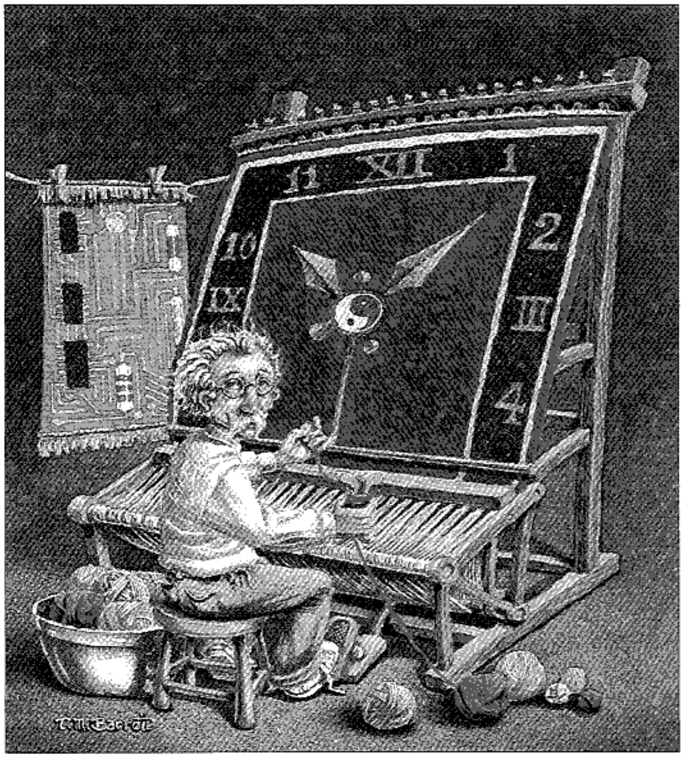
# Ask Mr. Protocol

*by Michael O'Brien*




TOM BARRETT

*"Never look back. Someone might be gainin' on you."*
– Satchel Paige

*"256 hosts is essentially infinite."* – Original ARPANET planners

*"Sometimes the Host Ready relay gets stuck. You have to thump it."*
– Steven Tepper

# Mr. Protocol Looks Back

**Q:** *How did Mr. Protocol get that way, anyway? And is it catching?*

**A:** To answer your second question first: Yes, occasionally, especially when it looks as if it might be profitable.

Your first question is much harder. Let's consider something easier instead, such as *Mary Poppins*. In the Disney movie of this remarkable tale, echoes of *Fantasia* are brought to bear as we listen and watch the bird woman on the steps of St. Paul's Cathedral ("Feed the birds, tuppence a bag."). The birds at the real St. Paul's are pigeons, not half so graceful as the abstract dove-like flocks in the movie, but St. Paul's is its own best advertisement and is beautifully rendered in the movie. In fact, this movie sequence is made memorable not by the flocks of birds, or the old woman on the steps below. It is St. Paul's that lends magnificence to the music.

Because England is not the United States, it is possible to climb to the top of the dome of St. Paul's, mostly because people have been doing it for hundreds of years. It is a mainstay of British society that anything that people have been doing for hundreds of years is something to be cherished and continued, including encouraging anything that's been going on for hundreds of years, so the whole thing sort of carries itself along, if you see what I mean.

The dome of St. Paul's is actually two domes, one inside the other. The inner dome carries the paintings that people see as they stand beneath, looking up from the floor of the church. This dome is shaped like a flattened beehive, so that people can actually see the paintings. If the paintings lay on the inside surface of the hemispherical dome we see from the street, most of them would be invisible, because they would be vertical.

The way to the cathedral spire lies between the two domes, inner and outer. At the last, one comes out to a dizzying prospect no more than 50 feet across: The "lantern," or square spire, sits within a round gallery, leaving no more than 18 inches of clearance between the corners of the spire and the railing around the gallery. No one minds. Beyond lies the London cityscape. It is this view that we must consider.

