TECHNICAL MEMORANDUM

21

A network for meteorological applications

F. Königshofer and P. Rakity

Research Department

January 1981

European Centre for Medium-Range Weather Forecasts
Europäisches Zentrum für mittelfristige Wettervorhersage
Centre européen pour les prévisions météorologiques à moyen

A NETWORK FOR METEOROLOGICAL APPLICATIONS

F.Königshofer

P.Rakity *

* SIA-Ganymede Ltd.
now with Perkin Elmer Computer Systems Division
227 Bath Road, Slough, Berks.

ABSTRACT

Software development for the Network Front-End Processor (NFEP), at ECMWF is aiming to achieve a maximum of manufacturer independency for connecting to ECMWF's mainframes or for connecting from ECMWF to the outside world. In defining such an "open" network architecture international standards - as far as available - and recent developments have been followed up closely. Some expected benefits are efficient and reliable data transfers,high lines utilisation, problem suited recovery procedures and flexibility towards future enhancements.

1. The ECMWF Environment

The European Centre for Medium Range Weather Forecasts (ECMWF) is an intergovernmental organisation with 17 Member States. It was founded in 1975 in recognition of the advantages that reliable medium-range (up to 10 days) weather forecasts would bring for the economy and in recognition of the considerable computing resources necessary for such tasks, especially if done in a daily, operational manner.

In consequence, the Centre has installed a CRAY-1 front-ended by a CDC/Cyber 175 (the FE).

Forecasts are produced on a 5-day per week operational basis since August 1979, and on a 7-day per week basis since August 1980.

2. The Telecommunications Aspect

The use of telecommunications enables the Centre to

a) acquire observational meteorological data

b) disseminate its forecast results

c) make its mainframes accessible to Member States via RJE

Meteorological observations are continuously taken on a world-wide basis, prepared into bulletins and distributed over a world-wide network set up by the World Meteorological Organisation. The Centre is not directly taking part in this network but receives the data on-line from the United Kingdom Meteorological Office.

The daily operational forecasting cycle at ECMWF starts at a fixed time every evening using the operational data already received. The resulting forecasting products in numeric grid-point format become available at quarter hour intervals as the computation progresses. They are arranged in files consisting of "fields" per individual Member State and distributed.

Member States may submit jobs to the Centre, and receive their output back (RJE). This facility is normally used during the day-time. Remote interactive access to the Centre's mainframes is not provided.

Analysis showed that a star-network would be the best solution. The network will ultimately comprise 17 leased point-to-point connections working at 2400 or 4800 bits per second (bps). These lines are implemented in a phased way from 1979 to 1984. Member States whose lines were due for 1980 or later had been given the option of an interim low speed connection. Over these telegraphic circuits forecasting products are distributed: RJE is not allowed. The current configuration consists of eight low speed telegraph lines (Spain, Yugoslavia, Turkey, Italy, France, Greece, Portugal and the Netherlands) and four X25 connections (United Kingdom, Sweden, Germany and Denmark; Ireland and France are under implementation).

One of the major considerations in the design of the telecommunications at ECMWF was that the European Postal Administration provided medium speed circuits on condition that the transmission protocols would be CCITT's X25. This would allow a future change to public packet switching services.

The decision to use layered data transmission protocols followed. A dedicated Network Front-End Processor (NFEP) should handle all remote communications. Its interface to the FE (the Host Interface) was the most critical aspect of the project.

The NFEP was to be the first non CDC machine connected to the new CDC Host software (INTERCOM 5). We decided to use a layered protocol approach to connect to the Host in order to minimise the risk by ensuring that the layers were isolated.

The Host and Network interfaces were separated. This separation has the advantages that the communication network availability was independent of the host, and the communication processor is available independent of the network user presence.

3. The Network Front-End Processor

3.1 The Hardware

The hardware is grouped around a Regnecentralen (RC) computer with 128K 24 bit words and two disk units of 33M bytes each. An RC3500 together with a CDI 100 channel coupler serve the link to the Cyber 175. Two 8301 peripheral processors handle HDLC (medium speed line) controllers, the asynchronous multiplexors, the printer, card-reader and magnetic tape. See figure 1. The operator display is integrated into a console containing various patch and jack fields, instruments for line and modem measurements and test pattern generation using a Data Analyzer and Oscilloscope.

