Some Research Problems and Directions in Computer Communications

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

1. Introduction

We have been investigating some fundamental problems in the modeling, analysis and construction of protocol systems. Although our concerns are applicable to distributed programs in general, we draw our examples and motivation primarily from the class of protocol systems that implement the various functions of a computer network. There is a wide range of technical problems here. They include, for examples, protocols for the management of logical connections, various handshaking protocols for the establishment of logical connections and the distribution of security keys, reliable data transfers at different levels (data link, host-to-host etc.), routing, flow and congestion control, avoidance of buffer deadlocks, multiple access in broadcast channels, concurrency control of multiple-copy databases, among others.

A general-purpose computer communication network needs to implement many, if not all, of the above-mentioned functions. The concept of a layered architecture helps to delineate a network's functions into different layers and organize the layers into a hierarchy. It is a step in the right direction towards a systematic approach for constructing software for a high-performance and reliable computer network. It is, however, a relatively small step towards that goal from the following observation. Each protocol layer in a typical layered architecture (ISO, SNA, ARPANET etc.) when implemented would be a highly complex set of distributed programs [CERF 78, CYPS 78, ANSI 81]. Here, "complexity" is measured in terms of the current state of the art in the design of distributed programs whose performance and 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.

There does not exist a systematic approach for the construction of a correct and efficient protocol system comparable in complexity to that of a protocol layer. In fact, there does not yet exist a sound theoretical foundation for analyzing the multi-faceted behavior of a protocol system having multiple functions to perform. Typically, for mathematical tractability, different models (abstractions) of a protocol system are used for analyzing independently the behavior/performance of different mechanisms in the system that implement the various protocol functions. Furthermore, depending upon the kind of performance characteristics under consideration (delay-throughput performance or some logical correctness property) a very different abstraction is used.

The failure to account for interactions between different mechanisms in a protocol system due to the use of different abstractions for these mechanisms will give rise to inaccurate performance predictions. This is undesirable but is often not disastrous. Now consider logical correctness properties that are verified for "toy protocols," which are abstractions of a multi-function protocol system. Such properties may be *totally invalid* for the protocol system itself if the abstractions have not been constructed to account for interactions between different mechanisms in the system [LAM 82d].

We have been working on the following three research topics:

- Analytic methods for network design based upon queueing network models with closed chains and population size constraints (intended as an alternative to Kleinrock's open-chain model [KLEI 76]).
- Establishment of a theoretical foundation for the use of abstractions to facilitate protocol verification and construction. This research is based upon the theory of protocol projections that we developed recently.
- Development of an event-driven process model that incorporates measures of time for the specification and verification of "time-dependent" protocol systems. Such a model is also intended to accommodate the analysis of a protocol system's throughput/delay performance characteristics (using simulation) as well as the verification of logical properties (via projections).

Our ultimate goal is a programming environment for the interactive design of protocol systems that have acceptable performance characteristics as well as desired logical properties. We give a brief description of our three research topics below.

2. A New Model for Network Design

Available analytic tools and methods for network design and analysis are mostly based upon an open-chain queueing network model, first formulated by Kleinrock in 1964 [KLEI 64, KLEI 76]. It has the advantage of having an explicit formula for the mean network delay in terms of the parameters of link (and processor) capacities and traffic flows, and is thus useful for optimal routing and channel capacity assignment studies [FRAT 73, GERL 73, GERL 77, SCHW 77]. However this model is an abstraction of a network in which the mechanisms of flow and congestion control are ignored. We investigated the conditions under which Kleinrock's model is accurate [LAM 81b]. We found that the interplay between routing and flow control has little effect on the accuracy of Kleinrock's model for networks that are lightly utilized. The accuracy of Kleinrock's model suffers significantly however when some of the links in the network are moderately or heavily utilized.

