we have

$$\frac{dH}{dp} = 0 = g(t)m(y) \left[(a - bp)e^{-rt} - b(pe^{-rt} + \lambda) \right]$$

so

$$2bpe^{-rt} = ae^{-rt} - b\lambda$$

and

$$p = \frac{a}{2b} - \frac{\lambda \cdot e^{rt}}{2}$$

To complete the solution, we solve F.O.C.2 to get lambda in terms of y and p, and we solve F.O.C.3 to get y in terms of p. Then substituting the latter into the former, we have lambda in terms of p, which we substitute into the preceding solution for p.

recall F.O.C.2, in which

$$\dot{\lambda} = -\frac{dH}{dy}$$

so we have

$$\frac{d\lambda}{dt} + g(t)q(p)m'(y)\lambda = -g(t)q(p)m'(y)(pe^{-rt})$$

which we can solve using the standard integrating factor from

$$\frac{d\lambda}{dt} + P(t)\lambda = Q(t) \Rightarrow \lambda = e^{-\int P(t)dt} \cdot \int e^{\int P(t)dt} Q(t)dt + C$$

giving us

$$\lambda = e^{-\int g(t)q(p)m'(y)dt} \int e^{\int g(t)q(p)m'(y)dt} [-g(t)q(p)m'(y)(pe^{-rt})] dt$$

The only form of y that we have in the above equation is m'. (So if we were to use a linear function for our market size network externality function m then y would disappear from the above, whereas our quadratic term will remain.) We will solve for y through F.O.C.3 below.

Recall F.O.C.3:

$$\dot{y} = \frac{dH}{d\lambda}$$

which evaluates to our constraint

$$\dot{y} = g(t) \cdot q(p) \cdot m(y)$$

so

$$\int \frac{dy}{m(y)} = \int g(t)q(p)dt$$

using our quadratic network externality function $m(y) = cy^2$, we have

$$y = -\left(\frac{1}{c} \int g(t)q(p)dt\right)^{-1} + C$$

where C is an arbitrary integration constant.

So we return to our solution above for p

$$p = \frac{a}{2b} - \frac{\lambda \cdot e^{-\lambda t}}{2}$$

and substitute our solution for lambda

$$p = \frac{a}{2b} - \frac{e^{-\lambda t}}{2} e^{-\int g(t)q(p)m(y)dt} \int e^{\int g(t)q(p)m(y)dt} \left[-g(t)q(p)m(y)(pe^{-\lambda t}) \right] dt$$

and substituting in $m'(y) = 2cy = -\left(2 \int g(t)q(p)dt\right)^{-1} + C$ we have

$$p = \frac{a}{2b} + pe^{\lambda t + 2z} \int ze^{-(\lambda + 2z)t} dt$$

or

$$p = \frac{\left(\frac{a}{2b}\right)}{\left[1 - e^{r+2z} \int z e^{-(r+2z)} dt\right]}$$

where

$$z = z(t) = \int g(t) \left[\int g(t) dt \right]^{-1} dt$$

which we can reduce by the formula

$$\int \frac{f'(x)}{f(x)} dx = \ln f(x) + C$$

to

$$z = \ln \left(\int g(t) dt \right) + C$$

In the simplest case, where r is zero (no interest rate factor), we have

$$p = \frac{\left(\frac{a}{2b}\right)}{\left[1 - e^{2z} \int z e^{-2z} dt\right]}$$

$$= \frac{\left(\frac{a}{2b}\right)}{\left[1 - \left(\int g(t) dt\right)^2 \int \frac{\ln \left(\int g(t) dt\right) + C}{\left(\int g(t) dt\right)^2} dt\right]}$$

Then using the formula

$$\int x^m \log x dx = x^{m+1} \left[\frac{\log x}{m+1} - \frac{1}{(m+1)^2} \right] + C$$

(where $m = -2$), we would have

$$p = \left(\frac{a}{2b} \right) \left[1 + \left(\int g(t) dt \right) \left(\ln \left(\int g(t) dt \right) + 1 - \int \frac{C}{\left(\int g(t) dt \right)^2} dt \right) \right]$$

The simplest solution (depending on the endpoint restrictions as embodied by the transversality conditions), is that in which the integration constant C is zero. Then with the integral of the market growth factor g equal to 1 at time 0, then the price at time zero would be half the usual single period revenue maximization price.

Also, the price drops over time, with the speed of the price drop proportionate to the increase in the growth function. (The effects of the constant of integration are diminished over time.) This is contra-intuitive, but is because our model only recognizes market growth and has an infinite horizon.

In reality our horizon is not infinite, so we might also add a term indicating the number of users who discontinue using our product (as each users' time horizon is finite). This would of course affect the user base, and would be an accurate representation of a new product in the market superseding the existing one.

In this case our analysis would be similar, and only our market growth function would be different. Here the market growth function would turn negative when the number of defections to competing products equalled the number of sales to new users. At this point our optimal price would then begin

to rise. It would continue rising until it approached the one period optimum as our market share continued to diminish and approached zero.

Finally, we might also add a term for upgrades, as our current model includes only new sales.

We would then have several changes to our model, including the following

$$n = g(t, l) \cdot q(p) \cdot m(y)$$

$$v = u \cdot y$$

$$d = g(t, 2l) \cdot y$$

where n is the number of new users, v is the number of upgrades, and d is the number of former users that discontinue use of the product. We use a similar growth function with double the lifetime to denote the number of defections.

Our maximization problem is then given by

$$R = \int_0^l (p_t \cdot n + p_t \cdot v) dt = \int_0^l p_t [g_1(t) \cdot q(p) \cdot m(y) + u \cdot y] dt$$

where we want to maximize the combined revenue from the sales to new customers and the sales to existing customers. We assume that the price is the same for all customers.

Our maximization is subject to the market size change function of

$$\dot{y} := n - d$$

where n is the number of new users and d is the number of former users who discontinue using the product.

Using the same maximization steps as in the previous, the first of the first order conditions evaluates to

$$p = \frac{a}{2b} + \frac{u \cdot y}{2bg(t)m(y)} - \frac{\lambda \cdot e^{\alpha}}{2}$$

Only the middle term is new, and it demonstrates that the optimal price may initially be higher when including upgrades in the analysis. However it would be only slightly higher as the numerator of the term is the market size, which at time zero is small. Also, the inclusion of upgrades (the middle term) will at least dampen if not entirely offset the price drop in the early stages of the product life cycle.

Note that the u term above is related to the portion of existing users that will upgrade at any given point. The value of u may range from 0, meaning no existing customers can be expected to upgrade, to 1, indicating that all existing customers can be expected to upgrade. Although we modeled it as a constant, it could also be a function of time. At time 0 it would certainly be zero, and the middle term would then drop out, leaving us with the initial optimal price equal to one half the single-period revenue maximizing price.

2. Model Implications; Optimal Price and Time to Market

With all things considered, our optimal price then would start at one half the single-period revenue maximizing price. During the period of market segment growth, various factors contribute both positively and negatively to the optimal price, and the net effect is not clear. However, at the point that the market growth begins to plateau, the optimal price would begin to rise gradually towards the single-period revenue maximizing price.

Also, it is clear from our model that earlier to market is exponentially important. Our model embodies the effects of competitors' products in that the market size of a given product is limited by its particular time 0 on the market segment growth curve. If a given product's time 0 (initial release time) is close to the market plateau, then that product's maximum market size is severely limited (other products have implicitly realized most of the market segment potential). However, if time zero for a given product is early in the market segment lifecycle, then that product has vast growth potential.

