Enabling Internet Technologies in Guinea

A White Paper for Development

by Wayne Marshall

INTRODUCTION

Consider briefly the cycle of development in a rural African village supplying its people with electricity. Long before anyone is ever connected to a national electrical grid, the village will go through several stages of technology and adaptation. At first some people may simply connect radios and a few light bulbs to a car battery. The barrier to entry is not insurmountable, and access encourages appreciation of the benefits of electricity. And even with crude circuitry and basic wiring, the people begin to gain a growing and empowering knowledge of the technology.

One day a small generator may appear in the village. At first the generator may run for short periods, and refrigeration and pumping water may be added to the growing list of services the technology provides. Then as demand and usage grows, the generator will run for longer and longer periods, giving its users more and more persistent access. Switches, outlets, and fixtures are added to the knowledge base and local micro infrastructure.

As development progresses, the village may be supplied with its own larger generator, initially to power a health clinic, secondary school, and/or government office. But as the village starts stringing wire and forming its own local grid, a growing number of users are added. The village chief, the district commissioner, the school headmaster get their new houses wired. The mosque, the entrepreneur, the international NGO get their establishments connected. And so it continues, until the technology is well embedded and the local infrastructure is well established.[1]

Wayne Marshall • 2110 Conakry Place • Washington, DC 20521-2110

Then, when the happy and ceremonious day comes at last that brings the country's hydropower grid to the village, they are ready. It is a matter of throwing a switch.

A similar development continuum could be described for other technologies, such as the transition from hauling water from the river in buckets to centralized water treatment and household plumbing. Along the way are the accessible technologies, such as rainwater catchment, storage tanks and pumps, that enable people to learn basic plumbing technologies and that build basic local infrastructure. If and when a centralized water supply comes on line, the village is ready; the valve is opened. And perhaps more importantly, the village was not denied the many benefits of plumbing technologies while waiting for a larger public works project to come their way.

The purpose of this paper is to similarly describe the many internet technologies that are available to Guinea—right here, right now—in its own continuum of development toward more complete access to the global Internet. Certainly Guinea today lacks some key telecommunications and other technological infrastructures that impede persistent access to the Internet in the near term. But these barriers need not deter Guineans from broad and rapid adaptation of those same internet technologies that can be used to good purpose on a local scale immediately. Schools, ministries, and NGOs need not wait for local dial-up connections to commercial internet service providers (ISP) to become widely available: they can enjoy the many benefits of email, newsgroups, file transfer, database access, and web publishing on their own networks right away. And in doing so, they will be empowering themselves with key internet technologies, while learning and expanding a knowledge base that will continue to serve their empowerment into the future.

Best of all, many of the barriers to entry for these technologies are extremely low. The state-of-the-art software systems on which the Internet is built are all free and freely available at absolutely no cost. These systems run well on a variety of modest hardware, often considered throw-away equipment in the U.S. And the user community is international, multi-lingual, and self-supporting. With the internet technologies described in this paper, Guineans can literally begin to build the bridge across the digital divide with their own hands.

THE INTERNET EVOLUTION

The phenomenon known popularly today as the "Internet" is itself composed of a number of key networking and communication technologies that evolved largely through academic research and development over the past 30 years. In fact, it is only the general public awareness of the Internet that is a relatively recent phenomenon, due to the emergence of widespread and low-cost access points through commercial ISPs in the U.S. only over the past 7 years or so.

But for a number of years before Netscape and the World Wide Web (WWW) entered the collective consciousness, computer users were going on-line and communicating with each other over campus networks and a growing number of dedicated, multi-user bulletin board systems (BBSs). These systems could host a number of users through persistent connections on a local network, or short-term dial-up sessions over normal telephone lines. These systems enabled email, discussion groups, and file sharing among users through a number of technologies such as UUNET, telnet, kermit and the like, long before America On-Line was ever conceived.

At first each network was a discrete and separate entity: users on-line with one network could communicate with other users on that same network, even if not on other networks. So, a student on a campus network could send email to another student at the same university, or one Compuserve user to another. Businesses and government used similar—even if proprietary—technologies, that allowed users on their own networks to collaborate and exchange information.

