The Intractability of Bounded Protocols for Non-FIFO Channels

Extended Summary

Yishay Mansour * *Baruch Schieber* †

Abstract

We discuss the efficiency of data link protocols for non-FIFO physical channels. We consider three resources: the number of packets that have to be sent, the number of headers, and the amount of space required by the protocol. We prove three lower bounds. First, we show that the space required by any protocol for delivering n messages using less than n headers can not be bounded by any function of n. Second, we prove that the number of packets that have to be sent by any data link protocol using a fixed number of headers in order to deliver a message is linear in the number of packets that are delayed on the channel at the time the message is sent. Finally, we introduce the notion of a probabilistic physical channel, in which a packet is lost with probability q. We prove an exponential lower bound, with overwhelming probability, on the number of packets that have to be sent by any data link protocol using a fixed number of headers, when it is implemented over a probabilistic physical channel.

*Laboratory for Computer Science, Massachusetts Institute of Technology, 545 Technology Square, Cambridge, MA 02139. Supported by an ISEF fellowship and NSF contract 8657527-CCR. Part of the research of this author was done while visiting IBM - T.J. Watson Research Center.

†IBM Research Division, T.J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598.

1. Introduction

One of the basic tasks of a communication network is to deliver a sequence of messages from a transmitting station to a receiving station. In the hierarchical decomposition of communication networks into abstract layers ([Tan81]), this task has to be accomplished by the data link layer. Although this task is so basic, the protocol that is used to implement the data link layer may be very complex, especially, when the communication channel is unreliable.

In this paper we discuss the tractability, or rather intractability, of bounded data link layer protocols for unreliable channels. The unreliable channels we consider are non-FIFO channels. Such channels may either delete or impose an arbitrary delay on any packet sent over it. We measure the efficiency of a data link protocol using three parameters: (i) The number of packets sent by the protocol. (ii) The amount of additional information (headers) sent by the data link layer protocol with each packet. (iii) The *boundness* of the protocol. We show that the boundness of the protocol is an abstraction of the space required by the protocol.

Our first result is that any k-bounded protocol requires n headers to deliver n messages, even if k is a function of n. This means that the space required by any protocol that uses at most $n-1$ headers to deliver n messages can

not be bounded by any function of n. In contrast, the naive protocol (which delivers the i-th message using the i-th header) uses n headers to deliver n messages in $O(\log n)$ space.

The second lower bound is on the number of packets that have to be sent in order to deliver a message by any protocol with a fixed number k of headers. We show that this number is at least $1/k$ times the number of packets that are delayed on the channel at the time the message is sent. This lower bound shows that the protocol proposed by [Afe88] is optimal in that sense.

For the third lower bound we introduce a model of a probabilistic physical layer with error probability q. In a probabilistic physical layer a packet can be delayed on the channel with probability q. This model is more realistic in the sense that the "average" behavior of the channel is considered and not its "worst case" behavior which may happen only with very low probability. For such channels we prove an exponential lower bound on the number of packets sent. Specifically, in order to deliver n messages at least $(1+q-\epsilon_n)^{\Omega(n)}$ (where $\epsilon_n = O(1/\sqrt{n})$) packets have to be sent with overwhelming probability. The protocols proposed by [AFWZ88,Afe88] match our lower bound, showing it to be tight.

Our conclusion from all these lower bounds is that any data link protocol with a bounded number of headers will suffer from major drawbacks. From a practical stand point it is probably better to pay the penalty of unbounded headers rather than paying in the enormous growth of both the number of packets and the space that any bounded protocol consumes.

The data link layer has received a lot of attention lately [LMF88,AFWZ88,WZ89]. In [LMF88] the first formal specification of the data link layer was given. For a non-FIFO channel, they introduced the notion of k-boundness, and proved that there is no protocol that is k-bounded, for some fixed k, and uses a bounded number of headers. Specifically, they showed that $\Omega(n/k)$ headers are required to deliver n messages. Note that if we allow k to be linear in n then this lower bound becomes trivial. One may suspect that the major reason for the impossibility is the restriction that k is a constant, and enabling k to depend on the number of messages would in fact admit a protocol with $o(n)$ headers. Our first lower bound shows that this is not the case.

In [AFWZ88] a protocol with a bounded number of headers for communicating over a non-FIFO channel is given. There are a few disadvantages that make the protocol of [AFWZ88] not suitable for practical applications. First, the amount of information that has to be recorded is very large. This number is unbounded by any function of the number of messages delivered. Our first lower bound shows that this must be the case. Second, the number of packets that have to be sent in order to deliver a message depends on the number of previous packets exchanged. The number of these packets grows very rapidly, and even in the best case it is exponential in the number of messages delivered. In [Afe88] the dependency was improved to be linear in the number of packets that are delayed on the channel at the time the message is sent. Our second lower bound shows that this the best one can do.

[WZ89] prove some impossibility results in a non-uniform model, in which the transmitter knows all the messages that have to be delivered in advance. They also assume that the set of possible input sequences is restricted to some countable set.

Finally, we remark that all our results can be extended to transport layer protocols (see [Tan81]) over non-FIFO virtual links. Recall that the task of the transport layer is to estab-

lish reliable host to host communication.

The paper is organized as follows. In Section 2 the communication model is described. In Sections 3 and 4 the lower bounds on the boundness and number of packets sent by any protocol for delivering n messages using less than n headers are proved. The analysis of the probabilistic physical layer is given in Section 5.

2. The communication model

In this section we describe the model of the communication network used in the paper. Our model is based on the model defined in [LMF88]. For the sake of brevity we tried to keep the description less formal. A formal specification of the model can be found in [LMF88].

Communication networks are decomposed into layers. (See [Tan81].) Each layer has a particular abstract behavior. This abstract behavior is provided for the use of the next higher layer, and is implemented in terms of the abstract behavior of the next lower layer. In this paper we consider the two lowest layers: the *physical* layer and the *data link* layer.

