

Router man

The creator of the multiprotocol router reflects on the development of the device that fueled the growth of networking.

By [John Dix](#) , Network World , 03/27/2006

Sponsored by:

 <IMG

src="http://ad.doubleclick.net/ad/idg.us.nwf.lanwan/printer;pos=imu;sz=336x280;tile=4;ord=1208885652?" border=0 height="280" width="336">

The creator of the multiprotocol [router](#) reflects on the development of the device that fueled the growth of networking.

William (Bill) Yeager is 66 and still gets peeved when someone trots out the Silicon Valley fable about how the founders of [Cisco](#) invented the router. He was the guy at Stanford University that made it happen. The history of Network World roughly parallels the commercialization of routing, so we tracked Yeager down for a glimpse into the scene back then.

You're credited with developing the first router while you were a staff researcher at Stanford. Tell us the tale.

This project started for me in January of 1980, when essentially the boss said, 'You're our networking guy. Go do something to connect the computer science department, medical center and department of electrical engineering.

What kind of gear did you need to connect?

We had mainframes, of course, DEC10 Systems, a number of Xerox PARC Lisp machines, Altos file servers and printers, and over the next year or so added DEC VAXs, Texas Instruments' Explorers and Symbolic systems. All of these things had to be hooked together, because we were spread across buildings on campus, and people were tired of carrying tapes around.

I thought about this for a bit, and I said, well really what you need is an [operating system](#). So while the cables were being pulled and tested, I developed a network operating system [NOS] and routing code [to run on a] DEC PDP11/05. But the Alan Snyder Portable C compiler generated too much code. So I had to go into the compiler and improve the code generators. And that wasn't even good enough. So then I had to write an optimizer for PDP11/05 assembler so I could reduce the code by about 30%. This was major engineering, because you had your hands

In this package

[20 people who changed the industry](#)

[20 network-changing products in 20 years](#)

[20 most important stories of the past 20 years](#)

[Then and now: \\$20 buys you . . .](#)

:

[A special nod of thanks to Noorda, Metcalfe and Gore](#)

[Router man](#)

[With new pay, new responsibilities](#)

[Readers weigh in on industry](#)

[Giga flops](#)

[Five futurists to describe the what is to come in networking](#)

[Are you a net know-it-all?](#)

[Advice for the next generation of IT managers](#)

[A brief timeline of network industry milestones](#)

into everything. It's important to remember the PDP11/05 only had 56KB of user memory, and was diskless.



The struggle was always a balance between how many input buffers you could have. You really had to squeeze things, because there was no disk and if you ran out of memory for input buffers you were dead in the water. So you had all of these constraints, which actually had a lot to do with how good it ended up being, because I had to do a lot of work to both assure the memory allocation algorithms would never run out of memory, as well as get things scheduled right. I spent an entire summer making sure the NOS scheduling and packet-switching algorithms were optimal.

All in all we had the basic systems put together and working in about three months, and at six months the first router was in place in Pine Hall in a telephone closet. Pine Hall was midway between the medical center and the department of computer science. It was about a 2,000 feet cable run on either side of the router.

What protocols did the box support?

Initially, the code routed Parc Universal Packet (PUP) for the Xerox PARC systems and mainframes. Late in '81 my boss said, 'IP is coming down the pipe. Figure out what you can do with it.' So I put a little IP router in, and I didn't have to worry about things like [ARP](#) [Address Resolution Protocol], because it was 3 megabit Ethernet, so your IP address was just 2 bytes, one for the network and the other for the host, and the host byte was also the [media access control] address. But we were ready by '82, when the computer science department started dropping IP in all of these VAX750s, and by 1983 the routers supported XNS - which is Xerox Network Services - CHAOSnet for the TI Explorer and Symbolics Lisp machines, and IP. And it was just about then that Stanford University started to make the big transition to 10Mbps Ethernet.

Is that roughly when you made the shift from the PDP11/05-based router to the device based on the 68000 board developed by Andy Bechtolsheim (who later went on to found [Sun](#))?

Andy was a master's student, and that collision was fortuitous. We had heard about his board, and we talked to him and he said we could have it. We plugged that sucker into a multibus backplane, plugged in some [3Com](#) Ethernet boards and then rattled off a few copies, and I sat down and did a full transition of the code. One of the key aspects of these routers I put together is they really could route. I had a tremendous amount of instrumentation in there. I worked very, very hard to get that right and they could really pass stuff through as fast as the hardware could move. My limitation was the bus speed, that was it.



The original Bechtolsheim boards had 256KB of RAM, and that was huge at the time. To me it seemed like paradise.

Is that when the school's network started to take off?

People were skeptical at first, but by 1983 it was clear this was the way to go. Initially, just technical people were hooking up, but then the rest of the campus got wind of it and it was made official and the thing started growing like hell. I completed the serious development around '85.

What happened in the interim? A lot of tweaking and refinement?

