Importance of Specifications in Designing of Protocols

Samiksha Nikam[1] & Dr. B.T. Jadhav[2]

Abstract: **New demands by advances in communication technology and wide use of distributed, net-based, and mobile applications make protocol design more complex. While designing a protocol two types of errors are unavoidable i.e. designing an incomplete set of rules or designing rules that are contradictory. Hence it is necessary to design both complete and consistent set of rules. It requires to specify very precisely all the specifications required to design a protocol. In this paper we focus on specification design parameters of a protocol. This is a most important step in protocol designing. If specifications are not designed properly then protocol designing may lead to wrong designing.**

Keywords :**protocol specification, service specification, formal method, informal method, protocol vocabulary.**

1. INTRODUCTION

Protocol is a set of rules governing data exchange between peer entities within a layer. Protocols are used for communication between entities in a system. Entities use protocols in order to implement their service definitions. The key elements of a protocol are Syntax and Semantics. Syntax includes time data formats and signal levels. Semantics includes control information and error handling.

Protocol specification description plays important role in all stages of protocol design. Protocol description comprises service specification and protocol specification [8]. Service specification contains services provided by layers of distributed communication system. Protocol specification provides operation of each entity within a layer. Specification design phase of protocol design allows the designer to prepare an abstract model of the protocol for testing and analysis.

Basically Informal and Formal methodologies are used to design protocol. In informal methodology specifications are designed using natural language.

Because of this blurred specification or ambiguity present in the design [10], [11], [13], [16], [19]. In this method protocol is designed with step by step approach. At each step it is checked, verified and accordingly specifications are changed till programmer is satisfied or correct protocol is designed. In this case there are chances of designing wrong protocol due to changes in the specifications at each stage. [12], [15], [19].

In Formal design method informal specifications are converted into formal specification and it is simulated and validated with a tool. In designing phase it is possible simulate the model in order to find errors. In formal method chances of designing correct protocol is more compared with Informal method because automated tool can be used for design and implementation. Formal methodology is very useful for debugging the specifications. Hence Formal designing technique is suitable methodology to design protocol. Still specification designing is user or human sensitive. [11].

2. INTERFACES AND SERVICES

The function of each layer is to provide services to the layer above it. The relationship between interfaces and services are shown in fig1.The active elements in each layer are often called entities. Entities in the same layer are called peer entities. The entities in layer n implement a service used by layer n+1.layer n is called service provider and layer n+1 is called the service user. Services are available at service access point SAPs. The layer n SAPs are the places where layer n+1 can access the service offered. Each SAP has an address that uniquely identifies it. Two layers agreed upon a set of rules to exchanged information between them. At an interface the layer n+1 entity passes an IDU (Information Data Unit) to the layer entity through the SAP. The IDU consist of some control information and SDU (Service Data Unit).the SDU is the information passed across the network to the peer entity and then up to layer n+1. Specifications designing are depending upon services provided by the lower layer to the upper layer.

1. College of Computer Application for Women, Satara.
nikam_samiksha@rediffmail.com

2. Department of Electronics and Computer Sciences, Y.C. Institute of Sciences, Affiliated to S.N.D.T. university, Mumbai ,India- Affiliated to Shivaji University, Kolhapur, India
btj21875@indiatimes.com

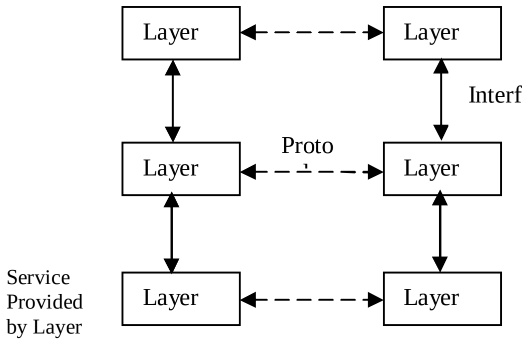

Fig1. Protocol Layers

3. SPECIFICATION DESIGN

This is an important stage in protocol designing. Specification decides input output behaviour of protocol. Specifications are designed according to user requirement. It must be easy to understand and expressive enough to avoid ambiguity [11]. A protocol specification consists of five distinct parts [7], [17],

- Service specification.
- Assumptions about the environment.
- Protocol vocabulary.
- Message format.
- Procedure rules.

