Geographic Implications of DNS Infrastructure Distribution

by Steve Gibbard, Packet Clearing House

The past several years have seen significant efforts to keep local Internet communications local in places far from the well-connected core of the Internet. Although considerable work remains to be done, Internet traffic now stays local in many places where it once would have traveled to other continents, lowering costs while improving performance and reliability. Data sent directly between users in those areas no longer leaves the region. Applications and services have become more localized as well, not only lowering costs but keeping those services available at times when the region's connectivity to the outside world has been disrupted. I discussed the need for localization in a previous paper, "Internet Mini-Cores: Local connectivity in the Internet's spur regions."[1] What follows here is a more specific look at a particular application, the *Domain Name System* (DNS).

Most Internet applications depend on the DNS, which maps human-readable domain names to the *Internet Protocol* (IP) addresses computers understand. Two Internet hosts may have connectivity to each other but be unable to communicate because no DNS server can be reached. This article examines the placement of DNS servers for root and top-level domains and the implications of that placement on the reliability of the services these servers provide in different parts of the world. It is not a "how-to" guide to the construction of DNS infrastructure and does not contain recommendations on DNS policy; it is rather a look at the placement of DNS infrastructure as currently constructed.

Although it is possible to access Internet resources without the DNS by entering numeric IP addresses directly, this type of access is not generally done. IP addresses, such as **`209.131.36.158`**, are difficult to remember, are generally unpublished outside the DNS, and often change without notice. Local caching of DNS information can mask temporary problems with DNS data for commonly accessed domain names, but caches are emptied when caching resolvers are restarted, data in caches expires, and nothing is cached until the first time it is accessed by a local user.

It should be noted that information about DNS deployment is changing rapidly. Several organizations are working on new DNS deployment. Information in this article can be considered current, to the best of my knowledge, as of May 2006.

DNS Hierarchy

The DNS is a hierarchy of domains within domains. The levels of the hierarchy are separated by dots. At the top of the hierarchy is the *root,* usually invisible but sometimes represented as a trailing dot. Using **www.yahoo.com** as an example, the **com** domain is contained within the root. **Com** contains **yahoo**, and **yahoo** contains **www**. Domains in the position **com** takes in this example are known as *Top-Level Domains,* or TLDs; they are the first level in the root domain. Domains in the position of **yahoo** are known as *Second-Level Domains*. In this example, **www** occupies the third level, and so forth.

The information that makes up the Domain Name System is stored on DNS *servers*. That information is divided into *zones,* which for our purposes are synonymous with domains. Each zone is stored on a set of *authoritative servers,* which are queried when users or applications attempt to access a service on the Internet. In the simplest case, a domain name query works like the following:

A *caching resolver* (so named because it caches information it receives) that has not yet cached any DNS zone data receives a query for **www.yahoo.com**. Because its cache is empty, it uses the *hints* distributed with the DNS software to contact one of the root servers and asks, "Where is **www.yahoo.com**?" The root server replies with a list of servers for **.com**. The caching server then asks one of the **.com** servers, "Where is **www.yahoo.com**?" and gets a response that directs it to servers for **yahoo.com**. It asks the same question of those servers and finally gets an answer to the question it was asking.

Generally several servers can answer questions about any domain, but if all the servers for any single level are broken or unreachable, the query fails and the service the user is looking for is unreachable. It is therefore important that the DNS be reliable, and that the servers for each zone throughout the hierarchy be reachable from anywhere the servers they point at are being used.

Root Servers

Without root servers, none of the DNS works. As of this writing, 117 root servers exist worldwide, operated by 12 different organizations.[2] Root servers are added frequently, so the number may be significantly greater by the time this article is in circulation.

Because of protocol limitations, the root servers can use only 13 IP addresses. Each root-server operator is responsible for one or two of those addresses. Using a technique called *anycast,* which allows servers in separate locations to share a *single* IP address, six of those operators operate multiple servers using the same IP address[3], meaning that only 13 of them are visible at the same time from any single location, but those 13 should in most cases include the topologically closest one.[4]

The distribution of root servers is rather uneven. North America and Europe have similar numbers: 38 in North America and 35 in Europe. The 35 in Europe are distributed fairly evenly, with the largest concentrations (four each) in London and Amsterdam, Europe's two largest Internet hubs. North America has 8 in Washington, D.C., 8 in the San Francisco Bay Area, and 5 in Los Angeles. In the United States, all cities that host root servers are on the coasts except Atlanta and Chicago. All seven of the remaining IP addresses represented by only a single server, known as *unicast roots,* are in the Washington, D.C., San Francisco, and Los Angeles areas.

