X.25: WHAT TO DO AFTER THE NETWORK IS IN PLACE

Stephen J. Coya
MCI Digital Information Services, Washington, D.C., USA

Summary

Many organizations are discovering the advantages of using X.25 packet switching networks to support their data communication requirements. However, it's not until the network is installed that the real "fun" begins: connecting all the hosts and getting them to talk to each other. This is especially true when hosts from different vendors are to be connected to the network. While the OSI model provides a good theoretical framework, many vendors have implemented this model in different ways. As a result, diverse, and sometimes incompatible, protocols must be accommodated and coerced into cooperation before they can communicate.

This paper will review the problems (and solutions) encountered connecting HP3000s, HP1000s, and DEC VAX 11/785s together over a nationwide private X.25 network. It will touch on the physical connections to the packet switch (OSI Level 1), a special Transport Layer protocol that had to be developed (OSI Level 4), and some of the problems encountered with DSN/DS (combination of OSI Levels 3 and 4) and what we had to do to make it all work.

Introduction

We knew from the start it would be interesting. Many companies have made the transition to X.25 networks with relative ease, but this can often be traced to systems where only one vendor host is involved. For example, a network consisting of HP3000s and the selection of a network switch vendor that supports such an environment. The MCI Mail system, an electronic mail system with hardcopy support, was designed around the use of VAX hosts for the mail application and HP3000s to process and print hardcopy traffic on 2680A Laser printers, and we knew the hosts would be connected to the network switches (PSNs) at 56 kilobits per second (kbps). We drew network (cloud) and system diagrams, examined technical specifications for the hosts and the switches, realized what could and could not work, and reached for the bottle of aspirin!

I should mention that there were time constraints involved that did not permit extensive experimentation or research, and the capabilities that exist on today's equipment did not exist three years ago. The system, designed in the latter part of 1982 was built in six months during 1983. Even after details were worked out in joint design meetings, special plugs and cables had to be developed on the fly as we tried to meet our target date, which we did.

Transport Layer (OSI Level 4)

The first "Gotcha" was getting information from the VAX hosts to the HP3000s. All VAX hosts were located in a central facility while the HP3000 based print sites were set up across the country. Tape transfers were out of the question, and the old standby, RJE, did not fit the overall system design. When DSN/DS was initially developed, it was for point to point communications between HP equipment. It was expanded to support X.25 interfaces, but still required the use of DS as a transport layer protocol. So one still had to use DS to transfer files to an HP3000. Well, DS was never implemented on VAX equipment, and we needed something that would talk straight X.25 with the VAXen and DS with the HP3000s. Enter the HP1000.

The HP1000 was inserted into the network diagrams between the VAX hosts and the HP3000s (we also started using pencils instead of pens to draw the diagrams). We used two LAPB modem cards and two physical connections to the packet network. One modem card was set up to use straight X.25 (DSN/X.25) to communicate with the VAX hosts. The second card was configured to use DS (DSN/1000-IV) to communicate with the HP3000s. Simple? Not quite. While the HP1000 could use DS to communicate with the HP3000, there was no transport layer protocol to communicate with the VAX.

The solution was to design a Simple File Transfer Protocol (SFTP) to accommodate the need for a transport layer. This protocol was implemented on the VAX equipment and on the HP1000s. As its name implies, it is a simple protocol: no CRCs, just simple checksumming and byte counts. It was also implemented on an IBM 4341 which was connected to the network and served the application as the accounting and invoicing host.

So, we had finally identified the host equipment that would be connected to the packet network and had designed a protocol that would provide the OSI level 4 Transport Level between the VAX and HP1000 equipment. Back we go to the basics, OSI Level 1, and connect the equipment to the packet switches.

Physical Layer (OSI Level 1)

OSI Level 1 pertains to physical connections, which pins are used for what signals, and the electrical levels of those signals. Unfortunately, there are a number of different standards that have been recommended and implemented: RS232c, RS449/RS422, and V.35 were the ones we had to work with (See figure on last page).

