

An Overview of TCP/IP Protocols and the Internet

Gary C. Kessler
gck@garykessler.net
21 July 2019

© 1994-2019 — This paper was originally submitted to the InterNIC and posted on their Gopher site on 5 August 1994. This document is a continually updated version of that paper.

Readers may also be interested in the GKA [TCP/IP Pocket Reference Guide](#).

Contents

- [1. Introduction](#)
 - [2. What are TCP/IP and the Internet?](#)
 - [2.1. The Evolution of TCP/IP \(and the Internet\)](#)
 - [2.2. Internet Growth](#)
 - [2.3. Internet Administration](#)
 - [2.4. Domain Names and IP Addresses \(and Politics\)](#)
 - [3. The TCP/IP Protocol Architecture](#)
 - [3.1. The Network Interface Layer](#)
 - [3.1.1. PPP](#)
 - [3.2. The Internet Layer](#)
 - [3.2.1. IP Addressing and Subnet Masks](#)
 - [3.2.2. Conserving IP Addresses: CIDR, DHCP, NAT, and PAT](#)
 - [3.2.3. The Domain Name System](#)
 - [3.2.4. ARP and Address Resolution](#)
 - [3.2.5. IP Routing: OSPF, RIP, and BGP](#)
 - [3.2.6. IP version 6](#)
 - [3.3. The Transport Layer Protocols](#)
 - [3.3.1. Ports](#)
 - [3.3.2. TCP](#)
 - [3.3.3. UDP](#)
 - [3.3.4. ICMP](#)
 - [3.3.5. TCP Logical Connections and ICMP](#)
 - [3.4. The TCP/IP Application Layer](#)
 - [3.4.1. TCP and UDP Applications](#)
 - [3.4.2. Protocol Analysis](#)
 - [3.5. Summary](#)
 - [4. Other Information Sources](#)
 - [5. Acronyms and Abbreviations](#)
 - [6. Author's Address](#)
-

1. Introduction

In 1994, I wrote the following paragraph as the introduction to this paper: "An increasing number of people are using the Internet and, many for the first time, are using the tools and utilities that at one time were only available on a limited number of computer systems (and only for really intense users!). One sign of this growth in use has been the significant number of Transmission Control Protocol/Internet Protocol (TCP/IP) and Internet books, articles, courses, and even TV shows that have become available in the last several years; there are so many such books that publishers are reluctant to authorize more because bookstores have reached their limit of shelf space! This memo provides a broad overview of the Internet and TCP/IP, with an emphasis on history, terms, and concepts. It is meant as a brief guide and starting point, referring to many other sources for more detailed information."

I was introduced to the ARPANET in 1982 and was using BITNET and CSNET throughout the 1980s. By 1992, I knew that the Internet was not just some fad but I can't honestly say that I anticipated what it has become today. This is an historical document. While I have tried to keep it up-to-date, I have also tried to maintain the language that I wrote at the time that I wrote it.

2. What are TCP/IP and the Internet?

While the TCP/IP protocols and the Internet *are* different, their histories are most definitely intertwined! This section will discuss some of the history. For additional information and insight, readers are urged to read two excellent histories of the Internet: *Casting The Net: From ARPANET to INTERNET and beyond...* by Peter Salus (Addison-Wesley, 1995) and *Where Wizards Stay Up Late: The Origins of the Internet* by Katie Hafner and Mark Lyon (Simon & Schuster, 1997). In addition, the Internet Society maintains a number of on-line "Internet history" papers at <http://www.isoc.org/internet/history/>.

2.1. The Evolution of TCP/IP (and the Internet)

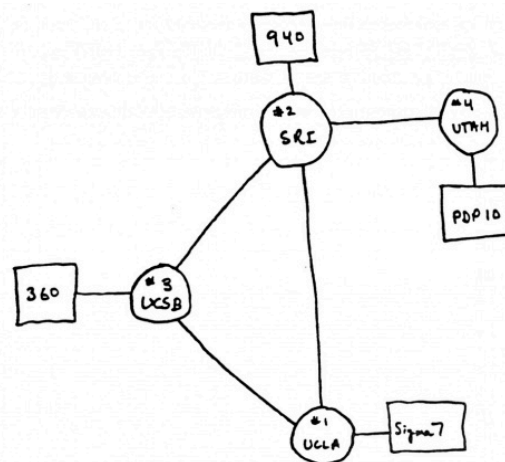
While the Internet today is recognized as a network that is fundamentally changing social, political, and economic structures, and in many ways obviating geographic boundaries, this potential is merely the realization of predictions that go back almost sixty years. In a series of memos dating back to August 1962, [J.C.R. Licklider](#) of MIT discussed his "Galactic Network" and how social interactions could be enabled through networking. The Internet certainly provides such a national and global infrastructure and, in fact, interplanetary Internet communication has already been seriously discussed.

Prior to the 1960s, what little computer communication existed comprised simple text and binary data, carried by the most common telecommunications network technology of the day; namely, circuit switching, the technology of the telephone networks for nearly a hundred years. Because most data traffic is bursty in nature (i.e., most of the transmissions occur during a very short period of time), circuit switching results in highly inefficient use of network resources.

The fundamental technology that makes the Internet work is called *packet switching*, a data network in which all components (i.e., hosts and switches) operate independently, eliminating single point-of-failure problems. In addition, network communication resources appear to be dedicated to individual users but, in fact, statistical multiplexing and an upper limit on the size of a transmitted data unit result in fast, economical networks.

In the 1960s, packet switching was ready to be discovered. [Leonard Kleinrock](#) published the first paper on packet switching theory in 1962 as his MIT Ph.D. dissertation (and the first book on the subject in 1964 while at UCLA). In 1962, [Paul Baran](#) of the Rand Corporation described a robust, efficient, store-and-forward data network in a report for the U.S. Air Force. At about the same time, [Donald Davies](#) and Roger Scantlebury suggested a similar idea from work at the National Physical Laboratory (NPL) in the U.K. The research at MIT/UCLA (1961-1967), RAND (1962-1965), and NPL (1964-1967) occurred independently and the principal researchers did not all meet together until the Association for Computing Machinery (ACM) meeting in 1967. The term *packet* was adopted from the work at NPL.

The modern Internet began as a U.S. Department of Defense (DoD) funded experiment to interconnect DoD-funded research sites in the U.S. The 1967 ACM meeting was also where the initial design for the so-called ARPANET — named for the DoD's Advanced Research Projects Agency (ARPA) — was first published by [Larry Roberts](#). In December 1968, ARPA awarded a contract to Bolt Beranek and Newman (BBN) to design and deploy a packet switching network with a proposed line speed of 50 kbps. In September 1969, the first node of the ARPANET, a Scientific Data Systems (SDS) Sigma 7, was installed at the University of California at Los Angeles (UCLA), followed monthly with nodes at Stanford Research Institute (SRI, an SDS 940), the University of California at Santa Barbara (UCSB, an IBM 360), and the University of Utah (a Digital Equipment Corp. PDP-10). With four nodes by the end of 1969, the ARPANET spanned the continental U.S. by 1971 and had connections to Europe by 1973.



The original ARPANET gave life to a number of protocols that were new to packet switching. One of the most lasting results of the ARPANET was the development of a user-network protocol that has become the standard interface between users and packet switched networks; namely, International Telecommunication Union - Telecommunication Standardization Sector (ITU-T, formerly CCITT) Recommendation X.25. This "standard" interface encouraged BBN to start Telenet, a commercial packet-switched data service, in 1974; after much renaming, Telenet became a part of Sprint's X.25 service.

The initial host-to-host communications protocol introduced in the ARPANET was called the Network Control Protocol (NCP). Over time, however, NCP proved to be incapable of keeping up with the growing network traffic load. In 1974, a new, more robust suite of communications protocols was proposed and implemented throughout the ARPANET, based upon the Transmission Control Protocol (TCP) for end-to-end network communication. But it seemed like overkill for the intermediate gateways (what we would

today call *routers*) to needlessly have to deal with an end-to-end protocol so in 1978 a new design split responsibilities between a pair of protocols; the new Internet Protocol (IP) for routing packets and device-to-device communication (i.e., host-to-gateway or gateway-to-gateway) and TCP for reliable, end-to-end host communication. Since TCP and IP were originally envisioned functionally as a single protocol, the protocol suite, which actually refers to a large collection of protocols and applications, is usually referred to simply as *TCP/IP*.

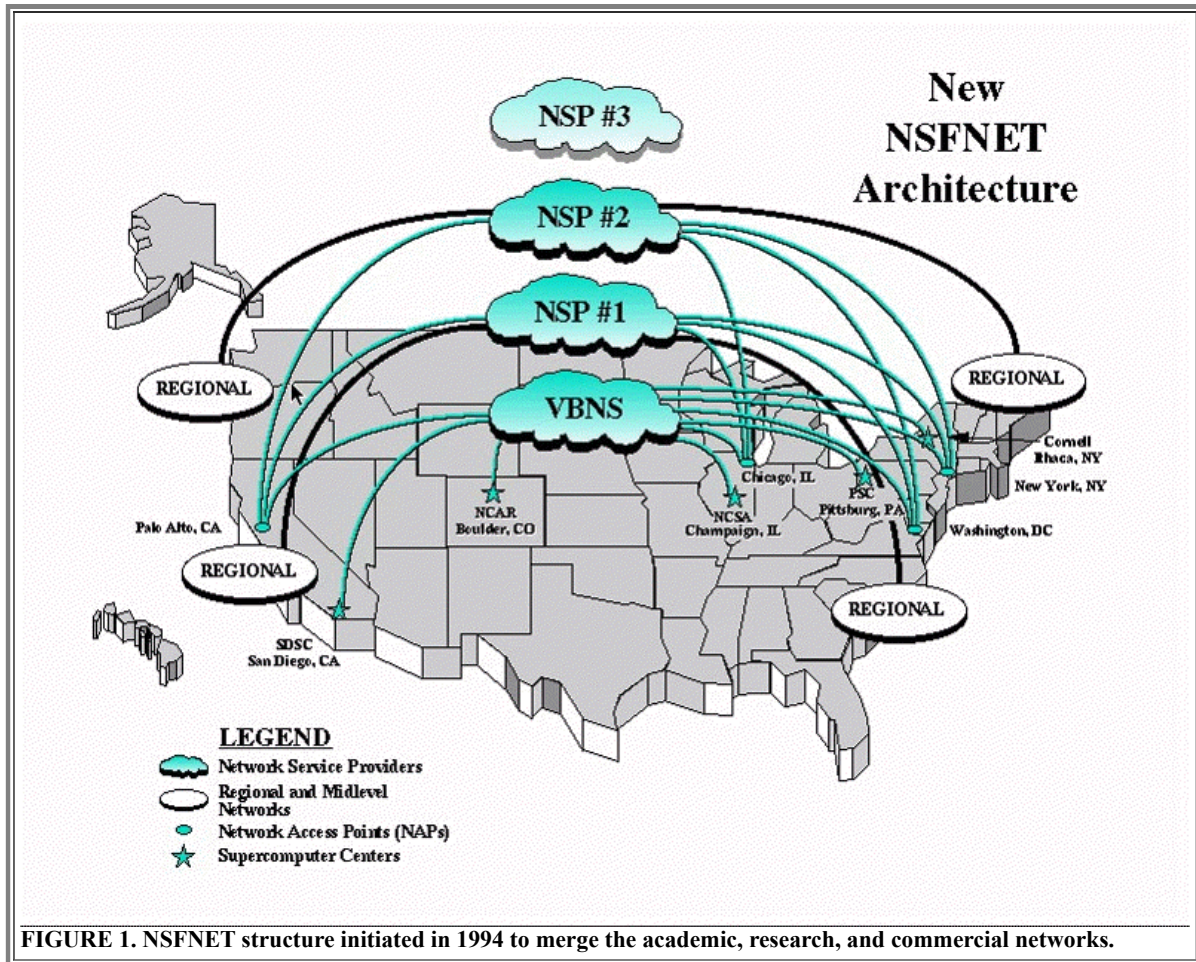
The original versions of both TCP and IP that are in common use today were written in September 1981, although both have had several modifications applied to them (in addition, the IP version 6, or IPv6, specification was released in December 1995). In 1983, the DoD mandated that all of their computer systems would use the TCP/IP protocol suite for long-haul communications, further enhancing the scope and importance of the ARPANET.

In 1983, the ARPANET was split into two components. One component, still called ARPANET, was used to interconnect research/development and academic sites; the other, called MILNET, was used to carry military traffic and became part of the Defense Data Network. That year also saw a huge boost in the popularity of TCP/IP with its inclusion in the communications kernel for the University of California's UNIX implementation, 4.2BSD (Berkeley Software Distribution) UNIX.

In 1986, the National Science Foundation (NSF) built a backbone network to interconnect four NSF-funded regional supercomputer centers and the National Center for Atmospheric Research (NCAR). This network, dubbed the NSFNET, was originally intended as a backbone for other networks, not as an interconnection mechanism for individual systems. Furthermore, the "Appropriate Use Policy" defined by the NSF limited traffic to non-commercial use. The NSFNET continued to grow and provide connectivity between both NSF-funded and non-NSF regional networks, eventually becoming the backbone that we know today as the Internet. Although early NSFNET applications were largely multiprotocol in nature, TCP/IP was employed for interconnectivity (with the ultimate goal of migration to a standardized Open Systems Interconnection [OSI] set of standards — that never appeared).

The NSFNET originally comprised 56-kbps links and was completely upgraded to T1 (1.544 Mbps) links in 1989. Migration to a "professionally-managed" network was supervised by a consortium comprising Merit (a Michigan state regional network headquartered at the University of Michigan), IBM, and MCI. Advanced Network & Services, Inc. (ANS), a non-profit company formed by IBM and MCI, was responsible for managing the NSFNET and supervising the transition of the NSFNET backbone to T3 (44.736 Mbps) rates by the end of 1991. During this period of time, the NSF also funded a number of regional Internet service providers (ISPs) to provide local connection points for educational institutions and NSF-funded sites.

In 1993, the NSF decided that it did not want to be in the business of running and funding networks, but wanted instead to go back to the funding of research in the areas of supercomputing and high-speed communications. In addition, there was increased pressure to commercialize the Internet; in 1989, a trial gateway connected MCI, CompuServe, and Internet mail services, and commercial users were now finding out about all of the capabilities of the Internet that once belonged exclusively to academic and hard-core users! In 1991, the [Commercial Internet Exchange \(CIX\) Association](#) was formed by General Atomics, Performance Systems International (PSI), and UUNET Technologies to promote and provide a commercial Internet backbone service. Nevertheless, there remained intense pressure from non-NSF ISPs to open the network to all users.



In 1994, a plan was put in place to reduce the NSF's role in the public Internet. The new structure was composed of three parts:

1. [Network Access Points \(NAPs\)](#), where individual ISPs would interconnect, as suggested in Figure 1. The NSF originally funded four such NAPs: Chicago (operated by Ameritech), New York (really Pensauken, NJ, operated by Sprint), San Francisco (operated by Pacific Bell, now SBC), and Washington, D.C. (MAE-East, operated by MFS, now part of Worldcom).
2. The [very High Speed Backbone Network Service](#), a network interconnecting the NAPs and NSF-funded centers, operated by MCI. This network was installed in 1995 and operated at OC-3 (155.52 Mbps); it was completely upgraded to OC-12 (622.08 Mbps) in 1997.
3. The [Routing Arbiter](#), to ensure adequate routing protocols for the Internet.

