ON USING PROTEAN TO VERIFY ISO FTAM PROTOCOL

R. Lai(*), K.R. Parker(@), T.S. Dillon(*)

(*)Department of Computer Science and Computer Engineering
La Trobe University
Melbourne, Victoria, Australia

(@)Telecom Research Laboratories
Telecom Australia
Melbourne, Victoria, Australia

Abstract

This paper describes the use of PROTEAN and associated methodology to verify the ISO FTAM (File Transfer, Access and Management) DIS (Draft international standard) protocol and discusses the analysis of its behaviour using PROTEAN facilities. PROTEAN is an automated validation tool developed by Telecom Australia. The formal description technique used is Numerical Petri Nets (NPNs), an extension of Petri Nets. The procedures carried out were based primarily on reachability analysis. The behaviour of the protocol as specified was compared to its service specification. There are two protocol machines specified for FTAM : the basic protocol and the error recovery protocol machines.

1 Introduction

Protocol verification is the demonstration of the correctness, completeness and consistency of the protocol design represented by its formal specification. Techniques for specification and verification of computer network protocols have progressed significantly in the past decade. The success is largely due to the development of the state transition approach for formal specification and to the greater automation of the verification process. Automated protocol verification is the use of computer tools to verify a communication protocol based on its formal specification [2,10].

PROTEAN (PROTocol Emulation and ANalysis) [3], developed by Telecom Australia, is a computer aided tool for analysis of computer communication protocols. It finds faults such as deadlocks, livelocks and maloperations specific to the protocol under test. It is based on a formal description technique called Numerical Petri Nets (NPNs) [11]. An NPN specification is the starting point for verification using PROTEAN. In practice, some faults can be uncovered during the process of creating a precise NPN specification.

ISO 8571 [4] defines a file transfer service and protocol, known as File Transfer, Access and Management (FTAM) [7]. It controls the transfer of whole files or parts of files between end-systems. The protocol is available within the application layer of the OSI (Open Systems Interconnection) reference model [1]. Primitive is an OSI term essentially meaning commands or messages passed between the OSI entities such as the user and the protocol machine.

FTAM supports two services: the Reliable File Service and the User Correctable File Service. For the Reliable File Service, the user states its quality of service requirements, but has no control of error recovery, delegating such considerations to the service provider. For the User Correctable File service, the user has primitives available for error recovery and transfer management. There are 2 protocols specified for FTAM : basic file protocol, supporting the user correctable file service, and error recovery protocol, supporting reliable file service.

This paper describes the use of PROTEAN to verify FTAM DIS protocols specified in NPNs.

2 Numerical Petri Nets

Numerical Petri Nets (NPNs) is a formal description technique that belongs to the family of state transition models. It is an extension of Petri Nets [8]. It was originally developed by Symons [9]. It can be specified algebraically or graphically. The graphical NPN consists of a bipartite directed graph, fixed firing rules and an initial marking. An NPN may have global variables (called P-variables) which transitions can read and write. An NPN has transition enabling conditions and operations which refer not only to the tokens in the input places but may also refer to the global variables. By a marking of a Petri Net we mean a complete specification of the tokens in each of its places and the value of all its global variables.

An NPN, labelled G, can be defined by a quintuple(P,T,Fi,Fo,Mo), where

"P" is the set of all places in the net G, where a place is represented by a circle in the NPN graph and can be used to represent a machine state;

"T" is the set of all transitions in the net G, where a transition is represented by a bar in the NPN graph and can be used to represent an event; each transition includes it's transition conditions and transition operations;

"Fi" are the input firing rules, usually written on the arcs between the input places and their respective transitions; this specifies what tokens are required to be in the input places to enable the transition and what tokens are destroyed in (i.e. removed from) the input places when the transition is fired;

"Fo" are the output firing rules, usually written on the arcs between the transitions and their respective output places; this specifies what tokens are created in (i.e. added to) the output places;

"Mo" is the initial marking of the net, this specifies all the tokens in each place and global variables.

3 Reachability Analysis

A system can be analysed by considering every possible sequence of events and thus every possible "state" of the system. The set of all possible states of the system is known as the Reachability Set, which is unique for a given system NPN with a given initial marking.

In Petri Nets, the relationship between states is given by the Reachability Graph (RG). It may be presented in tabular or graphical form. Reachability analysis is the study of the reachability graph generated for the system being verified.

The RG is used to investigate the properties of the protocol being studied. The main problem with the RG is that it gets very big and unmanageable as the complexity of the system modelled increases, and thus analysis becomes very tedious.

4 PROTEAN

PROTEAN is a user friendly menu-driven system, with on-line help and simple error messages. It has two parts. The first is the NPN Analyzer Program, which handles the NPNs and the generation of the reachability graph. The second is a collection of programs which helps the user detect maloperations and other properties of the protocol using the results from the first stage.