Greater London burned to the ground in the great fire of 1666. There are few buildings in the center of London that date from before this time, and no large ones. What we are left with is a skyline that consists of a mishmash of buildings constructed between 1667 and the present.

London consists of about two dozen parishes, each with its own parish church. All of these churches burned in the fire, along with the original St. Paul's. England's greatest architect, Christopher Wren, not only designed St. Paul's, but also many if not most of the parish churches that can be seen from the lantern of St. Paul's.

Knowing this, the thoughtful observer will see not a mishmash, but a palimpsest. One or two of these churches now lie in ruins due to the blitz of World War II, but the rest stand. The London of Christopher Wren still lies before the observer, in the dozens of steeples that rise above the common buildings below. Shakespeare, Wren, naval clerk Pepys and Marlowe, the tavern brawler, could name many if not most of the churches you see before you, and the Houses of Parliament and Westminster Abbey would also be familiar. Their London is still there to be seen and visited by anyone who will take the time, and who has the will and knowledge to see it.

The old does not often make way for the new. The new grows up in and around the old, shaped and influenced by it.

The Internet grew out of the ARPANET. Most folks would say the ARPANET made way for the Internet. The ARPANET was made up of Interface Message Processors (IMPs), Host/IMP interfaces, leased lines, Terminal Interface Processors (TIPs) and other large, heavy pieces of jargon-laden hardware that are now sought after by museums. No one dials in to TIPs any more to make terminal connections on the road. Space doesn't have to be made for a six-foot rack with a computer in it, just to connect four other computers (maximum!) to the Net. Yes, the ARPANET is gone all right, and good riddance, they say. Hmmm.

Sherman, set the Wayback Machine for 1977 in Santa Monica, CA. IMP #1 was still in service at the University of California at Los Angeles (UCLA) in nearby Westwood, with a big, glossy photo of Robbie the Robot plastered on it. Hollywood being nearby, Robbie had paid the IMP a visit and left a publicity still behind. The IMP was a favorite trophy to display, not least because of the huge steel ring bolted to the top: The first IMPs were designed to be craned into submarines if necessary.

IMP #7 was buried in the basement of The Rand Corp. (not yet The RAND Corp.) in Santa Monica, across the street from the beach. The machine room that contained it had previously held the JOHNNIAC computer. Now, the IMP shared space with a large Digital Equipment Corp. minicomputer running the first commercially licensed copy of the UNIX operating system.

## An IBM World

The ARPANET was developed in a computing world that, although dominated by IBM Corp., was far more pluralistic than today's. Every mainframe machine room contained an IBM 360 or 370, with few holdouts running Univac or Honeywell hardware. Rand was no exception. Host #1 on IMP #7 was an IBM 370 mainframe running MVT. At Bell Labs, scientists interested in what we'd call scientific supercomputing today were figuring out how to change the microcode in a 370. They did this successfully and reported in the academic paper describing the process that one of the more interesting results is that IBM did nothing to hinder them–IBM at that time being something of a natural force to be researched rather than a company to be negotiated with.

IBM was the center of the computing universe. Its operating systems were oriented to serving batch computing jobs and running large numbers of terminals doing transaction processing through terminal concentrators. Then, as now, its mainframe business was oriented more toward moving large amounts of data from hither to yon than supporting large amounts of computation on the data. This led to a highly optimized and highly specialized I/O system that so dominated its system architectures that support of what we now think of as general-purpose computing was baroque in the extreme. Word processing was so far from the IBM mainframe model that Prof. An Wang created a gigantic corporation to fill the gap with special-purpose engines...which crashed years later when the special-purpose engines were swept away by cheaper general-purpose engines that filled the same ecological niche.

So, with IBM firmly established in the center of the computer room, it came as no surprise that many machines on the ARPANET were IBM mainframes. UCLA supported a nearly unique Model 360/90 that was a supercomputer for its time.

