The Common Printer Access Protocol

James D. Jones, Ajay P. Kachrani, and Thomas E. Powers

1 Abstract

The concept of a "Printer Access Protocol" or "PAP" was introduced during the mid-1980s to provide for comprehensive access to the new generation of more capable, networked printers. The design goals for a PAP need to consider the variety of data presentation protocols in use and the heterogeneous nature of distributed interconnection methods. A Digital printing architecture team adopted an existing prototype, the Reid-Kent Print Server Protocol, as the basis for a proposed "Common Printer Access Protocol," or CPAP. Digital's first server instantiation of the CPAP, PrintServer Supporting Host Software version 4.0, also needed to address implementation practicalities to ensure interoperability and backwards compatibility. The solutions selected are applicable to a broad range of client-server systems where clients and servers may be independently developed, and where such components may be installed and upgraded asynchronously from one another. CPAP is being considered for adoption as an Internet standard, and is an element of the Palladium Printing System, which has been accepted as the printing systems component of OSF/1. [Protocol paper starts here.]

The presentation of computerized data has become a remarkably sophisticated and subtle operation. Video displays now support windows with complex allocations of display space, variable fonts, and real-time user input operations. Printing devices now offer support for publication-quality fonts, line art, and images. These devices can present visual objects on a variety of media, from many sources, and in variable orientations and presentation modes. In addition, both video and printing devices are now decoupled from dedicated computing environments, and are shareable from many hosts and by many users or programs.

Now, only the simplest printing devices are limited to presenting just characters, and many users are finding such restricted capabilities inadequate. Also, most printing devices still require dedicated connections to single computers. However, more printers now offer full network accessibility; i.e., network printers are capable of offering sophisticated services to a wide variety of users and their applications.

The paper entitled "Design of the DECprint Common Printer Supervisor for VMS Systems" in this issue of the Digital Technical Journal describes access methods and interrelations among services that provide for these increasingly sophisticated data presentation capabilities.[1] The printer access protocol (PAP), a service interface in the DECprint architecture, couples the printer supervisor component to the logical printer for presenting data and otherwise controlling a physical printing device. The common printer access protocol (CPAP) described in this paper provides the fundamental services required by a printer supervisor for the presentation of data and collection of accounting information. In addition, the CPAP supplies easier network access between printer supervisors and printers, as well as ancillary control of printers for network management and device configuration. The CPAP also provides services to distribute the processing requirements of the printer itself, most notably a mechanism for delivery of network font services. This last capability allows a printer to offer what amounts to virtual services, i.e., the ability to configure itself dynamically to the demands of a print job without the involvement of the printer supervisor.

This paper begins with a discussion of the influence of existing protocols and the DECprint architecture on our CPAP design goals. The sections that follow present the printer session concepts and the functional interface between the protocol and applications. We then describe the implementation of the new protocol in a server environment, including interoperability, compatibility, and the translation of the older PrintServer protocol. At the close of the paper, we discuss ongoing standardization issues.

2 History

Digital's first fully networked printer, the PrintServer 40, was first shipped in 1986. Its local area print server (LAPS) protocol was analogous to later printer access protocols. The PrintServer 40 was a ground-breaking product for Digital, and the LAPS protocol was a major aspect of the PrintServer development effort, portions of which date back to 1983. The LAPS protocol was designed and developed with particular product-oriented deliverables in mind, and was optimized for VMS access and DECnet transport. While this protocol predates much of the architectural work now being implemented in Digital's printing products, it was (and still is) a significant element of PrintServer architecture and implementation.

Work began on more general PAPs in 1987 as part of the early work on the DECprint architecture (known at the time as the Printing Systems Model). The specifics of what would become the CPAP emerged in late 1988 in two internal papers by Brian Reid and Chris Kent of Digital's Western Research Laboratory. These papers presented the initial design concepts for a PostScript-based, TCP/IP-connected (transmission control protocol/internet protocol) print server in a clearly defined client-server environment. This print server protocol came to be known as the Reid-Kent protocol.

Design Rationale and Goals

By early 1988, design goals for (and constraints on) a PAP were well understood, and had been collected and published as part of Digital's Printing Systems Model. Chief among these goals and constraints was the need to support a variety of data presentation protocols, and to allow printers to be connected to driving applications by a variety of communications and process-to-process interfaces.

