1 Introduction

1.1 Internet

Some thirty years ago, the RAND Corporation, America's foremost Cold War think-tank, faced a strange strategic problem. How could the US authorities successfully communicate after a nuclear war? Post nuclear America would need a command-and-control network, linked from city to city, state to state, base to base. But no matter how thoroughly that network was armored or protected, its switches and wiring would always be vulnerable to the impact of atomic bombs. A nuclear attack would reduce any conceivable network to tatters. And how would the network itself be commanded and controlled? Any central authority, any network central citadel, would be an obvious and immediate target for an enemy missile. The center of the network would be the very first place to go.

RAND mulled over this grim puzzle in deep military secrecy, and arrived at a daring solution. The RAND proposal (the brainchild of RAND staffer Paul Baran) was made public in 1964. In the first place, the network would *have no central authorities. * Furthermore, it would be *designed from the beginning to operate while in tatters. *

The principles were simple. The network itself would be assumed to be unreliable at all times. It would be designed from the get-go to transcend its own unreliability. All the nodes in the network would be equal in status to all other nodes, each node with its own authority to originate, pass, and receive messages. The messages themselves would be divided into packets, each packet separately addressed. Each packet would begin at some specified source node, and end at some other specified destination node. Each packet would wind its way through the network on an individual basis.

The particular route that the packet took would be unimportant. Only final results would count. Basically, the packet would be tossed like a hot potato from node to node to node, more or less in the direction of its destination, until it ended up in the proper place. If big pieces of the network had been blown away, that simply wouldn't matter; the packets would still stay airborne, laterally wildly across the field by whatever nodes happened to survive. This rather haphazard delivery system might be "inefficient" in the usual sense (especially compared to, say, the telephone system) -- but it would be extremely rugged.

During the 60s, this intriguing concept of a decentralized, blast proof, RAND, MIT and UCLA kicked around packet-switching network. The National Physical Laboratory in Great Britain set up the first test network on these principles in 1968.

Shortly afterward, the Pentagon's Advanced Research Projects Agency decided to fund a larger, more ambitious project in the USA. The nodes of the network were to be high-speed supercomputers (or what passed for supercomputers at the time). These were rare and valuable machines which were in real need of good solid networking, for the sake of national research-and-development projects. In fall 1969, the first such node was installed in UCLA. By December 1969, there were four nodes on the infant network, which was named ARPANET, after its Pentagon sponsor. The four computers could transfer data on dedicated high-speed transmission lines. They could even be programmed remotely from the other nodes. Thanks to ARPANET, scientists and researchers could share one another's computer facilities by long-distance. This was a very handy service, for computer-time was precious in the early '70s. In 1971 there were fifteen nodes in ARPANET by 1972, thirty-seven nodes. And it was good. By the second year of operation, however, an odd fact became clear. ARPANET's users had warped the computer-sharing network into a dedicated, high-speed, federally subsidized electronic post-office. The main traffic on ARPANET was not long-distance computing. Instead, it was news and personal messages. Researchers were using ARPANET to collaborate on projects, to trade notes on work, and eventually, to downright gossip and schmooze. People had their own personal user accounts on the ARPANET computers, and their own personal addresses for electronic mail. Not only were they using ARPANET for person-to-person communication, but also they were very enthusiastic about this particular service -- far more enthusiastic than they were about long-distance computation. It wasn't long before the invention of the mailing list, an ARPANET broadcasting technique in which an identical message could be sent automatically to large numbers of network subscribers. Interestingly, one of the first really big mailing lists was "SF-LOVERS," for science fiction fans. Discussing science fiction on the network was not work-related and was frowned upon by many ARPANET computer administrators, but this didn't stop it from happening. Throughout the '70s, ARPA's network grew. Its decentralized structure made expansion easy. Unlike standard corporate computer networks, the ARPA network could accommodate many different kinds of machine. As long as individual machines could speak the packet-switching lingua franca of the new, anarchic network, their brand names, and their content, and even their ownership were irrelevant. The ARPA's original standard for communication was known as NCP, "Network Control Protocol," but as time passed and the technique advanced, NCP was superceded by a higher-level, more sophisticated standard known as TCP/IP. TCP, or "Transmission Control Protocol," converts messages into streams of packets at the source, then reassembles them back into messages at the destination. IP, or "Internet Protocol," handles the addressing, seeing to it that packets are routed across multiple nodes and even across multiple networks with multiple standards -- not only ARPA's pioneering NCP standard, but also others like Ethernet, FDDI, and X.25.