In addition, NSF-funded ISPs were given five years of reduced funding to become commercially self-sufficient. This funding ended by 1998 and a proliferation of additional NAPs have created a "melting pot" of services. New terminology started to refer to three *tiers* of ISP:

- A [Tier 1 network](#) referred to national ISPs, or those that have a national presence and connected to at least three of the original four NAPs. National ISPs included AT&T, Sprint, and Worldcom. Today, a Tier 1 network refers to any network that can communicate with every other network via direct peering; i.e., not having to pay for transit services.
- A [Tier 2 network](#) refers to regional ISPs, or those that have primarily a regional presence and connect to less than three of the original four NAPs. Regional ISPs include Cogent Communications, Comcast, and Spectrum. Today, a Tier 2 network refers to a provider that can reach every other network either directly or by purchasing upstream Internet services.
- *Tier 3* refers to local ISPs, or those that do not connect to a NAP but offer services via an upstream ISP.

It is worth saying a few words about the NAPs. The NSF provided major funding for the four NAPs mentioned above but they needed to have additional customers to remain economically viable. Some companies — such as then-Metropolitan Fiber Systems (MFS) — decided to build other NAP sites. One of MFS' first sites was MAE-East, where "MAE" stood for "Metropolitan Area Ethernet." MAE-East was merely a point where ISPs could interconnect which they did by buying a router and placing it at the MAE-East facility. The original MAE-East provided a 10 Mbps Ethernet LAN to interconnect the ISPs' routers, hence the name. The Ethernet LAN was eventually replaced with a 100 Mbps FDDI ring and the "E" then became "Exchange." Over the years, MFS/MCI Worldcom has added sites in San Jose, CA (MAE-West), Los Angeles, Dallas, and Houston.

Other companies also operate their own NAPs. Savvis (now CenturyLink), for example, operated an international Internet service and built more than a dozen private NAPs in North America. Many large service providers go around the NAPs entirely by creating

bilateral agreement whereby the directly route traffic coming from one network and going to the other; before their merger in 1998, for example, MCI and LDDS Worldcom had more than 10 DS-3 (44.736 Mbps) lines interconnecting the two networks.

The [North American Network Operators Group \(NANOG\)](#) provides a forum for the exchange of technical information and the discussion of implementation issues that require coordination among network service providers. Meeting three times a year, NANOG is an essential element in maintaining stable Internet services in North America. Initially funded by the NSF, NANOG currently receives funds from conference registration fees and vendor donations.

In 1988, meanwhile, the DoD and most of the U.S. Government chose to adopt OSI protocols. TCP/IP was now viewed as an interim, proprietary solution since it ran only on limited hardware platforms and OSI products were only a couple of years away (and, by the way, ran on no plaforms!). The DoD mandated that all computer communications products would have to use OSI protocols by August 1990 and use of TCP/IP would be phased out. Subsequently, the U.S. Government OSI Profile (GOSIP) defined the set of protocols that would have to be supported by products sold to the federal government and TCP/IP was not included.

Despite this mandate, development of TCP/IP continued during the late 1980s as the Internet grew. TCP/IP development had always been carried out in an open environment (although the size of this open community was small due to the small number of ARPA/NSF sites), based upon the creed, "We reject kings, presidents, and voting. We believe in rough consensus and running code" ([Dave Clark](#)). OSI products were still a couple of years away while TCP/IP became, in the minds of many, the real open systems interconnection protocol suite.

It was never the purpose of this memo to take a position on the OSI vs. TCP/IP debate — although it is absolutely clear that TCP/IP offers the primary goals of OSI; namely, a universal, non-proprietary data communications protocol. (In fact, TCP/IP does far more than was ever envisioned for OSI — or for packet switching and TCP/IP themselves, for that matter). But before TCP/IP prevailed and OSI sort of dwindled into nothingness, many efforts were made to bring the two communities together. The International Organization for Standardization (ISO) Development Environment (ISODE) was developed in 1990, for example, to provide an approach for OSI migration for the DoD. ISODE software allows OSI applications to operate over TCP/IP. During this same period, the Internet and OSI communities started to work together to bring about the best of both worlds as many TCP and IP features started to migrate into OSI protocols, particularly the OSI Transport Protocol class 4 (TP4) and the Connectionless Network Layer Protocol (CLNP), respectively. Finally, a report from the National Institute for Standards and Technology (NIST) in 1994 suggested that GOSIP should incorporate TCP/IP and drop the "OSI-only" requirement. As it was, many industry observers have pointed out that OSI represented the ultimate example of a *sliding window*; OSI protocols were "two years away" pretty consistently between the mid-1980s to mid-1990s.

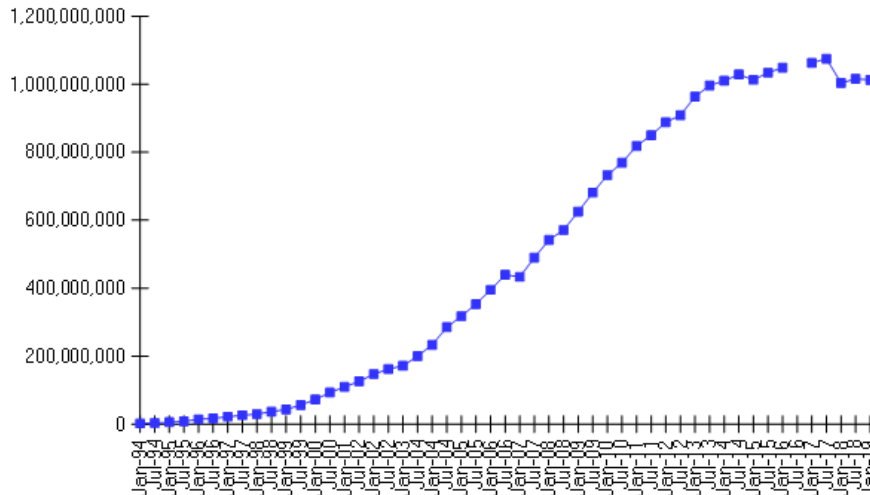
None of this is meant to suggest that the NSF isn't funding Internet-class research networks anymore. That is just the function of [Internet2](#), a consortium of nearly 600 universities, corporations, and non-profit research oriented organizations working in partnership to develop and deploy advanced network applications and technologies for the next generation Internet. Goals of Internet2 are to create a leading edge network capability for the national research community, enable the development of new Internet-based applications, and to quickly move these new network services and applications to the commercial sector.

2.2. Internet Growth

In Douglas Adams' *The Hitchhiker's Guide to the Galaxy* (Pocket Books, 1979), the hitchhiker describes outer space as being "...big. Really big. ...vastly hugely mind-bogglingly big..." A similar description can be applied to the Internet. To paraphrase the hitchhiker, you may think that your 750 node LAN is big, but that's just peanuts compared to the Internet.

The ARPANET started with four nodes in 1969 and grew to just under 600 nodes before it was split in 1983. The NSFNET also started with a modest number of sites in 1986. After that, the network experienced literally exponential growth. Internet growth between 1981 and 1991 is documented in "Internet Growth (1981-1991)" ([RFC 1296](#)).

Internet Domain Survey Host Count



Source: Internet Systems Consortium (www.isc.org)

The Internet Software Consortium hosts the [Internet Domain Survey](#) (with technical support from Network Wizards, who originated the survey). According to their chart, the Internet had nearly 30 million reachable hosts by January 1998 and just over a billion in July 2017. Dedicated residential access methods, such as cable modem and asymmetrical digital subscriber line (ADSL) technologies, are undoubtedly the reason that this number shot up during the 2000-2010 decade and Internet of Things (IoT) devices will add more exponential growth into the 2020s. During the boom-1990s, the Internet was growing at a rate of about a new network attachment every half-hour, interconnecting hundreds of thousands of networks. It was estimated that the Internet was doubling in size every ten to twelve months and traffic was doubling every 100 days (for 1000% annual growth). For the last several years, the number of nodes has been growing at a rate of about 50% annually and traffic continues to keep pace with that growth.

| WORLD INTERNET USAGE AND POPULATION STATISTICS JUNE 30, 2018 - Update | | | | | | |
|--|------------------------|-----------------------|-----------------------------|---------------------------|------------------|------------------|
| World Regions | Population (2018 Est.) | Population % of World | Internet Users 30 June 2018 | Penetration Rate (% Pop.) | Growth 2000-2018 | Internet Users % |
| Africa | 1,287,914,329 | 16.9 % | 464,923,169 | 36.1 % | 10,199 % | 11.0 % |
| Asia | 4,207,588,157 | 55.1 % | 2,062,197,366 | 49.0 % | 1,704 % | 49.0 % |
| Europe | 827,650,849 | 10.8 % | 705,064,923 | 85.2 % | 570 % | 16.8 % |
| Latin America / Caribbean | 652,047,996 | 8.5 % | 438,248,446 | 67.2 % | 2,325 % | 10.4 % |
| Middle East | 254,438,981 | 3.3 % | 164,037,259 | 64.5 % | 4,894 % | 3.9 % |
| North America | 363,844,662 | 4.8 % | 345,660,847 | 95.0 % | 219 % | 8.2 % |
| Oceania / Australia | 41,273,454 | 0.6 % | 28,439,277 | 68.9 % | 273 % | 0.7 % |
| WORLD TOTAL | 7,634,758,428 | 100.0 % | 4,208,571,287 | 55.1 % | 1,066 % | 100.0 % |

NOTES: (1) Internet Usage and World Population Statistics estimates in June 30, 2018. (2) CLICK on each world region name for detailed regional usage information. (3) Demographic (Population) numbers are based on data from the United Nations Population Division. (4) Internet usage information comes from data published by Nielsen Online, by the International Telecommunications Union, by GfK, by local ICT Regulators and other reliable sources. (5) For definitions, navigation help and disclaimers, please refer to the Website Surfing Guide. (6) The information from this website may be cited, giving the due credit and placing a link back to www.internetworldstats.com. Copyright © 2018, Miniwatts Marketing Group. All rights reserved worldwide.

It is also worth noting that the vast majority of Internet growth since 2000 has been outside of North America, Europe, and Oceania/Australia. The reason is simple — those regions of the world had explosive Internet growth in the 1990s! The [Internet World Stats](#) site is about the best to start learning about the demographics of the Internet.

And what of the original ARPANET? It grew smaller and smaller during the late 1980s as sites and traffic moved to the Internet, and was decommissioned in July 1990. Cerf & Kahn ("[Selected ARPANET Maps](#)," *Computer Communications Review*, October 1990) re-printed a number of network maps documenting the growth (and demise) of the ARPANET.

2.3. Internet Administration

The Internet is a collection of autonomous, crash-independent networks. The Internet has no single owner, yet everyone owns (a portion of) the Internet. The Internet has no central operator, yet everyone operates (a portion of) the Internet. The Internet has been

compared to anarchy, but some claim that it is not nearly that well organized!

Some central authority is required for the Internet, however, to manage those things that can only be managed centrally, such as addressing, naming, protocol development, standardization, etc. Among the significant Internet authorities are:

- The [Internet Society \(ISOC\)](#), chartered in 1992, is a non-governmental international organization providing coordination for the Internet, and its internetworking technologies and applications. ISOC also provides oversight and communications for the Internet Activities Board.
- The [Internet Architecture \(nee Activities\) Board \(IAB\)](#) governs administrative and technical activities on the Internet.
- The [Internet Engineering Task Force \(IETF\)](#) is one of the two primary bodies of the IAB. The IETF's working groups have primary responsibility for the technical activities of the Internet, including writing specifications and protocols. The impact of these specifications is significant enough that ISO accredited the IETF as an international standards body at the end of 1994. RFCs [2028](#) and [2031](#) describe the organizations involved in the IETF standards process and the relationship between the IETF and ISOC, respectively, while [RFC 8179](#) (BCP 79) describes the IETF policies about Intellectual Property Rights (IPR).
- The [Internet Engineering Steering Group \(IESG\)](#) is the other body of the IAB. The IESG provides direction to the IETF.
- The [Internet Research Task Force \(IRTF\)](#) comprises a number of long-term reassert groups, promoting research of importance to the evolution of the future Internet.
- The [Internet Engineering Planning Group \(IEPG\)](#) coordinates worldwide Internet operations. This group also assists Internet Service Providers (ISPs) to interoperate within the global Internet.
- The [Forum of Incident Response and Security Teams \(FIRST\)](#) is the coordinator of a number of Computer Emergency Response Teams (CERTs) representing many countries, governmental agencies, and ISPs throughout the world. Internet network security is greatly enhanced and facilitated by the FIRST member organizations.
- The [World Wide Web Consortium \(W3C\)](#) is not an Internet administrative body, per se, but since October 1994 has taken a lead role in developing common protocols for the World Wide Web to promote its evolution and ensure its interoperability. W3C has more than 400 Member organizations internationally. The W3C, then, is leading the technical evolution of the Web, having already developed more than 20 technical specifications for the Web's infrastructure.

2.4. Domain Names and IP Addresses (and Politics)

Although not directly related to the administration of the Internet for operational purposes, the assignment of Internet domain names (and IP addresses) is the subject of some controversy and a lot of current activity. Internet hosts use a hierarchical naming structure comprising a top-level domain (TLD), domain and subdomain (optional), and host name. The IP address space, and all TCP/IP-related numbers, have historically been managed by the [Internet Assigned Numbers Authority \(IANA\)](#). Domain names are assigned by the TLD naming authority; until April 1998, the [Internet Network Information Center \(InterNIC\)](#) had overall authority of these names, with NICs around the world handling non-U.S. domains. The InterNIC was also responsible for the overall coordination and management of the Domain Name System (DNS), the distributed database that reconciles host names and IP addresses on the Internet.

The InterNIC is an interesting example of the changes that occurred in the Internet during the early days of commercialization and the withdrawal of the NSF from the network operation. Starting in early 1993, [Network Solutions, Inc. \(NSI\)](#) operated the registry tasks of the InterNIC on behalf of the NSF and had exclusive registration authority for the *.com*, *.org*, *.net*, and *.edu* domains. NSI's contract ran out in April 1998 and was extended several times because no other agency was in place to continue the registration for those domains. In October 1998, it was decided that NSI would remain the sole administrator for those domains but that a plan needed to be put into place so that users could register names with other firms (pending final approval from the domain administrator). In addition, NSI's contract was extended to September 2000, although the registration business was opened to competition in June 1999. In addition, when NSI's original InterNIC contract expired, IP address assignments moved to a new entity called the [American Registry for Internet Numbers \(ARIN\)](#). (And NSI itself was purchased by VeriSign in March 2000.)

The newest body to handle governance of global Top Level Domain (gTLD) registrations is the [Internet Corporation for Assigned Names and Numbers \(ICANN\)](#). Formed in October 1998, ICANN is the organization designated by the U.S. [National Telecommunications and Information Administration \(NTIA\)](#) to administer the DNS. Although surrounded in some early controversy (which is well beyond the scope of this paper!), ICANN has received wide industry support. ICANN has created several Support Organizations (SOs) to create policy for the administration of its areas of responsibility, including domain names (DNSO), IP addresses (ASO), and protocol parameter assignments (PSO).

In April 1999, ICANN announced that five companies had been selected to be part of this new *competitive* Shared Registry System for the *.com*, *.net*, and *.org* domains. The concept was that these five organizations could accept applications for names in these gTLD name spaces, with final approval by ICANN. By the end of 1999, ICANN had added an additional 29 registrars and there are several hundred registrars accredited by ICANN today. Definitive ICANN registrar accreditation information can be found at the [ICANN-Accredited Registrars](#) page.