The packet switch nodes (PSN) supported RS449 physical connections. The switches did not supply a clocking signal, and the hosts were to be connected directly to the PSNs (no modems) at 56 kbps. I mention clocking because at that time neither the HP1000, HP3000, or the VAX could supply clocking at 56 kbps (if they could we didn't know about it). So we had to connect three different hosts that support three different interfaces to a packet switch that supported only one of the three.

The HP1000 was the easiest as it uses RS449 as a physical interface. All we had to do was insert a RS449 Synchronous Modem Eliminator (SME) between the HP1000 and the PSN to provide clocking at 56 kbps. Since then, we have been successful in operating without an SME as the HP1000 will now supply the clocking signal.

The HP3000, however, uses V.35 as its physical interface. To establish this connection, a V.35 cable from the INP is used which terminates in a Winchester type connector. The next link is a cable which terminates on one end with a Winchester connector (connected to the HP3000 INP cable), and on the other end with a DB-37 connector. The DB-37 connector is wired to conform to the RS449 specification, the electrical characteristics of which conform to RS422 (RS422 is the electrical specification for signals carried on an RS449 interface). The DB-37 connector is then plugged into a SME. From the SME comes a cable to a special V.35/RS422 convertor that is plugged directly into the HP3000's port on the PSN. Two down, one to go.

The VAX supports RS232 connections, which is rated at 19.2 kbps up to 75 feet. However, the interface from the VAX to the packet switch goes through a KMS11 processor which has a rated speed of 56 kbps. A shielded cable is used to connect the KMS11 to a RS232/RS449 SME which is then connected to the packet switch.

So we now have all the hosts connected to the network, transport layers are in place, and we can transfer files where they need to go. All done? Not yet - the system must be operated and strange things can happen!

To Flush or not

Our network supports data packets of up to 1024 bytes in length. When data is to be transmitted from the HP3000 over the packet network, the DS protocol sets up a buffer containing four 1024 byte packets. The packets are sent one at a time to the local packet switch which acknowledges receipt of each packet. After the fourth packet is transmitted and the local PSN acknowledges receipt, the buffer is flushed and the next group of 1024 byte packets are placed in the buffer. This looks acceptable on paper, but in practice caused some problems.

As the packets traveled through the network on their way to the destination PSN (the switch connected to the host to receive and process the data from the HP3000; in this case, another HP3000), a network problem caused a RESET to be generated by a packet switch. This RESET packet is returned through the network to the originating HP3000, essentially stating that something bad happened and requesting that the packet be retransmitted. If the reset is received by the HP3000 before the fourth packet has been acknowledged, the packet is located in the buffer and retransmitted.

However, if the reset is received after the fourth packet has been acknowledged, the buffer has been flushed and the packet is not available to be retransmitted. When this happens, DS just sits there wondering what to do. Recovery from this state required bringing down the DS connection, bringing it back up, and starting the transmission all over again.

The problem was reported to HP, and our SE captured numerous dumps to substantiate our claim, especially after HP informed us that 1) they couldn't reproduce it and 2) the local PSN had acknowledged receipt of the fourth packet, so it was a network problem, not an HP problem. That they couldn't reproduce it was understandable; they'd need to replicate our system switches, configuration, traffic patterns, line speeds, etc. Their second claim was a little hard to live with as it wasn't a "network" problem, but a SYSTEM problem. HP's position was understandable albeit cavalier. By the way, the occurrence of a RESET packet is part of the CCITT X.25 recommendations.

The solution? You won't believe it. We received a DS patch from HP that essentially said "If network=01 (BBN), don't flush the buffer until acknowledgment received from destination PSN."

Error Checking and DS

As I mentioned earlier, DS was written originally as a point-to-point transport agent for the HP3000s. It was later modified to support X.25 connections as a short term solution until a better defined interface was developed. It should be noted that error checking in DS is only done at the packet level (not message level) with the local PSN.