Queueing networks with closed chains and other population size constraints permit the modeling of a network's flow and congestion control constraints as well as its routing behavior [LAM 77, LAM 79, LAM 81a, LAM 81b, LAM 82a, LAM 82b, LAM 82c, LIEN

81]. A serious drawback of closed-chain models is that they require algorithmic solutions for network performance measures (virtual channel throughputs and their mean delays). The computational time and space requiremets of such algorithms grow exponentially with K, the number of closed chains (virtual channels). Hence, existing algorithms (designed primarily for multiprogramming system models) can only solve closed-chain models of networks with just a few virtual channels.

A contribution of ours (sponsored by a previous NSF grant) was the discovery, formulation and study of a new algorithm, called the tree convolution algorithm [LAM 83]. Our algorithm exploits the routing information of a given network which had not been utilized by previous algorithms. The tree convolution algorithm provides an *exact* solution to networks with tens of closed chains having time and space requirements that are manageable on current computers.

The objective of our current work is to develop methods for the design and analysis of networks including most of the following design variables:

- Network topology
- Capacities of resources
    - communication channels
    - nodal buffers
    - nodal processors
- Resource allocation protocols and their parameters
    - routing
    - flow control
    - congestion control
    - nodal buffer management

Our starting point is the tree convolution algorithm. We made the observation that this algorithm calculates a tree of arrays. Such a *tree-structured database* of information encapsulates the network parameters of interest, and can be used for network design in a role similar to that of Kleinrock's mean delay formula. Network design problems that require the evaluation of network performance measures when the network topology, routes, resource capacities and protocol parameters are perturbed can be obtained by suitable modifications to the tree database; a complete recalculation of all the information in the database is not needed. We are developing algorithms for use in conjunction with the tree database to address the following classes of problems:

1. analytic studies of network performance characteristics as functions of traffic levels, resource capacities and protocol parameters, and the behavior of any interactions between protocols;

2. development of various network design procedures based upon the tree database (e.g., routing of incremental flows, topology perturbations etc.)

To date, two accomplishments of ours are noteworthy [HSIE 84]. First, we derived a criterion for optimal channel capacity assignment for a closed-chain model of flow-controlled networks. Utilizing the tree convolution algorithm, the assignment of channel capacities to minimize the mean network delay subject to a cost constraint can be performed. This is a much more difficult problem than the channel capacity assignment problem formulated by Kleinrock, since our model includes the mechanism of window flow control.

We have also been investigating various approximate solution techniques for the class of queueing network models with closed chains and population size constraints. The accuracy of approximate solution techniques can be validated by the tree convolution algorithm over a sizeable region of the design space, thus avoiding expensive simulations. There are two reasons for having approximate solution techniques. First, approximate solution techniques are needed for extremely large networks (say, with 100 or more active virtual channels) that are beyond the capability of the tree algorithm. Second, approximate solution techniques are useful for speeding up intermediate steps in large optimization procedures. The tree algorithm can be used at various "checkpoints" in such procedures to avoid accumulation of errors.

We have also derived throughput bounds for both closed single-chain and multi-chain queueing networks. The throughput bounds of single-chain networks are very tight and can be calculated very efficiently. In addition, we have developed some approximation solution techniques for multi-chain networks that utilize these bounds. Comparisons with exact results calculated by the tree convolution algorithm indicate that the approximate techniques are accurate enough to be used in network design procedures.

3. Verification and Construction of Protocols

The theory of projections was developed by us to reduce the analysis of complex multi-function protocols to the analysis of simpler single-function "image protocols" [LAM 82d, SHAN 82]. Each image protocol is an abstraction of the given protocol but is specified just like a real protocol. Each image protocol is of the same complexity of toy protocols that are analyzed in the literature. Our construction method guarantees that each image protocol is faithful in the sense that any logical correctness property, safety or liveness property, that is valid for the image protocol must also be valid for the original protocol.

