Algebraic Specification-Based Performance Analysis of Communication Protocols*

Nihal Nounou and Yechiam Yemini

Computer Science Department
Columbia University
New York, N.Y. 10027

CUCS-141-84

June 1984

ABSTRACT

Safe and live protocols have been shown to exhibit timing errors. To avoid such errors, timing requirements of protocols should be specified and verified. In this paper, a method for mapping algebraic functional behavior descriptions into corresponding timing behavior descriptions is introduced. Constraints on timing behavior are then expressed and used in specifying and verifying protocol timing requirements. In addition, various protocol performance measures are defined and analyzed. Using the Alternating Bit protocol as an example, an upper bound on the protocol's timeout rate, such that it meets a given timeout requirement, is computed and its maximum throughput and mean transfer time are analyzed.

*This research was supported in part by the Defense Advanced Research Projects Agency, under contract N00039-84-C-0165, Nasional Science foundation, under contract MCS-81-10319, and New York State Center for Advanced Technonolgy in Computers and Information Systems, under NYSSTF CAT(83)-8.

1. Introduction

The correct operation of a communication protocol may typically depend not only on given functional requirements (such as deadlock-freeness), but also on some timing requirements. Indeed, it has been shown [Yemi 82] that the Alternating Bit (AB) protocol [Bart 69], though proven to be safe and live, could take forever to achieve its goals. In order to avoid this timing error, timing requirements should be specified and verified. One objective of this paper is to present a method for specifying such requirements and computing optimal values of protocol parameters in order to meet them. Another objective is to define and analyze protocol performance measures. Based on these measures, performance of protocols is predicted and the effect of the various protocol parameters on it is analyzed.

Instead of following the *direct approach* to analyzing protocol performance, in which a protocol performance model is extracted from first principles [Tows 79, Yu 79, Bux 80], we introduce an *algebraic specification-based approach*. In this approach, protocol functional behavior is described using an algebraic specification method that is a variant of Milner's Calculus of Communicating Processes (CCS) [Miln 80]. The corresponding protocol timing behavior is then obtained through algebraic mappings from the functional behavior to protocol probability and time attributes. Formal protocol functional specifications are the basis of any protocol functional analysis (e.g., formal verification, testing); similarly, a formal specification of protocol timing behavior could be the basis of various protocol timing and performance analyses.

There are several advantages of a specification-based approach to analyzing protocol performance. First, it facilitates the automation of protocol performance analysis, thus saving human ingenuity and time often involved in the direct approach. Second, it can be integrated with other protocol analysis tools (such as verification) in a protocol development environment. Third, it facilitates predicting protocol performance starting from early design phases when formal functional specifications are usually available.

Previous works on the specification and verification of protocol timing requirements by Shankar et al [Shan 82] assume only deterministic values for times between protocol events and assume that these requirements must be always satisfied. We assume that times between protocol events are represented by random variables which could possibly have deterministic or other distributions and allow for timing requirements to be specified such that time constraints (used in expressing timing requirements) are satisfied with a given desirable probability. These capabilities allow us to compute optimal settings of protocol parameters to meet given timing requirements. Also, previous work on specification-based evaluation of protocol performance measures done by Molloy [Moll 81] (the specification model used is petri nets with transitions in the net being augmented with firing rates) assumes a markov process as a model for the stochastic behavior of protocols thus limiting distributions of transition firing rates to have only exponential distribution. The markov model is then solved for probabilities of protocol states from which some performance measures such as throughput can be evaluated. In this paper, we choose a protocol timing model that allows for any distribution of times between events and both probability and time attributes of the model are considered.

The work described in this paper is part of Columbia's Unified Protocol Implementation and Design environment (CUPID) project [Yemi 83], which utilizes algebraic specifications as canonical representations of protocols. One of the project's goals is to design algorithms for translating other protocol specification (e.g. finite state machines) into this canonical representation. Therefore, the results given in this paper are likely to be applicable to other protocol representations.

The organization of this paper is as follows: the algebraic specification method is described and then applied to the AB protocol in section 2. Protocol timing behavior is defined and discussed in section 3. In section 4, methods for specifying and verifying protocol timing requirements and evaluating its throughput and transfer time are presented. Applying these concepts to the AB protocol, a timeout requirement is specified and verified resulting in computing an upper bound on the protocol's timeout rate and the protocol's maximum throughput and mean transfer time are analyzed. Finally, we summarize the results of this paper in section 5.

2. Algebraic Specifications of Protocols

Algebraic specification derives its name from its relationship to universal algebra [Grat 88]. A universal algebra (briefly algebra) consists of a nonempty set of *objects* and a set of *operations*. Each operation takes a finite number of inputs from the set of objects and produces an element of the set of objects. *Equational-axioms* are an important aspect of an algebra since they define the semantics of expressions in that algebra. In appendix I, definitions of some related algebraic concepts to be used in this paper are given.

The algebraic approach to specifying protocols has been employed in the AFFIRM system [Suns 82] in which protocols are modeled in terms of abstract data types and state-transition machines. We use the algebraic approach with an event-based protocol model described as follows. Typically, a protocol functional model consists of *functional* local processes that are distributed and that communicate via send and receive events across *interface points* or *ports*. We assume that only two processes can own pairs of send and corresponding receive ports, thus indicating a *one-to-one* addressing between the communicating processes; and that interactions between the processes are modeled via rendezvous interaction events. In what follows, a variant of the CCS algebraic specification method is used to formally capture the communication behavior of such a protocol model.

2.1. An Algebraic Specification Method

Let $\Upsilon = \Sigma \cup \overline{\Sigma} \cup \mathcal{T}$ be a set of *event symbols* whose components are defined as follows:

Σ — is a fixed set of send-event symbols.

$\overline{\Sigma}$ — is a set of receive-event symbols obtained by affixing a bar to elements of Σ.

$\mathcal{T}$ — is a set of rendezvous-event symbols of the form τ_g where g ranges over Σ.

Definition 1: A *functional algebra* over $\mathcal{T}$ is the free algebra over the set $\mathcal{T}$ generated by the following operations:

$\bullet$	(binary) *sequential composition*
$+$	(binary) *non-deterministic choice*
$\times$	(binary) *concurrent composition*
$\emptyset$	(nullary i.e. constant) *inaction or deadlock*

$\mathcal{F} = (\mathcal{B}\ ;\ \bullet,+,\times,\emptyset)$ denotes such a functional algebra. Elements of $\mathcal{B}$ are called *behavior expressions* (BEs). An example of equational-axioms of $\mathcal{F}$ is A+B $\equiv$ B+A (commutativity), where "$\equiv$" is the *observational equivalence* relation introduced by Milner. Unlike the algebra of regular events [Salo 66] the distributive law A$\bullet$(B+C) $\equiv$ A$\bullet$B + A$\bullet$C is not an axiom of $\mathcal{F}$ since the nondeterministic choices of behaviors on both sides of the axiom are different (between B and C on the left hand side; but between A$\bullet$B and A$\bullet$C on the right hand side) and hence the two expressions are not observationally equivalent. The complete set of axioms will be provided in a forthcoming paper. Throughout the rest of this paper, the following notations will be used: lower-case italic letters to range over Σ, τ with lower-case letter subscripts to range over $\mathcal{T}$ and capital letters to range over $\mathcal{B}$.

A *process specification* describing the process communication behavior may be expressed as a BE. Such a BE may be either given explicitly or as a set of *behavior equations* of which it is the unique solution. Assuming a fixed set X of identifier symbols (capital italic letters will be used to range over X), each behavior equation is a pair of polynomials $p[X]$ each of which is an element of the free algebra $\mathcal{F}[X]$ over $\mathcal{B} \cup X$. We are interested only in (possibly recursive) equations of the form $I=p[X]$. Conditions for obtaining unique solutions of such equations can be found in [Miln 80]. A collection of protocol local process specifications constitute a *protocol specification*; the concurrent composition of these local specifications produces a *global* protocol specification.

