

The ongoing implementation of a prototype medical communications system at the University of North Carolina.

B.M. Hemminger, B.G. Thompson, S.M. Pizer, P. Stancil

Department of Radiology, University of North Carolina  
Chapel Hill, North Carolina, 27514

#### Abstract

The communications network is the backbone of a Picture Archiving and Communications System (PACS). A working communications network also facilitates research into the different aspects of a prototype PACS. In our previous PACS communications paper [Thompson82], we outlined the design criteria for a medical communications system. These same criteria are in effect except that we have replaced the analog transmission of images by digital transmission. To meet the communications needs of our prototype PACS configuration, the University of North Carolina (UNC) at Chapel Hill chose to develop a rudimentary implementation of the ACR-NEMA standard to allow communication between acquisition devices, host computers, consoles and other medical imaging equipment.

This paper reflects on the decision to use digital transmission of images and existing digital modalities. We describe the implementation being carried out at UNC, and how the communications network interrelates with the other ongoing UNC PACS research efforts.

#### Introduction

The University of North Carolina at Chapel Hill is currently in the process of implementing a prototype PACS within the Department of Radiology in the North Carolina Memorial Hospital. The major objective of this PACS is research into and the evaluation of PACS configurations and PACS components in a clinical environment. The system is designed to be flexible so that many different types of PACS modules can be easily incorporated into the system. Early use of the PACS will emphasize research and analysis of the individual PACS modules attached to the system and study of the overall system characteristics of different PACS configurations. Although studies will take advantage of the clinical setting at UNC, the initial implementation is not designed as a working clinical product, but rather as an investigative tool for studying the performance of various PACS modules and system parameters in a clinical setting. The initial implementation of the UNC PACS prototype consists of several modules confined to the neuro-service area of the radiology department [Thompson85]. Included in the core facility are the communications network, an archive, several image processing computers, and hardcopy devices. Provisions are being made to interface digitizers, a DSR unit, an MRI unit, several CT units, an ultrasound device, and several nuclear medicine devices. Initial connections of these units will begin in the spring of 1986.

In order to evaluate PACS components and system configurations more efficiently we are extensively modeling the structure of our radiology department including clinician patient flow, equipment location, and information flow [Parrish85, Rogers86]. Using this approach, we are able to quantify and define the radiology operation and test the effect of various resource configurations on the operation without actually modifying the radiology department. This allows us to compare the performance characteristics of different resource configurations without costly empirical studies and the disruption they would cause to the operation of the department. Evaluation of PACS configurations will be developed with regard to the following expected areas of benefit:

- rapid access to images and related information
- rapid access to non-image related data
- access to remote data storage
- better diagnostic capabilities, e.g.
  - rapid comparison of image data from different modalities
  - ability to perform image processing tasks and store/access results
- reduced procedure time
- improved communications in a distributed department
- removal of physical proximity constraints
  - close location of examination and support areas no longer required
  - close location of reading rooms no longer required
  - film related equipment and dark rooms no longer required
- provide for the automatic gathering of statistics on the usage and utilization of equipment, space and personnel.

With the results from our modeling and the analysis of PACS configurations the first stages of clinical implementation, based on PACS, will begin.

#### Communications network: a revision

At the 1982 conference we presented a paper describing in detail the expected implementation of our prototype PACS configuration [Thompson82]. In that paper we listed the following criteria for making a selection of a communications network:

- it must employ proven technology
- it must be of the distributive variety
- it must have an isolation device buffering each tap
- it must provide for both analog and digital information
- it must be of adequate bandwidth
- it must be capable of transmitting the data error free
- it must be capable of future expansion

At that time we choose to implement a network based on a framework of broad band coaxial cable structured in a linear network architecture. Originally, it was our intention to have three coaxial cables available at each tap. Two would be used for the analog video distribution of images while the third would be used to transfer digital RS232 signals to provide the control necessary for remote operating stations. As a first experiment, we had planned to investigate the use of a single screen remote console hooked to one of our fluoroscopy imaging devices. Prior to this we had realized the need for multiple screens [Perry83], however, it was not until actually undertaking the first experiment that we realized the constraints attached to using analog video distribution of images. We found that research describing and evaluating PACS configurations and PACS components was much better suited to digital transmission because of several reasons.