For example, using the normal distribution formula as our growth rate

$$g(t) = N \cdot e^{-\frac{1}{2}(t-t/2)^2}$$

with mean 0 and standard deviation(σ) 1, then at $t=0$ we have the following calculations (where a greater value for $1/2$, half the lifecycle, means that the product was earlier to market and further from its peak growth rate):

if $1/2=\sigma/2$ then $g(0)=0.88N$, so the max growth rate is 1.13 times the initial rate

if $1/2=\sigma$ then $g(0)=0.61N$, so the max growth rate is 1.65 times the initial rate

if $1/2=3\sigma/2$ then $g(0)=0.32N$, so the max growth rate is 3.08 times the initial rate

if $1/2=2\sigma$ then $g(0)=0.14N$, so the max growth rate is 7.39 times the initial rate

and since we always normalized the initial growth rate to 1 in our model, the above represent a direct comparison in potential growth rates. And the growth rate differentials are compounded over time in the market size. So clearly the time to market in our model is of critical importance to the potential profitability of the product. We will discuss some of the reasons below.

3. Upgrade Frequency

Our model implicitly assumed a continuous product upgrade sequence. In practice, product releases are discrete. In addition to the technical challenge and feasibility question of producing and supporting continuous releases, we have the marketing question. Would users prefer, or even accept such a process?

The answer may depend on the type of customer. We can consider here that there are two basic categories of customers for any given release; existing customers who are upgrading, and new customers (here we group users that are new to the market segment and those converting from competing products).

Existing customers may not want to see continuous, or even very frequent releases. This is primarily because of the time cost involved in implementing new releases (in addition to the actual upgrade price). From an existing customer's perspective, he may always want to have the latest release, but he may not want to have to upgrade too frequently to do so. And from the firm's perspective, it's relatively safer to make existing customers wait for features. They are less likely to switch to competing products due to the costs and difficulty involved in conversions.

New customers, on the other hand, have no upgrade overhead and would like to have all possible features. Prior to their purchase, they are likely comparing the product with other similar products. From the firm's perspective, it would like for the customer to have all possible features immediately available for more favorable comparisons.

Against this background, are the firm's version control and support requirements. It is obviously easier for the firm to have fewer releases. Moreover, the development process is more straightforward when all potential features are analyzed and designed together in a single static specification.

So early in the market segment lifecycle, when new customers are likely to dominate upgrade customers, it would be economically beneficial to the firm to have rather rapid releases. This insures that as new customers enter the market segment and evaluate products, the firm's products have as

many features as possible available. It is also intuitive that in a new market there are rapid changes. Rapid releases are required.

However, as the market matures and becomes primarily an upgrade market, a more deliberate series of releases would likely be optimal. The increasing amount of upgrade customers for subsequent releases argues for an increasing release period. The optimal length of this period is of course affected by numerous other factors, such as the time cost involved in upgrading, and the competitive landscape. Nevertheless, we can deduce that an optimal release schedule might be one that initially had a very short cycle between releases with the cycle time gradually lengthening as the number of existing customers increases.

C. Other Profit and Revenue Considerations and Discussion

In addition to our model, which assumes that we cannot price discriminate, it may be useful to consider that there are distinct types of customers; those new to market, those upgrading, and those switching from competitors' products. These customers have differing intrinsic costs in terms of conversion and learning time. The time and effort cost for a user to switch products is significant. A software firm may however be able to adjust its price with either the sale or free distribution of upgrade and conversion tools. It may effectively reduce the price of the product for competitor's customers by offering free conversion tools and support. It could also effectively increase the price to existing customers by the sale of upgrade tools.

Also, our analysis has suggested an approach of many releases, adding features and planning and possibly announcing other features in advance. This is a also marketing issue and an issue of competitive "fairness" (the vaporware argument)¹⁴⁰. Studies have shown that the network externalities depend upon consumers' expectations^{141 142}. The announcement of vaporware can be an effective marketing tool if the firm is credible. If a firms' customers know that the product will soon contain a feature that they want, they will be less inclined to switch to another vendor, especially given the time cost of learning a new system.

The feature pre-announcement issue is also related to the optimal timing of releases, and to the mix of existing customers versus new customers. If the sales of a given release are likely to include a significant proportion of (existing) upgrade customers, then feature pre-announcement may be effective in satisfying these customers that additional features are forthcoming, and allowing them to

¹⁴⁰ Farrell, J. & Saloner, G. (1986) "Installed Base and Compatibility: Innovations, Product Pre-announcements, and Predation", *American Economic Review* (76) 940-55.

¹⁴¹ Katz, M. & Shapiro, C. (1985)

¹⁴² Katz, M. & Shapiro, C. (1986) "Technology Adoption in the Presence of Network Externalities", *Journal of Political Economy*, 94(4) p.822-41

time their upgrades to their advantage. However, if it is early in the product lifecycle and a large majority of customers will be new customers, then feature pre-announcement will not be effective. For one thing, the firm may not have established credibility in the market. Also, these announcements may cause customers to delay their purchases, and possibly reconsider their purchase decision altogether.

While the use of vaporware as a marketing tool is generally decried, there is perhaps an even more cynical force at work in the profit picture. This is revenue from technical support. If a firm releases a popular product that is generally intuitive but has a number of areas that are not entirely clear, then many customers will be compelled to use the firm's technical support services. Technical support can become a major source of income for a software firm. Witness Oracle which recently derived more than 45% of its revenue from "services" (with the cost of services being only 26% of total costs)¹⁴³.

Given the network externalities of a common standard: a wide base of expertise to draw on, a reduction in the risk of obsolescence and standard abandonment, limited conversion requirements and maximum interoperability, it is not surprising that we have rapid convergence toward standards in any given software area¹⁴⁴. As a corollary, the first to market has a "strategic, first-mover advantage" in the bid to become the standard¹⁴⁵. Furthermore, once the market has chosen a de facto standard, it will be extremely reluctant to move to a different standard¹⁴⁶. These notions are captured in our model with its premium on early entry.

The emergence of a product as the standard is a rapid process, as well as an interesting one. The process is related to the phenomenon of information cascades. In this scenario, early adopters disproportionately influence mass behavior by their decisions (which may even have a random

¹⁴³ <http://www.oracle.com/corporate/html/earningstable2.html>

¹⁴⁴ Swann, P., & Gill, J. (1993) Corporate Vision and Rapid Technological Change, Routledge, London

¹⁴⁵ Katz, M. & Shapiro, C. (1986)

¹⁴⁶ Farrell, J. & Saloner, G. (1985) "Standardization, Compatibility, and Innovation", *Rand Journal of Economics*, 16(1) p.70-83

element)¹⁴⁷. The phenomenon is often described in relation to the financial markets¹⁴⁸. In one light, however, a product market (like software) has significant differences with the financial markets: In financial markets the early trendsetters have significant potential for monetary gain. In much of technology, in fact, it's likely that the price will fall, so early adopters may even pay a premium. But there are also similarities to the financial markets in terms of information cascades. Studies have shown herd behavior, or clustering, among financial analysts and have also shown why it is practiced; risk aversion¹⁴⁹ ¹⁵⁰. This factor, risk reduction, is precisely why we would expect information cascades in software. Technology is complex. Often the resident expert is not in fact expert in all areas. In areas of unfamiliarity, one can not be faulted much for going with the accepted wisdom. (A related anecdote involves IBM of the 1970s. It was widely regarded in the industry that other vendors frequently had superior products at equal or lower prices, but users were afraid to take a chance. They knew that they couldn't be accused of making a big mistake by going with IBM.) Given the uncertainty in software product evaluation, it is also not surprising to see clustering of agents' decisions in terms of time¹⁵¹. This gives further credence to the speed with which a standard will emerge.