The fifth element of specification is the most difficult to design and also hard to verify [7].

Specification designing is challenging task. Because initially we have to specify that what the protocol does without constraining the implementation, detailing the services that it provides. The services, legal sequences of message exchanges, and the behaviour under all exceptional conditions must be defined in such a way that there can be no semantic ambiguity [20]. Specification designing is categories as service specifications and protocol specifications.

A.The concept of service and protocol

Services and protocols are particular architectural concepts which are commonly used in the design and implementation of communication systems. [7]

A service is a set of operation that a layer provides to the layer above it. Service relate with interface between two layers. A Layer reports to upper layer entities on an action taken by its entity using primitives as well. The primitives and their parameters are defined as Service Data Units (SDUs).

Services can be categorizes as connection oriented and connectionless. Services between adjacent layers are expressed in terms of primitives and parameters. Primitive specifies the function to be performed. Parameters express Data and control information. Service primitives can be categorized as given below,

1. Request primitive issued by a service user to invoke some service and to pass the parameters needed to fully specify the service request [1].

2. Indication primitive issued by the service provider to:

(a) Indicate the primitive has been invoked by the peer service user on the connection and provide associated parameters [1].

(b) Notify the service user of a provider-initiated action [1].

3. Response primitive issued by a service user to acknowledge or complete some primitive previously invoked by an indication to that user [1].

4. Confirm primitive issued by service provider to acknowledge or complete some primitive previously invoked by a request by the service user [1].

Protocol is a set of rules deciding the format, syntax and semantics of packet that exchanged by peer entities within layer. Entities used protocol to implement their service definition. Protocols can be changed against service. But services cannot be changed. Service defines operations that can be performed on object but does not provide implementation details. Protocols are invisible to the user [1]. From this point of view, we mentioned that the stability of service specifications is more important than the stability of protocol specifications. If a protocol is changed within a given layer, the effects are limited to the layer, while a change in a service specification may have consequences that will propagate upwards and affect several protocols [4].

B. Service Specifications

User requirements are converted into service specification. Service specifications are specified as a set of primitives available to user process to access the network service. Some basic service primitives are LISTENING, CONNECT, SEND, RECEIVE, DISCONNECT etc. [1]. Service specification relate with interface between layers. Lower layer is service provider and upper layer is service receiver. The passing information and data down through the layers of sending device and back up through the layers of receiving device is made possible by an interface between layers. Service specification for particular layer can be changed without changing surrounding layers [2].

C.Protocol Specifications

To reduce design complexity networks are organized as a stack of layers. The number of layers and function of layers are different from network to network. Layer n on one machine carries on a conversation with layer n on another machine. The rules and convections used in this data transformation is collectively known as protocols. The entities comprising the corresponding layers on different machines are called peers. Peer communicates by using protocols [1].

Protocol specification describes operation of each entity within layers in response to commands from its users, messages from other entities. [11]. It includes a list of type and format of message exchange between entities, rules regarding reaction of each entity to user commands and message from other entity [11]. Protocol specification contains both control and data processing function. Data processing consist of encryption, coding algorithm, error correction etc. control functions are of two types i.e. real time and data dependent [10].

Service and protocol specification are based on user's perception. It is based on user or human experience and highly prone to human errors. Such languages can be easily understood using fuzzy logic [18]. Hence we propose specification design should be use fuzzy logic for better result.

4. ASSUMPTIONS ABOUT THE ENVIRONMENT

The environment in which the protocol is to be executed is different for various type of network. for example for the file transfer service it requires minimum two users and a transmission channel. the user can be assumed to simply submit a request for file transfer and await its completion. the transmission channel is assumed to cause arbitrary message distortion, but not to lose, duplicate, insert or reorder messages [7].

5. PROTOCOL VOCABULARY

It defines three distinct types of messages: ack for a message combined with a positive acknowledge, nak: for message with negative acknowledge.err for a message with transmission error. The vocablury can be expressed as a set, v = {ack, err, nak}[7].