3.2 Software Overview

The software of the NFEP was designed to completely separate the function of driving the Network and the Host. This approach gave three major advantages over CDC's solution. The first is that the Network is no longer tightly coupled to the availability of the Host, the second is that the communication protocols of the terminals are independent of the Host configuration and the third, the NFEP is capable of automatic recovery in case of fatal error.

The CDC software is configured to logically support a number of HASP workstations while the Network supports a synchronous five and eight bit telegraph terminals and X25 synchronous lines. Extentions to HASP, CDC MODE 4, BISYNC and other communication line protocols are possible without affecting the Host interface.

To implement an independent front-end system while retaining the standard CDC interfaces required that we use layered software processes. This approach led to the use of the disc device to act as the functional separator or buffer between the Host and Network interfaces.

3.2.1 Cyber Interface Layers

An early decision was taken to use the standard CDC interfaces available between the Cyber software (INTERCOM 5) and the CDC 2550 front end processor. This decision allows us to operate a 2550 in parallel with the NFEP. Minor modifications were made to the CDC drivers to implement auto recognition of the NFEP. This was necessary to avoid loading and dumping the NFEP from the Host. Modifications were also made to allow transparent and non-transparent files to be sent from the host on the same stream. Management of files for the network was thus removed from the Host and placed in the NFEP. The total modifications allow the NFEP to be free standing.

The interface into the host software consists of separate modules or processes.

a) The Channel Interface (CI)

This module is located in the RC 3500 and allows the RC coupler to appear functionally similar to the CDC coupler. The major consideration in using a separate computer was to avoid the problem of interrupt thrashing. This separation allows the Channel Interface to be completely independent of the NFEP and has the added advantage that the RC 8000 concentrates on delivering or receiving blocks of data.

b) The Block Flow Control Layer (BTI)

The BTI logically emulates the Cyber host transport protocol (1). The BTI interfaces to the CI over two high speed data channels. One channel is used for upline and the other for downline data blocks. The BTI accepts both transparent and non-transparent data.

c) The Front-end Application Layer (FAI)

The FAI accepts blocks of data from the BTI, code converts the blocks if required and routes them to disc to form files which are logically suitable for transmission to the network. The FAI controls the pseudo HASP terminals by automatically logging the terminals into the host, and regulates the input and output streams depending on the state of the disc. The disc catalog is examined for files suitable for sending upline.

3.2.2 Network Interface Layers

a) Network Application Layers (NAI)

This layer consists of a File Transfer Interface and of a limited Interactive Interface permitting messages to and from the Cyber console operator and file (job) status enquiries. Files are transferred to and from a Member State according to a File Transfer Protocol developed in the Centre (2). It uses the services of b) or c) - see below - to set up a logical link if not yet established, then opens the file transfer with the other end giving the attributes of the file and then passes "buffers" read from the disk as the entities to be transported by b) or c). The receiving end works in a corresponding way. The interactive message exchange is also based on b), but on a very simple, informal, interim protocol.

b) End-to-End Transport Layer (EI)

This interface works towards the Member States according to an end-to-end transport protocol. It receives from a) requests to establish or terminate logical links ("liaisons") and to transport variable length entities ("letters") safely to the other end. It fragments these entities into smaller units in order to accommodate the needs of the underlying data transmission protocol. The "letter" is normally logically meaningful in the higher level a), e.g. the "buffer" in file transfers. The Centre has chosen a simple subset of the IFIP proposal for an end-to-end protocol - see (3) and (4).

c) Terminal Transport Layer (TA)

This interface works towards the Member States according to the asynchronous telegraph protocols. It receives from a) requests to establish and terminate logical links and to transport variable length letters to the other end. It multiplexes interactive and batch data to the terminal, and allows terminal control of file transfers.

d) Data Link Layer

This concerns the reliable transmission of smaller size data units ("frames" derived from the "fragments" of b)) over a communication line; CCITT's X25, level 2, LAP B is the interface on this level. As this is not defined to work in point to point connections, we had to enhance the definition (5). Connection to the modems use the V24 standard and the modems work with modulation according to V26 and V27 standards.

