Overview of the GMRT Online Computer System (Software)

A. Pramesh Rao
12 Dec 91

I.Introduction

The Online computer system of the GMRT controls 30 antennas distributed over an area of 25 km. A number of reports by CR Subrahmanya (GMRT report numbers 90100 to 90105) have given an overall concept of what was visualised as the structure of the system. The present document concentrates mainly on details of the implemention zeroth version of the Online system giving the status of the various components as of Dec 91. A rough projection of when the various components required to put together a minimal system are also given.

II.Scope of the Online System

There seems to be some confusion on what is the Online system and where it ends and the Offline system begins. In the first phase, I will adopt the extreme view that the Online system consists of the set of functions and programs that have to be done in real time. While an astronomer may want to produce high dynamic range maps of sources or estimates of source parameters in real time, these do not fall under the purview of the Online system since, apart from the astronomers wish, there is no technical reason why they should not be done days later. In contrast, the calculation of the altitude and azimuth of a source is a part of the Online system since without this, one does not know where to point the telescope. One can think of the Online system as that which is used by the telescope operator from the operator's console for controlling the antenna systems. It is generally a good practice not to allow the astronomer wanting to analyse the data to work from the operator's console while the observations are going on. The astronomer accesses the data recorded by the Online system through the Offline system which gets its inputs from files on the hard disks of the Unix file systems. Since its inputs is from files in the file system, the Offline programs can be run at any time, even during the observations, so long as the astronomer is given "read" access to the files, but clearly, during abservations, the Offline programs will run at a lower priority than the Online programs if both are running on the same computer.However, with computer networking, this is not an issue, since the astronomer can be running his programs on a different computer and access the files containg the Online data over the network.

III.General Overview

The Online computer system consists of a variety of cpu's (Sun class machine, PC-like 80186 microprocessors and smaller 80C535 microprocessors), a variety of operating systems (Unix on the Sun and locally written kernels on the microprocessors), programs written in a variety of languages (Assembler, C, Pascal and Fortran) and a variety of users (the telescope operator, the engineers, the programmers, the system astronomers and astronomers wishing to use the telescope). Each user will have a different expectation of the Online system and their expectations will be articulated only after using the system. Even if a complete specification is drawn up in advance, its implementation will necessarily have to go in stages since the programmers are new, one needs time to get familiar with the newly acquired software tools, the hardware platforms on which the software runs are still being debugged and modified, there is no experience with the system in the field and no guideline as to what the software should do when something malfunctions. With these considerations in view, the development of the Online system was planned in phases.

A. Hardware Overview

The hardware aspects of the control and monitor system have been reported elsewhere in GMRT reports by A. Ramakrishna (GMRT report no 90196). Here we a brief description of the hardware only for completeness. The configuration of the hardware is shown in Fig. 1.

The Central Electronics building houses the Unix system on which the correlator outputs and the Control and Monitor information are stored. In the first phase the Unix system will be a Sun 3 workstation with a few hundred megabytes of disk, to be replaced as the need arises, by a file server, the Sun 3/60 (called "tifr") which in turn will be replaced by the Sun Sparc 4/90 ("gmrt"). In the final setup, the operator console workstation (initially just a monitor) and other terminals for looking at the Control and Monitor system will be connected to the the file server through an ethernet. An independent ethernet will connect the server to the correlator computer (Sparc 1E card) and the correlator data will be recorded on the file server through this dedicated ethernet connection. The server will communicate to the Control and Monitor system through a 19.2 or 38.4 kbps asynchronous serial link to the Communication Handler card (COMH). The COMH is a standard Antenna Base Computer card sited in the Central Electronics building whose function is to act as an interface between the asynchronous serial link required by the Unix system and the 128 kbps synchronous HDLC protocol used over the optical fibre system. The Antenna Base Computers (one for each antenna) are connected to the COMH by the optical fibre. The ABC which is a PC class 80186 microcomputer is connected through a 9.6 kbps asynchronous serial link to the Servo Control Computer and to an as yet unspecified (maximum of 16) number of 80C535 based Monitor and Control Modules. The MCMs can control 16/32 switches and read the voltages (with 8 bit accuracy) from 62 monitor points. The Servo Control Computer (SCC) is a 8086 microcomputer based system designed by BARC that controls the opertations of tthe Servo system. The SCC and the MCMs are the basic elements of the Control and Monitor systems - the rest of the system essentially handling communication to ensure that the user commands reach these modules and the data from these modules reach the user.