The increasing corporate commitment to open systems made it clear that a PAP would also have to couple entities running various operating systems across different networks. Thus, the DECprint PAP architecture team decided early in the design process that a PAP should be designed for public access; that is, the specification for the protocol should be put into the public domain and submitted for industry standardization.

Interoperability is a most serious constraint. Digital has a strong tradition of maintaining backward compatibility within and among its product families. In a distributed processing environment, however, backward compatibility takes on the added burden of interoperability. Multiple clients must communicate with multiple servers, any of which can be upgraded to new versions of supported protocols asynchronously. Addressing this problem was a major conceptual test in the first implementation of a CPAP server. This is discussed in more detail in the section The CPAP Server Implementation.

The Reid-Kent protocol met many of the technical design requirements for a new PAP. It was built on industry-standard components, and contained no proprietary technology that would prevent its publication.

However, certain PAP design goals were not covered by the Reid-Kent protocol in its 1988 version.

- There was no facility to select a specific page description language (PDL) for printers supporting multiple interpreters.

- There was no method for soliciting the capabilities and media available on the printer.

- The only language supported was English (contrary to the corporate guidelines for internationalization).

- Data sent from the printer was not categorized; user-specific information was mixed with operator and service data.

- No means was provided to solicit the status of the printer.

- There was no encoding to discriminate between binary and text files.

However, these flaws were largely omissions from the design goals, not fundamental conflicts with them. The architecture team decided that the Reid-Kent protocol could be extended to address these omissions without serious conflict. In fact, the necessary extensions were designed to allow clients and servers conforming to the original Reid-Kent protocol to remain in conformity with the full CPAP specification.

3 Architecture

The CPAP is primarily a communication-oriented protocol, i.e., the presentation of its function is closely coupled with its encoding. The major syntactic features of the CPAP derived from the Reid-Kent protocol are the following.

- All encodings are ASCII strings. This eases the generation of protocol streams and ensures independence from the underlying communications channels.

- No data fields are fixed length. This provides for extensibility of the protocol and eases the generation of a protocol stream.

- Multiple channels of communication use the same basic format. Common parsing of separate channels simplifies implementations.

- Simple numeric tokens define the operators.

Session Concepts

The CPAP architecture defines separate contexts for each type of work the CPAP can perform. Each context requires that a separate session be established for its own tasks, and each session involves the creation and use of a separate network connection between the controlling client and the server. Each connection identifies the type of session the initiator requires. The CPAP defines three different session types: print, management, and console.

The set of CPAP operators allowed for a session is restricted to those needed to support that type of session. All session types have access to printer status and configuration information. In addition, multiple concurrent sessions are permitted. Print sessions and management sessions may have one or more virtual circuits active to a printer at a time. The use of multiple circuits permits the streaming of data to the printer over logically separate channels, thereby eliminating application protocol overhead for the most frequent operations. In contrast, console sessions use a single virtual circuit for exchange of data with remote terminals.

Print Sessions Print sessions usually consist of a series of documents printed for a user on a given host by a printing service (a "printer supervisor" as defined by the DECprint architecture). With the operators provided by the CPAP, the printing service can determine the language interpreters, printer options, fonts, prologues, and media that are currently installed at the server. These operators also provide the current operational state, number of jobs queued to the printer, and the current job status. These features permit the printing service to select the printer (server) that can satisfy the user's request and to determine a method for submitting the job to the printer.

Once the printing service has begun a session and identified itself, it identifies the user and the user's job code to the printer. This information may be used by the printer to provide usage information to a centralized accounting service. The printing service can then present documents to the printer. A transaction between the printing service and the printer establishes which interpreter the printer will use for each document and which virtual circuit will be used for its transmission.

Selection of the proper virtual circuit for transmission of documents to the printer is performed by passing tokens from the printer to the printing service. The tokens are then mapped to whichever virtual-circuit service is being used by both the printing service and the print server. This mapping approach avoids passing network-specific information within the protocol. Not only does the approach make the CPAP independent of the networks on which it might run, it ensures that the network services need no knowledge of CPAP encodings. Such virtual-circuit mapping is critical to allow CPAP client-server processing to be implemented in a heterogeneous, internetworking environment.