As early as 1977, other networks to link to ARPANET were using TCP/IP. ARPANET itself remained fairly tightly controlled, at least until 1983, when its military segment broke off and became MILNET. But TCP/IP linked them all. And ARPANET itself, though it was growing, became a smaller and smaller neighborhood amid the vastly growing galaxy of other linked machines. As the '70s and '80s advanced, many very different social groups found themselves in possession of powerful computers. It was fairly easy to link these computers to the growing network-of- networks. As the use of TCP/IP became more common, entire other networks fell into the digital embrace of the Internet, and messily adhered. Since the software called TCP/IP was public-domain, and the basic technology was decentralized and rather anarchic by its very nature, it was difficult to stop people from barging in and linking up somewhere-or-other. In point of fact, nobody *wanted* to stop them from joining this branching complex of networks, which came to be known as the "Internet." Connecting to the Internet cost the taxpayer little or nothing, since each node was independent, and had to handle its own financing and its own technical requirements. The more, the merrier. Like the phone network, the computer network became steadily more valuable as it embraced larger and larger territories of people and resources. A fax machine is only valuable if *everybody else* has a fax machine. Until they do, a fax machine is just a curiosity. ARPANET, too, was a curiosity for a while. Then computer-networking became an utter necessity.

In 1984 the National Science Foundation got into the act, through its Office of Advanced Scientific Computing.

The new NSFNET set a blistering pace for technical advancement, linking newer, faster, shinier supercomputers, through thicker, faster links, upgraded and expanded, again and again, in 1986, 1988, and 1990. And other government agencies leapt in: NASA, the National Institutes of Health, the Department of Energy, each of them maintaining a digital satrapy in the Internet confederation. The nodes in this growing network-of-networks were divvied up into basic varieties. Foreign computers, and a few American ones, choose to be denoted by their geographical locations. The six basic Internet "domains" grouped the others: gov, mil, edu, com, org and net. (Graceless abbreviations such as this are a standard feature of the TCP/IP protocols.) Gov, Mil, and Edu denoted governmental, military and educational institutions, which were, of course, the pioneers, since ARPANET had begun as a high-tech research exercise in national security. Com, however, stood for "commercial" institutions, which were soon bursting into the network like rodeo bulls, surrounded by a dust-cloud of eager nonprofit "orgs." (The "net" computers served as gateways between networks.)

ARPANET itself formally expired in 1989, a happy victim of its own overwhelming success. Its users scarcely noticed, for ARPANET's functions not only continued but also steadily improved. The use of TCP/IP standards for computer networking is now global. In 1971, mere twenty-one years ago, there were only four nodes in the ARPANET network. Today there are tens of thousands of nodes in the Internet, scattered over forty-two countries, with more coming on-line every day. Three million, possibly four million people use this gigantic mother-of-all-computer-networks.