B. Users Overview

1. The Minimal System (Phase I)

The first phase is a minimal single user system that tests the basic hardware and software. The system is lean and relatively simple so that overall system can be analysed and problems can be easily located and isolated. It is assumed that the complete chain of processors is available, but in a crunch, the system can also be run from a PC, in the place of the Unix machine, though with reduced capability.

In the first phase the main emphasis is on communication - how to get commands and responses to flow up and down the chain, protocols and so forth. Commands typed by the operator on the console, like set 3rd switch of 4th MCM of 5th antenna or stop antenna no 2 or read 43rd monitor point of 2nd MCM of the 1st antenna, should be carried to correct destination, executed and the results and error conditions should be brought up the link to the operators console. Issues of user friendliness, astronomical requirements and Online decision making have been relegated to second place. If some element in the system misbehaves, like an antenna does not move when commanded to or some MCM refuses to respond, the matter is reported on the operators console and the operator will have to intervene. In this phase of operation, a list of frequently occurring error situations will be compiled and if an officially approved standard action is available, in the next phase, these standard responses will be incorporated in software. If the operator's console is on a Unix class multitasking system, the astronomer can expect to track a specified source (including corrections for pointing), give offsets in RA-Dec or Elev- Azimuth coordinates and have these offsets vary linearly at a user specified rate (to give scans). All data coming to the Unix class machines will be recorded and interfaces provided so that the Offline system can access the recorded data.

The minimal system is expected to be integrated and be ready by the end of January 1992.

2. Enhancements to the Minimal System (Phase II)

The minimal systems will be easily upgradeable to provide most of the features required by the user. After the minimal system (and the associated hardware) is debugged and in routine operation, additional facilities required for regular operation will be incorporated. Some of the features envisaged to be available by the middle of '92 are:

1. The programs on the Unix machine will be enhanced to allow multiple sub-arrays with a number of users (upto 5?) looking at the Control and Monitor data and possibly controlling different sub-arrays from different terminals. This facility is required if engineers are to debug an antenna that is giving problems without stopping observations with the rest of the antennas. This facility requires writing a supervisory program that accepts inputs from the various users, checks the users authorisation and privileges and passes valid commands down the communication chain. This program will be primarily controlled by the operator.
2 A better user interface and the facility to control a sub-array through user created OBSERVE files. The format of the display of the control and monitor data will be tailored to suit the needs of individual groups. Thus there will be different display formats available for people wanting to make astronomical observations, look at and debug electronics, servo, mechanical and other subsystems.
3. A limited coupling between the Online and the Offline systems so that even during observations one can estimate the gain and system temperatures of the antennas and apply these corrections to the visibility before it is displayed, calculate and correct the antenna based phase error so that it is possible to use tha antennas in a phased array mode.
4. Based on experience of the reliability of the system, more and more of the operators decision making will be automated. The Antenna Base Computer will be enhanced to examine and filter out much of the monitored data which will reduce the data traffic on the optical fibre link.

Work on this phase will start after the minimal system is working in a stable manner. Many of the features listed above along with additional facilities can be expected to start appearing from the middle of 92. This enhanced system is expected to satify most of the users needs for at least a couple of years and could be the final system unless someone takes the initiative and completely redesigns the Online system.

3. Comprehensive rewrite of the Online System (Phase III)

At some point of time, after the whole system is working for a couple of years, and one has enough experience with its idiosyncracies as also the various modes in which the astronomers would like it to operate in, it might be worthwhile to completely redesign and rewrite the whole system. Since by this time, the hardware would have been completely debugged, one can design the system with the confidence that there will be no surprises. Further, since there is a working system in place, there will be no great urgency to get something working. This should be a professional software exercise using state of the art techniques.

C. Software Overview

The software of the GMRT Control and Monitor system has a hierarchical structure with the Operator Console Program at the top and the MCMs and the SCC at the bottom. To get an overview of the software structure, we will start at the bottom of the pyramid and go up.

