 

# M **Memo (Draft)**

**To:** Bill Gates

**From:** Steven Sinofsky

**Subject:** **Computing at Cornell and the Internet**

**Date:** Feb 13, 1994 (50)

**cc:** LIST

## Introduction

This memo is a brief overview of the use of the Internet at Cornell University. I was recently stranded there for 2 days because of the weather and used the time to meet with some people that I worked with when I was an undergraduate. While walking around campus it became abundantly clear to me that in the year since I was last there the computing environment had undergone a radical change.

The purpose of this memo is just to show how much can be done in a short time using mostly free software. The key backbone, as we will see, is the universal adoption of the TCP/IP protocol along with a large number of free interfaces and protocols. J Allard has written an excellent memo that gives a brief overview of the main services available on today's Internet. He also includes some detailed recommendations to Microsoft to follow. I have some similar recommendations at the end of this document.

I apologize in advance if I sound a little bit like a Cornell cheerleader or romanticize a bit too much. This is my alma mater after all ('87). I am in an interesting position to observe the changes outlined in the memo since I was at Cornell just prior to the introduction of the Macintosh and worked for the information technology group for the four years that I was a student. I have also been back to Cornell at least yearly since 1987.

## Background

First I just want to give a brief background on the university and on the computing environment. Cornell is located in a rural central New York town, Ithaca. The Cornell community is about 22,000 people including 12,500 undergraduates, 5000 graduate students, 1500 teaching faculty, 500 or so research associates, and the remainder are full-time administrators, including those that run the computing environment called Cornell Information Technology (CIT).

The university is divided up into a number of schools, which are located at various centers ("quads") across the 450 acre campus. The schools include Arts and Sciences (the largest), Engineering, Architecture Art and Planning, Agriculture and Life Sciences, Human Ecology, Hotel Administration, Johnson Graduate School of Management, and Industrial and Labor Relations. Each college acts at once as an autonomous unit, free to pursue any computing directive it can afford, and simultaneously as part of the university community by leveraging the main CIT maintained resources.

In the past Cornell has concentrated its computing efforts towards providing universal access, even if it meant access on relatively archaic systems. Since at least 1975, all incoming students were automatically given computer accounts and prior to that accounts were available for the asking. Starting in 1983, just before the Macintosh, most freshman writing courses used (and some required) word processors either on PCs (DisplayWrite) or on the mainframe systems.

Historically, CIT has maintained the several mainframe based computers, including two large IBM 3090's running VM/CMS, one which is available to students and one reserved primarily to run the business side of the university. There was also a public VAX running UNIX. Until the Fall of 1993, students would receive a free account on the IBM and on the VAX. These two accounts would be separately administered and students would use whichever one they were more comfortable with, which was increasingly the VAX. Larger departments maintain any number of smaller machines. The agriculture school has traditionally

 

managed the largest number of PDPs and MicroVAXen outside of DEC (apparently, VAXen an be easily connected to cows and plants). Engineering has a number of VAXen as well running UNIX. Computer Science has a large number (50+) of public Sun workstations, in addition to the workstations maintained by each research group in the department.

Cornell was one of the first major BITNET sites in the US. As one of the top three recipients of NSF and DARPA funding in the 80's, Cornell quickly became a major USENET node (BITNET evolved into USENET when DARPA funding was added). The fact that the IBM was one of largest public machines in the region also helped. With the receipt of one of four NSF super computer seed grants in 1986, Cornell became a major hub in the new NSFNET (when the NSF and DARPA could not agree on network use, NSF formed its own network). Cornell has T1 lines throughout the campus and leading to a number of other northeast computing sites. In 1986 the entire campus was rewired with fiber optic cable for a phone system, and at that time each dorm room added a tap for future computer use (with two large dorms participating in a pilot project for in-room net connections).