Assuming a general form of the binary choice operation "+" given as the *n-ary* choice operation

$$\sum_{i=1}^{n}$$

Then, one can express the concurrent composition operation in terms of the sequential composition and *n-ary* choice operations. Consider the concurrent composition of behavior expressions A and B. Let the *name* be a function of the set of events such that $name(a)=name(\bar{a})=a$ and $Scope(A,B)$ be a function of a pair of BEs denoting the set of event *names* on which the process described by A and the process described by B interact. A recursive definition of concurrent composition is then as follows

Let $A=\sum_{i=1}^{n} a_i \bullet A'_i$ and $B=\sum_{j=1}^{m} b_j \bullet B'_j$

then,

$$A\times B = \sum_{i=1}^{nn} a_i \bullet (A'_i \times B) \quad \text{for all } a_i \text{ such that } name(a_i) \notin Scope(A,B)$$

$$+ \sum_{j=1}^{mm} b_j \bullet (A\times B'_j \quad \text{for all } b_j \text{ such that } name(b_j) \notin Scope(A,B)$$

$$+ \sum_{b_j=\overline{a_i}} \tau_{a_i} \bullet (A'_i \times B'_j) \qquad 2.1$$

where $nn \leq n$ and $mm \leq m$ and

$A \times \emptyset = A$

Informally, concurrent composition of two BEs produces rendezvous events of the pairs of send and receive events belonging to the two BEs and all possible interleavings of all the other events appearing in the two BEs. Conditions given for the first two terms resulting from composition ensure a one-to-one type of addressing between the communicating processes since events on which the two process BEs interact can only contribute to rendezvous events and are not available for future composition with another process BE that might include an event with the same *name*. Note that one can compose global processes as well as local processes; rendezvous events appearing in a global process BE are then called *internal* rendezvous events since they are results of previous composition(s) and therefore are internal to the global process.

The above described algebraic model differs from CCS in three respects. First, we differentiate between τ actions according to the identity of the send and receive events involved. This distinction is necessary for the purpose of our analysis since different τ actions might have different timing behavior attributes. Second, addressing is one-to-one as opposed to many-to-many in CCS. Third, we excluded the restriction and relabeling operations that were needed in CCS to make the concurrent composition operation, in which addressing is many-to-many, associative and for generating copies of processes (an application that is unnecessary for protocol specifications) respectively.

2.2. An Algebraic Specification of the Alternating Bit Protocol

The AB protocol's consists of four functional processes: sender S, receiver R, sender-to-receiver transmission medium D, and receiver-to-sender transmission medium A. Fig. 2-1 depicts the model and the ports across which the processes communicate. In this figure, d,d' represent send ports for message d (with 0 or 1 value) from S and D respectively, and a,a' represent send ports for acknowledgment a (with 0 or 1 value) from R and A respectively. Similarly, events with overbar represent corresponding receive ports. Due to the symmetry of the protocol for the two half cycles of 0 and 1 bit values, we will only consider the

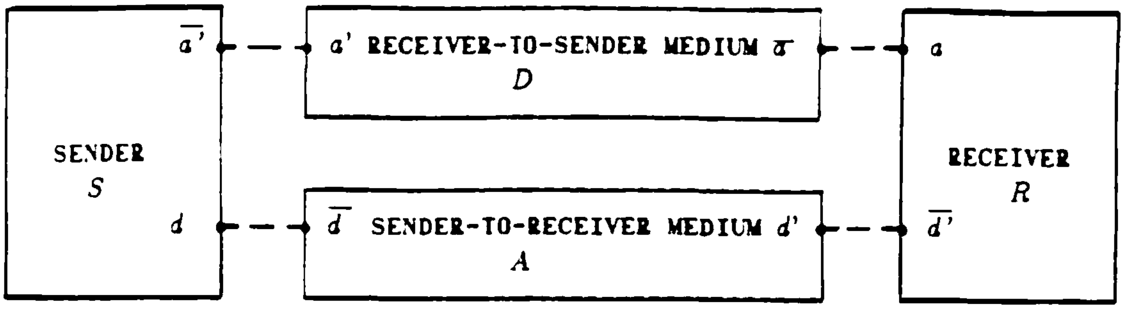


Figure 2-1: Functional model of the AB protocol

protocol operation during a single half cycle. We assume messages and/or acknowledgments may be lost and that there is an upper bound of 2 on the number of messages (acknowledgments) traveling through mediums at any one time. The latter assumption is used only to keep the length of the specifications manageable and will not restrict our analysis as will be explained in section 4.

Let τ_s, τ_x, τ_T, τ_{ld}, and τ_{la} denote internal rendezvous events resulting from the composition of the sender with a source process, the sender with a timer process (for timeout), the sender with the d_1 medium process (d_0) in the first (second) half cycle of the protocol, the message and acknowledgment mediums with a loss process respectively. Also, let τ_u, τ_v, τ_y and τ_z denote internal rendezvous events representing end of cycle operation with each of S, D, R and A with corresponding control processes respectively. Specifications of the behavior identifiers S, D, R and A denoting the four local processes are given by the following set of equations

$$S = \tau_s \cdot d \cdot S_1 \qquad S_1 = (\tau_x + \tau_T) \cdot d \cdot S_1 + \overline{a'} \cdot \tau_u$$

$$D = \overline{d} \cdot D_1 \qquad D_1 = (\tau_{ld} \cdot \overline{d} + \overline{d} \cdot [\tau_{ld} + d']) \cdot D_1 + d' \cdot [D + \tau_v]$$

$$R = \overline{d'} \cdot a \cdot [R + \tau_y]$$

$$A = \overline{a} \cdot A_1 \qquad A_1 = (\tau_{la} \cdot \overline{a} + \overline{a} \cdot [\tau_{la} + a']) \cdot A_1 + a' \cdot [A + \tau_z] .$$

Informally, the sender's specification indicates that upon a rendezvous interaction with a source it sends a message and then exhibits a repetition of the following behavior: it interacts with the medium on an incorrect acknowledgment or with the timeout timer, both of which cause it to send another copy of the same message. This repeated behavior ends upon receiving a correct acknowledgment and hence ending the half cycle. The receiver's specification indicates a repetition of receiving a message and sending its acknowledgment which ends upon receipt of the last message of the cycle and sending its acknowledgment. Both the message and acknowledgment mediums upon receiving data exhibit a repetition of losing data and then receiving another copy of the same message type, receiving another copy and losing one of the data copies, or delivering the data and then receiving another copy. This repeated behavior ends with either delivering the last copy of data received and then receiving another copy of the same data and going through the same repetitive behavior described above or delivering the last copy of data received and ending the half cycle.

The global protocol specification G is computed in two steps; this is possible since one of the axioms of $\mathcal{I}$ states that the concurrent composition operation is associative. First, the sender process is composed with the message medium to obtain G_1 and the receiver with the acknowledgment medium to obtain G_2. These two intermediate specifications, G_1 and G_2, are then composed to obtain G. To compute the concurrent composition of any two BEs, one has to provide the *Scope* set of the processes they represent and then use eq. 2.1 to get the *composite* BE. So, if

$$Scope(S,D) = \{d\} \text{ then}$$

$$G_1 = S \times D = \tau_s \bullet \tau_d \bullet B$$

$$B = S_1 \times D_1 = \overline{a'} \bullet (d' \bullet \tau_v \times \tau_u) + d' \bullet [\overline{a'} \bullet \tau_u \times \tau_v + (\tau_x + \tau_T) \bullet \tau_d \bullet B]$$
$$+ \tau_{ld} \bullet (\tau_x + \tau_T) \bullet \tau_d \bullet B + (\tau_x + \tau_T) \bullet [\tau_{ld} \bullet \tau_d + \tau_d \bullet (\tau_{ld} + d')] \bullet B$$