The hierarchical structure of domain names is best understood if the domain name is read from right-to-left. Internet host names end with a top-level domain name. Worldwide generic top-level domains (gTLDs) include the original *.com*, *.edu*, *.gov*, *.int*, *.mil*, *.net*, and *.org*. In November 2000, the first new set of TLDs were approved by ICANN, namely, *.aero*, *.biz*, *.coop*, *.info*, *.museum*, *.name*, and *.pro*:

- [.aero](#): Reserved for the global aviation community, sponsored by Societe Internationale de Telecommunications Aeronautiques SC (SITA)
- [.asia](#): Reserved for the Pan-Asia and Asia Pacific region, sponsored by DotAsia Organisation

- [.biz](#): Restricted to businesses, operated by NeuStar, Inc.
- [.cat](#): Reserved for the Catalan linguistic and cultural community, sponsored by Fundació puntCat.
- [.com](#): Used by commercial organizations, operated by VeriSign, Inc.
- [.coop](#): Reserved for business cooperatives, sponsored by Dot Cooperation LLC
- [.info](#): A general-purpose domain, operated by Afilias Limited
- [.jobs](#): Reserved for the human resource management community, sponsored by Employ Media LLC
- [.mobi](#): Reserved for consumers and providers of mobile products and services, sponsored by mTLD Top Level Domain, Ltd.
- [.museum](#): Reserved for museums and related persons, sponsored by the Museum Domain Management Association International (MDI)
- [.name](#): Restricted to individuals, operated by Verisign Information Services, Inc.
- [.net](#): Originally reserved for network service providers, operated by VeriSign, Inc.
- [.org](#): Reserved for non-profit organizations, operated by Public Interest Registry
- [.pro](#): Reserved for licensed professionals, operated by Registry Services Corporation (dba RegistryPro)
- [.tel](#): Reserved for individuals and businesses to store and manage their contact information in the DNS, sponsored by Telnic Limited
- [.travel](#): Reserved for entities whose primary area of activity is in the travel industry, sponsored by Tralliance Registry Management Company, LLC

Today, there are several hundred gTLDs and more are being added every day, including many using non-Latin characters (aka Internationalized Domain Names). An up-to-date list of gTLDs can be found at ICANN's [Delegated Strings](#) page.

Of course, the gTLDs that were there at the beginning still remain:

- [.edu](#): Educational institutions; largely limited to 4-year colleges and universities from about 1994 to 2001, but also includes some community colleges; administered by EDUCAUSE)
- [.int](#): Organizations established by international treaty, administered by IANA
- [.gov](#): U.S. Federal government agencies (managed by the U.S. General Services Administration, including the *fed.us* domain)
- [.mil](#): U.S. military (managed by the U.S. Department of Defense Network Information Center)

Other top-level domain names use the two-letter country codes defined in [ISO standard 3166](#); *munnari.oz.au*, for example, is the address of the 1990's Internet gateway to Australia and *myo.inst.keio.ac.jp* is a host at the Science and Technology Department of Keio University in Yokohama, Japan. Other ISO 3166-based domain country codes are *ca* (Canada), *de* (Germany), *es* (Spain), *fr* (France), *gb* (Great Britain) [NOTE: For some historical reasons, the TLD *.gb* is rarely used; the TLD *.uk* (United Kingdom) seems to be preferred although UK is not an official ISO 3166 country code.], *ie* (Ireland), *il* (Israel), *mx* (Mexico), and *us* (United States). It is important to note that there is not necessarily any correlation between a country code and where a host is actually physically located.

In addition, some countries allow their country codes to be used for other, commercial purposes. By way of example, the "[Web site](#)" domain *.ws* actually belongs to the country of Samoa and the [television \(tv\)](#) domain belongs to the country of Tuvalu.

Perhaps the most authoritative and up-to-date listings can be found at the "Domain name registries around the world" page maintained by [UNINETT Norid](#).

A host name generally contains at least three parts, namely, the name of the computer, the domain name, and the gTLD/ccTLD. For example, the host name *www.codb.us*, for example, is assigned to a computer named *www* (which just happens to be a Web server) in the City of Daytona Beach (*codb*) domain, within the U.S. ccTLD (*us*) namespace. The host name *mail.erau.edu* refers to a host (*mail*) in the Embry-Riddle Aeronautical University (*erau*) domain within the education gTLD (*edu*) space. Guidelines for selecting host names is the subject of [RFC 1178](#).

There are several registries responsible for blocks of IP addresses and domain naming policies around the globe. The [American Registry for Internet Numbers \(ARIN\)](#), was originally responsible for the Americas (western hemisphere) and parts of Africa. In 2002, the [Latin American and Caribbean Internet Addresses Registry \(LACNIC\)](#) was officially recognized and now covers Central and South America, as well as some Caribbean nations. The [African Regional Internet Registry \(AfrINIC\)](#) has responsibility for sub-Saharan Africa. At this time, ARIN only covers North America. The European and Asia-Pacific naming registries are managed by [Réseaux IP Européen \(RIPE\)](#) and the [Asia-Pacific NIC \(APNIC\)](#), respectively.

These authorities, in turn, delegate most of the country TLDs to [national registries](#) (such as RNP in Brazil and NIC-Mexico), which have ultimate authority to assign local domain names. An excellent overview of the recent history and anticipated future of the registry system can be found in "[Development of the Regional Internet Registry System](#)" (D. Karrenberg et al.) in the *IP Journal*, 4(4).

Different countries may organize the country-based subdomains in any way that they want. Many countries use a subdomain hierarchy similar to the original gTLDs, so that *.com.mx* and *.edu.mx* are the suffixes for commercial and educational institutions in Mexico, and *.co.uk* and *.ac.uk* are the suffixes for commercial and educational institutions in the United Kingdom.

The *us* domain was originally organized on the basis of geography or function. Geographical names in the *us* name space used names of the form *entity-name.city-telegraph-code.state-postal-code.us*. The domain name *cnri.reston.va.us*, for example, refers to the Corporation for National Research Initiatives in Reston, Virginia. Functional branches are also reserved within the name space for schools (K12), community colleges (CC), technical schools (TEC), state government agencies (STATE), councils of governments (COG), libraries (LIB), museums (MUS), and several other generic types of entities. Domain names in the state

government name space usually take the form *department.state.state-postal-code.us* (e.g., the domain name *dps.state.vt.us* points to the Vermont Department of Public Safety). The K12 name space can vary widely, usually using the form *school.school-district.k12.state-postal-code.us* (e.g., the domain *ccs.cssd.k12.vt.us* refers to the Charlotte Central School in the Chittenden South School District which happens to be in Charlotte, Vermont.) Today, the *.us* domain is used less hierarchically, but more information about the historical format can be found in [RFC 1480](#).

The biggest change to the TLD process was the introduction, in 2010, of TLDs that do not use Latin characters. These [Internationalized Domain Names](#) are described in detail at ICANN's Web site. To learn more about making non-Latin character codes readable, read up on Punycode ([RFC 3492](#)) and/or visit one of the many converter pages, such as the one at [punycoder.com](#).

More information about TLDs, the registration process, and new TLDs can be found at the ICANN [New TLD Program](#) Web page.

Domain name ownership — or, more precisely, the validity of a claim to own a domain name — has long had an issue of violation of trademark, service mark, or copyright. Consider this example from the 2001 era. A common Microsoft tag line is *Where Would You Like To Go Today?* It so happens that the domain name *wherewouldyouliketogotoday.com* was registered to The Eagles Nest in Corfu, NY. I don't know anything about The Eagles Nest of Corfu, NY but it should not be mistaken for either Eagles Nest Enterprises of Grapevine, TX (the owner of *eaglesnest.com*) nor The Eagles Nest Internet Services of Newark, OH (owner of *theeaglesnest.com*).

In any case, suppose that Microsoft decided that someone else using their service mark was not in their best interest and they pursued the issue; could they wrestle that domain name away from another registrant? Today's general rule of thumb is that if an organization believes that its name or mark is being used in someone else's domain name in an unfair or misleading way, then they can take legal action against the name holder and the assignment of the name will be held up pending the outcome of the legal action. More information about this issue can be found at ICANN's [Uniform Domain-Name Dispute-Resolution Policy](#) Web page. By the way, this is, of course, the genesis behind the industry of cybersquatting, where someone registers a domain name hoping that someone else will buy it from them later on!

And what about IP addresses? Prior to the widespread use of CIDR (pronounced "cider"; see [Section 3.2.1](#)), individual organizations were assigned an address (usually a Class C) and domain name at the same time. In general, the holder of the domain name owned the IP address and if they changed ISP, routing tables throughout the Internet were updated.

After 1994, domain name and IP number ownership were separated. Today, ISPs are assigned addresses in blocks called *CIDR blocks*. A customer today, whether they already own a domain name or are obtaining a new one, will be assigned an IP address from the ISP's CIDR block. If the customer changes ISP, they have to relinquish the IP address.

A good overview of the naming and addressing procedures can be found in [RFC 2901](#), titled "Guide to Administrative Procedures of the Internet Infrastructure."

3. The TCP/IP Protocol Architecture

TCP/IP is most commonly associated with the Unix operating system. While developed separately, they have been historically tied, as mentioned above, since 4.2BSD Unix started bundling TCP/IP protocols with the operating system. Nevertheless, TCP/IP protocols are available for all widely-used operating systems today and native TCP/IP support is provided in OS/2, OS/400, all Windows versions since Windows 9x, and all Linux and Unix variants.

Figure 2 shows the TCP/IP protocol architecture; this diagram is by no means exhaustive, but shows the major protocol and application components common to most commercial TCP/IP software packages and their relationship.

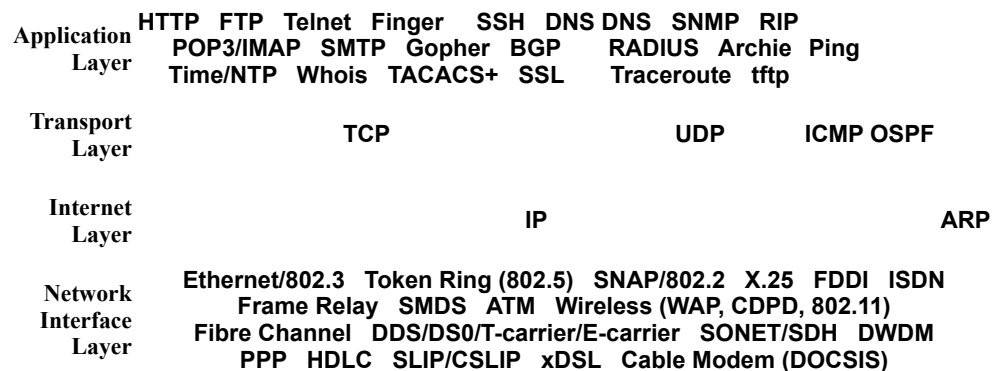


FIGURE 2. Abbreviated TCP/IP protocol stack.

The sections below will provide a brief overview of each of the layers in the TCP/IP suite and the protocols that compose those layers. A large number of books and papers have been written that describe all aspects of TCP/IP as a protocol suite, including detailed information about use and implementation of the protocols. Some good TCP/IP references are:

- *TCP/IP Illustrated, Volume I: The Protocols* by W.R. Stevens (Addison-Wesley, 1994)

- *Troubleshooting TCP/IP* by Mark Miller (John Wiley & Sons, 1999)
- *Guide to TCP/IP, 2/e* by Laura A. Cappell and Ed Tittel (Thomson Course Technology, 2004)
- *TCP/IP: Architecture, Protocols, and Implementation with IPv6 and IP Security* by S. Feit (McGraw-Hill, 2000)
- *Internetworking with TCP/IP, Vol. 1: Principles, Protocols, and Architecture, 2/e*, by D. Comer (Prentice-Hall, 1991)
- "TCP/IP Tutorial" by T.J. Socolofsky and C.J. Kale ([RFC 1180](#), Jan. 1991)
- "[TCP/IP and tcpdump Pocket Reference Guide](#)," developed by the author for The SANS Institute

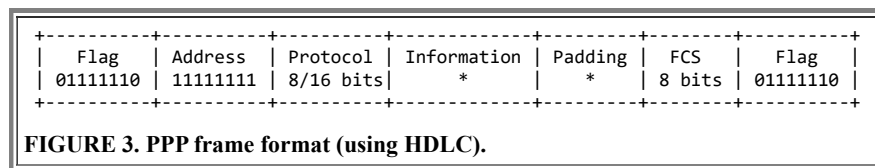
3.1. The Network Interface Layer

The TCP/IP protocols have been designed to operate over nearly any underlying local or wide area network technology. Although certain accommodations may need to be made, IP messages can be transported over all of the technologies shown in the figure, as well as numerous others. It is beyond the scope of this paper to describe most of these underlying protocols and technologies.

Two of the underlying network interface protocols, however, are particularly relevant to TCP/IP. The Serial Line Internet Protocol (SLIP, [RFC 1055](#)) and Point-to-Point Protocol (PPP, [RFC 1661](#)), respectively, may be used to provide data link layer protocol services where no other underlying data link protocol may be in use, such as in leased line or dial-up environments. Most commercial TCP/IP software packages for PC-class systems include these two protocols. With SLIP or PPP, a remote computer can attach directly to a host server and, therefore, connect to the Internet using IP rather than being limited to an asynchronous connection.

3.1.1. PPP

It is worth spending a little bit of time discussing PPP because of its importance in Internet access today. As its name implies, PPP was designed to be used over point-to-point links. In fact, it is the prevalent IP encapsulation scheme for dedicated Internet access as well as dial-up access. One of the significant strengths of PPP is its ability to negotiate a number of things upon initial connection, including passwords, IP addresses, compression schemes, and encryption schemes. In addition, PPP provides support for simultaneous multiple protocols over a single connection, an important consideration in those environments where dial-up users can employ either IP or another network Layer protocol. Finally, in environments such as ISDN, PPP supports inverse multiplexing and dynamic bandwidth allocation via the Multilink-PPP (ML-PPP) described in RFCs [1990](#) and [2125](#).



PPP generally uses an HDLC-like (bit-oriented protocol) frame format as shown in Figure 3, although RFC 1661 does not demand use of HDLC. HDLC defines the first and last two fields in the frame:

- *Flag*: The 8-bit pattern "01111110" used to delimit the beginning and end of the transmission.
- *Address*: For PPP, uses the 8-bit broadcast address, "11111111".
- *Frame Check Sequence (FCS)*: An 8-bit remainder from a cyclic redundancy check (CRC) calculation, used for bit error detection.

RFC 1661 actually describes the use of the three other fields in the frame:

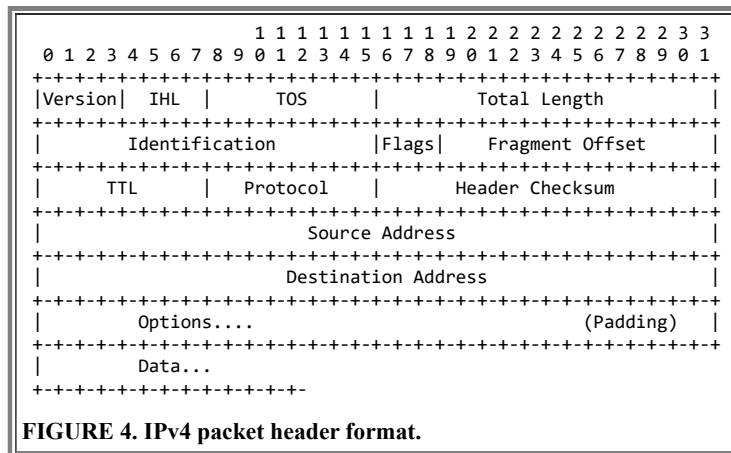
- *Protocol*: An 8- or 16-bit value that indicates the type of datagram carried in this frame's Information field. This field can indicate use of a particular Network Layer protocol (such as IP, IPX, or DDP), a Network Control Protocol (NCP) in support of one of the Network Layer protocols, or a PPP Link-layer Control Protocol (LCP). The entire list of possible PPP values in this field can be found in the [IANA list of PPP protocols](#).
- *Information*: Contains the datagram for the protocol specified in the Protocol field. This field is zero or more octets in length, up to a (default) maximum of 1500 octets (although a different value can be negotiated).
- *Padding*: Optional padding to add length to the Information field. May be required in some implementations to ensure some minimum frame length and/or to ensure some alignment on computer word boundaries.