In 1984 Apple Macintoshes took over the campus as far as undergraduates were concerned. Cornell was a charter member of Apple's university consortium. There are a number of staff programmers at CIT that are Macintosh specialists. Students can easily and cheaply purchase Macintoshes directly from CIT. At the height of CIT maintained machines, there were over 600 public Macintoshes on campus available in the library, residence halls, and in computing centers throughout the campus. As more and more students started purchasing their own machines and has more and more faculty purchased machines, CIT downsized the number of machines it maintains. Today there are easily 2000 "full-time" Macintoshes and each year roughly one-third of incoming students have a personal computer.[1]

Over the past two or three years I have seen the number of Windows machines grow on campus a lot. The biggest reason, from what I can gather, is that students are purchasing computers before they get to college. The computers that are cheaper and easier to obtain outside a college environment are Windows machines, especially because many parents can get discounts on PCs from their place of work. I saw a lot of ThinkPads and PS/1's in students hands.

Until 1993, the campus network consisted almost entirely of VT100 terminals connected to a switching system that allowed access to the IBM or DEC machines. In the larger CIT run clusters, Macintoshes replaced the VT100s and were used for dumb terminal emulation, and for word processing. In addition, these larger CIT clusters were connected with AppleTalk for simple file and printer sharing. Each cluster has a few dot matrix printers (original ImageWriters) with A/B/C switches. Each cluster also has a LaserWriter for use with a Vend-a-Card money card ($0.15/page).

This November, 1993, all of the Macintoshes were upgraded from their AppleTalk networks to full EtherTalk including MacTCP as the basic protocol. This helps a great deal as all the Internet client services assume TCP/IP. Each Macintosh on the campus network is a separate IP node and has its own IP address (i.e. 192.135.191.128). CIT provides help to staff and faculty in the selection and configuration of hardware so that new machines can be easily integrated into the campus network.

Although at first sight it would seem that the university is entirely a Macintosh school, a number of large sections of the university are entirely or mostly PCs, but only a small number of those are Windows machines. In fact, the current selection of a new dean in the Human Ecology school has basically become choice between two candidates, one of whom wants to move more towards PCs and another who wants to move more towards a Macintosh environment. Several other schools are mostly PC based included Management and Hotel (because of Lotus 1-2-3, though they are rapidly moving to Excel), and Human Ecology and Agriculture (because of a large investment in MS-DOS based courseware). Still, the students in these colleges do the mainstay of their work on Macintoshes, and concentrate only course project/homework specific work on the PCs. Right now the main issue stopping adoption of Windows is a lack of hardware, since most recent hardware purchases have been Macintoshes, and a lack of expertise in CIT on administering Windows based machines that use TCP/IP.[2] Additionally, most of the client-side software for Internet services are MS-DOS or very early pre-release Windows versions.

[1] Cornell is really expensive and adding another $1000+ dollars for a computer is simply not possible.

[2] In fact, CIT does not even know if Windows is capable of running TCP/IP like the Macs do. In addition, there is a lack of client side software for the various Internet services.

 

## Paradigm Shift

I have been back to Cornell at least once a year since I graduated in 1987 so I have seen the gradual changes over the past 6 1/2 years. The past six months have seen revolutionary changes in the way computers integrate into university life. I am sure there are smaller schools where computers are even more a part of every day life for students and faculty (Dartmouth), but to see an institution of over 22,000 people fundamentally change the way that Cornell has in this time is truly an inspiring sight. One really needs to understand just how huge, sprawling, and heterogeneous the Cornell computing environment is today. Every large research grant purchased unique hardware and software: PDP machines in Agriculture, vector processors in physics, a Prime in chemistry, Sun workstations in computer science, Macintoshes in the Freshman Writing Program, etc. Yet today all of these machines are held together under a common access umbrella, mostly driven by a few common interfaces and TCP/IP.