During the printing of the document, some data presentation interpreters (PostScript, for example) send data back to the user or print service. In addition, the printer may run out of paper or toner, may have a full output tray, or may encounter other exception conditions not directly related to the interpretation of page description data. The CPAP categorizes such conditions and delivers relevant messages to the user, the operator, or the event logs.

Upon completion of the job, the printing service is notified of the media used, the number of pages printed, and the printer processing time required to complete the job. The protocol also includes a provision to abort jobs, e.g., an improperly formed document that might otherwise hang the printer.

Management Sessions The CPAP supports certain printer services through management hosts. A management host is a network entity (not necessarily the same entity as the printing service) with which the printer can exchange information or request services. Such services include

- Time service
- Centralized event logging
- Centralized accounting
- Program loading and configuration
- Font services

An important aspect of the CPAP is that the printer is always passive with regard to initiating management services. A candidate management host advertises that it has services to offer, and a print server accepts or rejects the offer. Once a connection with one or more management hosts is established, the printer may use such hosts as servers for time synchronization, configuration file access, and font lookup. Additional functions for these hosts may be loading program images, event logging, accounting, and general file access.

File naming to access general file services is a problem that needs special attention if the server and the protocol are to maintain independence from the host operating systems. Commonly used files are identified in the CPAP by reserved tokens, such as $CONFIG, $DEFAULTS, $RESOURCES, and $SETUP. Arbitrary path names are allowed, but can access only a limited domain (from a known root directory) to preserve file system independence and to maintain security.

Translation to the host's services is provided by the host itself. This permits the printer to be served by different hosts using a wide variety of operating systems (and their implicitly different file-naming conventions and syntaxes) without any awareness of a management host's implementation by the server.

Console Sessions A console session is a form of printer management. The content of the data exchanged during a console session is specific to the printer, and is not specified by the CPAP. Services performed within a console session might include

- Operator services, such as telling a printer what media have been loaded (e.g., by color, weight, or transparency), or setting physical printer defaults (e.g., duplex versus simplex, or default medium selection)

- Network management configuration services, such as controlling domain access to or from the printer

- Troubleshooting or debugging services

Digital's implementation of console services on current PrintServer products conforms to the Enterprise Management Architecture.

Application Program Interface

The functional interface to any protocol provides an additional abstraction between an application and a protocol. This abstraction answers many of today's software application needs, including interoperability, portability, modularity, and reusability across multiple architectures. An application programming interface (API) that allows access to all CPAP facilities is included in the protocol's specification.

A connection block, which is passed as a parameter to all functions, provides support for various printer types, their device identifications, and descriptors for command and data channels. This support includes separate command and data channels for printers supporting multiple virtual circuits or channels. Just as in the case of the data-stream form of the protocol, the API form allows separate channels for data and commands.

A separate command channel allows ease of control flow between client and server. This may include the client receiving the server's status or events, or the client sending aborts to the server. For devices that support only a single channel, the generic printer driver can set both command and data channels to the same value. For supporting multiple jobs active at the same time (job overlap), a job identification (ID) parameter is passed with all functions.

To support various message types, the address of a read-callback routine is passed to the open printer function along with a pointer to read-callback arguments. These arguments may signal various events, or may consist of messages for the user, operator, accounting, or resources available in the printer.

An early version of the generic functional interface was part of MIT Project Athena's Palladium Print System. The printer supervisor in Digital's LN03R ScriptPrinter product was modified to create a generic printer interface for both the ScriptPrinter device and the PrintServer family. This conversion from an API-accessible base took one week to execute, whereas it typically takes six months of effort to develop a new printer supervisor for a device as complex as the PrintServer product.

4 The CPAP Server Implementation

The implementation of a protocol gives rise to problems different from those related to its design. When defining the architecture, one strives to provide an ideal that includes all of the desired features in an elegant manner. When performing an implementation, one finds that elegance often has to take a back seat to pragmatics. This is especially true when the new protocol is intended to replace two different protocols in a new version of an existing product. Merely implementing the new protocol is not enough-the implementation must somehow coexist with the protocols being replaced.