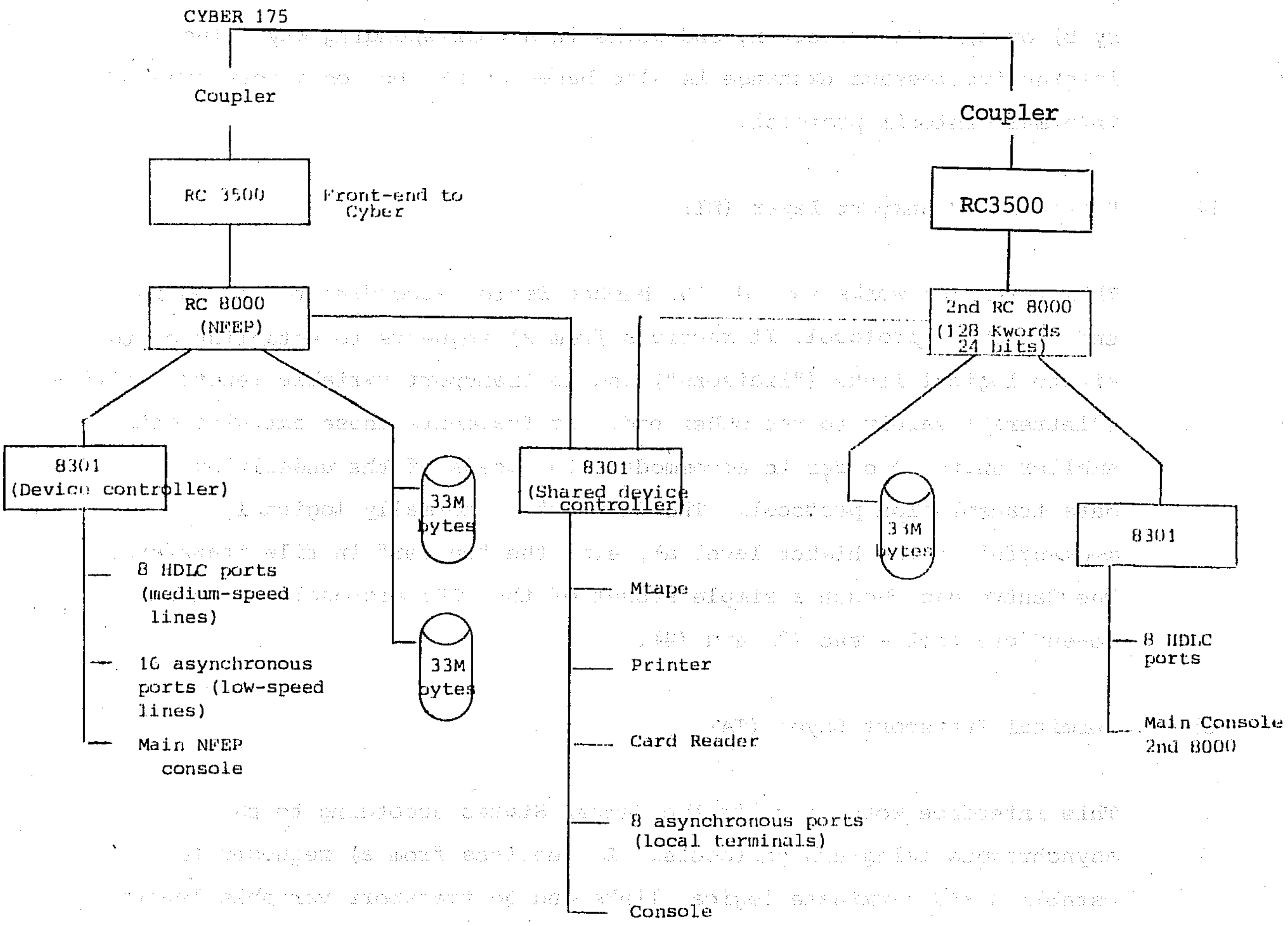


Figure 1 : Configuration ECTS

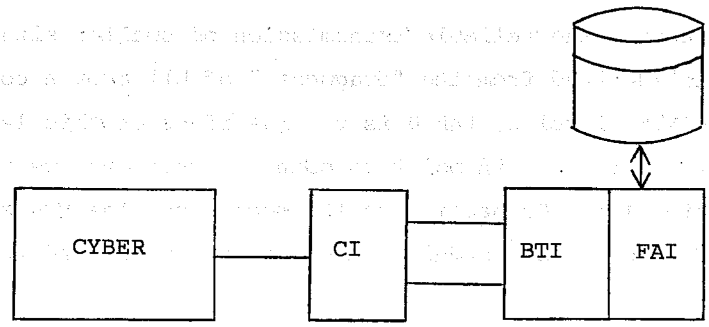


Figure 2 : Cyber Host Interface

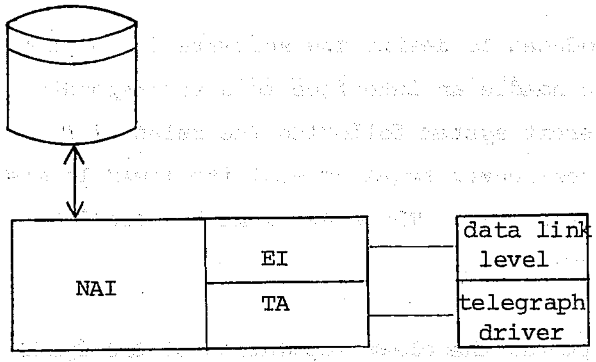
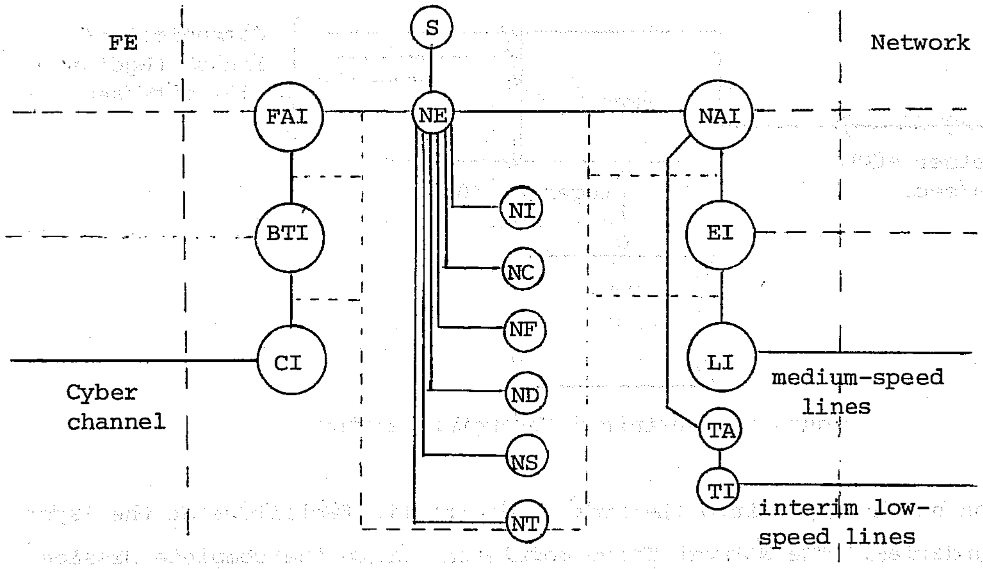


Figure 3 : Network Interface

3.2.3 Software Organisation

The software organisation is given in figure 4



S	System S (RC8000) Basic Operating System
NE	NFEP System Executive + Catalog Processor

FE-modules:

FAI	Front-end Application Interlocutor
BTI	Block Transport Interlocutor
CI	Channel Interlocutor

Network Modules:

NAI	Network Application Interlocutor
EI	End-to-end Interlocutor
LI	Line Interlocutor

TA	Terminal Adaptor
TI	Terminal Interlocutor

Central Modules:

NI	NFEP System Initialisation	ND	NFEP System Dump
NC	NFEP System Configuration	NS	NFEP (Network) Statistics
NF	NFEP File Control	NT	NFEP On-line Test Process

Figure 4 : NFEP Software Organisation

The term "interlocutor" is introduced to define the software (or on the lower levels: Hardware) modules which handle an interface to a corresponding interlocutor in a topologically different system following the rules of a certain protocol. The end-to-end interlocutor together with its lower layers comprises the "Transport Station" of the NFEP. There are some "central" modules with self-explanatory functions.

