# Where Cash and Technology Meet

## A history of Electronic Commerce

By
Stuart Taylor

Over the past quarter century, electronic commerce has helped revolutionized the way companies do business. It has done this several ways: One, it has assisted in internal communication and organization in business.[1] By storing information in databases that can be accessed and changed from anywhere in the world, businesses can reduce time spent on paper work and inventory. New methods of communications mean that people can share ideas and transfer data much faster and more effectively than before. Because this subset of electronic commerce is internalized, I will not cover it much in my paper. Business intranets do not suffer from language problems like open networks do. A completely contained network does not have to worry about peer-to-peer communications between partners that might not be active %100 of the time. As long as the code is written well, a private contained network should not suffer from an inability to communicate with its working sub-systems. Intranets should be at least mentioned, however, in order to showcase the differences between them and Internets.

The second way electronic commerce has changed business is by making business-to-business communications and sales much easier.[2] By automating the orders and payments for goods and services, businesses can streamline and eliminate paper work. For example, a company can automatically order products only when it needs them. By connecting a computer to the database that keeps track of inventory, the computer can automatically order and pay for whatever items a business is running low

[1] Riggins, Fredrick J. and Rhee, Hyeun-Suk (Sue). "Toward a Unified View of Electronic Commerce" *Communications of the ACM* Vol. 41, Iss 10, October, 1998, p. 91
[2] Riggins, 91

on. The automatic payment feature is based on technology known as electronic funds transfer, or EFT. EFT is the basis for electronic commerce, as it lets people and businesses transfer funds from one account to another instantly from any place with a connection to the relevant network.[3]

EFT has also been influential in a third area of electronic commerce, which is customer to business relations.[4] Electronic funds transfers, along with the Internet, have changed the way many people shop. While Internet commerce has been over-hyped at times, it has still created new markets and new modes of business that previously did not exist. Thanks to the Internet, customers can now view products first hand, and using electronic fund transfers, can purchase them as well.

Electronic funds transfers use various types of communication lines to send balance request and adjustment messages to a banks database, then process the banks return signal. Because banks store their customers' accounts in the form data in a database, electronic funds transfers can easily modify them according to how much was withdrawn or deposited. One of the oldest and most common types of electronic fund transfer systems are automatic teller machines.[5] These devices provide cash on demand, subtracting the amount dispensed from the user's bank account. The procedures developed for these devices are now used to conduct electronic and Internet commerce.

Another thing business must trust about electronic commerce, and the most important component of electronic funds transfers, is their ability to deliver information to its intended target. Electronic information transfer is in many ways the basis for electronic commerce. Information about goods and products go one way, and

[3] *Computer Encyclopedia*, "Electronic Commerce" 628
[4] Riggins, 91
[5] "Electronic Commerce" 635

information about bank account numbers and payments go the other way. The information travels to its destination by way of a communication network, usually the Internet.

The Internet is essentially a network of networks made up of networks.[6] If someone takes two independent networks and connects them via a router, they in effect become one network. The Internet works along these lines, making growth very easy. The only real regulation is the Internet Corporation for Assigned Names and Numbers, which keeps track of IP addresses and numbers.[7] The Internet shows this flexibility through its history. Originally, the technology that powers the Internet, called packet switching, was developed in the early 1960's to help communication networks survive nuclear attack.[8] Unlike traditional hub-and-spoke style networks, where information is routed through a central hub, information via multiple routers that are organized in a web. If one router goes down, say due to nuclear strikes, one or two packets will be lost while the other will use a different route that does not use that router.[9] In comparison, if a central hub were to go down in a hub-and-spoke system, the entire system would go down. As a result of being built on packet-switching technology, the Internet can keep functioning if routers go down, although at a reduced rate.

One problem this creates for electronic fund transfers is security. Because of the diffuse nature of the Internet, a user does not have any idea what routers his information in traveling through. It would be very easy for the owner of that router to access and

---

[6] Tannenbaum, Andrew S. "Computer Networks," 4th Ed. (Upper Saddle River, New Jersey: Prentice Hall, 2003), 58

[7] Internet Corporation for Assigned Names and Numbers, The. "ICANN Information," <http://www.icann.org/general/> (September 12th, 2004).

[8] Meleis, Hanafy. "Toward the Information Network." In "Computer" Vol. 29, Iss.10, October, 1996, 59

[9] Meleis, 59

copy the packets of information going through it. This is fine if the user is sending trivial information, but transmitting personal information like credit card and account numbers without any security is asking for trouble. Consequently, all personal data that a user wants to keep private must be sent encrypted. While electronic commerce is possible without secure encryption, it is not a good idea.

One of the most important requirements the Internet has is that all the separate networks a large network is made up of speak the same language. An analogy would be several people attending a conference, none of whom have a language in common. While it would be possible to hire translators, this would be expensive in terms of both time and money. It would be much easier if everyone spoke the same language, there by allowing quick and easy communication. The lingua franca for the Internet is Transmission Control Protocol/ Internet Protocol (TCP/IP). This protocol controls how and where information is sent along a network. The information itself is usually formatted into hypertext transport protocol form (HTTP) for a user on the World Wide Web, or into other protocols such as file-transfer protocol (FTP).