There is another factor at work with software, however, that might make it somewhat less prone to "erroneous" cascades. Information cascades and herd behavior arise largely by agents substituting an analysis of other agents behavior for an analysis of the product in question. The less the dependence on the collective wisdom, the less likely an incorrect cascade becomes¹⁵². It may be that this dependence is less prevalent with early technology adopters, as they may be more iconoclastic and analytically self-reliant. This would mean that there would truly have to be substance to gain

¹⁴⁷ Bikchandani, Hirshleifer, & Welch (1992) "A Theory of Fads, Fashion, Custom, and Cultural Change as Informational Cascades", *Journal of Political Economy*, 100(5) p 992-1026

¹⁴⁸ (See just about any arbitrary issue of *Journal of Finance* in the 1990s. The following is an example.)
Froot, Scharfstein, & Stein (1992) "Herd on the Street: Informational Inefficiencies in a Market with Short-Term Speculation", *Journal of Finance* 47(Sept) p1461-1484

¹⁴⁹ *ibid*

¹⁵⁰ Stickel, S. (1992) "Reputation and Performance among Security Analysts", *Journal of Finance* 47(December) ;1811-1836

¹⁵¹ Gul & Lundholm (1995) "Endogenous Timing and the Clustering of Agents' Decisions", *Journal of Political Economy* 103(5) p 1039-1066

¹⁵² Orlean, A. (1995) "Bayesian Interactions and Collective Dynamics of Opinion: Herd Behavior and Mimetic Contagion", *Journal of Economic Behavior and Organization*, 28(0) p257-274

favor. Of course this knowledge would intensify the cascade phenomenon once it got started and spread to those less analytically self-reliant.

So we have the market pressures for early entry coming from the trend of rapid market concentration, and we have the market and financial pressures for rapid releases previously described, all combined with the optimality of compelling users to utilize the firm's technical support. The result is one that may play into the trade-off between software quality and development time¹⁵³. The software engineering implications of this rapid and piecemeal release strategy are discussed further in the subsequent section.

The net result of this analysis is that a profit maximizing software development firm will have a low initial price for the product, will come to market quickly, and will release many upgrades fairly rapidly. The pace of upgrades will then slow as the market matures. The low package price will create a large customer base, not only enabling the firm to benefit from network externalities and upgrade sales, but also to take advantage of other revenue opportunities. These include technical support, service and advertising. A customer list is a valuable asset which can be used to advertise the firm's other products and services, or even to advertise other firms' products and services. In the current market environment, strategic alliances are of utmost importance and can make or break a product. A large customer base increases a firm's ability to strike strategic alliances with key partners.

¹⁵³ Yourdon, E. (1996) Rise and Resurrection of the American Programmer, Prentice-Hall, Englewood Cliffs, New Jersey

D. Software Engineering Implications

The prescription is then for a planned series of fairly rapid releases, with the timing of the releases as a critical issue. How does this affect the software engineering process? Contrary to established software engineering theory, the firm may not want to include all the desired features in the initial product or in a given release. The firm however would certainly want to have as many features as possible in mind when developing the initial product or a subsequent release so as to minimize the cost of future releases and to possibly to pre-announce these deferred features. But the firm may not actually want to put them all in a given release as this may delay the release. The design problem then is to predefine a series of releases with an emphasis on timing and speed to market.

Consider the case of Firm A which designs and implements all anticipated features into its product and takes a given amount of time to do it. Its competitor, Firm B, picks off the most important and readily implementable features, brings its product to market with these only and pre-announces the rest. Once a customer is using the latter's product, the cost of switching has increased. Firm B wins.

It is interesting to note the shift in emphasis from the total quality stressed in most software engineering texts, at the expense of schedules, to the emphasis here on coming to market quickly. The former viewpoint is summarized in the following quote

"The bitterness of poor quality lingers long after the sweetness of meeting schedules is forgotten."

154

This is a perfectly reasonable credo when one is developing a software project for a given customer. One already has the customer. However when getting customers is the paramount goal, speed to market is the important issue. An alternative to the preceding quote is something like "the sweetness of market dominance lingers long after the bitterness of fixing the shortcomings of the first release".

¹⁵⁴ N. Whitten, Managing Software Development Projects, (New York: John Wiley & Sons, 1990), p.96

It should be stressed that quality is still an important issue in mass-market software development. However, total quality is not the most important issue.

In designing our development process, we should also consider another externality of the costless delivery of software over the internet: the existence of freeware and shareware components, as well as commercial components. Not only does this affect product planning, but it affects product development as well.

One of the major intended benefits of object orientation is the ease of re-use of components (of course, efficient software developers have always re-used code - it's just that OO development means to make it a more well-defined process). With freeware and shareware, the available object library is explosively increased. In fact, it is critical that a software development effort have personnel that can do an effective component search. This is perhaps a more critical skill than programming. It is as if there is an ever-increasing available class library. And more than just freeware and shareware is involved. A working knowledge of commercially available component products and their features is equally important.

For example, in classical software engineering one wants to get from point A to point B, so one analyzes and designs their way to building a road directly from A to B. But with the existence of components, and with the realities of budgets (more on this later), the problem is to determine which roads already lie somewhere between A and B, and to do as little road-building as possible. Moreover, with multiple releases planned from the beginning, one would want to plan to do a little more road building at each release. For example, in a subsequent release, one may want to optimize the path from A to B while also extending it to C.

Another point to note about most established software engineering theory, is that all objectives or goals are treated as equals. This may be true when one is developing software to a given specification. But when the specifications themselves are being developed, this is almost never truly the case. There are always some elements that are absolutely critical for the project/product to perform. There are those that are quite important, but not critical, on down to those that would be nice but don't warrant great time or expense in terms of development. Add to this mix the budget and time constraints that are a part of every non-academic project (some might also say non-government

project). An approach which prioritizes objectives and explicitly recognizes schedule and budget constraints is required.

Object Oriented Software Engineering lends itself to rapid prototyping and the iterative model ¹⁵⁵. This is obviously well-suited to the reality of multiple releases, even though a "prototype" is not typically a releasable product. But even with these models, there are a couple of critical steps that we might want to add toward the beginning of this iterative process.

¹⁵⁵ E. Yourdon, Object Oriented Systems Design, An Integrated Approach, (Englewood Cliffs, New Jersey: Prentice-Hall, 1994)

1. Software Engineering Development and LifeCycle Models

First, let's look at a version of the standard waterfall model of the software life cycle. This type of model was first proposed by Royce in 1970¹⁵⁶. The version here is from Stephen Schach¹⁵⁷:

1. Requirements
2. Specifications
3. Design
4. Implementation
5. Integration
6. Operation

Another example by Barry Boehm¹⁵⁸ is

1. Feasibility
2. Requirements
3. Design
4. Detailed Design
5. Coding & Testing
6. Integration
7. Implementation
8. Maintenance
9. Phase-out

¹⁵⁶ W. Royce, "Managing the Development of Large Software Systems: Concepts and Techniques", Proceeding of WestCon, August 1970

¹⁵⁷ S. Schach, Software Engineering, (Homewood, Illinois: Aksen Associates, 1990)

¹⁵⁸ B. Boehm, Software Engineering Economics, (Englewood Cliffs, New Jersey: Prentice-Hall, 1981)

Again, these sequences stem from the contractual custom-software model predominant in the pre-PC days. In this model, the software developer could write to spec and be assured of success. Enhancements could be treated as separate projects, while maintenance was a separate phase. Total quality assurance was assumed to take precedence over schedule.

When developing for the mass market, there is no set of specifications that can be comprehensively promulgated. One has to try to anticipate the market, and plan on rapid incremental changes. Also there is no pre-defined final product. The process should describe an ongoing sequence of releases. The standard development cycle needs to be adjusted accordingly.

Of course most recent software engineering texts have duly disparaged the old waterfall model and proposed iterative or rapid prototyping models. These certainly go part of the way in meeting the exigencies of current software development. A variation on the previous stepwise model, made by the same author, Barry Boehm, describes a "spiral" model where each step is repeatedly revisited¹⁵⁹. In this model, prototypes are developed during each iteration, along with a consideration of alternatives and a risk analysis. The final iteration includes the Unit Test & Code, Integration & Test, Acceptance Test, and Implementation phases.