The usage of the spooling disk facilitates the clear separation of the Cyber and the Network interfaces. File transfers must be completed from one interface side to the disk before the other interface side can take over. As the sustained throughput rates of the disk system and of the FE-link are quite high - figure 5 - we do not expect considerable increases of file transfer delays by this method. Interactive data is not spooled via the disk.

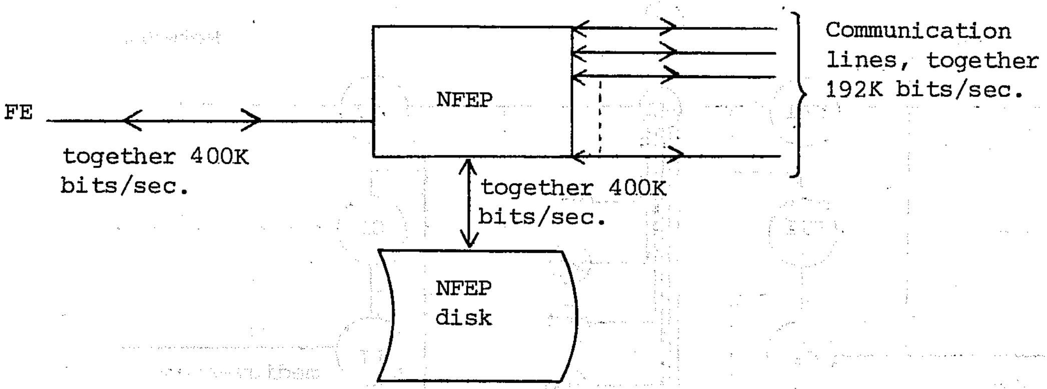


Figure 5 : Sustained throughput rates

Consideration has been put into the area of diagnostic facilities at the layer to layer boundaries. The Network Trace module can trace the complete message flow within the NFEP. This facility has proved invaluable in assisting Member States setting up new lines or experiencing difficulties on an existing link.

4. ECMWF's Network Protocols

4.1 The File Transfer Protocol (FTP)

This has been developed at ECMWF (2). The aim was to provide for an efficient procedure to transfer sequential files of a simple structure which could be mapped into the file structure of many computers. Apart from the file itself, the only recognised structure is the "logical record" which can be of arbitrary length. Files consist of one or more logical records. The FTP enables only the active sending of a file, since the request has to come from the sending end.

File transfers proceed by the sending of "Commands" in either direction. The following table lists the available commands:

RFT	REQUEST FILE TRANSFER	(File) sender to receiver
CFT	CONFIRMATION FILE TRANSFER	Receiver to sender (reply to RFT) The RFT-CFT exchange marks the start or restart of a file transfer.
LTR	LETTER	Sender to receiver.
BND	BOUNDARY (marks start of a logical record, can contain a request for ACK)	Sender to receiver
ACK	ACKNOWLEDGEMENT	Receiver to sender (if required by BND)
SFT	SUSPEND FILE TRANSFER	
EFT	END FILE TRANSFER	

File data is only contained in LTR commands which contain a "buffer" of data of up to 640 characters.

File transfers are started by the sender issuing an RFT together with the attributes for the file: the filename, the application type (e.g. Data Acquisition, Remote Job) and possible restart point. The receiver would normally answer an RFT request by sending a CFT. After receiving CFT, the sender could simply continue and pass the file by sending its data in LTRs. New logical records would start by a BND.

SFT or EFT can be issued at any time by the sender or by the receiver and should be confirmed by the other end with the same command. SFT permits a restart of the file transfer after a delay while after an EFT exchange, the file is irreversibly released at the sender's end as far as the receiver is concerned.

The asynchronous nature of SFT and EFT require the sender of a file to be prepared to receive an unexpected command from the receiver.

