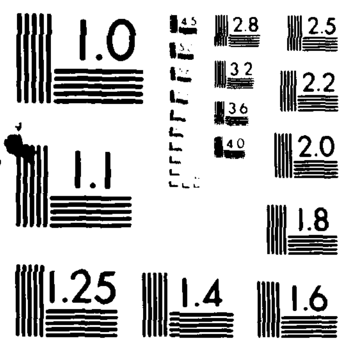
AD-A099 715 NAVAL OCEAN SYSTEMS CENTER SAN DIEGO CA F/G 9/2
COMMAND CENTER NETWORK PROTOCOLS SPECIFICATION.(U)
MAR 81 M A NEER
UNCLASSIFIED NOSC/TR-666 NL

NOSC

END
DATE
FILMED
7 81
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

LEVEL II

(12)

NOSC

NOSC TR 666

NOSC TR 666

AD A099715

Technical Report 666

Command Center Network Protocols SPECIFICATION

MA Neer

2 March 1981

Final Report for Period March 1979–March 1981

Prepared for
Naval Electronic Systems Command
NAVELEX 613

Approved for public release; distribution unlimited

NAVAL OCEAN SYSTEMS CENTER
SAN DIEGO, CALIFORNIA 92152

NAVAL OCEAN SYSTEMS CENTER, SAN DIEGO, CA 92152

AN ACTIVITY OF THE NAVAL MATERIAL COMMAND

SL GUILLE, CAPT, USN
Commander

HL BLOOD
Technical Director

ADMINISTRATIVE INFORMATION

Work was done by the C^2 Information Processing Branch (Code 8242) under PE62721N, Project F21241, Task XF21241100 (NOSC 824-CC70) for the Naval Electronic Systems Command (NAVELEX 613). This report covers work from March 1979 to March 1981 and was approved for publication 2 March 1981.

Released by
RC Kolb, Head
Tactical Command and
Control Division

Under authority of
JD Maynard, Head
Command Control—Electronic
Warfare Systems and
Technology Department

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NOSC Technical Report 666 (TR 666)	2. GOVT ACCESSION NO. AD-A099	3. RECIPIENT'S CATALOG NUMBER 725
4. TITLE (and Subtitle) Command Center Network Protocols SPECIFICATION		5. TYPE OF REPORT & PERIOD COVERED Final Rept. Mar[redacted]79—Mar[redacted]81
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) MA Neer		8. CONTRACT OR GRANT NUMBER(s) (12) 36
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Ocean Systems Center San Diego, CA 92152		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS PE 62721N Project F21241 Task XF21241100 (NOSC 824-CC70)
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Electronic Systems Command (NAVELEX 613) Washington, DC 20360		12. REPORT DATE 2 Mar[redacted]81
		13. NUMBER OF PAGES 34
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 14 NOSC/TR-666		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Command and control systems	Microcomputers
Command Center Network	Data links
Network Interface Units	Communication protocols

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A local network to give a commander flexible access to command and control subsystems is needed in the Navy. The Command Center Network is a proposed solution that would front-end Navy computers to a local data bus via microcomputers. These Network Interface Units (NIUs) would provide software to make the network transparent to each of the Navy subsystems.

This report describes the protocols necessary to carry out the functions described in a companion report, NOSC TR 665, Command Center Network Protocols—Functional Descriptions, by MA Neer, 2 March 1981.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

OBJECTIVE

Develop protocols which would allow for the transfer of command and control data over a local computer network in a distributed fashion.

RESULTS

Protocols were developed which allow the transfer of NAVMACS, Link 11, and terminal character strings.

RECOMMENDATION

Conduct experiments with the current protocols to see whether they fulfill the role envisioned for them.

PREFACE

A local network to give a commander flexible access to command and control subsystems is needed in the Navy. The Command Center Network (CCN) is a proposed solution which would front-end Navy computers to a local data bus via microcomputers. The microcomputers, or Network Interface Units (NIUs), would provide software to make the network transparent to each of the Navy subsystems.

A glossary of CCN acronyms and abbreviations is included at the end.

CONTENTS

ILLUSTRATIONS

1 INTRODUCTION

In recent years the Navy has initiated major efforts to provide user access to diverse subsystems which contain information of interest to the afloat commander. These efforts have been frustrated by the fact that these subsystems have been developed independently for specific functions and, as a consequence, are characterized by unique interfaces and protocols, fully committed memory and cpu cycles, complex software that is costly to modify, and an interface which expects an intelligent user on the other end. To address these issues, the Navy is developing a Command Center Network (CCN) for interconnecting these subsystems in a local environment such as a ship, building, or closely grouped set of buildings. The CCN builds upon recent developments in high speed data bus technology and protocols developed for the ARPANET. Microprocessors are used to "front end" these unique subsystems and to provide new protocols which facilitate C2 functions and process-to-process interactions without the requirement for an intelligent human user.

Figure 1 shows the protocol location to be employed in the initial CCN.

The protocols necessary to carry out the functions described in reference 1 are described in this report. For each C2 subsystem interfaced to the CCN, a set of user and server programs is defined. The server is a process in the NIU attached directly to the C2 subsystem, and the user is a process somewhere else on the CCN which requires interaction with the C2 subsystem.

The user and server programs will interface to the network via TCP4. (In this report TCP4 and TCP are equivalent protocols.) TCP4 is competely specified in reference 2, which describes this protocol competely.

2 INTERFACES

The user processes interface with a user (process or terminal) on one side and with TCP on the other. The server processes interface with the C2 subsystems on one side and with TCP on the other. In this section the user/TCP, server/TCP, and user/user process interfaces are discussed. The server/C2 subsystem interfaces are described in subsequent sections.

---

1. NOSC Technical Report 665, Command Center Network Protocols — Functional Descriptions, by MA Neer, 2 March 1981.
2. DOD Transmission Control Protocol, J Postel, ed; Defense Advanced Research Projects Agency, Information Processing Techniques Office, IEN 129, January 1980.