¹⁵⁹ B. Boehm, "A Spiral Model of Software Development and Enhancement", *Computer*, v.21, pp. 61-72, May 1968

A relatively recent example, from the Object Management Group¹⁶⁰, based on the OO iterative approach is:

1. Strategic Modelling
2. Analysis Modelling
3. Design Modelling
4. Implementation Modelling
5. Construction
6. Delivery

These models address the iterative approach required, however still produce a single product. A recent example from Edward Yourdon¹⁶¹ which more explicitly recognizes that numerous versions need to be produced sequentially is:

1. Requirements
2. Analyze
3. Design
4. Code
5. Test
6. Demonstrate
7. Revise Estimates

(Repeat steps 2-7 for subsequent versions)

¹⁶⁰ Object Management Group, Object Analysis and Design, (Framingham, Mass.: OMG, 1992)

¹⁶¹ E. Yourdon, Object Oriented Systems Design, An Integrated Approach, (Englewood Cliffs, New Jersey: Prentice-Hall, 1994), p. 52

Perhaps the existing model that comes closest to meeting the exigencies of mass-market software development is the Evolutionary Model by Tom Gilb ¹⁶². Although this model does not address the mass-market explicitly, an illustrative example given in the text is that of a PC user creating a system for himself. In this case, the user prioritizes the usefulness of various features, and builds and implements the system piecewise, modifying design where necessary along the way. This concept is advocated in large scale systems development as well; it recommends developing and implementing features in a prioritized incremental process, with the overall design subject to revision along the way. Although this model does not address the mass-market, and is not entirely feasible for it, it has many features which are applicable.

¹⁶² T. Gilb, Principles of Software Engineering Management, (Wokingham, England: Addison-Wesley, 1988), pp. 83-100

E. The Software Arts Model

1. Rationale and Discussion

This is a model that explicitly recognizes the constraints and opportunities in the production of software for the mass-market. It incorporates the market and financial pressures mitigating for numerous rapid releases by pre-planning incremental enhancements to the base product. It also recognizes the possibility of many alternative solutions to a particular problem, and the existence and availability of many components.

The existence of many possible solutions to a given problem has always been present in software development. This is true even in the software engineering of well-defined problems in well-defined technical environments. This solution ambiguity is expanded when the environment is less well-defined.

Internet connectivity has made vast numbers of software components and products immediately available. These pieces can be applied to many software development puzzles to great advantage. This unknown and evolving object library moves the software development process a little further from science and a little closer to art.

When one adds the vagaries of the mass-market, the problem itself becomes less well-defined. There is no specification or single customer to develop for. The developer (or manager) must try to anticipate the market's preferences and reactions. In this scenario we have not a single problem, but a problem set or problem domain. As such, we have a range of solution definitions, each solution definition itself having a range of potential implementation variations. We start with neither a well-defined problem nor a well-defined environment (but likely with budget constraints). The process becomes more like the production of popular music, from composition to performance to recording. To be sure, there are elements of competent engineering required in music production as well as in software production. However the successful process is more one of art than science. For these reasons our model is called Software Arts.

2. Notation

In our model, we will assume that for any given release, there is a set of features F that may potentially be included in that release. These features each have a priority value v , and a development cost x in terms of time. Priority values may range from 0 to h , where h is the highest priority and indicates that the feature is required regardless of other constraints. So we have

$$F = \{f \mid 0 < v(f) \leq h, 0 < x(f)\}$$

And as described above, for each release we have an optimal cycle time C that is a function of the point in the product life cycle t . We can consider that we also have a fixed maximum amount of manpower m (in per-period units, e.g. man-weeks per calendar week) that can be devoted to the project. This maximum may be the manifestation of a budget constraint, or of hiring and training logistics, or of the recognition of diminishing returns in terms of team size. In any case our optimal development effort is bounded by $M = C(t) \cdot m$ units of work.

So for our given release we would like to include the set of features S ($S \subseteq F$), such that we maximize the total priority values of the included features, while still producing the release at the optimal time.

The features that we will include in our release then can be described as the set of features whose combined development time cost is less than or equal to our maximum development effort, and whose combined priority values equal or exceed those of any other set of potential features meeting the first requirement. Alternatively we can write

$$S = \left\{ s \left| \begin{array}{l} s \in F, \\ \sum x(S) \leq M, \end{array} \right. \right\} \text{ and } \neg \exists W = \left\{ w \left| \begin{array}{l} w \in F, \\ \sum x(W) \leq M, \\ \sum v(W) > \sum v(S) \end{array} \right. \right\}$$

We should also include the restriction that we must include all features with a priority value of h . This is especially relevant for the first release where the optimal cycle time is zero (we would like to have our product to market immediately) but of course there will be a number of features that must be present for the product to be successful. We can make just a slight modification to the above to incorporate this requirement. We will call the set of all required features $H = \{f | v(f) = h\}$. We can then modify the meaning of M in our above notation to be

$$M = \left\{ \begin{array}{l} C(t) \cdot m - \sum x(H) \dots \text{if} \dots C(t) \cdot m > \sum x(H) \\ 0 \dots \dots \dots \text{otherwise} \end{array} \right\}$$

And using our revised definition of M , our definition of S above holds. So we have

Features included in release: $H \cup S$

This will guarantee inclusion of all critical features into our release at hand. The only case in which we might exceed our optimal time period, then, would be if the critical features alone exceeded it. In this case, we would have the $M=0$ case above. Any features with a priority of less than critical would not be included in that case, and will be subject to the constrained maximization calculation in any case.

3. Development Steps

The Software Arts model consists of the following steps:

1. Specification

This is a high-level specification of the product/project under consideration. It includes overall motivation and goals with reference to marketing and competition. It may be produced by a developer in conjunction with management, or by management via market research and/or strategic vision. This step is only present in the first release.

2. Analysis

This corresponds to the usual analysis phase in software engineering, with a few modifications. First, features of the product are prioritized, including the identification of those that are critical. Second, optimal release cycle constraints will be indicated (e.g. we need it by year-end, or this release cycle should be six months). Third, the level of detail is less than that described for the analysis phase in traditional software engineering texts.

3. Preliminary descriptions of alternative(s)

In this phase, several research team members investigate the problem and independently propose design alternatives. Through different technical approaches, each will be attempting to solve for S in our notation above, while maximizing $\sum v(S)$ and remaining within the optimal cycle schedule. Investigations may include web searches of freeware, shareware, and commercial software for potential components and a survey of competitive products. Researchers explicitly recognize the prioritized features and schedule constraints. The proposed solutions indicate which features should be accomplished in the initial release, and which will be deferred until later releases. Each analysis also indicates what existing components/products are to be used and contains a tentative manpower requirements/schedule section.

4. Selection of alternative(s) to pursue further

Alternatives are reviewed and discussed by the research team members and management. This will serve to update all researchers and management on the competitive landscape and on new developments and ideas. One or more solutions, or possibly a hybrid solution, will be chosen to investigate further.

5. Detailed research of selected alternative(s)

This is meant to insure that the preliminary investigations were correct in their assumptions and to refine the development time cost estimates. There should be a confirmation made as to which features are proposed to be included in the release, and which may be implemented in future releases. Those features deferred to future releases remain in consideration in the design, as the "hooks" will be designed into the current version to facilitate additional features. There should also be a refinement of the personnel, component, and equipment requirements.

6. Approval/assignment

This is the commit point for development. Specification and performance requirements will be defined. During this step, an implementation schedule (e.g. PERT chart) will be developed and personnel will be assigned. The target platform(s) will be defined, as well as planned and deferred future ports. The development environments will also be defined and made available.

7. Detailed design, and development of test specifications

Detailed design is to the module level. In the case that an object oriented language is being used, this phase corresponds almost directly to textbook OOD, and includes all the interface standardization. Additional consideration is given to ensuring that deferred features can be easily implemented in future releases. The necessary hooks are inserted and documented. Test specifications are developed concurrently. For most mass-market software products, a multiple language feature should likely be designed into the system. The details would be addressed in this step of the first release.