The ARPANET supported batch computing. Early RFCs describe a number of interesting attempts at creating general-purpose batch queuing systems that were really quite ingenious, failing only because no one developed any general-purpose batch jobs to go with them. Batch jobs were intrinsically non-portable, and the "software tools" culture was years in the future. So, while IBM mainframes resided on the ARPANET, most of the traffic was originated by other hosts–hosts with a more interactive usage style.

From its earliest days, the ARPANET shone in its interactive applications. Telnet acted as a generalized dial-up service. `Finger` told you where people were and what they were doing. You could even find out the temperature in the machine room: There was a special socket number reserved for a time and temperature service. The only serious background processing that took place was mail.

The result was that most people who used the ARPANET seriously were not IBM users. The ARPANET users favored a wide variety of other systems. Carnegie Mellon University had its C.mmp and CM* multiprocessor systems. One of these, interestingly, had no mail system–its users favored another system for email–and sent out precisely one electronic mail message in its entire lifetime, announcing its imminent decommissioning. The University of California at Santa Barbara sported a Culler-Fried graphics system, which had two full-size keyboards, one above the other. It had been designed for mathematical modeling, and its two keyboards, plus shift, alt and several meta keys, supported a mathematical character set of immense size. Locals called it the "Culler-Fried Chicken" system.

If the ARPANET had a dominant species, it was the DEC PDP-10 running TENEX. TENEX was written at Bolt, Beranek and Newman (BBN), which, funnily enough, was the same company that built and deployed the ARPANET under contract to Advanced Research Projects Agency (ARPA). BBN wanted to build a general-purpose time-sharing system with a unified architecture, unlike the ratbag of special cases that made up the IBM architecture. Time-sharing was the most powerful way of using the ARPANET, because local workstations and local-area networks were still experimental toys at Xerox Palo Alto Research Center (Xerox Corp. had not yet fumbled the future). In order to provide the page-oriented virtual memory they desired, the BBN researchers built their own paging hardware and bolted it onto the side of their PDP-10.

TENEX was wildly successful. People who believe that Macintosh users are fanatics have never met a TENEX fan. In fact there are a large number of former TENEX users who would really prefer to still be using TENEX, and who believe that it was, is and always will be far superior in design and implementation to UNIX. They aren't deranged, either, even though they sound like it. Of course, show them "@" as an interactive prompt and they go all soggy on you.

The PDP-10 had one real peculiarity, though. It was a 36-bit, word-addressed machine. Now, 36 has factors of 3, 3, 2 and 2. This gives you the possibility of defining a byte as four bits, six bits, nine bits, 12 bits, 18 bits or even seven bits or eight bits, if you don't mind a few odd bits left over. Lisp being the favored programming language of the day, a rich supply of tag bits was fairly universally felt to be a Good Thing, so byte lengths varied considerably in practice. SAIL, the Stanford Artificial Intelligence Laboratory, used to regularly hack on the microcode in its PDP-10's KI-10 processor, leading to the (eventually infamous) message of the day, which appeared with astonishing regularity, "The XYZ instruction has now been implemented. Please update your programs." Many of these instructions found their way back into DEC to become the basis of the KL-10 processor, the most powerful in the PDP-10 line.

So popular were the PDP-10 architecture and the TENEX operating system that Xerox PARC researchers, looking for a good central machine to hang their local net and workstations off of, tried to buy one. Xerox being itself at this time a commercial manufacturer of computers, having bought the Scientific Data Systems line of Sigma processors, refused to buy a competitor's top-of-the-line engine. (Years later, Apple showed no such signs of incipient senility when it bought a Cray to do circuit design. Apple's senility lies in other areas, such as marketing. ["What do you mean, people want to buy them? What on earth for? Tell them to make an appointment."]) Xerox researchers responded by beginning an initiative in microprogramming, designing and building a general-purpose microprogrammable mainframe engine that could be coded to emulate a variety of mainframes. The only emulation ever written was for the PDP-10. Once the emulation was working, TENEX was booted on the PARC-MAXC mainframe, and that was the quiet end of Xerox's research into mainframe emulation.

