Digital's Multiprotocol Routing Software Design

1 Abstract

The implementation of Digital's multiprotocol routing strategy required addressing various technical design issues, principally the stability of the distributed routing algorithms, network management, performance, and interactions between routing and bridging. Developers of Digital's DEC WANrouter and DECNIS products enhanced real-time kernel software, implemented performance-centered protocol software, and used high-coverage, high-quality testing and simulation methods to solve problems related to these issues. In particular, a packet management strategy ensured that queuing requirements were met to guarantee the stability of the routing algorithms. Also, network management costs were minimized by down-line loading software, using a menu-driven configuration program, and careful monitoring. Router performance was optimized by maximizing the packet forwarding rate while minimizing the transit delay.

Digital's implementation of multiprotocol routing software enables internetworking across complex topologies including local and wide area networks (LANs and WANs) and dial-up networks. Evolving from Digital's successful tradition in DECnet Phase IV networks, the implementation of multiprotocol routing currently supports numerous protocol and packet types including

- DECnet Phase IV
- Transmission control protocol/internet protocol (TCP/IP)
- Novell NetWare internetwork packet exchange (IPX) protocol
- AppleTalk protocol suite
- OSI CLNS, the open systems interconnection protocol for providing the connectionless-mode network service
- X.25, the packet switching standard specified by the Comité Consultatif Internationale de Télégraphique et Téléphonique (CCITT)

Additional extensions for Digital's DECnet Phase V and ADVANTAGE-NETWORKS architecture requirements are also supported by Digital's multiprotocol routers.[1,2] Many of these routers incorporate bridging technology, thus providing integrated bridging routers. This paper describes the most significant technical problems encountered and the solutions implemented when many internetworking operations are integrated into Digital's multiprotocol router system designs.

2 Digital's Router Product Overview

Digital's multiprotocol router products comprise two types: (1) access routers, which allow access to WAN services from branch offices for large LAN and WAN integration networks, and (2) backbone routers, which provide high-speed packet switching services for the network backbone of multiple types of high-speed media. Backbone sites offer a backbone network that often consolidates high-speed WAN lines, e.g., T1, T3, and SMDS. For high-speed local sites, backbone routers provide high-speed switching for many LAN ports and types, i.e., Ethernet, fiber distributed data interface (FDDI), and token ring. This section briefly discusses some of Digital's access routers-the DEC WANrouter 500, DEC WANrouter 250, and DEC WANrouter 90 products - and backbone routers-the DECNIS 500 and DECNIS 600 products.

The DEC WANrouter 500 is one of Digital's access routers and has been available in the marketplace since 1986. Originally a DECnet Phase IV-only router, this router has been upgraded and now offers multiprotocol routing that includes DECnet Phase IV, TCP/IP, and OSI. Additional support exists in this access router for common WAN services such as X.25 and frame relay. The DEC WANrouter 500 is a fixed-port configuration router offering one T1 WAN port and one Ethernet LAN port. This configuration permits branch office LANs to interconnect to backbone routers over relatively high-speed T1 lines. The DEC WANrouter 500 has an important place in router industry history as it was the first router ever to support the integrated intermediate system-to-intermediate system (Integrated IS-IS) routing algorithm.[3]

The DEC WANrouter 250, another of Digital's access routers, is significant due to its high density of WAN ports and its support for asynchronous WAN data link protocols. These two major features combine with the multiprotocol routing software to provide a router for the newly emerging computer networking needs of mobile computers. The increasing use of personal computers, including mobile laptop computers, has led to the development of new techniques for networking such remote computers. The DEC WANrouter 250 provides eight WAN ports with dial-in access for the internetworking of such remote and mobile computers.

The introduction of LAN hub technology has produced a need for new small router products for these platforms. Digital's DEChub 90 Ethernet backplane product set includes the DEC WANrouter 90 access router shown in Figure 1. One feature of the DEChub 90 technology is that this router can be configured to reside within the hub itself or as a standalone module. In addition, this router is completely self-contained and extremely small (i.e., similar in size to a VHS videocassette). Many WAN access services, such as X.25 network access, are provided for the DEChub 90 with the DEC WANrouter 90 router.

NOTE

Figure 1 (DEC WANrouter 90 Access Router) is a photograph and is unavailable.

The DECNIS 500 and DECNIS 600 (see Figure 2) bridging and routing products are Digital's highest performing and most flexible platforms. These backbone routing systems offer the power and interfaces necessary to meet the bridging and routing requirements of complex, high-speed networks, e.g., Ethernet, FDDI, T1/E1, and T3/SMDS.[4]

NOTE

Figure 2 (DECNIS 600 Backbone Router) is a photograph and is unavailable.

3 Router Software Development Methods

Software development for routing systems requires real-time kernel software, performance-centered protocol software development implementation, and high-coverage, high-quality testing and simulation methods. This section briefly describes some key techniques used in these development areas for the DEC WANrouter and DECNIS engineering programs.

Kernel Software

Digital has developed and refined different kernels with common interfaces to address the real-time software design environments required for their routers. A common router interface model has permitted different kernels to be turned to specific platforms as required. In some cases, a common portable kernel was developed that permitted quick retargeting of the total router software in support of short time-to-market development needs.