It's endless, right? New features, functions. I did a lot of Lisp work where objects were used, and I adapted that approach in C, so a router was a class, and a specific protocol, say IP, was an instance of that class, and the NOS was multitasking. When you added another router, then you ended up putting in an instance as another task or thread. In the network I/O drivers you would look for the link-level type in the packet to determine the protocol, and everything goes into nice queues under these router threads and it all works. That's why Cisco did so well in this, because you could add more and more stuff to the [operating system], no problem. Just add another task.

Speaking of Cisco, when did they enter the picture?

In the spring of 1985 Len Bosack [who was in charge of the computer science department's computer facilities and later went on to co-found Cisco] and another guy knocked on my office door and asked if they could have access to sources for the router code. I said, what do you want to do? They said, we want to improve it, add more features. I said, well that would be great, because I have other research tasks to do, and I gave them the password and away they went. I had no idea Cisco had been founded in '84. I'd never heard of it.

So your understanding was they wanted the code for the betterment of the school network?

Right. So we had weekly meetings and they were indeed working on the sources. The decision had been made to go with pure IP routers, so they took out XNS, CHAOSnet and PUP. And ultimately when they got it going about a year later their version of my code became the official Stanford routers. Things were working well and that was my only concern. We had connectivity.



PHOTOS: STEVE SKOLL

So I guess sometime in '86 I found out about Cisco. We all found out about Cisco and what Len was up to. And yeah, they were developing that code on Stanford time for Cisco. But this was not exactly bad, because other things had happened like that at Stanford before. But Stanford was deciding it was time to put its foot down. 'Guys, you develop something on Stanford's campus, we want to profit from it,' right?

Who was saying this?

This was just kind of the general tenor. So I was called into Stanford Legal and the lawyer told me to bring my sources on paper. Since [Len's partner] was in the Double E department he had the Double E sources. And I sat down, and the lawyer said, 'Will you do a comparison.' And I said, well let's start with the operating system. That's sort of the heart and soul of this. And it was identical except for changing variables names. I said, can you see this? She said, 'I'm a lawyer and I can see this is identical.'

Let's look at other things. Let's look at this network data logblock (a C structure). Well it's been broken into two pieces, big deal. Any time someone gets a chance to go over code again they refine it. It was refined, clearly, but absolutely the same stuff. Derivative. They changed and added a their new routing protocol, no big deal. If you knew networking you could do it. I only did what I had to do, because I was driven by my boss and he was driven by the department's needs. And when I stopped I stopped.

Well, then Stanford really put its foot down and Len [and his partners, including Cisco co-founder Sandy Lerner] left the university to focus on Cisco.

Did Cisco ever give you any credit, other than the \$100,000 in royalties?

The way royalties work, a third goes to the school, a third goes to the department and a third goes to the inventor. I gave my third back to my department because essentially all of this stuff is born out of a great research environment.

But Cisco has always had trouble giving me credit. They had a Web page that I was very irked by. 'Sandy Lerner and Len Bosack were in love and they had to go out and invent routers so they could talk across campus.' What a joke. And I'm like one of these bulldogs, you know, I get a hold of these guys' pant's leg and I won't let go of it.

I'm sort of a persona non grata down there at Cisco. But it was fun. I was very passionate about this stuff. I'm always passionate about what I do. And I learned a lot about how corporations work and these guys were great capitalists and obviously they turned out with a great company.

So you left Stanford after 20 years and went to Sun, right?

I left Stanford because it was getting more difficult to get grant money, so I did a bunch of consulting at Sun to make some extra money. Mostly dealing with [IMAP](#) e-mail stuff because there was a very interesting project at Sun called SPARCstation Voyager: a laptop with a fast matrix display, nice little footprint, running Solaris 2.4. Great system. One of the Voyager's special features was that it ran in disconnected mode. You could disconnect it from the network, and it would continued to function. My job was to create an IMAP server and client that worked when the client disconnected. This was tough because, at that time, IMAP2bis did not support disconnected e-mail, and I needed to modify the protocol to do this as well as support low bandwidth (IMAP can be very chatty). After one of the guys I was working with quit, his boss asked me to come save the e-mail part of the project. And I thought, I'm 53. I've been at universities too long. So I said sure.

How would you compare the academic to the commercial world?

I always ran into walls at Sun, company politics, and that never worked out too well. When I was at Stanford there was a rule: The best engineering wins. Simple, straightforward. If your engineering is better than the other guy's, yours got the blue ribbon. Well at Sun, and at companies in general, it's different. It's the politically correct software that gets productized. There are charters and vice presidents and presidents and all of that stuff, and I would find myself embroiled in these battles with people 10 levels above me [laughs], but I just kept battling. I didn't care, because I liked doing good engineering.

So I brought in the IMAP technology, and by '96 IMAP servers I had written were everywhere at Sun. And once that was in place they decided we should do something called mission-critical mail. So I invented something called Sun Internet Mail Servers [SIMS], which is a whole different type of server. We ended up getting hundreds of thousands of in-boxes on a single server.

The four patents I have, out of the 40 I filed, are on SIMS. The rest are really in peer-to-peer, which I did a lot with as I moved through Sun, ending up as the CTO of JXTA, Sun's open source peer-to-peer project.

What was JXTA all about?