Lets step back a moment and look at what a packet switch is: a computer. It's a special purpose computer, but a computer none the less. And what makes a computer work? Software. A packet switch contains an operating system, tables, microcode, routing algorithms, etc. Guess what happened when we upgraded the switch hardware and installed a new version of the PSN software designed to run only on the upgraded software.

The new equipment and software was tested for some time in a test mini-network we maintain that replicates our operational system, and we verified its functionality, new capabilities, and features. Unfortunately, there's one thing that cannot be tested in our mini-network: LOAD. The new hardware was installed throughout our network and the new software was propagated over some period of time. Two weeks after full propagation we got burned!

There was an obscure bug in the PSN software release that caused an "overlay" of data, overwriting the data contents of a packet in the memory of the PSN. The PSN then calculated the checksum and sent it on through the network. The "bad" data was not noticed by the HP3000 when received, since error checking is done on a packet level and the checksum for the packet was ok (as it was not calculated until after the overlay), and it was accepted. The "clobbered" message was printed and no error was encountered...but there was an error in the message, one that got through the network unnoticed by any of the software!

The PSN software was backed out and is only now being repropagated as the cause of the problem has been identified and corrected. I mention this here because there IS a deficiency (or hole) with DS and X.25 networks in that these errors are not discovered. What is needed is an end-to-end protocol that can be implemented on both the originator and receiver of the message so that error checking may be done on the message level itself, not just at the packet level which we now see is inadequate. But there is hope.

NS/3000

The latest release of communication software from HP will include, among other features, straight X.25 support; decoupling the requirement of DS as the transport layer protocol, though it will still be available and probably widely used. As an alternative to DS, NS/3000 will support the TCP/IP protocol which includes an end-to-end (host-to-host) checksumming algorithm that can be turned on and off. This is message level checking, not packet level, and data errors can be identified and recovered from by the software.

As TCP/IP has been implemented on VAX equipment (The Wollongong Group), it is feasible to use this protocol to permit VAXen and HP3000s to communicate directly with each other without the HP1000 being in the loop. However, there are costs involved that must be considered before we make the decision.

In our application, removing the HP1000s from the communication path would eliminate a central focal point for real time monitoring and controlling of the hardcopy system. On the other side of the coin, it would remove a potential bottleneck as 24 VAXen now communicate with 20 HP3000s through three HP1000s.

The network impact of removing HP1000s must also be considered as this would dramatically alter the flow of data over the packet switched network as all VAXen and HP3000s would communicate over virtual circuits, increasing the traffic and complexity of the network environment. A number of network topology studies would have to be conducted to anticipate the changes to the network, design host redeployment strategies, and determine the network modifications required to accommodate these changes.

Another hurdle is that DEC has not officially "sanctioned" the Wollongong Group implementation of TCP/IP. Indeed, DEC would prefer your using DECNET in an all VAX environment, just as HP would prefer you using DS in an all HP3000 environment. All this really means is that DEC is not actively marketing TCP/IP, and users must ask about it specifically. Just remember, user needs always outweigh vendor desires, especially since it is the user doing the buying.

And finally, the benefits of implementing TCP/IP must be compared to the costs. Error detection and recovery handling adds to the processing requirements, which is one reason HP will permit this function to be turned on and off (I do not know if the VAX implementation includes this option or not). If the network and transport protocols are solid and the encountered error rate is sufficiently low or within acceptable levels within the current environment, the benefits of introducing end-to-end error checking may not offset the costs incurred in modifying the system to support the protocol. If data integrity is critical and there is a history of damaged or incomplete messages/file transfers, the benefits may outweigh the costs of implementation.