Then there emerged a number of store-and-forward technologies that allowed networks to communicate with other networks over non-persistent dial-up connections. This was a much cheaper way of networking over long distances than having dedicated, hard-wired connections that otherwise made internetworking prohibitive for most users. The development of store-and-forward protocols made it possible for a batch of email from one network to be forwarded to another, say, twice a day, or for a day's discussions on one bulletin board to be circulated to others overnight. Communication protocols were designed to efficiently and reliably move data from one system to another, since transmission rates were slow and telephone line charges expensive.

Increasingly this "network of networks" paradigm gathered momentum. At the same time, the hardware supporting the technology got faster and cheaper. Connections became more persistent, data moved faster, more users got access. The networks that once were discrete now formed the permanent infrastructure of that larger network, what would become the capital 'I' Internet. These disparate existing local networks could simply "throw the switch" and merge with the growing Internet.

The purpose of reviewing the Internet evolution in the U.S. is to illustrate how Guinea could adopt internet technologies in similarly successive stages of development. For example, a school, government ministry, or NGO might set up a local network for their own users with freely available internet technologies. Users on these networks would then be able to send email and share work with one another, or put their databases on line with a browser-based front-end. Then, even without capital 'I' Internet access, such schools and minstries could make short, periodic dial-up connections to other networks, batch forwarding email and sharing electronic documents among themselves. As telecomms service permits, these connections may become increasingly persistent, and services such as on-line library catalogues may offer telnet access to remote users on demand. Finally, with the confluence of sufficient demand, investment and technology, viable ISPs and Internet access will become increasingly available in Guinea, and the local network infrastructures will readily transition to full access of the global Internet.

SERVERS AND THE OPEN SOURCE NETWORK

The Internet is built on the multitudiness connection of individual computers known as "servers." Each of these servers may provide one or more "services", what we otherwise experience as email and web pages and Usenet discussion groups. The Internet works because this vast collection of servers all agree to communicate with each other and their clients in the same way—using the same network "languages" and protocols—even if the underlying hardware of the computers themselves is quite different.

The dominant technology of the Internet—the *lingua franca* of network computing—is the UNIX[2] operating system. This operating system was originally developed by AT&T/Bell Labs over 30 years ago to control its massive network of telephone switches across the United States. AT&T subsequently licensed the technology of this operating system to universities and research institutions, where development and refinement of the system advanced continuously.

UNIX was traditionally a "big iron" operating system, running the large mainframes, super computers, and expensive engineering workstations throughout the 1970s and 1980s. A network was often designed around a single, central UNIX computer that would simultaneously serve up to hundreds of users through connected terminals. Because these computers were so expensive, most usage was on a time-sharing basis. It was unimaginable that a single person could ever afford their own computer that would be capable of running UNIX.

Then a cheaper class of computers emerged in the mid 1980s that became known as the personal computer, or PC. These computers were designed for single users to run a single application at a time, and a whole generation of users grew to believe that such devices would eventually make UNIX extinct (if they had even heard of UNIX in the first place), or at least relegate it to the backwater tasks of corporate data processing. For a time, PC usage prospered with the newfound freedom of limitless access to one's own machine—even if that machine itself was quite limited.

Businesses began installing PCs for their workers in mass number. Users crunched their own spreadsheets, processed their own memos. But something was missing: these PCs weren't connected, not to each other, not to the central mainframe database. In fact, there was no easy way to collaborate, share data, and communicate. To fill the gap, commercial vendors began competing with different networking solutions and ways to regain the connectivity lost with the transition from UNIX networks to individual PCs.

Each vendor developed a proprietary solution to add the missing networking technology that the underlying PC operating system was never designed for. Babel ensued, with IBM, Apple, Microsoft, Novell, Banyon, and any number of others vying to provide the different levels of networking services with their own proprietary designs.

All the while development of UNIX systems quietly continued through academic and institutional research, especially those aspects of UNIX related to networking and network protocols, even if much of this development was largely under the radar of the expanding PC industry. As described earlier, universities were figuring out how to connect their UNIX servers to one another, expanding the range of these networks around the world, in what would eventually become the "backbone" of the Internet. At the same time, UNIX programmers were also developing an advanced network-capable graphical user interface—known as the X Window System. Not only did this enable standardization of engineering and modeling applications on UNIX workstation computers, it also permitted the design of "friendlier" interfaces for UNIX users, especially as graphical hardware was becoming more affordable.[3]

