Fuzzy Approach for Formal Protocol Design Methodology

Samiksha Nikam
College of Computer Science for women Satara, Affiliated to S.N.D.T.University , Mumbai, India

B. T. Jadhav
Y.C. Institute of Sciences,Satara ,Affiliated to Shivaji University , Kolhapur ,India

ABSTRACT
A Protocol defines how communication takes place between computer systems. Protocol designing is complex task due to growing size of network and changing technology. Informal and Formal methodology used to design protocol. Informal method uses natural language or textual format to design specification. Formal methodology uses formal language to define specifications. Informal methodology have some shortcoming like blurred or ambiguous specification, analysis of protocol is complex and not suitable to design large protocols. Formal methodology uses automated tool to design specification and very useful for debugging of specification. Specification designing is key step in protocol design and development in both methods. Specification based on user requirement hence it is human sensitive. Fuzzy logic is best to understand natural language. It is multi-output and multi-input solution finding logic. It supports all designing phases of a protocol. Hence we proposed to use of fuzzy logic for specification design in formal protocol design methodology

Keywords
FormalDesign,Protocol,Specification,Validation.

1. INTRODUCTION
In computer communication protocol plays a vital role. Without protocol communication between computers is not possible. A protocol is a collection of rules which decides how communication governs between computers. Protocols are also responsible for deciding syntax and semantics of data exchange between computers. It is also responsible for error free, in time and in order delivery of data at destination. Protocol designing is complex task due to changing demands by advances in communication technology.

Basically Informal and Formal methodology 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 [9, 10, 12, 14, 16,]. 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. [11, 13, 16]

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. [10]..

2. PROTOCOL DESIGN
Protocol designing is basically carried out in two phases i.e. implementation independent and implementation dependent as shown in fig 1.1. In the first phase function of protocol is design without concerning implementation details. The resulting design support definition of open system specification and it is called the protocol design or simply the protocol [14].The next step in protocol design is implementation dependent design .During this step a protocol is refined until it works successfully as per design specification. This is called implementation of protocol. Protocol design and implementation must be correct with respect to service [14].

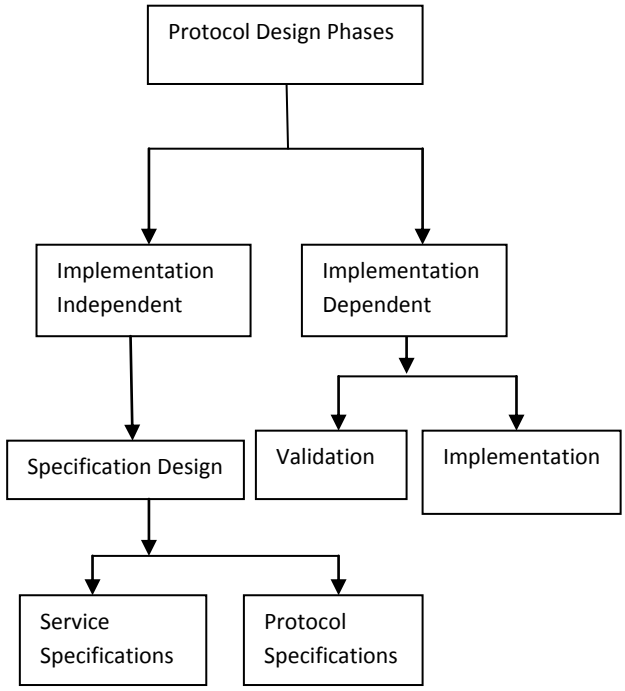

Fig.1: Protocol Design Phases

Formal Protocol design methodology consist of three steps,

- Specification Designing.
- Verification or validation
- Implementation.

2.1 Specification Design

This is an important stage in protocol designing. Specification decides input output behavior of protocol. Specifications are designed according to user requirement. It must be easy to understand and expressive enough to avoid ambiguity [10]. A protocol specification consists of five distinct parts [6],

- The service to be provides by the protocol.
- The assumptions about the environment in which the protocol is executed.
- The vocabulary of messages used to implement the protocol.
- The encoding of each message in the vocabulary.
- The procedure rules guarding the consistency of message exchanges.