The Internet is especially popular among scientists, and is probably the most important scientific instrument of the late twentieth century. The powerful, sophisticated access that it provides to specialize data and personal communication has sped up the pace of scientific research enormously. The Internet's pace of growth in the early 1990s is spectacular, almost ferocious. It is spreading faster than cellular phones, faster than fax machines. Last year the Internet was growing at a rate of twenty percent a *month. * The number of "host" machines with direct connection to TCP/IP has been doubling every year since 1988. The Internet is moving out of its original base in military and research institutions, into elementary and high schools, as well as into public libraries and the commercial sector. Why do people want to be "on the Internet?" One of the main reasons is simple freedom. The Internet is a rare example of a true, modern, functional anarchy. There is no "Internet Inc." There are no official censors, no bosses, no board of directors, and no stockholders. In principle, any node can speak as a peer to any other node, as long as it obeys the rules of the TCP/IP protocols, which are strictly technical, not social or political. (There has been some struggle over commercial use of the Internet, but that situation is changing as businesses supply their own links). The Internet is also a bargain. The Internet as a whole, unlike the phone system, doesn't charge for long-distance service. And unlike most commercial computer networks, it doesn't charge for access time, either. In fact the "Internet" itself, which doesn't even officially exist as an entity, never "charges" for anything. Each group of people accessing the Internet is responsible for their own machine and their own section of line. The Internet's "anarchy" may seem strange or even unnatural, but it makes a certain deep and basic sense. It's rather like the "anarchy" of the English language. Nobody rents English, and nobody owns English. As an English-speaking person, it's up to you to learn how to speak English properly and make whatever use you please of it (though the government provides certain subsidies to help you learn to read and write a bit). Otherwise, everybody just sort of pitches in, and somehow the thing evolves on its own, and somehow turns out workable. And interesting. Fascinating, even. Though a lot of people earn their living from using and exploiting and teaching English, "English" as an institution is public property, a public good. Much the same goes for the Internet. Would English be improved if the "The English Language, Inc." had a board of directors and a chief executive officer, or a President and a Congress? There'd probably be a lot fewer new words in English, and a lot fewer new ideas. People on the Internet feel much the same way about their own institution. It's an institution that resists institutionalization. The Internet belongs to everyone and no one. Still, its various interest groups all have a claim. Business people want the Internet put on a sounder financial footing. Government people want the Internet more fully regulated. Academics want it dedicated exclusively to scholarly research. Military people want it spy-proof and secure. And so on and so on. All these sources of conflict remain in a stumbling balance today, and the Internet, so far, remains in a thrivingly anarchical condition. Once upon a time, the NSFnet's high-speed, high-capacity lines were known as the "Internet Backbone," and their owners could rather lord it over the rest of the Internet; but today there are "backbones" in Canada, Japan, and Europe, and even privately owned commercial Internet backbones specially created for carrying business traffic. Today even privately owned desktop computers could become Internet nodes. You can carry one under your arm. Soon, perhaps, on your wrist. But what does one *do* with the Internet? Four things, basically: mail, discussion groups, long-distance computing, and file transfers. Internet mail is "e-mail," electronic mail, faster by several orders of magnitude than the US Mail, which is scornfully known by Internet regulars as "snailmail." Internet mail is somewhat like fax. It's electronic text. But you don't have to pay for it (at least not directly), and it's global in scope. E-mail can also send software and certain forms of compressed digital imagery. New forms of mail are in the works. The discussion groups, or "newsgroups," are a world of their own. This world of news, debate and argument is generally known as "USENET. " USENET is, in point of fact, quite different from the Internet. USENET is rather like an enormous billowing crowd of gossipy, news-hungry people, wandering in and through the Internet on their way to various private backyard barbecues. USENET is not so much a physical network as a set of social conventions. In any case, at the moment there are some 2,500 separate newsgroups on USENET, and their discussions generate about 7 million words of typed commentary every single day. Naturally there is a vast amount of talk about computers on USENET, but the variety of subjects discussed is enormous, and it's growing larger all the