The Servo Control Computer(SCC) has on it a control program developed by the BARC group which takes care of the servo system and the safety of the antennas. It expects asynchronous inputs over the serial line - commands like "STOP","MOVE Elevation to 40:30:00", "READ Analog Values" (encoders etc), etc and it return over the same link the status of previous commands, requested Analog and Digital values, and any unexpected events or fault situation. If no command is received, the SCC continues the last command and if the SCC thinks that no one is there at the other end of the link or that the link is down, it stops the antenna and if winds are high, it takes it to the safe position.

The MCM has on it a control program that takes commands from the serial port to set or unset user specified switches, to select which of the 64 analog channels should be monitored and to return the values of the selected analog channels. All the MCMs communicate to the ABC on a common bus. To avoid collisions, the ABC controls all communication on the bus and each MCM returns the acquired monitor data only when requested by the ABC.

The minimal program on the Antenna Base Computer just handle communication. It receives commands for the SCC and the MCMs from COMH, sends the commands to the right module, gathers information from the SCC and the various MCMs ands sends the information back to the COMH. The ABC program operates on a 1s cycle. Every second, it receives a packet of commands from the COMH (a null command packet if there are no real commands). The ABC immediately sends back to the COMH a packet containing the Control and Monitor data accumulated during the last second. During the rest of the second the ABC talks to and receives information from each of the subsystems under its control (SCC and MCMs). The ABC has to do this even if there are no commands to be distributed since the SCC and MCMs return information only when they are asked. Thus once a second the ABC sends back to the Central Electronics Building all the digital and analog information from the SCC like encoder values, motor currents, voltages, wind velocity etc, and all the selected monitor points from all the MCMs. If the number of selected monitor points for the MCM is too large for the bandwidth of the MCM bus, the MCM readout rate can slow to once every two seconds.

The software on the COMH handle the traffic on the optical fibre and converts the synchronous data on the optical fibre to asynchronous data required by the Unix system. Though there are independent optical fibres to each antenna, the ABCs cannot transmit at random since the hardware and COMH can handle messages only from one antenna at a time. To avoid collisions, the traffic on the optical fibre system is orchestrated by the COMH which also operates on a 1s cycles. The COMH broadcasts a packet of commands (real or null) to an ABC. On receipt of this, the ABC which has the attention of the COMH, transmits to the COMH the collected information of the previous second. This sequence is repeated for all the antennas in the system. Under normal conditions, this entire sequence will be completed for all 30 antennas once every second. While this is going on, the COMH will also be receiving commands for the antennas from the Unix system and sending monitor information from the antennas to the Unix system.

The software on the Unix system is a collection of independent but communicating processes that run on one or more Unix machines. The basic elements of the Unix system will include a process for handling the user inputs, a process for displaying the monitor information and a process for controlling the serial link to the COMH that sends commands to, receives and logs data from the COMH. Additional processes like one for tracking a source or to service other users can be added as needed.

IV.Elements of the Online System

A. Monitor and Control Module (MCM)

The MCM software has been written by Mukund Gadgil, who is also writing the documentation. The first draft is available with him. The software is written in Assembly language.

a. Functionality

When powered on, the software on the MCM card does a self test and if all is OK, enables addresss interrupts from the serial link to the ABC. When it receives an address interrupt, it checks if the address matches its own and if yes, it enables data interrupts. When it receives the packet (packet length is a part of the packet header), it checks the integrity of the packet (checksum) and if it is error free, it is decoded. Three kinds of commands are possible -

1. SET Analog mask to select which of the 62 analog channels are to be scanned and values returned
Digital mask (16/32 bit) to select which of the 16/32 switches should be turned on of off
Mode to select the mode in which the MCM should operate. Currently the only modes are IDLE (do nothing) and SCAN (get the values of the previously selected analog channels).
The set values stay set till another SET commands change them.

2. READ Self Test Results
Program Version
Analog Mask
Digital Mask
Mode
Reads and returns the current value of these quantities

3. NULL Continue doing whatever you were doing. In the SCAN mode, it gets the values of the previously selected analog channels. In the IDLE mode, it continues to idle.

