PERFORMANCE OF LAMS-DLC, A PROTOCOL FOR LOW ALTITUDE SATELLITE NETWORKS

Technical Report CSE-92-07

Christopher Ward
and
Cheong H. Choi

Department of Computer Science and Engineering
Auburn University, AL 36849-5347

Performance of LAMS-DLC, a Protocol for Low Altitude Satellite Networks

Christopher Ward and Cheong H. Choi

Department of Computer Science and Engineering,
Auburn University
Auburn, AL 36849

Abstract

This paper introduces a new data link control protocol appropriate for communication in low altitude satellite networks. A low altitude satellite network is assumed to be a system of satellites in the range 1000 to 15000 km, such that satellites may exchange messages while they are in line of sight with one another. In this environment data rates are expected to be Mbps, propagation delays are measured in ms, and SWAP (size weight and power) constraints restrict buffer size. The proposed data link control protocol is called LAMS-DLC (low altitude multiple satellite data link control). Unlike traditional ARQ schemes LAMS-DLC uses negative acknowledgment to provide a lossless packet stream without guaranteeing delivery of packets in sequence. Throughput and buffer size metrics are computed and compared with Selective Repeat HDLC, a typical acknowledgment-based ARQ protocol.

1 Introduction

In this paper we introduce a new data link control (DLC) protocol for communications in low altitude multiple satellite (LAMS) networks. The protocol is called "LAMS-DLC". The target network which we consider forms a unique environment which is composed of highly mobile satellites with store-and-forward and point-to-point communication capabilities. This network environment has distinctive characteristics, in particular, long propagation delays, high error rates, high bandwidth and short link lifetime (i.e. the time period of an active link) when compared with conventional network environments [1, 2]. LAMS networks also have a large retargeting overhead which occupies a significant portion of the link lifetime.

Many DLCPs assume reliable packet transfer as a constraint in their design [3, 4]. In this context, the term reliability implies that data is accepted at one end of a link in the same order as was transmitted at the other, without loss and without duplicates. These reliability constraints, which we call *strict reliability*, are summarized as 1) no loss 2) no duplicate 3) FIFO (in sequence) delivery [4]. Additional reliability constraints require error free procedures for link initialization, link failure detection, and resynchronization [4]. In this paper, we consider only those reliability constraints related to DLCP performance. In networks with small propagation delays and fixed links, this constraint does not significantly affect the network performance. However, this reliability constraint is one of the barriers which prevents performance improvements in networks with very long propagation delays. We believe that by relaxing the reliability constraint, it is possible that a new DLC protocol with higher performance can be designed.

Our first observation is that each link in a LAMS network is active during a relative short time period (in the order of several minutes) thus frequent changes occur at the link layer. Our second observation is that a LAMS network environment requires a large buffer size for continuous operation (since the buffer size requirement is proportional to the product of the link distance, the data rate and the queueing delay). Therefore, in LAMS-DLC we reduce the queueing delay by relaxing reliability constraints in so far as they do not cause serious problems. Notice that we may relax the in-sequence requirement without I-frame loss [5]. Thus a receiver is no longer required to store out-of-sequence I-frames, instead it immediately forwards them to the next node. Notice that to provide reliable message delivery for its users, the destination node now has responsibility for sequencing. Notice also that relaxing the in-sequence requirement is distinct from relaxing the no-loss requirement which would cause excessive delay if a packet were lost early in the route. Clearly we require that the destination node have sufficient buffer space to keep out-of-sequence I-frames and sufficient processing power to resequence them.

We structure our paper as follows. In Section 2 we introduce LAMS-DLC, a link-layer protocol appropriate for use in a LAMS network. In section 3 we derive performance measures for LAMS-DLC and compare them with those of SR-HDLC

2 LAMS-DLC

We now discuss LAMS-DLC. While there are several aspects to DLC protocol design in this paper we consider error control only.