6. MESSAGE FORMAT

Each message consists of a control field identifying the message type and a data field with character code. The general form of each message can be represented symbolically as a simple structure of two fields. {Control tag, data}.bit oriented, character oriented, byte counter oriented these three main formatting methods are used.

A bit oriented protocol transmits data as a stream of bits. Boundaries of the data frame can be recognizing using unique bit pattern or flags. In character oriented method some minimal structure is enforced on the bit stream. If number of bits per character is fixed to n bits, all communication take place in multiple of n in byte count oriented method the flags of bit oriented and control characters of a character oriented protocol are used to structure a raw data stream into larger fragments. This method is most widely used now a day [7].

7. PROCEDURE RULES

Procedure rules can be specified informally or formally. An important aspect of the protocol design problem is that the procedure rules are interpreted concurrently by a number of interacting processes. New rules can be added according to variation in environment. To formalize these rules we can use state transition diagram, flow charts, algebraic expression or program form descriptions. Procedure rules may be incomplete or contradictory due to changing demand by advanced technology. We proposed to use fuzzy logic to design procedure rules [7].

8. PROTOCOL SPECIFICATION AND DESCRIPTION LANGUAGES

Protocol specification describes operation of each entity within a layer in response to commands from users or messages from other entities. Protocol specification can be described using asynchronous and synchronous programming languages [6].

Asynchronous programming languages like CSP, Occam, and Ada described process as loosely coupled, independent execution unit. These languages support non deterministic behaviors i.e. more than one actions offered for simultaneous execution.

synchronous concurrent programming languages like estelle,lotos, esterel, sdl are used to model processes that react immediately to their input[6]. brief introduction of languages is given below.

ESTELLE (Extended State Transition Language): This language was developed to model Open System Interconnection services and protocol.

LOTOS (Language of Temporal Ordering Specification): It was developed to define implementation independent formal standards of OSI service and Protocols.

SDL (Specification and Description Language): It is a formal description technique that has been standardized by telecommunication industry. It has been designed to specify and described functional behaviour of telecommunication system.

ESTEREL: It is a synchronous reactive language have central clock for the system. It supports non deterministic and non- reactive programming well. Esterel compiler translate program written in higher level languages like C++ [6].

9. IMPORTANCE OF SPECIFICATION DESIGN STEP

This is a most important stage in protocol design and development. Once a protocol is developed it is tested according to the designed specification in the early stages. Specifications are human sensitive and depend on his experience. Hence chances to designed blurred or ambiguous specifications. If specifications are not designed properly then it will lead to wrong protocol development. Improper design of protocol affect performance of communication system following figure shows a diagrammatic representation that if specifications are not designed clearly it will to lead to incorrect protocol design or some uncorrectable errors may there in the developed protocol.

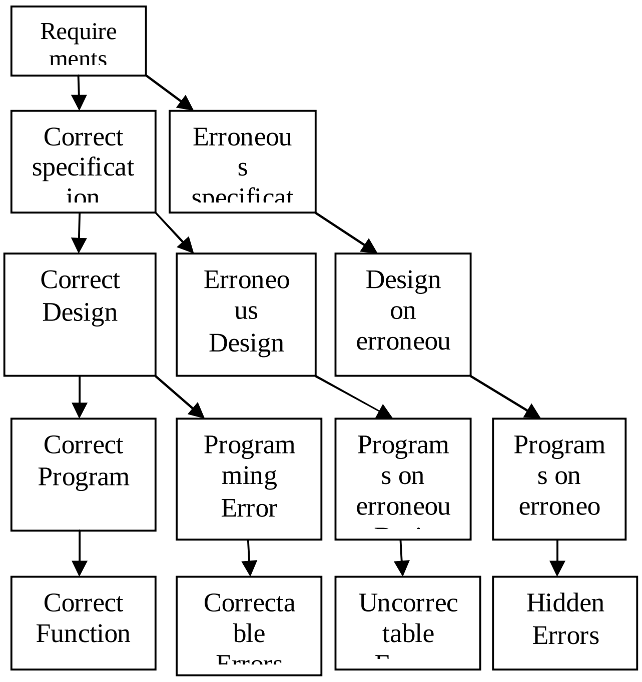

Fig .2 Early design stages

10. CONCLUSIONS

Specification design phase allows the designer to prepare an abstract model of the protocol for testing and analysis. Specification design phase decides service and protocol specification. This is a basis for protocol design and development. Protocol is design and implemented according to specifications design in early stage. A well-designed protocol should have simple and easily understood service specifications. Service and protocol specification are based on user's perception. It is based on user or human experience and highly prone to human errors. Such languages can be easily understood using fuzzy logic. Hence we propose specification design stage should be use fuzzy logic for better result.