time. USENET also distributes various free electronic journals and publications. Both Netnews and e-mail are very widely available, even outside the high-speed core of the Internet itself. News and e-mail are easily available over common phone-lines, from Internet fringe- realms like BITnet, UUCP and Fidonet. The last two Internet services, long-distance computing and file transfer, require what is known as "direct Internet access" -- using TCP/IP. Long-distance computing was an original inspiration for ARPANET and is still a very useful service, at least for some. Programmers can maintain accounts on distant, powerful computers, run programs there or write their own. Scientists can make use of powerful supercomputers a continent away. Libraries offer their electronic card catalogs for free search. Enormous CD-ROM catalogs are increasingly available through this service. And there are fantastic amounts of free software available. File transfers allow Internet users to access remote machines and retrieve programs or text. Many Internet computers – some two thousand of them, so far -- allow any person to access them anonymously, and to simply copy their public files, free of charge. This is no small deal, since entire books can be transferred through direct Internet access in a matter of minutes. Today, in 1992, there are over a million such public files available to anyone who asks for them (and many more millions of files are available to people with accounts). Internet file-transfers are becoming a new form of publishing, in which the reader simply electronically copies the work on demand, in any quantity that he or she wants, for free. New Internet programs, such as "archie," “gopher,” and “WAIS,” has been developed to catalog and explore these enormous archives of material. The headless, anarchic, million-limbed Internet is spreading like bread mold. Any computer of sufficient power is a potential spore for the Internet, and today such computers sell for less than $2,000 and are in the hands of people all over the world. ARPA's network, designed to assure control of a ravaged society after a nuclear holocaust, has been superceded by its mutant child the Internet, which is thoroughly out of control, and spreading exponentially through the post-Cold War electronic global village. The spread of the Internet in the 90s resembles the spread of personal computing in the 1970s, though it is even faster and perhaps more important. More important, perhaps, because it may give those personal computers a means of cheap, easy storage and access that is truly planetary in scale. The future of the Internet bids fair to be bigger and exponentially faster. Commercialization of the Internet is a very hot topic today, with every manner of wild new commercial information- service promised. The federal government, pleased with an unsought success, is also still very much in the act. NREN, the National Research and Education Network, was approved by the US Congress in fall 1991, as a five-year, $2 billion project to upgrade the Internet "backbone." NREN will be some fifty times faster than the fastest network available today, allowing the electronic transfer of the entire Encyclopedia Britannica in one hot second. Computer networks worldwide will feature 3-D animated graphics, radio and cellular phone-links to portable computers, as well as fax, voice, and high- definition television. A multimedia global circus! Or so it's hoped -- and planned. The real Internet of the future may bear very little resemblance to today's plans. Planning has never seemed to have much to do with the seething, fungal development of the Internet. After all, today's Internet bears little resemblance to those original grim plans for RAND's post- holocaust command grid. It's a fine and happy irony. How does one get access to the Internet? Well -- if you don't have a computer and a modem, get one. Your computer can act as a terminal, and you can use an ordinary telephone line to connect to an Internet-linked machine. These slower and simpler adjuncts to the Internet can provide you with the Netnews discussion groups and your own e-mail address. These are services worth having – though if you only have mail and news, you're not actually "on the Internet" proper. If you're on a campus, your university may have direct "dedicated access" to high-speed Internet TCP/IP lines. Apply for an Internet account on a dedicated campus machine, and you may be able to get those hot-dog long-distance computing and file-transfer functions. Some cities, such as Cleveland, supply "freenet" community access. Businesses increasingly have Internet access, and are willing to sell it to subscribers. The standard fee is about $40 a month -- about the same as TV cable service. As the Nineties proceed, finding a link to the Internet will become much cheaper and easier. Its ease of use will also improve, which is fine news, for the savage UNIX interface of TCP/IP leaves plenty of room for advancements in user-friendliness. Learning the Internet now, or at least learning about it, is wise. By the turn of the century, "network literacy," like "computer literacy" before it, will be forcing itself into the very texture of your life.