2.1 Frames, Command and Response

In LAMS-DLC, like HDLC we define two types of frame format: information frames (I-frames) contains user bits and are sequentially numbered with a sequence number N(S) and control frames (C-frames). Unlike HDLC, LAMS-DLC does not permit the use of piggybacking for acknowledgment, although it does

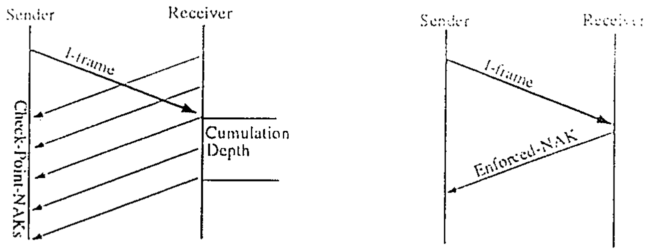

Figure 1: Check-point and Enforced NAKs in LAMS-DLC

use piggybacking for flow control. We define three control commands: Check-Point-NAK (Check-Point-Command), Enforced-NAK (Resolving Command), and Request-NAK. Check-point-NAK and Enforced-NAK contain sequence numbers (N(R)'s) which request retransmission of erroneous I-frames. Their length varies according to the number of the erroneous I-frames communicated. All control commands except for Request-NAK have the same format and include a Stop-Go bit for flow control. Check-Point-NAK (Command) and Enforced-NAK (Resolving Command) are distinguished by Enforced-bit. Operation of these C-frames is shown in Figure 1.

2.2 Error Recovery

LAMS-DLCP is unique in that unlike most ARQ schemes, its error control scheme is based primarily on the periodic issue of Negative Acknowledgment. That is, the protocol specifies that should the receiver detect errors during a pre-defined time interval (*Check Point Interval*), it responds with a NAK, however, if no error is reported by the NAK, the sender may assume that the corresponding I-frames successfully arrived. Should the sender not receive a response within a predefined duration (*Cumulation Depth*), it suspects the link as having failed. The sender then checks whether a link failure has occurred using an Enforced Recovery. This notion guarantees that an I-frame will not be lost. Note, however, that this may lead to I-frame duplication if the link failure is not recoverable during the link lifetime. [1] We assumed that link behavior of a LAMS network is deterministic, (i.e. the subnet nodes knows the precise distances and variance of the link) thus the sender will be able to compute when a failure occurred based on the expected arrival time of Enforced recovery C-frames.

We now discuss the details of how the NAK based protocol is used in LAMS-DLC. Like HDLC, error recovery of LAMS-DLC is divided into two different recovery methods, Check-Point Recovery and Enforced Recovery. First we discuss Check-Point Recovery.

In general, a NAK is issued by the receiver when an erroneous I-frame arrives. The NAK contains the sequence number(s) of one or more erroneous I-frames recognized since its last invocation, thus distinct NAKs contain disjoint information. A Check-Point NAK is again a NAK but includes selective cumulative information regarding previous erroneous I-frames during a cumulation depth and is used to ensure that erroneous frames are properly retransmitted. In LAMS-DLC, the receiver periodically sends a Check-Point command during regular communication. The interval between these Check-Point commands is called the "Checkpoint Interval", denoted W_{cp}. The primary objective of a Check-Point command is to relieve the sender of buffer space allocated for unacknowledged but error-free I-frames. The Check-Point command also implicitly functions as a positive acknowledgment. If there have been erroneous I-frames in the previous predefined number of check-point-intervals called the "Cumulation Depth" C_{depth}, the Check-Point command will also contain information on these erroneous I-frames. Thus, the Check-Point command with its imbedded error control information, is called "Check-Point-NAK". Since each Check-Point NAK contains information on the erroneous frames to a depth C_{depth}, the sender will be notified of an erroneous I-frame C_{depth} times, however, the sender responds with retransmission of the I-frame only one time after the first notification. After this response a new sequence number is assigned to the retransmitted I-frame. This is allowed because LAMS-DLC need not satisfy the in-sequence constraint. Therefore, if a sequence number reported by a Check-Point NAK does not match those unacknowledged I-frames in the sending buffer, the unacknowledged I-frame with that sequence number is assumed to have been already retransmitted. We call this recovery operation "Check-point Recovery".

Enforced-NAK is issued in response to a Request-NAK control frame, which acts like the P/F-bit check-point recovery of HDLC. If the sender suspects a link failure due to a lack of response during the previous C_{depth} check-point-intervals, the sender sends a Request-NAK to the receiver provided that the expected response time is within the remaining link lifetime (recoverable link failure). Simultaneously, the sender stops sending I-frames and starts a failure timer. The receiver keeps sending Check-Point commands regardless of link failure. If the sender again receives a Check-Point command (NAK) non-associated with Request-NAK, the sender may do Check-Point Recovery (i.e. retransmission) but cannot send new I-frames. Upon receiving a Request-NAK the receiver must respond immediately with an Enforced-NAK. This can be done by setting Enforced-bit of Check-Point NAK (Command) to 1 and including all sequence numbers of previously erroneous I-frames during the resolving period prior to the Request-NAK [2]. When the sender receives Enforced-NAK, the sender stops the failure timer and resends all the unacknowledged I-frames with the sequence numbers indicated by Enforced-NAK. If the receiver has no information on erroneous I-frames it may be regarded as simply a command for resynchronization which also permits buffer space to be released. Under these circumstances the Enforced-NAK is called a "Resolving Command", and the recovery sequence an "Enforced Recovery".