The physical layer is the lowest layer in the hierarchy, and is implemented directly in terms of the physical transmission media. The transmission media considered in the paper are assumed to be unreliable. Because of this unreliability the physical layer does not ensure that a message that is sent will be received, and also it does not ensure FIFO behavior; that is, messages are received on the physical channel not necessarily in the same order as they are sent. However, we assume that the physical layer ensures that the received messages are not corrupted.

The data link layer is the next higher layer in the network hierarchy. In contrast to the physical layer, the data link layer ensures reliable data transfer, though only across one hop in the network. This means that every message that is sent on a data link to a neighboring node is eventually received at the other end (unless a link failure occurs) and also that the data link exhibits FIFO behavior; that is, messages are received in the same order as they are sent. Data links are implemented using protocols that interact by communicating over physical channels.

In the next subsections we specify the behavior of the physical layer and the data link layer in more detail.

2.1. The physical layer

The physical layer interacts with higher layers at two endpoints, a "transmitting station" and a "receiving station". The physical layer receives messages called "packets" from the higher layer at the transmitting station, and delivers some of the packets to the higher layer at the receiving station. The packets are assumed to be taken from a fixed alphabet P. We denote a physical layer interacting between the transmitting station t to the receiving station r by $PL^{t \to r}$.

The physical layer has two actions: an input action $send\_pkt^{t \to r}(p)$, for $p \in P$, and an output action $receive\_pkt^{t \to r}(p)$, for $p \in P$. The $send\_pkt^{t \to r}(p)$ action represents the sending of a packet p on the physical channel by the transmitting station, and the $receive\_pkt^{t \to r}(p)$ action represents the receipt of a packet p by the receiving station.

We require that the following properties will be satisfied by the physical layer.

The first property is a safety property that says that the physical layer delivers only pack-

ets that were previously sent, and that it can not duplicate messages.

(PL1) There is a correspondence between the $send\_pkt^{t \to r}$ actions and the $receive\_pkt^{t \to r}$ actions which satisfies the following.

1. Each $receive\_pkt^{t \to r}(p)$ action corresponds to a unique preceding $send\_pkt^{t \to r}(p)$ action.
2. Each $send\_pkt^{t \to r}(p)$ action corresponds to *at most* one succeeding $receive\_pkt^{t \to r}(p)$ action.

The second property is a liveness property.

(PL2) Starting at any point, if infinitely many $send\_pkt^{t \to r}$ actions occur after this point, then some $receive\_pkt^{t \to r}$ action occurs after this point.

A physical layer protocol is a protocol that implements the physical layer on a specific transmission medium. This implementation relies heavily on the specific transmission medium used and is not considered in the paper.

In the above definition of a physical layer we took a very conservative approach. We allowed any packet to get lost, or be delivered far in the future. However, a more realistic approach would introduce some probability into the process. It is customary to consider channels as having some fixed probability of error. Thus, for a non-FIFO physical layer only a fraction of the packets are assumed to suffer a large delay.

We specify the properties satisfied by a *probabilistic physical layer* with a fixed error probability $q > 0$. Property (PL1) remains the same as for an ordinary physical layer. The modified Property (PL2) is as follows.

(PL2p) For any $send\_pkt^{t \to r}(p)$ a corresponding $receive\_pkt^{t \to r}(p)$ is generated immediately with probability $1 - q$.

Note that a probabilistic physical layer satisfies condition (PL2) with probability one.

2.2. The data link layer

The data link layer is implemented using the services of the physical layer. Generally, it is implemented in terms of two physical channels, one in each direction. It provides a reliable one-hop message delivery service.

We again assume that there are two endpoints, a "transmitting station" and a "receiving station". The data link layer receives messages from the higher layer at the transmitting station, and delivers them at the receiving station. The messages are assumed to be taken from a fixed alphabet M. We denote a data link layer interacting between the transmitting station t and the receiving station r by $DL^{t \to r}$. It is implemented in terms of the two physical layers $PL^{t \to r}$ and $PL^{r \to t}$. In this paper we consider a data link layer interacting between fixed transmitting and receiving stations, therefore, we omit the superscript $t \to r$ from all the notations that refer to the data link.

The data link layer has two actions, an input action $send\_msg(m)$, for $m \in M$, and output action $receive\_msg(m)$, for $m \in M$. The $send\_msg(m)$ action represents the sending of a message m on the data link by the transmitting station, and the $receive\_msg(m)$ action represents the receipt of a message m by the receiving station.

We require that the following properties will be satisfied by the data link layer. The first property is analogous to safety property already defined for the physical layer.

(DL1) There is a correspondence between the *send_msg* actions and the *receive_msg* actions which satisfies the following.

1. Each *receive_msg*(m) action corresponds to a unique preceding *send_msg*(m) action.
2. Each *send_msg*(m) action corresponds to *at most* one succeeding *receive_msg*(m) action.

The next property is the FIFO property; it guarantees that the messages sent are received in the same order.

(DL2) If *receive_msg*(m) occurs before *receive_msg*(m'), then the *send_msg*(m) action corresponding to *receive_msg*(m) occurs before the *send_msg*(m') action corresponding to *receive_msg*(m').

The last property is the liveness property.

(DL3) For every *send_msg*(m) action there is, eventually, a corresponding *receive_msg*(m) action.

Figure 1 gives a schematic description of the data link layer.

2.3. The data link layer protocol

A *data link layer protocol* is used to implement the data link layer using the services provided by the physical layer. As in [LMF88] the data link layer protocol is modeled by two I/O automata (see [LT87]), one at the transmitting station and one at the receiving station. Note that this can be done without loss of generality, since an I/O automaton has unlimited computation power. We denote the automaton at the transmitting station by A^t and the automaton at the receiving station by A^r. (See Figure 1.)