The tradition of UNIX in research and academic environments led to a style of open and collaborative development. This tradition generally promoted free sharing of source code, quick turn around of bug fixes, and interactive user support groups. Also, if one's source code was out there for anyone in the world to see—and as it represented one's credentials and programming prowess with one's peers—it was likely to express all the extra care and quality its authors could provide.[4] This style of development differs radically from that of commercial software production, with its closed source development, denial of bugs, and rush to market. (The difference between the *open source* style of development versus commercially developed software has been characterized as the difference between the dynamic interactions of a public bazaar, and the closed and cloistered rhetoric within a cathedral.)[5]

Still, through the late 1980s and early 1990s, the cost of commercial UNIX systems remained prohibitive for individuals and most small businesses, who continued to rely on offerings from the PC market. (Even in the early 1990s, a commercial source license for AT&T UNIX was $50,000.) At the same time PCs themselves were becoming more powerful, and their operating systems were slowly trying to add some of the missing networking and multi-tasking functionality. It was even thought that with each successive generation of personal computer, UNIX would eventually become obsolete.

But the increasing power of PC hardware, along with its decreasing costs, led some creative UNIX programmers to consider an interesting alternative scenario: why not develop a UNIX system that could fully exploit the multi-user, multi-tasking potential now possible in PC hardware? Wouldn't it be totally cool to have one's own UNIX workstation at home, a fully multi-tasking, multi-user, fully network capable computer at a PC price? The appeal of having one's own powerful UNIX workstation available at will and without time-sharing restrictions was extremely compelling for those priveledged to the computing power available in working with such systems. And, after all, the PC of 1990 now had more disk space, memory, and processing power than the "big iron" of just a decade earlier.

This idea seemed to germinate in the early 1990s in two different places at about the same time. The first was at the University of California, Berkeley, where their mature and proven BSD version of UNIX was first ported to run on the Intel 80386 processor by Willam Jolitz. The second place was in Helsinki, Finland, by a graduate student named Linus Torvalds, who proposed in a Usenet discussion group to write a UNIX-like operating system for the Intel processor from scratch, and in collaboration with other users who would volunteer their support over the Internet at that time.

Both projects were in the tradition of freely available, open source systems, where users were invited to participate in the development and ongoing improvement of the systems. The BSD project resulted in what are now known as the FreeBSD, NetBSD and OpenBSD systems. The Torvalds project resulted in what is now known as Linux.

Both projects have since become the powerful and robust systems that dominate the Internet server landscape today, as they have been singularly responsible for the growth of the Internet itself. Of the two, Linux is the most well-known, has the greatest degree of European internationalization, and receives ever-increasing coverage in the general press for its free availability and superior performance over commercial products. The BSD versions are probably the most stable; the author of this paper generally advocates a preference for the FreeBSD system whenever possible.[6] But for all practical purposes, these systems may be considered equivalent.

This equivalence is not only in terms of performance and stability, but also in terms of actual usage. These systems represent the years of effort that have gone into the standardization of UNIX by international standards organizations. All have been ported to a variety of hardware platforms, from Intel, DEC Alpha, Apple/PowerPC, Motorola, Sun, MIPS and others. Not only are these systems designed to be portable among hardware, they are also increasingly portable among nationalities. Substantial portions of these systems are available in German, French, Spanish, Russian, Japanese, Chinese, Vietnamese, and Korean, with locale-appropriate languages and character sets.

Learning to use and administer one system generally enables one to use any other, including the commercial UNIXes, such as Sun's Solaris, IBM's AIX, and HP's HP-UX. Throughout the rest of this paper, the BSD and Linux systems will be considered synonymous, and will be referred to collectively as open source UNIX systems.

KEY INTERNET TECHNOLOGIES

A typical distribution of an open source UNIX system, whether BSD or Linux, may be conveniently obtained from a variety of sources on a set of 2 or more CDROMs at a cost of less than $40. Unlike the legal restrictions imposed by the usual commercial software licenses, these disks can then be shared with others or otherwise copied without restriction. (Less conveniently, these systems may be obtained at no cost by download over the Internet from a number of sites around the world.)

A complete distribution will contain the operating system "kernel", all configuration and support programs, a standard C/C++ compiler, all system source code, a variety of text editors, the X-Window system, full documentation, and an incredible number (over 3,500 in the lastest FreeBSD distribution) of packages of user software. This software includes everything from spreadsheets and databases, word-processors and typesetters, Perl and Python interpreters, WWW browsers, Java virtual machines, photo and image manipulation programs, as well as recreational, educational, and multi-media software too numerous to mention.