Similarly, if

$$Scope(R,A) = \{a\} \text{ then}$$

$$G_2 = R \times A = \overline{d'} \bullet \tau_a \bullet C$$

$$C = R \times A_1 = a' \bullet (\tau_z \times \tau_y) + \overline{d'} \bullet [\tau_{la} \bullet \tau_a + \tau_a \bullet (a' + \tau_{la})] \bullet C + \tau_{la} \bullet \overline{d'} \bullet \tau_a \bullet C$$

Finally, if

$$Scope(G) = \{a', d'\} \text{ then}$$

$$G = G_1 \times G_2 = \tau_s \bullet \tau_d \bullet E$$

$$E = B \times G_2 = (\tau_x + \tau_T) \bullet F + \tau_{ld} \bullet (\tau_x + \tau_T) \bullet \tau_d \bullet E + \tau_{d'} \bullet H$$

$$F = \tau_{ld} \bullet \tau_d \bullet E + \tau_d \bullet [\tau_{d'} \bullet (B \times (\tau_a \bullet C)) + \tau_{ld} \bullet E]$$

$$H = (\tau_x + \tau_T) \bullet (\tau_d \times \tau_a) \bullet J + \tau_a \bullet I$$

$$I = \tau_{a'} \bullet (\tau_u \times \tau_v \times \tau_y \times \tau_z) + (\tau_x + \tau_T) \bullet \tau_d \bullet J + \tau_{la} \bullet (\tau_x + \tau_T) \bullet \tau_d \bullet E$$

Specification of behavior identifier J used above in G is given in appendix II in a complete specification of G. Examining the composite process specification G shows timeout occurring possibly at any instant between sending a message and successfully receiving its acknowledgment. In particular, timeout may occur in cases that do not involve medium loss. Consequently, one might get a repetition of undesirable timeouts causing the flooding of the transmission mediums with message copies and their acknowledgments and thus leading to the AB timing error reported in [Yemi 82]. So, before computing the AB global specification G we need to address this problem. Generally, in order to study such timing problems and predict protocol performance, protocol timing behavior is introduced next.

3. Protocol Timing Behavior

In this section we will propose a model for protocol timing behavior and provide algebraic functions to map a given protocol functional specification into attributes of this model. Assume that the observer of protocol behavior has a clock. Then, it is possible to observe the occurrence time of each event of that behavior. The occurrence time of a send or receive event is the time it is offered and the occurrence time of a rendezvous event is the time of completing the rendezvous interaction. At each of these times there might be a choice of events that can occur because of the possible non determinism involved in protocol behaviors. We assume that these occurrence times are modeled as continuous random variables.

Such protocol *timing behavior* is best modeled as a *marked point process* [Snyd 75] $[\{t_i, \mathcal{M}_i\},\ i \geq 0]$, where t_i is the random variable denoting the i-th occurrence time and $\mathcal{M}_i$ is the i-th mark denoting the set of possible events that can occur at t_i. In order to fully describe protocol timing behavior, *attributes* of the marked point process modeling this behavior have to be specified. We choose these attributes such that they are functions of the protocol BEs. So, attributes of protocol timing behavior are marks, *time durations* and probabilities of possible sample behaviors. Time durations are defined as differences between occurrence times, and hence are functions of BEs while occurrence times are functions of events. Behavior probabilities become an attribute of the marked point process when instead of being contended with just nondeterminism of behaviors (a qualitative aspect of behaviors that implies equally probable choices of behaviors), we are interested in a probabilistic (qualitative) representation of this nondeterminism.

Next, we will formally define these attributes as functions of BEs and provide evaluation rules for them. First, since protocol timing behavior, as described above, is based on observations of the dynamic execution of the protocol, we will consider how one can follow this execution given its functional specification. To do that, the following definitions are required. Let a *behavior derivative* function ∂: $\mathcal{B} \times \mathcal{B} \longrightarrow \mathcal{B}$ to be defined as follows

$$\text{Let } A = \sum_{i=1}^{n} \mathrm{A}_i \cdot \mathrm{B}_i \quad \text{then,}$$

$$\partial_{\mathrm{E}}(A) = \mathrm{B}_i \text{ if for some } i,\ 1 \leq i \leq n,\ \mathrm{E} = \mathrm{A}_i$$

$$= \emptyset \text{ otherwise} \qquad 3.1$$

Since the functional operation "$\cdot$" denotes sequential composition, then $\partial_{\mathrm{E}}(\mathrm{A})$ is the BE that occurs after the occurrence of behavior E in A.

Example 3.1 :

Let $A = a \bullet (b + c) \bullet d + e$

then using eq. 3.1

$$\partial_a(A) = (b + c) \bullet d \quad \text{and}$$

$$\partial_{(b + c)}(\partial_a(A)) = d$$

Definition 2: A *choice set* $CH(.) \subseteq \mathcal{T}$ is a function of $\mathcal{B}$ that is defined recursively by

C1. $CH(A \bullet B) = CH(A)$ C2. $CH(A + B) = CH(A) \cup CH(B)$

C3. $CH(a) = \{a\}$ C4. $CH(\emptyset) = \{\}$ i.e., empty set.

Example 3.2 :

Consider A given in example 3.1 and using C2

$$CH(A) = CH(a \bullet (b + c) \bullet d) \cup CH(e)$$

then using C1

$$CH(A) = CH(a) \cup CH(e)$$

finally using C3

$$CH(A) = \{a.e\}$$

Given a protocol specification C, one can describe the execution of the protocol starting at time t_0 as follows. At t_0, C represents all possible samples of protocol functional behavior observed starting from this occurrence time. A choice set $CH(C)$ is then associated with t_1 meaning that any event belonging to this set can possibly occur at that time. Then, computing $\partial_a(C)$ (which represents behavior observed after the occurrence of a), for every $a \in CH(C)$, a *collection* of choice sets $CH(\partial_a(C))$ is associated with t_2. This procedure of computing derivatives and then the collection of choice sets continues on until one gets a collection of empty choice sets signalling the end of protocol execution. A mark M_i in the timing behavior of C is then represented by the collection of choice sets at t_i, thus denoting the possible events that can occur at that time.

Note that at any t_i during an instance of protocol timing behavior, only one choice set can be *enabled* i.e. any of events belonging to this set can occur at t_i. Only events belonging to an enabled choice set can occur. So if an occurrence time t_i has associated with it a collection of more than one choice set, which one of them is enabled depends on which event actually occurred at t_{i-1}. For example, if a belonging to a choice set $CH(A)$ associated with t_{i-1} occurs, then only the choice set $CH(\partial_a(A))$ is enabled at t_i. When observing an instance of protocol behavior, at each occurrence time one choice set belonging to the mark of that time is enabled and hence any of its events could occur at the next occurrence time; accordingly another choice set is enabled at the following occurrence time and the same procedure goes on till protocol execution stops (if it is a terminating protocol).

Definition 3: A *probability attribute* is a mapping from $\mathcal{B}\times\mathcal{B}$ to the set $0\leq R^{+}\leq 1$.

Let $\mathbf{P}_A(B)$ denote such a probability attribute representing the probability of occurrence of B starting at the occurrence time with which A is associated. $\mathbf{P}_A(A)$ is then equal to 1.

Definition 4: A *time attribute* is a mapping from $\mathcal{B}\times\mathcal{B}$ to a set of random duration time variables.

Let $\mathbf{T}_B(A)$ denote such a time attribute representing the time duration of behavior A starting at the occurrence time with which behavior B is associated (recall from our description of protocol execution that each occurrence time has a BE associated with it).

In our discussion above for mapping protocol functional behavior into its corresponding timing behavior, we only required the protocol BE to be given. To evaluate probability and time attributes we need in addition to that the probability distributions of the random *event times* between the occurrence time of the concerned event and that of its immediate predecessor event. We assume that these random variables are continuous and mutually independent.

Evaluation rules for probability attributes, and ***mean*** and ***variance*** statistics of time attributes are defined recursively in the following theorems respectively. We use $\mu_C(A)$ to represent the mean of $\mathbf{T}_C(A)$ and $\sigma_C(A)$ to represent its variance. Also, $F_a(t)$ and $f_a(t)$ will denote the probability distribution and density functions of the event time of a.

