Development of TELNET’s Negotiated Options

Bernard Cosell and David Walden

Because the Internet’s TELNET protocol is primarily documented by a specification, much of the detail about who authored various bits of TELNET is undocumented. We are writing this description of our involvement in creating one aspect of TELNET before our memories become so dim we can’t remember our involvement at all.

1. Historical Context

TELNET was for years a key Internet protocol [9]. It is still in wide use and seems likely to survive indefinitely.

The TELNET protocol was developed as part of the original development of the ARPANET, the precursor of the Internet [10]. In an era when computer networks were typically either one-off special developments for one application or consisted of computers from only one computer vendor using proprietary protocols, the ARPANET was designed to be a common user network. As a common user network, ARPANET supported use by users of many kinds of applications on many kinds of computers with a variety of different operating systems — from any application on any computer (with its operating system) to any application on any computer (with its operating system) or to any user on any computer (with its operating system).

This was relatively early in the time sharing era (well before personal computers became prevalent) when users typically sat at terminals sharing access to a time-shared computer over a 110 baud (about 10 characters per second) telephone line. The plan for the ARPANET was that users at terminals on computers at their own locations would access time-sharing systems elsewhere on the network, thus permitting less expensive and faster access to time-shared computers regardless of their geographic location. Thus, the ARPANET faced the problem of having computers and operating systems expecting to be connecting with a specific type of terminal that were, in fact, communicating with another type of terminal. For instance, a line-at-a-time terminal using the EBSIDIC character set and meant to be connected to an IBM computer needed to be able to communicate with a PDP-10 computer expecting a character-at-a-time terminal using the ASCII character set, and vice versa.

Thus, the concept of the Network Virtual Terminal (NVT) was developed. An NVT is a terminal with a hypothetical set of characteristics [9, 4]. Every kind of computer and operating system was to be able to translate between its native terminal type and the network NVT, enabling any computer and operating system to communicate at some basic level with every kind of terminal.

2. Our Situation

In 1972, we were both involved with the support and on-going development of the ARPANET Terminal Interface Message Processor (TIP). The TIP was an ARPANET IMP (the packet switches that made up the ARPANET communications backbone [6]) that also had the capability to communicate with a number of local or dial-up terminals. TIPs were scattered throughout the geographical expanse of the ARPANET providing access to either nearby or remote time sharing systems. Thus, the TIP included an implementation of the TELNET protocol and NVT to allow the terminals connected to it (mostly simple, character-at-a-time, ASCII devices) to communicate with a variety of time sharing systems around the networks, independent of whatever type of physical terminals those time sharing systems supported [8].

Because many people were connected by TIPs to many different kinds of time sharing systems, we found ourselves to be highly affected by the then on-going refinement of the specification of TELNET and the definition of the NVT.

We had a breakthrough one day, which we called TELNET “negotiated options,” as described in the next section.

3. Our Solution

The details of TELNET negotiated options were worked out on the plane while Cosell and Walden were flying out to UCLA for a meeting with other ARPANET developers. The problem we were chatting about was how to make the TELNET protocol uncooperatively extensible — that is, how could we allow the protocol to change and evolve without having to make dozens of implementations change every time some new “feature” was added. In other words, the issues were: a) how to make the protocol extensible, and b) how to make it work asynchronously and symmetrically (no master or slave).

Cosell suggested and developed the basic ideas as we talked, with Walden asking about how to deal with various cases and nodding agreement as he began to understand what Cosell was saying.

As Cosell remembers the conversation, “The obvious thing to do was a negotiation, but there were problems with what happened if both sides started
negotiating at once and their requests (and refusals/acceptances) crossed in the pipe. And also so that the right thing would happen for an implementation that didn’t know from nothing about the option that appeared — so one could implement a computer’s protocol handler to understand what it understood and safely ignore everything else. I remember waving my hands (and perhaps scribbling a few partial examples on a cocktail napkin) and assuring you that the protocol couldn’t possibly ‘oscillate’ — that the negotiation would always quietly terminate and not go on and on, no matter how confused the two ends got” — that is, avoiding a negotiation like the following (in which the computer at which the user is sitting at a terminal and the computer which the user is accessing across the network attempt to agree which computer will echo the characters the user types):

At about the same time, one system sends “Hey would you please do echoing for me” while the other system sends “Please do local echoing”. The two requests pass like ships in the night.

Both systems are flexible, so the first system responds “OK, I’ll echo” while the second also responds “OK, I’ll echo for you”, with the messages again passing each other.

And so, still trying to be cooperative, both systems switch their echo modes “If you insist, I’ll shut off echos and let you echo” passing “If you insist, that’s fine by me I’ll let you echo”.

And so on.

At the UCLA meeting, Cosell used a blackboard and his waving hands to describe his idea for negotiated options to the assembled group.