Three cultural observations had the biggest impact on my thinking. First, new students at the university, having not been exposed to the antiquated systems, took the existing system totally for granted. For them the on-line services are as ubiquitous and expected as the regular phone service. In fact, the daily newspaper routinely prints editorial letters on the current inefficiencies in the computer system much like our MicroNews. For example, while I was there I read letters about the lack of an on-line course registration/sign-up mechanism and a lack of digital images for ID cards. The rise in information access at Cornell has been greater than exponential. Services like e-mail went from less than 1% penetration to 100% penetration in practically no calendar time. This pace of change in information access is faster than any other technology I have seen in my lifetime, including the personal computer itself.

Second, the fact remains that the vast majority of the software being employed for mission critical use at the university was free. In fact, all of the software described below was obtained and used without modification. The only use of commercial software is for word processing and spreadsheets. CIT employs a number of students every semester, usually computer science majors, to hack around on various tools and applications, sometimes even fixing bugs in the public domain packages. In fact, to maintain the public domain software, Cornell regularly submits bug fixes to the originators of the software (i.e. University of Minnesota for Gopher, University of Illinois for Eudora Mail, etc.)

Third, the Internet "fad" that is so prevalent in the commercial computing environment is nothing new at universities. In fact, the people I spoke with were even smug about it, with somewhat of an *I told you so* attitude. The phrase "Internet" has been part of university computing for at least 10 years and all indications are that each day more and more applications are being designed assuming the presence of this network.

The mainframe based systems were phased out over the course of the Fall 1993 semester. New accounts are all based on login IDs and except for legacy faculty projects, no new work is being done on the mainframe systems, except for university business processing.

It is easy for me to look at this system and see a lot of problems. Some of the software is not as stable as it could be, some of the interfaces are goofy, the fact that the heterogeneous nature of the environment makes for a lot of inconsistencies in access, etc. Cornell still has a lot of separate machines to administer accounts on, etc. But I walked away thinking far more about the possibilities for the future than the problems with the present.

For more details on the workings of the services and protocols described below please see J Allard's memo or the *Whole Internet User's Guide and Catalog* by Krol.

## User ID

Key to accessing the entire Cornell information infrastructure is a login ID. These used to be assigned randomly as 4 letter combinations. Today these are done using the initials you provide to Cornell at the time you register with the university. When there are collisions a number is automatically added to the end of your user name. The switch from these old IDs to new login IDs was not unlike our switch to NT Domain passwords, except far more people accomplished the change-over in far less time.

 

At the start of the semester CIT sets up a booth at registration where students can go to sign up for a user ID. This is even easier then getting your picture taken for the ID card. You simply swipe your validated student ID through a card reader and enter a new password and you have a login ID. Additionally, a post office (mailbox in our terminology) is created for you on one of 3 post office servers. You also receive a booklet detailing the computing environment.

Currently Cornell has over 22,000 login IDs in service and more than half are active in any given month. In the April alumni newsletter there will be instructions on how any alumnus can obtain a login ID and access the information services over either asyncronous lines or direct Internet connections if available, this includes e-mail (i.e. being a Cornell alumnus gives you free e-mail!)

An ID is required to access e-mail, Just the Facts, Net News, and a couple of other services. An ID is not required for CUINFO and Library catalogs.

## Bear Access

CIT has one single application that requires significant effort on their part to write and maintain. The application **Bear Access**[3] is a common front end to all of the university information providers, written in MacApp. This package is basically an application launcher that can also replace the Finder on the Macintosh. There is also a very unstable MS-DOS character mode version that is used at a few locations. All the public CIT Macintoshes run System 7.0.1 and Bear Access. These machines also run MacTCP and each one is a node on the Internet, externally visible to the world.

Two key features of Bear Access help CIT to manage the machines easily. First, the software can be automatically updated at runtime and version checks are done prior to running any of the sub modules. This is important because as with many in-house applications, new releases are done quite regularly. Second, the software is easily configurable. Menu items can be added or removed interactively and there is an administrator password scheme enforced throughout all the public machines that prevent arbitrary changes from being made.

Most students do not use Bear Access once they become familiar with the basic applications being launched, since each one can also have a desktop icon. Kiosk machines, however, can only be accessed through Bear Access. These machines are found in the library, and at various public locations (student union, department offices, etc.) all run Bear Access as a replacement for the Finder.