HTTP, the protocol for the World Wide Web, is something of a special case. It is a standardized, often used protocol that allows a user to transmit, store, or receive normal text, pictures, or hypertext. The information stored in hypertext markup language (HTML) and transported via HTTP is easily formatted and understood by most personal computers on the Internet. Its presence has allowed the third area of electronic commerce, the business-to-customer transfers, to flourish. Hyper text is text in an electronic document that when activated brings up another document. For example, a document about automobiles may imbed a link in the word ‘engine’ that brings up a page of information about internal combustion engines.  This is very useful as it allows readers who don't want to read about the way a gear box works skip that section while still giving that information to those who are interested.  In commercial websites, it lets users easily go to the products they are interested in.

The question that remains is how these various protocols and standards were developed.  The standards that define how the Internet works were not developed in the traditional way.  Unlike most software and hardware today, the protocols that run the Internet were not developed for profit.  They were instead developed and distributed for free by groups of individuals who worked for the government or many cases for free. Developing and freely licensing these protocols and programs for the common public good is often labeled as 'open source' development.  Open source means that a project is not developed by a company, but rather by a public effort, usually organized over the Web.

The way an open source project works is that its developers release the source code to the general public to run and critique.[10]  Based on the feedback they get, the developers then modify their project accordingly.  The advantage of this system is that there are hundreds and even thousands of people looking through the code and testing it for bugs.[11]  The developers themselves are programmers who have a problem they feel is not addresses by available software, and wish to create a program that will solve this problem.[12]  In an open source project, the developers are not the only one writing the code.  Hackers who have a general interest in the project will also contribute their

[10] Raymond, Eric S. "The Cathedral and The Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary."  (Cambridge: O'Rielly, 2001), 29
[11] Raymond, 30
[12] Raymond, 23

solutions to existing problems. Besides a need for software, hackers contribute to open source projects out of pleasure for a job well done, as well as the appreciation of their peers.[13] Many people contribute to projects specifically for the so-called 'ego-boost' that comes from their follow programmers in response to good code.[14]

It is important to note that while these projects are called 'open source,[15]' open source actually refers to just the source code and not the licensing. When Richard Stallman created the Free Software Foundation, he was responding not to idea of selling software, but rather to the idea of selling software the user is not able to modify.[16] He was mad that a program he had purchased had bugs in it that he was not able to fix, as he did not have the source code. He did not object to the idea of programmers benefiting from the products of their work.[17] Open source in its strictest sense means that a program comes with its source code so the purchaser may modify it to suite his demands. It does not mean that the purchaser of the software has the right to resell that software.

Open license software is software that is free to use in every way. Open licenses very, with the most common being the GNU General Public License and the BSD license template.[18] They generally ask that the developers of the software be credited. While some open licenses may stipulate otherwise, most publicly licensed software is free for any one to use in any manner they see fit. This can even include packaging and selling

---

[13] Raymond, 82
[14] Raymond, 81
[15] Note: Although I distinguish between open source and open license in these paragraphs, I will use the phrase 'open source' throughout the rest of the paper to indicate software that is both open source and open license. Although I think calling programs like Linux 'open source' a slight misnomer, it is the most common way of using that phrase. I suspect that most people do not distinguish between open source and open license because privately licensed software that has an open source is extremely rare.
[16] Castells, Manuel. "The Internet Galaxy: Reflections on the Internet, Business, and Society." (Oxford: Oxford University Press, 2001), 50
[17] Ibid.
[18] Open Source dot Org, "The Approved Licenses," <http://www.opensource.org/licenses/index.php> (December 12th, 2004), Open Source Licenses.

the software for profit.  While selling such a product is difficult, companies such as Red Hat can make money by selling open licensed software that has been modified and that is supplied with tech support.  Because open licenses do not limit the use of code in any way, it is common for developers to use one piece of open source code in another program.  For example, if another program, regardless of its licensing, needed a graphical user interface (GUI), there would nothing wrong with taking the code from an already existing open source GUI.

Open source development has many similarities with the two other types of development prevalent in the Internet, RFC pages and academic journals.  With an open source project, the code is released to the public to be commented on.  An academic journal works similarly, with a scientist, engineer or researcher releasing his or her ideas to the community for feedback.  Much of the early Internet was developed this, as well as much of the early encryption work.  When Ron Rivest, Shamir, and Adleman published their paper about public key encryption, a system that would later help secure the Internet from credit card fraud, it was developed off of a paper submitted a few years earlier by Diffie and Hellman.[19]  The request for comment (RFC) system is also similar to the academic journal system, but is less formal.  In that case, anyone can comment on an RFC if they are interested, and their thoughts will be edited into the document.  Much of the development of TCP/IP was based on RFC's as well as the development of the World Wide Web.