The fifth element of specification is the most difficult to design and also hard to verify [6].

Specification designing is categories as service specifications and protocol specifications. Service and protocol are distinct concept. A service is a set of operation that a layer provides to the layer above it shown in fig 1.2. Service relate with interface between two layers. 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 can not be changed. Service defines operations that can be performed on object but does not provide implementation details. Protocols are invisible to the user [1].

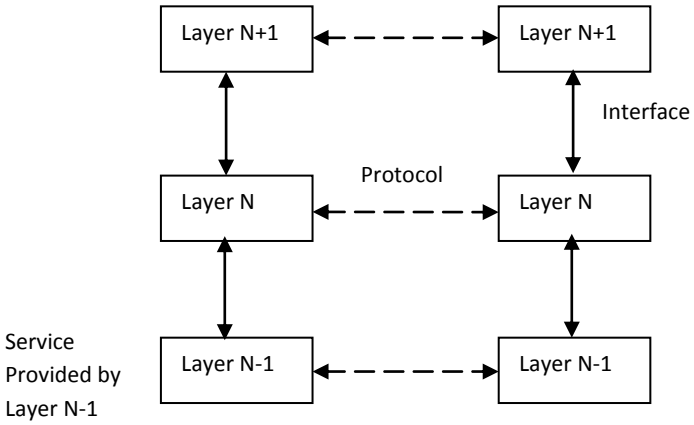

Fig.1.1: Protocol Layers

2.1.1 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].

2.1.2 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. [10]. 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 [10]. 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 [9].

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 [15]. Hence we propose protocol design should be use fuzzy logic for better result.

2.2 Protocol Design Principles

Protocol designing is a complex task. Below we mentioned basic principles used to design a protocol.

- Simplicity: protocol design should be simple and easy to understand. A well-defined protocol can be built from a small number of well designed and well understood pieces. Each piece performs one function and performs it well. Protocols designed such a way are easy to understand as well as easy to implement [6].
- Modularity: A protocol that performs a complex function can be built from smaller pieces that interact in well defined and simple way. Each small piece of software is separately developed, verified, implemented and maintained. Orthogonal functions are not mixed they are designed as separate entity [6].
- Well Formed Protocols: protocol should not be over-specified and incomplete. A well formed protocol should be bounded that it should know system limits. A well formed protocol should be self-stabilizing and self-adapting [6].
- Robustness: protocol is said to be robust when it is able to work under in unexpected environment [6].
- Consistency: Deadlocks, livelocks and improper termination are three situations in which protocol get fail. Such situations should be detected while designing a protocol [6].

2.2.1 Rules for Protocol design

Rules of protocol design are mentioned below. We propose fuzzy logic at some stages of design.

- Make sure that problem is well defined. Requirement for this is that Specifications should be easy to understand. There should not be ambiguity in specification design [6]. To remove ambiguity in specification fuzzy logic is best solution. We propose use of fuzzy logic for protocol specification design.

- Define the service to be performed at every level of abstraction before deciding which structure should be used to realize these services [6].
- Design External Functionality before internal functionality [6].
- Keep it simple. Fancy protocols are complex and bulky than simple ones. They are hard to implement and verify [6]. Here we proposed use of fuzzy logic to reduce complexity of protocol.
- Do not connect what is independent. Separate orthogonal concerns [6].
- Do not introduce what is immaterial. Do not restrict what is irrelevant. Design should be open ended [6].
- Build a prototype to verify protocol before implementing it [6].
- Implement the design, measure its performance and if necessary optimize it [6]. Optimization can be done by using fuzzy logic. We propose fuzzy logic to optimize protocol design.
- Check the final optimized implementation is as per the specification designed [6].