The operation of PPP is as follows:

1. After the link is physically established, each host sends LCP packets to configure and test the data link. It is here where the maximum frame length, authentication protocol (Password Authentication Protocol, PAP, or Challenge-Handshake Authentication Protocol, CHAP), link quality protocol, compression protocol, and other configuration parameters are negotiated. Authentication, *if* it used, will occur after the link has been established.
2. After the link is established, one or more Network Layer protocol connections are configured using the appropriate NCP. If IP is to be used, for example, it will be set up using PPP's IP Control Protocol (IPCP). Once each of the Network Layer protocols has been configured, datagrams from those protocols can be sent over the link. Control protocols may be used for IP, IPX (NetWare), DDP (AppleTalk), DECnet, and more.
3. The link will remain configured for communications until LCP and/or NCP packets close the link down.

3.2. The Internet Layer

The Internet Protocol version 4 ([RFC 791](#)), provides services that are roughly equivalent to the OSI Network Layer. IP provides a datagram (connectionless) transport service across the network. This service is sometimes referred to as *unreliable* because the network does not guarantee delivery nor notify the end host system about packets lost due to errors or network congestion. IP datagrams contain a message, or one fragment of a message, that may be up to 65,535 bytes (octets) in length. IP does not provide a mechanism for flow control.

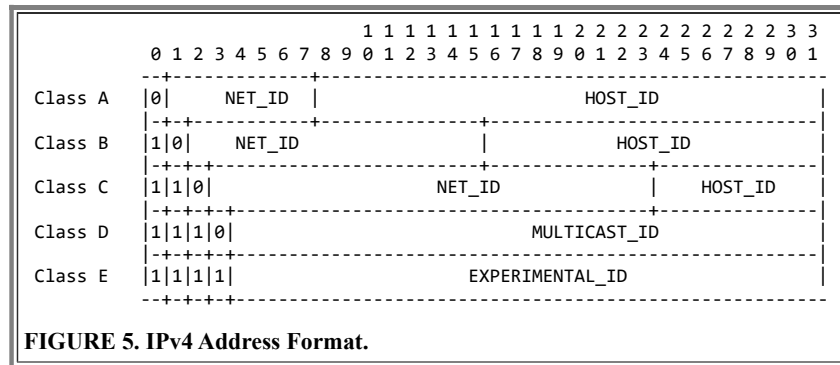


The basic IPv4 packet header format is shown in Figure 4. The format of the diagram is consistent with the RFC; bits are numbered from left-to-right, starting at 0. Each row represents a single 32-bit word; note that an IP header will be at least 5 words (20 bytes) in length. The fields contained in the header, and their functions, are:

- *Version*: Specifies the IP version of the packet. The most common version of IP in end-user system is version 4, so this field will contain the binary value 0100. IP version 6 is more commonly available, however, so this field would have the value 0110 in an IPv6 packet and the rest of the packet will be very different from what is shown here (see [Section 3.2.6](#)). [NOTE: Actually, many IP version numbers have been assigned besides 4 and 6; see the [IANA's list of IP Version Numbers](#).]
- *Internet Header Length (IHL)*: Indicates the length of the datagram header in 32 bit (4 octet) words. A minimum-length header is 20 octets, so this field always has a value of at least 5 (0101). Since the maximum value of this field is 15, the IP Header can be no longer than 60 octets.
- *Type of Service (TOS)*: Allows an originating host to request different classes of service for packets it transmits. Although not generally supported today in IPv4, the TOS field can be set by the originating host in response to service requests across the Transport Layer/Internet Layer service interface, and can specify a service priority (0-7) or can request that the route be optimized for either cost, delay, throughput, or reliability.
- *Total Length*: Indicates the length (in bytes, or octets) of the entire packet, including both header and data. Given the size of this field, the maximum size of an IP packet is 64 KB, or 65,535 bytes. In practice, packet sizes are limited to the maximum transmission unit (MTU).
- *Identification*: Used when a packet is fragmented into smaller pieces while traversing the Internet, this identifier is assigned by the transmitting host so that different fragments arriving at the destination can be associated with each other for reassembly.
- *Flags*: Also used for fragmentation and reassembly. The first bit is called the More Fragments (MF) bit, and is used to indicate the last fragment of a packet so that the receiver knows that the packet can be reassembled. The second bit is the Don't Fragment (DF) bit, which suppresses fragmentation. The third bit is unused (and always set to 0).
- *Fragment Offset*: Indicates the position of this fragment in the original packet. In the first packet of a fragment stream, the offset will be 0; in subsequent fragments, this field will indicate the offset in increments of 8 bytes.
- *Time-to-Live (TTL)*: A value from 0 to 255, indicating the number of hops that this packet is allowed to take before discarded within the network. Every router that sees this packet will decrement the TTL value by one; if it gets to 0, the packet will be discarded.
- *Protocol*: Indicates the higher layer protocol contents of the data carried in the packet; options include ICMP (1), TCP (6), UDP (17), or OSPF (89). A complete list of IP protocol numbers can be found at the [IANA's list of Protocol Numbers](#). An implementation-specific list of supported protocols can be found in the `protocol` file, generally found in the `/etc` (Linux/Unix/Mac OS X) or `c:\windows\system32\drivers\etc` (Windows) directory.
- *Header Checksum*: Carries information to ensure that the received IP header is error-free. Remember that IP provides an *unreliable* service and, therefore, this field only checks the IP header rather than the entire packet.
- *Source Address*: IP address of the host sending the packet.
- *Destination Address*: IP address of the host intended to receive the packet.
- *Options*: A set of options which may be applied to any given packet, such as sender-specified source routing or security indication. The option list may use up to 40 bytes (10 words), and will be padded to a word boundary; IP options are taken from the [IANA's list of IP Option Numbers](#).

3.2.1. IP Addresses

IP (version 4) addresses are 32 bits in length (Figure 5). They are typically written as a sequence of four numbers, representing the decimal value of each of the address bytes. Since the values are separated by periods, the notation is referred to as *dotted decimal*. A sample IP address is 208.162.106.17.



IP addresses are hierarchical for routing purposes and are subdivided into two subfields. IPv4 address have two levels of hierarchy, namely the Network Identifier (NET_ID) and Host Identifier (HOST_ID). The NET_ID subfield identifies the TCP/IP subnetwork connected to the Internet. The NET_ID is used for high-level routing between networks, much the same way as the country code, city code, or area code is used in the telephone network. The HOST_ID subfield indicates the specific host within a subnetwork.

To accommodate different size networks, IPv4 historically defined several *address classes*. Classes A, B, and C are used for host addressing and the only difference between the classes is the length of the NET_ID subfield:

- A Class A address has an 8-bit NET_ID and 24-bit HOST_ID. Class A addresses are intended for very large networks and can address up to 16,777,214 ($2^{24}-2$) hosts per network. The first bit of a Class A address is a 0 and the NETID occupies the first byte, so there are only 128 (2^7) possible Class A NETIDs. In fact, the first digit of a Class A address will be between 1 and 126, and only about 90 or so Class A addresses have been assigned.
- A Class B address has a 16-bit NET_ID and 16-bit HOST_ID. Class B addresses are intended for moderate sized networks and can address up to 65,534 ($2^{16}-2$) hosts per network. The first two bits of a Class B address are 10 so that the first digit of a Class B address will be a number between 128 and 191; there are 16,384 (2^{14}) possible Class B NETIDs. The Class B address space has long been threatened with being used up and it is has been very difficult to get a new Class B address for some time.
- A Class C address has a 24-bit NET_ID and 8-bit HOST_ID. These addresses are intended for small networks and can address only up to 254 (2^8-2) hosts per network. The first three bits of a Class C address are 110 so that the first digit of a Class C address will be a number between 192 and 223. There are 2,097,152 (2^{21}) possible Class C NETIDs and most addresses assigned to networks today are Class C (or sub-Class C!).

The remaining two address classes are used for special functions only and are not commonly assigned to individual hosts. Class D addresses may begin with a value between 224 and 239 (the first 4 bits are 1110), and are used for IP multicasting (i.e., sending a single datagram to multiple hosts); the IANA maintains a list of [Internet Multicast Addresses](#). Class E addresses begin with a value between 240 and 255 (the first 4 bits are 1111), and are reserved for experimental use.

Several address values are reserved and/or have special meaning. A HOST_ID of 0 (as used above) is a dummy value reserved as a place holder when referring to an entire subnetwork; the address 208.162.106.0, then, refers to the Class C address with a NET_ID of 208.162.106. A HOST_ID of all ones (usually written "255" when referring to an all-ones byte, but also denoted as "-1") is a broadcast address and refers to all hosts on a network. A NET_ID value of 127 is used for loopback testing and the specific host address 127.0.0.1 refers to the *localhost*.

Several NET_IDs have been reserved in [RFC 1918](#) for private network addresses and packets will not be routed over the Internet to these networks. Reserved NET_IDs include the Class A address 10.0.0.0 (formerly assigned to ARPANET), the sixteen Class B addresses 172.16.0.0-172.31.0.0, and the 256 Class C addresses 192.168.0.0-192.168.255.0. (These addresses are not routable over the Internet because all ISPs have agreed not to route them. (They are unusable as public addresses just as a telephone number such as 802-555-1234 is unusable on the public telephone network because the telephone service providers have "agreed" not to use the 555 exchange.)

An additional addressing tool is the *subnet mask*. Subnet masks are used to indicate the portion of the address that identifies the network (and/or subnetwork) for routing purposes. The subnet mask is written in dotted decimal and the number of 1s indicates the significant NET_ID bits. For "classful" IP addresses, the subnet mask and number of significant address bits for the NET_ID are:

| Class | Subnet Mask | Number of Bits |
|-------|---------------|----------------|
| A | 255.0.0.0 | 8 |
| B | 255.255.0.0 | 16 |
| C | 255.255.255.0 | 24 |

Depending upon the context and literature, subnet masks may be written in dotted decimal form or just as a number representing the number of significant address bits for the NET_ID. Thus, 208.162.106.17 255.255.255.0 and 208.162.106.17/24 both refer to a Class C NET_ID of 208.162.106. Some, in fact, might refer to this 24-bit NET_ID as a "slash-24."

Subnet masks can also be used to subdivide a large address space into subnetworks or to combine multiple small address spaces. In the former case, a network may subdivide their address space to define multiple logical networks by segmenting the HOST_ID subfield into a Subnetwork Identifier (SUBNET_ID) and (smaller) HOST_ID. For example, user assigned the Class B address space 172.16.0.0 could segment this into a 16-bit NET_ID, 4-bit SUBNET_ID, and 12-bit HOST_ID. In this case, the subnet mask for Internet routing purposes would be 255.255.0.0 (or "/16"), while the mask for routing to individual subnets within the larger Class B address space would be 255.255.240.0 (or "/20").

But how a subnet mask work? To determine the subnet portion of the address, we simply perform a bit-by-bit logical AND of the IP address and the mask. Consider the following example: suppose we have a host with the IP address 172.20.134.164 and a subnet mask 255.255.0.0. We write out the address and mask in decimal and binary as follows:

```

      172.020.134.164      10101100.00010100.10000110.10100100
AND  255.255.000.000      11111111.11111111.00000000.00000000
-----
      172.020.000.000      10101100.00010100.00000000.00000000

```

From this we can easily find the NET_ID 172.20.0.0 (and can also infer the HOST_ID 134.164).

As an aside, most ISPs use a /30 address for the WAN links between the network and the customer. The router on the customer's network will generally have two IP addresses; one on the LAN interface using an address from the customer's public IP address space and one on the WAN interface leading back to the ISP. Since the ISP would like to be able to ping both sides of the router for testing and maintenance, having an IP address for each router port is a good idea.

By using a /30 address, a single Class C address can be broken up into 64 smaller addresses. Here's an example. Suppose an ISP assigns a particular customer the address 24.48.165.130 and a subnet mask 255.255.255.252. That would look like the following:

```

      024.048.165.130      00011000.00110000.10100101.10000010
AND  255.255.255.252      11111111.11111111.11111111.11111100
-----
      024.048.165.128      00011000.00110000.10100101.10000000

```

So we find the NET_ID to be 24.48.165.128. Since there's a 30-bit NET_ID, we are left with a 2-bit HOST_ID; thus, there are four possible host addresses in this subnet: 24.48.165.128 (00), .129 (01), .130 (10), and .131 (11). The .128 address isn't used because it is all-zeroes; .131 isn't used because it is all-ones. That leave .129 and .130, which is ok since we only have two ends on the WAN link! So, in this case, the customer's router might be assigned 24.48.165.130/30 and the ISP's end of the link might get 24.48.165.129/30. Use of this subnet mask is very common today (so common that there is a proposal to allow the definition of 2-address NET_IDs specifically for point-to-point WAN links).

A very good IP addressing tutorial can be found in Chuck Semeria's *Understanding IP Addressing: Everything You Ever Wanted to Know* ([Part 1](#) | [Part 2](#) | [Part 3](#)). If you are really interested in subnet masks, there are a number of subnet calculators on the Internet, including [TunnelsUp.com's Subnet Calculator](#) and [Sohopen's IP Subnet Calculator](#).

Most Internet protocols specify that addresses be supplied in the form of a fully-qualified host name or an IP address in dotted decimal form. However, spammers and others have found a way to obfuscate IP addresses by supplying the IP address as a single large **decimal** number. Remember that IPv4 addresses are 32-bit quantities. We write the address in dotted decimal for the convenience of humans; the computer still interprets dotted decimal as a 32-bit quantity.

Therefore, writing the address as a single large decimal number will still allow the computer to see the address as a 32-bit number. For that reason, the following URLs will all take you to the same Web page (on most browsers):

- <https://www.garykessler.net>
- <https://207.204.17.246>
- <https://3486257654>

3.2.2. Conserving IP Addresses: CIDR, DHCP, NAT, and PAT

The use of class-based (or *classful*) addresses in IP is one of the reasons that IP address exhaustion has been a concern since the early 1990s. Consider an organization, for example, that needs 1000 IP addresses. A Class C address is obviously too small so a Class B address would get assigned. But a Class B address offers more than 64,000 address, so more than 63,000 addresses are wasted in this assignment.

An alternative approach is to assign this organization a block four Class C addresses, such as 192.168.128.0, 192.168.129.0, 192.168.130.0, and 192.168.131.0. By using a 22-bit subnet mask 255.255.252.0 (or "/22") for routing to this "block," the NET_ID assigned to this organization is 192.168.128.0 (this would generally be denoted 192.168.128.0/22).

This use of variable-size subnet masks is called *Classless Interdomain Routing (CIDR)*, described in RFCs [1518](#) and [4632](#). In the example above, routing information for what is essentially four Class C addresses can be specified in a single router table entry.

But this concept can be expanded even more. CIDR is an important contribution to the Internet because it has dramatically limited the size of the Internet backbone's routing tables. Today, IP addresses are not assigned strictly on a first-come, first-serve basis, but have been preallocated to various numbering authorities around the world. The numbering authorities in turn, assign blocks of addresses to major (or first-tier) ISPs; these address blocks are called *CIDR blocks*. An ISP's customer (which includes ISPs that are customers of a first-tier ISP) will be assigned an IP NET_ID that is part of the ISP's CIDR block. So, for example, let's say that the *Tomoka ISP* has a CIDR block containing the 256 Class C addresses in the range 196.168.0.0-196.168.255.0. This range of addresses could be represented in a routing table with the single entry 196.168.0.0/16. Once a packet hits the Tomoka ISP, it will be routed it to the correct end destination within the ISP's network.