Out of this vast array of system software, we will review below the core of programs related to the implementation of internet technologies in Guinea.

1. Distributed file systems. All open source UNIX systems include the NFS network file system server, the standard for serving files transparently over a network. In addition, these systems include the Samba and Netatalk packages, which enable the

UNIX system to seamlessly emulate Microsoft and Apple file servers. This means that a single UNIX box can serve as a central file server to a mixed collection of Windows and Mac desktop machines on a local network. To the desktop user, the files appear to be stored on their own machine. But since they are on a file server, a single administrator can regularly perform a safe backup for all users. By setting up a file server for its users, a UNIX system can also begin to provide its users with the potential for remote access to their documents.

2. Electronic mail. All systems include the Sendmail package, the predominant, heavy-weight mail transport agent used by the majority of email servers throughout the world. In addition to Sendmail, these systems provide Qmail, Postfix, Smail, and other mail transport agents that offer increased performance, security and/or ease of configuration, depending on the particular needs of the site. For single users and small networks, these systems also include the Fetchmail package to pull in email collected from an offline email server, such as from a local ISP. Any of these systems also supports the Majordomo package, to set up, administer, and automate group mailing lists, often used for targeted and particpatory discussion groups.

3. Remote access. All systems may be configured as telnet and ftp servers, to enable remote logins and file transfers over a network. Remote access allows a user or administrator to work on a machine from anywhere else on the network as though he/she were directly connected to it. Similary, ftp allows remote users to move files to and from the remote machine. Many open source UNIX systems now include OpenSSH, a secure shell server that encrypts remote access sessions for increased security against possible snoopers on the network.

4. Web server. The standard open source UNIX web server is Apache, representing nearly 70% of the web servers on the Internet. The WWW is the most hyped of Internet services, and many users may consider the Web and the Internet synonyms. But even without a direct connection to the Internet, a web server can be very useful on a local network—or intranet—to serve as a front end for electronic documents and locally produced HTML content. Users can browse from their Windows or Mac desktop to get the latest calendar and meeting information, staff directory, management reports, or any other information an organization wishes to dissimenate to its internal users. Furthermore, any experience users, programmers and administrators gain from designing and developing content for a local intranet is directly transferrable when they are able to move to the Internet. For networks with only intermittent Internet connections, it is also possible to schedule the UNIX server to pull in selected sites periodically, so users can browse this information as though they were on line.

5. Newsgroups. As an adjunct to the internal communications of a local intranet, a UNIX server may be configured to provide discussion groups known as Usenet News, with either the C News or INN packages. Unlike web browsing, which is a form of passive and one-way communication, newsgroups invite active, multi-way communication among participants. Usenet news is also a classic store-and-forward network application, and a UNIX news server can be configured to selectively pull in (and send out) any number of discussion group topics periodically from other news servers on the Internet. All desktop users on the local network then have the ability to participate in discussions with others around the world through their familiar Netscape browser interface. Also, much technical support for open source UNIX systems is available through Usenet, where peers tend to help one another eagerly and quickly.

6. Nameserver. All UNIX systems can be configured to provide the distributed translation between machine names and numeric IP addresses using BIND, the Berkeley Internet Name Daemon. Although this may be one of the least glamorous of the services described here, knowledge of BIND is one of the most essential for a network administrator to master. This is especially true if the network is of any size, and will one day host its own addresses on the global Internet. A properly configured caching nameserver on an Internet gateway machine will also allow the network to effectively reduce Internet traffic by as much as half, vastly increasing the performance of browser users on the network.

7. Database servers. Any of the open source UNIX systems can be used to install one of the freely availably SQL database management systems, such as MySQL or PostgreSQL. MySQL is extremely popular as a fast, lightweight database server used to provide dynamic web content, while PostgreSQL offers complete transaction processing control and reliably supports heavy demand. In addition to these open source packages, all major database vendors now offer their systems for Linux, including Oracle, Sybase, Informix and IBM's DB2. SQL is a standardized language of commands for accessing a database server, and is the lingua franca of database management programming. As a user gains familiarity and experience with either of the open source systems, he/she is readily able to transfer this knowledge to any of the commercial systems.