Theorem 3.1

P1. $\mathbf{P}_C(A.B) \quad = \quad \mathbf{P}_C(A)\ \mathbf{P}_{\partial_A(C)}(B)$

P2. $\mathbf{P}_C(A + B) \quad = \quad \mathbf{P}_C(A) + \mathbf{P}_C(B)$

P3. $$\mathbf{P}_C(a) \quad = \quad \int_0^{\infty} \prod_{e_i \in CH(C)\, e_i \neq a} [1 - F_{e_i}(t)]\ f_a(t)\ dt \qquad \textit{(minimum rule i)}$$

Proof P1 follows from noting that $\mathbf{P}_{\partial_A(C)}(B)$ is a conditional probability on the occurrence of A (since the choice set of behavior $\partial_A(C)$ is only enabled after A occurs) and P2 follows from noting that possible behaviors in a nondeterministic choice are mutually exclusive. Proof of P3 is given in appendix III.

Theorem 3.2

M1. $\mu_C(A \bullet B) \quad = \quad \mu_C(A) + \mu_{\partial_A(C)}(B)$

M2. $\mu_C(A + B) \quad = \quad [P_C(A)\mu_C(A) + P_C(B)\mu_C(B)]/P_C(A+B)$

M3. $\mu_C(a) \quad = \quad \int_0^\infty \prod_{e_i \in CH(C)} [1 - F_{e_i}(t)]dt$ *(minimum rule ii)*

Proof M1 and M2 follow from basic probability theory. Proof of M3 is given in appendix III.

Theorem 3.3

V1. $\sigma_C(A \bullet B) \quad = \quad \sigma_C(A) + \sigma_{\partial_A(C)}(B)$

V2. $\sigma_C(A+B) \quad = \quad \{P_C(A)[\sigma_C(A) + \mu_C^2(A)]+P_C(B)[\sigma_C(B) + \mu_C^2(B)]\}/P_C(A+B) - \mu_C^2(A+B)$

V3. $\sigma_C(a) \quad = \quad 2\int_0^\infty t \prod_{e_i \in CH(C)} [1 - F_{e_i}(t)]dt - \mu_C^2(a)$ *(minimum rule iii)*

Proof V1 and V2 follow from basic probability theory. Proof of V3 is given in appendix III.

Note that the event times whose distribution and density functions are used in P3. M3 and V3 are to be measured from the occurrence time of the immediate predecessor event. To avoid having to scale provided data for these event times according to the identity of this predecessor event we assume that all event time variables are exponentially distributed (by virtue of this distribution's memoryless property). Then P3, M3 and V3 are reduced to

$$P_C(a) = \lambda / \sum_{e_i \in CH(C)} \lambda_i$$

$$\mu_C(a) = 1 / \sum_{e_i \in CH(C)} \lambda_i$$

$$\sigma_C(a) = 1/\{ \sum_{e_i \in CH(C)} \lambda_i \}^2$$

Where λ_i and λ denote the rates of exponentially distributed event times of $e_i \in CH(C)$ and $a \in CH(C)$ respectively.

Given a recursively defined BE, one would expect that statistics of its time attribute would be the same as that of a geometrically distributed random variable. The next theorem formally states that this result can be obtained by only using evaluation rules of theorems 3.1, 3.2 and 3.3.

Theorem 3.4

Let $A = \mathrm{X} \bullet A + \mathrm{B}$, then

$$\mu_A(A) = \mathbf{P}_A(\mathrm{X})/[1-\mathbf{P}_A(\mathrm{X})]\ \mu_A(\mathrm{X}) + \mu_A(\mathrm{B})$$

$$\sigma_A(A) = \mathbf{P}_A(\mathrm{X})/[1-\mathbf{P}_A(\mathrm{X})]\ [\sigma_A(\mathrm{X}) + 1/[1-\mathbf{P}_A(\mathrm{X})]\mu_A^2(a)] + \sigma_A(\mathrm{B})$$

Proof See appendix IV.

In summary, we have proposed in this section a model for protocol timing behavior whose attributes are obtained through algebraic mappings of the protocol functional specification. Next, we will examine using such timing behavior in analyzing protocol performance.

4. Protocol Performance Analysis

Given a protocol timing behavior, one can analyze its performance in two respects. First, timing requirements on protocol timing behavior can be specified and verified. Second, after meeting the given timing requirements, some performance measures of the protocol can be defined and analyzed. The first task is addressed in section 4.1 and the second in section 4.2.

4.1. Specification and Verification of Timing Requirements

The objective of protocol functional analysis is to specify and verify protocol functional requirements (i.e. safety and liveness requirements). Meeting these functional requirements is often insufficient to ensure that a protocol achieves its goals since the correct operation of protocols usually depends on timing requirements. Therefore, protocol timing requirements should also be specified and verified. Specification and verification of timing requirements are analogues to the specification and verification of functional requirements. Timing requirements are expressed in terms of predicates, to be referred to as *time constraints*, on protocol timing behavior. However, unlike functional requirements which assert that safety and liveness predicates are *always* satisfied, timing requirements could only state that time constraints would be satisfied with a very high probability due to the random nature of protocol timing behavior.

In expressing time constraints, first order predicate calculus operations on the attributes of protocol timing behavior together with the following operation will be used.

Definition 5: "$\lessdot$" is a ternary relation on $\mathcal{T} \times \mathcal{B} \times \mathcal{T}$ such that given $a \lessdot_A b$ (read as a overrides b in the timing behavior of A) then

$\forall$ derivatives of A, C : $a,b \in CH(\mathrm{C})$ implies $\mathbf{P}_{\mathrm{C}}(b)=0$

In other words, $a \ll_{\mathrm{A}} b$ means that if in the timing behavior of A, a and b belong to the same choice set i.e., b and a are competing for an occurrence time, then b would never win since its probability attribute relative to this choice set is zero. Note that whenever in the timing behavior of A representing a choice between a number of BEs, the probability attribute of one of these BEs relative to A is zero, then the term in which this BE appears could be dropped from the functional specification of A.

Given a behavior A and a time constraint $a \ll_{\mathrm{A}} b$, how can one know if A satisfies this time constraint? There are three cases to consider. First, if for all derivatives of A, C, a and b do not belong to $CH(\mathrm{C})$ (i.e., left hand side of the implication in the definition above is false) then $a \ll_{\mathrm{A}} b$ is satisfied with probability 1. Second, if the event times given for the timing behavior of A are deterministic, then $a \ll_{\mathrm{A}} b$ is satisfied with probability 1 if the event time of a is less than that of b since then for all derivatives of A, C, we have $\mathbf{P}_{\mathrm{C}}(b)$ (probability that the event time of b is the minimum among those of elements of $CH(\mathrm{C})$) is equal to zero. Third, for the general case when such event times are random variables, $\mathbf{P}_{\mathrm{C}}(b)$ takes on any value from0 to 1 and the objective then is that A satisfies the time constraint with a probability close to 1. Hence, the timing requirement of A would have the form

Probability (A satisfies $a \ll_{\mathrm{A}} b) \geq 1 - \epsilon$
where ϵ is a very small probability error such as 10%.

From behavior A and time constraint $a \ll_{\mathrm{A}} b$, one can compute another behavior B which is the same as A except that it always satisfies the time constraint i.e., whenever the left hand side of the implication in definition 4 is true, the corresponding term is dropped. The probability that A satisfies $a \ll_{\mathrm{A}} b$ is then the same as $\mathbf{P}_{\mathrm{A}}(\mathrm{B})$ as illustrated in the following example.

Example 4.1: Let $A = c{\bullet}(a{\bullet}b + b{\bullet}a)$

and given the time constraint $a \ll_{A} b$, then the set of sample behaviors belonging to A that satisfy the time constraint is given by

$B = c{\bullet}(a{\bullet}b)$

and thus Probability(A satisfies $a \ll_{\mathrm{A}} b) = \mathbf{P}_{A}(B)$

where the probability attribute $\mathbf{P}_{A}(B)$ can be computed using theorems 3.4 and 3.1.