REFERENCES

[1] Andrew S. Tanenbaum ,Computer Networks, Fourth Edition: PHI 2008.

[2] Behrouz A Forouzan,Data Communications and Networking , FourthEdition :Tata McGraw- Hill 2006

[3] Behrouz A Forouzan, TCP/IP Protocol Suite, Fourth Edition: TataMcGraw- Hill.2000.

[4] Chris A. Vissers, Luigi Logrippo ," THE IMPORTANCE OF THE SERVICE CONCEPT IN THE DESIGN OF DATA COMMUNICATIONS PROTOCOLS" Fifth International Workshop on Protocol Specification, Testing and Verification Elsevier-North-Holland, 1986, 3-17.

[5] CHUNG-MING HUANG, JENQ-MUH HSU, "An incremental Protocol Verification Method", The Computer Journal , Vol 37, No 8, 1994.

[6] Edward A. Taft, Robert M. Metcalfe, "Formal Methods in Communication Protocol Design", IEEE Transactions on Communication, Vol.Com-28, No. 4, April 1980.

[7] Gerard J. Holzmann, 1991. Design And Validation Of Computer Protocols

[8] GERARD J.HOLZMANN, "An Improved Protocol Analysis Technique", Software Practice and Experience ,Vol 18, No. 2 PP. 137-161, 1988.

[9] Ge Ran, Huazhong Zhang, Shulan Gong ,"Improving on LEACH Protocol of Wireless Sensor Networks Using Fuzzy Logic", Journal on Information & Computational Science, volume 7 issue 3,2011,pg 767-775.

[10] Gregor V. Bochmann, " Finite State Description of Communication Protocols", North-Holland Publishing Company,Computre Networks 2,pg 361-372,1978.

[11] Gregor V. Bochmann, CARL A. SUNSHINE,"Formal Methods in Communication protocol Design", IEEE Transactions on Communication, Vol.Com-28, No. 4, pg 624-631,April 1980.

[12] J.L. da Silva Jr., MSgroi, F. De Bernardinis, S.F.Li, "Wirelsee protocols Design and Opportunities"

[13] K.Thramboulidis,A. Mikroyannidis,"Using UML for the Design of Communication protocols: the TCP Case Study.

[14] K. J. Turner, M. van Sinderen," LOTOS Specification Style for OSI" Proc. 3rd LotoSphere Workshop, pages 5/1-22, Pisa, September 1992. CNUCE.

[15] M.Sgroi,J.L.dasilva, Jr,F.DeBernardinis, F. Burghardt, " Designing wireless Protocols: Methodology and Applications. http://embedded.eecs.berkeley.edu/Respep/Research/asves/paper2000/Marco_icassp00.pdf

[16] Marten van Sinderen,Luis Ferreira Pires,Chris A. Vissrrs," Protocol Design and Implementation Using Formal Methods", This work has been partly supported the CEC research programme ESPRIT II(Lotosphere,ref.:2304).

[17] Nikolay Kakanakov, Ivan Stankov, Mitko Shopov, Grisha Spasov, "Controller Network Data Extracting Protocol – Design and Implementation", International Conference on Computer Systems and Technologies - CompSysTech'06

[18] Samiksha nikam ,B,T. Jadhav," Fuzzy Approach for Networking Protocol Design and Development",International Journal of Advance and Innovative Research ,ISSN 2278-7844,Vol 1, Issue 2 July 2012.

[19] Siegfried LOFFLER,Ahmed SERHROUCHNI ,"Protocol Design: From Specification to Implementation", 8march 1999

[20] Thomas Eugene Truman ,"A Methodology for the Design and Implementation of Communication Protocols for Embedded Wireless Systems", Spring 1998