Software Implementation

The following techniques were used in the development of the DEC WANrouter and DECNIS router software:

1. Implementing software directly from proprietary or standards-based architecture specifications

2. Licensing software from suppliers, e.g., external corporate software providers and government-funded university software projects

3. Importation of software from other implementations, i.e., host sources such as the ULTRIX, Open Software Foundation (OSF), and OpenVMS systems

Digital developed special-purpose, high-performance implementations of the Integrated IS-IS routing protocol. In addition, specific software kernels provide control and extensions for the special features required. Engineers enhanced the real-time software kernels with software interfaces commonly found in public domain software (e.g., the Berkeley Software

Development [BSD] UNIX socket model and system services). The inclusion of such interfaces has accelerated the addition of new software from external sources.

Digital's Multiprotocol Routing Software Design

Common router software has been developed for use across Digital's many internetworking platforms. The majority of this routing software, which is independent of the underlying hardware, has been developed to support the evolving standards of portability. For each platform, the performance-intensive and hardware-specific code have been customized to maximize the design center for each instance of a router product architecture.

Router Software Design Issues

Many technical problems had to be resolved when building Digital's multiprotocol routers. The following sections describe the most significant issues and how they were addressed in the DECNIS 600 backbone router, as an example of router software design. These issues were

1. Stability of the distributed routing algorithms

2. Network management

3. Performance

4. Interactions between routing and bridging

Memory size and usage and congestion control are also key issues. However, this paper does does not describe them in depth. Briefly, the amount of memory available is a major constraint on any router implementer. Usually, memory is largely consumed by code and by the databases the router must maintain to calculate the best route. In the case of routers that also perform connection-oriented functions (e.g., X.25 gateways and terminal servers), significant amounts of memory may be taken up by the per-connection state and counter information.

Since it is essential for routers in the network to agree on the best route to a destination, all such routers must be able to handle the route database for that network. Digital's router designs have an automatic shutdown mechanism that takes effect should a router run out of memory in which to store routing information. This mechanism prevents routing loops.

To control OSI congestion, the router must determine whether or not a packet experienced congestion by calculating the average transmission queue length over time. This calculation must be performed in an efficient real-time manner. Thus, for the DEC WANrouter and DECNIS products, Digital designed and implemented algorithms specific to the particular queue structures and hardware design.

4 Stability of the Distributed Routing Algorithms

Distributed routing algorithm stability was the most important issue considered in the design of Digital's router systems. A system design must guarantee successful results in support of routing control protocols even when the router is operating under a high load.

Whatever protocol is used, dynamic routing requires that all nodes that make decisions on how to forward data should agree on the correct path.

Otherwise, data packets will be discarded (e.g., if sent to a node that does not know how to reach the destination) or may loop (e.g., if two routers each believe the other is the correct next node on the path to the ultimate destination, then the packet will loop between the two routers).

If network configurations never changed, and lines and routers never got overloaded, then guaranteeing successful results would be easy. Unfortunately, actual networks are complex. In practice, for each protocol, the correct path agreement is reached using an algorithm distributed between multiple independent routers and operating on ever-changing data.

The distributed algorithm must converge rapidly so that when network conditions change, the new route is agreed upon quickly. However, the algorithm must also be stable. When changes occur at a fast rate or when the algorithm is trying to complete or has just completed, the algorithm must still converge to a consistent state between all the routers involved. In this way, the network remains useful. In addition, while the network is changing, a router or a line may suddenly be presented with an excessive load of packets to forward (e.g., because a routing loop occurred transiently). This situation must not be allowed to disturb the stability of the routing algorithm.

The stability of a well-designed routing algorithm is directly related to how well the algorithm meets the following main requirements:

- Line speed. The effective speed of lines between routers (allowing for error correction by the data link protocol or the modem) must be high enough to allow the routers to rapidly exchange routing control information. The maximum bandwidth required for routing control traffic can be calculated from the size of the network.[5] In a network of 4,000 end nodes, 100 level 1 routers, and 400 level 2 routers, approximately one Link State Packet (LSP) will be received every second. This LSP may contain 1,500 bytes, which would use a line bandwidth of 12,000 bits per second. This aspect of stability is under the control of the network designer; line speeds and network size must be continuously monitored and related.

- Processing power. The router CPU must be fast enough to forward routing updates to neighboring routers with minimum delay and must be able to recalculate the forwarding database quickly. Of course, this requirement relates only to that portion of the CPU time available for routing functions. A router that is also doing another job (e.g., acting as a file server) will have less CPU power available, unless routing is given priority over the other functions. Consequently, most networks now use dedicated routers instead of attempting to have routing tasks share the CPU with other functions.

- Queuing. The most important stability factor is to make sure that the systems are self-stabilizing. As the problem gets worse, progress to the solution should not become slower. For example, as the network configuration changes more rapidly, the calculation of the best route must not get slower. To meet this requirement, the routers must be

careful about queuing data and routing control messages internally so that excessive or unusual data forwarding loads do not affect the processing of routing control messages. Otherwise, when a network problem overloads a router, the routing algorithm may never converge to fix the problem.