Let us now examine how can we use timing requirements in addressing the timeout problem of the AB protocol discussed in section 2 in which timeout was shown to possibly occur at any instant in time between τ_d and τ_a, in the protocol's global specification G. Timeout is introduced in the AB protocol and other retransmission protocols in order to recover from situations in which messages or acknowledgments are lost. However, for such protocols to function correctly, a timing requirement is required stating that the timeout rate should be set such that unnecessary retransmissions are minimized. Unnecessary retransmissions are those caused by a premature timeout i.e., timeout in cases when no messages or acknowledgments were lost. In order to specify such a timeout requirement, the following timeout constraints that state that τ_{ld} τ_{ld} τ_a $\tau_{d'}$, and $\tau_{a'}$ all override τ_T are needed.

TC1. $\tau_{ld} <_G \tau_T$ TC2. $\tau_{la} <_G \tau_T$ TC3. $\tau_a <_G \tau_T$
TC4. $\tau_{d'} <_G \tau_T$ TC5. $\tau_{a'} <_G \tau_T$

then the timeout-constraint TC is given by

TC = TC1 and TC2 and TC3 and TC4 and TC5

and the AB timeout requirement is specified as

$$\mathrm{P}_G(G_{TC}) \geq 1\text{-}\epsilon$$

Where G_{TC} refers to the set of behavior sequences of the global AB specification G that satisfy TC.

In order to compute the probability attribute in this timeout requirement, first G_{TC} has to be computed and then its probability evaluated. For simplicity we assume a negligible rate for τ_x relative to other event time rates in the global BE. thus leading to negligible probability of behavior sequences including τ_x. Note that τ_x denotes an internal rendezvous event of the sender process between the sender and the acknowledgment medium of a previous half cycle in the AB protocol. Consequently, its rate depends on the number of retransmissions in the previous half cycle; considering it in the analysis would require studying the stability of the protocol random point process which is beyond the scope of this paper. From the specification of G in section 2., G_{TC} is given by the set of equations

$$\begin{aligned}
G_{TC} &= \tau_s \cdot \tau_d \cdot E_{TC} \\
E_{TC} &= \tau_{ld} \cdot \tau_T \cdot \tau_d \cdot E_{TC} + \tau_{d'} \cdot H_{TC} \\
H_{TC} &= \tau_a \cdot I_{TC} \\
I_{TC} &= \tau_{a'} \cdot (\tau_u \times \tau_v \times \tau_y \times \tau_z) + \tau_{la} \cdot \tau_T \cdot \tau_d \cdot E_{TC}
\end{aligned}$$

where E_{TC}, H_{TC} and I_{TC} denote E, H and I that satisfy TC.

Using theorem 3.4 and P1 and P2 in theorem 3.1, the probability of G satisfying TC is given by

$$\mathrm{P}_G(G_{TC}) = (1/1\text{-}p_{loop})\ \mathrm{P}_E(\tau_{d'})\ \mathrm{P}_H(\tau_a)\ \mathrm{P}_I(\tau_{a'}) \geq 1 - \epsilon \qquad 4.1$$

where $p_{loop} = \mathrm{P}_E(\tau_{ld}) + \mathrm{P}_E(\tau_{d'})\ \mathrm{P}_H(\tau_a)\ \mathrm{P}_I(\tau_{la})$

Choice set	Elements of choice set
$CH(E)$	τ_T τ_{ld} $\tau_{d'}$
$CH(H)$	τ_T τ_a
$CH(I)$	τ_T $\tau_{a'}$ τ_{la}

Table 4-1: Choice sets of the timing behavior of G

Probability attributes of events are then computed using P3 in theorem 3.1 and the choice sets listed in table 4-1.

Next we solve for the value of the timeout rate such that the timeout requirement is met. Let $\tau_d \sim \lambda_d$ (i.e., τ_d has exponential rate of λ_d), $\tau_{d'} \sim \lambda_{d'}$; $\tau_a \sim \lambda_a$; $\tau_{a'} \sim \lambda_{a'}$; $\tau_{ld} \sim \lambda_{ld}$; $\tau_{la} \sim \lambda_{la}$; $\tau_u \sim \lambda_u$; $\tau_v \sim \lambda_v$; $\tau_y \sim \lambda_y$; $\tau_z \sim \lambda_z$; and $\tau_T \sim \lambda_T$. Note that in the timing behavior of G, λ_d, λ_a represent message and acknowledgment transmission rates respectively; similarly $\lambda_{d'}$, $\lambda_{a'}$ represent the sum of propagation and processing rates. Then, for a transmission medium bandwidth b of 9600bits/sec, message and acknowledgment length l of 1024 bit, $\lambda_d = \lambda_a = 9.375$ (b/l), $\lambda_{d'} = \lambda_{a'} = 74.22$, $\lambda_{ld} = \lambda_{la} = 3.91$, $\lambda_u = \infty$, $\lambda_v = \infty$, $\lambda_y = \infty$, $\lambda_z = \infty$, ϵ of 15% and solving inequality 4.1 for positive λ_T we get the upper bound

$$\lambda_T \leq 1.22707$$

One can then conclude that the upper bound on the timeout rate of the AB protocol with a timeout requirement depends not only on transmission, processing, and propagation rates, but also on the medium rate of loss and the allowed ϵ. These results are independent of our assumption that the number of messages (acknowledgments) in the medium is bounded by 2 since only G_{TC}, which is not affected by the assumption, is considered in the analysis.

4.2. Analysis of Throughput and Transfer Time

In evaluating the AB performance measures, the same data given in the previous section will be used and the timeout rate λ_T will be set equal to 1 which is below the upper bound obtained from solving inequality 4.1. Hence the AB protocol's global behavior G could be approximated to G_{TC} with a probability of about 87.5%.

4.2.1. Maximum Throughput

A protocol *Maximum throughput TH* is defined as the average transmission rate of useful data between protocol entities (i.e., excluding any acknowledgments or retransmissions required by the protocol) achieved when the sender always has a new message to send [Bux 80]. This is a useful performance measure because it represents an upper bound on the amount of traffic the protocol system can carry without the sender's input buffer becoming unstable. Since the AB has always just one message type (i.e. copies of d_0 or d_1) occupying the protocol system at a time, *TH* is given by

$$TH = 1/t_m \tag{4.2}$$

where t_m denotes the mean *virtual transmission time* defined as the average time starting with the sending of a message by the sender until receiving its successful acknowledgment. From the AB protocol specification G_{TC}, this is the duration of the behavior starting from the first τ_d and ending with τ_a, and thus t_m is given by

$$t_m = \mu_{G_{TC}}(\tau_d \bullet E_{TC}) \tag{4.3}$$

It should be noted that E_{TC} ends with $\tau_{a'} \bullet (\tau_u \times \tau_v \times \tau_y \times \tau_z)$; however, since we assumed that rates for τ_u, τ_v, τ_y and τ_z are equal to ∞, then the time between τ_a, and $\tau_u \times \tau_v \times \tau_y \times \tau_z$ is zero and so E_{TC} (as well as G_{TC}) could be taken also as ending with τ_a.. Theorems 3.2 and 3.4 could be then used to evaluate t_m and eq. 4.2 to evaluate TH.