8. Coding of system & test programs

The system will be coded as will the test programs. A separate QA (Quality Assurance) group is given the test specifications and programs to use in testing releases. Revisions to the design documentation and the schedule are made as soon as they become necessary. Documentation is produced concurrently with coding, is kept in a common release control system, and is reviewed by QA.

9. Beta Pre-Release of the system

The system is made available for unsupported download via the web. This "beta" release will serve to have users perform additional testing, giving the software a thorough market-test before the official release. Developers handle serious bug reports immediately, and record less-serious bugs for future disposition in conjunction with QA. At this same time, help desk personnel are trained in preparation for the official release. If additional language versions of the product are to be available, user message translation would be done at this point.

10. Official Release, and Iteration of steps 2-9 for subsequent releases

For each release, there will likely have been features and fixes deferred. These will be considered in subsequent releases, along with additional ideas that may have come from customer feedback or new products on the market. During subsequent releases, base documentation will be maintained, noting modifications and additions for each release. Additional documentation on outstanding issues and bugs will also be kept in the release control system by QA.

As the foregoing suggests, QA and documentation are of course integral parts of the development process. The earlier discussion of schedule constraint precedence over total quality is not meant to slight the QA process, or to diminish the emphasis on quality.

Also, the model of costless delivery via the web is assumed. Furthermore, the model recognizes the prioritization of features, a planned series of releases, and the existence of schedule and budget constraints. More importantly from a pure software engineering viewpoint, the possibility of many alternatives and the existence of large chunks of existing components is recognized. Step three describes a number of research team members investigating alternatives. This is important because no matter how capable and experienced an individual developer is, it is likely that he or she has particular preferences and tendencies which are not optimal in all cases. Moreover, no developer can be aware of all that is currently available given the proliferation of freeware and shareware and the exceedingly rapid pace of commercial software release.

In recognition of these concerns, the Software Arts model primarily differs in expanding a single analysis phase to five phases: steps two through six. As has been noted many times before, the analysis is possibly the most important phase. It is also the proper place for the majority of the artistry. Leaving the art element to the design and coding phases, however satisfying to the developers, is more like improvisation and is an unmanageable process.

F. Summary and Conclusions

We have considered some of the unique aspects of software markets first from an economic perspective. We have seen the huge advantages to market size in terms of network externalities such as ease of interoperability, a greatly expanded set of compatible products, and reduced risk of unsupported obsolescence. These all serve to make the product much more valuable to the user. We have also seen the minimal costs of increased market size given zero marginal costs and costless delivery.

We have also noted and explained the rapid coalescence around standards in software markets. Forces that contribute to this process include the benefits of the network externalities mentioned above, as well as the uncertainty of independent evaluations of complex software products. These forces lend themselves to the creation of information cascades.

We took all these factors into account in building a unique economic model of a software product's lifecycle. We used a standard normal distribution curve to reflect clustering of agents purchase decisions in information cascades, for our market-segment lifecycle function. We also dynamically affected our demand curve by the existing market size to reflect network externalities.

The results gave us some insights in to optimal pricing, as well as into the importance of the timing of product releases; especially the first one.

Beyond our economic model, we also discussed other issues impacting software firms' profitability. These included an expansion on issues behind our economic model, such as the advantages to being first to market. We also discussed the pre-announcement of software product features. This is, of course, a common industry practice although it is criticized by many. We noted that it can be effective if the firm has proven itself credible in the past. In fact we recommend feature pre-announcement. The deferment of features can allow a firm to meet optimal timing or budget constraints. Nevertheless, it is important that existing customers know that a feature is in the pipeline, both for their upgrade planning decisions and to possibly dissuade them from switching to a competing product.

We also noted that while technical support is sometimes seen as a liability, it can be a profit opportunity. Within delicate constraints, it can even benefit from the tradeoff between schedule and total quality assurance.

Finally we brought all these issues to bear on the software development process. We reviewed a number of existing development models and noted some of the problems that we found with these models in the context of the requirements that we developed. Foremost among the shortcomings that we saw was the lack of explicit recognition of the process of multiple releases. Also lacking is the notion of schedule or budget constraints, and of the fact that all potential features do not necessarily have equal priority.

We then discussed further our views on how the current software development environment calls for a modified development paradigm. In addition to the economic considerations already reviewed, we noted the existence of vast bodies of components available as freeware or shareware. If nothing else these make excellent development templates. We also noted that the pace of change, in terms of the technical environment, has drastically increased. These factors cause us to suggest that the analysis phase in software development must be reconsidered. In fact we suggest that it should be expanded into a number of discrete steps. Furthermore, some of these steps should be done independently and simultaneously by multiple individuals or teams.

We also presented a novel and somewhat formal notation of the description and selection of potential features for inclusion in given software releases. This is in light of feature prioritization and schedule and budget constraints. We described potential features as having both a benefit value and a time cost, and described our constrained maximization problem of feature selection.

In the last section, we presented our unique software development model. This model is meant to incorporate all of our foregoing analysis to the benefit of both the software firm and its developers. The model represents a departure from existing models, but not a radical departure as analysis, design, coding, and testing are of course always required. The most significant innovations in our model include the expansion of the analysis phase into a number of distinct steps and the explicit recognition of prioritization and schedule considerations.

The software development environment has most certainly evolved in recent years, and the process should evolve likewise. By recognizing the underlying economics and the changed landscape we have taken a step in that direction.

IV. Conclusion

The goal of this dissertation has been to bring economic and business analysis to bear on emerging and evolving computer science issues. The development in the field of computer science with the biggest impact in recent years has been the growth of the inter-computer communications through the Internet. The reasons for the impact are many. Of course distributed processing and parallel processing have long studied issues of interconnectivity. The difference now is that the interconnectivity is ubiquitous and includes many autonomous and anonymous agents.

The recent development of the Internet, and its even more recent transition from government sponsorship, open questions of efficiency that are yet unanswered. This research has attempted to shed some light on the economic theory underlying shared resources such as the Internet, and on how a pricing scheme might be devised and implemented to increase efficiency. The most difficult issues involve not so much what would be economically efficient, but rather how an economically efficient scheme could be efficiently implemented, from an applied computer science viewpoint, into the current structure of the Internet. Business issues must also be considered. In practice, it is likely that service providers will try various pricing schemes in the future. The goal of this research has been to suggest schemes that are efficient in theory, practice and implementation, and would benefit both providers and users.

The evolution of the Computer market and the Internet has also created opportunities and changed the landscape in other areas. One of these is software development. This research has argued that software engineering methodologies must evolve to meet these new challenges and opportunities. No longer can software development be thought of as process bounded and defined solely by engineering. Rather it must be shaped by economic and market forces.

Accordingly, in the design of software development processes must start with the goal of the software development firm to maximize profits. The implications involve not only pricing issues, but

the timing of releases as well. In fact timing is critical to profitability. This research has therefore proposed that instead of the final specifications of a software product being fixed and the schedule made to suit, the schedule is the element that should be fixed and the included software features adjusted to suit.

In addition, the growth of the computer market and greatly increased computer interconnectivity has resulted in new challenges and opportunities in software design and analysis. The traditional analysis step in software engineering was a much more bounded problem than that currently faced by analysts. Again, this mitigates for changes to the accepted steps in the software development process. This research has suggested a development model that takes these issues into consideration.

The theme throughout has been the use of economic analysis to redesign computer science processes to increase efficiency. It is an approach that deserves consideration in many other areas of computer science and engineering as well as those considered herein.