8. Printer server, router and other. The capabilities of open source UNIX systems are extremely flexible and depend on their configuration to perform any number of special purpose tasks. For example, a UNIX box can be set up as a central printer server for an office network, to efficiently queue and spool print jobs from a heterogenous collection of Windows and Mac desktops to a single printer. An open source UNIX system can be installed on an unused, outdated PC to perform the duties of a network router, that would otherwise cost thousands of dollars if purchased as a separate piece of equipment. Depending on the network configuration and features desired, UNIX boxes are also set up as gateway machines, fax servers, modem pool servers, Internet relay chat servers, instant messaging servers, and internet telephony servers. Linux systems offer the unique possiblity of sending and receiving TCP/IP packets over shortwave radio frequencies using the AX.25 protocol, a feature that may represent a viable intermediate option to remote networking in Guinea where telecomm service is otherwise so limited. Because these systems are based on an efficient multi-tasking, multi-user UNIX kernel, it is customary for a single machine to perform any number of these services simultaneously.

9. Security and firewall systems. Once a network of any kind is publicly accessible, it needs to be protected from unauthorized access and the potential of attack. Any of the open source UNIX systems comes with a large collection of tools and configuration options to provide strong security and network defense.[7] While recognizing that no network is ever 100% free from security risks, open source UNIX systems have a proven track record of providing far more security than any commercial, closed-source, alternatives, simply because they are open. Any bugs or weaknesses that hackers might exploit in these systems are also visible to the legions of developers around the world who regularly build and maintain them. Security holes are “plugged” as soon as they are found, and alerts are disseminated immediately to all users advising them how to correct their systems. Any UNIX system administrator is responsible for, and will gain valuable experience in, configuring a network to be secure and monitoring for intrusion.

Because of its modest minimum hardware requirements, an open source UNIX system is frequently installed on an older unused computer, donoted equipment, or something put together from spare parts. At a minimum, a system should have the equivalent of an 80486 class processor, 16MB of RAM, and a 200MB hard disk. Naturally performance is better and installation is easier with a PCI-bus Pentium class computer, 32MB or more of RAM, 1GB or more of hard disk, and a CDROM drive, but these are still very modest requirements by today’s standards. Even if new, such systems can now be purchased or assembled from parts for less than $500(USD). All systems on the local network will also need a 100Mbs ethernet adapter, now widely available for under $20(USD) each.

The systems described here assume typical installation as unattended network servers, where a single UNIX box or two will just sit on the local ethernet network, each quietly performing its respective duties with aplomb, 24 hours a day, 7 days a week.[8]

Many of these open source UNIX systems are fully capable of multi-threaded, symmetrical multiple processing (SMP) with Intel or Alpha processors, meaning the ability to use motherboards with two or more CPUs. Such performance is only necessary under extremely heavy loads, or massively intensive computing tasks,[9] and most usage in Guinea will be presumed to be more modest. But it is important to note that the fundamental architecture of these systems is highly scalable to serve any future demand required.

NETWORKING IN GUINEA

Given the positive attributes of freely available, open source UNIX systems described here, it should not be surprising that a recent survey of web servers in several West and Central Africa nations found that 100% of these sites were hosted on Linux systems. Unfortunately with respect to the “digital divide”, there were only 32 such systems in all of Chad, Niger, Liberia, Equatorial Guinea, and Central African Republic *combined*. This compares to, say, Moldova, an extremely depressed former soviet state in Eastern Europe, yet still with over 800 web servers, two-thirds of which were noted to run Linux.[10]

Conspicuous by its absence in the African statistics is the Republic of Guinea. The balance of this paper discusses what can be done to address the “digital divide” in Guinea, particularly through the advancement of open source UNIX systems described above.

One of the main problems of adopting this technology anywhere—not just in Guinea—involves an issue of literacy. And by literacy we do not refer exclusively to a lack of technical knowledge by the non-technical. For it is frequently the case that even those who consider themselves technically fluent in, say, desktop systems, may be uninformed, underinformed, and/or misinformed in the arena of network technologies described above. Because of the intense and high-profile marketing campaigns by commercial system vendors, many people with technical responsibilities may still be unaware of the core technologies that underlie the Internet and that are freely available. The key point here is that those who would teach must first themselves learn.