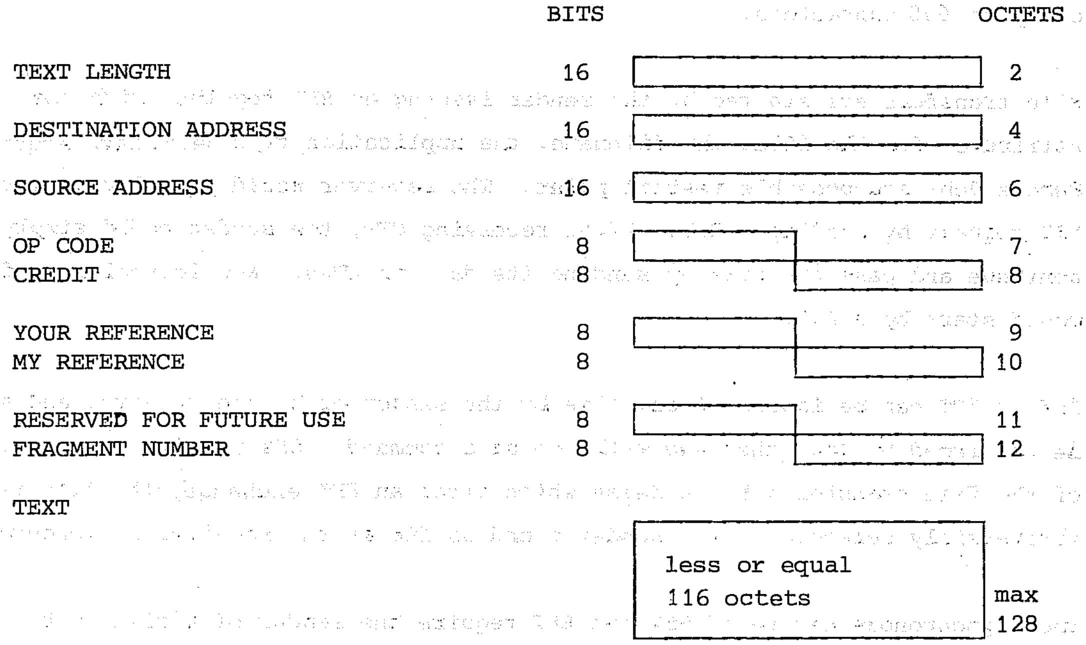
Synchronisation between sender and receiver is achieved by using the underlying end-to-end protocol which has the necessary flow control facilities. The ACK of the file transfer level is intended to allow "guaranteed" restart points ("checkpoints") but a more thorough definition is required before introducing it.

RJE is handled via the FTP. Card-image and line-image conventions follow those used in the CDC/Cyber under NOS/BE. At this very high level a strong Cyber/Cray dependency is the small price for the rather simple FTP and for the avoidance of extensive data image restructuring at the NFEP level.

4.2 The End-to-End (Transport) Protocol (EEP)

This had been derived from IFIP efforts to define a standard end-to-end protocol (4). It permits to establish "liaisons" between application processes in different systems and to exchange variable length "letters" on such liaisons between them. A letter is any entity of the higher level, input to the EEP for transport to the other end. This means, for instance, that all Commands of the FTP are letters in the sense of the EEP.

For the benefit of lower transmission levels, the transport protocol does not transfer letters in their full length but splits them into smaller size fragments. These are preceded by end-to-end control information for that particular liaison. Control information can travel without a data fragment attached to it. The unit of exchange between end-to-end interlocutors is the so-called Transport Command (TS Command), given in the following figure 6



	BITS	OCTETS
TEXT LENGTH	16	2
DESTINATION ADDRESS	16	4
SOURCE ADDRESS	16	6
OP CODE	8	7
CREDIT	8	8
YOUR REFERENCE	8	9
MY REFERENCE	8	10
RESERVED FOR FUTURE USE	8	11
FRAGMENT NUMBER	8	12
TEXT	less or equal 116 octets	max 128

Figure 6 : General Format of a TS Command

Destination and Source Address describe uniquely the liaison - i.e. whether it is a file transfer and who is the sender of the file.

Op-Code describes the nature of this TS command according to the following possibilities:

LI-INIT Initialise a liaison (acknowledged by LI-INIT)

LI-TERM Terminate a liaison (acknowledged by LI-TERM)

LI-LT This is part of a "letter" and therefore contains data in the TEXT field.