On receipt of a packet from the ABC, the MCM sends back a packet giving the status of the MCM and the link, and if in the SCAN mode, the data for the previous second. If the ABC command is a SET or a READ command the value of the quantity set or requested to be read, is also sent back as confirmation. After sending the data packet back to the ABC, the MCM goes into a cycle, doing nothing in the IDLE mode and repeatedly sampling the required analog channels when in the SCAN mode until it gets an address matching interrupt from the ABC. The ABC is programmed to collect data from each MCM once a second, but if the ABC fails contact the MCM in a particular second, data for that second is lost.

b. Limitations - General and of First Version

The 80C535 microprocessor on the MCM card has only 256 bytes of Random Access Memory which could be a limitation. After taking into account the various programming requirements like stack, register banks, buffer for incoming and outgoing messgages, etc there are only about 60 bytes of ram free for storage during program execution. In the current SCAN mode at most 60 of the possble 62 analog channels can be sampled and stored for transmission to the ABC. While this is not, at present, a serious limitation, if we want the MCM to calculate the mean of analog channels, since to accumulate the sum of the sampled points we need at least 2 bytes, we can at most calculate the means of 30 channels at a time. If we want to calculate the variance we need 5 bytes per channel and so we can estimate on the MCM card the mean and rms of at most 12 channels. In the current implementation of the software, it is assumed that the analog channels have time constants of the order of a second or more, so that they need not be sampled faster than once a second and then any calculation of mean, variance or histograms can be done higher up in the chain - either in the ABC or in the Unix machines.

The present implementation does not support different sampling rates for different analog channels.

The MCM program is burned onto EPROMs that are on the card. Modifying the MCM program involves changing the EPROMs on all the MCM cards which could be inconvenient since there are a large number of them spread over a large area. Since the programs are written in Assembly language, which is not easily read and hence verifiable by many, frequent modifications are discouraged.

c. Status and possible enhancements

Two versions of the MCM program are available. The first is for a single MCM being controlled by a PC (no 9 bit addressing problems) and the second is for a number of MCMs being controlled by the ABC. The first is useful for testing and integrating the MCMs with the actual elements they are supposed to control and the second is what is actually required for the Online system. Both versions, along with the PC and the ABC programs, are working and are being tested.

The user interface of the PC program can be improved and will be done by the end of December.

More modes of operation of the MCM like Mean and Variance, Spike detection etc can be added. In the first phase these will not be implemented unless there are compelling reasons.

B. Servo Control Computer (SCC)

The SCC software is written by the Reactor Control division of BARC. The software is written in Pascal and a description of its functionality and its interface with the ABC is available (GMRT Report No 90111).

a. Functionality

The SCC software is an independent system that controls the Servo system and takes care of the safety on the antennas. It accepts user inputs from the serial port, checks them and executes them if they are found to be acceptable. The SCC takes three kinds of commands -

1. DISPLAY command to return the encoder values, motor currents, wind, time, and various kind of Status flags.
2. SET commands to set time of day, safety threshold for wind velocity, etc
3. Actual commands like Coldstart, Stow release, Stow, Stop, Abort, Hold, Position a given axis at a given angle, Track a given axis to reach a specified position at a given time.

From the user's point of view, to go to a given position, one effectively tells the SCC "Position Elevation at +30:00:00". If the command is accepted (it can be rejected if the antenna is stow locked or if the wind is too high or if there is some malfunction), the user can forget about it until the antenna reaches the target, when the SCC sends a message out on the serial line saying that it has reached the target. To track a source, the position and the rate of change of position on each axis have to be specified, which is done by giving a target postion and a time and the SCC will move from the current position with a velocity such as to reach the target position at the specified time. The SCC knows only the horizon coordinates (Azimuth and Elevation) and the conversion from astronomical coordinate systems to the horizon system will be done in the Unix machines.

b. Status

The SCC software has been ready for some time but has not been realistically tested with the hardware. To simulate the ABC giving inputs to the SCC, the BARC group has developed a PC based program that simulates the ABC. This program has been with us for more than 6 months and has been used by us to test the communication protocol.