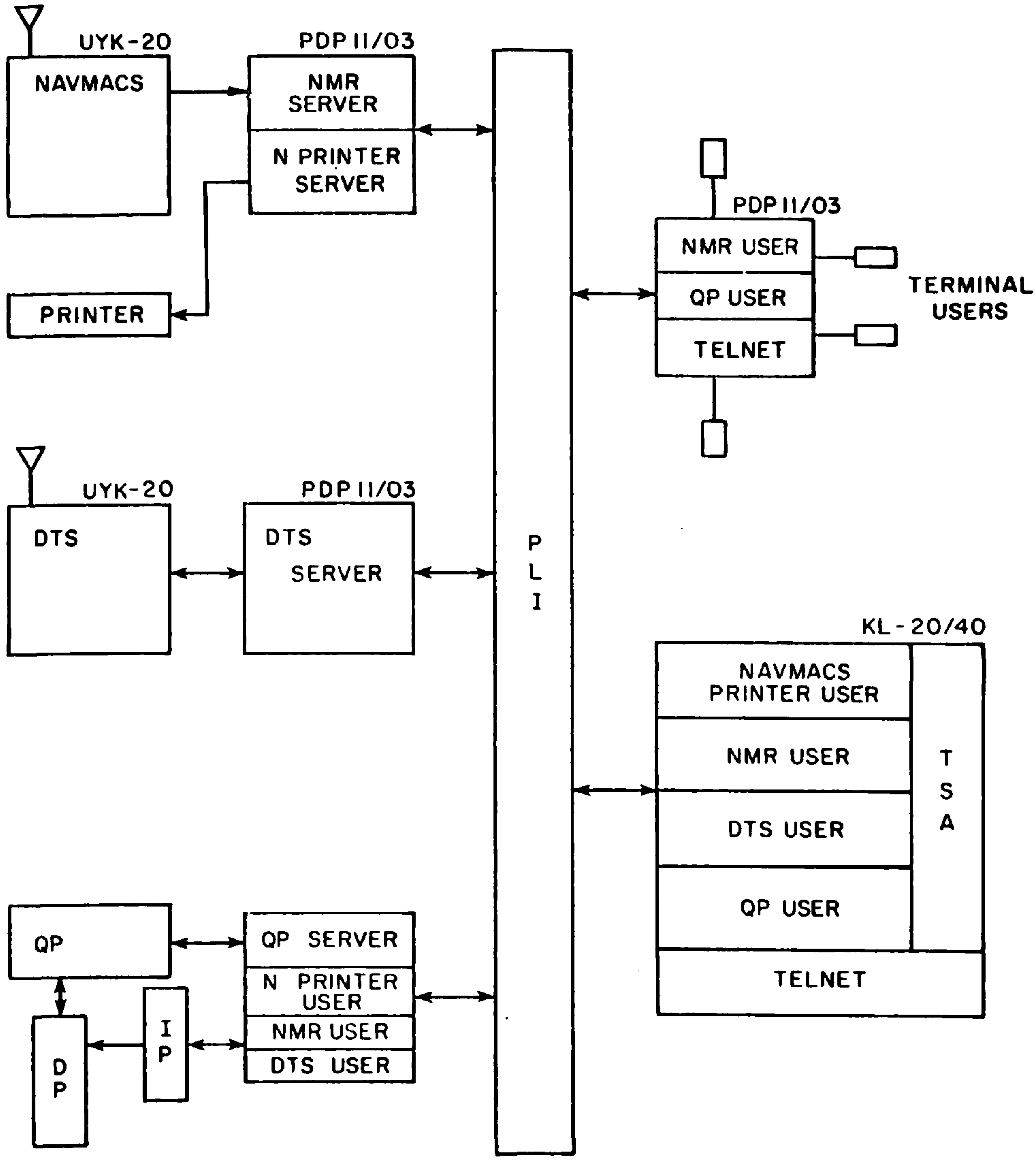


Figure 1. Protocol location in the initial CCN.

2.1 INTERFACE TO TCP

All TCPs must provide for the following minimum set of user calls:

OPEN(local port, foreign socket, active/passive). The terms port and socket are defined in the specification. If the indication is passive, this call is equivalent to LISTEN. If there is no foreign socket specified, the LISTEN would respond to any attempt by a foreign process to connect.

SEND(local connection name, buffer address). Here TCP is given the responsibility of delivering the named buffer to the destination. All errors due to transmission over the network are expected to be recovered by TCP.

RECEIVE(local connection name, buffer address). Here TCP provides a buffer for data arriving over the network.

CLOSE(local connection name). TCP will delete the connection.

It will be assumed throughout the following discussion that any TCP interfaced will provide for these four user calls.

In the initial CCN, SRI's TCP will run in the NIUs and TOPS20 TCP will run in the DEC 20/40. The interfaces to these particular implementations of TCP are completely described in references 3 and 4.

2.2 USER/USER PROCESS INTERFACE

For users on the CCN to take advantage of the services offered by the CCN user processes, the following signals are defined:

START. This is the signal that the user gives to the user process to initiate the connection to the C2 subsystem. The START signal will allow for the following parameters.

User name and password: for login purposes

File name: appropriate to the operating system being used. This file name will provide the user process the necessary information to store incoming data if the user so desires.

---

3. Terminal Interface Unit Notebook, by JE Mathis et al, Defense Advanced Research Projects Agency, May 1979.
4. TCP JSYS Calling Sequences, by W Plummer, of Bolt, Beranek, and Newman, Moulton MA, August 1979.

Filter field: contains an indication to the user process of data that will be used as a filter of incoming data.

Type field: specifies the type of data the user is interested in.

DONE. When the user is finished with the C2 subsystem, it issues this signal to close the TCP connection and terminate the user process.

CHANGE. This signal allows the user to change the filter or type specified in the START signal.

TRANSMIT. This signal will include a buffer address and byte count of data to send to the C2 subsystem.

RECEIVE. Indicates that the user is ready to process input from the C2 subsystem. If data are available when this signal is given, the user process will give a buffer to the user; otherwise, the request will be queued for processing on arrival of the data.

3 USER/SERVER SIGNALS

The user and server communicate with the following event signals:

PLEASE LOGIN. The server expects to receive a LOGIN message from the user within a timeout period.

LOGIN. A data packet containing all the necessary login information including user name, password, and filter (or message type).

ESTABLISHED. The LOGIN was accepted.

CONTROL. The control signal might be used to change filters or to inform the user of changing events such as printer ok or DTS ready.

ERROR. This can be sent at any time and will contain a field with an error code such as printer down, DTS down, printer out of paper.

4 INITIAL CCN AND TELNET

The initial CCN is the network which will perform a subset of the total functions desired in the long-range CCN. Throughout this document references are made to protocol activities which will not be implemented for the initial CCN. The functions which are not a part of the initial CCN were omitted primarily for simplicity's sake.

Terminal users on the CCN will have the Telnet protocol available for connecting their terminals to remote hosts. Telnet was originally designed for the ARPANET for the purpose of accessing interactive services available on time-sharing systems. In the CCN, SRI's Telnet developed for the Packet Radio Network will be used in the NIUs. This particular implementation is described in reference 3. This protocol will give terminal users on the CCN the ability to run processes like TSA on the KL-20/40.

5 NAVMACS PROTOCOLS

5.1 NAVMACS MESSAGE RECEIVING PROTOCOL

5.1.1 Introduction

The protocol described in this section describes the actions necessary to perform the following functions from reference 1:*

Deliver NAVMACS messages to terminal users on the CCN as well as processes like TSA and IP.

Allow a parameter indicating what kind of messages the user is interested in receiving. (The user, throughout this discussion, could be a process, a terminal, or a printer.) [In the initial CCN the only types permitted will be "all" and "RAINFORM".]

[Require a user to login or a process to authenticate itself.]

Signal a user when NAVMACS messages arrive.

[Allow a user to file messages for later retrieval.]

Allow a user to stop the process at any time.

[Inform the user of net errors which result in loss of messages.]

---

*The brackets [] enclose those functions which will not be implemented in the initial CCN.

Convert baudot to ASCII.

Employ a multiaddressing scheme to deliver the same NAVMACS message to several users.

Send each NAVMACS message to the NAVMACS TT-624 line printer. (The line printer is being shared with other processes on the CCN.)

[Allow the NSM to have NAVMACS messages sent to third parties.] (The NSM can arbitrarily decide that a process or terminal on the CCN should receive certain NAVMACS messages.)

[Filter messages based on subject or headers.]

[Print only the headers of messages for the user.]

[Inform the user when the NAVMACS processor is being held off.] (The processor is held off whenever buffer space is a problem or hardware is malfunctioning.)

[Allow a terminal user to direct NAVMACS messages to a third party.]

[Convert RAINFORM formatted messages to CCN format.]

The discussion will mention in bracketed statements those functions which will not be implemented in the initial CCN.

Note: The third-party transfer functions are not described in this report but will be added later.