The NPN analyzer program allows NPN subnets to be entered via keyboard or a file. The NPNs are checked for correct syntax. The subnet NPNs can be recalled and combined with other subnets to form larger NPNs. The NPNs can also be modified, listed and displayed graphically.

The NPN is initialised by placing tokens in the places and assigning values to the data variables. A user can then investigate the operation of the net manually or automatically. The manual method allows execution of the net by firing one transition at a time. In the automatic mode the complete reachability set and RG are generated. All deadlocked markings are identified.

PROTEAN contains several programs to investigate properties of the RG. They are RG graphics display, loop detection, livelock detection, reduction of RG and scenario generation. Some illustrations of the use of these are given in section 9.

5 FTAM Model

The operation of the FTAM protocol is modelled by the interaction of two file protocol machines (FPMs). The two FPMs communicate by means of the services available at their lower boundary, in such a way as to provide the service required at their upper boundary.

The file service is defined asymmetrically, with the file service user "A" being the initiator and file service user "B" being the responder. File user "A" can request the user correctable file service or the reliable file service.

The file service and its supporting protocol are concerned with creating a series of stages, a working environment in which the initiator's desired activities can take place. This leads to a set of contexts being established. The period for which some parts of the common state held by the service users is valid is called a "regime". As progressively more shared states are established a nest of corresponding regimes is built up. However, there will, in general, be a time lag between the establishment of a regime at the two ends of the association.

Four types of file regimes are defined: FTAM establishment, file selection, file open, and data transfer regime.

6 Verification Methodology

The procedure used to analyse FTAM follows a methodology which has three main stages.

The FTAM protocol is divided into the basic protocol and the error recovery protocol. For the basic protocol, it is further divided into the basic file protocol, the bulk data transfer protocol, and the basic file protocol under grouping control.

The next phase involves partitioning the protocol into subsystems and creating a complete specification of each of these. This is done by modelling with NPNs to show the systems state changes, data processing and signal flow. The NPNs for each subsystem are joined to form the model of the total system.

Finally the logical operation of the protocol model is then analysed. The possible sequences of events are considered, and the possible states of the system. In particular these are checked to see if the protocol conforms to the requirements of its service specification.

7 Formal Specification of FTAM

The nets are viewed as modelling the operation of the protocol machines. In our approach each service or function provided by the machine is specified via a separate net. We use places to represent ports through which data are sent and received, and global variables to represent states. Predicates are used to govern the conditions of the transition firings.

In this situation, the specification of a single function in a single net has logical advantages. This approach represents the flow of communication primitives closely and makes it easier to follow the behaviour of the entities involved. The data parameters passed by the protocol are modelled by the "attributes" of the tokens.

The specification is based on the Draft International Standard of ISO 8571. The NPN specifications of the whole FTAM protocols are based on the state tables in the Annex of ISO 8571, since the standard states that the Annex is to take precedence over the text.

The detailed formal specifications of the basic file and error recovery protocols are described in [5].

8 FTAM Verification

The NPNs must be first correctly entered into PROTEAN. It is entered in a net form, with each net modelling a service or function of the protocol machines. This allows the behaviour of various sub-nets to be analysed.

There are two methods available in the PROTEAN system to analyse a protocol : single step method and automatic generation. It is not feasible to use single step method to analyse FTAM as there are too many enabled transitions as more nets are added. Using the second method, a RG is automatically generated. The RG can be automatically checked for the presence of any deadlocks. The problem with this method is that the RG gets very large. It then becomes very tedious to analyse the system.

There are many primitives and regimes defined in the protocol. To analyse the protocol by combining all the nets will make the task insurmountable because the RG which would be thus generated will be too huge. For the basic protocol, the analysis was done in 3 phases: basic file protocol, bulk data transfer protocol and basic file protocol under grouping control. The RG was generated for each of the three phases. However, for the bulk data transfer phase, two separate RGs were created with the Cancel Data request included only in one, and the Restart Data request included only in the other.

The issuing of primitives follows the nesting of the regimes. The protocol was examined for behaviour upon receipt of combinations of primitives of interest. The protocol was also checked for situations with error conditions, for example, unsuccessful connection establishment, unsuccessful creation of files, etc. In the NPN model, these are specified by predicates. The net was simulated under all these conditions.

The properties of the protocol were investigated using the Liveness, Language, Scenario and Cycle facilities of PROTEAN. The results of verifying the FTAM basic file and error recovery protocols are described in [5,6].

9 Analysis Using PROTEAN

9.1 Liveness Analysis

The Liveness program determines all of the strongly connected components of the RG. A strongly connected component consists of one or more nodes for each of which a path can be found to every other node in that component. A liveness graph is generated showing how the strongly connected components of the RG are related. Livelocks occur as leaf nodes of this graph with two or more markings (leaf nodes with one marking are deadlocks). If there is only one strongly connected component, the protocol has no livelocks or deadlocks.