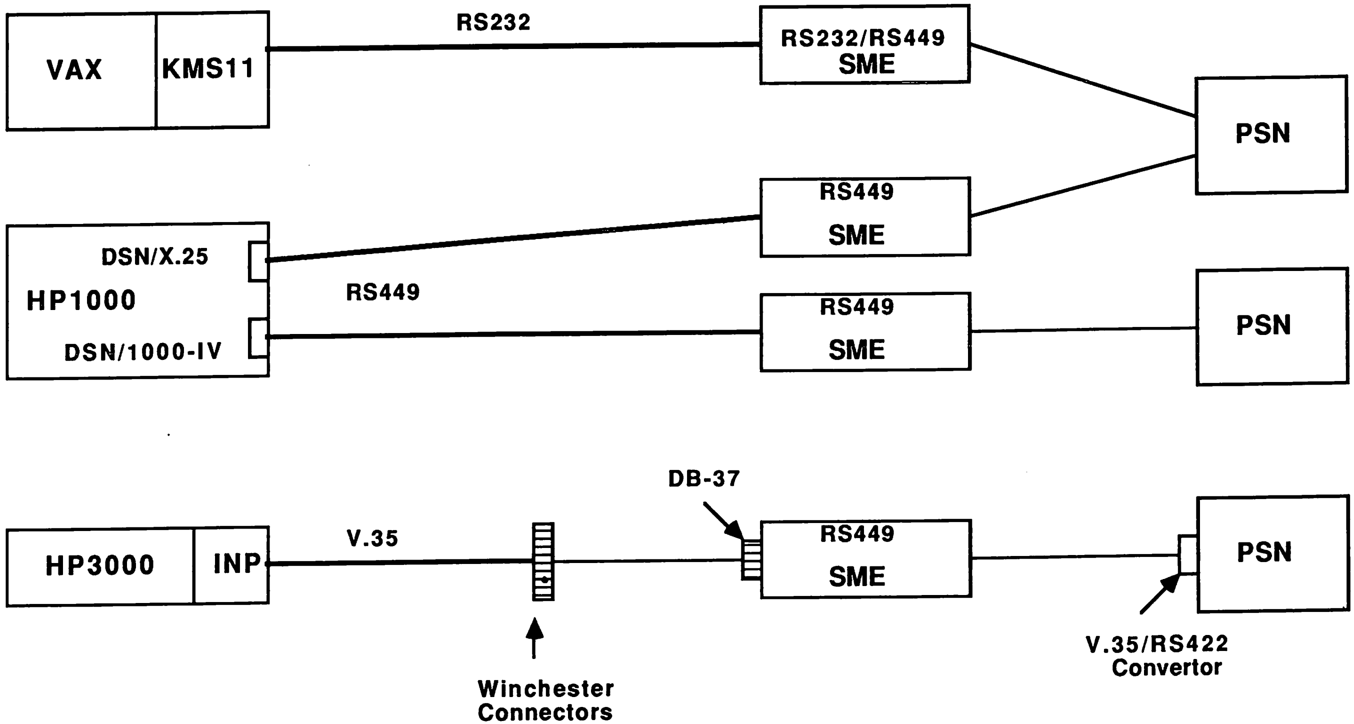
Conclusion

In this paper, I have attempted to provide a little insight (and humor) to the difficulties of bringing up a multi-vendor system on a packet switching network; that there is more to do than wait for the network to be installed before you just plug in the hosts. I included a couple of "horror" stories to illustrate that the job isn't done when everything has been connected and initial communication tests completed, but that it is an on-going effort to operate, improve, and even upgrade the system to support current and future requirements.

MCI believes in open architectures and makes no secret as to how the MCI Mail system is implemented. In fact, a more detailed description of our application and the network was the topic of another paper delivered at the Madrid INTEREX conference. I have limited the length of this paper and presentation so that there will be enough time to answer any questions that might be raised, or to discuss other related topics; including the experiences or plans of other organizations that might be building a system from scratch as we did, or who might be planning a transition to a X.25 network based system.

Biography

Steve Coya has been with MCI Digital Information Services for the past three years. He is the Senior Project Manager for MCI Mail Hardcopy Systems Development, and is also responsible for the overall planning and scheduling of system integration tests and for managing and coordinating the implementation of new software releases into the operational environment.



Physical Connections to the PSN