Figure 3 illustrates a case where an incorrectly designed router (one that gives priority to data forwarding over routing control message reception and processing) could cause a permanent routing loop and thus isolate a portion of the network. In this example, node A is sending a large amount of data to node F over the high-speed T1 line. The lower-speed (64 kilobit-per-second [kb/s]) line is available as a backup line. Because the backup line runs at only 64 kb/s, node C need only be a low-power router. For example, a router rated at 128 packets per second would be sufficient because a fully saturated full-duplex 64-kb/s line with 128-byte packets handles 128 packets per second.

Consider what happens if the T1 line fails. Router B notices immediately and begins to forward data to router C. Initially, however, router C still believes the best route to node F is over the T1 line and so forwards the data back to router B. B resends the data to C and so on; a routing loop has been created. This problem is common during routing transitions. The loop will be broken as soon as router C runs the decision process and updates its routing tables. However, if router C is incorrectly designed and gives priority to forwarding data, then the unexpectedly large amount of data will "swamp" the router and prevent it from running the decision process.

In addition, since router C is only a low-speed router, it will be forced to discard many data packets. Eventually, the transport connections between node A and node F will fail, because packets are not being delivered (presumably causing the applications to fail). This situation will reduce the number of packets being introduced into the loop. However, each packet can go around the loop many times, thus generating a high load. In this example, if nodes are set up such that a packet can travel the loop 64 times (a common value), then introducing only two packets into the loop per second will continue to swamp router C. Any node on the LAN might be sending those packets to discover when access to the remote LAN is restored. The effect is a long-lived routing loop that isolates the whole LAN, even though there was supposed to be a backup link available.

- Memory usage. Activities less important than routing should not consume the memory necessary for routing control processes to carry out their function. Even in a dedicated router, some lesser activities will be in progress. For example, network management and accounting are important activities, but they are not as critical as maintaining network stability-without a stable network, network management and accounting will fail. Therefore, other activities should not starve the routing control processes of memory. Consequently, traditional memory pools are not an appropriate way to allocate critical memory within the router; routing memory usage must be preallocated.

The remainder of this section describes the impact of the requirements on processing power, queuing, and memory allocation on the design of the DEC WANrouter and DECNIS products.

Requirements on Processing Power

The Digital Network Architecture (DNA) routing architecture requires that routing updates be propagated within 1 second of arriving and that the forwarding database calculation take no more than 5 seconds.[5] The forwarding database calculation is CPU-intensive, but the time is proportional to the number of links reported in LSPs. To meet the DNA requirement, various measurements were made for each product to determine the number of links the decision process could handle per second. This information indicates, for each product, the maximum number of links allowed in the network. Note that this number does not directly limit the number of nodes permitted in the network; a large network with an efficient connection strategy may have fewer links than a small network in which every node is connected directly to every other.

The update process latency requirement means that the CPU time must be fairly allocated between the decision process and the update process. If the update process was required to wait until the decision process had completed, then the delays on forwarding LSPs would be too large (i.e., 6 seconds).

We considered three possible solutions.

- Process priorities. Give the update process a strictly higher priority than the decision process so that the database can be updated as required. The main issues to resolve are synchronizing access to the shared LSP database and allowing the decision process to complete, if a faulty router generates LSPs at an excessive rate.

- Timeslicing. As in a traditional timesharing system, allow both processes to run simultaneously, thus sharing the CPU. This solution also requires synchronizing access to the LSP database.

- Voluntary preemption. The decision process periodically checks to see if the update process is required and, if so, dispatches to it. This check can occur at time intervals frequent enough to meet the latency requirements and at times convenient to the decision process so that no synchronization problems occur.

To avoid the synchronization problems, Digital's DECNIS 600 software developers chose the third solution for two reasons.

1. Synchronization issues often cause problems that are serious and difficult to debug in complex systems. By avoiding these issues entirely, we simplified the software and increased its reliability.

2. The addition of synchronization mechanisms for parallel tasks can

decrease the performance of the total system (for example by causing excessive rescheduling operations). Using voluntary preemption allowed a very efficient solution that still met the architectural requirements.

Requirements on Queuing

Queuing constraints ensure that high loads do not cause routing control information to be discarded. Initially, separating the data for forwarding from routing control messages might appear to be the logical solution to preserving routing control information. However, this solution works only if the router can process all the routing control messages without getting behind.

Many practical routers, including the DEC WANrouter products, do not have a CPU that is fast enough to guarantee such processing performance. Digital's routers can guarantee to meet the timing requirements on the decision and update processes (even under worst-case loads), but if that load is combined with a flood of End-node Hello messages, Router Hello messages, and other control traffic, then some of those messages have to be discarded or queued for later processing. Since there might be 1,000 or more nodes on the LAN, the worst situation would be if all these nodes were to decide to send Hello messages at the same time.

Careful software design means that the routers can meet the network stability requirements and still not lose connectivity to end nodes on the LAN. For the DEC WANrouter software, Digital designed and implemented a packet management policy that differentiates between routing packet types to meet their respective processing requirements for network stability. The following list summarizes the classes of packet types:

- Data
- End-node Hello messages
- Router Hello messages
- Link State Packets and their acknowledgments, Sequence Number Packets (SNPs) and Complete Sequence Number Packets (CSNPs)

The parameters controlling the minimum and maximum numbers of packets to be used for each differentiated type are carefully calculated based on their architected behavior and the network configurations supported by each product. For example, a router's architected design center for supporting a given maximum number of adjacent routers on an attached LAN will affect the policy selected for managing the Router Hello message queues and packet buffers. Such mechanisms are implemented to guarantee that, for network stability, forwarding performance, and network convergence, the minimum levels of forward progress per packet type are met.