In LAMS-DLC, the enforced recovery is initiated by expiration of a timer called the "checkpoint timer". The checkpoint timer is started upon reception of the first check-point command during a communication. Then the checkpoint timer is reset to zero after each Check-Point Command. The timeout of the checkpoint timer is $C_{depth} \cdot W_{cp}$. A Request-NAK also triggers the failure timer associated with failure checking. If the sender receives neither an enforced-NAK nor a resolving-command within the normally expected response time plus $C_{depth} \cdot W_{cp}$, the sender regards the receiver as having failed. This is justified by noting that the probability of successive check-point-commands all failing is negligible ($P_C^{C_{depth}} < \epsilon$ where ϵ is vanish-

---

[1] A minor modification to LAMS-DLC guarantees zero duplication as well as zero loss, however, analysis for this model has yet to be completed.

[2] Where the resolving period is the time period that transmission of a I-frame is acknowledged as in error or is a success, in LAMS-DLC this is bounded.

ingly small[3]). Figure 2 demonstrates this concept. Once the sender determines a link failure has occurred it stops transmitting I-frames and informs the network layer.

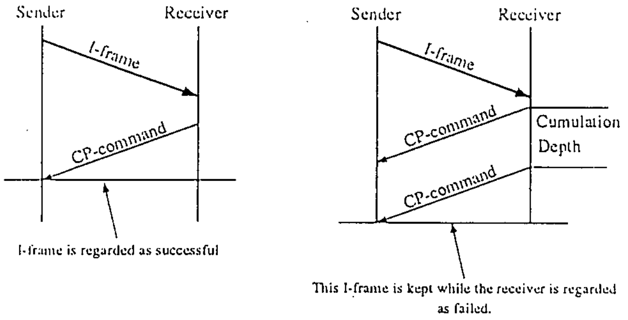

Figure 2: Normal operation vs link failure in LAMS-DLC

3 Performance of LAMS-DLC

In this section we compute the performance of LAMS-DLC. In particular we compute throughput efficiency and buffer size for different bit error rates.

Let a random variable S be the number of periods including the transmission period and retransmission periods. Regardless of how many I-frames are sent in a transmission period, the distribution of the random variable S follows the behavior of a single I-frame since erroneous I-frames are considered independent events. We first calculate $\bar{s}$ the mean number of the total periods (S) required for the successful delivery of an I-frame available in the sending buffer. This includes the transmission and retransmission periods.
The density function follows the Geometric distribution as:

$$\text{Prob}[S = k] = (1 - P_R)P_R^{k-1}$$

where P_R is the probability that an I-frame will be retransmitted due to some error.

Thus, the expected (i.e. mean) value is:

$$\bar{s} = E[S] = \frac{1}{1 - P_R}$$

P_R varies according to the protocol used. We note that the only invariant is the bit error rate. However, we assume that both the I-frame error rate and the command frame error rate are also invariant in order to derive expressions using only these probabilities. The question is therefore, how is P_R expressed by these probabilities? Let P_F and P_C be the probabilities that an I-frame and a control command are erroneous respectively.

Recall that LAMS-DLC is a NAK based protocol. An I-frame is retransmitted only when a received I-frame is in error and the receiver sends a NAK. Since the probability of failure of all the check-point-commands associated with the failed I-frame is nearly zero, the probability that a I-frame is retransmitted is identical to the probability that the I-frame was in error.

$$\begin{aligned} P_R^{LAMS} &= \text{Prob[I-frame has an error]} \\ &= P_F \end{aligned}$$

So

$$\bar{s}_{LAMS} = \frac{1}{1 - P_F}$$

We now calculate the mean length of the transmission period and the retransmission period, denoted by D_{trans} and D_{retrn} respectively. We make the following assumptions to simplify the derivation: every I-frame is of the same size; and the sender receives no new I-frames until N I-frames are successfully transmitted. The latter assumptions imply low traffic. We also assume that the buffer space is sufficient to accommodate all frames.
Let