Yet having suggested the issue of literacy, we will not pursue it further, at least not directly, not formally. Rather, if it has been said that the Internet developed like a public bazaar, and if it is true that markets are conversations, then it should follow that networks themselves are built with conversations. The means to advancing networking technologies in Guinea, then, is to have conversations with people.

This is the informal, person-to-person, voluntary, enthusiastic and personal involvement approach on which the Internet was built in the first place. For in spite of the high-profile, billion-dollar IPOs, and gold-rush mania of the Internet in the media, by and large the Internet is what it is—internet technologies are what they are—simply because some people talked about what would be cool to do with computers.

Similarly in Guinea, the means to success will be to arouse interest, provide compelling demos and hands-on experience, help with installation assistance, stimulate the learning and sharing of ideas in configuration and administration, all through growing the networking conversation with an active and mutually supportive, peer-to-peer user community.

One typical example with which the author is personally familiar—though which is instanced thousands of times over throughout the world, from Mexico to Italy to India to China—is the Portland Linux User's Group (PLUG), in the U.S. state of Oregon.

The PLUG has several components that make it a successful model for advancing open source UNIX and internet technologies. These include:

- Regular user group gatherings
- Beer drinking
- Monthly installation clinics
- Eating pizza
- "Linux in the schools" demonstration project
- Sharing disks and CDROMs
- Active web page and consultants listing

Some of the perhaps lesser of these points are considered in turn below:

1. Gatherings/meetings

Open and largely informal meetings are scheduled monthly in a room made available by the local university. Attendees are self-selected and represent anyone interested in open source technologies. There are no fees, dues, or restrictions. Participants come from a broad range of technical abilities and professional backgrounds, including: employees of local ISPs; network administrators from business, government and non-profit; computer science students; web developers and technical writers; and general computer "geeks" come to find out what this Linux thing is all about anyway. People are friendly, even if many frequently have a tendency to shyness, and no question is considered stupid. Everyone recognizes they were "newbies" once themselves, and take pleasure in helping their peers.

Meetings usually follow a format of open conversation for awhile, somebody might mention an article he/she just read, or some other hot topic in the news/Internet. This is then followed by voluntary presentations on a particular topic, as scheduled from the preceeding meeting. For example, someone might demonstrate how he set up network address translation (NAT); another might show how she configured Usenet news. Sometimes a guest speaker is invited from the local industry to speak about wide area protocol (WAP) or geographical information systems (GIS). The university facilities are a little cramped sometimes, but they kindly allow the use of their projection equipment.

Some business does get done. Volunteers are solicited to staff the Saturday clinics; a timetable is agreed for an upcoming trade show; suggestions are made for PLUG's web page. The meetings (which are held on the first Thursday of the month) break up about 8:30pm or so, and many of the participants then slouch on over to the nearby brew pub for some beers.

2. Install Clinics

The PLUG organizes installation/help clinics one Saturday morning each month. The location is made available by a local secondary school, and the clinics are staffed by volunteer PLUG participants. These clinics are primarily oriented to help anyone who is interested in installing Linux on his/her own computer. People bring in their own computers, both IBMs and Macs, and PLUG volunteers take them through the installation process, with any current version of Linux they may want to try out. Installation can sometimes be a little tricky in spots, so having the help of someone who has been through the process a few times can make it a lot easier.

In the fall and winter of 1998, the popularity of the clinics was rising quickly, and it was typical for volunteers to assist with 15 to 20 installations per weekend. As with the regular monthly PLUG meetings, the install clinics are publicized in the local computer trade press, as well as the group's web page.

3. "Linux in the Schools"

The site of the installation clinic is also the home of PLUG's main demonstration project for their "Linux in the Schools" program. This project was first designed to demonstrate a "real-world" example of using inexpensive open source UNIX systems to power a mixed network and computer lab of Windows and Mac machines. The project also opened a gateway to the Internet that the school could otherwise have not afforded.

The project first began around 1996, when many people were still sceptical about the capabilities of a system that cost no money, and it proved persuasively how viable and valuable such systems could actually be. Not only did this Linux system provide networking tasks and Internet access for the school, it also enabled the students themselves to have a system on which to safely develop their own web content and gain valuable internet technology experience hands-on.