5.1.2 Server

This process reads and distributes messages the NAVMACS processor normally sends to its printer. The messages are distributed to all interested users on the CCN and may be filtered on the basis of message type or content. [For the initial CCN, messages will not be filtered on content and the only types allowed will be "all" or "RAINFORM".] The messages are also sent to the NAVMACS printer. Users connect to this process via the user program described in the next section. The server program maintains a well-known socket via a LISTEN call to TCP. When TCP informs this process that an attempt has been made by a foreign process to connect to the NAVMACS processor, this process will send a PLEASE LOGIN signal to the user process. [In the initial CCN, no user name or password will be necessary but a TYPE parameter will be accepted that is limited to two types: "all" and "RAINFORM".] If this is successful – ie if the user is indeed a legitimate user – this process will establish the type of traffic the user is interested in seeing. If the user failed to login, the connection will be closed. If the NAVMACS processor is being held off due to lack of buffer space, the server will return a signal to the user (CONTROL). [This will not take place in the initial CCN.] Once the LOGIN has been received successfully it will be evaluated to see whether it is legitimate. If it is not, an error (not logged in) will be returned to the user and the connection will be closed. [Again, the LOGIN procedure described herein will not be part of the initial CCN implementation.] If the LOGIN is acceptable, an entry will be made in a connection table showing the following information:

This user is connected.

The TCP connection name.

The type of traffic the user is interested in.

A parameter to filter on - for example, a subject or header.

An on/off indicator.

An ESTABLISHED indication will be returned once the connection entry is placed into the connection table.

As each message is received from NAVMACS and before the message is distributed, the server process will do some preliminary manipulating. At first, the process will delimit the NAVMACS message by looking for the SOM-EOM characters. Once these characters are found, the NAVMACS processor will be held off by the dropping of the ready line to a low voltage. At this time the server process will attempt to send the message to the printer. The printer is being shared, so it is possible that it is busy with text from some other CCN user. If the printer is busy, the server process will attempt to store this message for the printer process retrieval later. If no storage device is available, the server process will hold the message with the NAVMACS processor held off until the printer becomes available. [In the initial CCN, there will be no mass storage capability.] The user will be informed via a CONTROL of the state of the NAVMACS processor [but not in the initial CCN]. The server process will then convert the NAVMACS message from baudot to ASCII and will start searching the connection table for interested users. If the message is RAINFORM formatted it will be converted to CCN format [but not in the initial CCN]. The CCN format is a standard tactical data format that makes information of this kind available to all CCN users. This format will be defined so that users who want to make use of, for example, DTS Link 11 data can do so by recognizing the data format. Each connection will be examined in turn to see which ones have interest in the current message. If a connection is interested, the server process will turn it on by making an entry into the connection table. Once the connection table has been searched, the message processing is finished.

The message will then be sent via TCP to each connection which is turned on. TCP will indicate the success/failure of the sent text. If a message cannot be delivered to a user, the user will be informed of the mission message by a CONTROL signal [not to be implemented in the initial CCN]. Once TCP has accepted the message for transmission, the buffer space will be reused by having the NAVMACS processor turn back on (by a rise of voltage on the ready line high). The connection table will be reinitialized; ie all connections will be turned off.

This process will respond to a CHANGE by updating the connection table entry for this user.

If a CLOSE is received from TCP, the connection table will be updated by deleting the entry for that connection.

5.1.3 User

This process will supply NAVMACS messages of a specified type to a user process in the CCN. The user must supply a START signal to this process to initiate the connection to the server. The user must supply a TYPE parameter indicating the type of messages it wants to receive. [In the initial CCN the only types will be "all" and "RAINFORM".] The user will be able to have the messages filtered on subject or header [but not in the initial CCN]. The user process should establish that the TYPE given is valid. The user should supply a parameter indicating whether the messages are to be delivered to a specified user buffer or filed on mass storage. [In the initial CCN, the ability to store messages will be confined to user processes running on the KL-20/40.] The buffer size at this time is to be on the order of 2 bytes, allowing enough room for an average-sized NAVMACS message. Once the server process responds to the attempt to connect (by requesting a login), this process will send the necessary login information to the server via TCP. [In the initial CCN, there will be no requirement for user name or password.] This process will keep a status indication which reflects the state of the process at any given time (OPEN issued, login sent, etc). Once the connection is established (the LOGIN was accepted), this process will inform the user and supply a buffer for incoming messages.

Once messages arrive, they will be filed away or given to the user if so indicated. In either case the user will be signalled when the messages arrive. The user will also be informed if the server has indicated that the NAVMACS processor is being held off or messages were lost via CONTROL signals [but not in the initial CCN]. This process will handle error messages sent by the server and will respond to a user signal to stop, at which time it will issue a CLOSE to TCP.

5.1.4 Terminal Users

Terminal users will be asked to indicate what kind of NAVMACS messages they are interested in. [In the initial CCN, the only types will be "all" or "RAINFORM".] This process will establish that the type is legitimate. If the type is not legitimate, the user process will return an error indication to the user. If the type is legitimate, this process will then connect to the server via TCP. This process will keep a connection status indicating the successive states:

OPEN sent

LOGIN sent

connection established

etc

Once the connection is made, this process will log the user in and establish the type of messages desired. Once the connection is established, this process will supply a buffer for incoming messages. At this time the buffer size is on the order of 2 bytes. Once messages arrive, the user will be signalled of a newly arrived message. The terminal user can then ask for

The message to be printed.

The header to be printed.

The message to be filed away.

[In the initial CCN, the user will have only the option: print the message.] A new buffer will be allocated and a new NAVMACS message solicited. The user may change the type of messages being delivered. The user process will send CONTROL(type) with the new type to the server. The user process will inform the user if the NAVMACS processor is being held off or messages were lost. The terminal user can stop the process at any time via the DONE signal.

5.1.5 User/Server Interaction

Here is an example of a simple user/server session:

User	Server
Idle	LISTEN
OPEN to NAVMACS →	
	← PLEASE LOGIN
LOGIN →	
	Login correct ← Connection ESTABLISHED
Established	
	⇇ NAVMACS messages
CLOSE →	LISTEN

5.1.6 NMR Protocol Location In The Initial CCN

Figure 2 shows the location of the protocols (user/server) in the initial CCN.