1.2 History of the Internet

1960 - There is no Internet...

1961 - Still no Internet...

1962 - The RAND Corporation begins research into robust, distributed communication networks for military command and control.

1962-1969

The Internet is first conceived in the early '60s. Under the leadership of the Department of Defense's Advanced Research Project Agency (ARPA), it grows from a paper architecture into a small network (ARPANET) intended to promote the sharing of super-computers amongst researchers in the United States.

1963 - Beatles play for the Queen of England

1964 - 'Dr Strangelove' portrays nuclear holocaust which new network must survive

1965 - The DOD's Advanced Research Project Association begins work on 'ARPANET'

1965 - ARPA sponsors research into a "cooperative network of time-sharing computers."

1966 - US Surveyor probe lands safely on moon

1967 - First ARPANET papers presented at Association for Computing Machinery Symposium

1967 - Delegates at a symposium for the Association for Computing Machinery in Gatlingberg, TN discuss the first plans for the ARPANET.

1968 - First generation of networking hardware and software designed

1969 - ARPANET connects first 4 universities in the United States. Researchers at four US campuses create the first hosts of the ARPANET, connecting Stanford Research Institute, UCLA, UC Santa Barbara, and the University of Utah.

1970 - **ALOHANET** developed at the University of Hawaii

1970-1973 - The ARPANET is a success from the very beginning. Although originally designed to allow scientists to share data and access remote computers, email quickly becomes the most popular application. The ARPANET becomes a high-speed digital post office as people use it to collaborate on research projects and discuss topics of various interests.

1971 - The ARPANET grows to 23 hosts connecting universities and government research centers around the country.

1972 - The InterNetworking Working Group becomes the first of several standards-setting entities to govern the growing network. Vinton Cerf is elected the first chairman of the INWG, and later becomes known as a "Father of the Internet."

1973 - The ARPANET goes international with connections to University College in London, England and the Royal Radar Establishment in Norway.

1974 - Bolt, Beranek & Newman opens Telnet, the first commercial version of the ARPANET.

1974 - 1981 - The general public gets its first vague hint of how networked computers can be used in daily life as the commercial version of the ARPANET goes online. The ARPANET starts to move away from its military/research roots.

1975 - Internet operations transferred to the Defense Communications Agency

1976 - Queen Elizabeth goes online with the first royal email message.

1977 - UUCP provides email on THEORYNET

1978 - TCP checksum design finalized

1979 - Tom Truscott and Jim Ellis, two grad students at Duke University, and Steve Bellovin at the University of North Carolina establish the first USENET newsgroups. Users from all over the world join these discussion groups to talk about the net, politics, religion and thousands of other subjects.

1980 - Mark Andreesen turns 8. 14 more years till he revolutionizes the Web

1981 - ARPANET has 213 hosts. A new host is added approximately once every 20 days.

1982 - The term 'Internet' is used for the first time.

1982 – 1987 - Bob Kahn and Vint Cerf are key members of a team which creates TCP/IP, the common language of all Internet computers. For the first time the loose collection of networks which made up the ARPANET is seen as an "Internet", and the Internet as we know it today is born.

The mid-80s marks a boom in the personal computer and super-minicomputer industries. The combination of inexpensive desktop machines and powerful, network-ready servers allows many companies to join the Internet for the first time. Corporations begin to use the Internet to communicate with each other and with their customers.

1983 - TCP/IP becomes the universal language of the Internet

1984 - William Gibson coins the term "cyberspace" in his novel "Neuromancer." The number of Internet hosts exceeds 1,000.

1985 - Internet e-mail and newsgroups now part of life at many universities

1986 - Case Western Reserve University in Cleveland, Ohio creates the first "Freenet" for the Society for Public Access Computing.

1987 - The number of Internet hosts exceeds 10,000.

1988 - Internet worm unleashed

1988 – 1990 - By 1988 the Internet is an essential tool for communications, however it also begins to create concerns about privacy and security in the digital world. New words, such as "hacker," "cracker" and" electronic break-in", are created.

These new worries are dramatically demonstrated on Nov. 1, 1988 when a malicious program called the "Internet Worm" temporarily disables approximately 6,000 of the 60,000 Internet hosts.

1988 - The Computer Emergency Response Team (CERT) is formed to address security concerns raised by the Worm.

1989 - System administrator turned author, Clifford Stoll, catches a group of Cyberspies, and writes the best-seller "The Cuckoo's Egg." The number of Internet hosts exceeds 100,000.

1990 - A happy victim of its own unplanned, unexpected success, the ARPANET is decommissioned, leaving only the vast network-of-networks called the Internet. The number of hosts exceeds 300,000.