2.3 Protocol Validation

Once protocol is designed next step is validation of protocol. Protocol Validation means verifying that the protocol works accordingly designed specifications. It detects logical errors such as deadlocks, unspecified reception, state ambiguity, live lock or dynamic lock, non executable transactions, Channel overflow [4, 7]. Protocol verification is depends upon specification designed. It is also depends on properties of lower layer protocol. In verifying that a protocol meets its service specification, it will be necessary to assume the properties of lower layer service [3].

Most of the formal techniques are used to validate a protocol. Global state reachability analysis is most straight forward and automated validation technique. Using this technique global state reachability analysis graph is generated which contains all possible transition sequence an all reachable goal state in communication protocol [4, 7, 9]. Problem of State space explosions observed in reachability analysis method. This problem arises due to changing technology, growing network size and complexity of protocol design. To design complex protocol very large number of states are needed which involves counters and more complex data [4, 7, 9, 10].To overcome this problem various techniques used such as Partial specification and Verification, Choosing large units of Actions, composition into Sub Layers, Classification States by Assertion, Focusing Search, Automation etc., [10].

Most of the formal methods are proposed incremental verification [16], state vector model [17] to improve validation process. A validation of protocol is also done by model checking technique. This technique used automated tool to validate a system. SPIN is a tool used to simulate and validate Protocols. [16,17]. PROMELA, its source language, is a formal description technique like SDL and Estelle that is based on communicating machines. SPIN offers modern algorithm to validate protocol [16].

2.4 Implementation

Once a protocol is verified against specification mentioned then next step is its implementation. Some methods used for protocol design can be used for protocol implementation. In Lotosphere project two approaches used for implementation i.e. use of pre defined implementation constructs and Compilation. First approach assumes that the availability of a set of high level general purpose implementation component called pre defined implementation constructs (PDIC). Each PDIS is represented by its formal specification in LOTOS. The LOTOS (Language of Temporally Ordered Specification) language was developed during the standardization of OSI model. This language is used for design and implementation of protocol [14].

3. LIMITATION OF FORMAL METHODS

Following are some limitations of Formal Protocol design Methodology.

- State Explosion problem: due to growing size of network, changing technology protocol designing turn out to be complex. Hence it required to design large number of states which involves huge memory, complex data structure and. Also analysis of computer system required more time [4, 7, 9].
- Reusability is absent in the design phase and very limited in the implementation phase. This disadvantage can be removed by using object oriented paradigm [12].
- Human intervention is needed in many cases for distinguishing between useful and undesirable specification [10].

4. CONCLUSION

Protocol is design by using informal or formal method. Informal method use natural language for specification. But it leads to ambiguous and lengthy specification which results in incorrect protocol design. On the contrary formal protocol design method uses formal specification which avoids ambiguity and specifications are more expressive and understandable. Formal methodology have problem of state explosion and Reusability of code is absent in design and implementation phase.

Various automated tools are used for design and development of protocol. However Specification design is vital stage in protocol design because if specifications are well form then only protocol will verify successfully. Human factor is prime contributor responsible for network performance. Human intervention is needed for specification design and for most of the designing steps. In such a case fuzzy logic is best suitable logic because it is multi-input multi-output solution finding logic. It can understand natural language easily. So we prose use of fuzzy logic in protocol designing.

5. REFERENCES

[1] Andrew S. Tanenbaum, 2008. Computer Network.

[2] BehrouzA.Forouzan, 2002. TCP/IP Protocol Suite.

[3] Behrouz A ForouZan, 2006. Data Communication and Networking.

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

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

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

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

[8] 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.

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

[10] 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.

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

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

[13] 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

[14] 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 programmeESPRIT II(Lotosphere,ref.:2304).

[15] Samikshanikam ,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.

[16] Siegfried LOFFLER,Ahmed SERHROUCHNI ,"Protocol Design: From Specification to Implementation", 8march 1996.

[17] YoungJoonByun ,"A Tool for Design and Validation of Communication Protocol using State Transition Diagrams and Patterns",Journal of Software,volume 2,no 311,2011,pg56-63.