In general, to reduce the problem of directly constructing a multi-function protocol to that of composing it from several single-function protocols is a very difficult problem. However we observed that many real-life protocols go through different phases one at a time performing a distinct function in each phase. We developed a multi-phase model for such protocols [CHOW 83].

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 multi-function protocol is constructed by first constructing separate phases to perform its different functions. We developed a method to connect these phases together to implement the multi-function 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). The modularity inherent in multi-phase protocols facilitates not only their construction but also their understanding and modification. We found an abundance of real-life protocols that can be constructed as multi-phase protocols. Three fairly large protocol examples were constructed: (1) a version of IBM's BSC protocol for data link control, (2) a high-level session control protocol modeled after one in IBM's Systems Network Architecture, and (3) a token ring network protocol.

4. A Model for Time-Dependend Protocols

We consider time-dependent systems whose correct functioning depends upon time relationships between system event occurrences [SHAN 82, SHAN 84]. Real-life communication network protocols are invariably time-dependent systems. Specifically, we consider networks of processes that communicate exclusively by message-passing. In the context of communication network protocols, each process is either a communication channel or a protocol entity.

Each process is specified by a set of state variables and a set of events. Each event is described by a predicate that relates the values of the system state variables immediately before to their values immediately after the event occurrence. The predicate embodies specifications of both the event's enabling condition and action. Measures of time are explicitly included in our model. Furthermore, clocks are not coupled and they can tick at any rate within some specified error bounds. Inference rules for both safety and liveness are presented. Progress properties can be specified by both liveness assertions and real-time specifications. Liveness assertions are expressed in terms of inductive properties of bounded-length paths in a system's reachability space. Real-time specifications are stated as safety assertions.

We have applied our methodology to the verification of several large communication protocols including a version of the High-level Data Link Control (HDLC) protocol [SHAN 83].

In addition to modeling errors and timers in a more natural manner, there are several other advantages in favor of incorporating measures of time into a protocol system model. These are discussed below.

1. Time relationships are widely used in some 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].

2. The presence of time measures in our model facilitates the use of the same model to evaluate a protocol system's performance (response time, throughput etc.). The time measures are indispensable to a simulation environment. A unified model for analyzing both the logical behavior and performance characteristics of a protocol system allows us to identify and study tradeoffs in the design of the protocol system e.g., finding ways to increase a protocol's performance by relaxing some protocol rules which may not be critical for the logical correctness of the system's principal functions.
3. We can model various bounded time specifications of a protocol system's performance in addition to probabilistic measures of performance from simulation.
4. Although real-time specifications involving time variables are typically more complex (and less elegant) when compared to a temporal logic specification [SCHW 82] that is gaining popularity in current proof methods, they are generally safety assertions and not liveness assertions. Thus they are more amenable to the use of existing mechanical verification techniques and tools.

Acknowledgement

The work reported in this paper has been supported by the National Science Foundation under Grant No. ECS78-01803 and Grant No. ECS-8304734.

References

[ANSI 81] American National Standards Institute. "Data Processing--Open Systems Interconnection--Basic Reference Model." Document ISO/TC97/SC16. *Computer Networks*, Vol. 5, No. 2, 1981, pp. 81-118.

[CERF 78] Cerf, V. G. and P. T. Kirstein. "Issues in Packet-Network Interconnection." *Proc. IEEE* 66, 11, November 1978, pp. 1386-1408.

[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).

[CYPS 78] Cypser, R. J. *Communications Architecture for Distributed Systems.* Addison-Wesley, Reading, Mass., 1978.

[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 73] Fratta, L., M. Gerla and L. Kleinrock. "The Flow Deviation Method: An Approach to Store-and-Forward Network Design." *Networks*, No. 3, pp. 97-133, 1973.

[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.

[GERL 73] Gerla, M. *The Design of Store-and-Forward Networks for Computer Communications*. PhD thesis, Department of Computer Sciences, UCLA, January 1973.