REFERENCES

V. References

- Baker, Fred (ed.). "Requirements for IP Version 4 Routers". RFC 1812, IETF Network Working Group. <<http://ds.internic.net/rfc/rfc1812.txt>>. June 1995.
- Bell Atlantic. "Bell Atlantic and NYNEX Agree to Merger of Equals ". <<http://www.bell-atl.com/nynex/>>. 22 April 1996.
- Bidwell, M., B. Wang, and D. Zona. "Analysis of Asymmetric Demand Response to Price Changes". *Journal of Regulatory Economics* 8 (1995):285-298.
- Bikchandani, Hirshleifer, and Welch. "A Theory of Fads, Fashion, Custom, and Cultural Change as Informational Cascades". *Journal of Political Economy* 100(5) (1992): 992-1026.
- Boehm, Barry. "A Spiral Model of Software Development and Enhancement". *Computer*. v.21. pp. 61-72. May 1988
- Boehm, Barry. *Software Engineering Economics*. Englewood Cliffs, NJ: Prentice-Hall. 1981.
- Bohn, R., H. Braun, S. Wolff, and K. Claffy.. "Mitigating the coming Internet crunch: multiple service levels via Precedence". <<ftp://ftp.sdsc.edu/pub/sdsc/anr/papers/precedence.ps.Z>>. March 1994.
- Braden, R. [ed.], L. Zhang, S. Berson, S. Herzog, and S. Jamin. "Resource Reservation Protocol (RSVP) – Version 1 Functional Specifications". RFC 2205, IETF Network Working Group. <<http://ds.internic.net/rfc/rfc2205.txt>>. September 1997.
- Braden, R., D. Clark, and S. Shenker. "Integrated Services in the Internet Architecture: an Overview". RFC 1633, IETF Network Working Group. <<http://ds.internic.net/rfc/rfc1633.txt>>. June 1994.
- Braun, Hans-Werner. "Models of Policy Based Routing". RFC 1104, IETF Network Working Group. <<http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1104.txt>>. June 1989.
- Brownlee, N. "Traffic Flow Measurement: Experiences with NeTraMet". IETF Internet Draft. <<ftp://ietf.org/internet-drafts/draft-ietf-rtfm-acct-experiences-01.txt>>. August 1996.
- Bye, R. "Composite Demand and Joint Supply in Relation to Public Utility Rates". *Quarterly Journal of Economics* 44 (November 1929): 40-62.
- Bye, R. "The Nature of Fundamental Elements of Costs". *Quarterly Journal of Economics* 41 (November 1926): 30-63.
- CAIDA. Links to NAP and NSP statistics. <<http://www.caida.org/INFO/>> 1997.
- Cascade Communications Corporation. "Cascade Reaffirms Frame Relay Leadership Position With Priority Frame™; Industry's First Quality of Service Technology For Frame Relay". <<http://www.casc.com/company/press/pfrr4.html>>. 20 January 1997.
- Cerf, Vinton and the Computing Research Association. "Computer Networking: Global Infrastructure for the 21st Century". <<http://www.cs.washington.edu/homes/lazowska/cra/networks.html>>. 1995.
- Cerf, Vinton as told to Bernard Aboba. "How the Internet Came to Be". <ftp://ftp.isoc.org/internet/history/cerf_Internet.txt>. 1993.

- Cerf, Vinton. "A Brief History of the Internet and Related Networks".
<ftp://ftp.isoc.org/internet/history/unknown_brief.txt>. undated, accessed March 1997.
- Clark, D. "Policy Routing in Internet Protocols". RFC 1102, IETF Network Working Group.
<<http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1102.txt>>. May 1989.
- Cocchi, R., S. Shenker, E. Estrin, and L. Zhang.. "Pricing in Computer Networks: Motivation, Formulation, and Example". <<ftp://ftp.parc.xerox.com/pub/net-research/pricing2.ps.Z>>. November 1993.
- Commercial Internet Exchange Association. "CIX members".
<<http://www.cix.org/CIXInfo/members.html>>
- Cracknell, D. and M. Knott. "The Measurement of Price Elasticities - the BT Experience". The International Journal of Forecasting 11 (1995) 321-329.
- Crew, M. and Kleindorfer, P. The Economics of Public Utility Regulation. Cambridge, MA: MIT Press. 1986.
- Crew, M., C. Fernando, and P. Kleindorfer. "The Theory of Peak-Load Pricing: A Survey". Journal of Regulatory Economics 8 (1995): 215-48.
- Deering, S. "Internet Protocol, Version 6 (IPv6) Specification". RFC 1883, IETF Network Working Group. <<http://ds.internic.net/rfc/rfc1883.txt>>. December 1995.
- Deering, S. and R. Hinden. "Internet Protocol Version 6 (IPv6) Specification". IETF IPNG Working Group. <<ftp://ietf.org/internet-drafts/draft-ietf-ipngwg-ipv6-spec-v2-01.txt>>. November 1997.
- Delgrossi, L. and L. Berger. "Internet Stream Protocol Version 2 (ST2)". RFC 1819, IETF Network Working Group. <<http://ds.internic.net/rfc/rfc1819.txt>>. August 1995.
- Economides, N. and C. Himmelberg. "Critical Mass and Network Size with Application to the US FAX Market". Working Paper no. EC-95-11. Stern School of Business, N.Y.U. (1995).
<<http://raven.stern.nyu.edu/networks/95-11.ps>>
- Edell, R., McKeown, N., Varaiya, P.. "Billing Users and Pricing for TCP".
<<http://paleale.eecs.berkeley.edu/~edell/papers/Billing/article.ps>>. April 1995.
- Estrin, D., T.Li, Y. Rekhter, K. Varachan, D. Zappala. "Source Demand Routing: Packet Format and Forwarding Specification (Version 1)". RFC 1940, IETF Network Working Group.
<<http://ds.internic.net/rfc/rfc1940.txt>>. May 1996.
- Farrell, J. and G. Saloner. "Standardization, Compatibility, and Innovation". Rand Journal of Economics. 16(1) (1985): 70-83.
- Farrell, J. and G. Saloner. "Installed Base and Compatibility: Innovations, Product Pre-announcements, and Predation". American Economic Review (76) (1986): 940-55.
- Fielding, R. "Hypertext Transfer Protocol – HTTP/1.1". RFC 2068, IETF Network Working Group.
<<http://ds.internic.net/rfc/rfc2068.txt>>. January 1997.
- Forgie, J.. "ST - A Proposed Internet Stream Protocol". IEN119. MIT Lincoln Lab. September 1979.
- Froot, Scharfstein, and Stein. "Herd on the Street: Informational Inefficiencies in a Market with Short-Term Speculation". Journal of Finance 47(Sept 1992): 1461-1484.