The TCP/IP protocols were first developed under the Advanced Research Projects Administration (ARPA).  The ARPA first created the Arpanet in 1969 as a research

---

[19], Rivest, R.L., Shamir, A. and Adleman, L.  "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems." In *"Communications of the ACM"* Vol 21, No 2, February 1978, 120

network, with routers at few major universities.[20] Over the next several years, the ARPANET grew with new routers and hosts being added constantly. Originally, the technology that powers the ARPANET, called packet switching, was developed in the early 1960's to supposedly help communication networks survive nuclear attack.[21] Unlike traditional hub-and-spoke style networks, where information is routed through a central hub, information is sent via multiple routers that are organized in a web. If one router goes down, say due to nuclear strikes, one or two packets will be lost while the other will use a different route that does not use that router.[22] In comparison, if a central hub were to go down in a hub-and-spoke system, the entire system would go down. In reality, the nuclear attack scenario was a way to justify the idea of a distributed network to the Department of Defense run ARPA. The ARPANET was really a way of letting multiple ARPA offices across the country use a few large and important mainframes.[23] A network based on packet switching technology would be very useful, as new components could be added to a network without having to reconfigure the entire network. It could also deal with data lost due to static, a problem that had plagued early telephone based data transfer techniques.[24] Since distributed networks were solutions to both problems, the Information Processing Techniques Office under ARPA began development on such a system. The result of this was ARPANET, a network based on packet switching technology that could survive both nuclear attack and overcrowded circuits.

[20]Tannenbaum, Andrew S. "Computer Networks," 4th Ed. (Upper Saddle River, New Jersey: Prentice Hall, 2003), 53
[21] Meleis, 59
[22] Meleis, 59
[23] O'Neill, Judy. "The Role of ARPA in the Development of the ARPANET, 1961-1972." *Annals of the History of Computing,* Vol. 17, Num. 4, 77-78.
[24] Ibid, 77.

The early ARPANET was developed along proprietary lines by BBN, or Bolt, Beranek, and Newman Corp., for DARPA.[25] BBN also played other roles in the development of the ARPANET. Ray Tomlinson of BBN also helped write the very first e-mail program, an application that would later serve as a critical development tool in the creation of the Internet.[26] Ray Tomlinson also played a critical part in the creation of TCP/IP when was working for BBN. Along with Peter Kirstien and Vinton Cerf, he created the protocols with funding from DARPA. The ARPANET was also proprietary in the sense that only those universities that had contracts with the Department of Defense (DoD) could use it.[27] The big difference between ARPANET and many other proprietary technologies is that the systems and protocols developed by them were available for anyone to copy.

Proprietary software and hardware is so common it almost doesn't need to be described. It is software or equipment that is privately developed by a corporation, company, or other private entity in order to be sold. The company that developed this equipment holds the patent or copyright to this technology, giving them the rights to decide who can use it and under what conditions. These conditions are usually 'you have to give us money before you can use our software.' The actual development of proprietary technology varies from company to company, but follows the same general pattern. Usually, a few programmers get together to work on an assigned project, and don't release the code until it is done. This is also called the 'Cathedral' method of

[25] Postel, Jon et al. "The Past and Future History of the Internet." *Communications of the ACM*, Vol 40, no 2, February 1997, 103.
[26] Postel, 103
[27] Tannenbaum, 55

development.[28] This is because the development of the code resembles the construction of an old medieval cathedral. A limited number of specialists work under a few expert craftsmen to construct a program from the ground up. The next layer of stones in a cathedral only laid after the previous layer has been completed.

One problem with proprietary software is that it costs money to use. It cost IBM a lot of money to develop SNA, and so IBM charged money for SNA and the equipment that used it in order to recap expenses and make a profit. Open source software is by nature free, although acquiring some form of technical support may cost money. A by-product of the costs of proprietary development is reluctance among managers to take risks when developing new software. This reluctance is also present in consumers who don't want to risk spending money on a protocol or program that will become uselessly obsolete later. Another problem with proprietary development is the possible disconnect between engineering and marketing in many companies. A company may nay develop software that is very useful for a certain situation because the engineering team did not think of that situation. SNA, for example, did not have peer-to-peer features added onto its protocols until much later in its development because the software engineers did not realize companies would want to do this. Open source development avoids these problems. A project that seems very useful will attract the attention of interested developers.

The later ARPANET was developed using much more open source like methods. Specifically, the development of TCP/IP was handed over to the universities that ran the ARPANET. Instead of being developed by a few people at BBN, it was now being

[28] Raymond, 21

developed through the newly created RFC process run by Jon Postel.[29] The RFC's helped the development of TCP/IP in several ways. These documents let everyone who used the ARPANET input on what they thought of TCP/IP, including known bugs, how it could be improved, and what additions were needed. Protocols like File Transfer Protocol (FTP) and Simple Mail Transport Protocol (SMTP)[30] were developed by people working with the RFC process.