The input actions of A^t are: *send_msg*(m), for $m \in M$, *receive_pkt*$^{r\to t}(p)$, for $p \in P$, its output action is *send_pkt*$^{t\to r}(p)$, for $p \in P$. Similarly, the input actions of A^r are: *receive_pkt*$^{t\to r}(p)$, for $p \in P$, its output actions are *send_pkt*$^{r\to t}(p)$, for $p \in P$ and *receive_msg*(m), for $m \in M$.

We define some of the terminology used for data link protocols.

Definition 1: *An* execution *of a data link protocol is a sequence α of possible data link layer protocol actions; that is, α is consistent with the local transition functions of the I/O automata A^t, A^r, $PL^{t\to r}$ and $PL^{r\to t}$.*

Definition 2: *For an excution α, $sm(\alpha)$ and $rm(\alpha)$ are the numbers of send_msg and receive_msg actions in α, $sp^{t\to r}(\alpha)$, $rp^{t\to r}(\alpha)$, $sp^{r\to t}(\alpha)$ and $rp^{r\to t}(\alpha)$ are the numbers of send_pkt$^{t\to r}$ actions, receive_pkt$^{t\to r}$ actions, send_pkt$^{r\to t}$ actions and receive_pkt$^{r\to t}$ actions in α, respectively.*

Definition 3: *An execution α is* valid *if it satisfies (DL1) to (DL3).*

Definition 4: *An execution α is* semi-valid *if there exists α_1 and α_2 such that (1) $\alpha = \alpha_1\alpha_2$, (2) α_1 is valid, and (3) $sm(\alpha_2) = 1$.*

To measure the efficiency of a data link protocol we use three parameters: (i) The number of packets sent by the protocol. (ii) The amount of additional information sent by the data link layer protocol with each packet. This amount is referred to as the *header* size. (iii) The *boundness* of the protocol. The relevance of the first parameter is obvious. Below, we define the latter two parameters and give some motivation to why they measure the efficiency of a protocol.

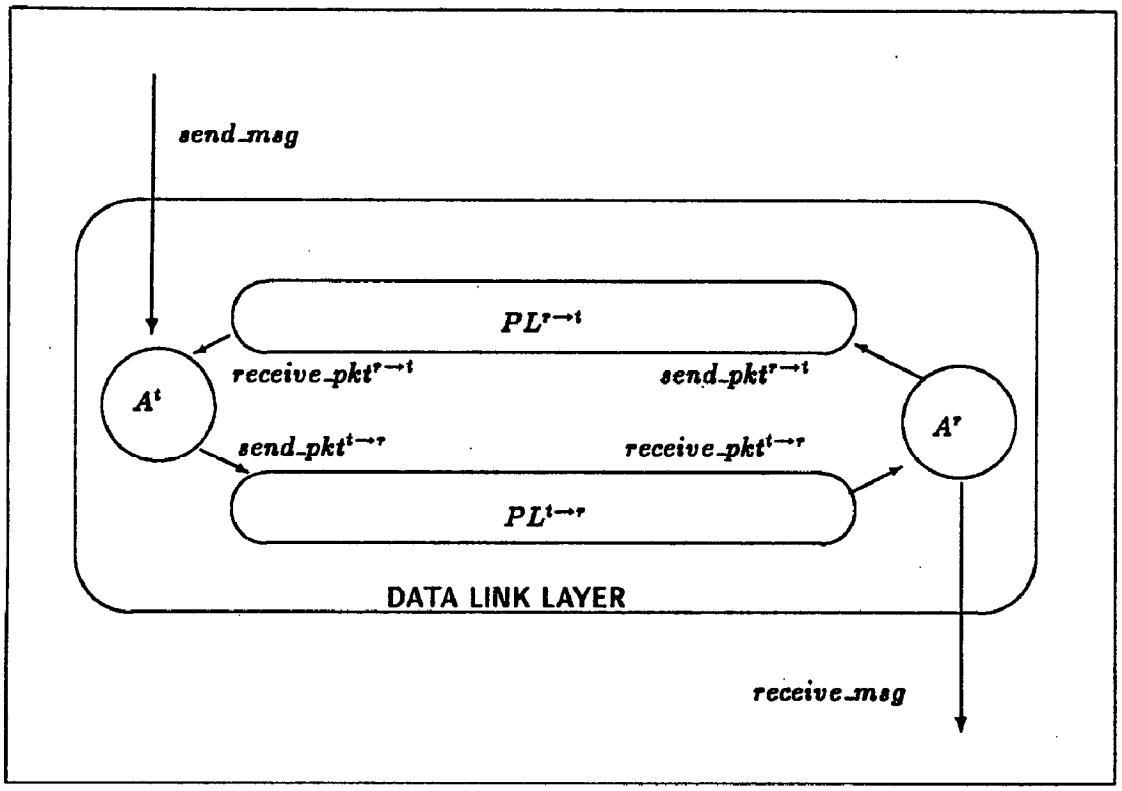

Figure 1: The data link layer

Headers Data link protocols often have to distinguish between packets sent over the channel. Since the protocol must work also in the case where all the messages that are sent are the same, it can not use the content of the messages in order to make this distinction. Thus, some additional information has to be appended by the protocol. A good example for this phenomenon is the "alternating bit" protocol (see [BSW69]). In this protocol two kinds of packets has to be distinguished, this is done by appending either "0" or "1" to the original messages. To measure the amount of additional information [LMF88] suggested an abstract notion based on an equivalence relation between packets. In this paper we choose a more simplistic approach. We assume that all messages sent are the same. In this case the number of headers becomes $|P|$. Notice that this number reflects the length of the packets, which must be at least logarithmic in the size of $|P|$. We refer to this parameter as the number of *headers*. This is done, mainly, in order to be consistent with previous terminology.

We consider the growth of the number of headers as a function of the number of messages. For a function h and a data link protocol, we say that the number of headers used by the protocol grows proportional to h, if for any $n > 0$, in any valid execution α of the protocol with $sm(\alpha) = n$ at most $h(n)$ distinct packets are sent.

