Chapter 7
Formal Methods for Protocol Verification and Construction

7.1. OVERVIEW OF MODELS AND METHODS

Numerous formal models and verification methods have been proposed and applied to the verification and construction of protocols. Excellent survey articles were written by Bochmann and Sunshine [BOCH 80] and Sunshine [SUNS 79]. ([SUNS 79] is reprinted below.) We mention a few of the models and methods below.

Stenning was one of the first to apply program verification techniques to the study of protocols. He proved some safety properties for a one-way data transfer protocol. His protocol gives a very good illustration of the window mechanism for error and sequence control and was specified using Pascal-like code. He formulated invariant safety assertions and proved them by induction [STEN 76].

Communicating finite state machines provide a different formalism for specifying protocols. With such a formalism, some realistic protocols have been shown to have certain desirable logical properties (e.g. freedom from deadlocks and unspecified receptions). The reader is referred to the articles in [WEST 78, ZAFI 80, RAZO 80]. In each case verification is carried out by a state-space exploration performed automatically by a program.

There were various attempts at protocol verification in a semi-automatic fashion. Brand and Joyner [BRAN 78] employed a program that can perform symbolic execution. DiVito [DIVI 83] employed automatic theorem provers.

To prove liveness properties of protocols, Hailpern and Owicki proposed the use of temporal logic [HAIL 80].

Instead of verifying that a given protocol has certain desirable properties, several researchers investigated methods to construct protocols that are guaranteed to have certain logical properties [ZAFI 80, MERL 83, GOUD 83b, CHOW 83].

7.2. STATE-SPACE REDUCTION METHODS

The concept of a layered architecture helps to delineate a network's functions into different layers and to organize the layers into a hierarchy. It is a step in the right direction toward a systematic approach for constructing software for a high-performance and reliable computer network. It is, however, a relatively small step toward that goal from the following observation. Each protocol layer in a typical layered architecture (ISO, SNA, ARPANET, etc.) when implemented would still be a highly complex set of parallel programs. Here, "complexity" is measured in terms of the current state of the art in the design of parallel programs whose logical behavior can be analyzed rigorously. In simpler terms, a protocol layer is complex because it typically has several functions (tasks) to perform. For example, the software for a basic data link layer in most architectures would have to implement at least three functions: connection management and two one-way data transfers in opposite directions.

A systematic approach does not exist for the construction of correct and efficient protocol systems comparable in complexity to that of a protocol layer. Typically, different abstractions of a protocol system are used for independently analyzing the behavior of different mechanisms in the system, that implement various protocol functions. As a result, logical correctness properties verified for "toy protocols," that are abstractions of a multifunction protocol system, may in fact be *invalid* for the protocol system itself if the abstractions have not been constructed to account for interactions between different mechanisms in the system.

The theory of projections was developed to reduce the analysis of a complex multifunction protocol to that of simpler single-function "image protocols" [LAM 82a, LAM 82b, LAM 82c]. Each image protocol is an abstraction of the given protocol but it is specified just like a real protocol. Each image protocol is of the same complexity as that of toy protocols that have been analyzed in the literature. The construction method guarantees that each image protocol is faithful in the sense that any logical correctness property, a safety or liveness property, that is valid for the image protocol must also be valid for the original protocol. ([LAM 82c] is reprinted below.) An application of the method of projections to verify a version of the HDLC protocol is presented in [SHAN 83].

In general, to reduce the problem of directly constructing a multifunction protocol to that of composing it from several single-function protocols is very difficult. However many real-life protocols can be observed to go through different phases, one at a time, performing a distinct function in each phase. A multiphase model for such protocols was developed in [CHOW 83, CHOW 84]. A phase is formally defined to be a network of communicating finite state machines with certain desirable correctness properties; these include proper termination and freedom from deadlocks and unspecified receptions. A multifunction protocol is constructed by first constructing separate phases to perform its different functions. Chow et al. presented a method to connect these phases together to implement the multifunction protocol such that the resulting network of communicating finite state machines is also a phase (i.e., it possesses the desirable properties defined for phases). We reprint [CHOW 84] below.

The method of closed covers presented by Gouda [GOUD 83a] is another state-space reduction approach. This method can provide finite representations of some protocols whose reachable states are unbounded. This approach was also used to formulate algorithms for detecting livelocks in networks of communicating finite state machines [GOUD 84].

7.3. MODELING OF TIME-DEPENDENT PROTOCOLS

Time-dependent systems are those whose correct functioning depends upon certain time relationships between system event occurrences [SHAN 82]. Most real-life communication protocols are time-dependent systems. Time relationships are widely used in protocol systems to provide concurrency control and to guarantee the ordering of certain remote events [ESWA 81, FRAT 83]. When used properly, they can simplify a protocol and make it more efficient by reducing the amount of handshaking required in reaching an agreement [WATS 81].

A model for representing time-dependent systems is presented in [SHAN 84] together with inference rules for proving safety and progress properties. [SHAN 84], reprinted below, is written specially for this tutorial text.

References

[BOCH 80] Bochmann, G. V. and C. A. Sunshine, "Formal Methods in Communication Protocol Design," *IEEE Trans. on Commun.*, April, 1980.