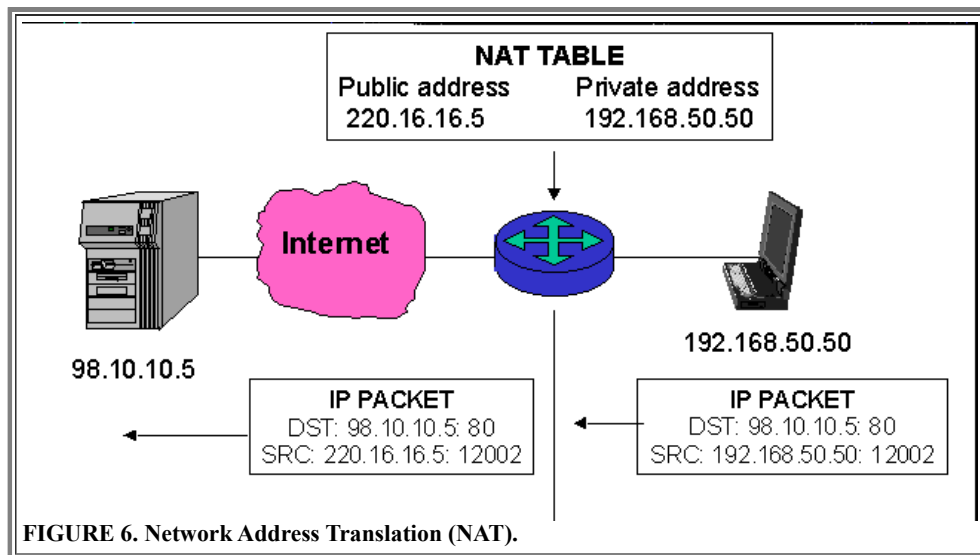
But don't stop now! By shrinking the size of the subnet mask so that a single NET_ID refers to multiple addresses (resulting in shrinking router tables), we could *extend* the size of the subnet mask to actually assign to an organization something smaller than a Class C address. As the Class C address space grew nearer and nearer to being exhausted, users were under increasing pressure to accept assignment of these *sub-Class C addresses*. An organization with just a few servers, for example, might be assigned, say, 64 addresses rather than the full 256. The standard subnet mask for a Class C is 24 bits, yielding a 24-bit NET_ID and 8-bit HOST_ID. If we use a "/26" mask (255.255.255.192), we can assign the same "Class C" to four different users, each getting 1/4 of the address space (and a 6-bit HOST_ID). So, for example, the IP address space 208.162.106.0 might be assigned as follows:

| NET_ID | HOST_ID range | Valid HOST_IDs |
|-----------------|---------------|----------------|
| 208.162.106.0 | 0-63 | 1-62 |
| 208.162.106.64 | 64-127 | 65-126 |
| 208.162.106.128 | 128-191 | 129-190 |
| 208.162.106.192 | 192-255 | 193-254 |

Note that in ordinary Class C usage, we would lose two addresses from the space — 0 and 255 — because addresses of all 0s and all 1s cannot be assigned as a HOST_ID. In the usage above, we would lose eight addresses from this space, because 0, 64, 128, and 192 have an all 0s HOST_ID and 63, 127, 191, and 255 have an all 1s HOST_ID. Each user, then, has 62 addresses that can be assigned to hosts.

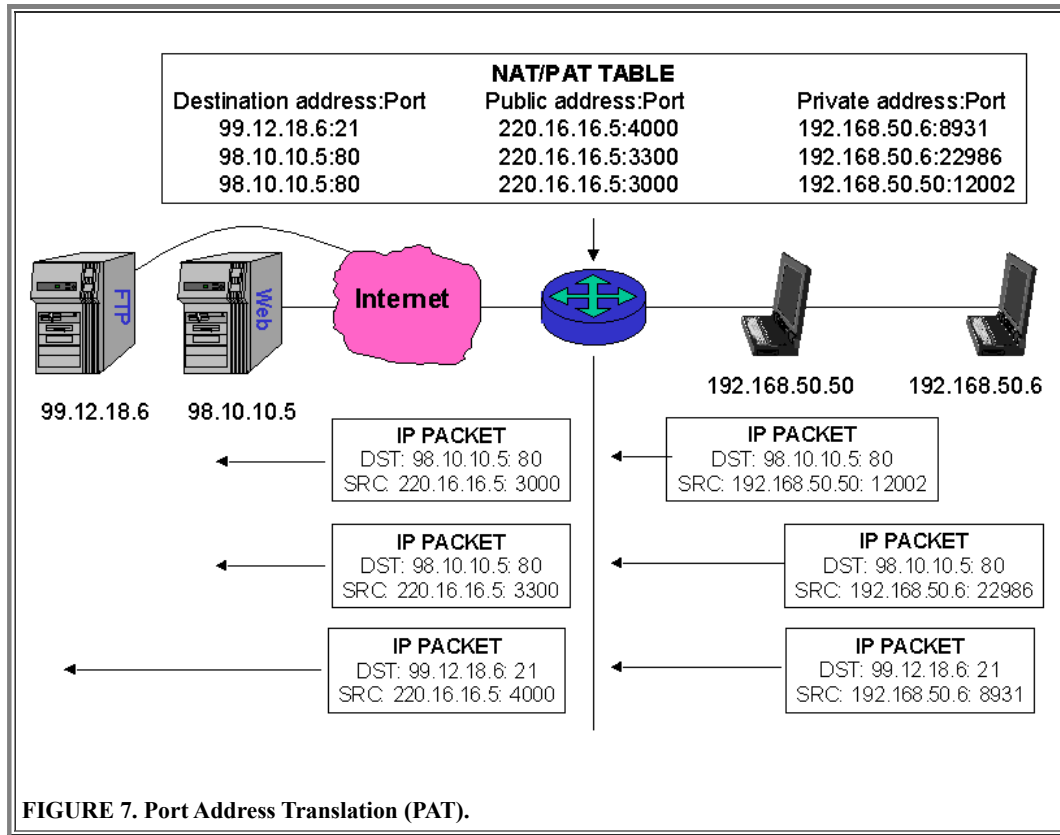
The pressure on the Class C address space is continuing in intensity. Today, the pressure is not only to limit the number of addresses assigned, but organizations need to show *why* they need as many addresses as they want. Consider a company with 64 hosts and 3 servers. The ISP may request that that company only obtain 32 IP addresses. The rationale: the 3 servers need 3 addresses but the other hosts might be able to "share" the remaining pool of 27 addresses (recall that we lost HOST_ID addresses 0 and 31).

A pool of IP addresses can be shared by multiple hosts using a mechanism called Network Address Translation (NAT). NAT, described in [RFC 3022](#), is typically implemented in hosts, proxy servers, or routers. The scheme works because every host on the user's network can be assigned an IP address from the pool of RFC 1918 private addresses; since these addresses are never seen on the Internet, this is not a problem.



Consider the scenario shown in Figure 6. When the user accesses a Web site on the Internet, the NAT server will translate the "private" IP address of the host (192.168.50.50) into a "public" IP address (220.16.16.5) from the pool of assigned addresses. NAT works because of the assumption that, in this example, no more than 27 of the 64 hosts will ever be accessing the Internet at a single time.

But suppose that assumption is wrong. Another enhancement, called Port Address Translation (PAT) or Network Address Port Translation (NAPT), allows multiple hosts to share a single IP address by using different "port numbers" (ports are described more in [Section 3.3](#)).



Port numbers are used by higher layer protocols (e.g., TCP and UDP) to identify a higher layer application. A TCP connection, for example, is uniquely identified on the Internet by the four values (aka *4-tuple*) <source IP address, source port, destination IP address, destination port>. The server's port number is defined by the standards while client port numbers can be any number greater than 1023. The scenario in Figure 7 shows the following three connections:

- The client with the "private" IP address 192.168.50.50 (using port number 12002) connects to a Web server at address 98.10.10.5 (port 80).
- The client with the "private" IP address 192.168.50.6 (using port number 22986) connects to the same Web server at address 98.10.10.5 (port 80).
- The client with the "private" IP address 192.168.50.6 (using port number 8931) connects to an FTP server at address 99.12.18.6 (port 21).

PAT works in this scenario as follows. The router (running PAT software) can assign both local hosts with the same "public" IP address (220.16.16.5) and differentiate between the three packet flows by the source port.

A final note about NAT and PAT. Both of these solutions work and work fine, but they require that *every* packet be buffered, disassembled, provided with a new IP address, a new checksum calculated, and the packet reassembled. In addition, PAT requires that a new port number be placed in the higher layer protocol data unit and new checksum calculated at the protocol layer above IP, too. The point is that NAT, and particularly PAT, results in a tremendous performance hit.

One advantage of NAT is that it makes IP address renumbering a thing of the past. If a customer has an IP NET_ID assigned from its ISP's CIDR block and then they change ISPs, they will get a new NET_ID. With NAT, only the servers need to be renumbered.

Another way to deal with renumbering is to dynamically assign IP addresses to host systems using the Dynamic Host Configuration Protocol (DHCP). DHCP is also an excellent solution for those environments where users move around frequently; it prevents the user from having to reconfigure their system when they move from, say, the Los Angeles office network to the New York office. For an introduction to DHCP, see [RFC 2131](#) or "[The Dynamic Host Configuration Protocol \(DHCP\) and Windows NT](#)" by G. Kessler and C. Monaghan.

3.2.3. The Domain Name System

While IPv4 addresses are 32 bits in length, most users do not memorize the numeric addresses of the hosts to which they attach; instead, people are more comfortable with host names. Most IP hosts, then, have both a numeric IP address and a name. While this is convenient for people, however, the name must be translated back to a numeric address for routing purposes.

Earlier discussion in this paper described the domain naming structure of the Internet. In the early ARPANET, every host maintained a file called `hosts` that contained a list of all hosts, which included the IP address, host name, and alias(es). This was an adequate measure while the ARPANET was small and had a slow rate of growth, but was not a scalable solution as the network

grew. [NOTE: The `hosts` file can be found on Linux, Max OS X, and Unix systems in the `/etc` directory and on Microsoft Windows systems in the `c:\$systemroot\system32\drivers\etc` directory.]

To handle the fast rate of new names on the network, the Domain Name System (DNS) was created. The DNS is a distributed database containing host name and IP address information for all domains on the Internet. There is a single *authoritative name server* for every domain that contains all DNS-related information about the domain; each domain also has at least one secondary name server that also contains a copy of this information. Thirteen *root servers* around the globe maintain a list of all of these authoritative name servers. Although most of the root servers have multiple instances around the globe to improve performance and minimize vulnerability to attack, most of the primary DNS root servers are in the U.S. with the remainder in Asia and Europe).

When a host on the Internet needs to obtain a host's IP address based upon the host's name, a DNS request is made by the initial host to a local name server. The local name server may be able to respond to the request with information that is either configured or cached at the name server; if necessary information is not available, the local name server forwards the request to one of the root servers. The root server, then, will determine an appropriate name server for the target host and the DNS request will be forwarded to the domain's name server.

Name server data files contain the following types of records including:

- *A-record*: An address record maps a hostname to an IPv4 (32 bit) address.
- *AAAA-record*: A quad-A address record maps a hostname to an IPv6 (128 bit) address.
- *PTR-record*: A pointer record maps an IP address to a hostname.
- *NS-record*: A name server record lists the authoritative name server(s) for a given domain.
- *MX-record*: A mail exchange record lists the mail servers for a given domain. As an example, consider the author's e-mail address, `gck@garykessler.net`. The "garykessler.net" portion of the address is a domain name, not a host name, and mail has to be sent to a specific host. The MX-record(s) in the `garykessler.net` name database specifies the host `mx01.register.com` is the preferred mail server for this domain. (Register.com hosts the garykessler.net domain.)
- *CNAME-record*: Canonical name records provide a mechanism of assigning aliases to host names, so that a single host with a IP address can be known by multiple names.

The use of a CNAME record might be a bit confusing since it appears to be redundant. The purpose is to minimize the amount of reconfiguration — and possible introduction of errors — if you have to change the IP address of a system. Consider this example. Suppose you have a server with the IP address 192.0.2.107 that is known as `foo.example.com`, `mail.example.com`, `name.example.com`, and `www.example.com`. You could set this up in DNS with four A-records, as follows:

```
foo.example.com. IN A 192.0.2.107
mail.example.com. IN A 192.0.2.107
name.example.com. IN A 192.0.2.107
www.example.com. IN A 192.0.2.107
```

Note that if you ever need to change the IP address of this system, you need to change all four A-records. If you use CNAME-records, however, you only need one A-record:

```
foo.example.com. IN A 192.0.2.107
mail.example.com. IN CNAME foo.example.com.
name.example.com. IN CNAME foo.example.com.
www.example.com. IN CNAME foo.example.com.
```

In this case, if you ever need to change the address of the server, you only need to change a single A-record. To be fair, however, the former method also has its advantages; if you ever migrate, say, `www.example.com` to another server but leave the other hosts on the original server, it's quite easy to just change a server's address by changing its A-record.

The IANA administers the root zone (i.e., ".") of the DNS. It maintains a list of all authoritative zone administrations at its [Root Zone Database](#).

More information about the DNS can be found in [DNS and BIND](#), 5th ed. by P. Albitz and C. Liu (O'Reilly & Associates) and ["Setting up Your own DNS"](#) by G. Kessler. The concepts, structure, and delegation of the DNS are described in RFCs [1034](#) and [1591](#). In addition, the IANA maintains a list of [DNS parameters](#).

[ANOTHER NOTE: For Microsoft NetBIOS applications, the moral equivalent to the DNS is the Windows Internet Name Service (WINS), used to reconcile the NetBIOS name of a computer (e.g., `\ALTAMONT`) to an IP address. A local WINS database can be created in the `LMHOSTS` file.]

3.2.4. ARP and Address Resolution

Early IP implementations ran on hosts commonly interconnected by Ethernet local area networks (LAN). Every transmission on the LAN contains the local network, or medium access control (MAC), address of the source and destination nodes. MAC addresses are 48-bits in length and are non-hierarchical, so routing cannot be performed using the MAC address. MAC addresses are never the same as IP addresses.

When a host needs to send a datagram to another host on the same network, the sending application must know both the IP and MAC addresses of the intended receiver; this is because the destination IP address is placed in the IP packet and the destination MAC address is placed in the LAN MAC protocol frame. (If the destination host is on another network, the sender will look instead for the MAC address of the default gateway, or router.)

Unfortunately, the sender's IP process may not know the MAC address of the intended receiver on the same network. The Address Resolution Protocol (ARP), described in [RFC 826](#), provides a mechanism so that a host can learn a receiver's MAC address when knowing only the IP address. The process is actually relatively simple: the host sends an ARP Request packet in a frame containing the MAC broadcast address; the ARP request advertises the destination IP address and asks for the associated MAC address. The station on the LAN that recognizes its own IP address will send an ARP Response with its own MAC address. As [Figure 2](#) shows, ARP messages are carried directly in the LAN frame and ARP is an independent protocol from IP. The IANA maintains a list of all [ARP parameters](#).

Other address resolution procedures have also been defined, including:

- Reverse ARP (RARP), which allows a disk-less processor to determine its IP address based on knowing its own MAC address
- Inverse ARP (InARP), which provides a mapping between an IP address and a frame relay virtual circuit identifier
- ATMARP and ATMInARP provide a mapping between an IP address and ATM virtual path/channel identifiers.
- LAN Emulation ARP (LEARP), which maps a recipient's ATM address to its LAN Emulation (LE) address (which takes the form of an IEEE 802 MAC address).

[NOTE: IP hosts maintain a cache storing recent ARP information. The ARP cache can be viewed from a Unix, Linux, or DOS command line using the `arp -a` command.]