## From the Sublime to the Ridiculous

To be sure, there were other systems out there. In large measure, that was the whole point of the ARPANET: to make unique resources available to researchers at other institutions. There were some doozies. The National Center for Atmospheric Research mounted the Datacomputer, which consisted of a central processor and a ridiculous amount of disk storage, all fronting for an Ampex Terabit Memory System. This set of about 10 special-purpose tape drives, recording onto two-inch videotape, represented by far the largest storage capacity available anywhere. Of course, a terabit is only on the order of 100 gigabytes, plus check bits, so anyone with a Platinum card can go down to Fry's and build their own terabit memory system today. It'll hold most of the games on sale at Fry's, too. At the time, though, the space made available by the Datacomputer was so immense that people seriously considered doing their daily backups to it over the ARPANET. Given the fact that the ARPANET was glued together with 56-Kb/s leased lines, when someone in Washington, D.C., fell into the daily habit of dumping several hundred megabytes of weather data to the Datacomputer in the middle of the night, the ARPANET became more useful during the day than during the night. This is the only period during the entire history of the Internet when this has been so.

*Several different institutions wrote code to put UNIX on the ARPANET, and most of the efforts were underwhelming.*

When the Datacomputer was eventually decommissioned, the public at large was given two entire weeks to get all their data off it over the Net and do something else with it. Those who had not become dependent on the Datacomputer breathed a self-congratulatory sigh of relief.

Another one-of-a-kinder was the ARPANET Terminal System, ANTS. This was developed at the University of Illinois at Urbana (aka "Shampoo-Banana"), and consisted of a large PDP-11 system running special-purpose software. It was a terminal concentrator for the ARPANET, and it never worked right. Its most successful feature was that someone had carefully created a trail of large ant cutouts along the top facing of the machine.

ANTS shared machine time with a few researchers who were playing with a system they'd gotten for $150 from Bell Labs, called UNIX. The idea was to see if this system might make a good candidate for being put on the ARPANET. It seemed a logical match, because it supported interactive use better than most other systems, had a number of useful tools in its own right, was cheaper than dirt compared with mainframe prices and came with full source code.

The answer turned out to be, "Well, sort of, if you try hard." Several different institutions wrote code to put UNIX on the ARPANET, and most of the efforts were underwhelming. One natural extension was to put host names out in the file system name space. On several of these early attempts, every host on the ARPANET was represented by a hostname in the `/dev` directory. This didn't work out too well. OK, you've done an `open()`, so now what? The semantics of the UNIX file system turned out to be a poor match to the requirements of networking semantics.

Someone finally wrote an ARPANET interface daemon that worked right. Because it had been foreseen that ARPANET hosts would be flaky, and indeed almost all ARPANET hosts were flaky, the ARPANET was designed to be connected via a group of special-purpose computers called Interface Message Processors (IMPs). These IMPs, in turn, were connected via 56-Kb/s leased lines. IMPs did not go down, or if they did, were designed to be rebooted remotely over the Net. Physical human intervention, outside of the monthly periodic maintenance that just about all computers of the era suffered, was almost never required. Because the weird byte sizes of the PDP-10 systems had shown people that different hardware architectures stored their bits differently, the connection between a host and its IMP was a bit-serial connection. Given the speeds the ARPANET ran at, this was not a performance problem. Hosts also had control over a one-bit relay, called the Host Ready relay, which they would turn on or off depending on the host's readiness to accept connection requests over the Net. It was part of the protocol of bringing a host up on the Net after a reboot that this line be "waggled" up and down three times, according to a specified timing interval, to alert the IMP to the reboot. And, for the sake of reliability, it was part of the hardware specification that this Host Ready relay be a mechanical reed relay, not a piece of logic.