Boundness Informally, the boundness of a protocol is an upper bound on the number of packets that have to be sent, from *any point* when the physical layer starts behaving in the optimal way, until the current message is received. As we show below there is a relation

between the boundness of the protocol and its space complexity. Before stating this relation we give a formal definition of boundness.

We first define *constant* boundness. For a constant k, a protocol is k-bounded if for every semi-valid execution α, there is an extension β of the execution, such that: (i) $\alpha\beta$ is a valid execution, (ii) β does not include any $receive\_pkt^{t\to r}(p)$ actions whose corresponding $send\_pkt^{t\to r}(p)$ occurred in α, and (iii) $sp^{t\to r}(\beta) \leq k$.

The relation between the efficiency of a protocol and its boundness is given in the following theorem.

Theorem 2.1: *Any data link protocol $A = (A^t, A^r)$ is $k_t k_r$-bounded, where k_t and k_r are the number of states of the automata A^t and A^r, respectively.*

Proof: Let α be a semi-valid execution. We prove that there is an extension β, such that: (i) $\alpha\beta$ is a valid execution, (ii) β does not include any $receive\_pkt^{t\to r}(p)$ actions whose corresponding $send\_pkt^{t\to r}(p)$ occurred in α, and (iii) $sp^{t\to r}(\beta) \leq k_t k_r$. The proof is by contradiction. To obtain a contradiction we show that if such an extension does not exist then there is an infinite extension that does not include any *receive_msg* action. This violates (DL3), contradicting the correctness of the protocol A.

To define the infinite extension, we consider the extension γ resulting from the behaviour of the physical layer that satisfies the following two properties: (1) No packet that has been sent while executing α is delivered while executing γ. This implies that the extension γ does not include any $receive\_pkt^{t\to r}(p)$ (resp. $receive\_pkt^{r\to t}(p)$) actions that correspond to $send\_pkt^{t\to r}(p)$ (resp. $send\_pkt^{r\to t}(p)$) actions that occurred in α. (2) A packet that is sent while executing γ is delivered immediately. This implies that each $send\_pkt^{t\to r}(p)$ (resp. $send\_pkt^{r\to t}(p)$) action is followed by the corresponding $receive\_pkt^{t\to r}(p)$ (resp. $receive\_pkt^{r\to t}(p)$) action. Let γ be the shortest such extension where $sp^{t\to r}(\gamma) > k_t k_r$. By our hypothesis, $sm(\gamma) = 0$, or otherwise a valid extension β exists.

Let (q_t^0, q_r^0) be the states of A^t and A^r at the end of α and let (q_t^i, q_r^i) be the states of A^t and A^r after the i-th $receive\_pkt^{t\to r}(p)$ action. Notice that there must be some $0 \leq i < j \leq k_t k_r$ such that $(q_t^i, q_r^i) = (q_t^j, q_r^j)$. Let $\gamma = \gamma_1\gamma_2\gamma_3$, where γ_1 is the prefix of γ until (and including) the i-th $receive\_pkt^{t\to r}(p)$ action, γ_2 is the part of γ consisting of the actions after the i-th $receive\_pkt^{t\to r}(p)$ action, until (and including) the j-th $receive\_pkt^{t\to r}(p)$ action, and γ_3 is the suffix of γ consisting of the actions after the j-th $receive\_pkt^{t\to r}(p)$ action.

It can be shown by a simple induction that the extensions γ_1 and $\gamma_1 \overbrace{\gamma_2 \ldots \gamma_2}^{l \text{ times}}$, for any $l > 0$, are indistinguishable by both A^t and A^r. This is, since $(q_t^i, q_r^i) = (q_t^j, q_r^j)$ and all the packets delayed on the channel at the end of $\alpha\gamma_1$ are the same as the packets delayed at the end of $\alpha\gamma_1\gamma_2$. We conclude that since $\alpha\gamma$ is a possible execution, the infinite execution $\alpha\gamma_1\gamma_2\gamma_2\ldots$ is also possible. However, this infinite execution does not consist any $receive\_msg(m)$ action; a contradiction. □

The above theorem shows that the boundness condition can be viewed as an abstraction of the space complexity of the protocol. By proving a lower bound on the boundness we prove a lower bound on the space required by the protocol.

In [LMF88] it was shown that any k-bounded data link protocol (for any constant k) constructed to use a non-FIFO physical layer for delivering n messages requires n/k headers. In

this paper we consider the case when the protocol is not constant bounded, but bounded by a function of the execution. It seems reasonable to make the boundness of a protocol be a function of the specific execution. For example, it would be reasonable to ask that the length of the extension of a semi-valid execution α will be shorter in the case α reflects an optimal behavior of the physical layer rather than in the case α reflects a "noisy" behavior of the physical layer. Also, it would be reasonable to ask that the length of the extension will be proportional to the number of messages delivered so far.

In view of this we consider two possible functions for the boundness. The first is a function of the number of messages delivered so far, and the second is a function of the number of packets that were sent from t but were not received by r. Let f be a function.

Definition 5: *A protocol is M_f-bounded if for every semi-valid execution α, there is an extension β of the execution, such that: (i) $\alpha\beta$ is a valid execution, (ii) β does not include any receive_pkt$^{t\to r}(p)$ actions whose corresponding send_pkt$^{t\to r}(p)$ occurred in α, and (iii) $sp^{t\to r}(\beta) \leq f(sm(\alpha))$.*

Definition 6: *A protocol is P_f-bounded if for every semi-valid execution α, there is an extension β of the execution, such that: (i) $\alpha\beta$ is a valid execution, (ii) β does not include any receive_pkt$^{t\to r}(p)$ actions whose corresponding send_pkt$^{t\to r}(p)$ occurred in α, and (iii) $sp^{t\to r}(\beta) \leq f(sp^{t\to r}(\alpha) - rp^{t\to r}(\alpha))$.*

We remark that the definition of boundness used here differs from the one used in [LMF88,AFWZ88]. In [LMF88,AFWZ88] the boundness is defined only with respect to extensions of semi-valid executions that end with a *send_msg*(m) action, while in our definition the boundness is defined with respect to extensions of any semi-valid execution. Our definition is similar to the one used in [WZ89].