Applications are chained together using a public domain program called Launcher that makes sure Bear Access continues to run along with only one copy of each of the other applications.

Through Bear Access one can get to most of the information services at Cornell including:

- Electronic Mail
- Traveler's Electronic Mail
- CUINFO/Gopher
- Library Resources
- Network News (USENET)
- Just the Facts for bursar and registration information
- Who I Am community directory service
- Chat
- File Transfers between TCP/IP machines (FTP)
- Campus Host Machines (TELNET)
- Campus Store
- Bear Access Information

Each of these services will be described in some detail below.

[3] Cornell's mascot is a bear that looks surprisingly like Misha the Soviet bear from the 1980 olympics.

 

Since students cannot currently have TCP/IP access into residence halls[4] and there are no off-campus TCP/IP connections, CIT has a system called *EZ Link Remote Access*. Remote access takes advantage of a dial-up service maintained by the university as a fee for use system ($8.50 for 60 hours for an individual, with shared accounts and group accounts available). The software required is a MS-DOS or Macintosh implementation of SLIP, the serial line Internet protocol. On the Macintosh there are three versions of SLIP software currently available, two commercial, and one freeware version released by InfoCon, a company that sells other communications packages. A large number of faculty have SLIP connections from their homes. Because of the costs, not as many students have full Bear Access in their residence halls and apartments, which is sort of a shame.

Along with MacTCP, each Macintosh is setup with Comet. Comet is a CIT written terminal emulation program. It has some neat features such as screen printing and multi-line buffering. On the whole, though, it is somewhat clunky and shows its roots in VT100 and 3270 emulation. Some of the 3270 functions are required when accessing the library's on-line catalog.

The Macintosh clusters supported by CIT all run a standard suite of software including System 7.0.1, Bear Access, MacTCP, Symantec THINK C, Microsoft Word, Microsoft Excel, Lotus 1-2-3, Write Now, and some specialized clusters include Adobe PhotoShop, Quark Express, and/or PageMaker. At each site these machines are all connected on AppleTalk to a Quadra server that allows for file sharing for course work, and for administration. In order to keep these machines consistent, stable, and up to date, each day a script is run on the Quadra that goes to each machine and resets the contents of the hard drive to a known state, including removing any user created files, checking for viruses, etc. The standard Macintosh configuration is roughly a Mac LCsi, with an 80MB drive, and 12MB of memory (the memory was added this Fall). There are more powerful machines in some of the graphics design parts of campus, Human Ecology and Architecture.

## Electronic Mail

The single application driving the student to use the on-line services is electronic mail. As stated, the creation of a login ID automatically creates a post office (i.e. mailbox) on one of three Cornell e-mail servers (UNIX boxes). These postoffices run POP, a Cornell post office protocol.

E-mail had already made significant inroads into various departments at Cornell over the past few years. Of course computer science was the main driver, having an ARPANET connection 10 years ago. Several administrative branches had developed a reliance on QuickMail for the Macintosh, administering these systems with some help from CIT. This heterogeneous e-mail world created problems in transitioning to the post office based system. CIT created a single address book mechanism shared by all users with login IDs, which is discussed below in the section Who I Am. The key entry that ties the disparate e-mail systems together is a preferred e-mail location, which can be any valid Internet address. This is totally end-user administered so if a user moves to a new department or changes workstations, one simple change can have all mail, internal and external, rerouted automatically. This is essentially a UNIX .forward file with a nice user-interface. The main use for this feature is so that everyone can give out their e-mail address in a canonical format, for example `sjs@cornell.edu`. If I then get assigned a workstation called `nimble`, I can simply change my address book to forward mail to that machine, rather than tell all my friends to change my e-mail to `sjs@nimble.cornell.edu`

The e-mail software used by most people is the Macintosh application Eudora, written as a freeware program by Steve Dorner at the University of Illinois. The program and author are now part of QualComm, which will continue to provide a free version, but will also have a commercial version (much like StuffIt has evolved). Eudora is a relatively complete mail program and includes all the features we commonly associate with e-mail, and then some. For example, it supports attachments using the new Internet MIME protocol. It also includes a number of features designed for slow link aware access, such as browsing headers and not downloading "big" messages. It also includes the famous Internet *signature* feature so each message has a bunch of text at the bottom identifying you as a unique person with a quote

[4] There are two pilot residence halls with direct TCP/IP taps in rooms. Other halls have clusters in the basement for TCP/IP access using Macintosh machines.