Big mistake.

At Rand, the Host Ready relay had a tendency to become stuck. This necessitated taking the PDP-11 down, shutting it off, pulling the interface card out of the Unibus cage and thwacking the relay just so with the middle finger released off the thumb. If your fingernail hurt, you'd done it right. Plug the card back in, reboot the system, and when the Network Control Protocol (NCP) daemon started, look at the IMP to see if the Host Ready light on the front panel blinked on and off three times.

Meanwhile, upstairs, in the showcase computer center with the big IBM machine, the IBM's ARPANET interface was an entire box, a cube two-and-a-half-feet on a side, covered in blinking lights. The sad thing was, IBM's NCP was one of the best NCPs ever written. Stable, correct and with outstanding hardware to match (that cube, one of Advanced Computer Communication's earliest products, was a sterling piece of engineering), the IBM machine was one of the most capable ARPANET hosts around, but it was so hard to get at the ARPANET software that few people ever found a use for it, except as a remote printer. Everyone stayed on the flaky PDP-11 UNIX system, because Rand had converted a whole bunch of ARPA's finest dollars into software that is still being used today, under ARPA's Office of the Future initiative. This was the period when the Rand editor and the MH mail system were first written.

Along with a working NCP daemon, UNIX received a working Telnet, FTP and mail suite. Other useful packages came along too, which Mr. Protocol has mentioned before, such as SUPDUP, a Massachusetts Institute of Technology invention that was Termcap and Terminfo done right. Some stayed the course, some disappeared. But eventually, everything was forced to change.

The scientific workstation had been invented, and it cried out for a network. Several competing local-area networks existed at this time, and which one you ran depended on which workstations you were using. If you went the Xerox route, you strung 3-Mb/s Ethernet coaxial cable and learned how to deal with PUPnet software. If you ran Symbolics Lisp machines, you strung 10-Mb/s Ethernet in big fat coaxial cable and ran Chaosnet protocols.

And none of these workstations could talk to the ARPANET, because the ARPANET was based on a mainframe view of the world, where each IMP could only be hooked to four hosts (or, in rare cases, eight), at a cost of about $150,000 per year for each host. And the NCP protocols didn't allow for any such thing as routing. You connected the host directly to the IMP and the IMP did all the routing.

***It was the LSI-11 routers that superseded the IMPs as Internet routers that made it possible to connect local networks to the big wide world.***

All sorts of cockamamie schemes having to do with virtual host addresses were dreamed up to allow these new campus-wide LANs to be connected to the ARPANET, but wiser heads prevailed. This was not time for a kludge. This was a time for transition to a whole new set of protocols.

Not everyone was willing to see things this way. For one thing, many legacy machines lacked the dollar support necessary to develop an implementation of the new network protocols for them. For another, a transition to a new protocol suite required the establishment of a "flag day" on which the IMPs would be commanded, at midnight, to quit routing NCP traffic, and to route only the new protocols. They sounded funny. For one thing, there were two of them, somehow related. One was called IP and the other was called TCP. And IP wasn't even reliable. Gaaah. What a terrible idea this was. Political pressure forced the "flag day" to be put off several times, for more than a year.

## A Victim of Its Own Success

OK, so here's this brave new Internet. There's only one Net, and it's Net #10. What happened to Nets 1 through 9? Good question. Mr. Protocol is glad you asked. They were assigned to campus nets that had done work in developing TCP/IP, and needed IP network addresses before TCP/IP was deployed on the ARPANET at large.

Now, the ARPANET had more troubles than just this. It was a victim of its own success. Real-world, practical routing theory was in its infancy at this point. All the IMPs ran the same routing protocol, and there were only about two other IMP-based nets in the world that could be used to test new routing algorithms for the IMPs. The original protocol broke down badly under conditions of load, causing routing thrashing on the ARPANET's cross-country backbone links. This made the Net almost unusable during the day, far worse than the Internet congestion seen today. In fact, this happened several times, for months at a time, first because the routing protocols needed improvement, and later because the links were honestly saturated, and the phone company lacked the high-speed long-haul links necessary to replace them. Lead times of nine months meant nine months of an ARPANET that was practically useless for interactive applications during the day.