The foremost reason for using digital transmission is the concern for having many images and related data available to any user at any console. The data may be from different sources, of different modalities, and recorded at different times. The important consideration here is that analog distribution of a video signal requires a live video connection between an image source and an image sink. This constrains the network in two ways. First it allows only a few images to be displayed concurrently. The number of images transmitted at the same time is limited by the bandwidth of the network, which is a function of the number of cables in the network. Secondly, it occupies both the input capability of the console (sink) and the output capability of the imaging device (source) until the session is completed. Thus a radiologist at a workstation is limited at the outset to display no more images simultaneously than the number of input cables connected to his console. Additionally, another radiologist wishing to use a second console at the same time to display different images from the same source would be forced to wait until the first radiologist's session was concluded. The important consideration is that unless all of the consoles are to be attached to their own specific imaging devices, the use of analog video to transmit images has serious drawbacks. Since our PACS efforts are directed towards studying different PACS components and system configurations under widely varying conditions rather than studying remote consoles, digital transmission was more appropriate for our research.

A second reason for using digital transmission is to facilitate communications with other information systems, e.g. Radiology Information Systems (RIS) and Hospital Information Systems (HIS). Using digital communications allows the different systems to easily communicate requests and to transfer the many types of data associated with PACS (e.g. images, image data, patient data, scheduling information). Furthermore it facilitates the expansion of such systems to remote sites which can access the PACS by existing mediums (e.g. phone lines, fiber optic cables, microwave links, satellite links).

Another reason is the use of digital archives, which play an important role in PACS configurations. Digital archives allow the error-free storage and retrieval of images and associated image data. Requests to the archive may take the form of simple menu selections or complex database queries. All of these cases, however, may be handled by digital communications between the archive and other components of the PACS.

One final reason for using digital transmission and storage of the images is image processing techniques. The image may be stored and then processed as desired at a console workstation. The results may be stored by describing the image processing technique and the original image or by storing the resulting image. The important advantages are that many enhanced images may be created and then displayed simultaneously and that the resulting images are exactly defined, i.e. if you repeated the process on the original image you would achieve exactly the same resulting image.

The UNC implementation strategy has been modified to include the use of digital

technology as a basis for image transmission and component design. This has resulted in the selection of digital medical imaging modalities as a primary image source and the radiology neuro service as a clinical test bed. The remainder of the paper will describe the specifics of our new work in implementing a digital transmission capability.

#### Implementation of the ACR-NEMA standard

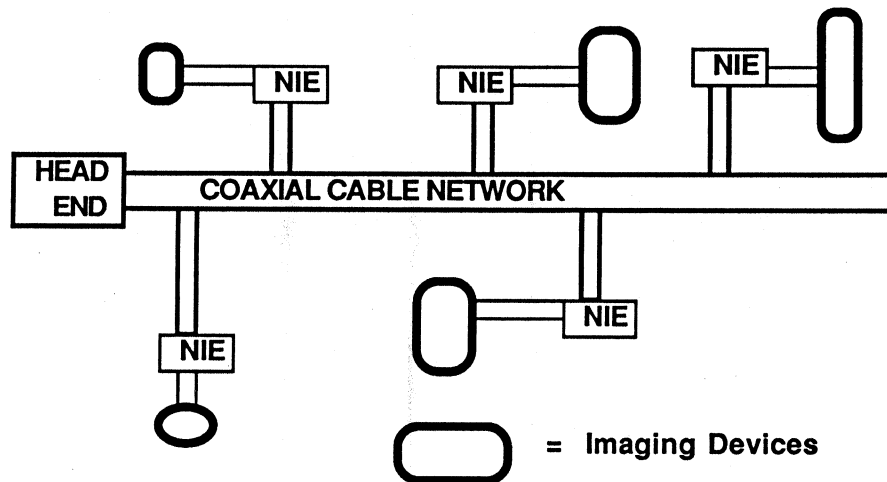
As a result of the above reasons, we moved to an examination of digital transmission protocols in the fall of 1983. At the same time similar interests had convinced the members of the American College of Radiology and the National Electrical Manufacturers Association (ACR-NEMA) to pursue mechanisms for allowing different radiologically oriented computing devices to communicate in a digital fashion. After careful examination of existing alternatives (e.g. DECNET, ETHERNET, etc.) we felt that none of the existing communications protocols met the requirement of being able to communicate between the many types of imaging devices in our Radiology department. Hence we selected the ACR-NEMA standard and began working towards an implementation in conjunction with 3M in the spring of 1984.

