Networked Vessel Systems

Thanks Nigel!

We took great pleasure reading Nigel Calder's article in the March/April issue of *Ocean Navigator*[1]. The article is about networked systems on boats as opposed to discretely wired systems which we all now have. By using networked smart controls to replace breakers, and switches, lots of wires can be eliminated, saving weight and copper costs. At the same time installation costs will be less and the time to checkout the system at the builder will be less . . . money saved!

Do beg, borrow or steal a copy of Ocean Navigator and read Nigel's article. Its well written, very informative, and we'll refer to and respond to some of the subjects he discusses.

Offshore by Wire

Are you ready to go offshore in a networked boat? Most cars made in the last 10-15 years have one or more networks. Airplanes have their share of networks. Industrial sites all over the world rely on networks. Almost certainly there is a network where you work. However, anytime I've broached the subject of networked boats to sailors, I've gotten the look that suggests they're one step away from calling the looney farm to make reservations for me.

I asked myself a similiar question when I was building my sailboat in the 1970s. Would I venture offshore with an electrical system typical of that era? Of course not, and that was really the start of Ample Power, although it wasn't until returning from cruising that Ample Power products could be enjoyed by someone other than ourselves.

Once upon a time

Networks are about as old as computers. In the 1960s I was employed by a small company designing digital voltmeters and automatic test systems. We were bought out by a major U.S. corporation that was building process control systems for their own use, and to sell to others. Projects we worked on included systems for a carpet manufacturer, a fiberglass manufacturer, a magnetic tape manufacturer, the NASA space center in Houston, traffic signal coordination, and NY State Gas and Electric. These systems used one or more small computers with networks to end point electronics. By nature, process control systems are spread about because of the physical space required by the machinery.

Moving from mini-computers to microcomputer in the 1970s reduced the cost of a networked node, and lead to more networking. Leaving the bureaucracy of the large corporation I began to work as a contractor/consultant and was the technical lead for other networked systems such as early parking and revenue control systems, the second and third electronic funds transfer systems installed in the U.S., and an online, realtime embossing system that stamped out vehicle identification tags just in time to be riveted to the chassis.

Keep it Operating

As you can imagine, car manufacturers don't like to have their production line stalled because the i.d. tag that must be riveted to the inside dashboard is not in the output hopper when the dash assembly comes past the i.d. station. Failures cost money . . . lots of money. More money than maintaining another device that instantly takes over if the primary device fails. How does that work? By networking.

There are worse failures than stalling an automobile manufacturing plant. How about melting down a nuclear power plant or spewing radiactive uranium hexachloride from an enrichment facility into the atmosphere?

One of the people I worked with on the embosser project had been hired to manage system designers working on two major projects. One was a networked system that connected a nuclear power plant to its control center, several miles distant. The other was a system to manage the process of enriching uranium. The former system was fairly well along, but still lacking a reliable network between the generating facility and the management center.

The system for uranium enrichment was just getting off the ground. The mandate for the system was 100% redundancy such that no single failure could lead to faulty control. There were two mainframes computers and up to 500 mini–computers spread about the complex. Each one of them had an associated network controller. Tens of thousand of parameters had to be measured, filtered, compared against one another and communicated throughout.

While the sheer number of parameters was daunting, they could be reduced to a few dozen measurement types, resolution and accuracy. After doing that, I turned the requirements over to the engineering staff who would design the circuits.

Numbers First

As the technical right hand for the project manager I got to sit through many boring meetings, not the least of which was focused on the state of software development. At one of those meetings, a presenter displayed a graph that showed a measurement cycle from the many process electronic racks. He traced the flow of data through the system to the two mainframe computers and how control information would flow back to keep the enrichment process on track.

Because I had cataloged the measurements and control signals, and had previously considered what the network bandwidth had to be, I saw right away that response to measurements would be at least 10 times slower than what the presenter described. After the meeting I talked with the presenter about the source of his numbers and discovered that he was using an *estimate* of the network traffic. The network designers were operating without hard numbers!

The specification from the DOE required that an abnormal event at the process must be annunciated in the main control center no later than two seconds, without interruption of normal communications. Another clause in the specification mandated the use of commercial operating systems for all computers, which in this case meant a Unix variant from Digital Equipment Corporation. Almost everyone agreed that two seconds wasn't possible with the real data volume using Unix system calls. Based on the ongoing problems with the communications controller for the power plant, which we were expecting to use as well, the consensus was even worse. We were headed for a train wreck.