What's interesting about the Internet is not the first Net, which was the ARPANET. What's interesting is the second Net, which was MILNET. This consisted of the military bases, as opposed to the not-for-profit research and academic centers. Many of these had been doing classified networking over the ARPANET for years, using link-level encryption. Now the plan was to partition the ARPANET into two nets, the original ARPANET and the new MILNET, and allow traffic to flow from one to the other only via electronic mail, through "mail bridges." Nice try.

Strong political factors, of the "Your head is mine" variety, forced the mail bridges to become the first Internet routers. They were overloaded and didn't do too well. Eventually, more were commissioned, and finally the Internet was opened up to the campus networks. Network numbers other than 10 were finally seen, but only when new routers were installed, connected via leased lines. IMPs still couldn't be convinced that networks other than ARPANET and MILNET existed, and they still only connected mainframes, not entire networks.

The result was that dinky little LSI-11 microcomputers became the world's second routers. And desktop workstations could finally Telnet off campus.

The Internet as we know it today is not purely a product of the TCP/IP protocol suite–that made it possible to express the notion of connectivity between networks in a sensible fashion. But it was the LSI-11 routers that superseded the IMPs as Internet routers that made it possible to connect local networks to the big wide world. The Internet began to grow into the entity we know today.

Commercial activity was verboten at first, of course, because the U.S. government was still footing the bills. It wasn't until CSNET, the Computer Science Research Network, was established as the first commercial Internet service provider, that people could just pay money to go on the Net. They still had to show some sort of research orientation to satisfy CSNET's charter, but this was sometimes rather loosely construed, as in the case of Omnibus Computer Graphics, a now-defunct computer special effects house that used CSNET not to connect to the Internet at large, but to act as a WAN connecting its own far-flung operations. If virtual private nets had existed in 1984, Omnibus would have done that instead.

The major players still hung back, because although CSNET was profitable, it was not wildly so. It took a while for Sprint, MCI and the rest to become large-scale Internet service providers in their own right. UUNET was probably the first one to go fully commercial in a big way.

And, gradually, as new services such as WAIS and then the Web came along, the character of the Net began to change. No longer was there traffic bringing unique resources to far-flung researchers. Now it was a matter of uniform resources being available everywhere, and the information itself being unique.

It is a principle of ecology that the smaller the number of species making up an ecology, the less resilient and healthy the ecology is. The ARPANET was extremely resilient, all the way through to the early days of the Internet, because although a large amount of legacy hardware was left behind forever on IP "flag day," there was still a lot left. People were running

TENEX, UNIX, TOPS-10, MULTICS, MVS, MVT, VM, DEC hardware, IBM hardware, Univac hardware, Honeywell hardware, homebrew hardware, whatever could be coerced into running an NCP or a TCP/IP stack. Even versions of UNIX varied among large research centers. Rand never even bothered to run UNIX Version 7. It stayed with its own highly hacked UNIX Version 6 because most of the features of Version 7 had either been duplicated by parallel evolution, or had been retrofitted on a case-by-case basis. This doesn't even mention the system calls introduced by Rand (it had named pipes in 1976). Mr. Protocol spent a happy couple of years ripping extraneous code out of the Rand kernel until it quit crashing. Others, at other institutions, can tell similar stories. Rand had one of the first VAXen on the Internet, even before native VAX network code was available, by writing a daemon that took network requests and shipped them via a 9,600-baud terminal line to a PDP-11 running a stand-alone NCP. Almost all of Rand's net traffic moved over that line...people still weren't using the IBM mainframe.