The Internet Engineering Task Force (IETF), a non-profit group dedicated to standardizing and improving the Internet, governs the RFC process. The IETF is open to any corporation or individual who is interested or concerned with the evolution of the Internet.[31] The IEFT is organized into a series of work groups organized roughly into areas. These groups, all made up of interested individuals, work on RFC's under the guidance of the Internet Engineering Steering Group (IESG).[32] The IESG does not exert direct control over the IEFT work groups. Instead, it starts and finishes work groups when needed, and ratifies and corrects their output.[33]

In order to submit an RFC to be published, it must first be posted as an Internet document.[34] The members of the IEFT then comment on it, suggesting changes and pointing out errors. A prospective RFC will probably be published as an Internet Document several times before it is submitted to the IESG for final publication. The IESG will probably also request the author make changes to the document. Once they approve, it is submitted to an RFC editor who checks the document for adherence to RFC

---

[29] "Jon Postel Biography," *Postel Center*, < http://www.postel.org/postel.html > (September 12th, 2004).
[30] Meleis, 59
[31] "Internet Engineering Task Force", <http://www.ietf.org/glossary.html (December 9th, 2004)> (September 12th, 2004), Glossary.
[32] Ibid.
[33] Hoffman, P. and Harris, S., "The Tao of IEFT – A Novices Guide to the Internet Engineering Task Force," *The IEFT Education Site*, <http://edu.ietf.org/tao>, (December 9th, 2004)
[34] Ibid.

guidelines.[35] The ISEG may also submit RFC's directly to the RFC editor, although they usually go through IEFT Work Groups.

There are currently six types of RFC's, each of which must go through a slightly different process to be accepted. They are: Proposed Standards, which are Internet Drafts that have been approved as standards; Draft Standards which are Proposed Standards that have been successfully implemented; Internet or Full standards, which have been around for several years, and are critical for the running of the internet; Experimental Protocols, which are ideas that the author feels are interesting, but are in no way implemented yet; Informational Documents which are not standards, but meant to help clarify concepts; and Historic Standards which are standards that are now obsolete.[36] It is important to note that to become a Full Internet Standard, a protocol must first be implemented then run for several years.

All of these standards are developed and approved by rough consensus. This means that before a standard can be approved, everyone must have the opportunity to comment on it. If enough people find serious objections to a new standard, it will not be approved. It is very important to note that while the IEFT is a standards organization, it does not 'police' the Internet.[37] Developers follow the standards laid out by the IEFT because it is far easier than developing their own standards. Because the IEFT is very open in its development process, most developers of Internet software will have some voice in the way an Internet Standard they care about is designed.

While the RFC process was not quite so formalized back during the early days of the Internet, it still functioned much the same way the modern RFC process does now.

---

[35] Ibid
[36] Ibid.
[37] Ibid.

Documents that were proposed as standards were first presented to the general community to be approved or change. If there was any significant objection to the way a protocol was designed, it was changed. While the RFC process is more complicated than back in the day when Jon Postel ran it on his own, it still is based on the same goal of achieving rough consensus among the internet community before publishing it.

While TCP/IP v.4 became the official ARPANET standard by 1983,[38] it was not the only network protocol in existence, nor was it even the only major one. The Digital Equipment Corporations DECnet, IBM's SNA protocol, and the International Standards Organization's Open Systems Interconnect (i.e. the ISO's OSI) were all protocols created in the late 1970's and early 1980's to compete with TCP/IP. DECnet and SNA were both developed under a proprietary model. The OSI protocol was developed in order to be a 'standard' protocol. The committee that ran the ISO wanted to develop an internationally used standard for networks protocols. They viewed TCP/IP as a US DoD protocol that "would not be acceptable in international work."[39]

The standards method of development is third major method of development I will talk about in my paper, specifically the type of standardization used by the International Standards Organization. A government or independent organization to try to standardize a product or technology uses the standards method. Various companies and individuals who wish to see a particular technology standardized in order to increase ease of use usually create independent standards organizations. In the case of OSI, its creator the International Standards Organization is an international group made up of

---

[38] Meleis, 60

[39] Foley, Jerry, and Severance, Charles. "OSI Retrospect and Prospect," *Computer*, (September 1997), 123.

national standards groups dedicated to standardizing a vast number of products.[40] Standards and standards organizations exist to solve a problem outlined above; what do you do when several people want to talk to each other, but no one understands the same language? The solution is to make sure each person has at least one language in common. Standards in the communications world exist to make sure this language exists and is the same everywhere.

While both the ISO and the IEFT are standards organizations, there are significant differences between the ways these two organizations create standards. The ISO is very hierarchical in the way it is organized and in the way it creates standards. The structure of the ISO is made up of representatives from various national standards organization. Each of these national standards organizations has a representative with a predefined number of votes.[41] These representatives then approve of the various standards in a 'formal democratic process' before they are released.[42] The OSI then designs and creates these standards through the 'cathedral' method of development, with the protocol being developed step-by-step from the ground up.[43] The OSI does not actually produce any code implementations. Instead, it specifies the needed protocols and standards and leaves the actual implementation to the developers. As a result, the people who specified the Open Systems Interconnect standard had little communication with the developers who tried to produce the OSI stacks.