 

or other information. This is a very popular university feature not found on most commercial packages. The Mac Eudora client runs in under 400K of memory.

Students are forced to save their e-mail on a floppy if they wish to keep it more than 60 days since the POP servers delete mail older than that. The downside of this is that it takes a while to fill a floppy with your mail. If you have a direct Internet connection on a home or office machine, of course you can use a hard drive. Eurdoa supports multiple mail boxes, so archiving mail on a hard drive is not a problem. One of the most interesting cultural things I witnessed was the mass of students that jump into a computer center at lunch and between classes, sticking their floppy into a Mac and checking e-mail–not too far off of what things are like at Microsoft.

For faculty and staff that travel a great deal or for people that do not want to deal with SLIP, CIT has a terminal emulation interface to the POP servers. This way you can easily read mail on the road using a relatively low-tech solution. This is not unlike how many Microsoft people use the MILDRED accounts still today. Mail is viewed, replied to, deleted, etc. on the POP server and can be later downloaded to a Eudora mail box. This is used quite a bit since it does not cost anything, unlike SLIP dial-up access which has a monthly fee.

One of the most empowering e-mail features was the extensive use of mailing lists. In order to avoid administrative hassles, Cornell makes extensive use of the old line LISTSERV program to create and manage distribution lists. Essentially this is how most UNIX information is disseminated in the Internet world. One sends a message to `LISTSERV@hostname.edu` with a single line such as `SUBSCRIBE ALUMNI-NEWS`. Your name is then added to the distribution list. Only the creator of the list can send mail to the entire distribution list, though anyone can request a complete list of recipients. The White House is currently using LISTSERV software for the NII demo mailings that go out. Many faculty have started using these lists to maintain mailing lists for classes. The list to subscribe to is printed in the course syllabus. More frequently, these lists are used for student organizations.

Cornell has undergone *the* full cultural change with regards to e-mail, in a manner that is consistent with e-mail invasions at all organizations. E-mail has become an acceptable form of social interaction. Students even encourage their parents to get accounts, mostly on AOL, so they can exchange mail rather than call or write letters. Since Cornell is a notorious bastion of *political correctness*, it is only expected that there are all sorts of policy issues being debated regarding etiquette and harassment via electronic means.

It will still be another year or two before all courses use e-mail in a ubiquitous manner. For example, most technical courses set up mailing lists and faculty expect to be able to communicate with students electronically. This is especially helpful in large classes, or in smaller informal classes. The use of list servers clearly facilitates this, though there is still some hesitancy in some of the older departments towards total commitment to e-mail.

## CUINFO/Gopher

Starting in 1979 Cornell developed an on-line information service called CUINFO. The system provided information about lectures, entertainment, class schedules, and a phone directory. The system was based on VM/CMS and used a huge 370 assembly language news reader program. Content was massaged via REXX scripts and imported into the program. Simple terminal emulation was all that was needed to access the service. There were public kiosks in a number of key locations (student unions, information center, and computer clusters).