In the spring of 1998, PLUG received six computer systems in a grant from Intel, and then solicited applications from other schools in the area to receive these systems as part of a package that included formal training in Linux and network administration. PLUG members designed the training curriculum that included the basic installation, configuration, and administration knowledge that would be necessary to successfully run these systems in a school setting. Many more applications were received than could be accepted, and in my last connection with the group in the late 1998, plans were underway to continue to solicit more computers to expand the program.

What can be concluded from this example of PLUG, especially as these conclusions might relate to Guinea? Certainly no statistical measure is available, to say just how much access to internet technologies increased due to PLUG's modest activities and activism. But it should be evident that internet technologies by their very nature are highly leveraging in their effects: one single server can bring a whole school on-line; a single Linux "newbie" can empower dozens of his/her colleagues and coworkers. As a personal example, in the fall of 1997 the author put the American Red Cross in Portland on-line, with a single Linux server that gave over 100 employees and volunteers unlimited Internet access from their desktops, as well as powerful new intranet communication technologies for the organization's management. This project might otherwise have not have been attempted without the encouragement of PLUG's demonstration sites.

Of couse this is not to understate the challenges faced in a developing country, nor to fail to recognize the numerous social, political, and economic circumstances where open source UNIX technologies cannot help:

- No literacy among users to read screens and operate keyboards
- No electricity
- No facility to house equipment
- No rule of law that protects property and investment

There is no panacea for these conditions, and certainly no silver bullet of technology is suggested here.

But the informal, conversational style of PLUG is a microcosm of the same energy and volunteer enthusiasm that built the networks that became the Internet, where none existed before. One could reasonably expect similar results from similarly modest efforts in Guinea. Certainly in Conakry, if not throughout Guinea, there are already fledgeling ISPs, network administrators in schools, ministries and NGOs, and the technical enthusiasts who are eager for such networking conversations to begin. Gather these key people together, a meeting room, a few computers and some good beer, and find there the humble beginnings of a GLUG...

CONCLUSIONS

This paper has described the development of the Internet itself as a model for enabling internet technologies in Guinea.

One of the key propositions is that access to the capital 'I' global Internet is not a preliminary condition to the productive use of internet technologies, right here, right now. Specifically, a host of powerful—and empowering—world-class information technologies are freely available for schools, government ministries, NGOs, entrepreneurs and other individuals, with extremely low barriers to entry.

The more these systems are in place, the more widespread will be the infrastructure and technical expertise to support even further growth of these systems. The more this infrastructure is in place, the more rapidly will access to the global Internet be spread throughout the country, as ISP and telecommunication resources do inevitably become more available. And that availability itself will be at least partially stimulated by the demand created through the growth and development of local information networks discussed here.

Finally, this paper has described networks as conversations, and conversations as a means of advancing networking technologies among the people interested in using them. The conversations themselves may take a variety of forms, and one particular example has been illustrated by the successes of a local user's group in the author's hometown. But whatever the actual form, the key attributes of an effective conversation will be those of the open source technologies themselves: peer-to-peer communication, openness, involvement, energy, commitment and enthusiasm.

NOTES AND REFERENCES

1. The author observed at first hand this stage of development during two years in Tutume, Botswana, a small village on the edge of the Kgalagadi Desert in southern Africa. In the early 1990s generator usage was sporadic and intermittent, except that one always knew when there were bodies in the morgue, as this was the only time the generator would run all night. As time went on, though, the generator ran more reliably, and almost constantly.

2. UNIX is a now a registered trademark of The Open Group.

3. A superficial copy of the Motif user interface on UNIX would become the model for Microsoft Windows 3.1.

4. The developer of the TEX typesetting sytem, for example, openly offered users $2.56 for any bug found in his software, doubling that for each additional bug found. His last payment was $327.68(USD) in March 1995. *Digital Typography*, Donald E. Knuth. 1999, CSLI Publications; Stanford, California.

5. *The Cathedral & The Bazaar*, Eric S. Raymond. 1999, O'Reilly; Sebastopol, California.

6. The largest site on the Internet, Yahoo, is run exclusively on FreeBSD servers. So—ironically—is Microsoft's own Hotmail. And the world's most active ftp server, Walnut Creek CDROM, is run on FreeBSD servers. (See http://www.freebsd.org/). Apple's upcoming desktop operating system—MacOS X—is built on a FreeBSD foundation.

7. One BSD distribution in particular, OpenBSD, is designed specifically with network security in mind, and is widely regarded as the system of choice for protecting a network from intrusion: "Three years without a root hole in the default install." (See http://www.openbsd.org/).