3.2.5. IP Routing: OSPF, RIP, and BGP

As an OSI Network Layer protocol, IP has the responsibility to route packets. It performs this function by looking up a packet's destination IP NET_ID in a routing table and forwarding based on the information in the table. But it is *routing protocols*, and *not* IP, that populate the routing tables with routing information. There are three routing protocols commonly associated with IP and the Internet, namely, RIP, OSPF, and BGP.

OSPF and RIP are primarily used to provide routing within a particular domain, such as within a corporate network or within an ISP's network. Since the routing is *inside* of the domain, these protocols are generically referred to as *interior gateway protocols*.

The Routing Information Protocol version 2 (RIP-2), described in [RFC 2453](#), describes how routers will exchange routing table information using a distance-vector algorithm. With RIP, neighboring routers periodically exchange their entire routing tables. RIP uses hop count as the metric of a path's cost, and a path is limited to 16 hops. Unfortunately, RIP has become increasingly inefficient on the Internet as the network continues its fast rate of growth. Designed to be used within a single organization's network, RIP is an *interior gateway protocol*. Many classic LAN networks used RIP or a variant, including NetWare, AppleTalk, VINES, and DECnet. The IANA maintains a list of [RIP message types](#).

The Open Shortest Path First (OSPF) protocol is another *interior gateway protocol*. OSPF uses a link state routing algorithm that is more robust than RIP, converges faster, requires less network bandwidth, and is better able to scale to larger networks. With OSPF, a router broadcasts only changes in its links' status rather than entire routing tables, making it more robust and scalable than RIP. OSPF Version 2 is described in [RFC 2328](#). (The "open" in the name of OSPF refers to the non-proprietary nature of the algorithm rather than an "open shortest path.")

The Border Gateway Protocol version 4 (BGP-4) is an *exterior gateway protocol* commonly used between two ISPs or between a customer site and ISP if there are multiple links. BGP is a distance vector protocol, like RIP, but unlike almost all other distance vector protocols, BGP tables store the actual route to the destination network. BGP-4 also supports policy-based routing, which allows a network's administrator to create routing policies based on political, security, legal, or economic issues rather than technical ones. BGP-4 also supports CIDR. BGP-4 is described in [RFC 4271](#), while [RFC 1772](#) describes use of BGP in the Internet. In addition, the IANA maintains a list of [BGP parameters](#).

As an alternative to using a routing protocol, the routing table can be maintained using "static routing." One example of static routing is the configuration of a default gateway at a host system; if the host needs to send an IP packet off of the local LAN segment, it is just blindly forwarded to the default gateway (router). Edge routers, too, commonly use static routing; the single router connecting a site to an ISP, for example, will usually just have a static routing table entry indicating that all traffic leaving the local LAN be forwarded to the ISP's access router. Since there's only a single path into the ISP, a routing protocol is hardly necessary.

All IP hosts and routers maintain a table that lists the most up-to-date routing information that that device knows. On a Windows system, you can examine the routing table by issuing a `route print` command; on Unix systems, use `netstat -r`.

[Figure 2](#) shows the protocol relationship of RIP, OSPF, and BGP to IP. A RIP message is carried in a UDP datagram which, in turn, is carried in an IP packet. An OSPF message, on the other hand, is carried directly in an IP datagram. BGP messages, in a total departure, are carried in TCP segments over IP. Although all of the TCP/IP books mentioned above discuss IP routing to some level of detail, *Routing in the Internet* by Christian Huitema is one of the best available references on this specific subject.

3.2.6. IP version 6

The official version of IP that has been in use since the early 1980s is *version 4*. Due to the tremendous growth of the Internet and new emerging applications, it was recognized that a new version of IP was becoming necessary. In late 1995, IP version 6 (IPv6) was entered into the Internet Standards Track. In 2017, [RFC 8200/STD 86](#) was released as the latest IPv6 specification, also elevating [RFC 4443](#) describing ICMPv6 to an Internet Standard (STD 89).

IPv6 is designed as an evolution from IPv4, rather than a radical change, so as to allow for an orderly transition. Primary areas of change relate to:

- Increasing the IP address size to 128 bits, essentially making it future-proof
- Header format simplification, eliminating superfluous fields
- Better support for traffic types with different quality-of-service objectives
- Security extensions to support authentication, data integrity, and data confidentiality

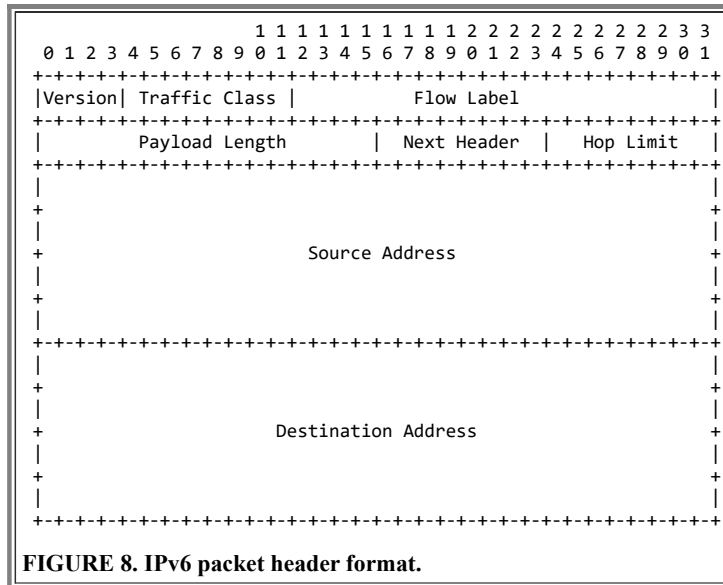


FIGURE 8. IPv6 packet header format.

The basic IPv6 packet header format is shown in Figure 8:

- *Version*: Specifies the IP version of the packet, in this case "6" (0110).
- *Traffic Class*: Eight-bit value used for traffic management.
- *Flow Label*: 20-bit value defining a *flow*, a mechanism to support different quality-of-service traffic flows (e.g. real-time vs. non-real-time traffic, or interactive data vs. bulk data).
- *Payload Length*: 16-bit value indicating the length (in bytes, or octets) of packet payload (i.e., the portion of the packet following the header).
- *Next Header*: Eight-bit value identifying the type of header immediately following the IPv6 header. Similar to the IPv4 *Protocol* field, except that this allows IPv6 with the flexibility to use only additional fields that it needs to use, thus reducing packet size (e.g., why include an empty *Routing* field if routing information is not going to appear in the packet).
- *Hop Limit*: Eight-bit value used like IPv4's *TTL* field; is assigned a value from 0 to 255 by the sender, indicating the number of hops that this packet is allowed to take before discarded within the network. Every router that sees this packet will decrement the TTL value by one; if it gets to 0, the packet will be discarded.
- *Source Address*: 128-bit IPv6 address of the host sending the packet.
- *Destination Address*: 128-bit IPv6 address of the host intended to receive the packet.

The architecture and structure of IPv6 addresses is described in [RFC 4291](#). As noted earlier, IPv6 addresses are 128 bits in length, yielding 2^{128} — or 340,282,366,920,938,463,463,374, 607,431,768,211,456 — possible host addresses. Even if you can't comprehend the magnitude of this number, it is pretty impressive given that there are only estimated to be 2^{80} stars in the galaxy!!!

IPv6 addresses are written in eight blocks of two-octets, separated by a colon (:), called *hex group* (or *hextet*) notation, e.g., 2001:0db8:3241:0000:0000:9a8f:00c9:951e. Leading zeroes within a group may be dropped, so you might see an address written as 2001:db8:3241:0:0:9a8f:c9:951e. As an additional shorthand, one or more consecutive all-zero groups can be replaced with a "::" (only once in an address), yielding 2001:db8:3241::9a8f:c9:951e.

In July 1999, the IANA delegated the initial IPv6 address space to the worldwide regional registries in order to begin immediate worldwide deployment of IPv6 addresses. Without going into too much gory detail about IPv6 addresses, suffice it to say that subnet masks can be of any length. Since the suggested address allocation is to assign every subnet as a /64, users will have a lot of address space and NAT/PAT become a thing of the past. Here is one example of the IPv6 address hierarchy:

- Entire IPv6 Address Space: 0000:: - FFFF::
- IANA space allocated to registries: 2001:: /16

- o APNIC: 2400::/12
- o ARIN: 2600::/12
- o LACNIC: 2800::/12
- o RIPE NCC: 2A00::/12
- o AfriNIC: 2C00::/12
- One country's address space: 2001:0D00::/23
- One ISP's address space: 2001:0DB8::/32
- One customer's space: 2001:0DB8:0326::/48
- One user's space: 2001:0DB8:0326:79F2:/64

More information about IPv6 addressing can be found at Wikipedia's [IPv6 Address](#) article.

Finally, many of the common IPv4 command line tools have IPv6 functionality. In general, Windows uses the same command but employs a -6 switch while Unix (including Linux and Mac OS X) employs a variant command name by adding a 6 to the end. Examples include:

| IPv4 (Windows) | IPv4 (Unix) | IPv6 (Windows) | IPv6 (Unix) |
|-----------------------|-------------|--------------------------|-------------|
| ping | | ping -6 | ping6 |
| tracert | tracert | tracert -6 | tracert6 |
| nslookup (set type=A) | | nslookup (set type=AAAA) | |
| whois | | whois | |
| | curl | | curl6 |
| | wget | | wget6 |

For more information about IPv6, check out:

- *IPng: Internet Protocol Next Generation* by Scott Bradner and Allison Mankin (Addison-Wesley, 1996)
- *IPv6: The New Internet Protocol* by Christian Huitema (Prentice-Hall, 1996).
- "[IPv6: The Next Generation Internet Protocol](#)" by Gary Kessler.
- *IPng and the TCP/IP Protocols* by Stephen Thomas (John Wiley & Sons, 1996)
- [IPv6](#) (Wikipedia)
- [6bone Testbed](#) (Wikipedia)

3.3. The Transport Layer Protocols

The TCP/IP protocol suite comprises two protocols that correspond roughly to the OSI Transport and Session Layers; these protocols are called the Transmission Control Protocol and the User Datagram Protocol (UDP).

3.3.1. Ports

Higher-layer applications are referred to by a port identifier in TCP/UDP messages. The port identifier and IP address together form a *socket*, and the end-to-end communication between two hosts is uniquely identified on the Internet by the four-tuple (source port, source address, destination port, destination address).

Port numbers are specified by a 16-bit number. Port numbers in the range 0-1023 are called *Well Known Ports*. These port numbers are assigned to the server side of an application and, on most systems, can only be used by processes with a high level of privilege (such as root or administrator). Port numbers in the range 1024-49151 are called *Registered Ports*, and these are numbers that have been publicly defined as a convenience for the Internet community to avoid vendor conflicts. Server or client applications can use the port numbers in this range. The remaining port numbers, in the range 49152-65535, are called *Dynamic and/or Private Ports* and can be used freely by any client or server.

Some well-known port numbers include:

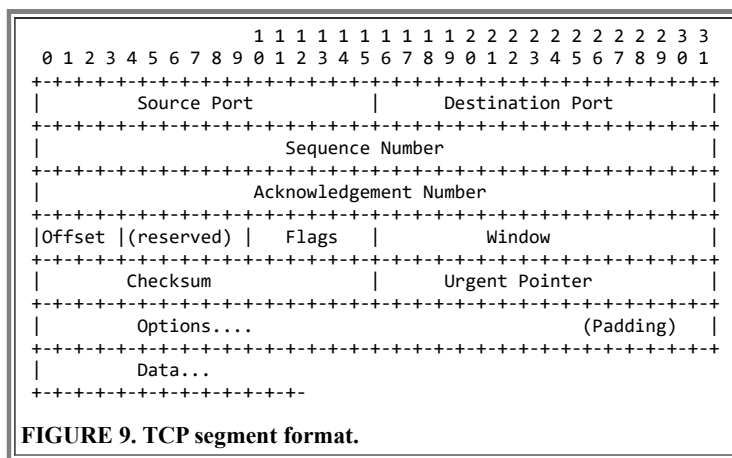
| Port # | Common Protocol | Service | Port # | Common Protocol | Service |
|--------|-----------------|-------------|--------|-----------------|-------------|
| 7 | TCP | echo | 80 | TCP | http |
| 9 | TCP | discard | 110 | TCP | pop3 |
| 13 | TCP | daytime | 111 | TCP | sunrpc |
| 19 | TCP | chargen | 119 | TCP | nntp |
| 20 | TCP | ftp-control | 123 | UDP | ntp |
| 21 | TCP | ftp-data | 137 | UDP | netbios-ns |
| 23 | TCP | telnet | 138 | UDP | netbios-dgm |
| 25 | TCP | smtp | 139 | TCP | netbios-ssn |

| | | | | | |
|----|---------|--------|-------|-----|---------------------|
| 37 | UDP | time | 143 | TCP | imap |
| 43 | TCP | whois | 161 | UDP | snmp |
| 53 | TCP/UDP | dns | 162 | UDP | snmp-trap |
| 67 | UDP | bootps | 179 | TCP | bgp |
| 68 | UDP | bootpc | 443 | TCP | https (http/ssl) |
| 69 | UDP | tftp | 520 | UDP | rip |
| 70 | TCP | gopher | 1080 | TCP | socks |
| 79 | TCP | finger | 33434 | UDP | traceroute |

A complete list of port numbers that have been assigned can be found in the [IANA's list of Port Numbers](#). An implementation-specific list of supported port numbers and services can be found in the `services` file, generally found in the `/etc` (Linux/Unix), `c:\windows` (Windows 9x, ME), or `c:\$systemroot\system32\drivers\etc` (Windows NT, 2000, et seq.) directory.

3.3.2. TCP

TCP, described in [RFC 793](#), provides a virtual circuit (connection-oriented) communication service between hosts across the network. TCP includes rules for formatting messages, establishing and terminating virtual circuits, sequencing, flow control, and error correction. Most of the applications in the TCP/IP suite operate over the *reliable* transport service provided by TCP.



The TCP data unit is called a *segment*; the name is due to the fact that TCP does not recognize messages, per se, but merely sends a block of bytes from the byte stream between sender and receiver. The fields of the segment (Figure 9) are:

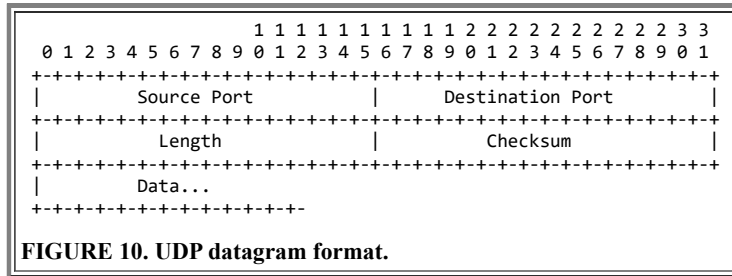
- **Source Port and Destination Port:** Identify the source and destination ports to identify the end-to-end connection and higher-layer application.
- **Sequence Number:** Contains the sequence number of this segment's first data byte in the overall connection byte stream; since the sequence number refers to a byte count rather than a segment count, sequence numbers in contiguous TCP segments are not numbered sequentially.
- **Acknowledgment Number:** Used by the sender to acknowledge receipt of data; this field indicates the sequence number of the next byte expected from the receiver.
- **Data Offset:** Points to the first data byte in this segment; this field, then, indicates the segment header length.
- **Control Flags:** A set of flags that control certain aspects of the TCP virtual connection. The flags include:
 - **Urgent Pointer Field Significant (URG):** When set, indicates that the current segment contains urgent (or high-priority) data and that the Urgent Pointer field value is valid.
 - **Acknowledgment Field Significant (ACK):** When set, indicates that the value contained in the Acknowledgment Number field is valid. This bit is usually set, except during the first message during connection establishment.
 - **Push Function (PSH):** Used when the transmitting application wants to force TCP to immediately transmit the data that is currently buffered without waiting for the buffer to fill; useful for transmitting small units of data.
 - **Reset Connection (RST):** When set, immediately terminates the end-to-end TCP connection.
 - **Synchronize Sequence Numbers (SYN):** Set in the initial segments used to establish a connection, indicating that the segments carry the initial sequence number.
 - **Finish (FIN):** Set to request normal termination of the TCP connection in the direction this segment is traveling; completely closing the connection requires one FIN segment in each direction.
- **Window:** Used for flow control, contains the value of the *receive window size* which is the number of transmitted bytes that the sender of this segment is willing to accept from the receiver.
- **Checksum:** Provides bit error detection for the TCP segment. The checksum field covers the TCP segment header and data, as well as a 96-bit *pseudo header* that contains the IP header Source Address, Destination Address, and Protocol fields, as well as the TCP segment length.
- **Urgent Pointer:** Urgent data is information that has been marked as high-priority by a higher layer application; this data, in turn, usually bypasses normal TCP buffering and is placed in a segment between the header and "normal" data. The Urgent

Pointer, valid when the URG flag is set, indicates the position of the first octet of nonexpedited data in the segment.