In 1993 all of the content was moved to a Cornell Gopher service. Gopher provides a hierarchical navigation facility ideal for the type of information contained in CUINFO. The service is now accessible from the Bear Access system and from remote telnet access (demo on request). The current Gopher hierarchy for the Cornell server is over 800 pages if printed out. The current information is all read only in the sense that only the Gopher administrators add new information, which is the intention of the gopher architecture. This is the top level of each subject. A sampling of information available on CU Gopher includes:

 

- Academic Life: course roster, text book requirements, fellowships, scholarships, undergraduate research projects, and in general anything that used to be posted on a bulletin board using standardized forms.
- Administration: student and staff directories including search, personnel services, workshops, administrative policies, job openings, day care, working family issues.
- Dialogs: interactive dialogs for sharing ideas including *Ask Uncle Ezra*[5](any and all issues), Letters from Eunice (diversity issues), and nutrition information.
- Library: schedule information, acquisitions since 1973 (see section On-line Library Catalog), electronic books (text based).
- Student Life: jobs for work study students, campus events (Cornell Cinema, music, theater, sports), movies and reviews for all local theaters (non-Cornell), concerts (Chamber orchestra, folk music[6], etc.)
- Campus: history, geography, trivia, calendars, exam schedules, news, and weather (from Cornell meteorology department), building hours, dining hall menus and hours, UPI news feed.
- Ithaca and environs: bus schedules, hotels, restaurants, local airline schedules.

In addition, the Cornell Gopher server will connect to all the other public gopher servers around the world and allow you to access their hierarchies. Although Gopher content is interesting, it is limited to text based articles. There is a **huge** movement to start to publish all content in WWW[7] format. By next year the Cornell gopher server will probably become a WWW server. Estimates are that WWW content is growing faster than the Internet itself, perhaps **3% per week**!

The Gopher server also has indexed searches on a large number of key areas. Some of the examples of searches that are enabled are:

- locate a work study job that only requires 10 hours per week during the day near the ag quad (to find a job that meets in between classes near my other classes)
- locate an Art History class that meets MWF afternoon and is 4 credits (to meet a requirement)

The Bear Access program launches a Macintosh Gopher viewer called Turbo Gopher. The user interface is not unlike the file manager since it is basically a hierarchical viewer. There are also a large number of other viewers available to anyone in the public domain, including X, Motif, Windows, MS-DOS, UNIX curses, etc.

The disappointing part about the Gopher server was that the current manual systems for many of these things are still in place. It will take time, and many union negotiations, to reallocate the huge amount of bureaucratic resources used to maintain, publish, and distribute, things like job listings, movie schedules, etc.

## On-line Library Catalog

Cornell began to computerize the library catalog before the Gopher architecture was in place. Gopher is not an ideal protocol for searching library collections, unless one wishes to use the Library of Congress hierarchy for organizing the collection. Cornell has nearly six million volumes. The on-line catalog is for materials acquired after 1973, which is about half. The on-line system is currently run from an IBM mainframe. There are no manual card catalogs on the floor of main libraries; these have been replaced by Bear Access kiosks. Manual systems are maintained for the special collections and older volumes. The

[5] The postings to this forum are done anonymously and answered by trained people throughout the university. The answers have been collected in two volumes that sell well in the Campus Store.

[6] Hey this is Cornell.

[7] WWW is the World Wide Web, sometimes just called the Web.

 

Bear Access system spawns a 3270 terminal emulation package for library access. Unfortunately the user-interface is very command-line based.

Through the Gopher server one does have access to a large number of bibliography entries used to build reference lists for research. The on-line library also has access to several specialized databases, such as GEOREF (geology), medicine (Cornell Medical College), and legal/financial (Cornell Law School).

## *Just the Facts* Service

This is one of the most impressive and useful parts of the use of the Internet access provided all over campus. Through the Just the Facts Service students have a gateway into the university bureaucracy, which is maintained on the other IBM VM/CMS machine. Students can:

- display their current course schedule and registration
- check on previous semester grades (as soon as they have been verified by the registrar)
- check current bursar balance for tuition
- verify CornellCard balance (university-wide credit card which is the same physical card as ID, meal plan card, and library card, all using the same magnetic stripe)
- obtain financial aid status/information
- look up official cornell address (usually parents)

Some two way communication is also possible. Students can order transcripts and can change their permanent address.

