Protocol Correction : Methods and Tools

Richard Castanet, Bernard Cousin, Omar Rafiq, César Viho

Computer Networks '91

Key words: Protocol, Finite State Machine, Reachability Graph, Correction, Suppression.

Richard CASTANET*
Bernard COUSIN*
Omar RAFIQ†
Gagnon VIHO*

PROTOCOL CORRECTION: METHODS AND TOOLS.

From their informal specification to their implementation, the different steps of protocols development still represent a hard work for designers. In this paper, we develop a method to help protocols designers in specification and verification steps, such that the considered protocols are error-free and correspond to the expected service. It allows the suppression of errors such as unspecified receptions, deadlocks and blocking loops.
After a correction oriented analysis of the reachability graph, our method proposes groups of transitions (of one or another finite state machines representing the communicating systems) which resolve the non-regression problem: the suppression of transitions in any of these groups eliminates the considered error without generating other errors.

1 Introduction

The goal of formal specification and verification steps is respectively to describe unambiguously protocols and to ensure their correct functionning.

Today, with fomal description languages such as SDL [2] and ESTELLE [4], specification allows complete description of all mechanisms which can occur in a protocol: segmentation, error recovering, data concatenation, flow control, multiplexing, synchronization etc...

From the established results in protocol verification [8, 9], tools allowing protocols errors detection have been developped during the last years. The contribution brought by such tools is certainly

*LaBRI-ENSERB – 351, Cours de la libération – 33400 Talence, FRANCE
†Laboratoire TASC – Université de Pau – Av. de l'Université – 6400 Pau, FRANCE

important but still much more has to be done by designers to correct the detected errors. This difficulty leads certain designers to pass over the verification step which is yet required for error-free protocol development. Others use techniques based on trial and error methods. Protocols thus implemented may work partially or be entirely unusable. The costs of implementation of any system and particularly of a protocol is such that "*it is better to correct the specification rather than implementation*".

Our aim is to develop an approch to cope with this problem.

2 Specification and error detection

Protocols are described by communicating systems exchanging messages throughout FIFO channels. Each communicating system is represented by a Finite State Machine (FSM).

The verification step generates the reachability graph [10]. In this graph, nodes are called **global states** and each one describes a situation of the communication between the communicating systems. An edge represents an action in one of the systems that generates the global state it leads to.

Gradualy as those global states are generated, errors such as unspecified reception, deadlock or blocking loop, are detected.

3 Problems to be resolved

Errors are located on global states in the reachability graph. To eliminate errors, we have then to suppress those global states. But, this graph is built upon communicating systems. So, the suppression of global states cannot be made without consideration of those systems.

In fact, suppressing a node in a graph consists in the suppression of all edges which lead to this node and all nodes issued only from it. In a reachability graph, the label of an edge is an edge of one of the communicating systems and, the label of an edge which leads to a global state can be the label of other edges in the reachability graph.

Consequently, the suppression of an edge in the reachability graph causes the suppression of all edges which have the same label. This means that, the suppression of global state is much more complex than a suppression of a state in a simple graph.

Moreover, by suppressing edges in a reachability graph, we can suppress reachability to global states other than those concerned by this suppression. According to the definition of the reachablity graph, suppressing edges to eliminate an error can generate other errors. This problem, called the **non-regression problem**, obliges one to make a judicious choice of edges to be suppressed.

4 Solutions we propose

A solution to the non-regression problem requires a study of the cause-and-effect relationship between the suppression of edges or nodes in a process and the suppression of edges or global states in a reachability graph (see [1] for details and results obtained from this study).

In what follows below, we give main steps that permit us to obtain solutions to the problem of error's suppression without generating other errors.

In order to make the different steps developped below able to understand, an edge in FSM is called **transition.**

Let E be a set of global states representing the detected error to be suppressed.

Step 1: Frontier Cut Plane

The analysis of the reachability graph obtained after the verification step gives a first set of transitions called the **Frontier Cut Plane** of E and denoted by $FCP(E)$.

The suppression of $FCP(E)$ eliminates all reachability to every global states in E.

But, the suppression of these transitions indues the suppression of all edges which are labeled by these transitions. Thus, we can eliminate other global states than those contained in E.

Step 2: Reduced Cut Plane

Reductions of the Frontier Cut Plane generate sub-sets called **Reduced Cut Plane** of E and denoted by $RCP(E)$.

Every $RCP(E)$ satisfies the two conditions: Its suppression eliminates all reachability to E and minimize the number of global states (other than those in E) suppressed.

This suppression can generate other errors.