The charter was to create an open source project for the creation of peer-to-peer protocols that would yield a virtual layer on top of the [TCP/IP](#) stack. That would return end-to-end connectivity to the Internet by making the traversal of [NATs](#) and [firewalls](#) transparent, and provide host endpoints with globally unique identifiers. Another goal was to work toward peer-to-peer protocol standards. I personally pushed this forward in the IETF and that resulted in an IRTF Peer-to-Peer Research Group that I still co-chair.

Open source was new territory for Sun, and the Project JXTA group were the pioneers. We had a very tight organization and a charter to do disruptive technology, so it was a grand experiment. An engineer was two degrees of separation from the vice president and they were always available for discussions. Amazing! We received an introduction to how to do an open source project from CollabNet and they hosted Project JXTA. Initially, most of the engineering was done by Sun but then the JXTA community began to grow exponentially and great contributions came from non-Sun members.

But then a lot of things happened to the organization. JXTA was put under the product side of things, which kind of gave me the shivers. I mean, you get into product stuff, and you're in a box. You can't get out. I always managed, but just because I was irritating enough for my vice presidents that they would say, 'Go do something else.'

So in 1998 I'm talking to my vice president, and he says go do what you want. And I said I'm doing [wireless](#). It's the next big thing. He says OK, if you believe it, go do it.

I wrote something called the iPlanet Wireless Server, which sat between IMAP e-mail on the back end, and on the other side you could go to [Wireless Application Protocol] servers or any kind of wireless device. It was presentation language stuff so, depending on the device, you put out screens for phones or whatever. It was quite cool. It ended up being probably one of their only money-making wireless projects.

Based on some of the projects I know you've been involved in, a common thread seems to be handheld devices. Do you see particular promise there?

Over the years I've developed a real interest in mobile devices, which was one of my reasons to go to Sun in the first place, to do this mobile laptop, which they ultimately end-of-lifed (in error, but they did it anyway). So I saw the power of these devices, and I saw the power of integrating these devices. You could see wireless moving in, see all of this happening. It was very clear.

I felt we ought to do something to get some decent user interfaces on these devices. That's going to be a big next step. I don't think everybody in the world's going to have a computer, and it's stupid to ask everybody to learn to type. If you can use a mobile phone there are ways around this, and that's part of what I'm working on if I can get this new company going.

What's the focus of the new company?

It's called Peerouette, and it's a new twist on peer-to-peer. I've created what's called a deterministic peer-to-peer network. That is, the peers are never down, because the peers are not your devices. The peers are in the network and hosted by ISPs. Your device just authenticates strongly with public key and gets in there. And all your content lives in the network and is shareable 24 by 7.

You drop your mobile phone in the toilet, it's done, but it's all backed up. Automatically. My colleague says 'Bill, go to this URL.' I do. An image of his mobile phone appears on my laptop. He says 'press the menu key.' I do. I'm looking at his menu. He says 'take a picture.' I do. A picture of him appears. We've really gotten into these operating systems, how they work. We can totally control mobile phones from other devices. This is great for mobile phone people doing IT. All under very strong encryption.

So it's a lot about that and a lot about giving computing back to the people. I'm very big on the garage rock band having a way to sell their stuff. So in my world, you create your community out there in what we call the Peerouette Network, you take your MP3 files, push them out there, we give you billing, give you advertising, and you can sell them for whatever price you want. We'll take maybe 10%, something like that. What we're really doing is giving the user, the wireless ISP and the content provider a fair share of all the content revenue.

That's kind of what I'm up to, if we can fund it and get it going. We are very close. Cross your fingers. The Internet will surely be a better place if we succeed.

Sounds great. Good luck with that. In closing, let's change the subject. I understand you have a wine cave. What's that?

In French a wine cellar is called a cave because originally all the wine bottles were literally stored in caves. So we have a wine cellar and keep a reasonable supply . . . about 500 bottles going back to the '80s. Always fun when you have friends over. Go down to the cave and bring out a bottle or two.

What's your favorite wine?

Pinot Noir. Without any doubt. They are the most subtly complex wines. They have a spectrum of flavors that show a taste of the earth.

William (Bill) Yeager

Where born, year:	San Francisco, 1940
Education:	<ul style="list-style-type: none"> • Bachelor of arts in mathematics, University of California at Berkeley, 1964 • Master of arts in mathematics, California
Work history:	1971-1975 — Systems programmer, NASA AMES 1975-1994 — Research staff in Stanford University's Knowledge Systems Laboratory 1994-2004 — Sun Present — Final stages of completing financing for start-up Peerouette
Interests/hobbies:	<ul style="list-style-type: none"> • Fluent in French, learning Mandarin • Tennis • Collecting wine • Travel to Europe, Asia
Tidbit few people would know:	After high school I attended the U.S. Merchant Marine Academy in Kings Point, N.Y. I left the academy after 18 months, six of which were spent seeing places like Pago Pago, Samoa, Australia, Tasmania, Japan, China and the Philippines. I found a calculus book published in 1895 in the library of the USS Golden Bear of Pacific Far East Lines. I read it, liked the math and ended up at UC Berkeley, where the mascot is the Golden Bear.