The ACR-NEMA standard satisfied all of our requirements by providing for the general digital transfer of image and related data between any two devices. Thus we could transfer data between different ACR-NEMA compatible imaging devices by simply plugging them into our network. This would allow us to carry out our current studies involving

- evaluation of a PACS communications network with many imaging devices (sources) and many consoles (sinks) attached,
- design and evaluation of a digital archive in a PACS,
- design and evaluation of multi-screen multi-image console,
- design and evaluation of a PACS/RIS interface.

Pictorially, our prototype PACS configuration is represented in the diagram in figure 1.

Figure 1: UNC PACS block diagram



NIE = Network Interface Element

In order to perform the evaluations of desired PACS configurations and PACS components, we need to have the ability to easily incorporate PACS components into our communications network. This will be accomplished when manufacturers and others build PACS components which allow communication via the ACR-NEMA standard. Clearly, we are not yet at such a point, as much testing and refinement of the ACR-NEMA standard has yet to be accomplished. While the main goal of the communications group is the development and testing of an ACR-NEMA protocol network, the needs of our other research efforts affect the prioritization of tasks. Having the ability to store experimental image data on the archive benefits several areas of research. With this in mind, we refined our goals as

- 1) choosing, implementing and validating a network that met the previously outlined criteria,
- 2) developing and testing of an ACR-NEMA communications protocol subset between two different imaging devices attached to the network,
- 3) use of limited ACR-NEMA communications on the network to

- allow the storage of images to the archive and retrieval of images from the archive by other nodes on the network,
- 4) develop and implement a full ACR-NEMA implementation allowing other imaging devices to easily attach to our communications network for the purpose of evaluation of PACS components and overall PACS configurations.

The first step in accomplishing our overall goals was to establish a working network. As part of a previous 3M and UNC contract, 3M had already put in place a broad-band coaxial network in the summer of 1983. In late 1984 they attached several Network Interface Elements (NIE's) to allow ACR-NEMA communications over the network. The first imaging device placed on the network was the 3M/1, consisting of a network control unit (NCU) and an attached image archiver. The details of the network and interface elements are contained in [Nelson86].

After the network was in place, it was tested using software in the NIE's to simulate imaginary imaging devices attached to the NIEs. Testing was done to verify that correct image transmission occurred between the NIEs operating in test mode and the NCU.

The second step was to connect another imaging device to the network and to test communication between it and the 3M/1. The computer we choose to attach was our radiology research machine, a Vax 11/730 running under the Unix (4.2BSD) operating system. In addition to functioning as our general purpose research computer, the Vax, in conjunction with a Comtal image processor, is currently functioning as our prototype console machine for our console experiments. Connecting the Vax to the network allows us to add a general purpose image computer resource and an experimental console resource to the network at the same time.

In order for our Vax to communicate with the network we needed three things

- 1) a hardware device to attach to the Vax to allow ACR-NEMA communications to the network at the physical level,
- 2) a software driver for the Vax to properly handle the hardware device,
- 3) higher level software to implement the portions of the ACR-NEMA standard necessary for us to accomplish testing of basic image transmission.