This packet management policy uses both buffer pools and queuing to implement the required policies. Inbound traffic is placed on queues that

are serviced using variants of round-robin algorithms. These algorithms give different weightings to each queue to ensure that progress is made for every packet type, although at different rates.[6] For example, for every data packet processed, the router may process 5 LSPs, 5 End-node Hellos, and 10 Router Hellos. The actual weightings used are selected when

the software is designed and depend on the performance characteristics and expected network configuration of each product.

Some alternatives that were considered are

- Alternative buffer pools. A completely separate pool can be used for each of the different types of packets. The disadvantage is that in small configurations or ones that are not under heavy stress, the pool of buffers available for forwarding is limited unnecessarily.

- Strict priorities. Setting strict priorities for processing different types of routing control messages is undesirable, because a flood of one type of routing control message could cause another type to be ignored for a long time. In such a case, it is better to process some of each type of message than to give one type absolute priority.

In the DECNIS routers, several queues exist at the boundaries between the different DECNIS processors.[4] Digital designed a mechanism for these queues similar to that described for the DEC WANrouter products. When the network interface cards, i.e., linecards, receive a packet destined to be passed to the management processor card (MPC), they analyze the packet and tell the MPC whether it is data, routing control, bridging control, or system control (which includes linecard responses to commands from the MPC). Thus, queues analogous to those described for the DEC WANrouters are used at all the interfaces within the system. For example, the assistance processor on the MPC recognizes the different types of messages and queues them on separate internal queues.

Requirements on Memory Allocation

Routers must have sufficient buffer space to handle the routing control messages. Consequently, all of Digital's router products guarantee this memory allocation. To preserve these buffers, the DECNIS MPC implements buffer swapping between layers, as illustrated in Figure 4. The data link layer must never be starved of buffers; otherwise, packets regarded as important by routing may be discarded without ever being received. To ensure that an adequate number of buffers is available to the data link layer, the MPC gives the data link a certain number of buffers and maintains that number. Every time a buffer is passed from the data link layer to the routing layer, another buffer is swapped back in return. If routing currently has no free buffers, it selects a less important packet to discard (freeing up the buffer containing the packet). In this way, the data link layer always has buffers available.

In the DECNIS linecard buffers, the arrangements are similar to those just described, but the details differ. The linecards and the MPC perform buffer swapping among themselves.[4]

5 Network Management

Some of the highest costs involved in running a network are those related to obtaining and maintaining trained and experienced network managers and operators. Minimizing these costs requires routers that can be easily and efficiently managed. The major network management issues are

- Installation/loading. How are software updates distributed and installed? How long does the router take to load after a power failure?
- Configuration. How is the software told about changes to the lines or the network parameters? Does the network require a reboot to change information?
- Monitoring. How does the manager get immediate reports of problems and unexpected changes, and long-term reports of traffic patterns and usage for network planning?
- Control. How can the manager shut down a line or even a whole router?
- Problem solving. What tools are available to detect the problem and then to investigate and correct the problem?

In all networks, though, a remote management capability is essential. Skilled network management staff may not be available at all sites (e.g., a small branch office). In fact, some sites may have no staff at all (e.g., a lights-out computing center).

Installation and Loading

All DEC WANrouter and DECNIS products update their software by down-line loading new software over the network. In the case of the DECNIS, the software is stored in nonvolatile memory and so does not need to be reloaded on each boot. However, the DEC WANrouter products down-line load the software each time they are booted.

Digital considered two other alternatives.

- Read-only memory (ROM). This means of distribution has the disadvantage of being expensive to modify and difficult to replace remotely.
- Floppy disk or other interface on the router. This mechanism increases cost and reduces reliability. Loading from a floppy disk may also be slower than loading over a network. Again, remote updating may not be possible, and physical security issues (e.g., preventing unauthorized users from supplying uncontrolled router software) may be introduced.

For the DECNIS product, Digital chose to use nonvolatile memory, e.g., flash random-access memory (RAM), for fast and reliable loading combined with backup down-line load operation when software updates are required. The down-line load can be from a DECnet system using the maintenance operations protocol (MOP) or from a TCP/IP host using the boot protocol (BOOTP) and the trivial file transfer protocol (TFTP).[1] The down-line load provides an easy way to update software when required; the

software can be installed on a load host using any of the standard software distribution mechanisms (e.g., CD-ROM, magnetic tape, or the network).

Configuration

Configuring a router is notoriously difficult. Therefore, Digital developed a tool to assist the network manager with configuration. Each of the DEC WANrouter and DECNIS products comes with a configuration program. This menu-driven program leads the network manager through a series of forms to define the information needed to configure the router or to modify an existing configuration. On-line help is available, and steps may be retraced. Consequently, the network manager has no need to learn the network control language (NCL).