- Gilb, T. *Principles of Software Engineering Management*. Wokingham, England: Addison-Wesley, 1988.
- Gul and Lundholm. "Endogenous Timing and the Clustering of Agents' Decisions". *Journal of Political Economy* 103(5) (1995).
- Gupta, A., D. Stahl, and A. Whinston. "An Economic Approach to Networked Computing with Priority Classes". <<http://cism.bus.utexas.edu/alok/nprice/netprice.html>>. December 1994.
- Gupta, A., D. Stahl, and A. Whinston. "Managing the Internet as an Economic System". <<http://cism.bus.utexas.edu/alok/nprice/netprice.html>>. July 1994.
- Gupta, A., D. Stahl, and A. Whinston.. "Priority Pricing of Integrated Services Networks". <<http://cism.bus.utexas.edu/>>. 1995.
- Hammond and O'Reilly. *Performance Analysis of Local Computer Networks*. Reading, MA: Addison-Wesley. 1986.
- Handelman, S., N. Brownlee, and G. Ruth. IETF Real Time Flow Measurement Working Group. <<ftp://ietf.org/internet-drafts/draft-ietf-rtfm-new-traffic-flow-00.txt>>. November 1996.
- Hanks, S., T. Li, D. Farinacci, and P. Traina. "Generic Routing Encapsulation over IPv4 Networks". RFC 1702, IETF Network Working Group. October 1994.
- Hanssens, D. and L. Parsons. "Econometric and Time Series Market Response Models". In *OR/MS in Marketing Handbook*, edited by Eliashberg and Lilien. New York: Elsevier. 1994.
- Harrison and Patel. *Performance Modelling of Communication Networks and Computer Architectures*. Reading, MA: Addison-Wesley. 1992.
- IETF. "IETF Home Page". <<http://www.ietf.cnri.reston.va.us/home.html>>. undated, accessed March 1996 -1997
- Information Sciences Institute, University of Southern California. "INTERNET PROTOCOL, DARPA INTERNET PROGRAM, PROTOCOL SPECIFICATION". <<http://ds.internic.net/rfc/rfc791.txt>>. September 1981
- Internet Society. "International Connectivity". <<http://www.isoc.org/images/mapv15.gif>>. June 1996.
- Kam, Phil . (©1992) KA9Q package. <<http://www.qualcomm.com/people/pkam/tcpip.html>>
- Katz, M. and C. Shapiro. "Network Externalities, Competition, and Compatibility". *American Economic Review* 75(3) (1985): 424-40.
- Katz, M. and C. Shapiro. "Technology Adoption in the Presence of Network Externalities". *Journal of Political Economy*. 94(4) (1986): 822-41.
- King, S., R. Fax, D. Haskin, W. Ling, T. Meehan, and R. Fink. "The Case for IPv6". IETF Internet Architecture Board. <<ftp://ietf.org/internet-drafts/draft-ietf-iab-case-for-ipv6-00.txt>>. November 1997.
- Kleinrock, Leonard. *Queueing Systems*. vol. 1. New York: John Wiley & Sons. 1975.

- Lash, Alex. "Pac Bell chimes in with Net service". *c/net*. May 28, 1996.
<<http://www.news.com/News/Item/0,4,1405,00.html>>.
- Mackie-Mason, Jeff, L. Murphy, and J. Murphy. "The Role of Responsive Pricing in the Internet".
<<http://www.spp.umich.edu/spp/papers/jmm/respons.ps.Z>>. August 1995.
- Mackie-Mason, Jeff and Hal Varian. "Economic FAQs About the Internet".
<http://www.spp.umich.edu/spp/papers/jmm/Economic_FAQs.ps.Z>. June 1995.
- Mackie-Mason, Jeff and Hal Varian. "Pricing Congestible Network Resources".
<http://www.spp.umich.edu/spp/papers/jmm/Pricing_Congestible/ieee.ps.Z>. revised November 1994.
- Mackie-Mason, Jeff and Hal Varian. "Pricing the Internet".
<http://www.spp.umich.edu/ipp/papers/info-nets/Pricing_Internet/Pricing_the_Internet.ps.Z>,
undated, accessed 1997.
- Mackie-Mason, Jeff and Hal Varian. "Some Economics of the Internet".
<http://www.spp.umich.edu/spp/papers/jmm/Economics_of_Internet.ps.Z>. revised February 1994.
- Mackie-Mason, Jeff. "Telecomm Information Resources On the Internet".
<<http://www.spp.umich.edu/telecom/telecom-info.html>>, undated, accessed 1997.
- Mankin, A. [ed.], F. Baker, B. Braden, S. Bradner, M. O'Dell, A. Romanow, A. Weinrib, and L. Zhang.
"Resource Reservation Protocol (RSVP), Version 1 Applicability Statement, Some Guidelines on
Deployment". RFC 2208, IETF Network Working Group. <<http://ds.internic.net/rfc/rfc2208.txt>>.
September 1997.
- Massachusetts Institute of Technology Research Program on Communications Policy.
<<http://far.mit.edu/Workshops/dist.html>>
- MCI Corp.. "CISCO AND MCI LEAD THE WAY IN BRINGING "PREMIUM" GRADE SERVICE TO
THE INTERNET". <<http://www.mci.com/aboutmci/news/indexnewsrefresh.shtml>>. 25 March 1997.
- Mecklermedia. "The List (Internet Service Providers)". <<http://thelist.iworld.com/>>
- Merit Network Inc.. "Merit Retires NSFNET Backbone Service". *MichNet News*. Vol 9(2).
<<http://www.michnet.net/michnet/michnet.news/mnn.1995-02/nsfnet.html>>. Spring 1995.
- Merit Network Inc.. "NSFNET: Transition to T-3".
<<http://www.merit.edu/nsfnet/final.report/transition.html>>. undated, accessed March 1996,1997.
- Merit Network. "NSFNET: Bringing the World of Ideas Together".
<<http://www.merit.edu/nsfnet/nsfnet.overview>>. April 1992.
- MFS Communications Company Inc.. "MFS and UUNET Announce Merger Agreement to Form
Premier Internet Business Communications Company".
<<http://www.mfsdatanet.com/mfs/news/Press/1996/Apr/30Apr96.html>>. 30 April, 1996.
- Mischler, David. (©1995,1996). "IPRoute". V0.9. <<http://www.mischler.com/iproute/>>. 14 February 1997.