Digital's first production implementation of the CPAP was targeted for the DEC PrintServer Supporting Host software version 4.0, which loads and drives the PrintServer family of printers. For the balance of this paper, we refer to this software by the PrintServer product designation of LPS version 4.0.

We started the implementation by modifying Digital's ULTRIX PrintServer client, which already used the Reid-Kent subset of the CPAP, to use DECnet network transport and run on the VMS operating system. We then updated the LPS server code to permit either DECnet or TCP/IP transport. This was accomplished by using the direct-to-port communication features of the VAXELN operating system. The server establishes a circuit using the appropriate transport and then spawns a process for dealing with each incoming connection. Thus, the same code can service print sessions, management sessions, and console sessions without concern for the type of network transport.

The CPAP was, by design, directly upward-compatible with the Reid-Kent protocol subset. However, Digital's PrintServer offerings prior to LPS version 4.0 were LAPS-based, and LAPS was not CPAP-compatible. To permit users of existing PrintServer printers to continue to use these products, we had to find a way for the new CPAP implementation to coexist with the older LAPS application protocol. We achieved this coexistence by having the server perform translations from the older protocol to the new one in the server itself. When the client establishes the initial connection, the server senses which protocol is being used by the client system. If the initial message indicates the use of LAPS, the server spawns incoming and outgoing filters to deal with the incoming connection, and a new internal circuit replaces the network connection to handle the interpretation of the CPAP.

The coding of the LAPS filters was the last step in implementation before testing began. The PrintServer 20, PrintServer 40, PrintServer 40 plus, and the new turbo PrintServer 20 all had to be tested using both LAPS and the Reid-Kent subset of the CPAP. In addition, the new implementations of the management client and the console client on the VMS system required verification. This verification entailed a multitude of tests using the LPS symbiont running on older versions of the VMS operating system, the newer common print symbiont (CPS), several versions of the ULTRIX operating system, and a source kit version running on a Sun Microsystems workstation.

Unfortunately, this testing uncovered latent defects in the implementation of the existing products. We had to analyze each of these defects and plan corrective action. Since updating the existing products in the field is a difficult process (both technically and procedurally), we corrected most of the defects by altering the server to deal with the problems. Retesting was performed over several baselevels to ensure that our changes caused no regression.

At one of the early baselevels, the interface between the network distribution software and the server's PostScript interpreter was updated to use a stream-based connection in place of the previous packet protocol. This update permitted the new CPAP data channel to be mapped by reference to the input of the PostScript PDL or any other PDL supported by the printer. This change alone permitted the performance of the server to be maintained even when the server was translating from the old LAPS protocol to the CPAP.

In general, development proceeded incrementally, i.e., key features were identified and added with each baselevel. While this technique limits the complexity of producing the product, it raises an important business issue. Specifically, the provision of enhanced services in a client-server environment often exposes aspects of the proverbial "chicken-and-egg" situation. There is little call to offer enhanced features in a server if clients have not been programmed to solicit the features. However, clients are not readily upgraded to solicit features that might not be widely available.

The LPS version 4.0 project team met its backward compatibility design goals by including the LAPS-to-CPAP filters. In doing so, they undercut the need to provide the enhanced feature support that the CPAP was designed to deliver, since existing clients (earlier versions) could not avail themselves of the added features. In addition, the risks of including full CPAP support in LPS version 4.0 (possible increase in time to market, and the creation or exposure of more latent defects in all supported environments) seemed to outweigh the benefits. However, a last-minute change to use the new protocol's data channel for loading fonts yielded such a large performance advantage that resistance to using the new features crumbled, and the project team was allowed to submit the full protocol to field test.

5 Standardization

Network printing became widely available in the mid-1980s, but products from different vendors were not compatible. Network printing protocols were largely proprietary efforts by vendors who had developed them for their own printer products. Digital's PrintServer 40 and its LAPS protocol were typical in this regard. By the late 1980s, network printing was an established and competitive technology, but there was still little interoperability among the various vendors' products.