The ACR-NEMA standard in section 7.1 defines several desirable hardware interface features. Included are

- 1) be simple enough to be retrofitted to existing equipment,
- 2) carry 16 bit parallel data with parity,
- 3) Support cable lengths up to 15 meters without comprising speed,
- 4) use asynchronous "handshake" word transfer protocol so that all types of devices can be supported,
- 5) use identical interface symmetry so any two devices can be connected as equals or peers (i.e., no master slave relationship).

We choose to attach a DR11-B to our Vax, primarily because it met the above criteria. Most of the remaining reasons for the choice of the DR11-B were not because it was better suited than other devices, but rather because of reasons such as ease of availability and public domain software drivers for DR11-B's in the Unix environment. In table 1 the correspondence between the ACR-NEMA physical layer and the DR11-B connections is sketched.

Table 1: Correspondence between DR11-B and ACR-NEMA signals

DR11-B		ACR-NEMA
-----		-----
STATUS LINE A	-----	REQO
STATUS LINE B	-----	INTI
STATUS LINE C	-----	Unused
FUNCTION LINE 1	-----	REQO
FUNCTION LINE 2	-----	Unused
FUNCTION LINE 3	-----	INTO
BUSY	-----	STBO
CYCLE REQUEST	-----	STBI
DATA LINES	-----	DATA LINES
PARITY	-----	PARITY

The next two steps were the development of a software driver for the DR11-B device and

the development of software packages to perform the sending and receiving of images between the network and the Vax. The functions implemented in both steps were a direct reflection of our initial goals. From the communications standpoint we were interested in testing some minimal ACR-NEMA functions to assure us that we could develop a working ACR-NEMA communications interface. From the viewpoint of our other research efforts it was beneficial at the time to have the ability to store and retrieve images from the archive because of disk space constraints on our Vax. As a result, the first device driver and applications programs were designed to send and receive images to and from the network, respectively.

Several simplifying assumptions were made for this trial case. The driver, which was responsible for the physical layer of communications was written with the assumption that only one logical connection could be open at a time. The send and receive programs were simple implementations of small parts of the ACR-NEMA standard. Both programs communicated only via the default channel (channel 0) so that they did not need to perform Open Channel Requests and Indications. They sent or received a single image per execution. Although the two programs implemented only a small portion of the ACR-NEMA standard, they implemented a vertical slice so that the implementation would require the testing and debugging of the physical layer through the session layer.

### Results

The primary result of the initial effort was the successful transmission of images in both directions between two machines using separately written ACR-NEMA implementations. The success of this preliminary test confirmed the ability of the ACR-NEMA standard to actually transmit images, thus allowing our communications research group to proceed on to developing a complete ACR-NEMA implementation. The secondary result was that our other research efforts now had access to the 3M/1 image archive.

Two basic tasks were tested. First was the transmission of images from the Vax to the archive, initiated by the Vax. Second, was the transmission of images from the archive to the Vax, initiated by a request to the database program that is used to interface to the archive on the 3M/1. The database system on our archive is discussed in more detail in [Creasy86].

In the first case, a user at the Vax would run a program which sent images to the network. The user would specify to the program the file name of the disk file on the Vax which contained the image. Because the images on the Vax were stored containing only the pixel data, the user would be required to provide the information associated with the image that went into the header of the image. Once this information was obtained from the user and the file had been found to contain a valid image, the program would read the image data from disk, break the data into packets, then add the packet and frame headers, and finally send it to the driver which would transfer it out to the network. Since the messages were sent over the default channel, no open request packets preceded the image. Thus the packets that the program on the Vax sent out were

```
Image Header Fields (i.e. groups 0H,8H,10H,18H,20H,28H,etc.)
Block 0 of image,      (i.e. group 7FE0H)
Block 1 of image,
.
.
.
Last Block of image.
```

The frames were sent as data acknowledge (ACK) so if the reply was not acknowledged (NACK) then the last frame was resent until an ACK was received or the number of NACKs overflowed a preset limit. The NCU in the 3M/1 would receive the frames and make the appropriate responses. If a valid image was received then the NCU would pass the image to the archive where it would be stored on one of the short term storage devices in the archive.