In contrast, the RFC process places a great deal of significance in already implemented protocols. Any proposed protocol that wants to be something other than an

[40] Tanenbaum, 74
[41] Hanseth, Ole, et al. "Developing Information Infrastructure: The Tension between Standardization and Flexibility," *Science, Technology, and Human Values*, vol. 21, no. 4 (Autumn, 1996), pg. 413
[42] Ibid.
[43] Ibid.

experimental protocol must first go through several implementations, checked for implementation problems, and then only approved when it is clear that the rest of the Internet community has accepted the standard as well.[44]

The so-called ‘protocol wars’ were resolved by the mid 1990’s with TCP/IP as the clear winner. Although it is clear now that TCP/IP is now the dominant network protocol, it was not so clear during the mid-1980’s. There were many people back then who predicted that TCP/IP would die out. One reason was that other protocols such as SNA were arguably better at internal network functions than TCP/IP was.[45] Another is that protocols like OSI were developed by well-known national and multinational groups that had previously defined other important standards. Anything produced by them would undoubtedly become the new standard for network communications. Many government entities, looking to develop there own Wide Area Networks and Backbones accepted the OSI standard purely because it was put out by the ISO, not realizing that TCP/IP was already widely adopted.

Why did TCP/IP dominate when at one point it was going to ‘die out?’ One reason was that other protocols were not as adaptable as TCP/IP.[46] DECnet for example could only be run on the VAX/VMS operating system that was developed by the Digital Equipment Corporation. In comparison, while TCP/IP was developed for the UNIX operating system, it was later ported over to VAX/VMS.[47] With the choice coming down to either a network protocol that could only talk to computers running VAX/VMS or a protocol that could talk to computers running VAX/VMS *or* UNIX, it was fairly obvious

[44] Ibid.
[45] English, Erin. “TCP/IP gets a face-lift,” *Computer* (October 1995), p. 13
[46] English, 12
[47] Berners-Lee, Tim. “Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by It Inventor,” (San Francisco: HarperCollins, 1999), 19

which protocol to choose. This difference between the two protocols is what made Tim Berners-Lee choose TCP/IP as his platform of choice when he developed the World Wide Web.[48] This may have been what made TCP/IP the clear winner in the protocol wars, and put the final nail in the other protocols coffins, something I will talk about later in more detail. Another reason TCP/IP won is the way in which it was distributed. Because TCP/IP was open source, anyone could use it free of charge. The other proprietary networks charged for the use of their systems. From nearly the beginning, TCP/IP was included free in the UNIX operating system, which was used by most of the computer research community.[49]

Another reason TCP/IP dominated over many of the other network protocols was the way TCP/IP was developed and distributed. Because it was open source, TCP/IP was constantly being changed and undated. Most other proprietary network systems had to under go a slower development process between updates. Further more, when TCP/IP was changed or updated, it was modified in the way its end users needed it to be. This was because it was the end users who were modifying the code, or at least advising the people who were modifying it. As a result, TCP/IP and its sub-protocols such as Simple Mail Forwarding Protocol (SMTP), File Transfer Protocol (FTP), and SNMP (Simple Network Managing Protocol) were able to create and adapt themselves to their users needs.

Probably the biggest reason TCP/IP was able to succeed was that it was able to avoid the classic 'chicken and egg' problems faced by many types of communication systems. The problem is that in order for people to adopt a new standard, they must first

---

[48] Berners-Lee, 19.
[49] Postel, 105.

see that everyone else is using it or will be soon. There is no point in spending time and effort to change all of your computer systems to use a new protocol if no one else is going to use it. All you will have accomplished is wasting energy to speak a language no one else speaks.

This is the trap that OSI fell victim to. OSI took a long time to develop because it was being created by a standards organization. Because the ISO had to please each of the vendors that it was writing its standard for, it had to make its product look to much like many different companies proprietary software.[50] TCP/IP was already being widely used, and because OSI did not fulfill its roles as well as it should have, people did not adopt OSI widely enough for it to become the standard. Unlike ISO, TCP/IP was able to avoid this problem because of the way it was developed. The separate protocols that made up TCP/IP were created and modified as soon as a need for them was realized, and so they came into a field that was at the time empty of competition. The OSI standard took several years to develop precisely because of the way the International Standards Organization was organized. The protocols developed for the ARPANET were also produced fully implemented, and consequently easy for developers to use. The OSI protocols for the some applications were much more complex, as they had been designed with out ever being tested to see how well they would work in real world situations.[51]

One good thing OSI did was to supply as its reference model to TCP/IP.[52] Prior to the OSI model, TCP/IP was well implemented but poorly conceptualized. The ISO development team was able to stand back and look at the full picture, and so develop a