Liveness analysis was applied to the RGs generated for the nets created in the specifications of the FTAM protocols. This revealed several leaf nodes which were found to be deadlocks as described in [5,6]. There are no actual livelocks uncovered in the analysis.

9.2 Language Analysis

The reachability graph shows all of the possible sequences of all transitions, ie. the language of all of the transitions. To determine if a protocol meets its service, or to verify other properties, it is useful to determine the language of selected key transitions. The Language program reduces the reachability or language graph to show only the sequences of these user-determined key transitions.

Language analysis was applied to the transitions representing FTAM services to study the behaviour of the FTAM file regimes. It verifies that the protocol does behave according to the FTAM regimes.

For example, a language graph for a net that consists of F-initialize, F-select, F-create, F-deselect, F-delete, and F-terminate is shown in figure 1, where the circles are markings labelled by a number. The transitions which fire to go from numbered one marking to another are as follows:

1	Finirq (F-initialize request)
	2
2	Finicf (F-initialize confirm)
	13
13	Fselrq (F-select request)
	15
13	Fcrerq (F-create request)
	14
14	Fcrecf (F-create confirm)
	36
15	Fselcf (F-select confirm)
	36
36	Fdesrq (F-deselect request)
	38
36	Fdelrq (F-delete request)
	37
37	Fdelcf (F-delete confirm)

```
38        Fdescf (F-deselect confirm)
        59
59        Fterrq (F-terminate request)
        60
60        Ftercf (F-terminate confirm)
        1
```

The initial marking is given on the left of the transition name, and the final marking number is below it. The language graph with the legend above shows possible sequences of behaviour. Initially there is no connection, then the FTAM regime is established with the issuing of an F-initialise request and subsequent receipt of an F-initialise confirm. Then the initiator may either issue an F-select request or F-create request, etc.

The language graph is essentially a reduction of the RG with unwanted details suppressed (hence the missing numbers in the the sequence of markings).

9.3 Scenario Generation

The Scenario program finds paths in a RG that match a specified transition or marking sequence. The user may exclude particular nodes or transitions from the sequence and may limit the number of paths found. This facility enabled the tracing of specific behaviour in the FTAM protocol and, in particular, was useful for debugging and tracing the cause of deadlocks.

For example, the deadlock uncovered for the File Open service has the path listed in below.

```
Path 1: 1 -> (Finirq) -> 2 -> (Ainirq) -> 3 -> (Aassrq) -> 4
        -> (Aassin) -> 5 -> (Binirq) -> 6 -> (Finiin) -> 7
        -> (Finirp) -> 8 -> (Binirp) -> 9 -> (Aassrp) -> 10
        -> (Aasscf) -> 11 -> (Ainirp) -> 12 -> (Finicf) -> 13
        -> (Fselrq) -> 14 -> (Aselrq) -> 15 -> (PdatrqAsel) -> 16
        -> (PdatinBsel) -> 17 -> (Bselrq) -> 18 -> (Fselin) -> 19
        -> (Fselrp) -> 20 -> (Bselrp) -> 21 -> (PdatrqBsel) -> 22
        -> (PdatinAsel) -> 23 -> (Aselrp) -> 24 -> (Fselcf) -> 25
        -> (Fopnrq) -> 26 -> (Aopnrq) -> 27 -> (PdatrqAopn) -> 28
        -> (PdatinBopn) -> 29 -> (Bopnrq) -> 30 -> (Fopnin) -> 31
        -> (Fopnrp) -> 32 -> (Bopnrp) -> 33 -> (PdatrqBopn) -> 34
        -> (PdatinAopn) -> 36 -> (PaltrqBopn) -> 37 -> (Aopnrp) -> 39
        -> (p4Fopncf) -> 41
```

Marking 41, the final marking in that path, is a deadlock. From this sequence generated by Scenario, the reason for the deadlock can be traced. In this instance the deadlock was due to an error in the FTAM state table, where a negation is missing from action 14.

9.4 Elementary Cycle

The Cycle program lists all of the loops contained in the reachability or language graph. It identifies the largest cycle and may be used for a loop layout of the graph. Cycle is useful in investigating the cyclic behaviour of the protocol.