3. A lower bound for M_f-bounded protocols

In this section we prove:

Theorem 3.1: *Let f be any function. Any M_f-bounded data link protocol for sending n messages requires n headers.*

This theorem extends the result of [LMF88] that considered only constant bounded data link protocols.

Proof: First, we show that it is sufficient to consider only functions which are monotonically non-decreasing. Suppose that f is not monotone. Define a new function $\hat{f}$ as follows. For all $x \geq 0$, $\hat{f}(x) = \sup_{y \leq x}\{f(y)\}$. Clearly, $\hat{f}$ is monotone and the protocol is $M_{\hat{f}}$ bounded. Thus, from now on we consider only monotone functions. We also assume w.l.o.g. that $f(1) \geq 2$.

The proof is by contradiction. Suppose that $A = (A^t, A^r)$ is an M_f-bounded protocol that uses less than n headers to deliver n messages. Let $P = \{p_1, \ldots, p_k\}$, for some $k < n$, be the set of packets used by A.

To obtain a contradiction we show that there exists an execution α' of the protocol that does not satisfy (DL1). Specifically, the execution α' will be an excution where $rm(\alpha') = sm(\alpha') + 1$. Such an execution does not satisfy (DL1), and can not be a prefix of any execution that does satisfy (DL1). We refer to such an execution as an invalid execution.

To get the execution α' we assume an adverse behavior of the physical layer. Recall that the physical layer may delay the packets

arbitrarily. Thus, at each point in time there is a set of packets which are *in transition* from the transmitting station to the receiving station. That is, these packets were sent by A^t but are still delayed on the channel. Consider a valid execution α_i of the protocol A, with $sm(\alpha_i) = i$, that ends with a *receive_msg*(m) action. Suppose that the semi-valid execution α_i*send_msg*(m) can be extended by β to a valid execution α_i*send_msg*$(m)\beta$ such that (i) β does not include any *send_msg* actions, and (ii) β does not include any *receive_pkt*$^{t\to r}(p)$ actions whose corresponding *send_pkt*$^{t\to r}(p)$ occurred in α_i. Observe that if for each *send_pkt*$^{t\to r}(p)$ action in β there is a copy of the packet p in transition at the end of α_i, then the extension β can be "simulated" by the physical layer, simply by replacing each packet which is sent by A^t in β by the respective packet in transition. Let β' be the resulting extension. Clearly, A^r can not distinguish between β and β'. Thus its actions in both executions are the same, implying that $rm(\beta') = 1$. Now consider the execution $\alpha' = \alpha_i\beta'$. This is a possible execution of the protocol. However, since $sm(\alpha') = i$ and $rm(\alpha') = i + 1$, α' is invalid, i.e., it does not satisfy (DL1).

To complete the proof we show that there exists an execution α_i*send_msg*(m), for some $i \leq k < n$, that can be extended to an invalid execution α'. We construct the execution α_i inductively, starting from the empty execution α_0. For the construction we define subsets $P_i \subseteq P$ of packets. The subset P_i contains i packets from P. The inductive claim is as follows.

Claim: For all $0 \leq i < k < n$, either the execution α_i can be extended to an invalid execution α', or there exists an extension of α_i*send_msg*(m) to a valid execution α_{i+1} and a subset P_{i+1} such that for each packet $p \in P_{i+1}$, $(k-i-1)!f(k+1)^{k-i}$ copies of p are in transition at the end of α_{i+1}.

Before proving the Claim we show how it proves Theorem 3.1. Observe that either we can extend α_i, for some $0 \leq i < k$, to an invalid execution, or $f(k+1)$ copies of p are in transition at the end of α_k, for each packet $p \in P_k = P$. By of the boundness of the protocol the semi-valid execution α_k*send_msg*(m) can be extended by an extension β to a valid execution α_k*send_msg*$(m)\beta$, such that (i) β does not include any *send_msg* actions, (ii) β does not include any *receive_pkt*$^{t\to r}(p)$ actions whose corresponding *send_pkt*$^{t\to r}(p)$ occurred in α_k, and (iii) $sp^{t\to r}(\beta) \leq f(k+1)$. Since $f(k+1)$ copies of each packet p are in transition, for each *send_pkt*$^{t\to r}(p)$ action in β there is a copy of the packet p in transition. Hence, the extension β can be "simulated" by the physical layer, to obtain an invalid execution $\alpha' = \alpha_k\beta'$. Theorem 3.1 follows.

We return to the proof of the Claim. The proof is by induction.

BASIS The basis is simple. Starting from the execution α_0*send_msg*(m) = *send_msg*(m), the first $k!f(k+1)^k - k + 1$ packets sent from the transmitting station are delayed on the channel. The execution α_1 is the valid execution resulting from the extension of α_0*send_msg*(m). Clearly, at the end of α_1, for at least one packet p, $(k-1)!f(k+1)^k$ copies of p are in transition. We let $P_1 = \{p\}$.

INDUCTION Suppose that, for some $1 \leq i < k$, α_i can not be extended to an invalid execution. By the hypothesis, there is a subset P_i such that at the end of α_i, $(k-i)!f(k+1)^{k+1-i}$ copies of each $p \in P_i$ are in transition. By of the boundness of the protocol the semi-valid execution α_i*send_msg*(m) can be extended by an extension β_1 to a valid execution, such that (i) β_1 does not include any *send_msg* actions, (ii) β_1 does not include any *receive_pkt*$^{t\to r}(p)$ actions whose corresponding *send_pkt*$^{t\to r}(p)$ occurred in α_i, and (iii) $sp^{t\to r}(\beta_1) \leq f(i+1) \leq f(k+1)$.