[50] Foley, 123
[51] Hanseth, Ole, et al., "Developing Information Infrastructure: The Tension between Standardization and Flexibility," *Science, Technology, and Human Values*, Vol. 21, No.4 (Autumn, 1996), pg. 413
[52] English, pg. 13

very good conceptualized picture of how the Internet worked. This theoretical model was then taken by Internet developers and applied to the TCP/IP model.[53] The ISO model was very important to the development of TCP/IP, as it introduced the idea of abstracted layers. Because TCP/IP was developed in an ad hoc fashion, no one took a wider look at the overall protocol model.[54] The advantage of a layered protocol model like OSI is that any individual protocol layer can be completely replaced, and as long the replacement layer communicates with the other layers in the same fashion it predecessor did, the other layers do not have to be changed. For example, one could edit the data link and physical layers of the OSI stack to use carrier pigeons instead of copper wire.[55] A more realistic example is the accessing of the Internet through wireless connections instead of the traditional copper wires. The application and transport layers are unchanged even though lower layers have to be significantly altered.

Private protocols like SNA and DECnet also fell victim to the same chicken and egg problem that OSI fell victim too, even though they were under development at the same time as TCP/IP. TCP/IP did not fall victim to this chicken and egg trap because of its openness in both development and in structuring. SNA, for example, could never have served as a network backbone due to its hierarchical layout. In order to operate, it needs to the exact hierarchy of all the machines in the network.[56] Considering the rather fluid nature of the Internet, this would cause problems. SNA is structured this way because it was originally designed for large mainframes,[57] where it is safe to assume that a network will remain relatively stable for any great length of time. The hierarchical

---

[53] Tanenbaum, pg. 48-49.
[54] Ibid
[55] Waitzman, D. "Request For Comments: 1149," (1 April, 1990)
[56] Jander, Mary, "Managing the Mix," *Data Communications*, (7/1/1998), vol. 27, iss. 10, pg. 73.
[57] Ibid.

layout is a direct result of the 'cathedral/proprietary' development method used to create it. A stable hierarchy is much easier to plan and predict than other more ad-hoc ways of organizing. Later, when TCP/IP was being developed, IBM, the owner of SNA, was not about to risk substantial sums of money to develop a network protocol that could run on a then experimental market.

The most significant change to the nature of the Internet since its creation came in the early- and mid-1990's with the creation of the World Wide Web. Tim Berners-Lee, the man widely credited with inventing the Web, developed it while working for a Geneva based physics research lab named CERN in the late 1980's.[58] The World Wide Web was built directly on the previously existing Arpanet inspired networks, but was designed to be much more intuitive to an average user.[59] The World Wide has had a huge impact on Internet commerce as it has effectively created the field of direct customer-to-business electronic commerce. Prior to the World Wide Web, the closest a customer could get to electronic commerce was ordering products over the phone, unless that customer had his own mainframe.

Before the development of the World Wide Web, there was Gopher. Mark McCahill developed gopher at the University of Minnesota from 1991 to 1993.[60] The purpose of Gopher was much like that of the World Wide Web; to create easy to use access software for the Internet.[61] Gopher did a very good job of this, structuring pages

[58] World Wide Web Consortium. "Longer Bio for Tim Berners-Lee," *The World Wide Web Consortium Website*, < http://www.w3.org/People/Berners-Lee/Longer.html > (September 12th, 2004).
[59] Berners-Lee, pg. 39
[60] Frana, Phillip L. "Before the Web There Was Gopher," *IEEE: Annals of the History of Computing*, v. 26 n.1, Jan – Mar 2004, 20.
[61] Frana, 22

of information on a hierarchical basis in an easy to use way. Yet by 1995, the World Wide Web had overtaken and effectively killed gopher.

Much of this can be attributed to the two systems development styles. Gopher, while developed like an open source project, was limited to the University of Minnesota in development.[62] More importantly, the technology was actually owned by the University, which caused hesitation among those worried that some organization would later charge for the rights to use Gopher.[63] The World Wide Web, while originally developed at the CERN physics lab, had a wider initial development base as well as a clearer release license; Tim Berners-Lee made sure that CERN released the code behind the World Wide Web under General Public License.[64] This meant that anyone who wished to could modify the code.

What really killed Gopher was the way it was organized. Gopher was organized along a tree-like hierarchical structure with links and nodes. This limited the way a Gopher site would be organized, and often led to poorly organized sites.[65] The World Wide Web in contrast used hypertext, where a word or phrase could link to any other web page the creator wanted. While there are certainly poorly organized websites, hypertext lets designers be more flexible in how they design their websites.

As mentioned above, the World Wide Web and TCP/IP helped each other to succeed. When Tim Berners-Lee set out to develop the Web, he chose to do it on a TCP/IP platform.[66] TCP/IP offered to him a popular, free protocol that was available for multiple operating systems. His development of the World Wide Web caused the

---

[62] Frana, 22
[63] Frana, 30
[64] Berners-Lee, 73-74
[65] Frana, 34
[66] Berners-Lee, 19

Internet, and consequently TCP/IP, to explode in volume by opening up the Internet to non-professional computer users.