For each RG, there is at least one cycle. For the example given in section 9.2, there are 4 cycles. (Cycle uses the language graph.) The cycles reveal different paths that can be taken for the FTAM file regimes. They can be obviously seen from the language graph in figure 1, for this simple example, and are as follows:

Cycle 1: 1 2 13 14 36 37 59 60
Cycle 2: 1 2 13 14 36 38 59 60
Cycle 3: 1 2 13 15 36 37 59 60
Cycle 4: 1 2 13 15 36 38 59 60

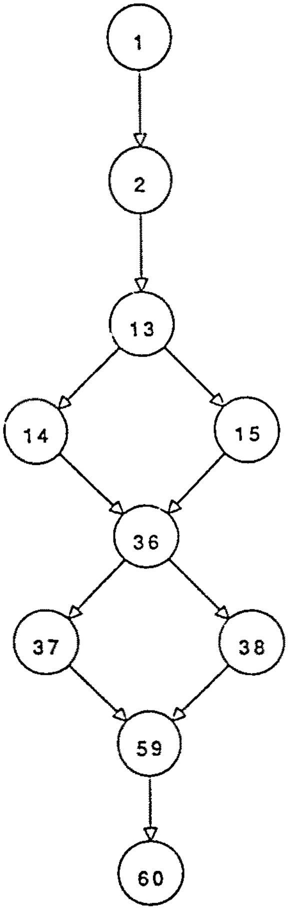

Figure 1. Language Graph

9.5 Limitations

There are several ways in which facilities currently available in PROTEAN limit the usefulness of it for the analysis of a complex protocol, like FTAM. These include :

- Lack of direct graphical input of NPNs
- Lack of auxiliary net input facilities such as macros
- Lack of refinement facilities for nets

e.g. representation of a subnet at one level as a single transition at a higher level

Other needs, as presented in [3], include handling of infinite reachability set, incorporation of a query language, language comparison, background simulation and improved performance for analysis and graphical layout. Alternative analysis techniques, such as improved reachability analysis, invariant analysis and net reduction, also have great potential for aiding practical analysis. This will greatly improve the range of analysis that will be possible for FTAM, and for other complex protocols.

10 Conclusions

The techniques used to perform a verification of the ISO FTAM DIS protocol using PROTEAN, after subdivision into manageable tasks, have been presented. We have reported results of the analysis of protocol properties performed using PROTEAN, whose utilities proved to be useful in this task. In particular the discovery of deadlocks in the NPN model of FTAM were found, as was an absence of livelocks and conformance to the FTAM service regimes.

While reachability graph generation is fundamentally an exhaustive searching process, we have indicated that with the aid of the PROTEAN automated analysis tool, reachability analysis can be of practical use, even for complex protocols. We thus conclude that reachability-based analysis of Petri-net specifications is a useful approach in the practical verification of complex protocols, and that there is great potential for further tool development.

11 Acknowledgment

We would like to thank Telecom Australia Research Laboratories for the use of their PROTEAN software and to J. Billington for his comments on an early draft of this paper.

References

[1] Bartoli, P.D., "The Application Layer of the Reference Model of Open Systems Interconnection", Proceedings of the IEEE, vol. 71, no. 12, pp. 1404-1407, December, 1983.

[2] Billington, J., Wilbur-Ham, M.C., and Bearman, M.Y., "Automated Protocol Verification", Proceedings of the Fifth International Workshop on Protocol Specification, Testing and Verification, June, 1985.

[3] Billington, J., Wheeler, G.R., and Wilbur-Ham, M.C., "PROTEAN: A High-level Petri Net Tool for the Specification and Verification of Communication Protocols", IEEE Transactions on Software Engineering, Vol. 14, No. 3, pp. 301-316, March 1988.

[4] ISO DIS 8571, "Information Processing Systems - Open Systems Interconnection - File Transfer Access and Management - Parts 1,2,3 and 4", Draft International Standard, ANSI, New York, 1986.

[5] Lai, R., "Formal Specification and Verification of ISO FTAM Protocol", PhD Thesis, La Trobe University, Australia, August, 1989.

[6] Lai, R., Dillon, T.S., Parker, K.R., "Verification Results for ISO FTAM Basic Protocol", Proceedings of Ninth Symposium on Protocol Specification, Testing and Verification, North-Holland, June, 1989.

[7] Lewan, D., and Long, H.G., "The OSI File Service", Proceedings of IEEE, Vol. 71, no. 12, pp. 1414-1419, December, 1983.

[8] Peterson, J.L., "Petri Nets Theory and the Modelling of Systems", Prentice-Hall, Englewood Cliffs, N.J., 1981.

[9] Symons, F.J.W., "Modelling and Analysis of Communication Protocols using Numerical Petri Nets" PhD Thesis, Department of Electrical Engineering Science and Telecommunications, University of Essex, May, 1978.

[10] West, C.H., "An Automated Technique of Communication Protocol Validation", IEEE Transaction on Communications, vol. COM-26, pp.1271-1275, Aug, 1978.

[11] Wheeler, G.R., "Numerical Petri Nets - A Definition", Telecom Australia Research Laboratories Report 7780, Telecom Australia Research Laboratories, May, 1985.