- *Options*: Used at connection establishment to negotiate a variety of options; maximum segment size (MSS) is the most commonly used option and, if absent, defaults to an MSS of 536. Another option is Selective Acknowledgement (SACK), which allows out-of-sequence segments to be accepted by a receiver. The IANA maintains a list of all [TCP Option Numbers](#).

3.3.3. UDP

UDP, described in [RFC 768](#), provides an end-to-end datagram (connectionless) service. Some applications, such as those that involve a simple query and response, are better suited to the datagram service of UDP because there is no time lost to virtual circuit establishment and termination. UDP's primary function is to add a port number to the IP address in order to provide a socket for the application.



The fields of a UDP datagram (Figure 10) are:

- *Source Port*: Identifies the UDP port being used by the sender of the datagram; use of this field is optional in UDP and may be set to 0.
- *Destination Port*: Identifies the port used by the datagram receiver.
- *Length*: Indicates the total length of the UDP datagram.
- *Checksum*: Provides bit error detection for the UDP datagram. The checksum field covers the UDP datagram header and data, as well as a 96-bit *pseudo header* that contains the IP header Source Address, Destination Address, and Protocol fields, as well as the UDP datagram length.

3.3.4. ICMP

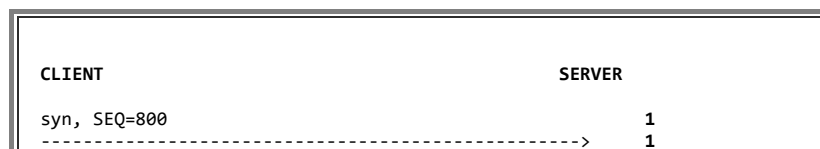
The Internet Control Message Protocol, described in [RFC 792](#), is an adjunct to IP that notifies the sender of IP datagrams about abnormal events. This collateral protocol is particularly important in the connectionless environment of IP. ICMP messages can be sent between two host systems or from a router to a host. Commonly employed ICMP message types include:

- *Destination Unreachable*: Indicates that a packet cannot be delivered because the destination host cannot be reached. The reason for the non-delivery may be that the host or network is unreachable or unknown, the protocol or port is unknown or unusable, fragmentation is required but not allowed (DF-flag is set), or the network or host is unreachable for this type of service.
- *Echo* and *Echo Reply*: These two messages are used to check whether hosts are reachable on the network. One host sends an Echo message to the other, optionally containing some data, and the receiving host responds with an Echo Reply containing the same data. These messages are the basis for the Ping command.
- *Parameter Problem*: Indicates that a router or host encountered a problem with some aspect of the packet's Header.
- *Redirect*: Used by a host or router to let the sending host know that packets should be forwarded to another address. *For security reasons, Redirect messages should usually be blocked at the firewall.*
- *Source Quench*: Sent by a router to indicate that it is experiencing congestion (usually due to limited buffer space) and is discarding datagrams.
- *TTL Exceeded*: Indicates that a datagram has been discarded because the TTL field reached 0 or because the entire packet was not received before the fragmentation timer expired.
- *Timestamp* and *Timestamp Reply*: These messages are similar to the Echo messages, but place a timestamp (with millisecond granularity) in the message, yielding a measure of how long remote systems spend buffering and processing datagrams, and providing a mechanism so that hosts can synchronize their clocks.

ICMP messages are carried in IP packets. The IANA maintains a complete list of [ICMP parameters](#).

3.3.5. TCP Logical Connections and ICMP

It is imperative to understand how a TCP connection is established to get a good feel for how TCP operates. TCP connections have three main parts: connection establishment, data exchange, and connection termination. Figure 11 shows a POP3 server (listening on TCP port 110) being contacted by a client (using TCP port 1967).



| | |
|----------------------------------|-----------------|
| src_port=1967, dst_port=110 | 1 |
| -----> | 1 |
| syn, ack, SEQ=1567, ACK=801 | 1 CONNECTION |
| <----- | 1 |
| src_port=110, dst_port=1967 | 1 ESTABLISHMENT |
| -----> | 1 |
| ack, SEQ=801, ACK=1568 | 1 |
| -----> | 1 |
| src_port=1967, dst_port=110 | 1 |
| -----> | 1 |
| ack, SEQ=1568, ACK=801 | 2 |
| <----- | 2 |
| src_port=110, dst_port=1967 | 2 |
| DataLen=18 (POP3 Server V1.12\n) | 2 |
| -----> | 2 |
| ack, SEQ=801, ACK=1586 | 2 |
| -----> | 2 |
| src_port=1967, dst_port=110 | 2 |
| DataLen=5 (quit\n) | 2 DATA |
| -----> | 2 |
| ack, SEQ=1586, ACK=806 | 2 EXCHANGE |
| <----- | 2 |
| src_port=110, dst_port=1967 | 2 |
| DataLen=9 (Sayonara\n) | 2 |
| -----> | 2 |
| ack, SEQ=806, ACK=1595 | 2 |
| -----> | 2 |
| src_port=1967, dst_port=110 | 2 |
| -----> | 2 |
| fin, ack, SEQ=806, ACK=1595 | 3 |
| -----> | 3 |
| src_port=1967, dst_port=110 | 3 |
| -----> | 3 |
| ack, SEQ=1595, ACK=807 | 3 |
| <----- | 3 |
| src_port=110, dst_port=1967 | 3 CONNECTION |
| -----> | 3 |
| fin, ack, SEQ=1595, ACK=807 | 3 TERMINATION |
| <----- | 3 |
| src_port=110, dst_port=1967 | 3 |
| -----> | 3 |
| ack, SEQ=807, ACK=1596 | 3 |
| -----> | 3 |
| src_port=1967, dst_port=110 | 3 |

FIGURE 11. TCP logical connection phases.

The *connection establishment* phase comprises a three-way handshake during which time the client and server exchange their initial sequence numbers (ISN) and acknowledge the other host's ISN. In this example, the client starts by sending the server a TCP segment with the syn-bit set and a Sequence Number of 800. The syn-bit tells the receiver (i.e., the server) that the sender (i.e., the client) is in "ISN initialization" mode and that the ISN hasn't yet been confirmed. The segment's Acknowledgement Number isn't shown because its value is, at this point, invalid.

The server responds with a segment with the syn- and ack-bits set, a Sequence Number of 1567, and an Acknowledgement Number of 801. The syn-bit and ISN of 1567 have the same meaning as above. The ack-bit indicates the value of the Acknowledgement Number field is valid and the ACK value of 801 is the way in which the server confirms the client's ISN.

The final part of the three-way handshake is when the client sends a segment with just the ack-bit set. Note that the Acknowledgement Number field (1568) is one greater than the server's ISN.

This three-way handshake is sometimes referred to as an exchange of "syn, syn/ack, and ack" segments. It is important for a number of reasons. For individuals looking at packet traces, recognition of the three-way handshake is how to find the start of a connection. For firewalls, proxy servers, intrusion detectors, and other systems, it provides a way of knowing the direction of a TCP connection setup since rules may differ for outbound and inbound connections.

The second part of the TCP connection is *data exchange*. The information here is more or less made up for example purposes only; it shows a POP server sending a banner message to the client system, the user sending the "quit" command, and the server signing off. (Note that the "\n" indicates an "end-of-line" indicator.) These segments show the changing of, and relationship between, the client's and server's sequence and acknowledgement numbers.

The final phase is *connection termination*. Although TCP connections are full-duplex (even if a given application does not allow two-way simultaneous communication), the TCP protocol views the logical connection as a pair of simplex links. Therefore, connection termination requires four segments or, more properly, two pair of segments. In this case, the client sends the server a

segment with the fin- and ack-bits set; the server responds with a segment with just the ack-bit set and the Acknowledgment Number is incremented. The server then sends a fin/ack segment to the client.

The paragraphs above describe a normal scenario setting up a TCP connection between a client and server. Two UDP hosts communicate in a much simpler fashion; one host sends a UDP datagram to the other which is presumably listening on the port indicated in the datagram. If the sender does not receive an expected response in a timely fashion, it doesn't try to figure out why but merely retransmits its message.

But what happens if a host isn't listening on a port to which a connection is attempted or the host doesn't actually exist? Here's what happens in these "abnormal" conditions:

- *Host not listening on TCP port*: If Host A attempts to contact Host B on a TCP port on which Host B is not listening, Host B responds with a TCP segment with the reset (RST) and acknowledge (ACK) flags set.
- *Host not listening on UDP port*: If Host A attempts to contact Host B on a UDP port on which Host B is not listening, Host B sends an ICMP *port unreachable* message to Host A.
- *Host does not exist*: If Host A attempts to contact Host B and Host B is not available (e.g., Host B's IP address either doesn't exist or Host B is down), Host B's subnet's router will send an ICMP *host unreachable* message to Host A.

3.4. The TCP/IP Application Layer

The TCP/IP Application Layer protocols support the applications and utilities that **are** the Internet. This section will list a number of these applications and show a sample packet decode of all protocol layers. (One might argue that it is a misnomer to refer to "TCP/IP applications," as most such applications actually run over TCP or UDP, as shown in [Figure 2](#). Thus, we refer here merely to the "TCP/IP Application Layer.")

3.4.1. TCP and UDP Applications

Commonly used — and a few long-gone — protocols (as shown in [Figure 2](#)) include:

- *Archie*: A utility that allowed a user to search all registered anonymous FTP sites for files on a specified topic. Obsolete by the late-1990s, obviated by the World Wide Web.
- *DNS*: The Domain Name System (described in slightly more detail in Section 3.2.3 above) defines the structure of Internet names and their association with IP addresses, as well as the association of mail and name servers with domains.
- *Finger*: Used to determine the status of other hosts and/or users ([RFC 1288](#)).
- *FTP*: The File Transfer Protocol allows a user to transfer files between local and remote host computers ([RFC 959](#)). Anonymous FTP allows a user to download files from a site without requiring that the user have an account at the site; in this case, the user employs the username *anonymous* and the password *guest* (although, later, users were asked to use their e-mail address as the anonymous password).
- *Gopher*: A tool that allowed users to search through data repositories using a menu-driven, hierarchical interface, with links to other sites. Obsolete today, obviated by the World Wide Web ([RFC 1436](#)).
- *HTTP*: The Hypertext Transfer Protocol is the basis for exchange of information over the World Wide Web (WWW). Various versions of HTTP are in use over the Internet, with HTTP version 1.1 (HTTP/1.1) being the most current ([RFC 7230](#) et seq.). Web site pages are written in the Hypertext Markup Language (HTML), an ASCII-based, platform-independent formatting language ([RFC 2854](#)).
- *Jughead*: A utility that allowed a user to search Gopher sites for files on a specified topic. Obsolete by the late-1990s, obviated by the World Wide Web.
- *IMAP*: The Internet Mail Access Protocol defines an alternative to POP as the interface between a user's mail client software and an e-mail server, used to download mail from the server to the client and providing significant flexibility in mailbox management. IMAP version 4 (IMAP4) is documented in [RFC 3501](#).
- *Ping*: A utility that allows a user at one system to determine the status another host and the latency in getting a message to that host. Uses ICMP Echo messages. For more information and insight, see [The Ping Page](#).
- *POP*: The Post Office Protocol defines a simple interface between a user's mail client software (e.g., Eudora, Outlook, or the e-mail capability of your browser) and an e-mail server, used to download mail from the server to the client and allows the user to manage their mailboxes. POP version 3 (POP3) is documented in [RFC 1939](#).
- *RADIUS*: The Remote Authentication Dial-In User Service (RADIUS) is a remote-access protocol that provides procedures for centralized authentication, authorization, and accounting (AAA) user management features ([RFC 6613](#)).
- *SMTP*: The Simple Mail Transfer Protocol is the standard protocol for the exchange of electronic mail over the Internet ([RFC 5321](#)). SMTP is used between e-mail servers on the Internet or to allow an e-mail client to send mail to a server. [RFC 5322](#) specifically describes the mail message body format, and RFCs [2045](#), [2046](#), [2047](#), [2048](#), and [2049](#) describe MIME (Multipurpose Internet Mail Extensions). Reference books on electronic mail systems include *!%@:: Addressing and Networks* by D. Frey and R. Adams (O'Reilly & Associates, 1993) and *THE INTERNET MESSAGE: Closing the Book With Electronic Mail* by M. Rose (PTR Prentice Hall, 1993).
- *SNMP*: The Simple Network Management Protocol defines procedures and management information databases for managing TCP/IP-based network devices. SNMP ([RFC 1157](#)) is widely deployed in local and wide area networks. [RFC 3411](#) describes the current SNMP architectural framework.
- *SSH*: The Secure Shell is a protocol that allows remote logon to a host across the Internet, much like Telnet. Unlike Telnet, however, SSH encrypts passwords and data traffic. [RFC 4251](#) describes the overall SSH architecture, with additional RFCs describing other operational aspects (e.g., [RFC 4252](#) describes the SSH Authentication Protocol and [RFC 4253](#) describes the

SSH Transport Layer Protocol, responsible for initial key exchange, server authentication, and encryption, compression and integrity verification).

- *SSL*: The Secure Sockets Layer (SSL), originally developed in 1996 by Netscape in order to enable e-commerce, provides a mechanism for secure communications over the Internet, based on certificates and public key cryptography. The most commonly known SSL application is HTTP over SSL, commonly designated *https*. The newest version of SSL is called Transport Layer Security (TLS); TLSv1.2 is described in [RFC 5246](#). SSL is *not*, however, HTTP-specific; protocols such as IMAP4 (imaps), FTP (fips), Telnet (telnets), and POP3 (pop3s) all have definitions for operation over SSL. (Additional details on the operation of SSL/TLS can be found in the [SSL section](#) of the author's cryptography overview paper.)
- *TACACS+*: The Terminal Access Controller Access Control System plus is a remote access protocol developed by Cisco and, like RADIUS, providing authentication, authorization, and accounting (AAA) services. Although TACACS+ has been around for decades, it is still an [Internet draft](#) as of 2017. A description of TACACS can be found in [RFC 1492](#).
- *Telnet*: Short for *Telecommunication Network*, a virtual terminal protocol allowing a user logged on to one TCP/IP host to access other hosts on the network ([RFC 854](#)). Telnet is one of the original ARPANET protocols.
- *TFTP*: The Trivial File Transfer Protocol (TFTP) is used for some specialized simple file transfer applications. Unlike FTP that operates over TCP's virtual connections, TFTP runs over UDP and has a very lightweight communication mechanism. TFTP version 2 is described in [RFC 1350](#).
- *Time/NTP*: Time and the Network Time Protocol (NTP) are used so that Internet hosts can synchronize their system time from well-known Internet time servers. There are many time servers on the Internet. At one time, the two standard time servers were *tick.usno.navy.mil* and *tock.usno.navy.mil*, operated by the U.S. Naval Observatory — a good choice since the U.S. Navy is the [official timekeeper](#) of the U.S., the Dept. of Defense, and the Global Positioning System (GPS). NIST also displays official U.S. time at <https://www.time.gov/>. NTP version 4 is described in [RFC 5905](#). Prior to NTPv4, the Simple Network Time Protocol (SNTP) used a subset of NTP procedures for those systems that did not require the granularity and precision of NTP. The former-SNTP guidelines have been incorporated into NTPv4.
- *Traceroute*: A tool that displays the route taken by packets across the Internet between a local and remote host. The *traceroute* command is available on Linux/Unix systems (including Mac OS X); Windows systems introduced the *tracert* command utility in Windows 95. More information about the operation and background of traceroute can be found at the [InetDaemon traceroute](#) page.
- *Veronica*: A utility that allowed a user to search all Gopher sites for files on a specified topic. Obsolete by the late-1990s, obviated by the World Wide Web.
- *Whois*: A utility that searches databases for information about Internet domains and domain contact information ([RFC 3912](#)). *NICNAME* is a former alias for Whois.