R = Round trip time between two connected nodes,
I_{cp} = The length of the interval between two consecutive cp-commands,
W = The window size for HDLC,
t_f, t_c = Transmission times of an I-frame and a control command respectively,
t_{proc} = The maximum time required to process an I-frame or a control command,
t_{out} = The timeout defined in HDLC,
$\bar{n}_{cp}$ = The mean number of cp-commands needed to ensure an I-frames reliable delivery, $= \frac{1}{1 - P_C}$,
C_{depth} = The number of the consecutive cp-commands covering a I-frame.

In LAMS-DLC, each of C_{depth} consecutive cp-commands contains the NAK associated with the erroneous I-frame. Most I-frames are successfully covered by the first one of these cp-commands. But if the first cp-command is corrupted, the next cp-command would be an acknowledgment to the I-frame instead. Thus the mean number $\bar{n}_{cp}$ of cp-commands required to acknowledge an I-frame must be considered to calculate the length the distinct periods, D_{trans} and D_{retrn}. Note that LAMS-DLC gives rise to two new delays. The first delay is caused by the period between the arrival of an I-frame and the transmission of the following cp-command. Since I-frames are assumed to arrive uniformly during the cp-command interval, the mean delay is $I_{cp}/2$. The second delay is due to the possible loss of the expected cp-command which increases the delay by as much as one I_{cp}. This delay is therefore $(\bar{n}_{cp} - 1) \cdot I_{cp}$.

$D_{trans}^{LAMS}(N)$ = Transmission time of N I-frames,
$(N \cdot t_f) +$
One way propagation time, $(R/2) +$
The mean delay from the last I-frame to the following cp, $(\frac{1}{2} \cdot I_{cp}) +$
$(\bar{n}_{cp} - 1)\cdot$ cp-interval, $((\bar{n}_{cp} - 1) \cdot I_{cp}) +$
One way propagation time, $(R/2) +$
Transmission time of the cp-command,
$(t_c) +$

---

[3] For example given a BER of 10^{-7}, $P_C^{depth} \leq 10^{-10}$

$$\begin{aligned} & \text{Processing time of the cp-command,} \\ & (t_{proc}) \\ = {} & N \cdot t_f + t_c + t_{proc} + R + (\bar{n}_{cp} - \frac{1}{2}) \cdot I_{cp} \end{aligned}$$

Consider now the length of the retransmission period, D_{retrn}. The retransmission of erroneous I-frames may begin once a valid cp-command concerning these I-frames arrives at the sender. If we assume that erroneous I-frames are uniformly distributed among the I-frames transmitted in a period, the final cp-command to close a period will report the erroneous I-frames that occurred during the cp-interval. Even given that the cp-command reports errors it is reasonable to assume that the number of associated I-frames is on average one. Thus D_{retrn} is equivalent to D_{trans} except for the transmission time.

$$\begin{aligned} D^{LAMS}_{retrn} = {} & \text{Transmission time of one I-frame, } (t_f) + \\ & \text{One way propagation time, } (R/2) + \\ & \text{The mean delay from the last I-frame to} \\ & \text{the following cp, } (I_{cp}/2) + \\ & \text{One way propagation time, } (R/2) + \\ & \text{Transmission time of a cp-command, } (t_c) + \\ & (\bar{n}_{cp} - 1)\cdot \text{ cp-interval, } ((\bar{n}_{cp} - 1) \cdot I_{cp}) + \\ & \text{processing time of the command, } (t_{proc}) \\ = {} & t_f + t_c + t_{proc} + R + (\bar{n}_{cp} - \frac{1}{2}) \cdot I_{cp} \end{aligned}$$

Therefore, the mean total time required for the safe delivery of N I-frames denoted as $D^{LAMS}_{low}(N)$ is:

$$\begin{aligned} D^{LAMS}_{low}(N) &= D^{LAMS}_{trans}(N) + (\bar{s}_{LAMS} - 1)D^{LAMS}_{retrn} \\ &= (N + \bar{s}_{LAMS} - 1)t_f + \\ &\quad \bar{s}_{LAMS}(R + t_c + t_{proc}) + \bar{s}(\bar{n}_{cp} - \frac{1}{2})I_{cp} \\ &\approx Nt_f + \bar{s}_{LAMS}R + \bar{s}_{LAMS}(\bar{n}_{cp} - \frac{1}{2})I_{cp} \end{aligned}$$