Digital used formal human factors testing during the design and development of the configurators to ensure that these tools were of high quality. Human interface testing continued through the router's customer field trials and provided additional feedback on our configurators' ease of use.

One thing that Digital did not originally anticipate is that users now tend to see the configurators as the user interface for the product. The configurator is often a customer's main means of interacting with the router and thus is an essential part of the product. Once people have used the configurator, they no longer regard it as an optional feature.

Monitoring

Digital's routers are fully manageable using Phase V network management. They all respond to NCL commands and can be managed using the DECmcc program, Digital's Enterprise Management Architecture (EMA)-compliant director. Therefore, DECmcc added-value functional modules are available for performance analysis and historical data recording. The DECmcc design enables these functions to work without changing the router design.

Many users, however, are now investing in management stations that use the simple network management protocol (SNMP). Thus, for monitoring purposes, Digital already implements basic read-only SNMP management, which is being enhanced over time to add more information.

Control

Whether managed by the NCL or the DECmcc director, access is controlled using passwords. In addition, Digital is focused on offering full SNMP management for the router products. As well as providing the standard public management information, Digital is defining private management information to allow unique features of the routers to be controlled. We designed the internal management interfaces of the routers to allow

us to write modules that are manageable from both the SNMP and the common management information protocol (CMIP), with minimal effort and duplication.

Problem Solving

One of the most time-consuming, and hence expensive, parts of a network manager's job is problem solving. Fortunately, many of the tools and techniques used for this task were required for debugging and testing router implementations and thus already exist.

Building initially on debugging and testing experience, and later on discussions with users, Digital has produced problem-solving guides for each DEC WANrouter and DECNIS product. These guides take the user through a step-by-step description of how to isolate and fix a problem. We have conducted human factors testing on these guides and have investigated different modes of making this information available. The DECNIS guide is currently available in hard copy and also in an on-line Bookreader form that allows moving through the flow to be automated using hot spots. Digital is currently evaluating Hypertext technology to further improve the usability. One main tool for problem solving is the common trace facility (CTF), a software tool that causes the router to record and display packets that are sent and received. Analysis routines automatically format the packets. Having the CTF is comparable to having a built-in line or LAN analyzer. The CTF is the main diagnostic tool used by Digital's service engineers when investigating a problem and also by the development engineers when debugging software.

Digital's routers also include diagnostic and maintenance facilities, which include loopback testing over all interfaces and low-level, limited, remote management directly at the data link layer. The remote management capabilities allow monitoring of counters from an adjacent node and also allow an adjacent node to force a reboot if a suitable password is supplied. This latter operation is referred to as a MOP boot (previously known as a MOP trigger in DECnet Phase IV).[1]

A MOP boot command may be the final attempt by a network manager to fix a problem with a router without having to go physically on site. For that reason, the command must be recognized and acted upon regardless of what else may be happening in the router. In the DECNIS routers, the MOP boot command is recognized by the linecards. In the DEC WANrouter, the MOP boot command is specially actioned by the lower layers of the software to make sure it is honored even if the higher layers have failed in some way or if the system is under an enormous load.

We also support the "TCP/IP ping" utility (more formally, ICMP Echo) and the similar "OSI ping" utility. These tools are commonly used for diagnosing reachability problems.

6 Router Performance

Today's large-scale computer data networks rely on bridge router components for the networks' total level of performance and quality of service. As such, data network designers and network managers must be knowledgeable about their chosen router platform's performance characteristics. This section of the paper discusses the performance aspects of Digital's routers.

Performance Metrics

In support of developing common metrics across the internetworking router industry, the Internet Engineering Task Force (IETF) has set up a Benchmarking Methodology Working Group, which has developed definitions for router performance.[7] Three key metrics defined by this group provide the background for our discussion of Digital's router software design.

- Throughput-the maximum (forwarding) rate at which none of the offered frames (packets) are dropped by the device (i.e., packets per second)

- Frame loss rate-the percent of frames (packets) that should have been forwarded by the network device (router) while under a constant load but which were not forwarded due to lack of resources (i.e., percent packets lost)

- Latency-for store-and-forward devices (i.e., routers), the time interval beginning when the last bit of the input frame reaches the input port and ending when the first bit of the output frame is seen on the output port (i.e., units of time)

In the design of Digital's router software and systems, a balance has been targeted with maximizing the packet throughput forwarding rates while minimizing the packet latency. Some vendors mistakenly compare loss-free throughput rates with forwarding rates that have high loss rates. Such comparisons must be studied carefully, because they do not compare route performance measures of equal impact to the total network. To reiterate, the throughput forwarding rate occurs only at the point when the frame loss rate is zero percent. Digital's routers target throughput designs which, as much as possible, run at "wire speed" with zero frame loss rates. Regardless of the throughput value quoted, router comparison should reference common packet loss rates because network applications need to retransmit any packets that are lost by the routers.