We make the channel delay all the packets in β_1 which are not from the set P_i. We claim that at least one of the packets in β_1 is not from the set P_i. To see this, observe that if all the packets in β_1 are from the set P_i then α_i can be extended to an invalid execution. (This is, since there are at least $f(k+1)$ copies from each packet in P_i.) Let $\hat{\beta}_1$ be the prefix of β_1 up to the first $receive\_pkt^{t\to r}(p)$, such that $p \notin P_i$. (This implies that p is in transit in $\alpha_i\hat{\beta}_1$.)

Now consider the semi-valid execution $\alpha_i send\_msg(m)\hat{\beta}_1$. Again, by the boundness of the protocol $\alpha_i send\_msg(m)\hat{\beta}_1$ can be extended to a valid execution by an extension β_2, such that (i) β_2 does not include any $send\_msg$ actions, (ii) β_2 does not include any $receive\_pkt^{t\to r}(p)$ actions whose corresponding $send\_pkt^{t\to r}(p)$ occurred in $\alpha_i send\_msg(m)\hat{\beta}_1$, and (iii) $sp^{t\to r}(\beta_2) \leq f(i+1) \leq f(k+1)$. We make the channel delay all the packets in β_2 which are not from the set P_i. We claim that at least one of the packets in β_2 is not from the set P_i. Otherwise, since at least $2f(k+1) \leq (k-i)!f(k+1)^{k+1-i}$ copies of each $p \in P_i$ are in transition the channel would be able to "simulate" the extension $\hat{\beta}_1\beta_2$, and hence to obtain an extension of α_i to an invalid execution, contradicting our assumption. Let $\hat{\beta}_2$ be the prefix of β_2 up to the first $receive\_pkt^{t\to r}(p)$, such that $p \notin P_i$.

Let $t = (k-i)!f(k+1)^{k-i}$. Since $(k-i)!f(k+1)^{k+1-i} = t\,f(k+1)$ copies of each $p \in P_i$ are in transition, the same procedure can be repeated t times, to produce the extension $\hat{\beta}_1\hat{\beta}_2\cdots\hat{\beta}_t$, causing the delay of $(k-i)!f(k+1)^{k-i}$ copies of packets which are not in P_i. Clearly, for at least one packet p of the $k-i$ packets not in P_i, $(k-i-1)!f(k+1)^{k-i}$ copies of it will be in transition. The new subset P_{i+1} will be $P_i \cup \{p\}$. □

4. A lower bound for P_f-bounded protocols

Suppose that we are given a data link protocol for delivering n messages using $k < n$ headers. In this section we prove a lower bound on the boundness of the protocol as a function of the number of packets that are in transit.

Theorem 4.1: *Any protocol for delivering n messages using $k < n$ headers can not be P_f-bounded for any monotonically increasing function f such that $f(l) \leq \lfloor l/k \rfloor$, for some $l < n$.*

Proof: The proof is by contradiction. Let f be some monotonically increasing function such that $f(l) \leq \lfloor l/k \rfloor$, for some integers $k < n$ and $l < n$. Suppose that $A = (A^t, A^r)$ is a P_f-bounded protocol that uses k headers to deliver n messages. Let $P = \{p_1, \ldots, p_k\}$ be the set of packets used by A.

To obtain a contradiction we show that there exists an invalid execution α' of the protocol that does not satisfy (DL1). Specifically, the execution α' will be an excution where $rm(\alpha') = sm(\alpha') + 1$.

We construct the execution α' using a method similar to the one used in the proof of Theorem 3.1. Let α_i be a valid execution of the protocol A, with $sp^{t\to r}(\alpha_i) - rp^{t\to r}(\alpha_i) = i$. We show that there exists an execution α_i, for some $i \leq l < n$, that can be extended to an invalid execution α'. The execution α_i is constructed inductively, starting from the empty execution α_0.

For $1 \leq i \leq l$ and $1 \leq j \leq k$, let $m_{i,j}$ be the number of copies of p_j that are in transition at the end of α_i. The inductive claim is as follows.

Claim: For all $0 \leq i < l < n$, either the execution α_i can be extended to an invalid execution α', or there exists an extension of

$\alpha_i send\_msg(m)$ to a valid execution α_{i+1} such that (i) for each $1 \leq j \leq k$, $m_{i+1,j} \leq \lfloor l/k \rfloor$; (ii) for some $1 \leq j \leq k$, $m_{i+1,j} = m_{i,j} + 1$.

Before proving the Claim we show how it proves Theorem 4.1. Let $\hat{l} = k \lfloor l/k \rfloor \leq l$. Observe that either we can extend α_i, for some $0 \leq i < \hat{l}$, to an invalid execution, or $\lfloor l/k \rfloor$ copies of each packet are in transition at the end of $\alpha_{\hat{l}}$. Recall that $sp^{t \to r}(\alpha_{\hat{l}}) - rp^{t \to r}(\alpha_{\hat{l}}) = \hat{l}$. By the boundness of the protocol the semi-valid execution $\alpha_{\hat{l}} send\_msg(m)$ can be extended by an extension β to a valid execution $\alpha_{\hat{l}} send\_msg(m)\beta$, such that (i) β does not include any $send\_msg$ actions, (ii) β does not include any $receive\_pkt^{t \to r}(p)$ actions whose corresponding $send\_pkt^{t \to r}(p)$ occurred in $\alpha_{\hat{l}}$, and (iii) $sp^{t \to r}(\beta) \leq f(\hat{l}) \leq f(l) \leq \lfloor l/k \rfloor$. Since $\lfloor l/k \rfloor$ copies of each packet p are in transition, for each $send\_pkt^{t \to r}(p)$ action in β there is a copy of the packet p in transition. Hence, the extension β can be "simulated" by the physical layer, to obtain an invalid execution $\alpha' = \alpha_{\hat{l}}\beta'$. Theorem 4.1 follows.

We return to the proof of the Claim. The proof is by induction.

BASIS Starting from the execution $\alpha_0 send\_msg(m) = send\_msg(m)$, the first packet sent from the transmitting station is delayed on the channel. W.l.o.g. assume that this packet is p_1. The execution α_1 is the valid execution resulting from the extension of $\alpha_0 send\_msg(m)$. We have that $m_{1,1} = 1$.