LI-ACK Acknowledgement to full letter, but has also EEP status enquiry functions to the other end.

LI-NAK Negative Acknowledgement, retransmission starting from a certain letter is required. This can only be a letter which has not yet been positively acknowledged.

One bit of the Op-Code (called R-bit) requests the other end to submit its status via any feasible TS command after processing the control information in the TS-command. We enhanced this facility and have status enquiry procedures which bring a liaison back into normal working condition after synchronisation problems.

FRAGMENT NUMBER provides sequential numeration of fragments within a letter and contains one bit to identify the last fragment of a letter.

Over a liaison letters can generally flow in both directions, provided CREDIT has been received from the other end. Each end sends letters numbered sequentially in the MY REFERENCE field. Letters coming from the other end are acknowledged by putting their reference number into the YOUR REFERENCE field of TS commands emitted from this end. To permit the other end to send "n" more letters,"n" would be put into the CREDIT field which therefore has a value relative to the YOUR REFERENCE field.

The receiving end of a letter controls whether a letter can be received, or should be re-transmitted. The sending end starts enquiries if it cannot transmit the letter by lack of credit.

Our EEP, in general, relies on the retention of the sequence of TS commands per liaison by the lower transmission layers. Loss of TS commands or their duplication can be recovered.

The EEP gives the higher level a service which if properly used can lead to high sustained throughput rates over certain liaisons.

4.3 The Development of ECMWF's Protocols from Selection and Definition to Implementation

When we were selecting our protocols, ISO and CCITT had just agreed on the HDLC/LAP B standard. Therefore, the decision for LAP B came quite naturally.

A difficult question was what to do with X25, level 3. We were attracted by the idea to incorporate this level into the transport protocol but soon realised the difficulties. These difficulties are described in (7). Finally, we realised we could not, with this approach, keep our options open with regard to later selection of permanent virtual circuits versus virtual calls if a too close correlation between transport liaisons and virtual calls was established. We have left out the packet level of X25 for the time being. When we proceed to the use of public packet switching services, we will implement this level between LAP B and our end-to-end protocol and confine it to its pure network related addressing and packet flow control functions.

Consequently, we chose the IFIP proposal for an internetwork end-to-end protocol, often referred to as INWG (Note 96) (3) as our transport end-to-end protocol.

In its first edition, INWG 96 represented the result of a lot of practical experience - e.g. in the French Cyclades network. At ECMWF, we cancelled the datagram mode, reduced the variety of operation codes but defined NAK usage and sophisticated the sender-receiver (of letters) relationship.

The latter means that for each of the two directions of data transfer over a liaison, the flow is controlled by the receiving end whereas the timers are controlled by the sending end. Our end-to-end protocol definition remained quite stable during the implementation phase, but the initialisation phase and the static/dynamic port structure had to be defined in much more detail.

At the same time, the datagram/liaison mode controversy and the X25-based transport protocol discussion influenced the INWG work on Note 96 resulting in less and less elegant compromises. For our chosen definition new aspects did not appear, and when the efforts on INWG 96 finally broke down, we were not affected.

To select a file transfer protocol we started by studying three proposals for standardised file transfer, namely (8), (9) and (10). As we were aiming at very efficient file transfers we wanted only a minimum of file attributes and corresponding mapping functions. The FTP had to match the facilities provided by our end-to-end protocol. Of the three proposals, Gien's protocol (8) had probably the greatest influence on us. We finally defined our own simple file transfer protocol. The FTP remained relatively stable during implementation. Notable amendments are that transparency of data is now definable per logical record instead of file, implementation dependent features have been removed, and new SFT and EFT codes have been introduced. For instance, it proved useful for a file receiver to have the possibility to end a suspend-mode earlier than given at the SFT exchange. Also useful was a new facility permitting the early termination of a file transfer but keeping that file at the sender with a much lower priority thereby enabling other file transfers to take place. The definition of the file transfer protocol ACK was not precise enough to show its intention to create "checkpoints" during long file transfers. Implementors could easily mis-interpret it as a flow synchronisation thereby just slowing down file transfer speeds. At present, we are studying the checkpoint problem again; should the receiver be allowed to reject a checkpoint request, e.g. if his operating system does not support the creation of checkpoints.