His basic idea involved four commands: DO, DON’T, WILL, WON’T. Since there a number of capabilities besides which computer echoes that needed to be agreed on between the two computers, depending on what each computer is capable of doing and what is desirable to the user given the capabilities of the computers (for instance, whether line-at-a-time or character-at-a-time input is supported), his solution involved using the four commands as prefixes for the various optional Telnet capabilities:

WILL (option code) Indicates the desire to begin

WON’T (option code) Indicates the refusal to perform, or continue performing, the indicated option.

DO (option code) Indicates the request that the other party perform, or confirmation that you are expecting the other party to perform, the indicated option.

DON’T (option code) Indicates the demand that the other party stop performing, or confirmation that you are no longer expecting the other party to perform, the indicated option.

Thus, with regard to which computer echoes characters, one or more of the following commands might be used:

WILL Echo
WON’T Echo
DO Echo
DON’T Echo

The purpose of this historical account is not to provide a comprehensive tutorial on how the DO/DON’T/WILL/WON’T prefixes work to negotiate the options specified following the prefixes. You can read the details in \[3, 4, 9\] or try to derive it yourself.

### 4. Diffusion of the Solution

Walden came out of the meeting at UCLA with the task of writing it all down, and Cosell was to proofread the writeup to make sure that the protocol really did “close” properly. Cosell remembers, “Even at the meeting we didn’t make the details of DO/DON’T/WILL/WON’T clear — I was confident that it was all OK, but it seemed too complicated to bother with trying to explain in detail, so the RFC was the first place (other than in the airplane, if you count hand waving…) that the whole thing was pinned down.”

Upon arrival back at BBN, we immediately went to find Jerry Burchiel and Ray Tomlinson. Jerry and Ray had been key members of the BBN team that developed the TENEX time-sharing system \[2\], and TENEX systems had been or were being installed at a number of sites on the ARPANET (there were probably more TENEXs on the ARPANET at that time they any other single kind of computer system). Among other things, Ray was responsible for TELNET on TENEX and, because he was in
effect supporting many instances of the system, he had the same problem we had with the TIP.

Jerry and Ray could usually be found several times a day having coffee in the BBN cafeteria and discussing their various technical projects, and that’s where we found them and described Cosell’s idea to them. They quickly understood the idea and agreed that it looked good.

Over the next little bit of time, Walden drafted a written description of negotiated options, had it reviewed by Cosell (and possibly Burchiel and Tomlinson and possibly others) and published the description in RFC 435 [3].

At the time, the TELNET specification was also being drafted (or perhaps an earlier draft was being revised). In any case, Walden also looked at the non-negotiated versions of a number of existing basic TELNET options and rewrote them in terms of negotiated options. He also added a few TELNET options to the negotiated options list. He remembers doing this writing on one weekend when he and his wife were baby-sitting for his sister-in-law’s children in a suburb outside of Boston. He had taken his SCM portable electric typewriter and plenty of white-out correction tape along to where he was baby sitting and drafted this rather major update to the TELNET specification essentially in one sitting. He mailed it to the person editing the TELNET specification (Jon Postel, perhaps) early the next week.\(^2\)

There was immediate general acceptance of the concept of negotiated options, and it was implemented relatively soon throughout the host computers of the ARPANET. That brings us to the end of our story.

A few years later, the history and status of TELNET until then, including negotiated options, was documented in a multi-author paper [4] (which Walden initiated and pulled together primarily with the help of Bob Thomas of BBN).

**Authors’ Biographies**

Bernard Cosell and David Walden were both members of the Bolt Beranek and Newman Inc. (BBN) team that developed the ARPANET IMP, a well known effort [1, 5]. In the years that followed, each worked on many other ARPANET and Internet projects, including the ARPANET TIP project.

Mr. Cosell attended MIT and received his Master’s degree from Northeastern University. After many years at BBN, he left in 1992 to raise sheep on a farm in Virginia. He also works at RevNet Technologies and teaches at Virginia Western Community College, and he continues to whip out and implement new computer communication ideas.

Mr. Walden received his undergraduate degree from San Francisco State College and did graduate study at MIT. After many years at BBN, he retired in 1995 and now lives primarily on Cape Cod, where he does a lot of writing on technical and management topics.

**References**


---

\(^2\) In [7], Alex McKenzie says Walden also had help with these options from Cosell, Tomlinson, Bob Thomas, Burchiel, and Dave Crocker. Walden does not remember if he ever lived up to the promise to document more options that Alex reported in the RFC.


◊ Bernard Cosell  
  252 Merino Lane  
  Pearisberg, VA 24134  
  bernie@fanstasyfarm.com  
  www.fantasyfarm.com

◊ David Walden  
  12 Linden Road  
  E. Sandwich, MA 02537  
  dave@walden-family.com  
  www.walden-family.com