With the popularity of the Internet came the need to secure it. Even if a user is not conducting a business transfer, he probably does not want a random people reading what he is sending and receiving. This growth in the Internet brought about the development of good Internet cryptography. The problem with any form of encryption is that it must be good enough to guaranty security, but must also be relatively easy to implement. At this point one also runs into the classic dilemma of encryption; how do you instruct the person your sending information to how to decrypt that information without telling everyone else? A solution to both these problems is found with public key encryption, which was originally developed by Whitfield Diffie and Martin Hellman in1976.[67] There idea was to use "trap door" function, so-called because they are very easy to compute one way, but very difficult to calculate in the other direction.[68] R.L. Rivest, A. Shamir, and L. Adleman later took Diffie and Hellman's thoery and turned it into a pratical method of encryption in 1978.[69] They created a one-way function by multiplying two very large (i.e. 100 digit length) prime numbers. While it is farily simple to culculate the product of two such numbers, it is extremely time consuming to try to factor that number.[70]

Rivest, Shamir, and Adleman were able to develop their algorithms because Diffie and Hellman freely published their studies. It is rather ironic then that Rivest, Shamir, and Adleman then patented the algorithms they developed and created a private

[67] Rivest, 120
[68] Rivest, 121
[69] Rivest, 120
[70] Rivest, 121

corporation, RSA Data Security Inc., to produce encryption programs based off those algorithms. This didn't stop Phil Zimmermann from effectively stealing those algorithms in 1991 to produce his Pretty Good Privacy program. This not only started a dispute with RSA Data Security Inc., but also with the Government which classifies encryption software as a munition and bans its export. While the fight with the government over export laws would last until 1996[71], the legal battle between RSA Data Security Inc. and Zimmermann was resolved by 1994 with a clever open sourced solution. While the RSA algorithms were proprietary, RSA Data Security Inc. had released a freeware e-mail encryption program called RSAREF. Zimmermann simply rebuilt his PGP encryption program on top of RSAREF thus bypassing the copy right restrictions on most of the RSA algorithms.[72]

Zimmermann's 'use' of RSA encryption raises isues about both the nature of buiness and the way patent law should work. Here is a situation that is legally clear; Zimmermann used Rivest, Shamir, and Adleman's intelectual property with out gaining permission to do so first. There are other issues involved that I will not cover that in this paper, like weather one can put a patent on an algorithm or the very significant free speech issues involved in letting private citizens use encryption even the government can not break. The issue I am pointing out is the idea of technology being used for the common good. By releasing Pretty Good Privacy, Zimmermann helped significantly increase electronic commerce. Because people can now use PGP on their home computers, they can conduct electronic commerce in a safe and secure manner. While

[71] Zimmerman, Phillip. "Background," Phil Zimmermann's Home Page, <http://www.philzimmermann.com/EN/background/index.html > (September 11th, 2004).

[72] Levy, Steven. "Cypher Wars: Pretty Good Privacy Gets Pretty Legal" *Wired.com* <http://www.wired.com/wired/archive/2.11/cypher.wars.html> (November, 1994)

companies would have been able to purchase RSA encryption, that overhead would have significantly reduced smaller compnaies ability to do electronic commerce. This is especially true considering that RSA had an effective monopoly on effective encryption systems.

In conclusion, the Internet is the product of a combination of open source and standards development. More importantly, the Internet is a product of ad hoc development, with a new feature added whenever someone felt the need for it. This piece by piece development also means that the Internet is very well suited for growth. Unlike a pre-designed hierarchical network, new equipment can be added to Internet easily with only a minimal amount of equipment recalibration. This ease of use means that electronic commerce can be easily implemented. If a company finds that its servers are being overworked, it can easily add new ones. Companies can also easily offer new services because of TCP/IP's flexibility. Thanks to the overlaid OSI model, it is now easy to add a new service at the application layer without modifying the underlying layers. This flexibilty in the Internet has let electronic commerce flourish.

Bibliography: Chicago style

Berinato, Scott. "Access To SNA over IP Getting Cheaper, Easier," *PC Week*, (08/10/98), Vol. 15 Issue 32, p8

Berners-Lee, Tim. “Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by It Inventor,” (San Francisco: HarperCollins, 1999). This is Tim Berners-Lee’s account of the creation of the world wide web. Useful for analyzing the open source methods of development

Bhimani, Anish. “Securing the Commercial Internet.” In “Communications of the ACM” Vol. 39, Iss. 6, June 1996, pg. 29-35. This discusses the issues of security and encryption in Electronic Commerce.

Castells, Manuel. “The Internet Galaxy: Reflections on the Internet, Business, and Society.” (Oxford: Oxford University Press, 2001).

*Computer Elencyclopedia,* “Electronic Commerce” 628

English, Erin. “TCP/IP Gets a Face-lift,” *Computer,* (October 1995), pg. 12 – 13.