INDUCTION Suppose that, for some $1 \leq i < l$, α_i can not be extended to an invalid execution. By of the boundness of the protocol the semi-valid execution $\alpha_i send\_msg(m)$ can be extended by an extension β to a valid execution satisfying properties (i)-(iii) above. We claim that for at least one packet p_j, such that $m_{i,j} < \lfloor l/k \rfloor$, the number of $send\_pkt^{t \to r}(p_j)$ actions in β is more than $m_{i,j}$. To see this, observe that otherwise the physical layer would be able to "simulate" β by an extension β', which would be indistinguishable by A^r from β. This would imply that the execution $\alpha_i\beta'$, which violates (DL3), is a possible execution of A.

We make the channel delay the first packet p_j sent in β. The execution α_{i+1} is the valid execution resulting from the extension of $\alpha_i send\_msg(m)\hat{\beta}$, where $\hat{\beta}$ is the prefix of β up to the first $receive\_pkt^{t \to r}(p_j)$ action. Clearly, $m_{i+1,j} = m_{i,j} + 1 \leq \lfloor l/k \rfloor$. □

The theorem implies that for any protocol with a fixed number of headers, there exists a semi-valid execution α such that the length of the shortest extension of α to a valid execution is linear in $1/k$ times the number of packets in transit at the end of α. This bound is tight up to a constant factor. [Afe88] describes a data link protocol with three headers which is P_f bounded, for some linear function f.

5. A probabilistic physical layer

Suppose that we are given a data link protocol with a fixed number, k, of headers that is implemented using a probabilistic physical layer with error probability q. From the previous lower bounds on the boundness of a protocol one may deduce lower bounds on the number of packets that have to be sent in order to deliver n messages. These lower bounds are assuming an adverse behavior of the physical layer. That is, they are assuming that the packets which will be delayed can be determined so that they will make the protocol behave in the worst scenario. Clearly, a probabilistic physical layer can have the same adverse behavior. Thus, the lower bounds hold also for probabilistic physical layers. However, the probability of such a behavior may be very small. Hence, one may think that although in the worst case there is

a lower bound on the number of packets required to deliver n messages, in most of the cases the number of packets required will be much smaller. In this section we prove that this is not the case. Specifically, we prove that with high probability the number of packets required to deliver n messages is *exponential* in the number of packets sent. This shows that even the "average" behavior of a data link protocol with bounded headers is intractable.

We remark that both protocols suggested in [AFWZ88,Afe88] achieve this "average" behavior. Thus showing that our lower bound is tight.

Theorem 5.1: *Any data link protocol with a fixed number, k, of headers that is implemented using a probabilistic physical layer with error probability q has to send, with probability $1 - e^{-\Omega(n)}$, at least $(1 + q - \epsilon_n)^{\Omega(n)}$ packets in order to deliver n messages, where $\epsilon_n = O(1/\sqrt{n})$.*

Proof: Let A be a data link protocol with k headers that is implemented using a probabilistic physical layer with error probability q. Let $P = \{p_1, \ldots, p_k\}$ be the set of distinct packets used by A. Consider a valid execution α of the protocol A, that includes the exchange of n messages in the following form, $\alpha = send\_msg(m)\beta_1 receive\_msg(m) send\_msg(m)$ $\beta_2 \cdots \beta_n receive\_msg(m)$ with $sm(\beta_i) = rm(\beta_i) = 0$. W.l.o.g. assume that n is a product of k. Define α_i to be the prefix of α up to and not including the i-th $send\_msg(m)$ action. For $1 \leq i \leq n$ and $1 \leq j \leq k$, let $m_{i,j}$ be the number of copies of p_j that are in transition at the end of α_i.

As in the proofs of theorems 3.1 and 4.1 we claim that for any β_i, for at least one packet p_j, the number of $send\_pkt^{t \to r}(p_j)$ actions in β_i is more than $m_{i,j}$. We call such a packet a *dominant* packet in β_i. To see this, obsreve that otherwise the physical layer would be able to "simulate" β_i by an extension β_i', which would be indistinguishable by A^r from β_i. This would imply that the execution $\alpha_i \beta_i' receive\_msg(m)$, which violates (DL3), is a possible execution of A.

Observe that at least one packet, say p_j, is dominant in at least n/k of the β_i's. Intuitively, we would like to argue that the number of copies of p_j sent by the protocol is $\Omega((1+q)^{n/k})$. To see this, consider an extension β_i in which p_j is dominant. The number of copies of p_j sent in β_i is at least $m_{i,j}$, implying that the expected number of copies of p_j that will be delayed in β_i is $qm_{i,j}$. Thus, the expected value of $m_{i+1,j}$ is $(1+q)m_{i+j}$. Intuitively, this will give the expected exponential behavior. However, there are three problems with this intuition.

1. We assume that the events of delaying a packet are independent. This may not be the case. In other words, it may be that a packet p_j is sent $m_{i,j}$ times by the protocol and still the expected number of delayed copies of p_j would be much less than $qm_{i,j}$. This may happen, for example, if the protocol would send a copy of p_j only it knows that the previous copy of p_j has been received.
2. Notice that we claim that the expected value of a *product* of random variables is the product of their expected values. This is true if the random variables are independent, but in our case it is not clear that they are independent, since an extension may depend on previous extensions.
3. When considering a product of random variables there is a difference between the expected value and the observed value; that is, the value we get with high probability. Unlike the case of adding independent random variables, it is not al-

ways true that the observed value converges with high probability to the expected value. There are examples where the expected value of a product of random variables diverges, but with high probability the value of this product converges to zero.