In general, the throughput, loss-free forwarding rate is the optimum value for discussions of router forwarding performance. The other critical value is the stability of the router under heavy overload. A "receive livelock" condition occurs when the offered load, i.e., input packets

received for subsequent forwarding by a given router, reaches the point where the delivered throughput, i.e., packets actually forwarded, decreases to zero.[8,9] Real-time systems, such as routers, have the potential to livelock under traffic loads above their throughput peaks. However, it is extremely important that routing implementations avoid such responses to post-throughput saturation. In the case of Digital's routers, in all

architectures and products, the routers do not livelock but remain stable even when the applied input load to a router exceeds the peak throughput forwarding packet rate. This key performance measure of router devices remains an underlying design characteristic of all Digital DECNIS and DEC WANrouter network devices.

Packet Throughput/Forwarding Rate

Digital's routing platforms offer a range of throughput measures. For each platform, the throughput is the most often quoted value used to characterize the router's aggregate capabilities. In the case of the DECNIS 600, an aggregate throughput of 80,000 packets per second is offered.[10] In smaller routers, the WAN line interface rates (i.e., 64 kb/s and T1) are often the limiting factor for the aggregate throughput. The software in all cases is optimized for the given router platforms mix of WAN and LAN interfaces.

Since the forwarding rate is the most important performance metric for a router, Digital carefully optimized the designs of its multiprotocol routers to allow data forwarding to occur as fast as possible. On the DEC WANrouter products, we handle all the forwarding on a central CPU with little hardware assistance. In the DECNIS products, forwarding and filtering operations are handled by linecards. A hardware assist for the performance-critical forwarding function's address lookup is used on DECNIS routers in support of requirements for very high-speed packet switching.[4] On each linecard, a streamlined software kernel has been developed along with all the required software. The linecard software kernel and modules were carefully constructed to have the minimum number of instructions and the lowest number of execution cycles necessary to perform the high-speed forwarding and filtering operations. On the DECNIS MPC, the software kernel is also fully capable of the routing forwarding operations. However, this kernel is mainly required to provide the software processing for the remaining non-performance-intensive operations of the router's software (i.e., the processing of updates to the router topology database and the network management commands/received packets). This partitioning of processing of received packets in the DECNIS router system permits such routers, and the networks that they comprise, to remain highly stable when traffic overloads occur.

For the DEC WANrouter software, the forwarding operation has no hardware assist. Software lookup assist algorithms have been researched and implemented to help meet the performance-intensive requirement. As in the microcoded DECNIS linecard adapter software, the software is highly tuned for performance. To minimize the additional maintenance overhead associated with highly tuned software, the amount of such code is kept to a minimum. The DEC WANrouter software design is an example of how Digital carefully balanced product performance requirements and product development

and maintenance costs to meet the required price/performance goals for its router product family.

Packet Latency (Transit Delay)

The next most frequently specified performance requirement is packet latency or packet transit delay. For bridge/router devices, this measurement clearly depends on software and hardware timings. However, the definition of latency utilized corresponds directly to the chosen system's design.

The previously quoted IETF definition for store-and-forward devices can be further refined to accommodate differing device designs. The IETF working group clarifies the difference between a "store-and-forward device" and a "bit-forwarding device" internal design model for a router. The latter design model is often referred to as a "cut-through" design and requires a different definition than previously listed for store-and-forward devices. The definition of latency used for this cut-through model is the time interval starting when the end of the first bit of the input frame reaches the input port and ending when the start of the first bit of the output frame is seen on the output port.[7]

The issue that distinguishes the two models is whether or not processing starts prior to the packet being completely received. However, another key point is whether or not the packet received can be sent out for transmission prior to complete reception. When reception, forwarding, and transmission can occur in parallel, the design is referred to as cut-through. For Digital's router designs, the DECNIS does process reception and forwarding in parallel prior to a packet being completely received. However, the DECNIS does not start transmission until a packet is completely received. Thus, the DECNIS latency model uses the original store-and-forward definition of the IETF.

In the case of the DEC WANrouter software, the model and definition used is again store-and-forward. The factors that control the packet latency in the DEC WANrouter design are as follows:

1. Receiving the packet. The packet must be completely received.

2. Performing the forwarding operation. This factor includes packet verification, analyzing the packet, performing any required address lookup, performing any required packet modifications, and queuing the packet for transmission on the destination interface.

3. Congestion queuing. If the destination interface is not idle, the packet will have to be queued before transmission. Some transit delay measurements use only uncongested media interfaces connected to the router. However, latency measurements must be made to measure the potential latency delays due to congestion at the router output interface. The packet latency due to queue occupation delays is also

included here. Congestion avoidance algorithms have been implemented to minimize this congestion delay.

4. Transmitting the packet. This factor is usually dominated by the time taken to clock the bits of the packet out of the interface but also includes media access times, i.e., delays due to another node already using a common connection.

We now examine how the DEC WANrouter and DECNIS routers separately minimize the transit delay.

The DEC WANrouters minimize the packet reception and transmission portions by allowing hardware to perform these functions using direct memory access (DMA). Because these systems have only a single processor, the forwarding delay is minimized by the same fast-path optimizations used to improve the forwarding rate.

On the other hand, the optimizations for the DECNIS routers are slightly different for the various linecards. The DEC WANcontroller 622 card has no DMA, and the linecard on-board processor is involved in receiving each byte of the packet. We parse the header as soon as there is enough information to do so. For example, the data link packet type field is decoded before the network address bytes have been received, and the network address lookup is initiated as soon as the address has been received (i.e., before the data has been received). The address lookup is then performed by the address recognition engine hardware without further involvement from the software.