The entire Just the Facts service is protected by use of a user ID. You cannot get access to anyone's personal information without their login ID (easily guessed) and password. The information on this service is a copy of the information on the IBM mainframes. I do not believe they are doing remote queries on the actual data at this time.

When alumni have access to the services this spring, the complete living alumni directory will also be available on line. This will make it easy to stay in contact with the school and vice versa.

## Who I Am

As described in the section on electronic mail, Cornell has implemented a central directory service that can be used to span the heterogeneous e-mail systems in use.

Along with e-mail information, the Who I Am service is used as a dynamic campus directory. In addition to e-mail forwarding, the service also provides fields for postal address, local residence, phone number, college, major, FAX number, department, job title, nick name, etc. You can omit any and all fields you want except for name and e-mail ID. This is the location for changing passwords as well. Cornell has not instituted any password aging

There is a rich query interface for the service. It allows for great scenarios such as "find the person named Jane in the Arts [and Sciences] school, class of '96, lives on College Ave." This lets you find people in your classes (or otherwise) that you only know by a few key points. These queries can be done via the **Ph** command (like our phone) from Eudora, or from the Gopher server (in the Campus and Directories section)

Interestingly, when I was at Cornell we added the ability to search for people's e-mail address to the CUINFO front end. It lasted about a week until people feared for loss of privacy. Even with the rich queries possible, the fact that people can omit information and are not required to have e-mail IDs at all has reduced the fear of big brother to near zero.

Along with this local directory service, through Bear Access one can connect to a large number of other servers that provide various directory services. Some are for other universities, specializations (physics for example), or public directory servers for people not affiliated with large organizations.

 

## Chat

Internet Relay Chat (IRC) is a communications protocol that lets you "talk" interactively via the keyboard with multiple users simultaneously over the Internet.

The Cornell community uses this extensively both for real work (faculty collaboration) and quite often for play. I saw a group of about 10 students Chatting with numerous people world wide until very late at night. Chat has clearly created a new form of cultural interaction, similar to e-mail.

There are several front ends to chat, some of which are very sophisticated and allow for multiple panes and other monitoring functions. From Bear Access a simple terminal emulation program is used.

## USENET News

USENET bulletin board access has been part of the university computing environment for a long time. Until the advent of easy to use readers like NewsWatcher (a Macintosh hosted MacTCP reader), access was available only via UNIX and usually only with the cryptic ReadNews, **rn**, program (like the one we use on our machines). Students now take part in the Internet community on a regular basis. From the data I could gather anecdotally, the computer center operators feel that each student subscribes to an average of about 10 newsgroups. The most popular groups are the `alt` groups related to music and movies.

## FTP and Telnet

Through the Bear Access front end Cornell provides an interface to FTP files around the various servers at Cornell. They are using a nice front end that helps out with Macintosh files conversions. The Macintosh files are an issue only because there are no large servers for general use that are Macintoshes, so users are forced to store files on UNIX servers. The front end is called Fetch and is another public domain program.

The telnet access is also for logging into any of the Cornell machines. This is a TCP/IP protocol that is used to support remote login on UNIX servers. Through Bear Access this is mostly just an easy way to launch the terminal emulator and execute the appropriate telnet line. Again, this just helps out in the heterogeneous environment.

## CU-SeeMe Conferencing

The most impressive demonstration I saw was a real-time video conferencing implementation over the Internet. That's right–multi-channel video communication for free! I participated in a conference with 5 people around the country (New York (2), Washington, DC, North Carolina, and California). The machine I was using was a Macintosh IIci, a low end machine.

Cornell has received some funding from NSF and the Dept. of Education (as part of a project called *Global Schoolhouse*) to work on getting video links between students in secondary school. They have implemented conferencing software that uses UDP (a derivative of TCP that uses fixed sized blocks without CRCs, as far as I understand). The video data is captured by a cheap $500 camera and board, which is then compressed and sent over the Internet line. If you do not have a camera you can still observe the conference without any special hardware.