A few small modifications that have been suggested by us have been agreed to but the updated versions of the programs and documentation have not been received.

C. Antenna Base Computer (ABC)

The software for the ABC is being developed by Madhura Anturkar in C using Intel's IC86 development system for 8086/8088 class processors.

a. Functionality

In the first phase, the ABC software is mainly concerned with communication- getting commands from the COMH to the SCC and the MCMs and getting data and status from the SCC and MCMs back to the COMH. The software on the ABC works on a 1 second cycle. Each second it receives a packet of commands from the

COMH and it immediately returns to the COMH a packet containing the accumulated data of the previous second. During the rest of the second, the ABC issues a DISPLAY command to the SCC to get its status, issues any commands from the COMH to the SCC taking care of all the communication protocols and ensuring that they are executed. If unable to reach the SCC or if unable to get the command confirmed by the SCC, the ABC software informs the COMH. In parallel, the ABC communicates to every MCM under its control, passing any commands from COMH or null commands and gets back the data accumulated by the MCMs in the previous second, again taking care of the communication protocols. At the end of the second, the ABC makes a packet of all the data and puts it in the transmission to COMH buffer which is automatically transmitted when the reception from the COMH is completed.

The software for the ABC consists of three communication handlers, one to the SCC, one to the MCMs and one to the COMH. All the communications are interrupt driven with highest priority to the COMH, next to the SCC and last to the MCMs. All the communication handlers use flags to indicate when transmission or reception are complete and when there are error conditions that cannot be handled by the communication layer. The main program consists of an endless loop, synchronised to the second tick, looking at all the communication handlers flags and taking appropriate action. The arrival of the packet from the COMH need not be synchronised with the second tick and if a command to an MCM is received after the ABC has talked to the MCM, the command will be sent to it only in the next second.

b. Limitations - General and of First Version

The main general limitation of the software on the ABC system is that we have not been able to get floating point arithmetic to work on the 80186 processor on the ABC card. We suspect that this is because floating point arithmetic is done by a floating point accelerator emulator that makes calls to the operating system and we have not bought any operating system for the processor. What operating system we have is one written locally to download an IC86 program developed on a PC down to the ABC card. The absence of floating point arithmetic does not hurt us in the first phase but could be a nuisance later if we want to make, say, the pointing corrections for the antenna on the ABC, which is the logical thing to do. One can overcome the problem by doing integerised real arithmetic or by writing or locating a floating point arithmetic subroutine library.

c. Status and possible enhancements

A simple operating system for the ABC card that downloads executable code generated by the IC86 development system has been developed and has been in routine use for more than 6 months. The communication handlers for the MCMs and the SCC are working and tested, the latter using an SCC simulated on a PC. A communication handler for the COMH (simulated on a PC) using asynchronous serial communication is working and the three have been integrated to the form the first version of the ABC software. With this system, one can enter commands to the SCC or the MCMs on a PC and display data from these systems on the PC. The system has just started working and needs to be tested more critically. Work on the communication handler to the COMH in the synchronous 128 kbps HDLC mode is just starting now that two working ABC cards are available. Realistic testing of the ABC can be done only we get BARC SCC and have it connected to something real.

The first version has no built in logic to scan MCM and SCC data to see if things are normal and handle error conditions, everything is passed up to the operator console. This leads to large amouts of data being transmitted to the COMH. While this is manageable when there are only a few antennas, this model cannot be extended for full GMRT. However without real experience with the system, one cannot finalise the logic and strategies to be adopted by the ABC software.

D. Communication Handler (COMH)

The COMH is a C program running on an ABC card located in the Central Electronics building, that controls the traffic on the optical fibre and acts as an interface between the 128 kbps synchronous communication on the optical fibre and the serial link required by the Unix system. The software for the COMH consists of a main program that controls two communication handlers, one to the Unix system and one to the ABCs over the optical fibre. The COMH will communicate with (send a packet of commands and get a packet of data from) each of the 30 antennas once every second, and push the antenna data to the Unix system as they come.

Work on the COMH software is just starting and one can expect stand alone versions of the two communication handlers to be ready by the end of December '91 and the integrated COMH a month later if priorities do not get changed.