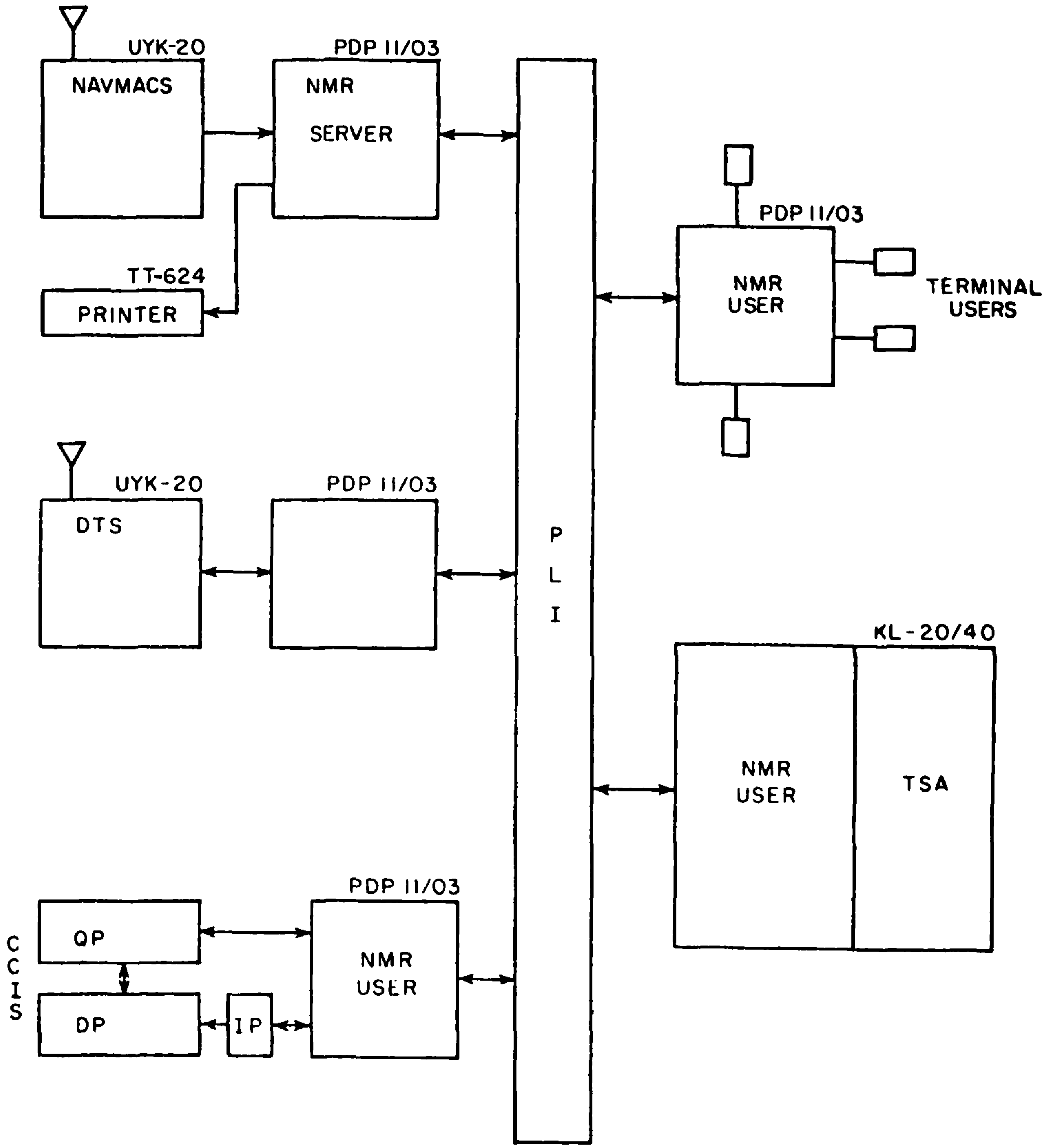


Figure 2. NMR protocol location in the CCN.

5.2 NAVMACS PRINTER PROTOCOL

5.2.1 Introduction

The protocol described in this section will perform the following functions described in reference 1:*

Print NAVMACS messages on the TT-624.

Direct user output on the CCN to the TT-624.

[Store user text for printing later if the printer is busy.]

Keep user text separate from other users.

Convert ASCII characters to baudot for the printer.

Prevent users from tying up the printer.

[Employ a priority scheme in granting access to the printer.]

Inform the users of success/failure of printed text.

[Require users to login.]

Advise users when the printer is not available.

Inform users when the printer buffer space is exhausted.

The functions that won't be performed in the initial CCN are mentioned in bracketed statements in the following discussion.

5.2.2 Server

The server process allows foreign processes to send text to the TT-624 line printer. The server process, when not being employed by a user on the CCN, will provide a buffer for the NAVMACS Message Receiving Protocol to use for NAVMACS messages. This buffer will be the same size as an average NAVMACS message: 2k bytes. This buffer will be sent to the line printer. Whenever the buffer has been completely printed, the server will return a count of characters printed to the user process in a CONTROL signal. This can be used to indicate the fact that the text was successfully printed. When a user on the CCN connects to this process, the NAVMACS Message Receiving Program will no longer be given this buffer. If text is in the buffer at the time of connection, it will be spooled in its entirety before the buffer is realeased or assigned to another user. This process will return a PLEASE LOGIN signal if the printer is available. [In the initial CCN, there will be no user name and password.] If the printer is already being used by some CCN user, an ERROR(busy) signal will be returned (if this user has a lower priority than the current user). [In the initial CCN, no priority scheme will be employed.]

*The brackets [] enclose those functions which will not be implemented in the initial CCN.

The server will queue a list of attempts to connect in prioritized order. When the printer becomes free, this list will be searched to see who gets the next PLEASE LOGIN. After the user has successfully logged in, it may send text to be printed. After the PLEASE LOGIN is sent, this process will wait a timeout period during which the user may send text to be printed. The server will generate a "form feed" character to separate users' text. If text does not arrive during the timeout period, a CLOSE will be issued to TCP and the printer freed for either NAVMACS messages or other CCN users. The timeout period will be adjustable to meet changing demands but may initially be on the order of 2 minutes. At this time the user-sent ASCII text will be converted to baudot for the printer. When the text has been printed, the server will send a character count to the user and ask TCP for more user text. If TCP returns a CLOSED, this will indicate that the user has closed the connection so that the printer can be freed for other users. If the printer malfunctions or runs out of paper, an ERROR(printer down) signal will be returned to the user.

5.2.3 User

The user process will allow other processes on the CCN to direct their output to the NAVMACS TT-624 line printer. The user process must be passed the address and byte count of a buffer of text to be printed. The user process will then ask TCP to connect to the NAVMACS processor. The user process will then wait for a signal (either ERROR(busy) or PLEASE LOGIN) to be returned from the server. If this signal is not returned or if a BUSY is returned, this process will store the text on mass storage and solicit more text from the user. [In the initial CCN, there will be no mass storage capability.] This process will then continue storing text until the user issues a DONE signal. When the user receives the PLEASE LOGIN signal from the server, the user process will send the login information and then wait for an ESTABLISHED signal to be returned from the server. When the user process receives the ESTABLISHED from the printer server, it will use TCP to send all stored text to the printer. The user can supply text continually until it tells the user process to stop (via a DONE signal), at which time a CLOSE will be given to TCP. The user process will keep a byte count of all text which is sent to the printer. As CONTROL words are returned from the server containing character counts, the user process will consider that number of characters as acknowledged.

5.2.4 User/Server Interaction

The following table depicts normal protocol interaction to get user text printed on the TT-624:

User	Server
OPEN PRINTER →	LISTEN
	← ERROR(busy)
(start storing user text) (State=busy)	
	← PLEASE LOGIN
LOGIN →	Verify login
	← ESTABLISHED
Text →	
	← ERROR(printer down)
(State=wait)	
	← CONTROL(go-ahead)
Text →	
CLOSE →	LISTEN

5.2.5 NAVMACS Printer Protocol Location In The Initial CCN

Figure 3 shows the location of the protocols (user/server) in the initial CCN.

6 DTS PROTOCOL

6.1 Introduction

The protocol described in this section describes the actions necessary to perform the following functions from reference 1:*

Deliver track data from the DTS computer to interested users.

[Deliver track data from users to the DTS computer.]

*The brackets [] enclose those functions which will not be implemented in the initial CCN.