Australia has two root servers in Brisbane, one in Perth, and one in Sydney. New Zealand has two, one in Wellington and one in Auckland. Singapore and the wealthier parts of East Asia are well-covered, and there are two root servers in Jakarta and one each in Bangkok and Kuala Lumpur. A year ago, there were none in the vast expanse between Bangkok and Dubai, but three have recently been added in India, along with others in Dhaka and Karachi. Mainland China and the former USSR each have two. There are three in Africa: two in Johannesburg and one in Nairobi. Another will be installed in Nairobi shortly, but most of the rest of Africa lacks direct connectivity to Johannesburg or Nairobi and must cross satellite or intercontinental fiber links to reach the nearest root servers. All four of the root servers in South America are in Brazil and Chile, with two in Sao Paulo and one each in Brasilia and Santiago de Chile.

With some exceptions, root-server density tends to correlate strongly with per-capita income. This fact is not surprising—it is true for other forms of infrastructure as well—but it means that those with the greatest dependence on external infrastructure are those least able to pay for external connectivity.

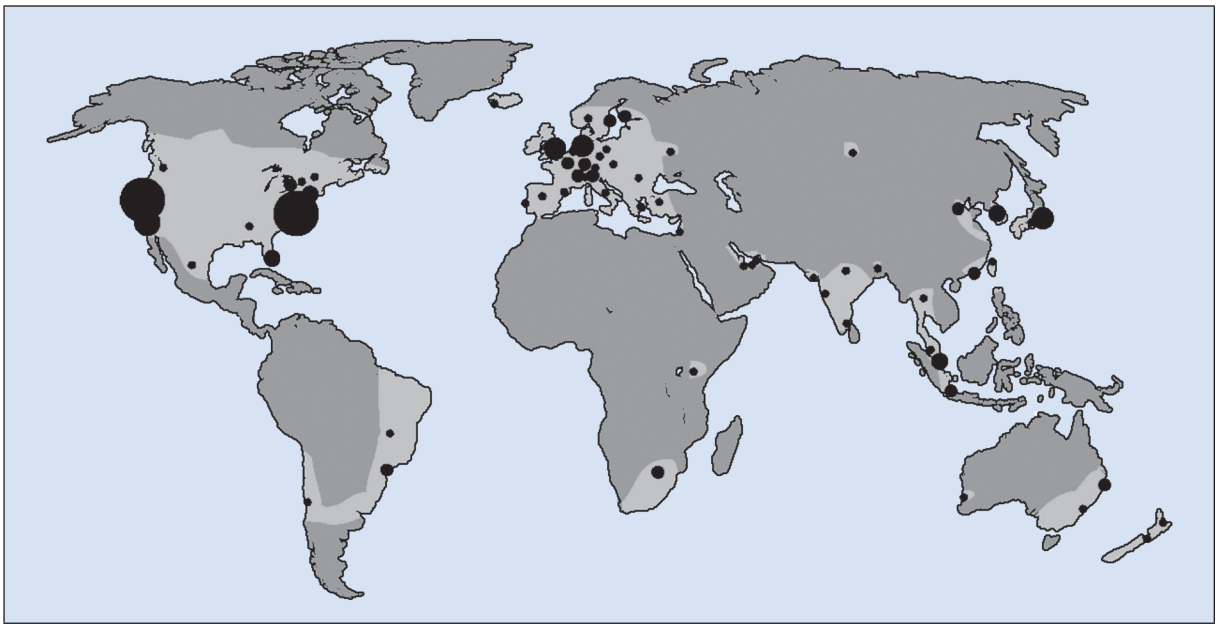
Root-Server Placement

In areas that have local root servers, finding the name servers for a top-level domain should be fairly reliable. In areas without local root servers, the ability to query the root servers is dependent on other long-distance infrastructure. In some places this infrastructure is well-developed, so this problem is not a significant one. Elsewhere long-distance infrastructure is slow, expensive, and unreliable, consisting of satellite links or a single fiber connection that may take several days to fix if it breaks.