E. Unix Systems (Server / Operators Console)

The Unix system will consist of a number of processes (three or more) running on one or more Sun computers. In the first phase all the processes will run on a Sun-3 workstation with a few hundred megabytes of disk space. Later, as the need arises the Sun 3/60 file server and additional work stations can be added. The Unix software will be written in a mix of C and Fortan, with the system oriented programs in C and the astronomy oriented programs in C or Fortran 77. The final version of the software will be planned and implemented by B. K. Ravi while the first version will have inputs from many different sources.

a. Functionality

The final version of the Unix software should allow multiple subarray operation, multiple users to control different subarrays, any user on the network, even from Pune to display the status of any subarray, logging of user specifiable monitor points and so forth. This will involve writing a supervisory program that will take care of issues of security, priorities, communications between various users and processes and so forth. In the first version we will be less ambitious and require that all commands to the antenna systems come from the operator console. Facilities for having multiple subarrays looking at different sources will be provides but all subarrays will be controlled from the Operator's console. From different terminals, users other than the Operator will be able to login and display the Control and Monitor data and also the visibility data in near real time. The Control and Monitor data and the visibility data will be routinely logged.

b. Software description

As said earlier the Unix software can be written as a collection of independent but communicating processes running on one or more machines. One process will control input output on the serial port that is connected to the COMH. It gets input commands from the process controlling the Operator's console and sends them to the COMH and writes the data from the COMH onto the fileserver. It also communicates with the process controlling the display on the Operator's console so that the latest status is displayed immediately. The process controlling display will have a variety of modes to suit the needs of individual groups. The display process controlling the Operator's console gets the Monitor data as it comes into the Unix system, the display processes on secondary terminals will take the Monitor data from the logged data on the disk.

The process controlling the Operator's console should parse the Operator's inputs and convert the Operator's commands into the form that is required to reach and be understood by the final processor actually executing the code. Thus when the Operator types "ANTEN=7;STOP", it gets converted into the hex code which the SCC recognises as the STOP command, with a preamble which COMH recognises as meaning send the following packet to ANTENNA number 7. The Operator's console process recognises and handles all the raw commands to the elements of the

Control and Monitor system. In addition addition, it will also have the facility of executing standard or user created macros or sequences of raw commands. Thus, there could be a macro called 327MHZ which is a text file in a standard area containing the sequence of raw commands required to set the antenna to 327 MHz and when the Operator type "ANTE=3;327MHZ ", the parser insets all the commands in the 327MHZ file, effectively setting the frequency of antenna 3 to 327 MHz. The Operator's console process should also have the facility to spawn processes which run nearly independent of it. Thus, if one wants to track a source, one has to periodically (every 5 seconds or so) calculate the Elevation and Azimuth of the source and issue a track command to the SCC. Ideally, this should be done on the ABC but at present it cant because floating point arithmetic is not possible on the ABC. On the Unix machine, the most convenient way to handle this is to spawn a subprocess that wakes itself up every 5 seconds, forms the TRACK command for the SCC, sends it to the COMH and goes back to sleep, and continues to this until the Operator tells it to stop. Another application of spawning subprocesses would the facility for the Operator to transfer control to a process that reads an astronomers OBSERVE file and proceeds to execute the instructions in the OBSERVE file in the background, periodically informing the Operator of what it is doing, through the Operator's display process.

All the above features are available in AIPS which uses POPS (People Oriented Parsing System). In AIPS the ADVERBS are keywords that the parser recognises, VERBs are the commands, the TASKs are spawned subprocesses that run in parallel to AIPS and the PROCEDURES and RUN files are the macros which are executable sequences of VERBs and TASKs. In the first phase, it is intended to take AIPS and modify it to suit the needs of the Online system. This would involve adding about twenty ADVERBs, a dozen or so VERBs and writing a few TASKs, one to track a source and one to handle an OBSERVE file being essential. This strategy is not very elegant, since in modifying AIPS like this, there will a number of features of AIPS that will not be used and so are a drag in some sense. But it has the advantage that it can be made to work at short notice (time scales of two months), and more important, give the software group a hands on feeling of what the astronomers want from the final system which they will have to design and implement.

c. Status

Work on the Unix system is just starting. Some work has been done on checking out how difficult it will be to adapt AIPS to suit our needs. No major problems are anticipated. A paper design of the second phase of the Unix system is also being started but serious impementation will not start till the first phase is in operation for a while.

