Communication Network Protocols: Formal Models and Analytic Methods

Simon S. Lam
Department of Computer Sciences
University of Texas at Austin
Austin, Texas 78712

This article consists of two parts: (1) a brief overview by the author of the status of this research area, and (2) a summary of discussions on this topic that took place during the NSF workshop.

Overview

What is a protocol? According to McQuillan and Cerf [1], "the word protocol has been borrowed from the parlance of conventional social behavior to describe the orderly exchange of information between separate pieces of equipment." We shall, however, give the word "protocol" a much broader meaning, and use it to refer to a set of parallel programs that provide one or more functions within a computer communication network environment. There is a wide variety of such functions. They include, for examples, the management of logical channels, the opening/closing of logical channels, reliable data transfers at different levels (data link, host-to-host, etc.), routing, flow and congestion controls, deadlock avoidance, multiple access in broadcast channels, concurrency control of distributed data bases, among many others. The reader is referred to [2] for protocol examples and details. Traditionally protocols involving user processes have been of interest to computer scientists, e.g., data base concurrency control protocols. On the other hand, low-level functions such as coding, modulation and clock synchronization are in the domain of electrical engineering. Research work in the middle ground (i.e. protocols for communication and network resource allocation) has been interdisciplinary. Much of such research has been on the development of performance models. Of particular interest have been mechanisms for implementing data link level functions and network level functions, such as ARQ error control techniques, multiple access techniques, routing algorithms, and flow and congestion control mechanisms. Different models are typically formulated to analyze independently different mechanisms. The models are often highly abstracted versions of the real mechanisms in order for them to be mathematically tractable. (Most are either queueing-theoretic or Markov chain models.) As such, they are not suitable for analyzing logical correctness properties of protocols.

Although research has been going on in the area of protocol verification for about a decade, a strong theoretical foundation is lacking and this remains a fertile area for future research. Especially needed are formal models and analytic methods that can accommodate the analyses of both the logical correctness properties and the performance characteristics (throughput, delay, etc.) of protocol systems.

Some examples of logical correctness properties of protocols are: freedom from deadlocks, livelocks or access conflicts; in-sequence delivery of data; reaching agreements of various kinds (the status of a connection, routing table updates etc.). Work on this topic is exemplified by formal models for the specification and verification of communication protocols [3], as well as the work on routing protocols that have loopfree and failsafe properties [4,5].

Formalisms and techniques for the specification and verification of parallel programs abound in the computer science literature. In protocol verification, the most widely used and successfully applied formalism by far is that of a state transition model. In this model, a protocol system is represented by a network of event-driven processes. Both protocol entities and communication channels are modelled as processes. The state of a process is given by the value of a variable or a set of variables. The state of the network is given by the value of all state variables in the network. Transitions between network states correspond to the occurrences of enabled events (i.e. the sending and receiving of messages by protocol entities). The "reachability graph" of such a network contains all available information on the logical properties of the network. (The reachability graph is a directed graph in which each node is a network state and each arc is a state transition corresponding to an event occurrence.)

Unlike queueing-theoretic models, state transition models are very close to real systems and can be converted to protocol implementations very easily. They are extremely powerful in the sense that almost any protocol can be specified as a network of such event-driven processes. But the reachability graph for any non-trivial protocol is typically very large (possibly infinite). As a result, to check whether a logical property holds for a protocol system, the straightforward method of traversing the reachability graph is not always computationally feasible.

There seem to be two general approaches to getting around the difficulty of a very large or infinite reachability graph. One is to avoid it altogether and employ proof techniques based upon induction. However, such proof techniques cannot be easily automated and considerable human ingenuity is necessary to construct proofs. Another technique that has been pursued primarily by this author and his colleagues is to construct "abstractions" of a given protocol that retain adequate information for proving specific protocol properties without knowledge of the protocol's reachability graph. Depending upon the protocol properties of interest, these abstractions may be abstractions of the protocol or abstractions of the reachability graph of the protocol [6,7,8].