Ensmenger, Nathan L. “Open Source’s Lessons for Historians,” *IEEE: Annals of the History of Computing*, Vol. 26, No. 4 (October-December 2004), pg. 102 –104

Frana, Phillip L. “Before the Web There Was Gopher,” *IEEE: Annals of the History of Computing*, v. 26 n.1, Jan – Mar 2004, pg. 20 – 41.

Foley, Jerry, and Severance, Charles. “OSI Retrospect and Prospect,” *Computer,* (September 1997), pg. 123 – 124.

Gray, James P. “Services Provided to Users of SNA Networks,” IEEE, 1979.

Hanseth, Ole, et al., “Developing Information Infrastructure: The Tension between Standardization and Flexibility,” *Science, Technology, and Human Values,* Vol. 21, No.4 (Autumn, 1996), pg. 407 – 425.

Hoffman, P. and Harris, S., “The Tao of IEFT – A Novices Guide to the Internet Engineering Task Force,” *The IEFT Education Site,* <http://edu.ietf.org/tao>, (December 9th, 2004), The Tao of IEFT

Internet Engineering Task Force, “RFC Editor: Publication Process,” *RFC Editor Web Page,* <http://www.rfc-editor.org/howtopub.html> (December 12th, 2004), How to Publish

Jander, Mary, “Managing the Mix,” *Data Communications,* (7/1/1998), vol. 27, iss. 10, pg. 72, 9p.

Levy, Steven. “Cypher Wars: Pretty Good Privacy Gets Pretty Legal” *Wired.com* <www.wired/archive/2.11/cypher.wars.html> (November, 1994).

Mandt, Hans W. “Data Management in Electronic Funds Transfer Systems,” (Oxford, Ohio: Miami University, 1978).

Meleis, Hanafy. “Toward the Information Network.” *Computer,* Vol. 29, Iss.10, October, 1996, pg.59-67. This talks about the early history of the internet, and goes into detail about the ARPANET.

Norberg, Arthur L. “Changing Computing: The Computing Community and DARPA” Vol. 18, No. 2, (Summer 1996), pg. 40 - 53

O’Neill, Judy E., “The Role of ARPA in the Development of the ARPANET, 1961-1972.” *Annals of the History of Computing,* Vol. 17, Num. 4, pg. 76 - 81.

Open Source dot Org, “The Approved Licenses,” <http://www.opensource.org/licenses/index.php> (December 12th, 2004), Open Source Licenses.

Paxson, Vern. “Growth Trends in Wide-Area TCP Connections,” *IEEE Networks,* (July/August 1994) pg. 8 – 17.

Postel, Jon et al. “The Past and Future History of the Internet.” In *“Communications of the ACM”* Vol 40, no 2, February 1997, pg. 102-108. While this is much shorter than Berners-Lee’s version, it offers a different perspective on the same topic.

Rivest, R.L., Shamir, A. and Adleman, L. “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems.” In *“Communications of the ACM”* Vol 21, No 2, February 1978, pg. 120-126. This is the initial publication of the RSA encryption system, which provides security for most electronic commerce.

Panurach, Patiwat. “Money in electronic commerce: digital cash, electronic fund transfer, and Ecash.” In *“Communications of the ACM”* Vol 39, iss 6, June 1996, pg. 45-50. Actually part of a six part series, this discusses online financial security.

Raymond, Eric S. “The Cathedral and The Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary.” (Cambridge: O’Rielly, 2001) This is one of my sources for analysis of the open source community.

Riggins, Fredrick J. and Rhee, Hyeun-Suk (Sue). "Toward a Unified View of Electronic Commerce" In "*Communications of the ACM*" Vol. 41, Iss 10, October, 1998, p. 91. This text is useful in defining what exactly constitutes electronic commerce.

Schien, Edgar H. "DEC is Dead - Long Live DEC: The Lasting Legacy of Digital Equipment Corporation," (San Francisco: Berrett-Koehler Publishers, Inc, 2003).

Tannenbaum, Andrew S. "Computer Networks," 4th ed. (Upper Saddle River, New Jersey: Prentice Hall, 2003).

Tillman, Matthew A. and Yen, David. "SNA and OSI: Three Strategies for Interconnection," *Communications of the ACM,* v. 33, no. 2, (February 1990), pg. 214 – 224.

Thyfault, Mary E., "Sears selects IP," *InformationWeek*, (06/09/97) Issue 634, p107

Wittie, Larry D. "Computer Networks and Distributed Systems," *Computer*, (September, 1991), pg. 67 – 75.

Waitzman, D. "Request For Comments: 1149," (1 April, 1990)

World Wide Web Consortium. "The World Wide Web Consortium Website" < http://www.w3.org/ > (September 12th, 2004). The World Wide Web consortium is a group that played a key role in the early internet protocol standardization movement, and is mentioned in several books and articles.

Zimmerman, Phillip. "Background," *Phil Zimmermann's Home Page*, < http://www.philzimmermann.com/EN/background/index.html > (September 11th, 2004).