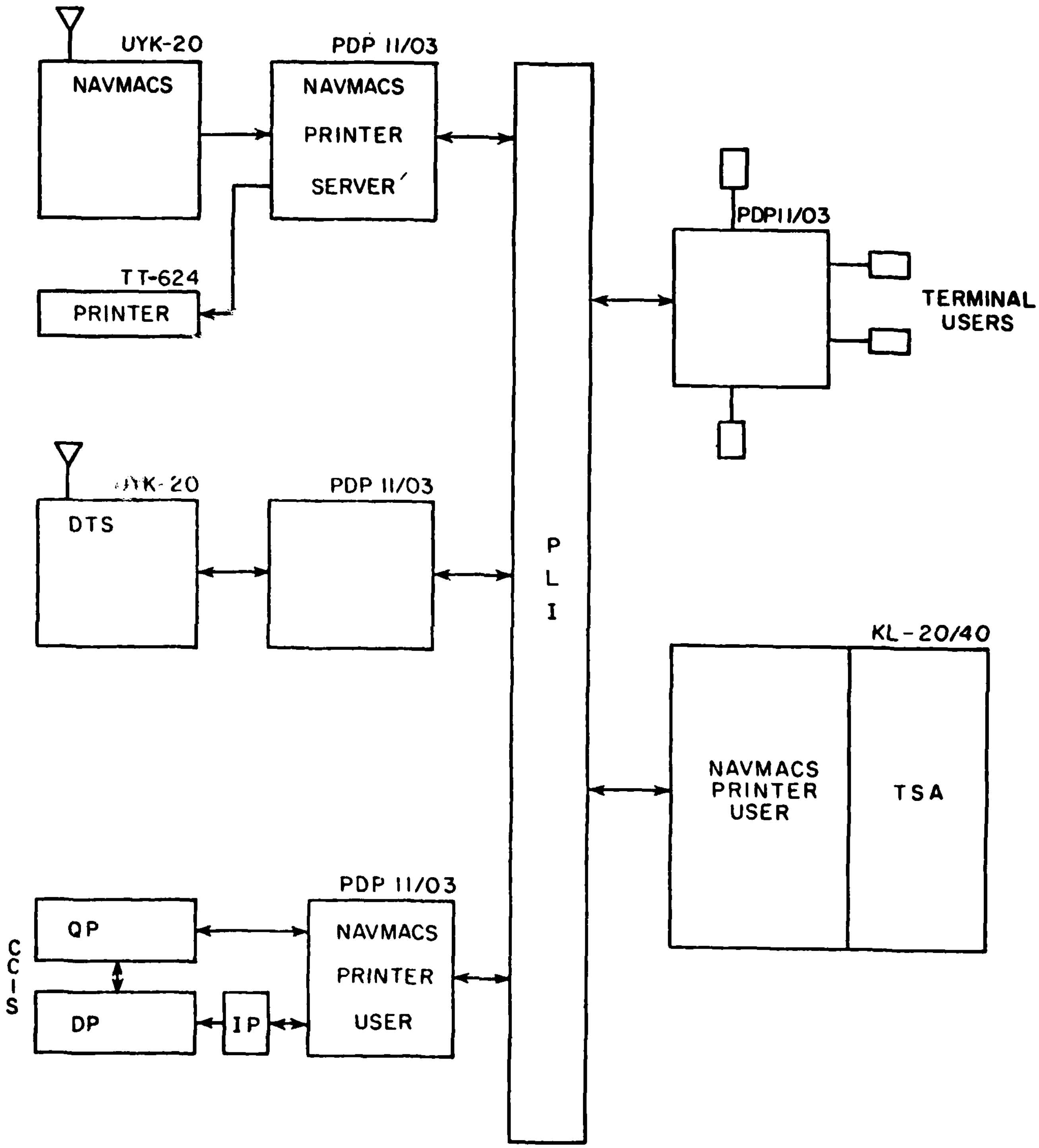


Figure 3. NAVMACS printer protocol location in the CCN.

[Require users to login or processes to identify themselves.]

Prevent transmission over CCN of track reports containing no change in data fields.

[Deliver tracks based on content — air tracks to some users, surface tracks to others, etc.]

Signal the user when tracks arrive from the DTS computer.

Employ a multiaddressing scheme in order to deliver the same tracks to several users.

[Inform users of success/failure of tracks sent to the DTS computer.]

Convert track data from binary to ASCII.

[Store track data for later retrieval.]

[Allow NSM to have tracks sent to a third party and filter on subject or content; ie the NSM can change the addressee list.] (For the purpose of insuring that certain processes on the CCN get all air track information or surface track information, etc.)

[Convert ASCII/binary.]

[Convert to/from CCN format.]

[Allow an option to disable the default of receiving all tracks and receive only certain tracks based on some filter.]

The discussion will mention in bracketed statements those functions which will not be implemented in the initial CCN.

Note: The NSM will not be a part of the initial CCN, and the third-party transfers are yet to be described.

6.2 Server

This process reads and distributes Link 11 track data which the DTS computer normally sends to an NTDS computer. The tracks are sent to all interested CCN users and may be filtered on content. [In the initial CCN, no content filtering of tracks will be performed.] Users connect to this process via the user process described in the next section. The server process maintains a well-known socket via a LISTEN call to TCP. When the user process establishes a connection with this socket, the server process will request that the user login. [In the initial CCN, no name and password will be required.] A PLEASE LOGIN signal will be sent to the user, and the user should respond with a LOGIN message containing the necessary information. If the user fails to login or the login information is not valid, the server will close the connection. Included in the LOGIN message will be an indication of the user's desire to filter tracks. The filter may be based on content or track number. [In the initial CCN there will be no opportunity to filter tracks.] If the LOGIN is acceptable, an ESTABLISHED signal will be returned to the user process. A data structure will be maintained by the server consisting of one entry per connection containing information like the following:

User name

TCP connection name

Filter/no filter indicator

Parameter to filter on

On/off indicator

As tracks arrive from the DTS, the server process will record the track numbers (and any other necessary information) in order that it may prevent transmission over CCN of track reports containing no change in the data fields. Because of the nature of the data receiving process, truncation of track data may occur before the duplicate is detected. If the newly arrived track is not a duplicate, a routine will be invoked to convert the binary track data into CCN ASCII format. This format will be the standard CCN format for tactical data, which will allow processes anywhere on the CCN to recognize and make use of the information. Once this conversion is done, the connection table will be scanned and each connection turned on/off depending on the appropriateness of the track. If a connection indicates some filter, the track will be analyzed to see if the information is pertinent to that user. [In the initial CCN, this will not be done.] The track will then be sent to all connections that were turned on during the scan. The connections will then be turned off and the process will repeat. If TCP closes the connection, the entry in the connection table will be deleted.

While connected, the user will be able to send signals to the server process. One signal will be a CONTROL(filter) signal so that the user can change the filter information on its connection. The server process will respond by inserting the new information into the appropriate entry in the connection table. The user may also send new track information on the DTS computer. [In the initial CCN, this will not be implemented.] The track data will arrive in CCN ASCII format, and the server process will convert the data to the binary format required by the DTS computer. The converted data will be sent to the DTS computer. CONTROL signals will be returned to the user process to indicate the success/failure of delivering the data to the DTS.

6.3 User

This process will supply Link 11 track data to users on the CCN obtained from the DTS computer. The user must supply a START signal to this process to initiate the connection to the server process. Included with this signal will be an indication of whether the user wants the data delivered to a buffer or to mass storage. [In the initial CCN, only the KL-20/40 will have mass storage.] The user can supply data to filter the tracks on if he so desires. [In the initial CCN, no filter will be allowed.] The user process will issue an OPEN to TCP and wait for a request to login from the server process. [In the initial CCN, no user name and password will be required.] The user process will respond to the login information, including the filter parameters if there are any. The user process will then wait for an ESTABLISHED signal from the server process before forwarding it to the user. When track data arrive, the user process will store them in a file if the user desires or will deliver the data to a specified buffer. In either case the user will be signalled that Link 11 track data have been delivered.