In conclusion, formalisms for specifying protocols, for specifying protocol properties, and proof techniques for their verification present challenging problems to be addressed. Computer scientists have been studying the verification of parallel programs for quite a while. But communication protocols constitute a special type of parallel programs with its own special characteristics that, as far as I know, have not yet been adequately addressed.

A related research topic that is pretty much wide open is the development of protocol models that incorporate (1) time variables (timers, clocks etc.), or (2) probabilities and random variables, and the development of analytic techniques for such models. The incorporation of time variables is important because many communication and networking protocols are time-dependent systems, i.e. their correct functioning depends upon time relationships between remote events. The incorporation of probabilities in some manner into a protocol model permits the evaluation of a protocol system's performance (delay, throughput etc.) in probabilistic terms. These problems are similar to research on timed and stochastic Petri Nets. However the application of Petri Nets to the analysis of protocols has been limited to very trivial protocols [9].

Summary of Discussions

1. Professor M. Schwartz conjectured that an approach to reducing the complexity of protocol analysis is to determine whether some protocol properties will hold with certain confident levels (instead of holding absolutely). To do so, it was pointed out by this author that probabilities and/or random variables need to be incorporated somehow into the protocol model. Someone mentioned the similarity of such models to stochastic Petri Nets (Professor A. Lazar?).

2. Professors R. Gallager and P. Humblet asserted that instead of finding ways to handle the analysis of complex protocols it will be better to write correct protocols in the first place and compose complex protocols from simple protocols. This author agreed that this would be the ideal approach to getting correct protocols. However, since protocols are typically designed and implemented well before they are analyzed, methods to handle the analysis of complex protocols are needed.

3. Professor I. Rubin pointed out that real-life protocols are so complex that abstraction techniques that will permit us to consider mechanisms of a protocol one at a time are valuable.

4. Someone suggested "multi-user protocols" as a research topic.

   Note by author: most communication protocols involve two users engaged in a dialog. But most resource allocation protocols (routing, multiple access etc.) involve more than two users. To handle protocols involving N users or a network of users, various induction techniques have been proposed [7,10].

5. Professor M. Schwartz stressed that the performance analysis of protocol systems is still a very important problem area.

6. Professor R. Gallager proposed the following as a significant open problem: What classes of protocols are "impossible?"

7. Someone made the comment that we must accept the fact that all programs have bugs (Professor I. Rubin?). We might want to just determine the frequency of occurrence of such bugs.

8. Professor Schwartz suggested "protocol testing" as a research topic.

9. Lastly, this author suggested the topic of network security protocols that are concerned with the management and distribution of security keys (private or public) and the implementation of digital signatures.

References

[1] McQuillan, J. M. and V. G. Cerf, *A Practical View of Computer Communications Protocols*, IEEE Computer Society Press, 1978.

[2] Lam, S. S., *Communication and Networking Protocols*, IEEE Computer Society Press, 1984 (to appear).

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

[4] Merlin, P. M. and A. Segall, "A Failsafe Distributed Routing Protocol." *IEEE Trans. Comm.*, Vol. COM-27, No. 9, Sept. 1979, pp. 1280-1287.

[5] Gallager, R. G., P. A. Humblet and P. M. Spira, "A Distributed Algorithm for Minimum-Weight Spanning Trees," *ACM TOPLAS*, Vol. 5, 1983.

[6] Lam, S. S. and A. U. Shankar, "Protocol Verification via Projections." Technical Report 207, Dept. of Computer Sciences, University of Texas at Austin, August, 1982; to appear in *IEEE Trans. on Software Engineering*, July 1984.

[7] 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).

[8] 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.

[9] Molloy, M. K., "Performance Analysis using Stochastic Petri Nets," *IEEE Trans. on Computers*, September 1982, pp. 913-917.

[10] Merlin, P. M., "Specification and Validation of Protocols," *IEEE Transactions on Communications*, November 1979.