Step 3: Total Cut Plane

The resolution of the non-regression problem consists in adding to every Reduced Cut Plane, transitions intended to eliminate the errors generated by the suppression of the transitions in a Reduced Cut Plane.

The reachability graph obtained after the suppression of $RCP(E)$ contains less global states and less edges. In this smaller reachability graph obtained after the suppression of $RCP(E)$, we apply step 1 and step 2 to suppress errors generated by this suppression.

We can iterate this, until there is no more error introduced or there is no more transition to be suppressed (this is the worst situation we can obtain).

The sets obtained in this manner are called **Total Cut Plane** of E.

We prove that the suppression of a Total Cut Plane either suppresses the considered error without generating other errors or suppresses all the reachability graph.

Finally, protocols are designed to provide an expected service. The problem of conserving provided service is not our main concern, but we propose two criteria for error suppression: the first allows the designer to choose the Total Cut Planes that contain minimum number of transitions and the second, those which suppresse a minimum of global states.

5 Termination and availability

We present here main arguments proving the termination and availibility of this method.

5.1 Termination

Communicating systems are Finite State Machines and we have supposed all the reachability graph is constructed. At all steps of our method, we propose the suppression of transition which eliminates global states and edges in the reachability graph.

Suppression method and finite domain assure that our method always terminates.

5.2 Availability

We have to prove that our method "*doesn't always propose the suppression of all transitions*" (the suppression of all the protocol).

When all global states are or lead to error, our method proposes the suppression of all the protocol. (It is difficult to correct a protocol where everything goes wrong. This also true for our method.)

Otherwise, our method proposes at least one Total Cut Plane such that its suppression preserves global states including the initial global state, that form a cycle in the reachability graph. Note only that, this is due to the notion of frontier introduced in step 1.

5.3 Application

This method has been implemented as a tool [7], and has given interesting results on real protocols such as "Packed Radio Network Management" [6] and a simplified connection/deconnection X.25 protocol.

For example, in the Packed Radio Network Protocol which describes communication between two stations responsible for network control in a packet radio network, the result given by our method can be resumed like this: Only the suppression of the transition of "REQUEST" reception by the second

station in its temporal "WAITING-FOR-ACK" state, is required to suppress a detected deadlock; this is however hard to see among about the sixty thousand global states contained in the whole reachability graph obtained.

6 Conclusion

The originality of this work is in the fact that it attacks a recognized difficult problem [3, 5, 10], which has no complete and formally proved solution. We present here a formal method and its tool allowing a given error to be eliminated without generating other errors. The difficulty comes from the fact that we have to use the reachability graph which is generally very large, according to the communicating systems upon which it has been built.

References

[1] R. Castanet, O. Rafiq et G. Viho, "De l'aide à la correction des anomalies dans les protocoles de communication", $7^{ième}$ Congrès *De Nouvelles Architectures pour les Communications*, Eyrolles, Paris, Oct. 1990.

[2] CCITT/SGXI/WPWP3-1, "Specification and Description Language", CCITT Recommandations Z100-Z104, May 1983.

[3] P. Guitton, "Description, validation, et test de conformité de protocoles de communication", Thèse No 1941, Université Bordeaux I, Jan. 1984.

[4] ISO-IS-9074, "ESTELLE, a formal description technique based on an extended state transition model", Sep. 1986.

[5] C. Jard et M. Raynal, "De la nécessité de spécifier des propriétés pour la vérification des algorithmes distribués," Rapport de recherche No 590, INRIA-Rennes, Dec. 1986.

[6] C. V. Ramamoorthy, S. T. Dong and Y. Usuda, "An Implemantation of an Automated Protocol Synthesizer (APS) and Its application to the X.21 Protocol," *IEEE Trans. on Soft. Eng.,* vol. SE-11, No. 9, pp. 886-908, Sep. 1985.

[7] G. Viho, "Manuel d'utilisation de VAP," Rapport interne No I-8916, LaBRI - Université Bordeaux I, Avr. 1989.

[8] C. H. West, "General techniques for communications protocol validation," in *Proc. Comput. Network Protocols Symp.,* Liège, Belgium, Feb. 1978.

[9] P. Zafiropulo, "Protocol validation by duologue-matrix analysis," *IEEE Trans. Commun.,* vol. COM-26, pp. 1187-1194, Aug. 1978.

[10] P. Zafiropulo, C. West, H. Rudin, D.D. Cowan and D.Brand, "Towards analyzing and synthesizing protocols", IEEE Tans. Commun., vol. COM-28, pp. 651-661, Apr. 1980.