4.2.2. Mean Transfer Time

A protocol *mean transfer time* t_f is defined as the average time from the arrival of a new message at the source until the arrival of its successful acknowledgment at the sender. t_f is then the time duration of the behavior starting with τ_s and ending with τ_a, and so is given by

$$t_f = \mu_{G_{TC}}(G_{TC})$$

which from M1 in theorem 3.2 gives

$$t_f = \mu_{G_{TC}}(\tau_s) + t_m \qquad 4.4$$

Note that if a new message arrives at the source and finds the protocol system occupied, then it has to wait in an input buffer as shown in the queueing model of the AB protocol depicted in Fig. 4.1. Therefore the time duration of τ_s corresponds to the waiting time in the input buffer. Let t_w represent the average waiting time for a message arriving at the input buffer. Then t_f is given by

$$t_f = t_w + t_m \qquad 4.5$$

t_m can be calculated from eq. 4.3 and assuming an exponential arrival rate of new messages λ. t_w can be calculated using the Pollaczek-Khinchine formula [Klei 75] as follows

$$t_w = \lambda[t_m^2 + \sigma_m]/2[1 - \lambda t_m] \qquad 4.6$$

σ_m is the variance of the virtual transmission time and could be computed using theorems 3.3 and 3.4.

4.2.3. Numerical results

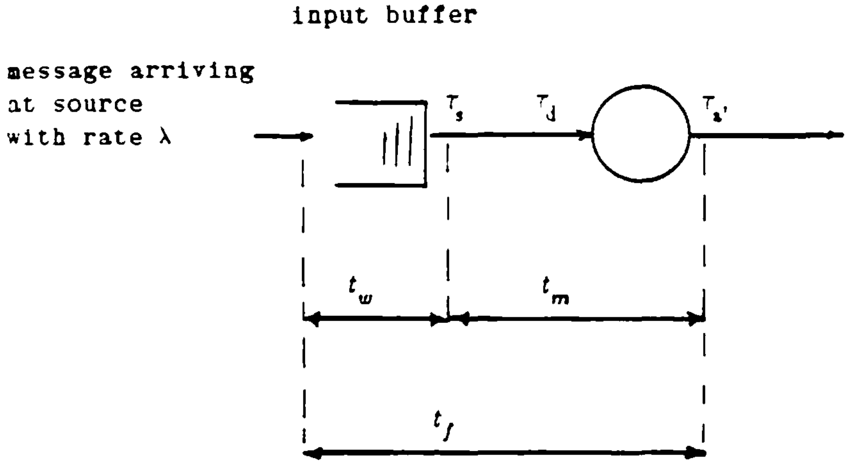
Using the data given in section 4.1 and for message arrival rate of λ=0.1mess/sec., the following results are obtained

TH = 2796.56 bits/sec $\qquad$ t_m = 0.36616 sec/message
t_f = 0.386 sec/message

The value of t_m agrees with that computed by Molloy [Moll 81] (t_m = 0.3662 sec/message) using the same data but obtained from solving a markov process that is extracted from the reachability graph of a stochastic petri net modeling the protocol.

Note that the rate of loss λ_{ld} (λ_{la}) could be calculated from the medium's bit error probability p_{bit} using equation

Figure 4-1: A queueing model for the AB protocol



Prob. of message loss $= P_C(\tau_{ld}) = 1 - (1 - p_b)^l$ 4.7

A plot of *TH* against message length l for several p_{bit} in Fig. 4-2. shows a degradation in *TH* for large values of l. This is due to the increase in probability of medium loss for large l. In addition, as p_{bit} decreases this degradation occurs at very long messages and *TH* saturates for a range of l. A similar result has been obtained by Bux et al [Bux 80] in analyzing the effect of changing the mediums error bit probability on the throughput of a class of HDLC procedures.

A plot of t_f against λ for several p_{bit} is given in Fig. 4-3. The figure shows that as λ approaches $1/t_m$, t_f approaches ∞. Plotting t_f against p_{bit} for several medium bandwidths in Fig. 4.3 shows t_f not affected by change in p_{bit} for small p_{bit}. A similar result has been given in [Tows 79] for stop-and-wait ARQ retransmission protocols which have the same global specification as G_{TC} given in this paper.

In summary, for terrestrial mediums (where p_{bit} is very small e.g., 10^{-10}) the AB throughput only suffer degradation at large message lengths and mean transfer time is also not affected by any slight change in p_{bit}. However, for satellite mediums with higher bit error probability all the protocol parameters should be considered.

5. Conclusions

The main results of this paper are summarized as follows

1. A CCS-variant algebraic specification method that facilitates mapping protocol functional behavior into corresponding timing behavior was introduced.
2. A model for protocol timing behavior has been proposed and rules for evaluating attributes of the model have been provided.

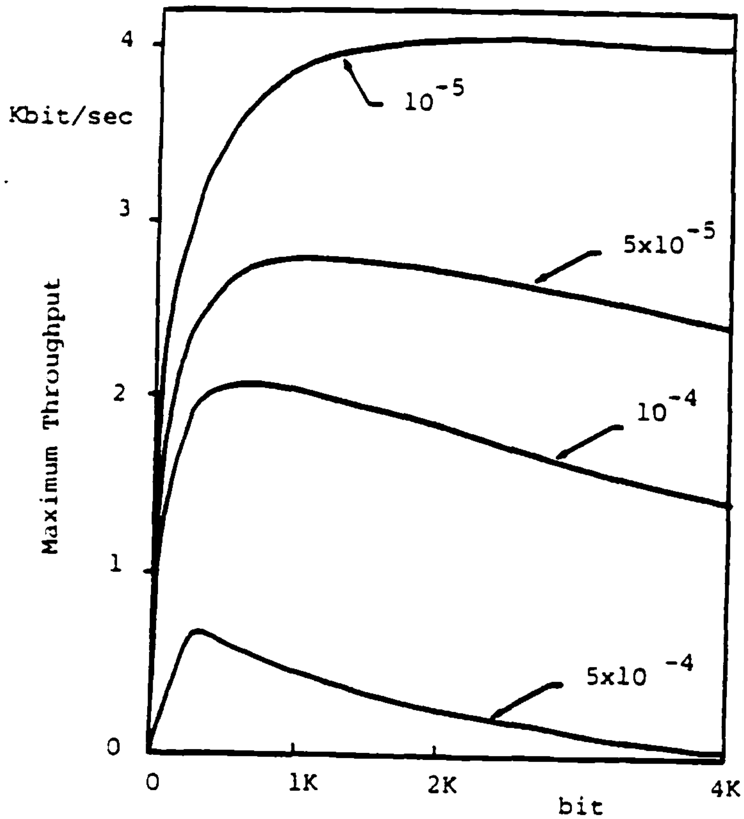


Figure 4-2: Throughput *TH* vs. message length l for various p_{bit}

3. We demonstrated how the model of protocol timing behavior could be used to first specify and verify protocol timing requirements and then to define and analyze performance measures for the AB protocol example. In particular, a timeout constraint has been formulated and used in specifying a timeout requirement on the protocol's timing behavior. In verifying this requirement, an upper bound on the timeout rate was computed. Also the protocol's maximum throughput and mean transfer time performance measures have been analyzed and the results shown to agree with similar results of Molloy, Bux et al and Towsley using other performance analysis approaches.

The approach to performance analysis of protocols described in this paper still needs to be generalized for cases when event times have distributions other than the exponential distribution. Also, much experience is required in applying it to other. more complex and diverse protocols.