Overview of the ON line System

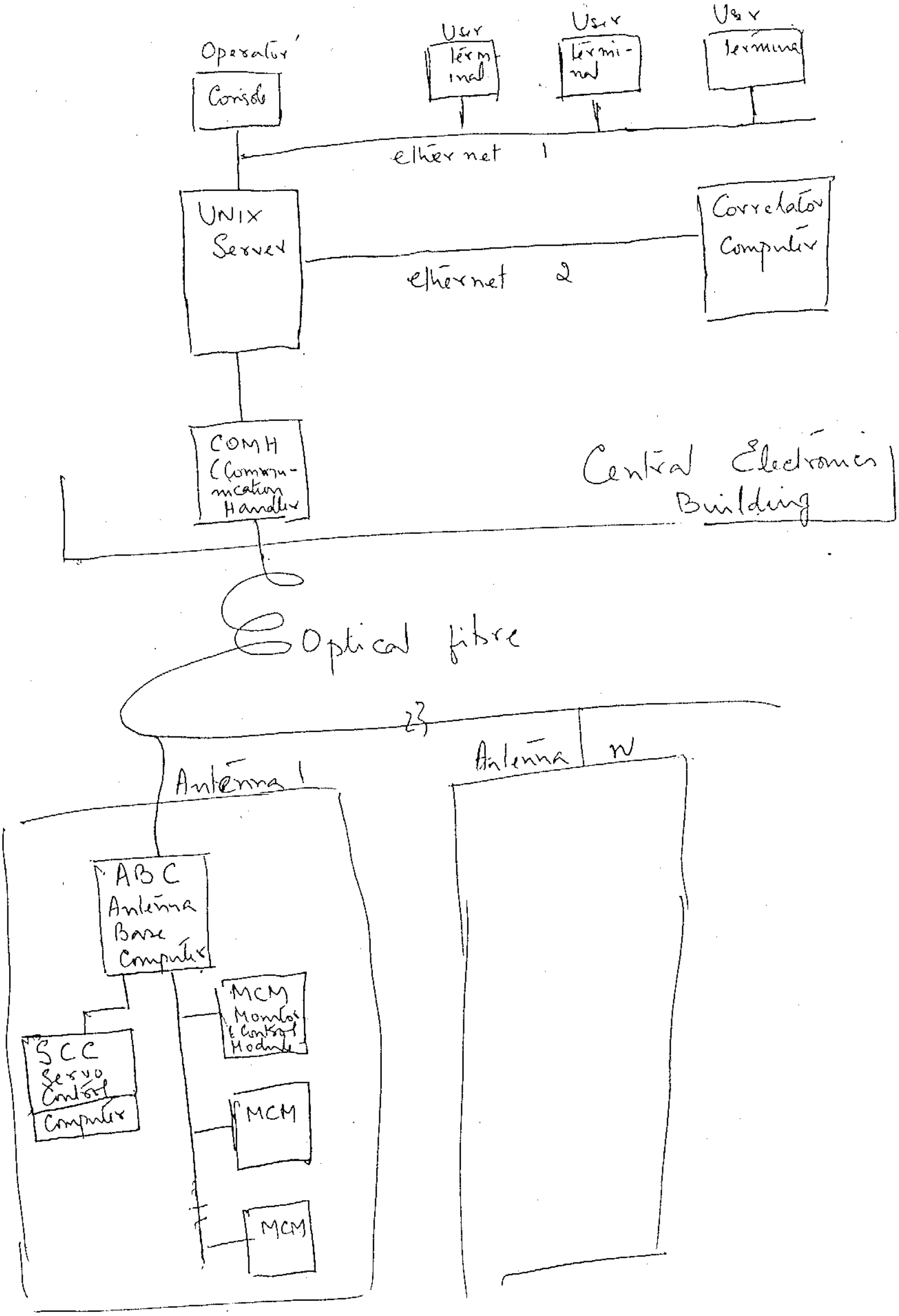

Fig 1