## UNIX – The Standard Internet OS

As the Internet grew, however, things changed. TENEX fell by the wayside, a victim of DEC's decision to go with the VAX at the expense of PDP-10 development. The Jupiter project, which was to develop a killer PDP-10, was disbanded and team members reassigned to the VAX.

VAX architecture was more mainstream than the PDP-10, being an extension of the PDP-11 to a 32-bit architecture with safe, sound, stable, mainstream 8-bit bytes. VMS, the VAX operating system, was confidently seen by DEC as a UNIX-killer. VAX hardware and VMS had been codesigned, so how could UNIX compete with VMS on a VAX?

UNIX spread like a virus. ARPA didn't help, or it did help, depending on your view. Refusing to pour any more millions into competing operating systems providing similar infrastructure, ARPA held a quiet meeting among its current crop of principal investigators and asked them to decide on a common operating system software base, which ARPA would support as a network operating system to be used across ARPA projects. UNIX was that choice, and the University of California at Berkeley was chosen as the institution to act as the architects and implementors of UNIX as the standard Internet operating system.

As a result, UNIX wiped the floor with just about everything else, at least on the Internet. Microcomputers and PCs might run CP/M or DOS or Windows, but these were laughable as network systems. Most researchers thought they were laughable as any kind of system. Nonresearchers didn't care, and got rich. Caveat emptor.

As a result, while the Internet continued to grow and apparently became robustly healthy and self-supporting, there was basically a single operating system on the Net: UNIX. Versions began to coalesce as it became commercially supported and source became scarcer. What cost a university $150 for UNIX Version 5 in 1975 came to cost a commercial company $70,000 for source code, if they could get it at all. In a little-noted contretemps, Sun Microsystems Inc., which when it was founded promised to fold developments back into the Berkeley release, reached the point within only about three years where it refused to sell source code to BBN at any price. Only the intervention of ARPA, which promised that without source, Suns were useless as ARPA research platforms, caused Sun to release source code.

Researchers today are profoundly depressed by the advent of greaseball operating systems emanating from Redmond, WA, and sullying the network scene.

People with a greater historical perspective know better. Windows 95, Windows 98 and Windows NT are not going to take over the Internet, no matter how badly Mr. Gates needs the entire output of Turkmenistan to carpet his front hall. The advent of new operating systems into the networking world is an unmitigated Good Thing, because it keeps the Net honest. The Netscape Navigator/Microsoft Explorer wars merely go to point out the disadvantages to all of ignoring standards in favor of proprietary software. Aside from the corporate sites of the two combatants, any Web designer worth her salt knows that a site designed to work with both browsers is far, far preferable to a "sabotage site" that breaks one or the other. In fact, it encourages a third party to do just what the original Mozillans did: build a better browser.

New operating systems, no matter where they come from, or how bad they are, are not going to kill the Internet. "Rough consensus and working code" has been the Net's watchword from the beginning, and it still works. The only appreciable strain that the Internet Engineering Task Force is showing, is scaling up its operations to deal with all the new initiatives in multimedia, bandwidth management and satellite service.

In fact, Mr. Protocol, at the conclusion of this, his hundredth column for *SunExpert*, feels confident that diversity, not uniformity, is going to mark the future of the Internet, as embedded systems and wireless connectivity combine to form a much richer, more multilayered network environment than we enjoy today.

Now, if only he could Telnet to the drip system that waters his front yard.

---

***Mike O'Brien** has been noodling around the UNIX world for far too long a time. He knows he started out with UNIX Research Version 5 (not System V, he hastens to point out), but forgets the year. He thinks it was around 1975 or so.*

*He founded and ran the first nationwide UNIX Users Group Software Distribution Center. He worked at Rand during the glory days of the Rand editor and the MH mail system, helped build CSNET (first at Rand and later at BBN Labs Inc.) and is now working at an aerospace research corporation.*

***Mr. Protocol** refuses to divulge his qualifications and may, in fact, have none whatsoever. His email address is* `amp@cpg.com`.