First practice has shown that EEP and FTP are fulfilling our expectations, i.e. they are reliable, easy to debug and permit very good efficiency.

This has encouraged us to go ahead with the introduction of multistreaming of files, i.e.we allow several files in one direction in a multiplexed way between a Member State and the Centre if the Member State's configuration includes several concurrently running peripherals. When we started operations, only one file transfer per direction was allowed to run simultaneously but to remove this restriction was in the full spirit of the EEP and its facilities.

5. Future Considerations

It has already been mentioned that level 3 of X25 would fit in smoothly between LAP B and EEP handling. This makes it possible to judge, whenever we want to, the economic trade-offs of switching certain or all of our links through a public packet switching service. As our Member States will have to run the same protocols as us, their situation in such a move will be similarly flexible.

We also have a medium-sized local network of alphanumeric, graphics and RJE terminals which are connected to the CDC/Cyber via a 2550. We shall investigate if and how all these terminals could be integrated via the NFEP. The RJE terminals obviously have to follow our FTP, EEP and LAP B protocols. The necessary software has been produced by SIA-Ganymede.

However, the integration of alphanumeric and possibly graphics terminals would require more investigations which would centre about the provision of some Virtual Terminal Protocol (VTP), maybe based on our EEP.

Our protocol structure would give us even flexibility towards application to application communication or remote data base direct access, etc., but at this stage, the Intercom interface from NFEP to CYBER would probably no longer allow a mapping for such facilities.

We are also re-assessing with interest the protocols we selected from the experience we get in using them, their suitability for future applications in our network and how they relate to the ISO OSI attempts (11).

CONCLUSION

The timetable for the implementation of the ECMWF network required the selection of protocols during a period of intense international efforts towards standardisation or towards finding the base for standardisation. We tried to choose advanced but solid protocols and are now gaining encouraging experiences with them. Quite clearly, there is always a price to be paid: simplicity - usually the work at the higher level either at sender or receiver must be more elaborate: generality - simple dedicated terminals with just one or two applications require the full protocol-layer set-up.

The implementation of the NFEP as described in this paper was carried out by the Ganymede division of SIA Limited on the basis of specifications prepared by ECMWF.

REFERENCES

1. Communication Control Intercom 3.0.
Communication Subsystem for NOS/BE INTERCOM 5.
Specification 74872890, December 1977, Control Data Corp.

2, Quoilin, P.
"The ECMWF File Transfer Protocol", ECMWF Computer Bulletin B3.3/3,
Revision 1, Dec. 1979.

3. Cerf, V., Mackenzie, A, Scantlebury, R. and Zimmermann, H.
"Proposal for an Internetwork End-to-end Protocol", Internation Network
Working Group (INWG) General Note 96, May 1975, with later revisions
and re-editing 1978.

4. Königshofer, F., Haag, A., Quoilin, P.
"The ECMWF End-to-end Protocol", ECMWF Computer Bulletin B3.3/2,
Revision 2, January 1980.

5. Haag, A.
"The ECMWF Data Link Protocol", ECMWF Computer Bulletin B3.3/1,
Revision 2, October 1979.

6. Wilke, K.
"Setup of Service to ECMWF and Testing Procedures", ECMWF/CAC(78)22,
August 1978.

7. Sexton, J.H.
"End-to-end Protocols, Virtual Calls and the IIASA Network", CSN 021,
International Institute for Applied Systems Analysis, Laxenburg, (Vienna)
June 1976.

8. Gien, M.
"Proposal for a Standard File Transfer Protocol (FTP)", DAT 519,
EIN/IRIA/77/4, May 1977.

9. Heinze, W. and Butscher, B.
"File Transfer in the HMI-Computer Network", Proceedings of 3rd European
Network Users Workshop, Laxenburg (Vienna), April 1977.

REFERENCES (continued)

10. "A Network Independent File Transfer Protocol", prepared by the High Level Protocol Group HLP/CP(78)1, INWG Protocol Note 86, December 1977.

11. "Open Systems Interconnection" ISO/TC97/SC16 N 117, June 1979, version 4.