1991 - The World Wide Web is born!

1991 – 1993 - Corporations wishing to use the Internet face a serious problem: commercial network traffic is banned from the National Science Foundation's NSFNET, the backbone of the Internet. In 1991 the NSF lifts the restriction on commercial use, clearing the way for the age of electronic commerce.

At the University of Minnesota, a team led by computer programmer Mark MaCahill releases "gopher," the first point-and-click way of navigating the files of the Internet in 1991. Originally designed to ease campus communications, gopher is freely distributed on the Internet. MaCahill calls it "the first Internet application my mom can use." 1991 is also the year in which Tim Berners-Lee, working at CERN in Switzerland, posts the first computer code of the World Wide Web in a relatively innocuous newsgroup, "alt.hypertext." The ability to combine words, pictures, and sounds on Web pages excites many computer programmers who see the potential for publishing information on the Internet in a way that can be as easy as using a word processor.

Marc Andreesen and a group of student programmers at NCSA (the National Center for Supercomputing Applications located on the campus of University of Illinois at Urbana Champaign) will eventually develop a graphical browser for the World Wide Web called Mosaic.

1991 - Traffic on the NSF backbone network exceeds 1 trillion bytes per month.

1992 - One million hosts have multi-media access to the Internet over the MBONE

1992 - The first audio and video broadcasts take place over a portion of the Internet known as the "MBONE." More than 1,000,000 hosts are part of the Internet.

1993 - Mosaic, the first graphics-based Web browser, becomes available. Traffic on the Internet expands at a 341,634% annual growth rate.

1994 - The Rolling Stones broadcast the Voodoo Lounge tour over the M-Bone. Marc Andreesen and Jim Clark form Netscape Communications Corp. Pizza Hut accepts orders for a mushroom, pepperoni with extra cheese over the net, and Japan's Prime Minister goes online at www.kantei.go.jp. Backbone traffic exceeds 10 trillion bytes per month.

1995 - NSFNET reverts back to a research project, leaving the Internet in commercial hands. The Web now comprises the bulk of Internet traffic. The Vatican launches www.vatican.va. James Gosling and a team of programmers at Sun Microsystems release an Internet programming language called Java, which radically alters the way applications and information can be retrieved, displayed, and used over the Internet.

1996 - Nearly 10 million hosts online. The Internet covers the globe

1996 - As the Internet celebrates its 25th anniversary, the military strategies that influenced its birth become historical footnotes. Approximately 40 million people are connected to the Internet. More than $1 billion per year changes hands at Internet shopping malls, and Internet related companies like Netscape are the darlings of high-tech investors.

Users in almost 150 countries around the world are now connected to the Internet. The number of computer hosts approaches 10 million.

Within 30 years, the Internet has grown from a Cold War concept for controlling the tattered remains of a post-nuclear society to the Information Superhighway. Just as the railroads of the 19th century enabled the Machine Age, and revolutionized the society of the time, the Internet takes us into the Information Age, and profoundly affects the world in which we live.

The Age of the Internet has arrived.

1997 - Today some people telecommute over the Internet, allowing them to choose where to live based on quality of life, not proximity to work. Many cities view the Internet as a solution to their clogged highways and fouled air. Schools use the Internet as a vast electronic library, with untold possibilities. Doctors use the Internet to consult with colleagues half a world away. And even as the Internet offers a single Global Village, it threatens to create a 2nd class citizenship among those without access. As a new generation grows up as accustomed to communicating through a keyboard as in person, life on the Internet will become an increasingly important part of life on Earth.

When Tim Berners-Lee invented the technology and built the infrastructure that would become the World Wide Web, he envisioned nothing more than a global application that would allow its users to publish and link documents without regard for location and platform.

It was a bold vision, and like many great ideas, it vastly underestimated the true utility of the World Wide Web.
With the benefit of hindsight, one may be surprised by the narrowness of the original vision. The benefits delivered by the technology underlying the Web are obvious in retrospect.