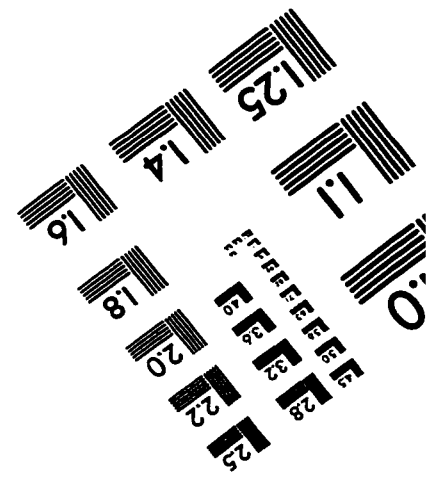
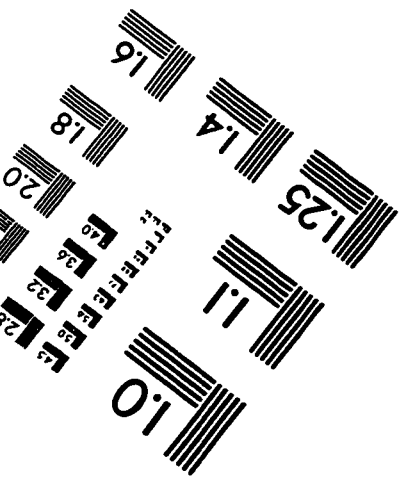
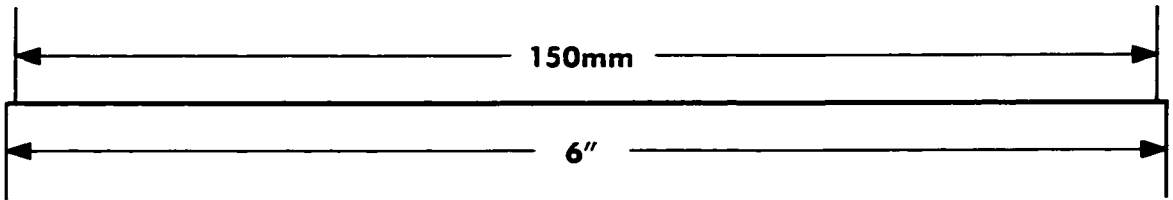
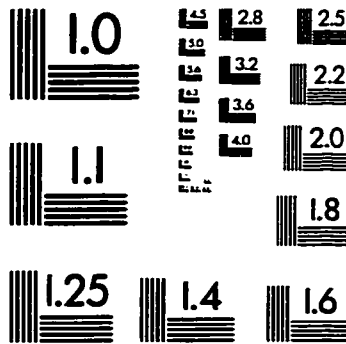
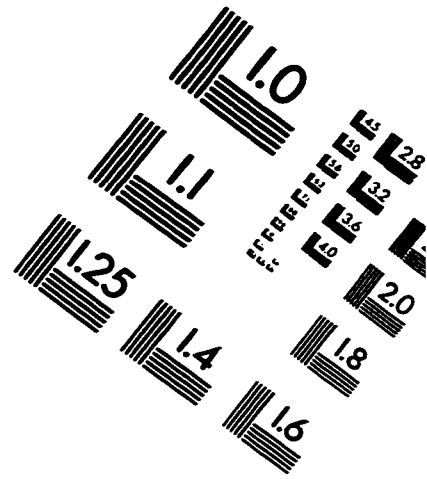
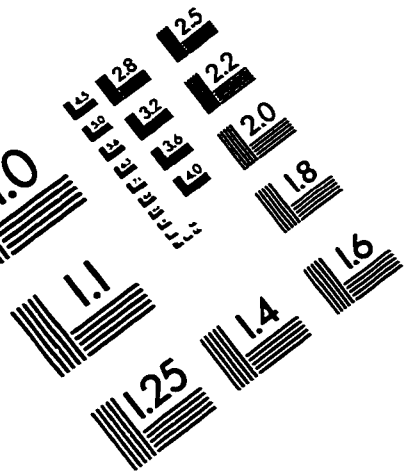
- Mitchell and Vogelsang. *Telecommunications Pricing: Theory & Practice*. Cambridge: Cambridge University Press. 1991.
- National Laboratory for Applied Network Research. "Internet Information Presentation". <<http://www.nlanr.net/INFO/>>
- Netscape Communications Corporation. US Securities and Exchange Commission Form 10-K. <<http://www.netscape.com/comprod/investor/10kpart1a.html>>, for the fiscal year ended Dec. 31, 1995.
- Object Analysis and Design. Framingham, MA: Object Management Group. 1992.
- Oi, W. "A Disneyland Dilemma: Two-Part Tariffs for a Mickey Mouse Monopoly". *Quarterly Journal of Economics*. 85(1: February 1971): 77-96.
- Oracle Corp. <<http://www.oracle.com/corporate/html/earningstable2.html>> 1997.
- Orlean, A. "Bayesian Interactions and Collective Dynamics of Opinion: Herd Behavior and Mimetic Contagion". *Journal of Economic Behavior and Organization* 28 (1995).
- Panzar, J., and D. Sibley. "Public Utility Pricing under Risk: The Case of Self-Rationing". *American Economic Review* 68(5) (1978): 888-95.
- Perkins, C. "IP Encapsulation within IP". RFC 2003, IETF Network Working Group. <<http://ds.internic.net/rfc/rfc2003.txt>>. October 1996.
- Perkins, C. "Minimal Encapsulation within IP". RFC 2004, IETF Network Working Group. <<http://ds.internic.net/rfc/rfc2004.txt>>. October 1996.
- Postel, Jon. "DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION", RFC 791 <<http://ds.internic.net/rfc/rfc791.txt>>. September 1981.
- Rekhter, Jacob. "EGP and Policy Based Routing in the New NSRNET Backbone". RFC 1092, IETF Network Working Group. <<http://info.internet.isi.edu:80/in-notes/rfc/files/rfc1092.txt>>. February 1989.
- Royce, W. "Managing the Development of Large Software Systems: Concepts and Techniques". *Proceedings of WestCon*, August 1970.
- Schach, Steven. *Software Engineering*. Homewood, IL: Aksen Associates. 1990.
- Scotchmer, S. "Two-tier pricing of shared facilities in a free-entry equilibrium". *Rand Journal of Economics* 16(4) (1985): 456-472.
- Segal, Ben. "A Short History of Internet Protocols at CERN". <<http://wwwcn.cern.ch/pdp/ns/ben/TCPHIST.html>>. April 1995.
- Shenker, S., C. Partridge, and R. Guerin. "Specification of Guaranteed Quality of Service". IETF Network Working Group. <<http://ds.internic.net/rfc/rfc2212.txt>>. September 1997.
- Shenker, Scott. "Service Models and Pricing Policies for an Integrated Services Internet". <[ftp://ftp.parc.xerox.com/pub/net-research/policy.ps.Z](http://ftp.parc.xerox.com/pub/net-research/policy.ps.Z)>. undated, accessed March 1996.
- Spulber, D. "Capacity-Contingent Nonlinear Pricing by Regulated Firms". *Journal of Regulatory Economics* 4(4) (1992): 299-320.

- Spulber, D. "Optimal Nonlinear Pricing and Contingent Contracts". *International Economic Review* 33(4) (1992): 747-72.
- Sterling, Bruce. "Short History of the Internet". <ftp://ftp.isoc.org/internet/history/sterling_Internet.txt>. February 1993.
- Stickel, S. "Reputation and Performance among Security Analysts". *Journal of Finance* 47(December 1992): 1811-1836.
- Swann, P., and J. Gill. *Corporate Vision and Rapid Technological Change*. London: Routledge. 1993.
- Taylor, Lester. *Telecommunications Demand: A Survey and Critique*. Cambridge, MA: Ballinger Publishing Company. 1980.
- Topolcic, D. "Experimental Internet Stream Protocol, Version 2 (ST-II)". RFC 1190, IETF Network Working Group. <<http://ds.internic.net/rfc/rfc1190.txt>>. October 1990.
- Ubois, Jeff. "Peer Pressure". *Internet World*, vol. 7(8) August 1996. <<http://www.iw.com/1996/08/peer.html>>
- Varian, Hal. "The Information Economy". <<http://www.sims.berkeley.edu/resources/infoecon>>
- Vickrey, W. "Responsive Pricing of Public Utility Services". *Bell Journal of Economics* 2(1:Spring 1971): 337-46.
- Villasenor, Anthony. "Survey of International Internet Connectivity". <<http://nic.nasa.gov/ni/survey/survey.html>>.
- Whitefield, Mimi. "BellSouth will offer Internet services, too". *Miami Herald*, 4 April 96 p. C1.
- Whitten, N. *Managing Software Development Projects*. New York: John Wiley & Sons, 1990.
- Wingfield, Nick. "Bell Atlantic gets into Net market". *c/net*, 10 April 1996. <<http://www.news.com/News/Item/0,4,1075,00.html>>.
- Wroclawski, J. "Specification of the Controlled-Load Network Element Service". RFC 2211, IETF Network Working Group. <<http://ds.internic.net/rfc/rfc2211.txt>>. September 1997.
- Wroclawski, J. "The Use of RSVP with IETF Integrated Services". RFC 2210, IETF Network Working Group. <<http://ds.internic.net/rfc/rfc2210.txt>>. September 1997.
- Yourdon, Edward. *Object Oriented Systems Design, An Integrated Approach*. Englewood Cliffs, NJ: Prentice-Hall, 1994.
- Yourdon, Edward. *Rise and Resurrection of the American Programmer*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

VITA

November 28, 1954	Chicago, Illinois
1975-77	Senior Programmer/Analyst Alachua County Data Processing Department Gainesville, Florida
1978-81	Computer Consultant Miami, Florida
1981-1984	Senior Software Engineer Harris Corporation, Controls Division Melbourne, Florida
1984	B.A., Anthropology University of Florida Gainesville, Florida
1986	M.B.A., International Business University of Florida Gainesville, Florida
1987-1992	Computer Consultant Miami, Florida
1993	M.A., Economics Florida International University Miami, Florida
1995	M.S., Computer Science Florida International University Miami, Florida
1995-1996	General Manager High Performance Database Research Center School of Computer Science Florida International University Miami, Florida
1997-1998	President Palm Computers Boca Raton, Florida
1998	Ph.D., Computer Science Florida International University Miami, Florida

IMAGE EVALUATION TEST TARGET (QA-3)



APPLIED IMAGE, Inc
 1653 East Main Street
 Rochester, NY 14609 USA
 Phone: 716/482-0300
 Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved