SDL Specification and Simulation of TDM/CDMA VSAT Integrated Service Satellite Communication Network

Jin Ye

Department of Electronics

Peking University, PR. China

Visiting Research in Department of Computer Science

Humboldt University, Germany

E-mail: jin@informatik.hu-berlin.de

Tel: 0049-30-20181235

Fax: 0049-30-20181234

March 2, 1995

Abstract

SDT'88, an implemented tool of SDL'88, is employed for description and simulation of TDM/CDMA VSAT Integrated Service Satellite Communication Network. A complete specification of this large scale system, which includes more than thousand network nodes and even more subscribers, has been reached. A dynamic simulator is generated and executed after syntax and semantics correction, through which theoretical design and logical data flow of this system are tested. During the same procedure, experience on SDL language and its application is developed together with discussions on related aspects.

KEYWORDs: SDL, SDT'88, VSAT, Integrated services, specification, instance, TDM/TDMA, CDM/CDMA, SDM/SDMA

Preview

This article is supposed to be the conclusion and report of my scientific visit in Department of Computer Science, Humboldt University, from Dec. 20, 1993 to end of 1994.

After I firstly arrived in Berlin, I got at once direction of Prof. Polze and help of other colleagues in System-Architecture group. As I tried at first to settle concrete research project, I began to know the research work in this group and further in other groups in this institute. In February, I gave a lecture to introduce my work in China to search for crossing point between my background and the advancing projects here. Wide opportunities have been offered by Prof. Polze and after several times of information collection and trial, I selected a project directed by Prof. Fischer on system analyze, 'Application of SDL in Communication System'.

SDL, Specification and Description Language, is a formal system specification language recommended by CCITT(now ITU). Compiler and simulation package are under development in System-Analyze Group of Department of Computer Science, Humboldt University. Under suggestion of Prof. Fischer, a topic "Application of SDL in Satellite Communication System" has been worked out to combine advancing project in group of Prof. Fischer and my earlier experience on protocol development in TDM/CDMA Integrated Service VSAT Satellite Communication Network. And SDT'88, an implementation package of SDL version 1988 developed by Telelogic Company of Sweden, is employed to realize full description and simulation for this large scale system. The results have been tested through static analyze and dynamic simulation and discussed in this paper. Experience has been gathered on application of SDL, especially SDT'88 tool, in specification and simulation of a large scale communication system, which is a reference for SDL implementation and application work going on in group of Prof. Fischer.

A lecture was given on this work at 11th of November. And this paper is to give more details,

Here I want to give my thanks to Prof. Polze and Prof. Fischer, who give me full support and direction, and also other colleagues in System-Architecture, System-Analyze and Network Management groups who have offered their help all through my work.

Contents

1 INTRODUCTION

SDL, Specification and Description Language, is CCITT (now ITU) recommended system specification language. After its first proposal in 1972, it has been continuously developed through the time. Version released in 1988 is fairly concrete for implementation of this language, although the one coming out in 1992 brings some new features, especially on service oriented programming. An implemented tool of SDL'88, SDT'88, developed by Telelogic Company of Sweden, is a compact software package including graphical editor, syntax and semantics analyzer and C code generator of dynamic simulation. Although there are still some restrictions in SDT'88 and not all features in SDL'88 are implemented or supported, SDT'88 supplies a basic platform for application of SDL in practical communication systems. Here, SDT'88 is employed for description and simulation of TDM/CDMA VSAT Integrated Service Satellite Communication Network.

TDM/CDMA VSAT Integrated Service Satellite Communication Network is a star topological centralized VSAT system. It is composed of a synchronous communication satellite, a central earth station called HUB and 1 - 2 thousands of VSAT user earth stations. CDMA multiple access scheme is used on links from VSATs to HUB, which is called Inbound links, while TDM multiplex scheme is applied on Outbound broadcasting link from HUB to VSATs. Apparently this is an unbalanced multi_scheme system. See Figure 1.

Basic unit of speed in this system is 2.4Kbps, in concerning to support as much capacity as possible with limited satellite retransmission power and microwave frequency bandwidth. And all throughoutput of other links are of times of this speed unit, for example, Outbound link of 153.6 Kbps, 64 times of 2.4Kbps, to be able to supply simultaneously 64 channels.

Two kinds of communication services are supplied in this network to user subscribers, packet switching data transmission service, with X.25 protocol access interface, and bitstream data relaying service through an end_to_end real_time transparent connection. The later one is supported by PBX form circuit switching to avoid processing delay. This is specially important for real time data transmission like vocoder service for talking. The signalling signals for setup and disconnection of bitstream service are carried as data frames of packet switching protocol for insurance of error free routing and relaying of these information.

In addition to these two services, administration protocol is also developed on base of a management center in HUB and monitor clients at each communication node. Operational information is collected, processed and stored by management center. Various manage steps can be carried through issues of command from management center to specified node. An operator interface is located at management center for informing and administration purpose. The infrastructure of this VSAT network is

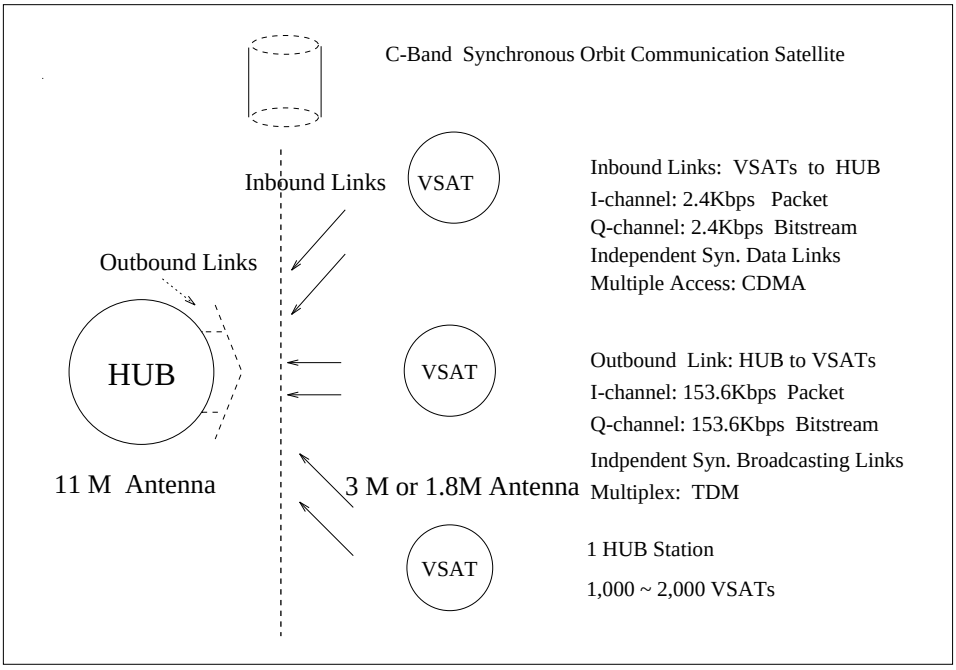

Figure 1: Configuration of VSAT network

informally described in Figure 2, which might be taken as a reference and comparison of later SDL formal specification.

In other chapters of this paper, a specification model in SDT'88 of this VSAT network are step by step introduced and discussed, while characters of SDT'88 implementation tool are not specially referred when it is not necessary. And full graphical specification of the system are supplied in Appendix A with tree list of structural information. In this model, service concept is applied for description of multi_task parallel operation mechanism. But as service concept has not been implemented in SDT'88 package, another subset using process concept as substitution of service concept is taken to generate C code simulation target file, which neglects the specification and simulation of multi_task parallel operation mechanism. This subset (mainly in VSATs references) is also supplied in appendix B.

Knowledge about SDL language can be found in [1] and [2], while [3] is the handbook for SDT'88, and service, infrastructure and protocol of TDM/CDMA Integrated Service Satellite Communication Network can be refered in [5], [4] and [7].

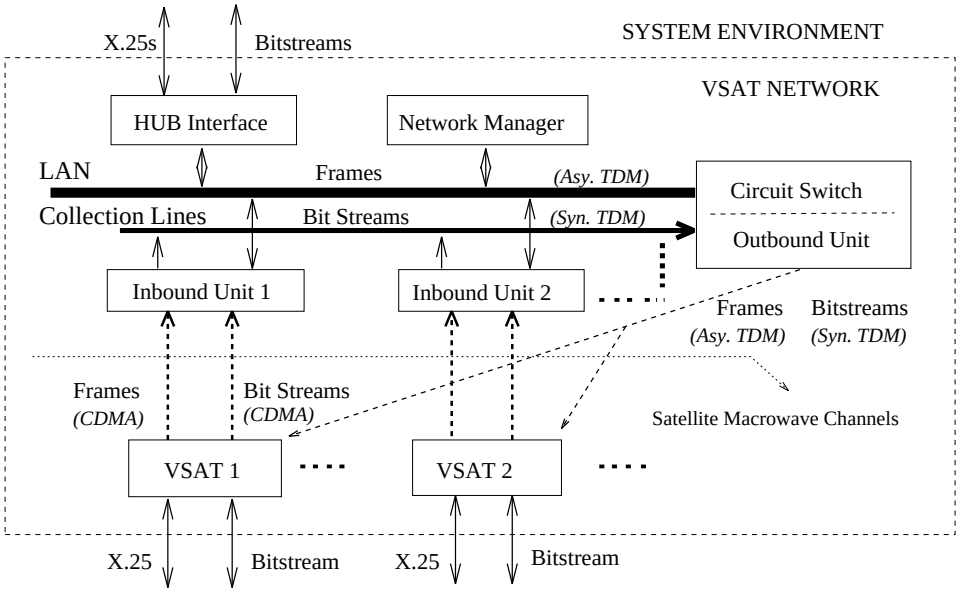

Figure 2: Informal description of system infrastructure

2 Infrastructure and general specification

2.1 Infrastructure and hardware/software interface

Infrastructure is the first concern in design of this communication system, in which hardware and software concepts are generally separated with interfaces to each other. There could be many different ways of infrastructure configuration and hardware/software separation, and hardware is settled at first while software has more flexibilities. Surplus hardware resources, such as processing speed of CPU, size of memories and input/outport ports, are prepared in hardware configuration to leave enough room for software development. In this VSAT system, general infrastructure are mainly determined by hardware parameters which are deserved from overall system specifications, such as available resources (power, frequency bandwidth, etc.), number of subscribers and quality of services (error rate, delay, voice evaluation, etc.). So topological structure with HUB and VSATs interconnected through CDMA Inbound links and TDM Outbound links and user service aspects (X.25, Bitstream, etc) are taken as background and preamble for SDL description. But this does not mean that the infrastructure of this system had been finally settled and might not be changed. In fact, one of the most important purposes for specification and simulation using a language tool like SDT'88 is to find out if system requirement can be fulfilled with presupposed hardware structure and properly developed software application. When answer is 'no', then new network infrastructure and hardware/software interfaces have to be defined to begin a new round of system specification and simulation.

With consideration mentioned above, it is natural to keep hardware infrastructure as preciously as possible with hardware/software interfaces as partitioning and

connecting points between structural concepts of SDL. And it is also reasonable to do this way concerning later work on system operational evaluation or further improvement, hardware or software.

At system level, HUB and VSATs are separated as two blocks with Inbound links and Outbound links for both services as interconnection channels. HUB and VSATs have also their own user interfaces represented as channels leading to and from system environment. These all agree with system infrastructure, and hardware/software interfaces are now channels carring signals between different structural concepts. See Figure 3.

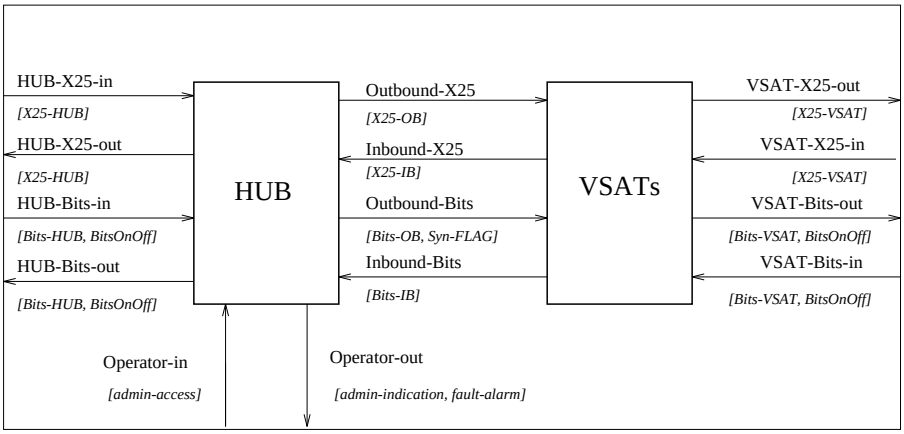

Figure 3: System structural specification

2.2 Structural reference of HUB

HUB station is the center of whole system. Physically or in protocol, all communication services must go through the corresponding switching center in HUB for routing, relaying and maintenance, no matter it is of packet switching, bitstream or administration, between VSAT stations and HUB or between VSATs. So different access modules for VSATs or HUB subscribers, switching centers for all services are located and interconnected in HUB. All these functional blocks are represented by block concept in SDL, and interconnection links by structural channels. Structural reference of HUB is SDL substructure HUB_modules. Graphical physical page HUB_modules(2) shows its remote reference. See Fig.4.And names of blocks in this figure are used later to refer corresponding functional blocks. It might be interesting to compare this graphical representation with the informal description in Figure 2, which is widely adopted in project files.

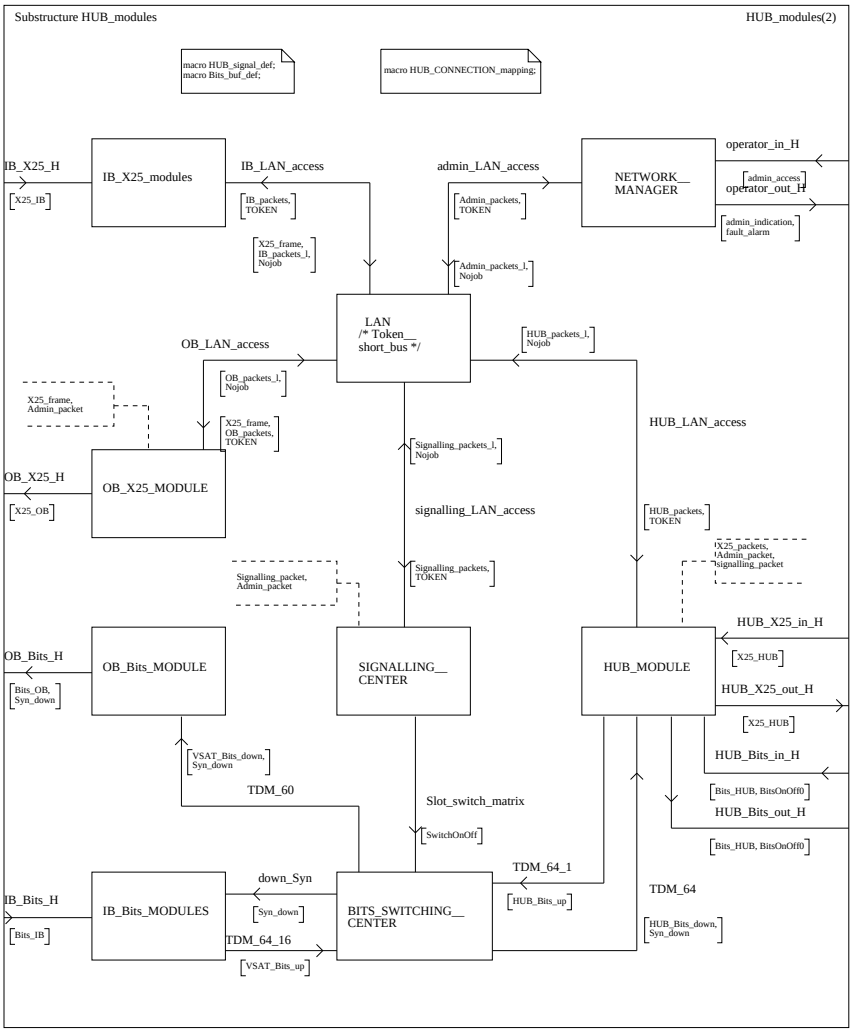

Figure 4: Structural specification of HUB

In this graphical layout, IB_X25_modules is of CDMA receivers for Inbound signals from VSATs into HUB. For each VSAT, there is a corresponding Inbound module, the CDMA demodulator. OB_X25_module is shared by all Outbound X.25 frame signals in asynchronous TDM scheme. HUB_module is access interface of HUB subscribers for both X.25 and Bitstream services. Signalling_center is responsible for all signalling process to keep and update connection matrix and send it to Bits_switching_center for circuit switching (slot switching) operation. And Network_manager is the management center of whole network, which exchanges administration information with monitor client of all communication nodes. All these functional blocks interconnect to each other in packet data form and are linked together through a LAN, Local Area Network, more concretely, a TOKEN short BUS, which is the only gateway for packet switching. See also informal representation in Figure 2.

In addition to this packet switching functional group, Bitstream switching group is composed of IB_Bits_module as CDMA Inbound receivers, OB_Bits_module as synchronous TDM outbound multiplexer and sender, and Bits_switching_center as circuit switching center, which operates similarly to PBX.

In this structural reference, each block represents an independent functional unit with particular hardware application, and all channels are physical links connecting these units. It reaches a agreement between HUB infrastructure and structural approach of SDL specification.

2.3 Structural reference of VSATs

At system level, all VSAT stations are generally represented as one block although they are many independent communication nodes. This is not only because that all VSAT stations play the same role in this system and have the same function, but also that it is impossible to setup explicit structural specifications for so many nodes. This is a necessary approach for specification of any large communication system. And some technical problems have to be handled within this '*simplification*' on structural specification, which will be discussed in the next chapter. The consideration is the same for further specification of IB_X25_modules and HUB_module, which have also many units of the same type.

3 Creation and interconnection of multi_instances

3.1 Dynamic representation and interconnection of instances

3.1.1 Dynamic representation by created process instances

As static or structural specification for all VSATs is impossible, the only mean in SDL is through dynamic representation, i.e. creation of many process instances of the same type to represent independent VSAT user stations with the same function. But as channels and signal routes can not be simultaneously created together with process instance, they keep to be static and it is undetermined with which one of process instances they are connected. So that interconnection routes can not be predefined through specifications of channels between blocks and signal routes between process types and must be dynamically setup after creation of related process instances.

3.1.2 Explicit routing through TO_ Process_Instance_Name expression

In this way a unique route is specified for current signal sending operation. But when considering a signal route with a process type having more than one process instances as receiver, the sender must specify identically the address of the receiver instance, no matter that they are in the same block or different blocks. This requirement conflicts with the structural concept of SDL and will bring much more complexities in system specification, especially in large scale system.

3.1.3 Relative routing (SELF, SENDER, PARENT, OFFSPRING)

All these PID type addresses of process instance are assigned through dynamical relations between related processes. They are temporary address names used by each process and may be refreshed with each output/input or create operation. But in connection oriented communication services, there is inner relation between signals from the sender to its paired receiver, and a stable link should be kept during the service. So relative routing, although it fits very well the structural concept of SDL, is not enough by itself. For example, as keyword SENDER points out the PID address of the sender of current received signal and can be used to send back response, the value may be updated by a new input from another signal route and not available further signal relay to the first sender.

3.1.4 Dynamic implicit routing

The addresses of receiver instances is known and can be stored by dynamic creator of them and viewed by sender through Import/Export implicit data exchange. In this way only a group of PID_type variables which have one-to-one relationship with the channel and signal routes between receiver and sender is to be defined, assigned during creation and viewed later without limitation from system structure partitioning (receiver and sender may be in different structural units.), or from process specifications (specification of process type can be freely renewed with little influence on addressing). For convenience in application, this one-to-one relationship could also be formally defined as attributes of channels and signal routes which connect different multi_instance process types. It is the same as that when process instances are created, attributes of corresponding channels and signal routes are also dynamically assigned for further routing. Or it can be said that the channels and signal routes are also dynamically 'created' together with its ending processing instances.

3.1.5 Absolute addressing

In this VSAT system, each communication node is identified with a unique network address. The simplest addressing plan is sequential numbering in natural number, and any other plan has a one_to_one correspondence with it. So it is essential to combine this address with the PID of its corresponding process instance in SDL specification and map network routing into inter_process_instance signal transmission. One point that should be mentioned here is that binary algorithm is not clearly specified in SDL'88, but it is very popular in both hardware and software application.

3.1.6 One more discussion

Relative addresses like SELF, PARENT, OFFSPRING and SENDER are basic elements supplied in SDL for dynamic addressing and routing. As a signal is simply sent to a process type with more than one instances, one of the instance will be randomly selected as receiver. So, there is not any provement for the next signal to

be sent to the same process instance and no stable link between sender and receiver. A particular rule can be defined to decide which one of the possible receiver instances of the same process type will be selected as receiver when a signal is sent to this process type without specifying an unique instance to be the receiver. For example, the signal will be arranged to the instance whose SENDER has the same value as the PID address of its current sender, or if there is not any one already existed, the creator of this receiver process will create a new instance. The signal is routed to it and its SENDER variable is although assigned. Then through a send_to_the_selected_one/send_response_back_to_sender, a call/response pairing, or maybe called as pre_training, a unique route could be setup and kept. This is very similar to DAMA (Demand Accordance Multiple Assignment) idea in communication system. Only the instance that has connection demand will be created and trained to be fixed on a link, and be stopped when this service round is over. This is likely to reduce the dynamic size of the simulator, just as it has done in communication system, especially in large scale system. But for this purpose unique PID type SENDER keyword has to be defined for each independent signal route to avoid unpredictable overlay.

3.2 Dynamic representation and interconnection in VSAT reference

3.2.1 VSAT_initializer and VSAT_nucleus

VSAT_initializer is the only active process when system is firstly started. It creates one VSAT_nucleus process instance for each VSAT station with a formal parameter which is the sequential absolute address of this station and is predefined through addressing plan of the whole system. The temporary PID address assigned with keyword 'Offspring' is stored up and then exported for other process instance, which will send signal to VSAT_nucleus instances of the same type, to view through import operation.

3.2.2 Interconnection between VSATs and environment

Environment represents here user subscribers of all VSAT stations. As one signal is sent from process env:1 to a VSAT_nucleus instance, it means a subscriber is sending a signal to its VSAT interface. To keep this one_to_one relation, a Space_division block is arranged for distribution of signals from env:1 to VSAT_nucleus instances and multiplex of signals visa verse. For this purpose the Division process in this block will firstly import and record the PIDs of all created VSAT_nucleus instances, which has been exported by VSAT_initializer. Each signal from or to env:1 should have additional addressing information to indicate to or from which of VSAT_nucleus instances or VSAT stations it is sent. Further discussion will be given in next chapter about data sort and signal definition.

3.2.3 Interconnection between VSATs and HUB

Another interface block is defined including Sender and Receiver process for both X25 and Bitstream services. Its routing issue is similar to that in Space_Division

block between VSATs and env:1 with the difference that outbound channel from HUB to VSATs is a single wireless broadcasting channel. This makes it possible for broadcasting service to send common information from HUB to all or a group of VSATs at the same time. So although the Receiver process needs also to know PID addresses of all VSAT_nucleus instances, it sends a copy of received information from HUB to each of them instead of specified one. This is also true in IB_X25_modules in HUB block, as all CDMA Inbound receivers can hear the mixed wireless signal from all VSATs and will later filter out the signal specified to it throu a 'demodulator' procedure. Although it is still not possible to describe or simulate the operation of CDMA demodulator and this procedure shows to an idle relay, it defines clearly structural concepts and interfaces, which is a necessary prepare for further simulation work, no matter it is done by refinement of SDL itself or introduction of other simulation tools. All these aspects can be seen in Appendix A.

3.2.4 Structural specification of VSATs

VSATs reference shows also to be a substructure composed of three blocks, See Figure 5.Block VSAT_U2 including VSAT_initializer and VSAT_nucleus, which is the functional specification block of VSAT station; block HV_Interface, interface between VSATs and HUB; and also block Space_division, the interface between VSATs and environment. As interface blocks, block HV_Interface and block Space_division can be replaced by channel substructure references of corresponding channels, and the later structural concept is closer to channel concept of communication system, although it might mean more structural partitioning pages due to more usage of substructure concept. And these two interface blocks can also be represented with process types, but this will break structural concept and compactness of VSATs specification in block VSAT_U2. And it means insolvable difficulties in further application or change of this specification package.

This custom is also discussed in SDL'88 recommendation concerning 'Consistent partitioning subset' concept. As there is a conflict between the principle that each structural concept should be treated as a 'black box' and no assumption should be made about its inner reference from outside and the fact that the sender process in one block must know 'something' about the receiver process in another block, at least the receiver exists and there is an identical type of signal link leading to it. For example, the multiple instances case in specification of this VSAT system. So interfacing parts of all structural blocks are related to each other and certain agreement should be kept by each independently specified block to maintain 'consistent partitioning subset' and 'consistent interconnection' between them. The interfacing structural concepts introduced above is one possible selection for this purpose, although further work should be done to testify if it is enough for general 'consistent partitioning subset' application and could setup a complete interfacing agreement in system view or not.

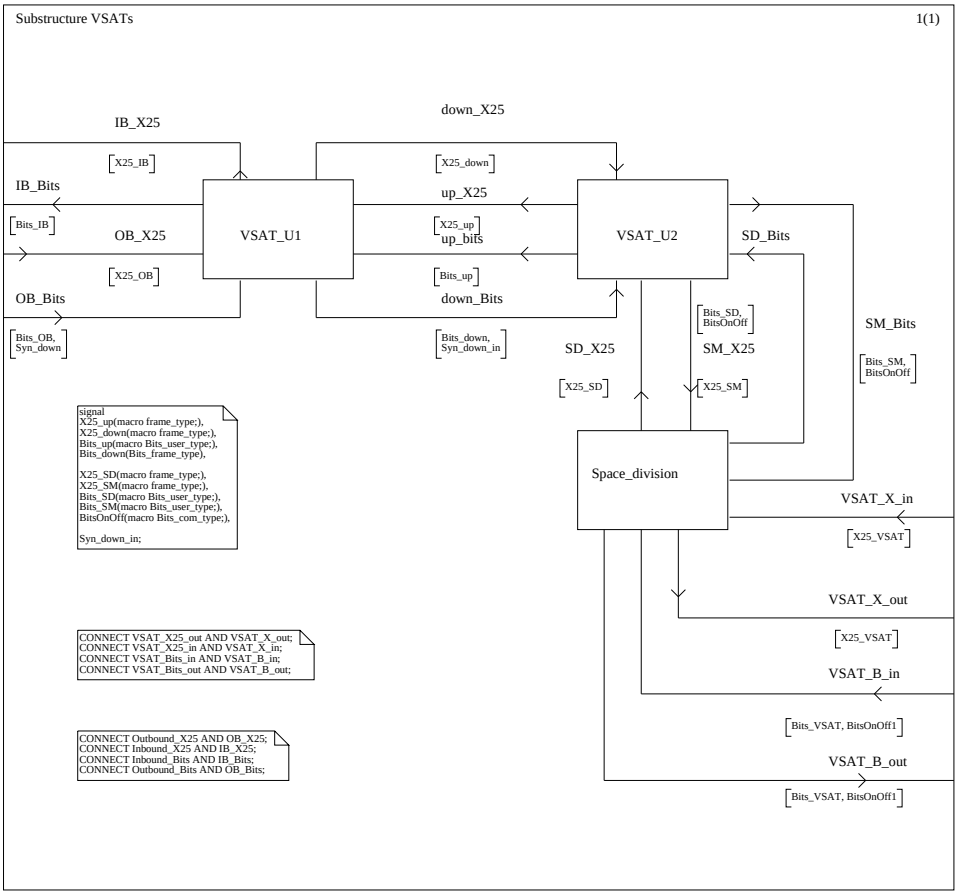

Figure 5: Structural specification of VSATs

4 Parameter, signals and variables

This VSAT network is composed of thousands of communication nodes, which is too large to be involved in one simulation code package. So formal parameters are used all through this specification and simulation effect can be reflected logically with a much smaller code package. See formal parameter definition in Figure 6, 3 VSAT users and two HUB subscribers are allocated in this model. At the same time, formal parameters of other units, such as number of Inbound receivers, are also relatively defined.

In this VSAT system, OSI layered structure is adopted to keep system open for interconnection with other communication systems, PSTN, packet switching network, ISDN, etc., in which X.25 is the direct reference for 3 lower levels of protocol. There are three types of higher level protocol on base of this X.25 platform, packet switching protocol (PSP), signalling protocol for Bitstream service (SP) and network management protocol (NMP). Basic protocol structure is shown in Figure 7.

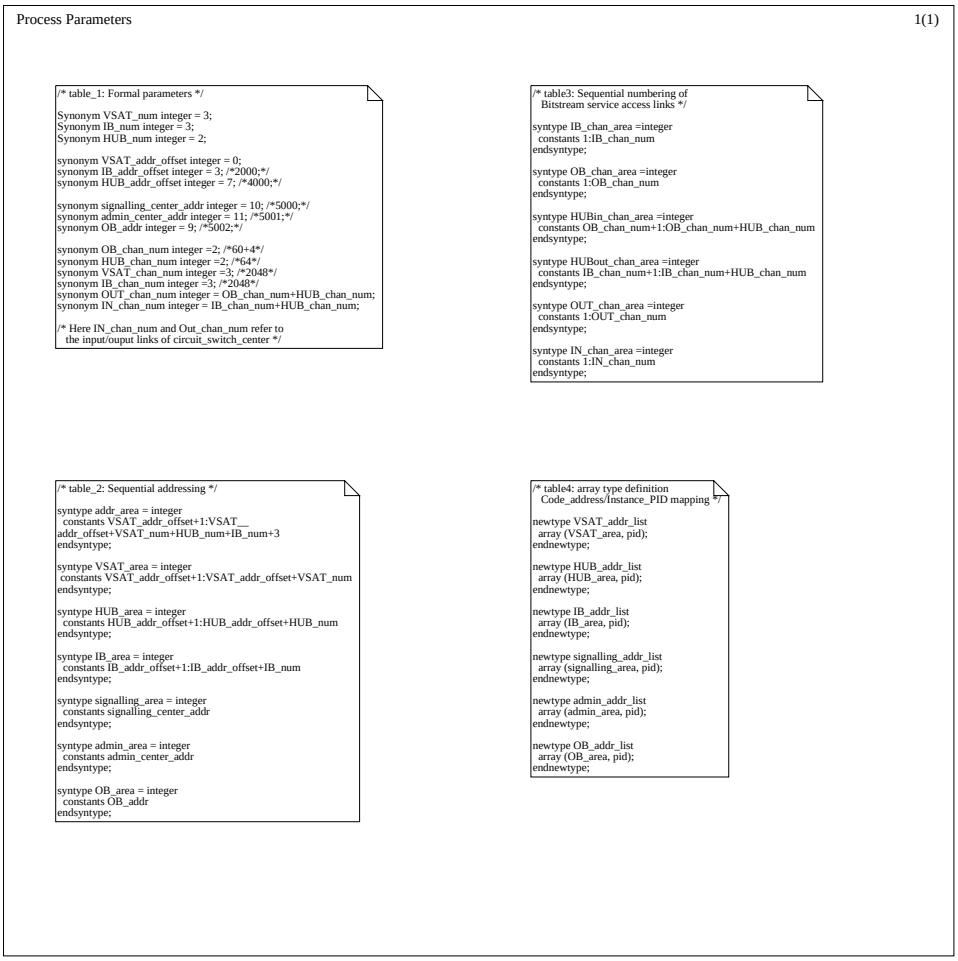

Figure 6: Parameter, signal and variable definition

4.1 Addressing and routing plan

The basic and simplest addressing plan, sequential natural numbering plan, is applied for addressing all communication nodes in system, VSATs, IB_X25_modules, OB_X25_module, HUBs, (different subscribers accessed through HUB_module), signalling_center and management center. Each is assigned one or a group of addresses. For example, for addressing of VSAT stations, VSAT_addr_offset is synonym type for address base or offset of VSATs, VSAT_num stands for the total number of VSAT stations. So, new data type VSAT_area is defined with integer value area from VSAT_addr_offset+1 to VSAT_addr_offset+VSAT_num, which covers all available and unique addresses for VSAT stations. Similarly, address boundary IB_area, HUB_area, etc., are also defined for other nodes or modules, and addr_area for all possible addresses in system. See Figure 6.

After this addressing plan is settled, a one_to_one relationship between address of each node and PID of its corresponding process instance is setup through definition of a PID sort array with address as index. See Figure 6. PID values are stored

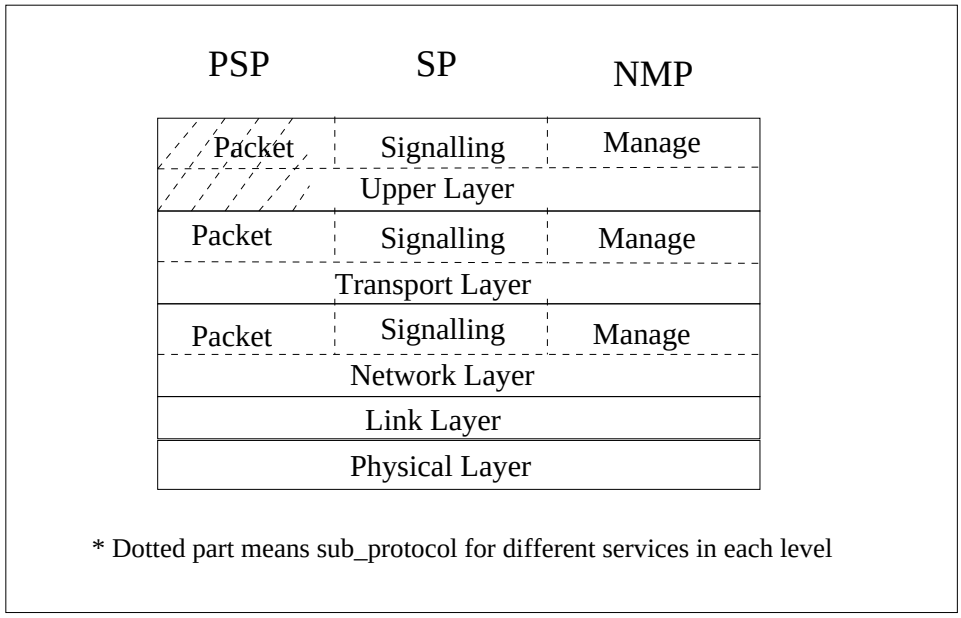

Figure 7: Structure of protocol in VSAT network

in arrays with the related unique node addresses as index by creators, which are various initializer of multi_instance process types, and then exported to possible senders. Each sender process type views PID arrays of its corresponding receiver process types through an input transition. So if a sender process instance knows the network address of the receiver under communication protocol, it can simply transmit this signal further simply with a TO_PID(address) output operation.

4.2 Numbering of Bitstream relaying links

Similar to addressing plan of packet switching service, links which carry Bitstream signals into and out of Bits_switching_center are also numbered this way for Bitstream service. The IN_chan_area includes sequential numbers of TDM collection links from IB_Bits_module to Switching_center and from HUB_module into Switching_center. And OUT_chan_area covers numbers of distribution links from Switching_center to OB_Bits_module and HUB_module. See Figure 6. While there are same number of links for both direction between Switching_center and HUB_module, there are only 64 outbound TDM links from Switching_center to OB_Bits_module and then later on Outbound channel from HUB to VSATs, in comparison to much more collection input links from IB_Bits_modules, one for each VSAT station. It is clear that Outbound channel is the bottleneck for circuit switching service or Bitstream service and the key factor of signalling protocol. See Figure 8.

4.3 Packet switching service and signal declaration

There are basically two types of packet switching service this VSAT system, communication between VSATs or between VSATs and HUB subscriber. Signal flow description can be found in Figure 9.

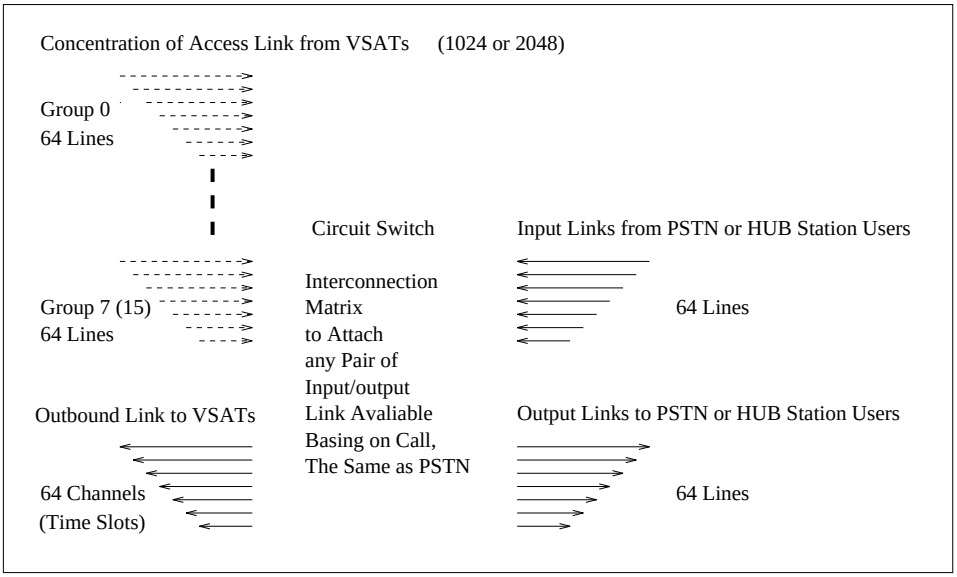

Figure 8: Configuration of Circuit_switching_center

Frame is the basic data structure for packet switching service, or more concretely, X.25 protocol. Different information units are carried sequentially in a frame as data elements. To consider simplest but sufficient application of frame signals in system specification and simulation, a variable list declaration is defined. As show in graphical text symbol in Figure 6, six sorts of element are declared as units of frame signal: frame_sorts, packet_sorts, addr_area, addr_area, info_type, and user_data.

This definition follows OSI layered protocol data structure. Frame_sorts is data link level unit and others are user data of this level. Sequentially, packet_sorts, addr_area (destination address), addr_area (source address), info_type (information of packet level) are packet or network level data units with the last one user_data as user information.

Frame_sorts, with possible values of X25_link_frame, X25_packet or ERR_frame, indicates that current frame is a data link level monitor frame, data frame or a frame with errors occurred during transmission. It is the processing object for Data Link level protocol. Here the error control and flow control aspects of Data Link level protocol are neglected which should have been respected through further defined data units.

Packet_sorts separates general packet types in this system with literals packet, admin or signalling. Destination and source addresses of addr_area type are carried by each frame for routing and addressing purpose. Only interruption data packet is concerned here to avoid details of X.25 protocol specification, which is not main issue here in system description. Info_type unit gives more information about the function of current packet with possible values: conn, disc, indi, data, etc.

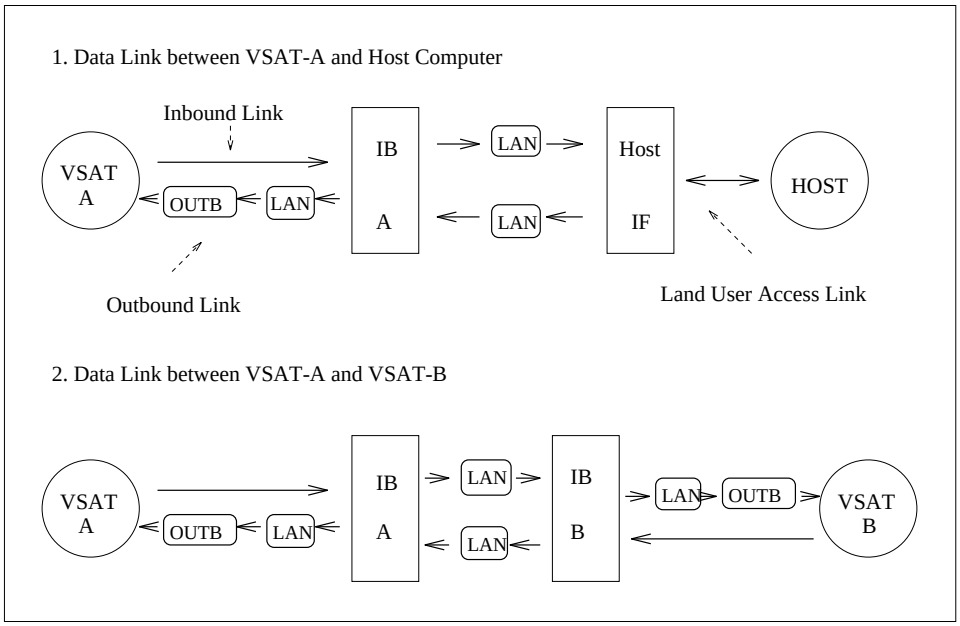

Figure 9: Signal flow of packet switching service

The user_data, a charstring type data unit, is carried transparently through the network between user subscribers, which are taken as environment of SDL specification.

It can be seen from frame signal declaration of packet switching service, signalling and administration packets are carried as higher level user data of packet switching service. Here they share packet_sorts and Info_type data units for simplicity and convenience.

There are other patterns of signal definition or variable declaration in this VSATs system model. Each one can select his own preference. This signal definition or variable declaration procedure will influence all further specification work, it is suggested to keep clear protocol structure instead of searching the most suitable candidates for current research step.

4.4 Bitstream service and signal declaration

Two kinds of signals are involved in Bitstream service: signalling signal BitsOnOff* for setup and clearance of Bitstream service, and Bits_user, the signal that will be relaid transparently through the network after setup of end_to_end link.

Signal BitsOnOff carries routing message for connection management, addr_area (destination address), addr_area (source address) and info_type for signalling command and confirmation. Signalling protocol has been clearly described and simulated in this system application. A simple call/reponce link setup procedure can be seen in Figure 10.

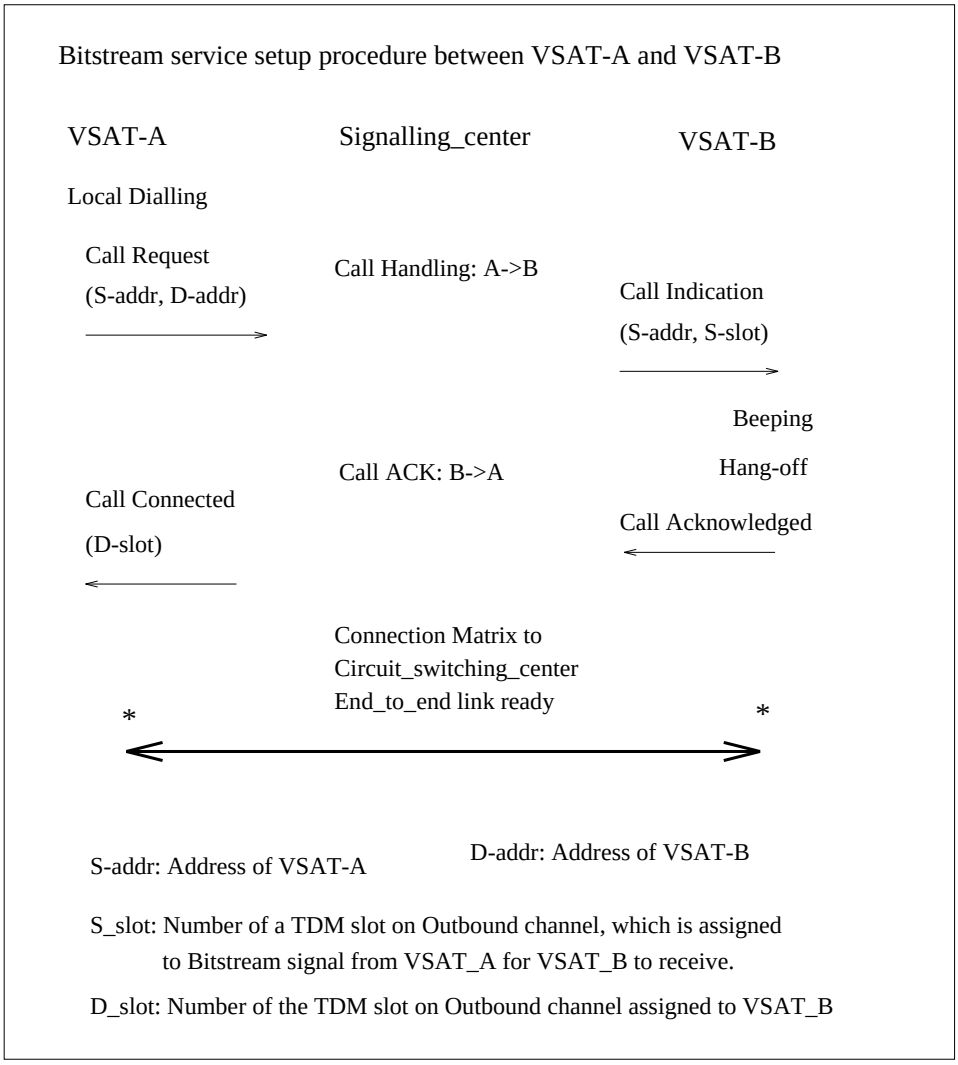

Figure 10: Signalling procedure of Bitstream service

Although destination address is not any more needed for Bits_user signal as an end_to_end link has been setup and kept, source address is still necessary for sender such as env:1 to address corresponding receiver process instance. In addition to that, a character type unit of Bits_user represents user data of Bitstream service. As there is no specified data unit defined for D_slot and S_slot, (see also Figure 10, User_data unit in signalling packet is exploited as temporary carrier, in case only two Outbound channels for Bitstream service are supplied in this mini model.

4.5 Administration service

Only information collection operation is described and simulated in this application. The network_manager sends out in period command packets to poll all created nucleus instances representing operating communication nodes. And each polled object will send back a report frame to network_manager. And when a new poll/report ope-

ration has been finished, a indication is given to env:1 through Operator_Interface.

5 SDL specification of parallel operation mechanism

5.1 Multi_task parallel operation mechanism

Multi_task parallel operation mechanism, just like the multi_task operational system on single processor platform, means the parallel execution and sharing of resources of independent communication procedures or subscribers. It is adopted in software configuration of all communication nodes in this system. With the help of this mechanism, independent services of different user subscribers can be supported in a single hardware platform to reduce size of equipment for higher efficiency. Meanwhile an on_line monitoring client can be conveniently attached to each node for management purpose and error alarming. A self_recover procedure can be automatically stirred up when fatal error has occurred.

Hardware interruption is applied for switching between procedures, which alternate to be either active or sleeping (current working area protected, no execution, waiting for being waken up by new interruption to become active). A local TOKEN, the sequential number of interruption, controls and indicates which procedure will be at active state. At any time, except that for interruption operation (Parallel Control), there is one and only one procedure to be active.

5.2 Multi-task specification using service concepts in SDL

Using service concept, different application procedures are services in a process instance. The communications between the services are statically defined in process reference and all services share the single process instance PID address, without additional problem for other process which sends signals to this process (the sender can simply use OUTPUT_TO_PID(address) operation, and the only one service will be identified as receiver with help of signal_route definition). It is clearly defined in SDL standard that 'Service signal routes are allowed only when signal routes are specified in the enclosing block'.

Priority input and output concepts available for signal exchange between service instances fit very well to interruption control mechanism. Otherwise normal input and output will be delayed by other signals and can not act as 'parallel control signal carrier' to avoid 'pile_up_result', which means 'the one have many gets more, while the one has few gets less'. And the Import/Export data exchanging has no synchronization mechanism between the sender and receiver, when to export has no restriction on when to import. This is also an important point that state concept and signal Input/Output can not always be well substituted by variable definition and exchanging.

The working area protection for interruption operation is now carried by current states and local variables of each service.

Although hardware and software interruption are basic issues in communication system development, they are not treated specifically in SDL. As priority signal might describe some hardware interruption application and Timer acts somehow like a software interruption trigger, they are far from being enough for specification of practical interruption concepts such as multi_priority interruption, interruption loop or insert, etc. which are necessary concern for almost all communication project, especially real time or parallel processing system. It could be foreseen that more related concepts and protocols would be introduced in SDL as complement.

5.3 Parallel control in VSAT_nucleus, an example

In VSAT_nucleus process specification, as shown in Figure 11, five services Link_1, Link_2, Packet, Signalling and Admin are referenced to represent five parallel operation procedures, while Bits_server is the interface for signalling operation and is practically of hardware. In actual situation, various subscriber accessed to the same single processor node are treated as different parallel operation procedure instead of different protocol package. But as here only one subscriber is accessed to each VSAT node in this model, service representation introduced above are developed for functional simulation. Service Parallel_control is the controller of this mechanism. Just as interruption aspect, a Timer TOKEN_clock is periodically set in Parallel_control as input to start a operation switching. When this Timer input firstly comes, Parallel_control sends a NULL TOKEN to all services to force them into 'sleeping' state for start synchronization. Then the first service in list, Link_1, will be given a TOKEN equal to its sequential number to turn into 'ready' state and be alive. At next time of Timer input, a NULL TOKEN is sent to Link_1 to alternate it into 'sleeping', then the TOKEN is renewed to be equal to sequential number of service Link_2 and then sent to it and Link_2 becomes the current alive procedure. As TOKEN cycles further in round, all services operate in a parallel mechanism.

6 Specification of multiplex/demultiplex schemes

Three different types of multiplex/demultiplex scheme have been employed in this VSAT network, TDM/TDMA through time separation, SDM/SDMA through dedicated physical medium, and CDM/CDMA though orthogonal distribution of power. Another general multiplex/demultiplex scheme is FDM/FDMA, which separate through frequency band, and is very similar to SDM/SDMA. A simple pattern about these schemes is shown in Figure 12.