In the other case, the user would select an image from the database program by one of several methods, and then ask to have it sent to the Vax. The database software would retrieve the image from the archive and pass it to the NCU who would perform the appropriate packeting and framing of the data blocks before sending them out to the network. A program on the Vax which sleeps until receiving incoming requests would be woken up by the request, and it would read the incoming image and store it on the Vax disk. The message format would be the same as that for the opposite direction.

Once the system had been debugged and was able to perform the above two tasks it was subjected to several performance studies. Two variables were examined. The first was the system load average on the Vax during the transmission. The other variable recorded was the

size and content of the messages sent. Different types of image data were used, including artificial data (single byte bit-patterns replicated to fill the image) and real images from digital CT scans and digitized film images from several modalities. The size of the images contained in the message varied from 0 blocks (header only) to 64 blocks (512\*512\*8 bits). These studies recorded the debugging information provided by the NIE's and the Vax describing timing, retransmissions, and handshaking at the physical level between the DR11-B and the NIEs. The data are summarized in table 2.

Table 2: Summary of Transmission Data

<u>Size of transfer</u>	<u>Direction</u>	<u>Load Average on Vax</u>	<u>Mean Time</u>
256K	Archive to Vax	light	40.3 secs
256K	Archive to Vax	medium	118.3 secs
256K	Archive to Vax	heavy	191.2 secs
256K	Vax to Archive	light	44.8 secs
256K	Vax to Archive	medium	130.3 secs
256K	Vax to Archive	heavy	210.7 secs
< 256K	Archive to Vax	light	*
< 256K	Archive to Vax	medium	*
< 256K	Archive to Vax	heavy	*
< 256K	Vax to Archive	light	**
< 256K	Vax to Archive	medium	**
< 256K	Vax to Archive	heavy	**

Notes regarding table 2:

- (1) The number of samples for transfer size of 256K bytes ranged from 10 to 30. The number of samples for transfer sizes of less than 256K bytes ranged between 5 and 10. Only the mean values are shown.
- (2) Number of retransmissions per 256K byte transfers was approximately zero in all cases and thus was not listed separately.
- (3) With one exception (see note (4)), there was no variance depending on the contents of the images so this information was not listed separately.
- (4) The modems used in the network had bit-slip problems when transmissions contained long strings of bytes of all zeros or all ones. Images sent that contained these patterns would commonly fail to make it across the network, or would take much longer times to do so. As this was a recognized problem with the modems, such images were not included in the trials that produced the preceding statistics.
- (5) Special notes from above:
  - \* : All the times were proportional fractions of the time required to send a 256K byte image from the Archive to the Vax.
  - \*\* : All the times were proportional fractions of the time required to send a 256K byte image from the Vax to the Archive.
- (6) Definition of load average times on the Vax 11/730 running 4.2BSD UNIX:
  - light: load average of approximately 0.80
  - medium: load average of approximately 2.0
  - heavy: load average of approximately 4.0

In all cases the throughput of the Vax is the limiting factor. The time required to transfer 256K bytes from the NCU to an NIE is approximately 10 seconds on our current system. The bottleneck is the checksum verification and the disk IO in the Vax. The higher cost associated with the Vax sending transmissions as opposed to receiving transmissions is due to the additional software overhead of packeting and framing the blocks. In addition this was done in user space as opposed to the kernel and thus had to compete with other processes for CPU time. In our new implementation the physical through the network/transport levels are all in the kernel and thus the real time costs for transfers will be reduced in all cases.