The user can supply signals to the user process at any time, indicating the following:

It wishes to stop the process (DONE).

It wishes to change the filter (CONTROL).

It wants to send track data to the DTS (TRANSMIT).

[In the initial CCN, only the DONE will be allowed.]

If the user wishes to stop the process, the user process will issue a CLOSE to TCP and await TCP's CLOSED signal before exiting. If the user wishes to change the filter, a CONTROL signal will be sent to the server process along with the new filter data. If the user wishes to send track data, it must issue a TRANSMIT along with the buffer address and byte count. It is the user's responsibility to insure that data are in CCN ASCII format. Once data have been sent, the user process will keep an indication that it is awaiting acknowledgment for the data. CONTROL signals will come from the server, indicating the success/failure of delivering the tracks to the DTS.

6.4 User/Server Interaction

Here is an example of a simple user/server session:

User	Server
IDLE →	LISTEN
	← PLEASE LOGIN
LOGIN →	
	← ESTABLISHED
(Established)	
	⇇ NTDS data
SEND →	
data →	(deliver data to DTS)
	← GO-AHEAD
CLOSE →	LISTEN

6.5 DTS Protocol Location In The Initial CCN

Figure 4 shows the location of the protocols (user/server) in the initial CCN.

7 CCIS PROGRAMS

7.1 INTRODUCTION

The protocol described in this section describes the actions necessary to perform the following functions from reference 1:*

Serve as the IP by delivering NAVMACS RAINFORM messages and DTS tracks to the DP. (The user processes which deliver this information are described in 2.1.3 and 3.3.)

Format the RAINFORM messages and DTS tracks appropriately for the DP.

Allows users to query the QP and return responses to them.

Allow users to send responses to the TT-624 printer. (The user process which sends text to the printer is described in 2.2.3.)

[Establish a priority scheme for query users.]

[Require users to login.]

[Store queries/responses if printer is busy.]

[Queue queries if QP is busy.]

[Queue messages in IP for DP.]

Send queries/responses to both the printer and the user.

The discussion will mention in bracketed statements those functions which will not be implemented in the initial CCN.

7.2 INTERFACE PROCESSOR PROGRAM (IFP)

In order for the CCIS Programs to deliver NAVMACS RAINFORM and DTS messages to the DP, the user programs described under the NAVMACS Message Receiving Programs and the DTS Programs will be used in the CCIS NIU by an IFP Program. This program will employ the user programs and will provide a buffer for delivery of RAINFORM messages and DTS track data. The buffer will be on the order of 2 bytes long. When Link 11 track data arrive as signalled by the

*The brackets [] enclose those functions which will not be implemented in the initial CCN.

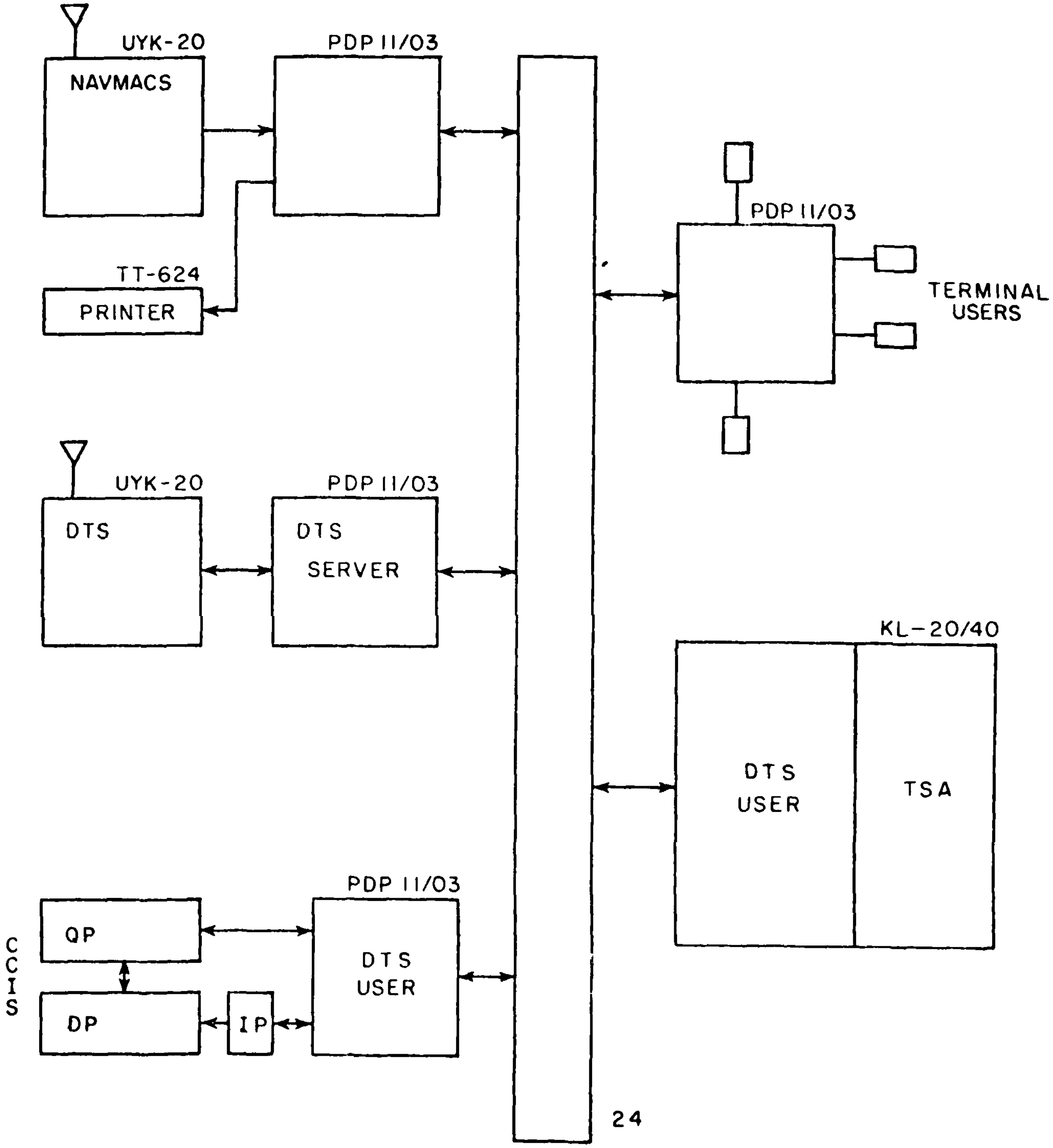


Figure 4. DTS protocol location.

DTS user program, the IFP will convert the data to the format the DP expects and send it to the DP. The NMR and DTS user processes will be prevented from delivering additional tracks or RAINFORM messages while the data are being sent to the DP. Likewise, if a RAINFORM message arrives, the NMR and DTS user programs will be prevented from delivering any additional messages until the current message is sent to the DP. Thus, it is the responsibility of the NMR and DTS user programs to store messages until the IFP is ready. Storage could be on mass storage or in core. [In the initial CCN, only one message at a time will be processed.] If the user program attempts to deliver new messages and the IFP is busy, it is the IFP's responsibility to inform the user process when it is ready to accept more data or to retrieve the information from a file if mass storage is available.