[1] Ocean Navigator, POB 569, Portland ME, 04112-0569

Shoot the Messenger

Since I was the messenger of bad news, I was assigned to come up with a solution. One of the seasoned programmers volunteered to help me. We left the office early that afternoon to drink a few beers and eat some Mexican food. Yes, engineers and programmers do actually make notes on napkins and within a few hours we had devised a framework.

To make it work, we needed to convince the engineers at the DOE to exclude the network controller from the Unix requirement. There was already an exemption for I/O processors, so reclassifying the network controller as an I/O processor could possibly be done without going through too many layers of bureaucracy. The DOE engineers could see the problem, and so, at their suggestion, the network controller became the network I/O controller.

The end result hadn't changed . . . we still had to provide ultimate reliability with end–to–end redundancy, but do it a lot faster. Hardware and software had to be much closer coupled, and I was the translator between the engineers and programmers. Everyone realized we couldn't rely on traditional methods unless we could show that they met our needs. New ideas and new terminology germinated, some to be uprooted, and some to be cultivated. From the threads came a weave and from that a cloth.

Toss out the old

The hardware changes were dramatic, but actually quite simple using standard modules. Into each minicomputer and the two mainframes we plugged in a second processor, to run the network, and a shared memory card that let the network controller dump packets directly into the memory space of the minicomputer and likewise, extract packets from the shared memory.

The network medium was fiber optic cable arranged in a double redundant fashion. There were four ways to deliver data packets and the system could operate normally with two of them defective. We pushed the baud rate up to a non-standard value just within the flat–out processing capabilities of the network computer. Then came the hard part, hand crafting assembly language programs to operate the network node and handle all the potential fault conditions. It didn't happen overnight, but those who said it couldn't be done . . . were wrong.

It's all quite simple . . . isn't it?

Having worked on large systems and small ones, standalone systems and distributed networked systems I know that design issues must pass muster from many engineering perspectives. Too often, however, the boilerplate part of the job blurs the vision of someone who might have caught the missing part.

Satellite launches fail and space telescopes end up disfunctional because metric measurements get mixed up with inches. Space shuttles fail from purely mechanical problems like loosing heat–shield tiles. Online computer systems crash from overloads because no one devised a way to gracefully throttle the system.

So, what kind of networked system will you trust for your next offshore vessel?

I've never met a transistor I didn't like

I owned a 1967 VW squareback with electronic fuel injection. A few months past the warranty period the vehicle went back to the dealer because it was running erratically. It needed a new *computer* board . . . $470 in 1968 money. I paid, because what else can you do? But we took the old part and with nothing more than a diode checker found a transistor that was dead. Replacing it with a *garden variety* transistor from my grab bag fixed the unit. I sold the vehicle, along with a spare *computer* board and went back to mechanical systems.

I stayed clear of electronic systems in vehicles until we purchased a used 1993 Ford pickup. It wasn't very long before it developed a shifting problem and we paid for a new torque converter, although that seemed an unlikely cause. Shortly thereafter, and still under the dealer warranty, the problem recurred . . . the dealer connected their diagnostic device and simply reset the error condition. The transmission worked again as it should, but 600 miles later it was again shifting *clunky*. Parking it for three weeks and replacing the battery in the meantime fixed the problem. The next time the tranny acted up, I stopped and disconnected the battery for a few minutes. That cured the problem until some random time later it would recur. Obviously there wasn't a real problem with the transmission, just one with the microcomputer monitoring it. Eventually I discovered the rpm/acceleration conditions that would trigger the fault. Bye, bye Ford.

To this day, I don't own a vehicle with a network. Why? The answer will emerge later. The first clue is in the paragraph below.

If a transmission breaks a gear and scatters itself on the roadway, the problem is obvious. If a bug in the microcomputer that manages a transmission randomly declares a fault condition, the average consumer doesn't have much choice but to return to the dealer and pay up. If there is anyone out there who thinks this system has been designed for the best interest of the consumer let me tell you about the great deal we can offer you on ocean front property in Ames, Iowa.

To be continued . . .

In coming installments we're going to look at networking systems and strip away some of the smoke and mirrors. We'll discuss relibility, maintainabilty, repairability, growth, and many other issues that determine how successful the system will be. We'll present some terminology, like fail safe and fail over. If you'd like to be notified when the next installment is released, send us an email. The subject should read *Network Blog*. See the image below for the email address.

In the meantime, don't forget to read Nigel's article in Ocean Navigator. One issue he discussed was troubleshooting and repairability by the owner. Is that possible?

smead@amplepower.com