So in low traffic given an infinite buffer, the mean total time for N I-frames is:

$$\begin{aligned} D^{LAMS}_{low}(N) &\approx Nt_f + \bar{s}_{LAMS}R + \bar{s}_{LAMS}(\bar{n}_{cp} - \frac{1}{2})I_{cp} \\ D^{HDLC}_{low}(N) &\approx Nt_f + \bar{s}_{HDLC}R + \\ &\quad ((\bar{s}_{HDLC} - 1)(1 - P_F - P_C + P_F P_C) \\ &\quad -P_C)\alpha \end{aligned}$$

We note the total period for successful delivery of N I-frames in SR-HDLC and LAMS-DLC are nearly equivalent if $\bar{s}_{LAMS}$ is equal to $\bar{s}_{HDLC}$ and α is small. However, it is likely that $\alpha \gg \bar{n}_{cp}$ in a rapidly changing network and $\bar{s}_{HDLC} > \bar{s}_{LAMS}$ in a high error environment. Therefore, we would expect the transmission delay for N I-frames in SR-HDLC to be greater than $D^{LAMS}_{low}(N)$ in a LAMS network even given low traffic and an infinite buffer. Although an infinite buffer is unreasonable in a LAMS network it can be realized with a transparent buffer size which allows the protocol to operate efficiently without suffering from buffer shortages. In low traffic, if the removal rate of frames from a buffer is equal to, or greater than, the incoming rate into the buffer, the transparent buffer size is bounded. The removal rate depends on the protocol scheme and the computational capability of a node, an upper bound will clearly be $1/t_f$. The incoming rate depends on the intensity of traffic and the number of incoming links ignoring the effects of flow control. Therefore its upper bound is c/t_f where c is parameter varying according to the number of links in a node, $c \geq 1$. We assume $c = 1$ for simplification because the transparent buffer size is linearly proportional to c in a deterministic model.

We now compute the transparent buffer size. From the perspective of a DLC there are two buffer types: the sending buffer and receiving buffer. In a communication between two peer DLC protocols, only the sending buffer of the sender and the receiving buffer of the receiver need be considered. For our analysis of transparent buffer size we therefore assume that the sending buffer of the sender is filled by the network layer at a rate of $1/t_f$. In this model, the receiver sends eligible I-frames to the network layer as soon as they are resolved. Since the sending and receiving buffers behave differently, we first consider the mean period that an I-frame has to wait in the sending buffer after it is transmitted, called the "Holding Time" of the sender. In LAMS-DLC, buffer behavior is associated with the mean holding time of an individual frame, not $D^{LAMS}_{low}(N)$ the mean total time for N frames. Thus we use a different approach to derive the mean holding time of the sending buffer.

Let

H^{LAMS}_{frame} = The mean holding time of a I-frame
H^{LAMS}_{succ} = The mean holding time of a successful I-frame
H^{LAMS}_{fail} = The mean holding time of an erroneous I-frame.

We can express H^{LAMS}_{frame} as follows:

$$H^{LAMS}_{frame} = (1 - P_F) \cdot H^{LAMS}_{succ} + P_F \cdot H^{LAMS}_{fail}$$

Then the mean holding time of the sending buffer for a successful I-frame is:

$$\begin{aligned} H^{LAMS}_{succ} &= D^{LAMS}_{trans}(1) \\ &= R + t_f + t_c + t_{proc} + (\bar{n}_{cp} - \frac{1}{2})I_{cp} \end{aligned}$$

In the case of H^{LAMS}_{fail}, we can express it with H^{LAMS}_{frame} and H^{LAMS}_{succ} recursively as follows: an erroneous I-frame is first retransmitted $D^{LAMS}_{trans}(1)$ after transmission of the I-frame. After the sender retransmits that I-frame, the mean holding time of the new I-frame is again H^{LAMS}_{frame} since the behavior of the I-frame is independent from its preceding transmission. Therefore the mean holding time for an erroneous I-frame is defined as follows:

$$\begin{aligned} H^{LAMS}_{fail} &= H^{LAMS}_{succ} + H^{LAMS}_{frame} \\ &= R + t_f + t_c + t_{proc} + (\bar{n}_{cp} + \frac{1}{2})I_{cp} + H^{LAMS}_{frame} \end{aligned}$$