6.1 TDM/TDMA, time division

Whole SDT'88 described system runs on a single time standard, different processes operate in parallel, and time consumed by transitions and signal transmission are thought to be none. All these prove that each process can execute on step of an exact clock and simply keep synchronization with any other process instance. These

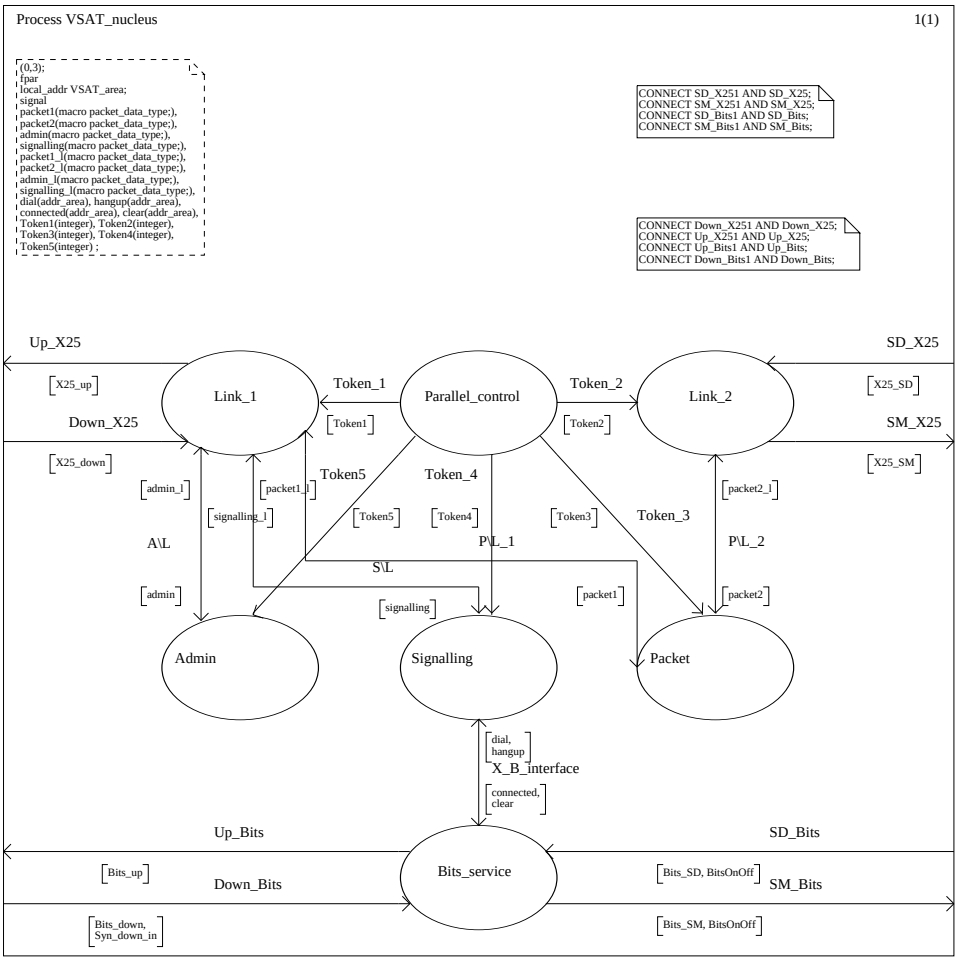

Figure 11: Specification of parallel operation mechanism in VSAT-nucleus

are all synchronous TDM concept. And signals are lined up in endless queues and sent in asynchronous TDM form.

A perfect application of this TDM/TDMA feature of SDL specification in this system is the synchronous relaying links of Bitstream service. A Timer Syn_clock is repeated in Bits_switching_center as frame synchronization flag for the whole system. Each time, when this Timer input comes, it is scattered to all process instances which send or receive Bitstream signal, for example, IB_Bits_module and OB_Bits_module in HUB, VSAT_Bits_sender and VSAT_Bits_receiver in substructure VSATs, etc.. And all these process instances work their sending and receiving on steps of this flag. As transmission delay and waiting delay in queue are treated to be null, a exact synchronization can be kept all around the system very easily. This is the key factor for TDM/TDMA scheme and it is proved by complicated synchronization equipment in practical system.

For in SDT'88 transmission of signal are considered as no delay, which means an unlimited speed, there is no modification of signal route or channel reflecting

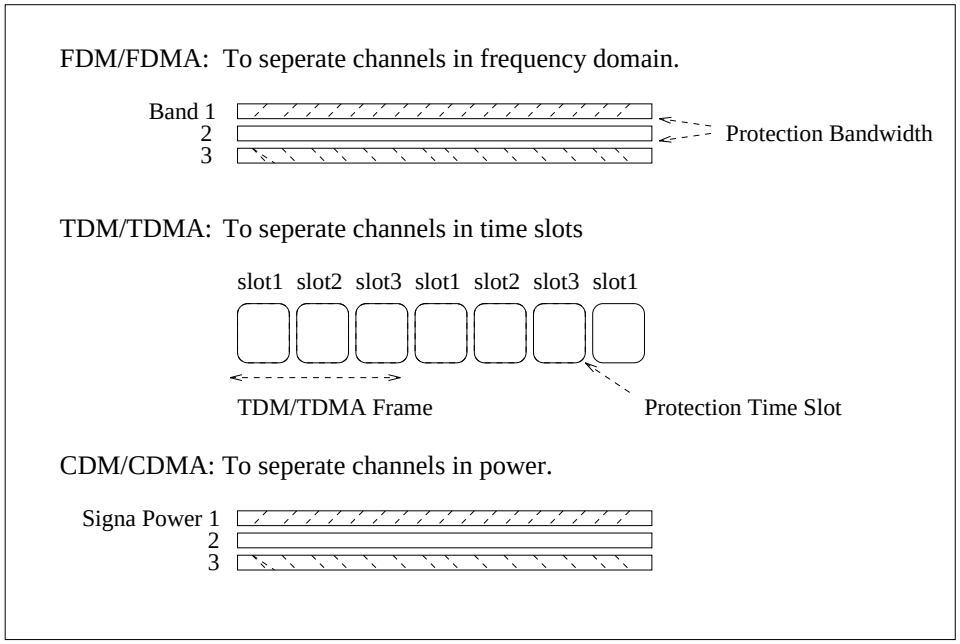

Figure 12: Description of 3 types of multiplex/demultiplex schemes

capacity factor. And certain tricks have to be taken in protocol process to reach capacity control, which is a key fact for communication system. For example, although countless Bitstream signal might be sent without time restriction to Circuit_switching_center, the switching protocol has to control that user_data can only go through connected links that have been setup through signalling procedure.

6.2 SDM/SDMA, space division

In VSAT system specification, SDM/SDMA is represented with structural concept and creation of process instance. This topic has been detailly introduced in chapter 2, where multi_node and multi_instance issue is dealt with. In other words, the problem is to represent SDM/SDMA scheme in TDM/TDMA mechanism of SDL specification. And an additional informational unit is necessary to setup a one_to_one correspondence between a TDM/TDMA signal and its represented SDM/SDMA one, such as the sour_addr(source address) unit in packet switching service signal definition.

6.3 CDM/CDMA, code division

CDM/CDMA is the orthogonal power distribution multiplex/demultiplex scheme. Different signals may coexist in the same frequency band at the same time. Existence of each signal means interference to other signals in channel. And this interference in determined by power distribution of all signals, phase difference, and correlation factors between random code of each signal.

As more than one signals may exist at the same time and be simultaneously received by receiver, CDM/CDMA is somehow similar to SDM/SDMA scheme, in comparison with TDM/TDMA scheme. For the same reason, it has to be turned into TDM/CDMA representation on channels and signal routes. In IB_X25_module. a single instance process type Power_distribution is receiver of all CDMA signals from VSATs to HUB and broadcasts each signal it has received to all the IB_link process instances, which are of corresponding IB_nucleus to VSAT_nucleus. After a demodulation procedure, only the unique IB_nucleus instance specified by routing and addressing information in this signal can filter out this CDMA signal deserved to it and the others will discard it. This is exactly the working algorithm of CDMA demodulator. As phase difference between signals is an essential factor for simulation of CDM/CDMA scheme, it is very difficult to use SDT'88 for its specification, although it is not impossible. Here in this model CDM/CDMA scheme is treated just as SDM/SDMA, except that each receiver will 'hear' all signals in channel instead of the dedicated one as in SDM/SDMA case.