[BRAN 78] Brand, D. and W. H. Joyner, "Verification of Protocols using Symbolic Execution," *Computer Networks*, September/October, 1978.

[CHOW 83] Chow, C. H., M. G. Gouda and

S. S. Lam, "A Discipline for Constructing Multi-Phase Communication Protocols," Dept. of Computer Sciences, University of Texas at Austin, Tech. Rep. TR-233, June 1983 (revised October 1983).

*[CHOW 84] Chow, C. H., M. G. Gouda and S. S. Lam, "An Exercise in Constructing Multi-phase Communication Protocols," *Proc. ACM SIGCOMM '84*, Montreal, June 1984.

[DIVI 83] DiVito, B. L., "Mechanical Verification of a Data Transport Protocol," *Proc. ACM SIGCOMM '83 Symp.*, University of Texas at Austin, March 1983, pp. 30-37.

[ESWA 81] Eswaran, K. P., G. S. Shedler, and U. C. Hamacher, "Collision-Free Access Control for Computer Communication Bus Networks," *IEEE Trans. on Software Eng.*, November 1981.

[FRAT 83] Fratta, L., "An Improved Access Protocol for Data Communication Bus Networks with Control Wire," *Proc. ACM SIGCOMM '83* Symposium, Univ. of Texas at Austin, March 1983, pp. 219-225.

[GOUD 83a] Gouda, M. G., "Closed Covers: To Verify Progress for Communicating Finite State Machines," TR-191, Dept. of Computer Sciences, Univ. of Texas at Austin, January 1982. Revised January 1983. To appear in *IEEE Trans. on Software* Engineering.

[GOUD 83b] Gouda, M. G. and Y. T. Yu, "Synthesis of Communicating Machines with Guaranteed Progress," TR-179, Dept. of Computer Sciences, Univ. of Texas at Austin, June 1981. Revised January 1983. Revised October 1983. To appear in *IEEE Trans. on Commun.*

[GOUD 84] Gouda, M. G., C. H. Chow and S. S. Lam, "Livelock Detection in Networks of Communicating Finite State Machines," Dept. of Computer Sciences, University of Texas at Austin, Tech. Rep. TR-84-10, March 1984.

[HAIL 80] Hailpern, B. T. and S. S. Owicki, "Verifying Network Protocols using Temporal Logic," Technical Report 192, Computer Systems Laboratories, Stanford Univ., June, 1980.

[LAM 82a] Lam, S. S. and A. U. Shankar, "Verification of Communication Protocols via Protocol Projections," *Proc. INFOCOM '82*, IEEE Computer Society Press, April 1982, Las Vegas.

[LAM 82b] Lam, S. S. and A. U. Shankar, "An Illustration of Protocol Projections," *Proc. 2nd Int. Workshop on Protocol Specification, Testing and Verification*, Idyllwild, Calif., North-Holland Publishing Co., Amsterdam, 1982.

*[LAM 84] Lam, S. S. and A. U. Shankar, "Protocol Verification via Projections," *IEEE Trans. on Software Engineering*, Vol. SE-10, No. 4, July 1984, pp. 325-342.

[MERL 83] Merlin, P. M. and G. V. Bochmann, "On the Construction of Submodule Specifications and Communication Protocols," *ACM TOPLAS*, Vol. 5, No. 1, January 1983, pp. 1-25.

[RAZO 80] Razouk, R. and G. Estrin, "Modeling and Verification of Communication Protocols in SARA: The X.21 Interface," *IEEE Trans. on Comput.*, December 1980.

[SHAN 82] Shankar, A. U. and S. S. Lam, "On

Time-Dependent Communication Protocols and their Projections," *Proc. 2nd Int. Workshop on Protocol Specification, Testing and Verification*, Idyllwild, Calif., North-Holland Publishing Co., Amsterdam, 1982.

[SHAN 83] Shankar, A. U. and S. S. Lam, "An HDLC Protocol Specification and its Verification using Image Protocols," *ACM Trans. on Computer* Systems, November 1983.

*[SHAN 84] Shankar, A. U. and S. S. Lam, "Time-Dependent Communication Protocols," Technical Report TR-84-26, Department of Computer Sciences, the University of Texas at Austin, 1984.

[STEN 76] Stenning, N. V., "A Data Transfer Protocol," *Computer networks*, September, 1976.

*[SUNS 79] Sunshine, C., "Formal Techniques for Protocol Specification and Verification," *Computer*, Vol. 12, No. 9, IEEE, September 1979, pp. 20-27.

[WATS 81] Watson, R. W., "Timer-Based Mechanisms in Reliable Transport Protocol Connection," *Computer Networks*, Vol. 5, No. 1, February 1981, pp. 47-56.

[WEST 78] West, C. H. and P. Zafiropulo, "Automated Validation of a Communications Protocol: The CCITT X.21 Recommendation," *IBM Journal of Res. and Develop.*, January, 1978.

[ZAFI 80] Zafiropulo, P., "Towards Analyzing and Synthesizing Protocols," *IEEE Trans. on Commun.*, Vol. COM-28, April 1980, pp. 651-661.

(* article reprinted below.)