Sri Lanka, for example, is connected to the rest of the world by a single fiber connection, which was cut in 2004 by a ship that dragged anchor in the Colombo harbor.[5] Although Sri Lanka has an exchange point that should have allowed connectivity to local Internet services, news reports said that Sri Lanka's "Internet and long-distance phone service" had been cut off. I have not received a good account of what Internet connectivity looked like from anyone in Sri Lanka at the time, but it is likely that even local Internet connections would not have worked without a local root server.

Sri Lanka is not an isolated case. The dots in Figure 1 show the locations of all root servers. The light grey areas are regions in which multiple fiber paths are available to root servers. The remainder of the world can reach root servers only by a single fiber path or by satellite. Large areas of the world are poorly covered.

Figure 1: Root Server Locations and Areas of Redundant Connectivity



Root-Server Expansion

Four of the 12 root-server operators are presently working to install root servers in areas that lack them. Although the 117 root servers currently in operation are a big improvement over the 13 that were in operation three years ago, many regions still do not have any. Those root-server operators are installing servers wherever they can get the funding to do so.

Funding is generally provided either by grants, especially from the *Asia Pacific Network Information Centre* (APNIC) in the Asia-Pacific region, by local governments or *Internet Service Provider* (ISP) associations. Because the addition of new anycast copies of root servers is relatively easy given sufficient funding, the main limitation preventing the installation of root servers in new locations is lack of funding.

One question probably best addressed in a more central manner is whether it makes sense to have many copies of one or two root-server IP addresses in some regions or whether it would be better to have more of a mix of root-server IP addresses. Currently, only 6 of the 13 root-server addresses are anycasted, only 4 are anycasted in large numbers, and 2 of those focus on specific regions, meaning that in many of the more remote parts of the world the only nearby root servers are *Internet Systems Consortium* (ISC)'s "F" and Autonomica's "I" roots, and some places have several of one of those closer than the next one of the other.

Because some DNS resolvers have their own mechanisms for finding the closest server and for handling failures of types that do not include route withdrawals, having multiple IP addresses nearby seems like a good thing. A more complex question is whether it would be worthwhile to anycast all 13 of them widely, or if there is some smaller number that would be sufficient to have nearby. Previous research on this topic has assumed a limit of 13 root servers, producing conclusions that are not applicable to the modern Internet.[6]

This article should not be seen as a criticism of the places with large numbers of root servers. Although the U.S. distribution looks strange, with the San Francisco Bay and D.C. area clusters perhaps excessive, it comes close to following the Internet topology in the United States. Indeed, the U.S. concentration may be appropriate to handle server load. Western Europe's dense but relatively even distribution of root servers through the region appears to be an optimal distribution, because most populated areas have multiple root servers nearby. Likewise, Jakarta is one of the very few cities in the developing world to have more than one, and that provides local redundancy that much of the developing world lacks. If root-server deployment were funded from a single global budget, the distribution across the world's regions would look very unfair. But because Internet infrastructure is mostly funded locally, Jakarta and Western Europe are examples other regions could emulate.

TLD Distribution

Use of the DNS also requires access to TLD servers. To access something in the **.com** domain, a user's local DNS resolver must be able to reach the **.com** servers. This statement is true for any TLD, whether it is a *generic TLD* (gTLD), such as **.com**, **.net**, and **.org**, or a *country code TLD* (ccTLD). Unlike the root, it is not necessary that all TLDs be reliable from all locations; if a TLD is not used to name local resources in a region, having local access to that TLD will not help if that the region gets cut off from the rest of the world.

gTLD Distribution

Of the gTLDs, **.com** is by far the largest. It is well-connected to the Internet core, the area with well-meshed internal connectivity mainly comprising North America, Western Europe, East Asia, and Singapore. (See Figure 2.) The **.com** servers are located in Australia, Brazil, Japan, South Korea, the Netherlands, Sweden, the United Kingdom, and the U.S. states of California, Florida, Georgia, Virginia, and Washington. The **.com** servers are well-connected to areas well-connected to those regions but poorly connected to Africa, South Asia, and parts of South America.

Figure 2: Server Locations for .com and .net and Areas of Redundant Connectivity

UltraDNS, the operator of **.org**, **.info**, **.mobi**, and **.coop**, among others, is also somewhat well-connected to the Internet core, although not to the extent the **.com** servers are. It has publicly accessible servers in four metropolitan areas in the United States as well as in London and Hong Kong. It has a couple of noncore locations, in Delhi and Johannesburg. UltraDNS also has servers in other locations, accessible only to the resolvers of certain large ISPs. Because those servers are not available to the general public in their regions, they are omitted from discussion here. (See Figure 3.)