[GERL 77] Gerla, M. and L. Kleinrock. "On the Topological Design of Distributed Computer Networks." *IEEE Trans. on Commun.*, January 1977.

[HSIE 84] Hsieh, C. T. "The Design of Store-and-Forward Communication Networks using a New Model," Ph.D. Thesis, Department of Computer Sciences, University of Texas, 1984 (in preparation).

[KLEI 64] Kleinrock, L. *Communication Nets: Stochastic Message Flow and Delay*. McGraw Hill, New York, 1964.

[KLEI 76] Kleinrock, L. *Queueing Systems Vol. 2: Computer Applications*. Wiley-Interscience, New York, 1976.

[LAM 77] Lam, S. S. "Queueing Networks with Population Size Constraints." *IBM Journal of Research and Development*, Vol. 21, July 1977.

[LAM 79] Lam, S. S. and M. Reiser. "Congestion Control of Store-and-Forward Networks by Input Buffer Limits--An Analysis." *IEEE Trans. on Commun.*, Vol. COM-27, Jan. 1979. Reprinted in *Advances in Computer Communications and Networks*, edited by W. W. Chu, Artech House, 1979.

[LAM 81a] Lam, S. S. and Y. L. Lien. "Congestion Control of Packet Communication Networks by Input Buffer Limits--A Simulation Study." *IEEE Trans. on Computers*, Vol. C-30, No. 10, October 1981.

[LAM 81b] Lam, S. S. and Y. L. Lien. "Modeling and Analysis of Flow Controlled Packet Switching Networks." *Proc. Seventh Data Communications* Symposium, Mexico City, October 1981, IEEE Computer Society Press.

[LAM 82a] Lam, S. S. "Dynamic Scaling and Growth Behavior of Queueing Network Normalization Constants." *Journal of the ACM*, Vol. 29, No. 2, April 1982.

[LAM 82b] Lam, S. S. and Y. L. Lien. "Optimal Routing in Networks with Flow-

Controlled Virtual Channels." *Performance Evaluation Review*, Vol. 11, No. 1, April 1982. (Computer Network Performance Symposium, University of Maryland.)

[LAM 82c] Lam, S. S. and J. W. Wong. "Queueing Network Models of Packet Switching Networks, Part 2: Networks with Population Size Constraints." *Performance Evaluation*, Vol. 2, No. 3, 1982.

[LAM 82d] Lam, S. S. and A. U. Shankar. "Protocol Verification via Projections." Technical Report 207, Dept. of Computer Sciences, University of Texas, August 1982; to appear in *IEEE Transactions on Software Engineering.*

[LAM 83] Lam, S. S. and Y. L. Lien. "A Tree Convolution Algorithm for the Solution of Queueing Networks." *Comm. ACM,* March 1983.

[LIEN 81] Lien, Y. L. *Modeling and Analysis of Flow-Controlled Computer Communication Networks.* PhD thesis, Dept. of Computer Sciences, University of Texas at Austin, December 1981.

[SCHW 77] Schwartz, M. *Computer-Communication Network Design and Analysis.* Prentice-Hall, Englewood Cliffs, N.J., 1977.

[SCHW 82] Schwartz, R. L. and P. M. Melliar-Smith. "From State Machines to Temporal Logic: Specification Methods for Protocol Standards." *IEEE Trans. on Commun.*, Vol. COM-30, December 1982.

[SHAN 82] Shankar, A. U. *Analysis of Communication Protocols via Protocol Projections.* PhD thesis, Dept. of Electrical Engineering, Univ. of Texas at Austin, December 1982.

[SHAN 83] Shankar, A. U. and S. S. Lam. "An HDLC Protocol Specification and its Verification using Image Protocols." *ACM Transactions on Computer Systems*, Vol. 1, No. 4, pp. 331-368.

[SHAN 84] Shankar, A. U. and S. S. Lam, "Proving Safety and Liveness of Time-Dependent Communication Networks." Draft Manuscript submitted for publication, 1984.

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