The effort spent accomplishing our initial task was largely in the area of debugging the low level software. The initial installation of the network and the 3M/1 required relatively little effort as the system had been previously tested by 3M. Designing and debugging the packeting and framing of the image data in our limited prototype was not difficult. The design was taken almost intact from descriptions in the ACR-NEMA standard. Combining the packeting and framing software with existing Unix input and output procedures completed the upper layers of software development. The design and especially the

implementation and debugging of the low level driver, however, was more time consuming. This was not because of inherent difficulties but because we lacked software experience in device drivers and had no hardware debugging tools to assist us. On the positive side, the existence of many public domain Unix device drivers, including several for DR11-B applications, facilitated the understanding and development of the device driver for our system. Additionally, visits by the authors of the NIE software significantly reduced the debugging time. Finally, the use of a logic analyzer helped resolve many of the low level electronic signal questions. Once the driver was performing consistently in a correct fashion the the higher level code was incorporated rapidly.

### Conclusion

After completing testing on our initial implementation we had accomplished our second goal, that of developing and testing ACR-NEMA communications between two different imaging devices attached to the network. Our communications effort is now directed towards the full implementation of the ACR-NEMA standard on our Vax so that our current Vax and other Vaxes in the department may utilize the communications network for their research efforts and for the evaluation of PACS components and overall PACS configurations. The full implementation is being based on the network software framework that exists in the 4.2BSD UNIX environment. When this implementation is finished our next step is to add additional upper level test programs to allow us to conveniently attach other ACR-NEMA devices and examine their operation in an ACR-NEMA communications environment.

A large time investment is required to develop any robust communications protocol. It is important to select a protocol that is capable of meeting current and future needs to avoid the costly task of integrating addition protocols in the future. We feel that the ACR-NEMA standard meets the current and the foreseeable future requirements.

### Acknowledgements

This work was supported in part by a grant from 3M.

### References

#### Creasy86:

Creasy, J.L., Loendorf, D.L., Hemminger, B.M., Initial experience with a prototype archival system at the University of North Carolina, 1986 SPIE PACS for Medical Applications, 1986.

#### Johnston86:

Johnston, R.E., Rogers, D.C., Perry, J.R., Pizer, S.M., Staab, E.V., Curnes, J.T., Hemminger, B.M., Multiscreen multi-image PACS console, 1986 SPIE PACS for Medical Applications, 1986.

#### Nelson86:

Nelson, O.L., Rennolet, C.L., Implementing the ACR-NEMA protocol on a network, 1986 SPIE PACS for Medical Applications, 1986.

#### Parrish85:

Parrish, D.M., Perry, J.R., Thompson, B.G., Johnston, R.E., Rogers, D.C., Staab, E.V., "A Radiology Information System: The Total Approach to Radiology Information Management", Presentation at the Intl. Conf. on The Computerization of Medical Records Creating Patient Information System, Boston, May 31, 1985.

#### Perry83:

Perry, J.R., Thompson, B.G., Staab, E.V., Pizer, S.M., Johnston, R.E., "Performance Features for a PACS Display Console", IEEE Computer, Vol. 16, #8, Aug 1983, pp51-56.

#### Rogers84:

Rogers, D.C., Wallace, R., Parrish, D.M., Information flow analysis as a tool for PACS development, 1986 SPIE PACS for Medical Applications, 1986.

#### Thompson85:

Thompson, B.G., Brenton, B.C., Staab, E.V., Perry, J.R., Parrish, D.M., Johnston, R.E., Rogers, D.C., Creasy, J.L., PACS: Part of an Intergrated Communications Systems, 1985 SPIE PACS for Medical Applications, 1985.

#### Thompson82:

Thompson, B.G., Anderson, D.J., Chaney, E.L., Delaney, D.J., DiBianca, F.A., Guilford, W.B., Jaques, P.F., Johnston, R.E., McCartney, W.H., Pizer, S.M., Scatliff, J.H., Staab, E.V., Washburn, D.B., Medical image communications system: expansion from a prototype at the University of North Carolina, 1982 SPIE PACS for Medical Applications, 1982.