The Web is built upon a family of simple, robust, nonproprietary protocols. The HyperText Transfer Protocol, or HTTP (which overlays a number of more complicated transport protocols), unifies communication between clients and servers. The Universal Resource Locator, or URL, solves the difficult problem of uniquely identifying resources and data on a widely dispersed network without centralized control and organization. The HyperText Markup Language (HTML) provides a presentation layer that is lightweight and independent of the client platform.

These characteristics made the World Wide Web and Web technology the ideal catalyst for a set of trends that had been developing in corporate computing circles for a number of years.

Web technologies could easily replace limited, expensive, proprietary tools and protocols for sharing information over the corporate LAN with robust, open protocols and tools. This was the beginning of the corporate intranet.

Then, with the intranet in place, Web technologies provided a platform that enabled users with desktop machines to access the information and resources present on legacy systems. Tools that began as simple, Web-based reporting front-ends for legacy databases quickly grew in functionality until they were capable of integrating multiple back-end systems under a single interface. HTML made this step possible. HTML made the activity of designing attractive, lightweight user interfaces relatively painless. And the resulting screens were both portable across client platforms and free of the problems that often came with distributing and updating traditional client code.

Beyond the bounds of the corporate intranet, the World Wide Web allowed companies to bring these applications to remote users without the trouble of a dedicated wide-area network (WAN) or dialup connections. More importantly, the ubiquitous nature of the World Wide Web, and the now-universal nature of the protocols and technologies upon which it was built, made it possible to extend these same corporate applications out to suppliers, partners, and customers.

Since its inception as an implementation in the NCSA Web server, CGI has become a powerful and widely used standard for interfacing external applications with Web servers.

There are various advantages of CGI:

- **Simplicity.** CGI provides a simple way of running programs on the server when a request is received and it is conceptually easy to understand the underlying process.

- **Process Isolation.** Since CGI applications run in separate processes, buggy applications will not crash the Web server or access the server's private internal state.

- **Portability.** CGI is an open standard. CGI is not tied to any particular (such as single- or multi-threaded) server architecture. CGIs are far more portable over other alternatives such as server extension APIs.

- **Language Independency.** CGI applications can be written in nearly any language.

- **Support.** CGI is a proven technology, and some form of CGI has been implemented on almost every Web server on a variety of platforms. There are many CGI scripts available for free for a variety of applications: as user-friendly front-ends to databases, search engines, scientific analysis tools, traditional inventory systems, gateways to network services such as gopher or whois.

The CGI approach, however, also has both implicit and explicit limitations. In this article, we discuss these drawbacks, and suggest solutions to minimize, and even eliminate them in certain cases.

Although any programming language can be used to write CGI scripts, Perl has established itself as the ***lingua franca*** for CGI programming. When referring to a programming language, we will refer to Perl exclusively.