To prove the theorem we have to refine our arguments. For $1 \leq i \leq n$, define B_i to be the random variable whose value is the extension β_i, given the semi-valid execution $\alpha_i send\_msg(m)$. Observe that B_i is a random variable since it depends on the behavior of the probabilistic physical layer. Also, the value of B_i is not defined apriori but only after the execution α_i is defined. Consider all the possible values of B_i, each such value (extension) has a certain probability. For each such extension we have at least one packet p_j which is dominant in that extension. Thus, by summing the probabilities of all the extensions in which a packet p_j is dominant, we can get the probability of the packet p_j to be dominant in the i-th extension. Since the sum of these probabilities for all packets is at least one (it may be more than one if there is more than one dominant packet in the same extension), there is at least one packet p_j whose probability to be dominant in the i-th extension is $\geq 1/k$. We call such a packet a *probable dominant* for β_i. Clearly, at least one packet p_j is probable dominant in at least n/k of the extensions. In the rest of the proof we consider only this packet and show that $m_{i,j}$ grows exponentially. (We remark that we do not use the *value* of j, which is not known apriori, but only the fact that such a j exists, which is known apriori.)

For the proof we need the following two lemmas. Let l be the index of the $(n/2k+1)$-th extension in which p_j is the probable dominant packet.

Lemma 5.2: *With probability* $1 - e^{-\Omega(n)}$, $m_{l,j} \geq nq/4k^2$.

Lemma 5.3: *If* $m_{l,j} \geq nq/4k^2$ *then with probability* $1 - e^{-\Omega(n)}$, $m_{n,j} \geq (1+q-\epsilon_n)^{\Omega(n)}$.

The theorem readily follows from these two lemmas. □

In the proofs of both lemmas we use a bound on the tail of the binomial distribution, known as Hoeffding bound. Let $\{X_i\}_{i=1}^n$ be independent $(0,1)$ random variables with probability q of being one.

Theorem 5.4 ([Hoe63]): *For* $\alpha < q$

$$Prob\left\{\sum_{i=1}^{n} X_i \leq \alpha n\right\} \leq e^{-2n(\alpha-q)^2}.$$

Proof of Lemma 5.2: For each of the first $n/2k$ extensions in which p_j is probable dominant, consider the random variable whose value is one if at least one copy of p_j is delayed in the extension. To get the probability that this random variable is one, note that if a packet is dominant in an extension then with probability q at least one of its copies sent in that extension is delayed. Thus, the probability that at least one copy of p_j will be delayed in an extension in which p_j is probable dominant is q/k. Since all the $n/2k$ random variables corresponding to the extensions in which p_j is probable dominant are independent, we may use Hoeffding bound to get a bound on the probability that their sum is $< nq/4k^2$ and get that this probability is $< e^{-nq^2/4k^2} = e^{-\Omega(n)}$. □

Proof of Lemma 5.3: Let i be the index of one of the last $n/2k$ extensions in which p_j is probable dominant. Consider the random variable X whose value is one if $m_{i+1,j} \geq (1+q-\epsilon_n)m_{i,j}$, given that $m_{i,j} \geq nq/4k^2$. We claim that for $\epsilon_n = O(1/\sqrt{n})$ the probability of this random variable to be one $\geq 1/2k$.

We bound the probability of the intersection of two events: $X = 1$ and p_j is dominant in the i-th extension. Clearly, the probability that $X = 1$ is at least the probability of this intersection. Note that the probability of this intersection is the probability that $X = 1$ given that p_j is dominant times the probability that p_j is dominant, which is $1/k$. The probability that $X = 1$ given that p_j is dominant is one minus the probability that $X = 0$ given that p_j is dominant. We will show that the probability of all the extensions in which $X = 0$ given that p_j is dominant is less than $\frac{1}{2}$. This implies that the probability that p_j is dominant and $X = 1$ is at least $1/2k$.

In any extension in which p_j is dominant at least $m_{i,j}$ copies of p_j are sent. To bound the probability of the extensions in which only $(q - \epsilon_n)m_{i,j}$ of the $m_{i,j}$ copies of p_j are delayed we use Hoeffding bound. This probability is at most $e^{-2m_{i,j}\epsilon_n^2} < e^{-nq\epsilon_n^2/2k^2}$. For $\epsilon_n = O(1/\sqrt{n})$ this probability is $\leq \frac{1}{2}$. Therefore, the probability that p_j is dominant and more than $(q + \epsilon_n)m_{i,j}$ packets are delayed is at least $1/2k$.

To conclude the proof we show that with high probability in at least $n/8k^2$ of the last $n/2k$ extensions in which p_j is probable dominant the number of its delayed copies is increased by a factor of $(1 + q - \epsilon_n)$. We already have that the probability that the number of copies of p_j is be increased in one of these $n/2k$ extensions is $\geq 1/2k$. Thus the probability that it is increased in $n/8k^2$ of the $n/2k$ extensions is at least $1 - e^{-n/16k^3} = 1 - e^{-\Omega(n)}$.
□

References

[Afe88] Y. Afek. 1988. Personal communication.

[AFWZ88] H. Attiya, M.J. Fischer, D. Wang, and L.D. Zuck. Reliable communication using unreliable channels. 1988. Manuscript.

[BSW69] K.A. Bartlett, R.A. Scantlebury, and P.T. Wilkinson. A note on reliable full-duplex transmission over half-duplex links. *Communications of the ACM*, 12:260–261, 1969.

[Hoe63] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[LMF88] N.A. Lynch, Y. Mansour, and A. Fekete. Data link layer: two impossibility results. In *Proc. of the 7th ACM Symp. on Principles of Distributed Computing*, pages 149–170, Toronto, Canada, August 1988.

[LT87] N.A. Lynch and M.R. Tuttle. Hierarchical correctness proofs for distributed algorithms. In *Proc. of the 6th ACM Symp. on Principles of Distributed Computing*, pages 137–151, August 1987.

[Tan81] A. S. Tanenbaum. *Computer Networks*. Prentice-Hall, 1981.

[WZ89] D. Wang and L.D. Zuck. Tight bounds for the sequence transmission problem. In *Proc. of the 8th ACM Symp. on Principles of Distributed Computing*, Edmonton, Alberta, August 1989. To appear.