We can make a new recursive expression using these three expressions specified above:

$$\begin{aligned} H^{LAMS}_{frame} &= (1 - P_F) \cdot H^{LAMS}_{succ} + \\ &\quad P_F \cdot (H^{LAMS}_{succ} + H^{LAMS}_{frame}) \end{aligned}$$

So

$$\begin{aligned} H^{LAMS}_{frame} &= \frac{1}{1-P_F} \cdot H^{LAMS}_{succ} \\ &= \frac{1}{1-P_F} \cdot (R + t_f + t_c + t_{proc} + \\ &\quad (\bar{n}_{cp} - \frac{1}{2})I_{cp}) \\ &= \bar{s}_{LAMS} \cdot (R + t_f + t_c + t_{proc} + \\ &\quad (\bar{n}_{cp} - \frac{1}{2})I_{cp}) \\ &\approx \bar{s}_{LAMS} \cdot (R + (\bar{n}_{cp} - \frac{1}{2})I_{cp}) \end{aligned}$$

In LAMS-DLC, valid I-frames will be sent to the packet layer immediately after their arrival since LAMS-DLC provides an out-of-sequence zero packet loss service. As a result, provided the receiving buffer can hold t_{proc}/t_f frames at a time, that size is sufficient for transparency. Let B_{LAMS} be the transparent buffer sizes for LAMS-DLC. Initially the sending buffer increases during H^{LAMS}_{frame} because all frames are unresolved and these frames must be kept in the sending buffer. But after time period H^{LAMS}_{frame}, the removed rate from the sending buffer will to be approximately $1/t_f$. The sending buffer is now stable in the deterministic model (although in the M/M/1 queueing model this rate implies an infinite buffer). Therefore, the transparent sending buffer size is equal to the number of frames flowing into the sending buffer during H^{LAMS}_{frame}. So, the mean buffer size B_{LAMS} for transparent operation (i.e. the mean buffer size) is:

$$\begin{aligned} B_{LAMS} &= \text{size(the sending buffer)} + \\ &\quad \text{size(the receiving buffer)} \\ &= \frac{1}{t_f} H^{LAMS}_{frame} + \frac{t_{proc}}{t_f} \\ &= \frac{1}{t_f}(\bar{s}_{LAMS} \cdot (R + t_f + t_c + t_{proc} + \\ &\quad (\bar{n}_{cp} - \frac{1}{2})I_{cp})) + \frac{t_{proc}}{t_f} \\ &\approx \frac{1}{t_f}\bar{s}_{LAMS}(R + (\bar{n}_{cp} - \frac{1}{2})I_{cp}) \end{aligned}$$

In high traffic $N > H^{LAMS}_{frame}/t_f$, therefore the mean total times required for successful transmission of N frames is:

$$D^{LAMS}_{high}(N) = D^{LAMS}_{low}(N^{LAMS}_{total})$$

As a consequence, in high traffic, LAMS-DLC shows the following throughput efficeincy η_{LAMS} with the transparent buffer size B_{LAMS}:

$$\begin{aligned} \eta_{LAMS} &= \frac{N}{D^{LAMS}_{high}(N)} \\ &= \frac{N}{N^{LAMS}_{total} t_f + \bar{s}_{LAMS} R + \delta_{LAMS}} \end{aligned}$$

where $B_{LAMS} = \frac{1}{t_f}\bar{s}_{LAMS}(R + (\bar{n}_{cp} - \frac{1}{2})I_{cp})$ and $\delta_{LAMS} = \bar{s}_{LAMS}(\bar{n}_{cp} - \frac{1}{2})I_{cp}$
Results for throughput efficiency based on these equations are given in Figures 3 and 4.

Notice that in heavy traffic LAMS-DLC performance favorably when compared with SR-HDLC.