1.3 Constraints of CGI

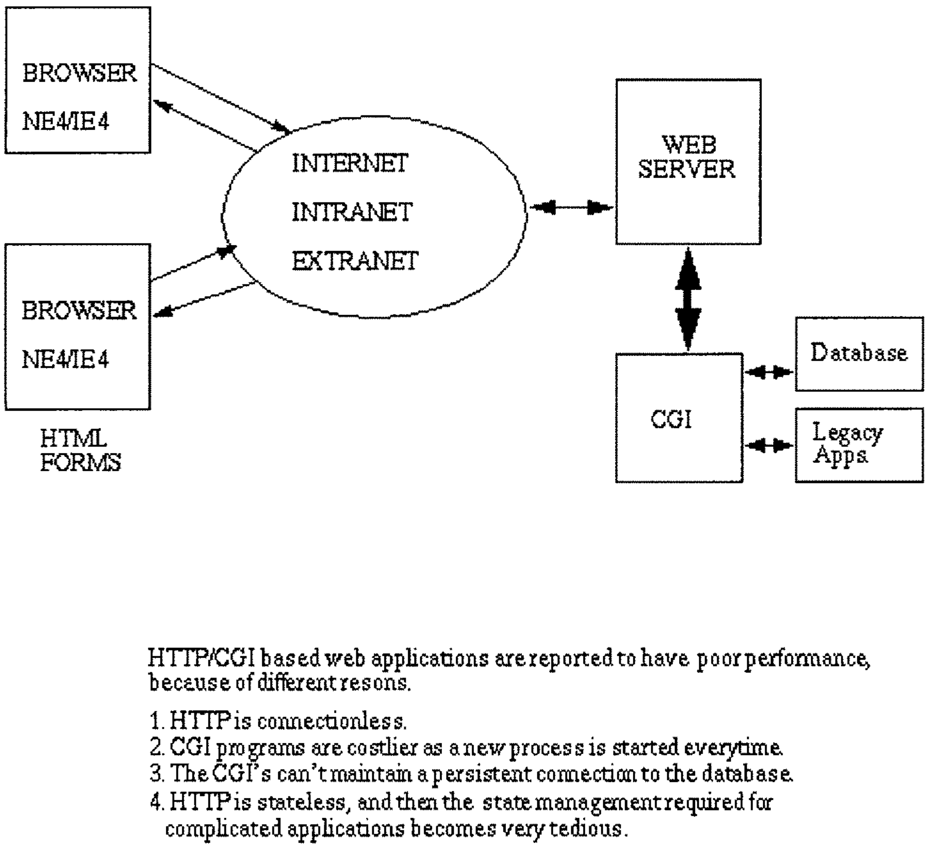

Figure 1 : Constraints of CGI

1.4 Performance

The leading CGI problem is performance, and is reflected in various forms.

- **Time Lag.** There is a *lag* between a Web server receiving a CGI request and an external program handling it. If an HTML file must be parsed, the lag can become even worse. Some solutions attempt to keep the CGI loaded in memory or take advantage of operating system facilities to keep processing time to a minimum, but these still do not eliminate the lag.

- **Forking.** Since a CGI script is external to a Web server, it *must* be launched as a separate process. This usually means the server is forking a new process. The overhead of launching and running a script can be significant on some operating systems. It can become even more important on a frequently-used server when the number of processes grows to problem levels. If the CGI scripts are themselves lengthy, this is heavily exacerbated.

- **Initialization.** A CGI program must initialize. CGI programs are loaded from disk (or disk cache) into memory and run each time their services are requested. A separate process is created and the program is executed. This process can be resource-intensive in terms of CPU and memory usage. This problem is compounded on a site where multiple users are requesting services from the CGI program simultaneously.

In the case of a compiled language such as C/C++ the overhead is negligible but sacrifices portability in many instances. Writing a CGI script in Perl gives it a semblance of platform independence, but it also requires that each *separate* Perl *interpreter* which takes more time and requires extra resources.

1.5 Statelessness

The CGI overhead is primarily as a consequence of HTTP being a *stateless* protocol. CGI scripts exit as soon as they are finished processing the current request. If the CGI script does some time-consuming task at start-up (such as connecting to a database), the overhead of reestablishing the state each time it is needed is considerable.

1.6 Scalability

When a server responds to a request that accesses a CGI program, it *must* create a *new process* to run the CGI program, and then pass to it, via environment variables and standard output, the information that might be necessary to generate a response. Creating a process for every such request requires time and significant server resources, which limits the number of requests a server can handle concurrently.

Thus, CGI scripts do not scale well. With the invocation of each CGI process, the load on the server increases and performance degrades. CGI is often used as frontend to other applications, such as a database, which may take considerable time to initialize. This can also represent a major overhead.

1.7 Functionality

CGI has a limited functionality: it only plays a simple role of a "responder", where the application generates the response that is returned to the client. Once it begins execution, a CGI program runs in a separate process. Therefore, it cannot interact with the Web server (to link into other stages of Web server request processing, such as authorization and logging) or take advantage of the server's capabilities. (For example, a CGI script cannot write to the server log files.) This can have an indirect impact on performance as well.