In this system specification, importance of looking for precious specification for different multiplex/demultiplex schemes seems not to be so obvious. But if further simulation work is concerned to get more meaningful results, such as concerning transmission through certain physical medium condition, these approaches prepare clear, well defined interfaces to dynamic simulation librarys which simulate performance of physical channels.

7 Specification of LAN

The Local Area Network is composed of the process instance LAN_nucleus in block LAN and LAN_access processes in each block connected to block LAN. Although in HUB substructure specification there are one dedicated bidirectional channel between LAN and each connected block, they are actually part of one single physical BUS line with LAN block as the transparent joint point. This means that at any time there is at most only one signal to be carried on these channels and in LAN_nucleus process. Each signal is switched by LAN_nucleus from the first channel between sender and LAN_nucleus to the second channel between LAN_nucleus and receiver. This dynamic bridging functional LAN is the packet switching center (PSE).

To avoid collision of signals on BUS, a cycling TOKEN is controlled and distributed by packet switching center, LAN_nucleus. In LAN_nucleus, a Timer TOKEN_clock is set periodically to drive the gate signal TOKEN. The TOKEN takes sequentially in each period of TOKEN_cycle valid gate value for one of the nodes attached on LAN BUS, IB for IB_X25_modules, OB for OB_X25_module, HUB for HUB_module, SC for signalling_center and AC for management_center. When a node gets its valid TOKEN gate value from LAN_nucleus, it can send the firstly buffered frame in its queue to LAN_nucleus for switching. If its queue is empty, a Nojob indication will be sent to LAN_nucleus. See Figure 4.

There are two different algorithms for LAN_nucleus to handle this Nojob indication. It may neglect it and wait simply the next TOKEN_clock input. This is fixed length TOKEN arrangement, as fixed length of time, a period of TOKEN_clock, is arranged to each node no matter it has demand for packet switching or not. Otherwise LAN_nucleus will renew the value of TOKEN to valid gate value of the next node on BUS and send to it. This is the demand accordance arrangement with higher throughout. See Figure 13. But when TOKEN has finished a round of cycle, and is beyond the valid gate value of the last node on BUS (now it is management_center), it will be kept in LAN_nucleus for one cycle. This is because that physically when signal reachs end of BUS, it must be consumed by a load to avoid electrical reflection, which may cause shivering of TOKEN_clock and unstability of BUS, just as one period of TOKEN_clock is consumed in LAN_nucleus in this specification.

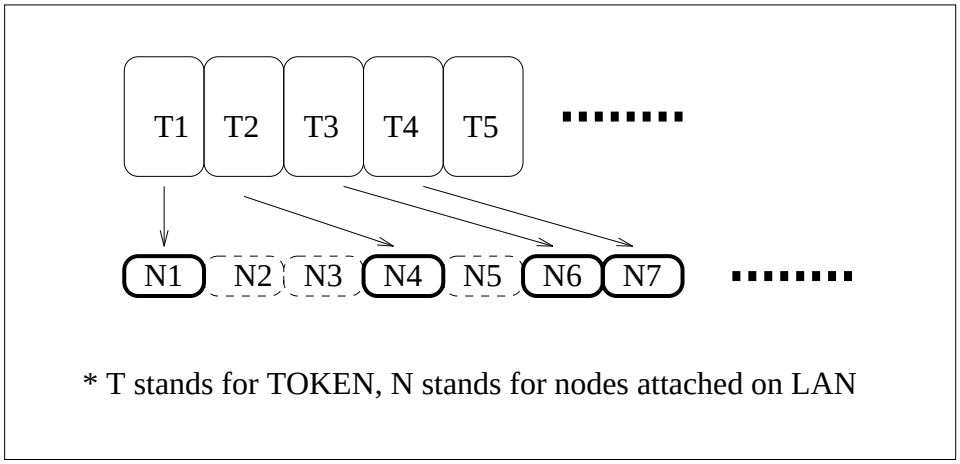

Figure 13: Demand accordance TOKEN arrangement on LAN

Here one more time it is shown the fitness of SDL application for a synchronous system without delay. This is the physical character of this short BUS LAN, whose propagation delay can be ignored. But for other LAN types in which propagation delay plays a key role, as in CSMA style LAN, TDM synchronous feature of SDL specification might no longer be an advantage.

8 System simulation

Within SDT'88 compact tool, after graphical edition, MACRO expansion and syntax, semantic check, a C code simulator is generated and then dynamic simulation is executed. Specification is proved to be correct and compact. Packet switching service, Bitstream manipulation and relaying service, administration polling service are tested. TDM/TDMA transmission feature is observed in global time standard. And proper interfaces are reserved for none TDM/TDMA schemes, SDM/SDMA and CDM/CDMA, and ready for further development.

Following aspects in SDL specification and SDT'88 simulation tool are discussed.

(1) Time consumed by transitions and transmission delay of signals should be taken into account for real time system simulation, considering that no delay transitions may cause endless loop for correct logical design. Timer input can not exactly reflect time delay while it is influenced by other signals. Timer with priority could be better but will add complexity considering that there are different delay for different decision choice and different signals. So it might be suggested to define a type of parameter for transition, signal input/output operation, and signal route or channel concepts to modify the time consumed by these processing objects.

(2) Statistical results, such as throughoutput and delay, are important aspect of system simulation, and new data processing features (functions, libs, etc.) should be developed also in SDL or its implemented tools.

(3) Consistent partitioning subset is a importment aspect of SDL and substructural concept is adopted in VSAT system specification. But this concept is not well supported in SDT'88 package. In SDT'88, only the biggest (with most leave blocks) configuration is taken to generate C code simulator without choice for user, although the other branches which are of 'no use' may also make problems during edition and analyze procedure.

(4) Macro concept is very useful in simplification of SDL representation. According to macro definition in SDL, a macro is a textual or graphical substitution of certain part of specification with one or more than one inlet or outlet points. In SDT'88, only one inlet is allowed. At the same time, although more than one outlet can be defined and labeled with unique names in macro definition, errors occur during analyze if more than one outlets are used.

(5) Service concept is not implemented in SDT'88 C code package. So final simulator of VSATs is of process description and multi_task parallel operation mechanism is neglected. As show in Appendix B.

(6) MSE is a direct way for monitoring and adjusting simulator. But the graphical output is apparently too large to be read conveniently, on screen or on paper. Even for this mini simulation package, it consists over 30 coexisting process instances.

Conclusion

A whole procedure has been gone through using SDT'88 tool for SDL specification of TDM/CDMA VSAT Integrated Service Satellite Communication Network. Graphical representation, MACRO definition, Syntax and semantic analyze are errorless passed to generate a C code simulator. Designing plan are proved theoretically and logically correct through dynamic simulation.

This experience shows that SDL is a powerful language for specification of communication system, especially synchronous system, and SDT'88 is a well structured development tool even though it has some restrictions and unimplemented features.

9 Appendix A

Note: Due to limitation on size of this paper, Appendix A and Appendix B are printed as an independent copy.

10 Appendix B

Note: Due to limitation on size of this paper, Appendix A and Appendix B are printed as an independent copy.

References

[1] ITU. Recommendation Z.100, Geneva, 1993.

[2] Ferenc Belina, Dieter Hogrefe and Amardeo Sarma. "SDL with APPLICATIONS from PROTOCOL SPECIFICATION", *Prentice Hall Publication*, 1991.

[3] TeleLOGIC Company, Sweden. "SDT2.2 Reference Manual", Volume.1 and Volume.2, 1992.

[4] Jin Ye, Xuqiang Liao and Haige Xiang, "Protocol of TDM/CDMA VSAT Integrated Service Satellite Communication Network", *Paper to ICCT'92*, International Communication Conference Technique, Beijing, China, 1992.

[5] Huisheng Chi and Haige Xiang, " TDM/CDMA VSAT Integrated Service Satellite Communication Network", *Paper to ISCS'90*, 6th International Satellite Communication Symposium, Nanjing, China, 1990.

[6] Jin Ye, " Application of SDL in Communication System " *Paper to PSTV'95*, International Conference on Protocol Specification, Testing and Verification, Warsaw, Poland, 1995.

[7] CCITT Recommendation X.25, International Standard Blue Book, 1988.

[8] Joachim Fischer, "Contributions for the Formal Specification of the ODP Trader using SDL'92 and ASN.1", *Informatik-Bericht Nr.37*, Department of Computer Science, Humboldt University, Berlin, Germany, 1994.