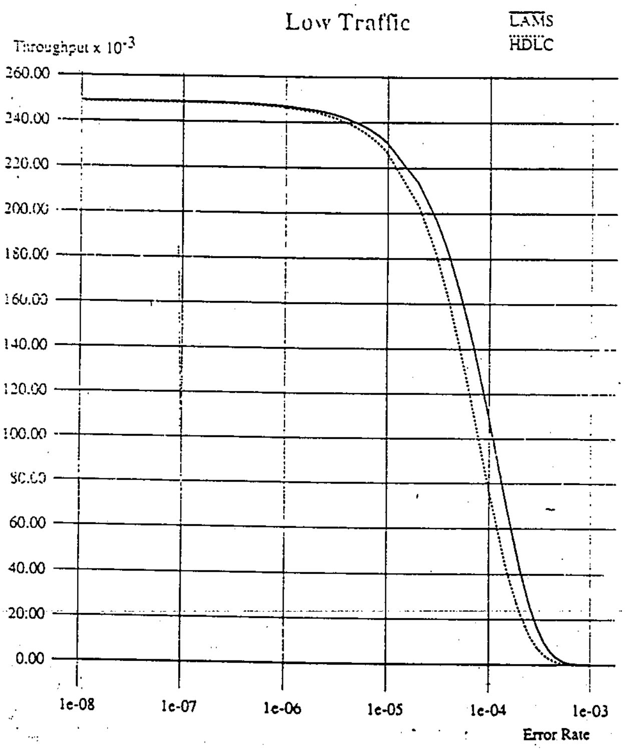

Figure 3: Throughput efficiency vs BER for low traffic rates using SR-HDLC and LAMS-DLC

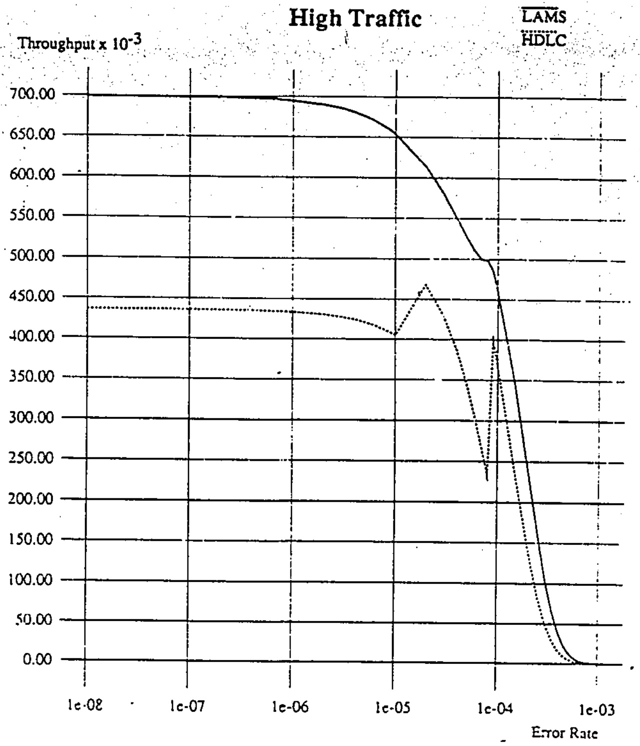

Figure 4: Throughput efficiency vs BER for high traffic rates using SR-HDLC and LAMS-DLC

4 Summary

In this paper we have presented LAMS-DLC, a DLCP for use in low altitude multiple satellite networks. These networks are expected to have distinctive characteristics which include long propagation delays, high error rates, high bandwidth and short link life-times when compared with traditional networks. The LAMS-DLC protocol is designed to take advantage of these characteristics by using NAKs and a check-point mechanmism to provide a zero-loss data stream. In the envisaged application, where data will be highly encoded, this is expected to be sufficient. The paper includes a derivation for the throughput efficiency and buffer size for LAMS-DLC. Results indicate that LAMS-DLC will perform well in its target environment.

References

[1] D. K. Paul, F. Hemmati, and R. G. Marshalek. Forward error correction and its impact on high-data-rate, free-space laser communication system design. Joint sposorship, COMSAT Laboratories, Clarkburg, Maryland 20871, 1989.

[2] W.D. Brodd and R.A. Donnan. Data Link Control Improvements for Satellite Transmission. In J.L.Grange, editor, *Satellite and Computer Communications*, pages 201–213. IFIP TC 6, North-Holland, February 1983.

[3] Malcolm C. Easton. Design Choices for Selective-Repeat Retransmission Protocls. *IEEE transactions on Communications*, 29(07), July 1981.

[4] Alan E. Baratz and Adrian Segall. Reliable Link Initialization Procedures. *IEEE Transactions on Communications*, 36(2), February 1988.

[5] Christopher Ward and Cheong H. Choi. The lams-dlc arq protocol. In *Sigcomm'91*, Zurich, Switzerland, September 3-6 1991.