8. For other uses, such as desktop workstations for business, school or home, a good video graphics card will also be desirable. In fact, open source UNIX systems now make great desktop/laptop machines, and several free or low-cost commercial office suites are now available, including StarOffice, WordPerfect, and Applix. In the author's opinion, open source UNIX systems are the perfect and most strategic choice for developing countries such as Guinea, and as some countries such as China may themselves be deciding as official policy. (See "China Stiffens Its Resistance To Microsoft: Price and National Wariness help Linux-Based Red Flag", International Herald Tribune, September 5, 2000.) But here we have intentionally reserved the topic to the server-based applications, and otherwise avoid the religious controversies that may inflame discussions of preferred desktop operating systems.

9. The high tech graphics imagery of the 1997 blockbuster film *Titanic* was performed by a cluster of 105 interconnected Linux computers. See "Linux Helps Bring *Titanic* to Life," Linux Journal, February 1998.

10. From "LJ Index—July 2000," Linux Journal, July 2000. For reference, Germany alone recorded over 460,000 web servers in this same report (nearly 200,000 hosted on Linux.)

ABOUT THE AUTHOR

The author is a UNIX programmer and technical consultant who has specialized in the use of open source UNIX systems since 1993. He has also worked as the senior financial and technical officer of a number of volunteer development and social service organizations since 1986. Currently he lives in Conakry, the capitol city of the Republic of Guinea, in West Africa.

SIDEBAR: SO WHAT ABOUT MICROSOFT?

In general, Microsoft operating system products are not considered appropriate for providing the internet server technologies described in this paper, primarily for reasons of performance and reliability. In spite of extensive marketing claims to the contrary, Microsoft's efforts to implement prorietary networking technologies cannot hope to compare with the decades of open source development that have gone into the UNIX systems. As a consequence, those users of Microsoft systems are burdened with regular system crashes, data loss, and security breaches that rarely, if ever, affect the UNIX systems described here.

But as specifically concerns the development of internet technology in Guinea, there are even more compelling reasons to steer a path clear of Microsoft. First is the cost. License fees for (legitimately) setting up a Microsoft network server will run into the thousands of dollars. And because these systems are not as efficient as open source UNIX systems, their hardware requirements are much greater, also resulting in increased costs. The total increase in cost of several thousand dollars per server represents a significant barrier to entry for the typical Guinean individual or organization that might otherwise be able to access internet technologies with an open source UNIX system. Guinea is an environment where it is particularly crucial to remove any barriers to entry due to cost.

Second is the issue of empowerment. Open source UNIX systems are designed to empower their users, with vast and increasing quantities of high quality documentation online and thousands of enthusiastic and supportive users throughout the world. Microsoft systems, on the other hand, are more of a closed box, good documentation is an added cost, and service and support are by contract. The Microsoft paradigm is to keep users rather uninformed and dependent. These are the natural financial objectives of a commercial entity, and result in the binding of users in a perpetual cycle of dependency, paid upgrades, and paid support. While they offer the promise of ease of use and point-and-click administration, the truth is, if they don't give you a button to push, or if nothing happens when you click, you are pretty much out of luck. Open source UNIX systems, on the other hand, attempt to be transparent, flexible, and extremely configurable through simple and accessible plain-text files. UNIX users are empowered to do whatever they want to do with their systems, modify them at will, and even give copies to anyone they choose.

Finally is the sense of community. Users of open source UNIX systems tend to gain pride and a sense of belonging in joining a global community of their peers. As we have discussed elsewhere in this paper, the Internet is a conversation, a bazaar and free-flowing exchange of ideas, owned by no-one, superseded by no entity. The idea is as empowering as democracy, and the sense of belonging to that borderless community can be profound and liberating. Microsoft cannot hope to engage this same sense of community and activism amongst its users, who are merely as serfs to a feudal lord. Only among those who feel the freedom to do so will there develop the activism and selfless energy that are crucial for the building of network communities.

It is vital for anyone or organization working to stimulate the advance of internet technologies in Guinea to both understand and embrace the fundamental differences between these two paradigms. Not only is it important to be able to articulate these issues, but credibility and leadership will require an actual demonstration of these principles in action. Consequently, any group working to effectively promote the effort of building networking bridges in Guinea should also be active in building the networks in their own organizations similarly.