A guide to using many of these applications can be found in "A Primer on Internet and TCP/IP Tools and Utilities" (FYI 30/[RFC 2151](#)) by Gary Kessler & Steve Shepard (also available in [HTML](#) or [PDF](#)).

3.4.2. Protocol Analysis

Full-blown protocol analysis is well beyond the scope of this paper. But a little introduction is ok!!

In the deep past, network managers would use serial line analyzer hardware to display the 0s and 1s on the line; most had a mode that displayed hexadecimal so that we had a fighting chance to know the octet boundaries. Protocol analyzers had software to interpret the bit stream and display the interpreted protocol values. In the late-1990s, we started to see specialized packet sniffing software that would perform protocol analysis. When using such software, a computer's network interface card (NIC) operates in *promiscuous mode* so that the NIC captures every packet that flies by on the wire rather than only those packets addressed to this particular NIC. The most popular packet sniffing/protocol analysis software today is [Wireshark](#) (formerly known as Ethereal), which runs on Linux, Mac OS X, Unix, and Windows platforms.

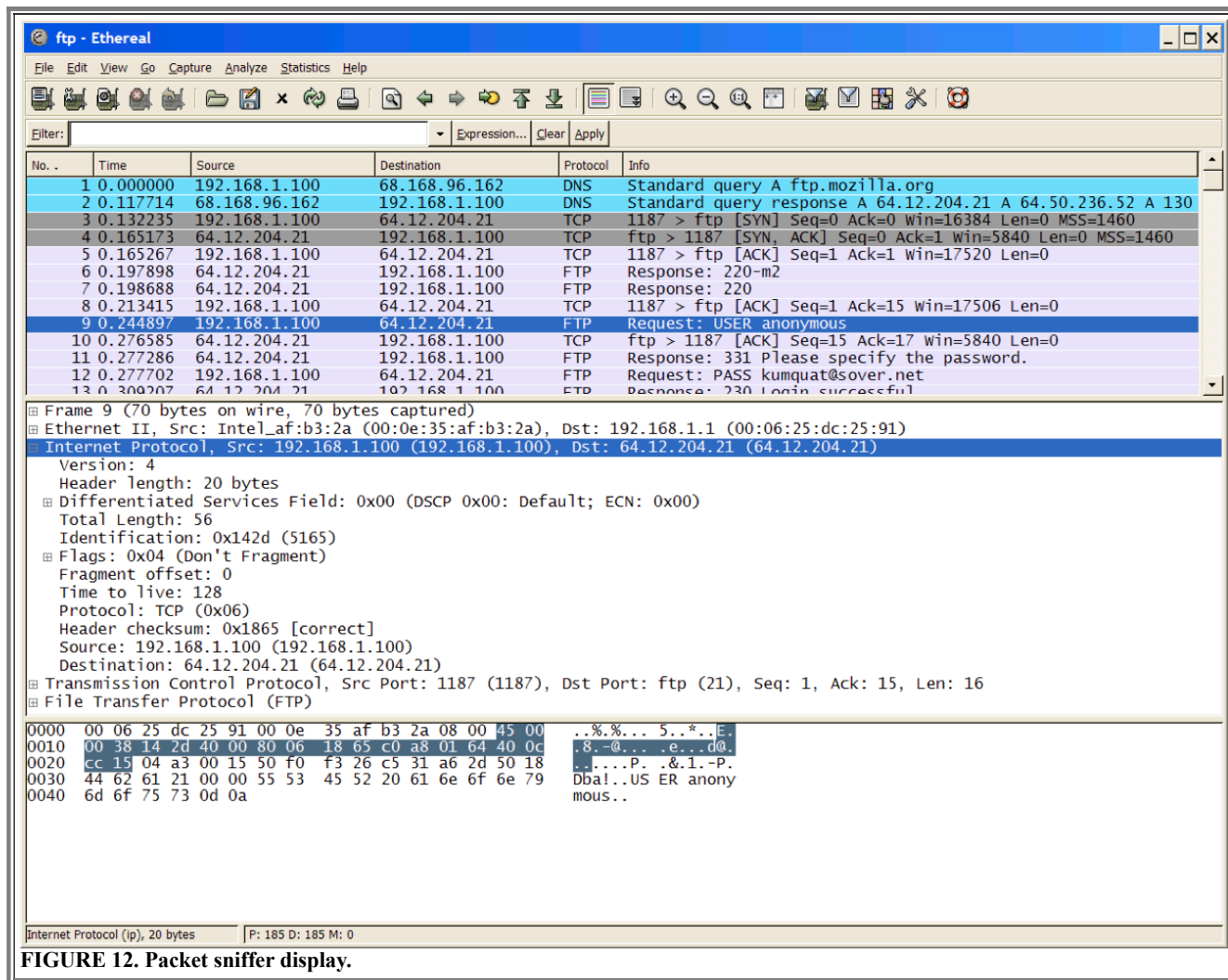


FIGURE 12. Packet sniffer display.

Figure 12 shows an Ethereal screenshot. To understand the display of the protocol analyzer, recall that TCP/IP is a layered protocol architecture, as shown in Figure 2. The lowest level unit of transmission at the Network Interface Layer protocol is called a *frame*. Within the frame is an IP *packet*. Packets contain Transport Layer data unit, such as a TCP *segment*, UDP, or ICMP *message*. Data units at the Application Layer are also referred to as *messages*.

Note that the packet sniffer display has three sections. The top part shows a summary of the frames in the capture buffer. Note here that we see frames numbered 1-13 (*No.* column). The *Time* column displays the relative time of the packet transmission, where the first packet is at time 0. The next two columns list the source (sender's) and destination (receiver's) IP addresses. The *Protocol* column indicates the protocol associated with this packet, while the *Info* column provides some summary information about the packet. In this case we see:

- In Frame 1, the local host (192.168.1.100) sends a DNS request to a local name server (68.168.96.162), asking for the IP address of a host named *ftp.mozilla.org*. In Frame 2, the server responds with several addresses, the first of which is 64.12.204.21.
- Frame 3-5 show the three-way handshake to set up a TCP virtual connection between the local host and the FTP server; note the settings of the SYN, SYN/ACK, and ACK flags in the three frames.
- FTP traffic exchange occurs in Frames 6-13; note where the user sends a username (Anonymous) in Frame 9, the host requests a password in Frame 11, the user sends a password (kumquat@sover.net) in Frame 12, and the server indicates successful login in Frame 13.

The middle section of the display shows the decode and interpretation of a single transmission, in this case, Frame 9:

- The very first line tells us that the frame is 70 bytes in lengths.
- The second line indicates that this is an Ethernet II frame, meaning that Ethernet is the Network Interface Layer. We also see the source (i.e., local host) and destination (i.e., local router) hardware (i.e., medium access control, or MAC) addresses.
- The next block of output is a detailed examination of the IP packet. Here we see the source and destination host IP addresses (i.e., local host and FTP server, respectively), indication of IPv4, and other protocol information. Note that the IP packet's Protocol field indicates use of TCP.

- The next-to-last line in the display is the TCP block, where TCP-specific information would be displayed. Note that the destination port is 21, which indicates use of FTP.
- The last line is the block where FTP-specific information would be displayed.

The bottom section shows the selected transmission in hexadecimal. This is the raw bit stream. Note that some of the bytes are highlighted here; they match the block highlighted in the middle part of the display, namely the entire IP header portion of the transmission. Note that this frame contains the Anonymous user name (as shown in the upper section of the display); this can also be seen in the raw byte stream.

The discussion here is only meant to give readers a taste of one of the coolest tools that we get to play with in data communications; it is also an important tool for network managers and security administrators.

3.5. Summary

As this discussion has shown, *TCP/IP* is not merely a pair of communication protocols but is a suite of protocols, applications, and utilities. Increasingly, these protocols are referred to as the *Internet Protocol Suite*, but the older name will not disappear anytime soon.

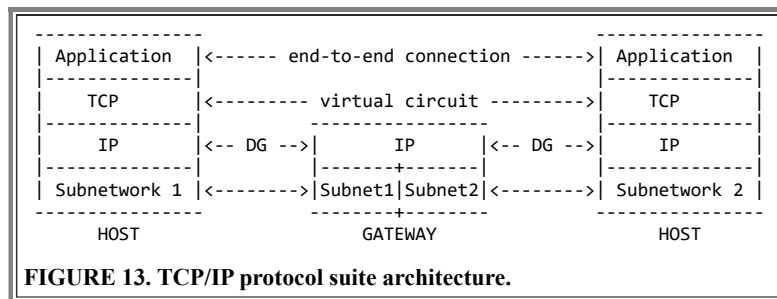


Figure 13 shows the relationship between the various protocol layers of TCP/IP. Applications and utilities reside in host, or end-communicating, systems. TCP provides a reliable, virtual circuit connection between the two hosts. (UDP, not shown, provides an end-to-end datagram connection at this layer.) IP provides a datagram (DG) transport service over any intervening subnetworks, including local and wide area networks. The underlying subnetwork may employ nearly any common local or wide area network technology.

Note that the term *gateway* is used for the device interconnecting the two subnets, a device usually called a *router* in LAN environments or *intermediate system* in OSI environments. In OSI terminology, a *gateway* is used to provide protocol conversion between two networks and/or applications.

4. Other Information Sources

This memo has only provided background information about the TCP/IP protocols and the Internet. There is a wide range of additional information that the reader can access to further use and understand the tools and scope of the Internet. The real fun begins now!

Internet specifications, standards, reports, humor, and tutorials are distributed as Request for Comments (RFC) documents. RFCs are all freely available on-line, and most are available in ASCII text format from the [RFC Editor](#) Web site; you can also sign up to receive notification of new RFCs at the [RFC Announcements](#) page.

Internet standards are documented in a subset of the RFCs, identified with an "STD" designation; [RFC 2026](#) describes the Internet standards process and the RFC Editor's [Official Internet Protocol Standards](#) page contains the official list of Internet standards.

For Your Information (FYI) documents are another RFC subset, specifically providing background information for the Internet community. The FYI notes are described in [RFC 1150](#) although the FYI sub-series has been discontinued. Similarly, Internet Best Current Practices (BCP) are another subset of RFCs, described in [RFC 1818](#).

Frequently Asked Question (FAQ) lists may be found for a number of topics, ranging from ISDN and cryptography to the Internet and Gopher. Two such FAQs are of particular interest to Internet users: "FYI on Questions and Answers - Answers to Commonly asked 'New Internet User' Questions" ([RFC 2664](#)) and "FYI on Questions and Answers: Answers to Commonly Asked 'Experienced Internet User' Questions" ([RFC 1207](#)). All of these documents point to even more information sources.

5. Acronyms and Abbreviations

| | |
|---------|--|
| ARP | Address Resolution Protocol |
| ARIN | American Registry for Internet Numbers |
| ARPANET | Advanced Research Projects Agency Network |
| ASCII | American Standard Code for Information Interchange |

| | |
|----------|--|
| ATM | Asynchronous Transfer Mode |
| BGP | Border Gateway Protocol |
| BSD | Berkeley Software Development |
| CCITT | International Telegraph and Telephone Consultative Committee |
| CIX | Commercial Internet Exchange |
| CDPD | Cellular Digital Packet Data protocol |
| CSLIP | Compressed Serial Line Internet Protocol |
| DARPA | Defense Advanced Research Projects Agency |
| DDP | Datagram Delivery Protocol |
| DDS | Digital data service |
| DNS | Domain Name System |
| DOCSIS | Data Over Cable System Interface Specification |
| DoD | U.S. Department of Defense |
| DWDM | Dense Wave Division Multiplexing |
| FAQ | Frequently Asked Questions lists |
| FDDI | Fiber Distributed Data Interface |
| FTP | File Transfer Protocol |
| FYI | For Your Information series of RFCs |
| GOSIP | U.S. Government Open Systems Interconnection Profile |
| HDLC | High-level Data Link Control |
| HTML | Hypertext Markup Language |
| HTTP | Hypertext Transfer Protocol |
| IAB | Internet Activities Board |
| IANA | Internet Assigned Numbers Authority |
| ICANN | Internet Corporation for Assigned Names and Numbers |
| ICMP | Internet Control Message Protocol |
| IESG | Internet Engineering Steering Group |
| IETF | Internet Engineering Task Force |
| IMAP | Internet Message Access Protocol |
| InterNIC | Internet Network Information Center |
| IP | Internet Protocol |
| IPX | Internetwork Packet Exchange |
| ISDN | Integrated Services Digital Network |
| ISO | International Organization for Standardization |
| ISOC | Internet Society |
| ITU-T | International Telecommunication Union Telecommunication Standardization Sector |
| MAC | Medium (or media) access control |
| Mbps | Megabits (millions of bits) per second |
| NICNAME | Network Information Center name service |
| NSF | National Science Foundation |
| NSFNET | National Science Foundation Network |
| NTP | Network Time Protocol |
| OSI | Open Systems Interconnection |
| OSPF | Open Shortest Path First |
| PING | Packet Internet Groper |
| POP3 | Post Office Protocol v3 |
| PPP | Point-to-Point Protocol |
| RADIUS | Remote Authentication Dial-In User Service |
| RARP | Reverse Address Resolution Protocol |
| RIP | Routing Information Protocol |
| RFC | Request For Comments |
| SDH | Synchronous Digital Hierarchy |
| SLIP | Serial Line Internet Protocol |
| SMDS | Switched Multimegabit Data Service |
| SMTP | Simple Mail Transfer Protocol |
| SNAP | Subnetwork Access Protocol |

| | |
|---------|---|
| SNMP | Simple Network Management Protocol |
| SONET | Synchronous Optical Network |
| SSL | Secure Sockets Layer |
| STD | Internet Standards series of RFCs |
| TACACS+ | Terminal Access Controller Access Control System plus |
| TCP | Transmission Control Protocol |
| TFTP | Trivial File Transfer Protocol |
| TLD | Top-level domain |
| UDP | User Datagram Protocol |
| WAP | Wireless Application Protocol |
| xDSL | Digital Subscriber Line family of technologies |

6. About the Author

[Gary Kessler](#), Ph.D., CCE, CISSP is the president and janitor of [Gary Kessler Associates](#), an independent consulting and training firm specializing in computer and network forensics, information security, Internet access issues, and TCP/IP networking. He is also a Professor of Cybersecurity at Embry-Riddle Aeronautical University in Daytona Beach, Florida. Gary has written more than 70 papers for industry publications; his main areas of interest include TCP/IP and the Internet, computer and network forensics, and network security. His e-mail address is gck@garykessler.net.