The DEC WANcontroller 618 card and the DEC LANcontroller 601 and 602 cards all receive packets one segment at a time. Internally, these cards use small fixed-size buffers that are linked together as necessary to store a whole packet. Again, they perform the analysis and forwarding lookup as soon as the data is available (i.e., when the first segment is received).

Thus, for a large packet, the entire forwarding decision will have been made before the last byte has been received. However, note that until the last byte has been received, it is not known whether the cyclic redundancy check (CRC) is correct or the packet has been corrupted. So the packet is not actually passed to the destination linecard until that check has been completed. As discussed before, this design is still store-and-forward, rather than cut-through. The DECNIS design goals were easily met without using cut-through; however, Digital has used the cut-through design on a number of LAN host-based adapters.

When a packet is to be transmitted, certain changes must be made in the data. For example, the IP and OSI protocols require that time-to-live fields and, in some cases, other options be modified. Bridged packets may need address bits modified or conversion between Ethernet and IEEE 802 forms. As with reception, all DEC WANcontrollers perform these operations as the data is transmitted. All cards have hardware assistance

for recalculating header checksums and CRCs.

These features are designed to reduce the forwarding delay as much as possible, so that the transit delay is mainly controlled by the time it takes to receive and send the packet. The type of architecture that best describes the DECNIS design is a data-flow, which blends traditional store-and-forward designs with newer cut-through designs. This data-flow architecture processes packets in a distributed manner (i.e., linecards process packets) without transmitting packets prior to complete reception validation of these packets. This design limits the forwarding of packets that are found to be in error, whereas the similar full cut-through design would propagate invalid packets.

7 Interaction Between Routing and Bridging

Designing a combined router and bridge product is complicated by the relationship between the routing and bridging functions.[11] A received packet must be subjected to either the bridge forwarding or the routing forwarding process (or maybe both).

Several designs are possible and are illustrated in Figure 5.

a. Protocol split. In this design, some protocols are bridged, e.g., Local Area Transport (LAT), and others are routed, e.g., TCP/IP. The bridging and routing functions are completely separate; they merely share line interfaces. Every packet received is passed to either routing (if intended for a protocol that is being routed) or bridging.

b. Integrated with one interface. In this design, the routing function is modeled as being layered on top of the bridging function. Theoretically, packets are subjected to the bridging process and then, if they are addressed to the router, subjected to the routing process. In this form of the model, the router uses a single logical interface seemingly connected to a private LAN contained within the bridge/router.

c. Integrated with multiple interfaces. This design is similar to the integrated design with one interface, but the router uses all the available interfaces and logically connects to the same extended LAN multiple times.

Each design model has advantages and disadvantages, and we considered all three models for the design of the DECNIS routers. The protocol-splitting model has the advantage of simplicity. The major disadvantage is that any particular protocol must be either bridged or routed. The integrated models have the disadvantage of requiring specific management to prevent a routed protocol from also being bridged. In most cases, a protocol is being routed specifically to avoid the problems associated with bridging. The model with one interface also has the disadvantage that the network manager may get confused attempting to work out which interface is being used for routing.

We chose the protocol-splitting model because of its effectiveness and ease of use.

8 Special Considerations of the DECNIS Architecture

We have discussed special features of the DECNIS system architecture. Now we present some additional DECNIS software design issues.

Control and Management of Linecards

Each linecard is a separate software environment and must be managed and controlled by the management processor. The main tasks required are

- "Watchdog" polling. In a standalone network server product, it is necessary to guard against the software getting caught in an infinite loop and hence not responding to management and control messages. The management processor is protected by a hardware watchdog timer, but the linecards do not have a timer. To protect the linecard software, we designed the management processor software to poll each linecard every 400 milliseconds (ms). If there is no response, we reset the card.

- Counters. The network interface cards handle data forwarding and therefore must maintain the required counters (e.g., the number of data bytes received). However, to avoid requiring the linecard to maintain 64-bit counters (which costs memory and requires 64-bit arithmetic), the management processor maintains the full counters and polls the linecards frequently enough to guarantee that the on-card counters do not wrap. Each counter is sized to support the design of the management processor polling every 400 ms.

- Control. When a data link protocol or a routing protocol is started or stopped on an interface, the management processor receives the network management command and issues appropriate control messages to the network interface card.

Distributed Forwarding

Each linecard normally handles the forwarding of bridged and routed data without involving the management processor. This design requires a different approach to meeting the stability requirements from that described for the DEC WANrouter devices.

For example, the DEC WANrouter products discard data packets to meet the routing stability requirements. This discard is limited by the packet management mechanisms to guarantee a minimum level of forwarding performance for the other routing packets, even under worst-case conditions such as those caused by network topology changes. The DECNIS routers do not normally have to discard packets, because the network interface cards can continue to forward data while the management processor handles the routing protocol operations. In addition, correctly designed linecard software

guarantees that control traffic is passed to the MPC, even in cases where the software is also passing large amounts of data traffic to the MPC.