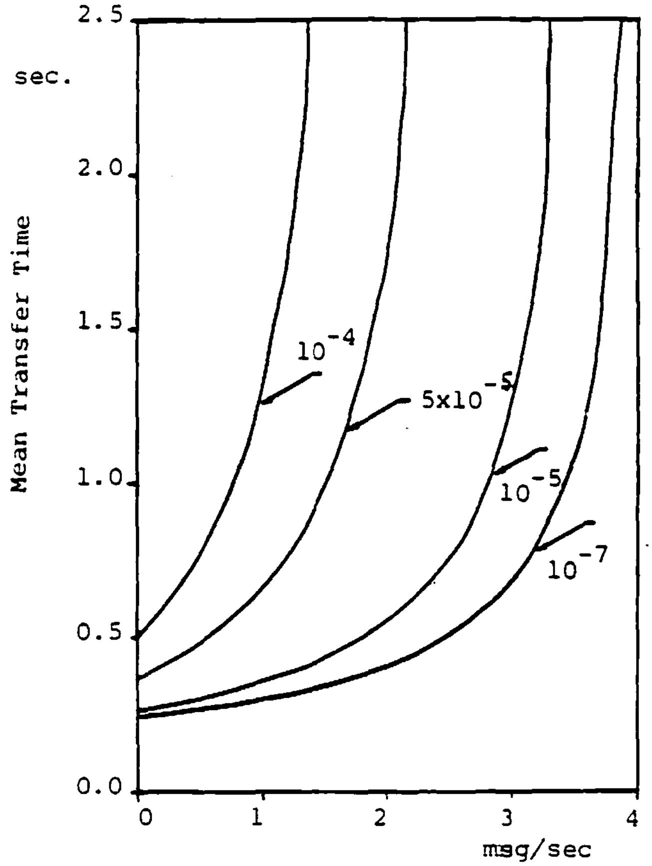


Figure 4-3: Mean transfer time t_f vs. λ for various p_{bit}

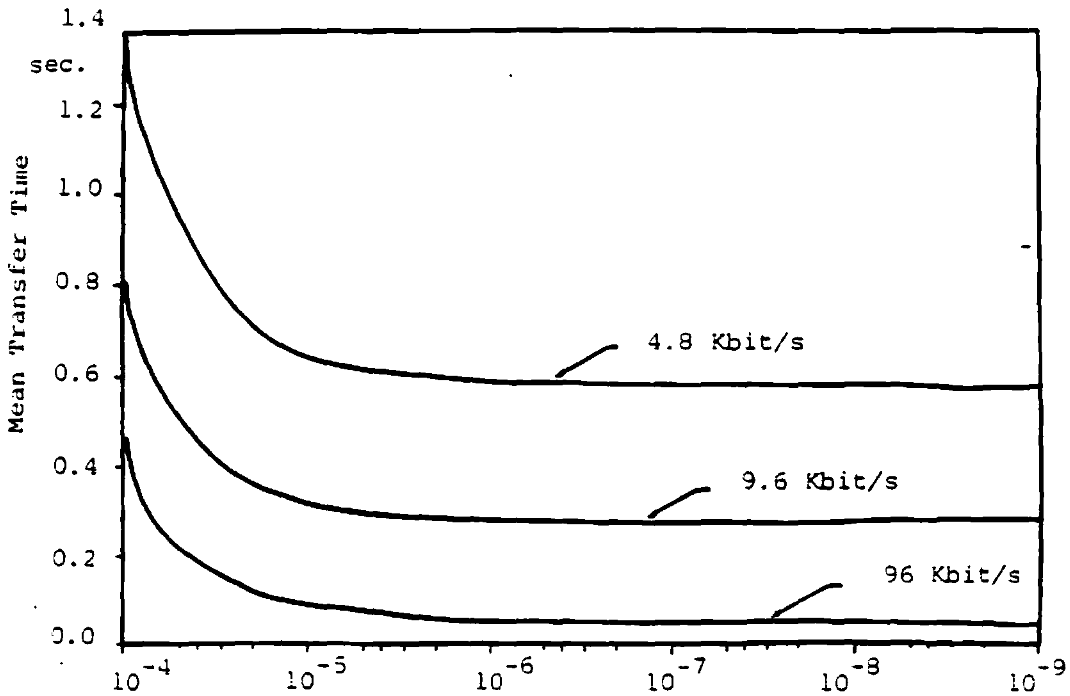


Figure 4-4: Mean transfer time t_f vs. p_{bit} for various bandwidth

I. An Overview of Related Definitions from Universal Algebra

An algebra $\mathcal{A}=<A;\ F>$ consists of an object set A and a finite set F of operations (functions) over A. An algebra $\mathcal{A}$ is the free algebra over a set Υ if A is the minimal set containing Υ and closed under the operations of F. That is, if $f\in F$ is an *n-ary* operation and $a_1,a_2,\ldots,a_n\in A$, then $f(a_1,a_2,\ldots,a_n)\in A$.

Assuming a set of variables X, then $\mathcal{A}[X]$ can be used to denote the free algebra over $A\cup X$. Elements of $\mathcal{A}[X]$ are called polynomials over $\mathcal{A}$ and a pair of polynomials form an equation of the form $\sigma=\tau$ where $\sigma,\tau\in\mathcal{A}[X]$. An equation is said to hold identically in $\mathcal{A}$ and that $\mathcal{A}$ satisfies $\sigma=\tau$ if for every mapping $\Phi: X\longrightarrow A$ we have $\sigma=\tau$, where $\Phi(\sigma)$ is the element of A obtained by substituting all occurrences of variables $x\in X$ with Φx.

II. A Complete Specification of G for the AB Protocol

From the specifications of G_1 and G_2 given in section 2 and if

$Scope(G) = \{a',\ d'\}$ then

$$G = G_1\times G_2 = \tau_s\bullet\tau_d\bullet E$$

$$E = B\times G_2 = (\tau_x + \tau_T)\bullet F + \tau_{ld}\bullet(\tau_x + \tau_T)\bullet\tau_d\bullet E + \tau_{d'}\bullet H$$

$$F = \tau_{ld}\bullet\tau_d\bullet E + \tau_d\bullet[\tau_{d'}\bullet(B\times(\tau_a\bullet C)) + \tau_{ld}\bullet E]$$

$$H = (\tau_x + \tau_T)\bullet(\tau_d\times\tau_a)\bullet J + \tau_a\bullet I$$

$$I = \tau_{a'}\bullet(\tau_u\times\tau_v\times\tau_y\times\tau_z) + (\tau_x+\tau_T)\bullet\tau_d\bullet J + \tau_{la}\bullet(\tau_x + \tau_T)\bullet\tau_d\bullet E$$

$$J = B\times C = \tau_{ld}\bullet\{(\tau_x+\tau_T)\bullet[\tau_d\bullet J+\tau_{la}\bullet\tau_d\bullet N] + \tau_{la}\bullet(\tau_x+\tau_T)\bullet\tau_d\bullet N\} \\ + (\tau_x + \tau_T)\bullet O + \tau_{la}\bullet N + \tau_{d'}\bullet K$$

$$N = \tau_{d'}\bullet L + (\tau_x + \tau_T)\bullet\{\tau_{ld}\bullet\tau_d\bullet E + \tau_d[\tau_{ld}\bullet E+\tau_{d'}\bullet(B\times(\tau_a\bullet C))]\} \\ + \tau_{ld}\bullet(\tau_x+\tau_T)\bullet\tau_d\bullet E$$

$$L = \tau_a\bullet M + (\tau_x + \tau_T)\bullet(\tau_d\times\tau_a)\bullet J$$

$$M = \tau_{a'}\bullet(\tau_u\times\tau_v\times\tau_y\times\tau_z) + (\tau_x + \tau_T)\bullet\tau_d\bullet J \\ + \tau_{la}\bullet(\tau_x+\tau_T)\bullet\tau_d\bullet N$$

$$K = [\overline{a'}\bullet\tau_u\times\tau_v + (\tau_x+\tau_T)\bullet\tau_d\bullet B]\times[(\tau_{la}\bullet\tau_a+\tau_a\bullet(a'+\tau_{la}))\bullet C]$$

$$O = C\times[\tau_{ld}\bullet\tau_d+\tau_d\bullet(\tau_{ld}+d')]\bullet B$$

III. Proof of Minimum Rules (i), (ii) and (iii)

Consider the timing behavior of some process and assume that an event e occurs at some t_i, thus enabling the choice set $CH(A)$ at the next occurrence time t_{i+1}. Then the events belonging to $CH(A) = \{a_i \mid i=1,n\}$ are actually *contending* with each other for this occurrence time. Let $\mathcal{X}[a]$ denote the event time of event a. Also, let F_a and $f_a(t)$ denote the probability distribution and density functions of this random variable. Then the probability that an event a_i belonging to $CH(A)$ occurs is given by

$$\mathbf{P}_A(a_i) = \text{Prob}(\mathcal{X}[a_i] < \mathcal{X}[a_1] \text{ and } .. \ \mathcal{X}[a_i] < \mathcal{X}[a_{i-1}] \text{ and } \mathcal{X}[a_i] < \mathcal{X}[a_{i+1}] \text{ and } .. \ \mathcal{X}[a_i] < \mathcal{X}[a_n])$$

$$= \int_0^\infty \text{Prob}(\mathcal{X}[a_1] \geq t \text{ and } .. \ \mathcal{X}[a_{i-1}] \geq t \text{ and } \mathcal{X}[a_{i+1}] \geq t \text{ and } .. \ \mathcal{X}[a_n] \geq t \mid \mathcal{X}[a_i] = t) \ f_{a_i}(t) \ dt$$

and by independence assumption of the event times, we get P3 (minimum rule(i))

$$\mathbf{P}_A(a_i) = \int_0^\infty \prod_{j=1, j\neq i}^{n} [1 - F_{a_j}(t)] \ f_{a_i}(t) \ dt$$