7.3 QP PROGRAMS

7.3.1 Server

The server process will deliver queries sent by a CCN user to the QP and return the query/response to the user or optionally to the NAVMACS printer. This process will maintain a well-known socket for users to connect to via TCP. When a connection is attempted, the server will respond with a PLEASE LOGIN request. The user should then supply a LOGIN message consisting of a user name and password [not required in the initial CCN] and an indication of whether the user wants the queries and responses sent to the printer, the user, or both. If the LOGIN is acceptable, the server will mark this connection as established and return an indication to the user. The user may then send queries to the server. When a query is received, the server will check to see if the user has logged in and then put the query in the queue for the QP. [In the initial CCN, one query/response will be processed at a time.] The position in the queue will be based on the priority of the user. [In the initial CCN, no priorities will be assigned.] Normally queries will be processed FIFO(first in, first out). Both the query and the "owner" will be stored in the queue. When responses come back from the QP, they will be matched with the query at the front of the queue. Care must be taken to insure that a query which is being processed by the QP is not disturbed from its position at the front of the queue. When the response returns, the connection will be checked to see whether the response should be sent to the line printer, the user, or both. If the response is to go to the NAVMACS printer, the NAVMACS Printer Program user process will be invoked and will receive the query and response to deliver to the printer. (See 2.2.3.) It is the printer program's responsibility from that point on to deliver the query/response to the line printer. The QP server process will send the next query in the queue to the QP. A CLOSED from TCP will tell the server process when the user has finished with the QP. The connection will be deleted and the server will return to the LISTEN state. If the response is going to the user, it will be sent along with the query.

7.3.2 User

This process will have the responsibility for accepting and delivering queries from users (processes or terminal users) on the CCN to the QP in the CCIS system. Upon receiving a signal to start up, this process will issue an OPEN to QP via TCP. This process will accept a parameter from the user, allowing the user to select the NAVMACS printer as the output device. The user will have the option of having the responses returned to him, sent to the printer, or both. When the server responds with a PLEASE LOGIN request, this process will return a LOGIN message containing user name and password [not required in the initial CCN] and the printer select parameter. This process will then await an ESTABLISHED signal from the server to see whether the login was accepted. When the ESTABLISHED signal is received, this process will inform the user that it is now permissible to submit queries to the QP. If the user has indicated that the responses are to be sent to the printer only, the user process forgets about the query. If the responses are returning to the user, a buffer will be reserved for the returning responses. (The buffer is about 2k bytes.) An event signal from the user process will tell the user when a response has been returned. The user process records each query sent and each response returned in order to tell when there are outstanding queries. This process will try to match each query with a response before exiting or closing the connection. The user can signal this process at any time to close the connection.

7.3.3 User/Server Interaction

The following is an example of a simple user/server session:

User	Server
Idle	LISTEN
OPEN TO QP →	
	← PLEASE LOGIN
LOGIN →	
	← ESTABLISHED
query →	(put query on queue)
query →	(put query on queue)
	← query/response
CLOSE →	LISTEN

7.4 CCIS PROTOCOL LOCATION IN THE INITIAL CCN

Figure 5 shows the location of the protocols(user/server) in the initial CCN.

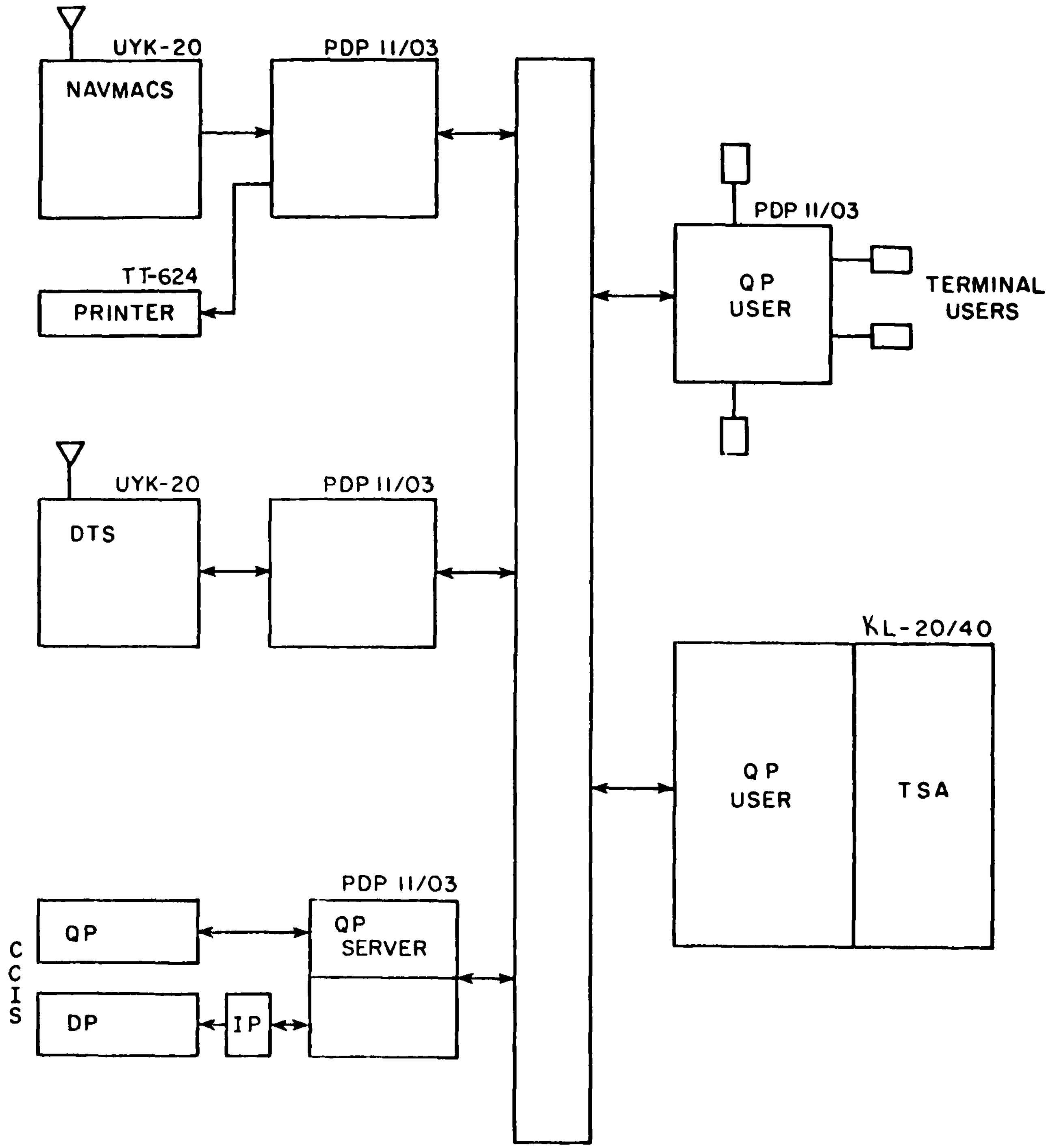


Figure 5. CCIS protocol location in the CCN.