9 Conclusion

This paper describes the complex nature of the design decisions required in the development of Digital's multiprotocol router systems and software. The issues and solutions discussed show how many conflicting technical requirements can be addressed. One example of such a conflict is related to the designs goals for the performance of Digital's multiprotocol routers. While on one hand achieving extremely high system throughput (i.e., the DECNIS 600 router supports a forwarding throughput rate of over 80,000 packets per second), the DECNIS 600 design also addresses the equally critical metric of router stability (i.e., the DECNIS 600 product remains stable under extreme network loads).[10] This balancing of requirements is key to justifying Digital's approach toward router product engineering. As summarized in his recent book on computer systems performance analysis, Raj Jain states that

> The performance of a network ... is measured by the speed (throughput and delay), accuracy (error rate) and availability of the packets sent.[12]

Routers that can forward packets but cannot remain stable under heavy loads, or meet the requirements for bursty packet rates as required by many of the newer network applications (e.g., packet-based videoconferencing systems such as Digital's DECspin product), will fail to satisfy customers.[13] As such, Digital provides a well-tuned, optimized total network solution with DECNIS 600 routers and DECspin products. This synergy of Digital's network applications and network infrastructure components is the ultimate justification for the multiprotocol router design decisions outlined in this paper.

10 Acknowledgments

Many engineers in Australia, England, Ireland, and the United States participated in the design and implementation of the Digital's multiprotocol routers. We wish to thank all of them.

11 References

1. DECnet Digital Network Architecture (Phase V) General Description (Maynard: Digital Equipment Corporation, Order No. EK-DNAPV-GD-001, 1987).

2. J. Martin and J. Leben, DECnet Phase V (Englewood Cliffs, NJ: Prentice-Hall, Inc., 1992).

3. R. Perlman, R. Callon, and M. Shand, "Routing Architecture," Digital Technical Journal, vol. 5, no. 1 (Winter 1993, this issue)

4. S. Bryant and D. Brash, "The DECNIS 500/600 Multiprotocol Bridge Router and Gateway," Digital Technical Journal, vol. 5, no. 1 (Winter 1993, this issue)

5. DECnet Digital Network Architecture (Phase V) Network Routing Layer Functional Specification) (Maynard: Digital Equipment Corporation, Order No. EK-DNA03-FS001, 1991).

6. E. Coffman, Jr., and P. Denning, Operating Systems Theory (Englewood Cliffs, NJ: Prentice-Hall, Inc., 1973): 169.

7. S. Bradner, "Benchmarking Terminology for Network Interconnection Devices," Internet Engineering Task Force RFC 1242 (July 1991).

8. K. Ramakrishnan and W. Hawe, "The Workstation on the Network: Performance Considerations for the Communications Interface," IEEE Computer Society Technical Committee on Operating Systems, vol. 3, no. 3 (Fall 1989): 29-32.

9. K. Ramakrishnan, "Scheduling Issues for Interfacing for High Speed Networks," Proceedings of Globecom '92, IEEE Global Telecommunications Conference, Session 18.04, Orlando, FL (December 1992): 622-626.

10. S. Bradner, "Interop Fall 1992 Router Performance Study," technical presentation, Harvard University, 1992.

11. W. Hawe, M. Kempf, and A. Kirby, "The Extended Local Area Network Architecture and LANBridge 100," Digital Technical Journal, vol. 1, no. 3 (September 1986): 54-72.

12. R. Jain, The Art of Computer Systems Performance Analysis, ISBN 0-471-50336-3 (New York: John Wiley & Sons, 1991): 23.

13. R. Palmer and L. Palmer, "DECspin: Networked Multimedia Conferencing for the Desktop," Digital Technical Journal, vol. 5, no. 2 (Spring 1993, forthcoming).

12 Biographies

Graham R. Cobb Graham Cobb is a consulting engineer in the Internet Products Engineering Group and was software project leader for the DECNIS 500/600 router development. Graham holds an M.A. in mathematics from the University of Cambridge and joined Digital as a communications software engineer in 1982. He has worked on many Digital communications products including X.25 products and routers and was a major contributor to the DEC WANrouter 100/500 software immediately prior to leading the DECNIS development. Most recently, Graham has been working on new-generation routing software.

Elliot C. Gerberg Elliot Gerberg is a senior engineering manager in Digital's Networks Engineering Division, managing the Routing Engineering

Group (USA). Since joining Digital in 1977, he has worked on numerous projects including the DEUNA, Digital's first LAN adapter; the DECserver 100, Digital's first low-cost terminal server; the SGEC, a high-performance Ethernet semiconductor interface; and various multiprotocol routers. Elliot has a B.S. in physics from SUNY and an M.S. in computer science from Boston University. He holds professional memberships with the IEEE, the ACM, and the Internet Society.

13 Trademarks

The following are trademarks of Digital Equipment Corporation:

ADVANTAGE-NETWORKS, Bookreader, DEC,
DEChub, DECmcc, DECnet, DECNIS, Digital, DNA, OpenVMS, and ULTRIX.

AppleTalk is a registered trademark of Apple Computer, Inc.

BSD is a trademark of the University of California at Berkeley.

NetWare and Novell are registered trademarks of Novell, Inc.

OSF is a registered trademark of Open Software Foundation, Inc.

UNIX is a registered trademark of UNIX System Laboratories, Inc.