In the absence of printing protocol standards, the Internet Engineering Task Force (IETF) formed a Network Printing Protocol working group in early 1990. This group's charter was to examine printing protocols then in existence or under development, assess their applicability to Internet-wide use, and suggest changes. Digital's representatives to the IETF working group on the Palladium Printing Systems standardization reported the interest shown in Digital's Reid-Kent protocol. Thus, in July of 1990, Digital submitted a version of the PAP that was under consideration by the DECprint PAP architecture team.

Early consideration of this PAP by IETF and the LPS version 4.0 implementation effort ran concurrently. This provided a unique opportunity for Digital's implementers to obtain feedback from a very knowledgeable architectural community. In turn, they could report implementation experiences that affected the review and progress of the specification towards standardization. Implementations of CPAP clients and servers by companies other than Digital are in progress.

As part of Project Athena's Palladium Printing System, the CPAP has been accepted by the Open Software Foundation for inclusion in a future release of OSF/1.

A draft of the CPAP is being circulated among Internet members for comment. Meanwhile, work on future enhancements continues. Work is now in progress to specify a superset of the existing protocol that deals with authentication and encryption to strengthen security. This work is being done in the spirit of the original migration from the Reid-Kent protocol to the CPAP; i.e., the security features being added will not adversely impact users who do not need the new features.

6 Acknowledgements

The CPAP effort has been the work of many developers. Chris Kent and Brian Reid drafted the base architecture and created the first prototype implementations. Jim Jones championed the protocol in the DECprint PAP architecture team (Alan Guenther, Tom Hastings, Jim Jones, Tom Powers, and Eric Rosen) and coded the LPS version 4.0 server. Carol Gallagher wrote the LAPS filters to translate from the old protocol to the new. Mike Augeri and John McLain ported the management and console clients to the VMS system from the ULTRIX system. J.K. Martin rewrote the Berkeley Software Development (BSD) source kit to use the new protocol. Ajay Kachrani developed our ULTRIX and MIT Athena clients and represented the protocol during early phases of the IETF standardization effort. Many others supported these efforts, and others are yet beginning to develop new CPAP clients. We thank them all for their efforts.

7 Reference

1. R. Landau and A. Guenther, "Design of the DECprint Common Printer Supervisor for VMS Systems," Digital Technical Journal, vol. 3, no. 4 (Fall 1991, this issue):43-54.

James D. Jones Jim Jones is a principal engineer in the Hardcopy Systems Engineering Group. He joined Digital in 1974 and was part of a team developing diagnostic programs for the DECsystem-10 and DECSYSTEM-20 systems. After running his own software business for five years, Jim rejoined Digital to design printer controllers and software. Most recently, he provided software for the PrintServer products, authored the Common Printer Access

Protocol specification, and is helping to define the next generation of network printers. Jim is a member of IEEE and ACM and participates in the IETF.

Ajay P. Kachrani Principal software engineer Ajay Kachrani currently works on the OSF/1 socket and XTI kernel interfaces and security project. Previously, he led the development of the overall PrintServer software version 4.0 with dual network protocol support (DECnet and TCP/IP), from inception through field test. Ajay presented the CPAP protocol as an Internet standard to the IETF and added PrintServer support in version 1.0 of the Palladium Print System at MIT/Project Athena. Ajay holds a B.S.E.E. (honors) from the University of Mysore, India, and an M.S.C.S. from the University of Lowell.

Thomas E. Powers As a consultant engineer in the Hardcopy Engineering Firmware/Software Group, Tom Powers is a vendor liaison for desktop PostScript printer products. He chairs the DECprint PAP Architecture Team and was a contributor to the PrintServer 40 internal hardware/firmware architecture. Tom represented Digital on American and international standards committees on computer graphics from 1979 to 1989. He led several firmware teams and is coinventor of the ReGIS Graphics Protocol. Tom received his B.S.E.E. from Tufts University and holds an M.S.E.C.E. from the University of Massachusetts at Amherst.

DECnet, DECprint, Digital, LN03, PrintServer, ULTRIX, VAXELN, and VMS are trademarks of Digital Equipment Corporation.

PostScript is a registered trademark of Adobe Systems, Inc.

Open Software Foundation is a trademark and OSF and OSF/1 are registered trademarks of Open Software Foundation, Inc.

===

===