Figure 3: Server Locations for .org, .info, and .mobi and Areas of Redundant Connectivity

Other gTLDs do not do considerably better. Table 1 shows the locations of all the gTLDs.

Table 1: Locations of TLD Servers

gTLD	Locations by Country or U.S. State
.aero	Switzerland, Germany, India, Hong Kong, United Kingdom, and the following states in the United States: California, Illinois, and Virginia
.biz	Australia, Hong Kong, Netherlands, New Zealand, Singapore, United Kingdom, and the following states in the United States: California, Florida, Georgia, New York, Virginia, and Washington
.com	Australia, Brazil, Canada, Japan, South Korea, Netherlands, Sweden, Singapore, United Kingdom, and the following states in the United States: California, Florida, Georgia, Virginia, and Washington
.coop	United Kingdom and the following states in the United States: California, Illinois, and Massachusetts
.edu	Netherlands, Singapore, and the following states in the United States: California, Florida, Georgia, and Virginia
.gov	Canada, Germany, and the following states in the United States: California, Florida, New Jersey, Pennsylvania, and Texas
.info	India, Hong Kong, South Africa, United Kingdom, and the following states in the United States: California, Illinois, and Virginia
.int	Netherlands, United Kingdom, and California in the United States
.jobs	Netherlands, Singapore, and the following states in the United States: California, Florida, Georgia, and Virginia
.mil	The following states in the United States: California, Maryland, Virginia, and other unknown locations
.mobi	India, Hong Kong, South Africa, United Kingdom, and the following states in the United States: California, Illinois, and Virginia
.museum	Sweden and California in the United States
.name	Singapore, United Kingdom, and the following states in the United States: California, Florida, Georgia, Virginia, and Washington
.net	Australia, Brazil, Canada, Japan, South Korea, Netherlands, Sweden, Singapore, United Kingdom, and the following states in the United States: California, Florida, Georgia, Virginia, and Washington
.org	India, Hong Kong, South Africa, United Kingdom, and the following states in the United States: California, Illinois, and Virginia
.pro	Canada and the following states in the United States: Illinois and Texas
.travel	Australia, Hong Kong, Netherlands, New Zealand, Singapore, United Kingdom, and the following states in the United States: California, Florida, Georgia, New York, Virginia, and Washington

Although gTLDs are typically marketed for their applicability to specific types of organization, or in the case of **.com** because it is the only domain many people have heard of, geography should also be considered in selecting domains. Most of the gTLDs have reasonable coverage throughout the Internet core region, but there are exceptions. The **.int** and **.museum** domains are hosted only in North America and Europe, and **.pro** is hosted only in North America.

Outside the Internet core there is little gTLD presence. Only **.biz**, **.travel**, **.com**, and **.net** are present in Australia and New Zealand. South Africa and India have **.aero**, **.info**, **.mobi**, and **.org**, making them the only gTLDs hosted in either Africa or the South Asian region. South America hosts only **.com** and **.net**, with servers in two cities in Brazil. Taken together, these are the only Southern Hemisphere gTLD locations as of this writing, and no gTLD has any presence in parts of the world without external fiber-optic connectivity, although that may be changing.

Where gTLDs should be hosted, and with what scope, are somewhat open questions. Should these domains address resources anywhere, or should their scope be local? This question is really one for the *Internet Corporation for Assigned Names and Numbers* (ICANN), or for the gTLD sponsors or registries, and beyond the scope of this article. Verisign, the company that administers **.com** and **.net**, points out that database replication with the amount of changes in the **.com** zone is a significant problem over slow network links.

ccTLD Distribution

Questions about where ccTLDs, the top-level domains assigned to individual countries, ought to work seem more straightforward. Working effectively in their own countries seems like the top priority, with connectivity to the Internet core and to other regions with which people in the country communicate regularly being somewhat lower priorities. Just over two-thirds of ccTLDs are hosted in their own countries; refer to Figure 4 for the bigger countries, and the online appendices for the full list. Although the third of ccTLDs not hosted in their own countries include some marketed more for international use than global use—Cocos Island's **.cc**, Tonga's **.to**, Turkmenistan's **.tm**, and Tuvalu's **.tv**, among others—those are very much the exception.