In order to prove minimum rules (ii) and (iii), note that among the contending events belonging to $CH(A)$, if a_i wins, then it occurs after a time duration of

$$\mathbf{T}_A(a_i) = \min\{\mathcal{X}[a_1], \ldots, \mathcal{X}[a_n]\}$$

which has a distribution given by

$$\text{Prob}(\min\{\mathcal{X}[a_1], \ldots, \mathcal{X}[a_n]\} \leq t) = 1 - \text{Prob}(\min\{\mathcal{X}[a_1], \ldots, \mathcal{X}[a_n]\} > t)$$

$$= 1 - \text{Prob}(\mathcal{X}[a_1] \text{ and } \mathcal{X}[a_2] \text{ and } .. \ \mathcal{X}[a_n] > t)$$

and from the independence assumption

$$\text{Prob}(\min\{\mathcal{X}[a_1], \ldots, \mathcal{X}[a_n]\} \leq t) = 1 - \prod_{j=1}^{n} [1 - F_{a_j}(t)]$$

By computing the average and variance of $\mathbf{T}_a(a_i)$, we get $\mu_a(a_i)$ and $\sigma_a(a_i)$ of minimum rules (ii) and (iii) respectively.

IV. Proof of theorem 3.4

Let $A \quad = \quad \mathrm{Y}{\bullet}A + \mathrm{B}$, then

using M2 in theorem 3.2 and since $\mathbf{P}_A(\mathrm{Y}{\bullet}A) + \mathbf{P}_A(\mathrm{B}) = 1$, then

$$\mu_A(A) \quad = \quad \mathbf{P}_A(\mathrm{Y}{\bullet}A)\mu_A(\mathrm{Y}{\bullet}A) + \mathbf{P}_A(\mathrm{B})\ \mu_A(\mathrm{B})$$

note that from P1 in theorem 3.1 we have

$$\mathbf{P}_A(\mathrm{Y}{\bullet}A) = \quad \mathbf{P}_A(\mathrm{Y})\ \mathbf{P}_A(A) = \mathbf{P}_A(\mathrm{Y}) \tag{IV.1}$$

and from M1 in theorem 3.2 we have

$$\mu_A(\mathrm{Y}{\bullet}A) = \mu_A(\mathrm{Y}) + \mu_A(A) \tag{IV.2}$$

From IV.1 and IV.2 we get

$$\mu_A(A) \quad = \quad \mathbf{P}_A(\mathrm{Y})/[1\text{-}\mathbf{P}_A(\mathrm{Y})]\mu_A(\mathrm{Y}) + \mu_A(\mathrm{B})$$

thus proving the first equation of theorem 3.4. To compute the variance in the second equation: from V2 in theorem 3.3 and IV.1 we get

$$\sigma_A(A) \quad = \quad \mathbf{P}_A(\mathrm{Y})[\sigma_A(\mathrm{Y}{\bullet}A) + \mu_A^2(\mathrm{Y}{\bullet}A)] + \mathbf{P}_A(\mathrm{B})[\sigma_A(\mathrm{B}) + \mu_A^2(\mathrm{B})] - \mu_A^2(A)$$

then by using V1 in theorem 3.3, M1 in theorem 3.2 and the first equation in theorem 3.4 for $\mu_A(A)$, we get

$$\begin{aligned}\sigma_A(A) \quad = \quad & \mathbf{P}_A(\mathrm{Y})\ [\sigma_A(\mathrm{Y}) + \mu_A^2(\mathrm{Y}) + \sigma_A(A)] + [\mathbf{P}_A(\mathrm{Y})]^2\ \mu_A^2(\mathrm{Y})/[1\text{-}\mathbf{P}_A(\mathrm{Y})] \\ & - [1\text{-}\mathbf{P}_A(\mathrm{Y})]\mu_A^2(\mathrm{B})] + \mathbf{P}_A(\mathrm{B})\ [\sigma_A(\mathrm{B}) + \mu_A^2(\mathrm{B})]\end{aligned}$$

which reduces to

$$\sigma_A(A) \quad = \quad \mathbf{P}_A(\mathrm{Y})/[1\text{-}\mathbf{P}_A(\mathrm{Y})]\ [\sigma_A(\mathrm{Y}) + 1/[1\text{-}\mathbf{P}_A(\mathrm{Y})]\mu_A^2(a)] + \sigma_A(\mathrm{B})$$

thus proving the second equation in theorem 3.4.

References

[Bart 69] K.Bartlett, R.Scantlebury, and P.Wilkinson.
A Note on Reliable Full-Duplex Transmission over Half-Duplex Lines.
CACM Vol.12,No.(5):260-261, May, 1969.

[Bux 80] W.Bux,K.Kummerle, and H.Truong.
Balanced HDLC Procedures: A Performance Analysis.
IEEE Transactions on Communications Vol.COM-28,No.(11):1889-1898, November, 1980.

[Grat 68] G.Gratzer.
Universal Algebra.
Springer-Verlag, 1968.

[Klei 75] L.Kleinrock.
Queueing Systems.
Wiley Interscience, 1975.

[Miln 80] R. Milner.
A Calculus of Communicating Systems.
Springer Verlag, 1980.

[Moll 81] M.Molloy.
On the Integration of Delay and Throughput Measures in Distributed Processing Models.
PhD thesis, Univ. of California Los Angeles, 1981.

[Salo 66] A.Salomaa.
Two Complete Axiom Systems for the Algebra of Regular Events.
jacm 13(1):158:169, 1966.

[Shan 82] U.Shankar and S.Lam.
On Time-Dependent Communication Protocols and Their Projections.
Proceedings of the Second IFIP International Workshop on Protocol Specification, Testing and Verification :215-236. May, 1982.

[Snyd 75] D.Snyder.
Random Point Processes.
Wiley-Interscience, 1975.

[Suns 82] C.Sunshine, D.Thompson, R.Erickson, S.Gerhart, and D.Shwabe.
Specification and Verification of Communication Protocols in AFFIRM Using State Transition Models.
IEEE Transactions on Software Engineering Vol.SE-8,No.(5):460-489, September, 1982.

[Tows 79] D.Towsley and J.Wolf.
On the Statistical Analysis of Queue Lengths and Waiting Times for Statistical Multiplexers with ARQ Retransmission Schemes.
IEEE Transactions on Communications Vol.COM-27,No.(4):693- 702, April, 1979.

[Yemi 82] Y.Yemini and J.Kurose.
Towards the Unification of the Functional and Performance Analysis of Protocols, or is the Alternating-Bit Protocol Really Correct?
Proceedings of the Second IFIP International Workshop on Protocol Specification, Testing and Verification , 1982.

[Yemi 83] Y. Yemini and N. Nounou.
CUPID: A Protocol Develeopment Environment.
Proceedings of the Third IFIP International Workshop on Protocol Specification, Testing and Verification , May, 1983.

[Yu 79] L.Yu and J.Majthia.
An Analysis of One Direction of Window Mechanism.
IEEE Transactions on Communications Vol.COM-27,No.(5):778-788, May, 1979.