8 TSA PROTOCOLS

Since the protocols applicable to TSA have been described elsewhere in this report, the section numbers of the description are given after each function. The protocols that apply to TSA fulfill the following functions from reference 1:*

Deliver NAVMACS messages to the KL-20/40 for TSA. (2.1.3)

Allow a parameter indicating what kinds of NAVMACS messages TSA is interested in. (2.1.3)

[Require TSA to authenticate itself as a NAVMACS user, NAVMACS Printer user, DTS user, and QP user.] (2.1.3, 2.2.3, 3.3, 4.3.2)

Signal TSA when NAVMACS messages arrive. (2.1.3)

Allow TSA to file the NAVMACS messages for later retrieval. (2.1.3)

Allow TSA to stop the process at any time. (2.1.3)

[Inform TSA of net errors which result in the loss of NAVMACS messages.] (2.1.3)

[Inform TSA when the NAVMACS processor is being held off.] (2.1.3)

Allow TSA to send text to the NAVMACS TT-624 line printer. (2.2.3)

Allow TSA to file text for later printing if the line printer is busy. (2.2.3)

Keep TSA's text separate from other users' text. (2.2.2)

Prevent TSA from tying up the line printer. (2.2.2)

Inform TSA of success/failure of printer text. (2.2.3)

Deliver DTS track data to TSA. (3.3)

[Send track data from TSA to DTS.] (3.3)

Signal TSA when track data arrive. (3.3)

Allow TSA to store tracks for later retrieval. (3.3)

[Inform TSA of success/failure of sent tracks.] (3.3)

Allow TSA to query the QP. (4.3.2)

---

*The brackets [] enclose those functions which will not be implemented in the initial CCN.

Allow TSA to direct the query/response from the QP to the line printer. (4.3.2)

[Filter on subject or header in NAVMACS messages.] (2.1.3)

[Store the queries/responses sent to the QP for later retrieval if the printer is busy.] (2.2.3)

[Send tracks from TSA to the DP.] (Not described in this report at present; however, will be added later.)

9 TERMINAL USERS PROTOCOLS

Telnet and the user protocols described in this report are available to terminal users. Since these protocols have already been addressed in this report, the following discussion will reference sections from this report that are applicable to the named function.

The protocols in the CCN will perform the following functions from reference 1:*

Deliver NAVMACS messages to terminal users. (2.1.4)

Accept a parameter defining the type of NAVMACS messages the user is interested in. (2.1.4)

[Require the user to login.] (2.1.4)

Signal the user when NAVMACS messages arrive. (2.1.4)

[Allow the user to file the NAVMACS messages for later retrieval.] (2.1.4)

Allow the user to stop the process at any time. (2.1.4)

[Inform the user of net errors which result in the loss of NAVMACS messages.] (2.1.4)

[Print only the headers of NAVMACS messages.] (2.1.4)

[Inform the user when the NAVMACS processor is being held off.] (2.1.4)

Allow the user to query the QP. (4.3.2)

Allow the user to direct queries/responses from the QP to the NAVMACS line printer. (4.3.2)

*The brackets [] enclose those functions which will not be implemented in the initial CCN.

[Store queries/responses for later printing if the TT-624 is busy.] (2.2.3)

Allow the user to run TSA. (1.4)

[Allow a user to have NAVMACS messages sent to a third party.] (Not described in this report but will be later.)

Allow users to both receive NAVMACS messages and send them to the printer. (This is a third-party transfer.)

10 NETWORK SERVICES MANAGER PROTOCOLS

Since the Network Services Manager (NSM) will not be a part of the initial CCN, all of the functions associated with it are to be implemented in the future. The NSM is anticipated to perform many functions to relieve the burden of processing from the NIUs. At present, however, it is not possible to define all the functions it may perform. Some of the functions that the protocols will provide are as follows:

Have NAVMACS and DTS data sent to third parties.

Have NAVMACS and DTS data filtered on content.

11 JUSTIFICATION FOR THE CCN PROTOCOLS

In the CCN, the C2 subsystems will be interfaced to a network in order that information can be shared and made available to a user (human) for the exercise of command and control. The C2 subsystems are to be front-ended; as a result, the C2 subsystems will not be altered, yet this information will still be made available to CCN users. Since it is desirable to make information generally available in the CCN, the protocols divide naturally into server and user, where the server provides a service (access to the C2 subsystem's data) to the user (human at a terminal or process) anywhere on the CCN. Thus the user initiates the action, since the server is waiting for any CCN user to request service. The server is capable of servicing multiple users who will have the ability to start and stop the process of accessing the C2 subsystem's data asynchronously. This design is more flexible than having the server initiate the action, since new users can be incorporated into the CCN with little or no modification to existing software. The user in this design is given more control since he is free to start and stop the process at any time. The synchronous exchange for the protocol functions PLEASE LOGIN, LOGIN, and ESTABLISHED is justified by the simplicity they bring to the protocol and the anticipated low delay in a high bandwidth local network. In general, simplicity in the protocols is a desirable feature in a local network.

The advantage of PLEASE LOGIN over merely sending the LOGIN is that the server has the option of accepting the connection from TCP but denying service (by returning BUSY) until it is ready to give it.

None of the C2 protocols has a mechanism for flow control other than to declare the C2 system inoperative since TCP provides that function. It is felt that, in general, functions should not be duplicated in layered protocols.

REFERENCES

1. NOSC Technical Report 665, Command Center Network Protocols — Functional Descriptions, by MA Neer, 2 March 1981.

2. DOD Transmission Control Protocol, J Postel, ed; Defense Advanced Research Projects Agency, Information Processing Techniques Office, IEN 129, January 1980.

3. Terminal Interface Unit Notebook, by JE Mathis et al, Defense Advanced Research Projects Agency, May 1979.

4. TCP JSYS Calling Sequences, by W Plummer, of Bolt, Beranek, and Newman, Moulton MA, August 1979.

GLOSSARY OF CCN ACRONYMS AND ABBREVIATIONS

ARPANET	Advanced Research Projects Agency Network
ASCII	American Standard Code for Information Interchange — a seven-bit code used to represent 128 symbols. In the CCN, the ASCII format used is that code that is described in the ARPANET Protocol Handbook (NIC 7104), pages 251-263, and called USASCII.
Baudot	Seven-level International Telegraph Code 2 using one start bit, one stop bit, and five bits representing one of 62 symbols.
Bit	Binary digit
Byte	Eight-bit word
C^2	Command Control
CCIS	Command Center Information Subsystem
CCN	Command Center Network
cpu	Central processing unit

DEC	Digital Equipment Corporation
DP	Data Processor (one of three processors used in the CCIS)
DTS	Data Terminal Set
EOM	End of message
IMP	Interface Message Processor
IP	Interface Processor (one of three processors used in the CCIS)
kbaud	One thousand bits per second
LSI-11	Digital Equipment Corporation processor model
LSI-11/03	Digital Equipment Corporation processor model
MOS	Micro Operating System
NAVMACS	Naval Modular Automated Communications System
NIU	Network Interface Unit
NSM	Network Service Manager
NTDS	Navy Tactical Data System
NTDS Slow	Interface requirements specified in MIL-STD-1397
PDP-11/03	Digital Equipment Corporation computer model built on an LSI-11/03
PLI	Private Line Interface
QP	Query Processor (one of three processors used in the CCIS)
RAM	Random access memory
RS-232	EIA standard electrical interface defining control lines, voltage levels and signals for exchange of binary data.
SOM	Start of message
SRI	Stanford Research Institute
TCP	Transport Control Protocol
TCP4	Transmission Control Protocol version 4
Telnet	ARPANET protocol which allows a user at a remote site to login to a time-sharing system as if he were at a directly connected terminal

TENEX	Operating system for PDP-10 computers manufactured by Digital Equipment Corporation
TSA	Tactical Situation Assessment
TT-624	Data General Corporation line printer

INITIAL DISTRIBUTION

NAVAL ELECTRONIC SYSTEMS COMMAND
  CODE 330 (R. FRATILLA)
  CODE 330 (F. DECKELMAN) (2)
  PME-108 (R. WEIAND)

COMMANDER IN CHIEF U. S.
  PACIFIC FLEET
    CDR R. MEINHOLD

DEFENSE TECHNICAL INFORMATION CENTER (12)