Indonesia has local access to the root and to its ccTLD (**.id**). Pakistan has a root server, but no local access to its ccTLD (**.pk**). Let's compare what happens when someone in Indonesia does a lookup on an **.id** domain name with what happens when someone in Pakistan does a lookup of a name in the **.pk** domain.

In Indonesia, the query goes to a root DNS server at the Indonesian Internet Exchange in Jakarta, where it is answered with the locations of the **.id** servers, several of which are also in Indonesia. The query then goes to the local **.id** server and is answered locally, whereupon the user can start sending traffic to the host he or she was trying to connect to, which is presumably also local. The traffic need not leave Indonesia, and if all the parties involved are in Jakarta it need not leave town.

The Pakistani case is quite different. Until early 2006, there were no root servers in Pakistan, nor were there local servers for the **.pk** domain. There is now a root server in Karachi, but lacking servers for any TLDs it is of limited utility. DNS resolvers start out querying the local root server, but the response directs them to servers for the **.pk** domain, all located in the United States, at least 10 time zones away. They then send their lookup packets across the single fiber connection all the way to the United States and wait for the response. At best, this process is slow. If that fiber connection goes down, or if there is any other problem between Pakistan and these U.S. servers, local communications in Pakistan are crippled.

The situation with traditional gTLDs (**.com**, **.net**, and **.org**) in Indonesia and Pakistan is somewhat different. In Indonesia, local root servers provide addresses for the **.com**, **.net**, and **.org** servers. The **.com** and **.net** lookups can be handled in Singapore, 18 milliseconds away. Theoretically, **.org** lookups can be handled in Hong Kong, but *traceroutes* indicate **.org** queries being answered in California instead. Thus, in Indonesia, **.id** is hosted mostly locally, **.com** and **.net** are nearby, and **.org** is considerably farther away. In Pakistan, in contrast, **.pk** queries and **.org** queries are answered from the United States, more than 200 milliseconds away, while **.com** and **.net** are answered from Singapore, 80 milliseconds away. For Pakistani users of all TLDs, there are single points of failure, but **.com** and **.net** do appear to be somewhat better connected than **.pk**.

In Nairobi, Kenya, there are local copies of a root server and the local ccTLD (**.ke**). All external connectivity is by satellite, and most ISPs have only a single satellite link. Two Internet users in Nairobi wanting to communicate can do a lookup on the local root server to find the servers for **.ke** and can do a lookup on a local **.ke** server to find the servers for a subdomain of **.ke**. Assuming the subdomain being used is served locally, they can do a local lookup for a host within that subdomain and then send data across the local exchange point. Thus the two users in the same town can send data back and forth without having to send any data elsewhere.

According to Verisign, Nairobi will soon have servers for **.com** and **.net** as well. In contrast, to use the **.org** domain they can again obtain addresses of the **.org** server from their local root server, but the lookup of the **.org** domain must go over a satellite link to Europe in order to be answered by a server in London. If the satellite link is up, this process adds half a second of latency to the query. If the satellite link is down, whatever local resource they are trying to connect to is out of reach.

Figure 4: Countries that Host Their Own ccTLDs in Grey; Those that Do Not in Black

Note: Western Sahara and North Korea do not have active ccTLDs, hence the white areas.

There is also a concern about ccTLDs not served from the global core; if their region or upstream provider is cut off from the Internet outside their region, the rest of the world is unable to see that ccTLD. (See Table 2). This situation may or may not be of concern; if all Internet resources within that ccTLD become unreachable in the same outage, the DNS portion of the outage may have no additional effect. However, if there is anything in that ccTLD that is not in the ccTLD's region, or if people or systems outside prefer to get a DNS response for an unreachable IP address rather than no DNS response at all, it may be of concern. Indeed, having servers that are well-connected to "the Internet as a whole" is a recommendation of RFC 2182, though the RFC does not consider the case of large portions of the Internet not being well-connected to each other.