In order to multicast the conference a UNIX based *reflector* runs on an server. This reflector enables any number of people to connect to the conference. There are currently about a dozen reflectors running around the country. All participants must connect to the same reflector. The current user-interface is still a little technoidy, in that the details of the reflector are in your face. As the reflectors become more stable this can change. Because of the nature of the software, the security aspects are lacking a little bit right now so it is possible for people to connect to a conference without obtaining permission. This will change very soon, of course.

Two key assumptions were made that make the system reasonable. First, there is no simultaneous voice and video. The system assumes the voice link will be done over standard phone lines. This makes it easy to avoid the problems associated with choppy voice and video or synchronization.

 

Second, the video is black and white with a variable frame rate. Basically the system will produce a frame rate as high as it can over the connection you have. The conference I participated in varied from 3 to 12 frames per second. Since the voice is detached from the video you really do not notice the slow frame rate. The images are typical 200x300 or they can be doubled if you have a higher frame rate. The compression just uses a simple frame difference algorithm so could obviously be improved. They wanted to be software only. In order to be good Internet citizens the frame rate is selectable from the user-interface. They have successfully obtained rates in the upper 20's with connections within Ithaca.

The system is another example of something that is taking off. They have the reflector and client software on public access FTP servers and the growth of reflectors and clients is very fast. As with all such technology, they have been fascinated watching the high school students use the system. They have already connected the US and Europe and have ongoing "exchange" programs.

A Windows client is being developed at the University of North Carolina with the help of IBM. They have already started asking for help with questions on Video for Windows.

 

## Suggestions

The following are some brief suggestions for things we might do to improve or look at in order to take advantage of the "Internet." I think Chicago can totally take over the university environment for productivity application use and as a client, if we do a few simple things. Making inroads with Windows NT is also possible, especially as schools prepare to buy more servers to implement their on-line services. The current crop of Macintoshes is getting pretty old and purchasing is on hold awaiting PowerPC machines. There is a big opportunity.

Some of these suggestions are already underway and this is just meant to underscore their importance in a university environment. Others are hard, controversial, or maybe not very strategic, but they are here to spur more thought and debate on this topic. Please see J Allard's memo for more detailed suggestions.

1. Implement TCP/IP network protocols for both Chicago and Windows NT. These need to be easy to install and configure, with the baseline being the ease of installation of MacTCP.
2. Implement a SLIP network protocol for Chicago (client) to support remote IP use.
3. Support someone who is willing to implement the current Internet "fad" server services including gopher, WAIS, WWW, Archie, list services, and name services natively on Windows NT. For some services it might be appropriate to implement them on Chicago.
4. Support someone who is willing to implement client access to News, Gopher, and WWW for Windows running on commodity Chicago machines.
5. Investigate an implementation of mailing lists that require no administrative intervention, along the lines of the ancient LISTSERV services.
6. Seed one or two universities with Windows NT and Chicago machines as test cases for the integration of TCP/IP services within a university environment (University of Washington, Cornell University?)
7. Investigate the use of network based conferencing as shown in the CU-SeeMe project at Cornell. This overlaps with the work being done in multi-media. Perhaps Microsoft should further the efforts at Cornell for a Windows client. We might also think of helping to get a reflector on Windows NT.
8. Remind ourselves that no matter how much we like the Chicago shell some people will need to replace the shell or have a way to disable it completely for machines that will be placed as unattended kiosks.
9. Investigate the file formats and protocols used in WWW for cross platform viewing and content indexing. This is clearly an emerging standard in the Internet world.
10. Understand that our content should be viewable on WWW servers, at least some portion, and what that will mean for our authoring tools. WWW content is increasing as much as **3% per week**. This includes understanding HTML (the derivative of SGML used for these rich documents).
11. Really understand why we cannot provide access to the Marvel content via Gopher, or more likely WWW.

Will Marvel allow Chat to take place across Internet servers. The IRC protocol is well established and could be useful for some people. It would also add a cheap international mystique to Marvel, even in the first release.

 