Table 2: TLDs Not Served in the "Internet Core" Region

TLD	Country	Location of DNS Servers
BB	Barbados	Barbados
BD	Bangladesh	Bangladesh
BH	Bahrain	Bahrain
CN	China	China
EC	Ecuador	Ecuador
GF	French Guiana	French Guiana and Guadeloupe
JM	Jamaica	Jamaica
KG	Kyrgyzstan	Kazakhstan
KW	Kuwait	Kuwait
MP	Northern Mariana Islands	Guam
MQ	Martinique	Guadeloupe and Martinique
PA	Panama	Brazil, Chile, Costa Rica, and Panama
PF	French Polynesia	French Polynesia
QA	Qatar	Qatar
SR	Suriname	Suriname
TJ	Tajikistan	Tajikistan
ZM	Zambia	South Africa and Zambia

Lack of Exchange Points and Local Peering

In the "Internet Mini-Cores" article[1], I noted that local hosting of critical infrastructure is moot if there is not either a local exchange point or a monopoly transit provider in the region. If data needed in a poorly connected region must leave the area and return to reach the user requesting it, the communication has double the latency, and possibly double the reliability problems, that it would have if it were hosted somewhere in the core. For the specific examples used in this article, I have mostly chosen areas that do have exchange points. I have not analyzed the underlying local infrastructure in all countries.

Methodology

The addresses of DNS servers for a TLD are available through several means: by looking at the root zone, by doing *digs* for the name servers, and by looking in the *Internet Assigned Numbers Authority* (IANA) *whois* data, among others. I did lookups against an anycast root server on my own network, because that seemed easiest to automate. My script then did a lookup for the address of each name server, stripped off the last octet, and produced a list of TLDs hosted in each /24 subnet.

There are 635 /24s containing name servers for TLDs; 142 of them host multiple TLDs; the rest host just one. I assumed that all DNS servers in a given /24 were likely to be in the same or nearby locations. This situation appears not to be the case for the UUNet name servers, and there are probably a few other exceptions that will show up as errors in my data.

I looked at a few automated geolocation systems to attempt to attach locations to the DNS servers, but none of them appeared to be producing accurate information. Instead, I guessed at the locations of the 600 subnets, using *traceroutes* from a variety of locations, paying attention to DNS, latency, and the results of whois queries for address space along the way. I also asked lots of questions of DNS operators and others and am particularly grateful to several anycast DNS operators, whose locations would not have all been found by my *traceroutes*. Some of my guesses are likely incorrect, and corrections are appreciated.

I may be missing some information about the UltraDNS TLD servers, because UltraDNS has locations it regards as confidential. This information about UltraDNS servers is from Afilias's **.net** application, *traceroutes* from a variety of locations, and UltraDNS.[7]

Locations of root servers are easier to find; they are listed at **http://www.root-servers.org**. Some supplemental information about **j.root-servers.net** was supplied by Verisign. If there are operational root servers not included on **www.root-servers.org** other than the J-roots, I did not count them.

The full lists of locations of all TLDs and TLD servers are in the appendices to this article, at:
http://www.pch.net/resources/papers/infrastructure-distribution/dns-distribution-appendices.pdf.

References

[1] Steve Gibbard, "Internet Mini-Cores: Local connectivity in the Internet's spur regions" (2005):
http://www.pch.net/resources/papers/Gibbard-mini-cores.pdf

[2] Root Server Technical Operations Association:
http://www.root-servers.org

[3] Joe Abley, "Hierarchical anycast for global service distribution," ISC Tech Notes (2003):
http://www.isc.org/index.pl?/pubs/tn/index.pl?tn=isc-tn-2003-1.html

[4] Bradley Huffaker, "Two days in the life of three DNS root servers" (2006):
http://www.caida.org/publications/presentations/2006/brad_wide0611_anycast_analysis

[5] Tim Richardson, "Ship's anchor cuts cable to Sri Lanka," *The Register,* August 24, 2004:
http://www.theregister.co.uk/2004/08/24/sri_lanka_anchor

[6] Tony Lee, Bradley Huffaker, Marina Fomenkov, and kc claffy, "On the problem of optimization of DNS root servers' placement" (2003):
http://www.caida.org/publications/papers/2003/dns-placement/

[7] Afilias, ".NET Application Form":
http://www.icann.org/tlds/net-rfp/applications/afilias.htm

STEVE GIBBARD is a Network Architect for the nonprofit organization, Packet Clearing House (**www.pch.net**), based in Berkeley, California. He runs an anycast DNS network that hosts the top-level domains for several countries and several of the "I" root anycast DNS servers, maintains PCH's network of route collectors and route servers at exchange points around the world, and researches the interconnection of Internet networks. In addition, Steve carries out network architecture and peering work as a consultant for several ISPs in the San Francisco Bay Area and elsewhere. Steve is a former Senior Network Engineer at Cable & Wireless, and has held network engineering positions at Digital Island and World Wide Net. E-mail: **scg@pch.net**