THE LAMS-DLC ARQ PROTOCOL

Technical Report CSE-91-03

Christopher Ward, Cheong H. Choi

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

March 8, 1991
The LAMS-DLC ARQ Protocol

Christopher Ward
Cheong H. Choi
Computer Science and Engineering Department,
Auburn University
March 8, 1991

Abstract

Abstract-In this paper we introduce a new data link control protocol appropriate for a low earth orbit (LEO) satellite network. The principal characteristics of an LEO satellite network are high error rates, long propagation delay, and high mobility. Although this area is traditionally dominated by forward error correction (FEC) we present a reliable protocol suited to this environment and derive the throughput for it given a finite buffer model. In our target network environment, reliability constraints may be relaxed provided that the protocol provides zero packet loss capability. The proposed scheme (called LAMS-DLC) provides a datagram service at the link level based on a Negative Acknowledgement ARQ scheme to accomplish error recovery. In LAMS-DLC the receiver sends commands for the following reasons only: to enable the sender to release buffer space held for previously transferred information frames (I-frames); to retransmit unsuccessful I-frames; and to identify when the sender has failed. Commands are sent by the receiver so long as the link is active. The performance of the protocol is compared with the popular ARQ protocol HDLC. We show that LAMS-DLC performs better than the selective repeat HDLC and offers certain advantages in satellite networks.
1 Introduction

During the late 1970s and 1980s Go-Back-N (GBN) and Selective-Repeat (SR) Automatic Request (ARQ) schemes were developed to provide highly reliable link layer communications [1, 2, 3, 4, 5, 6]. These two ARQ schemes show near optimal performance in their target communication environments, which are characterized by a relatively small propagation delay, relatively low data rates and error rate, and are typical of many terrestrial network environments. Variations of these protocols include Stutter GBN ARQ which was motivated by using the idle time to repeatedly send unacknowledged frames [1]. Miller and Lin [3] proposed a mixed mode ARQs—Selective Repeat plus Stutter (SR + ST) and Selective Repeat plus Go-Back-N (SR + GBN). These mixed mode ARQs were designed with the same motivation as the Stutter GBN except that their emphasis is placed on the finite buffer size.

In addition, some variations of HDLC for Point-to-point satellite links were proposed in [7, 8]. NBDT (NADIR Bulk Data Transfer Protocol) [7] applied a variety of retransmission strategies to HDLC, using the absolute numbering scheme and selective acknowledgement scheme. The protocol was motivated from the numerous drawbacks of standard HDLC in a point-to-point satellite link—the need of a huge memory size and the throughput degradation. Absolute numbering uses 32 bit sequence number field ($2^{32}$ numbers) which allows the frame size to be controlled for the optimal size. NBDT had two improved modes (Multiphase mode and Continuous mode) that attempt to use the idle time of HDLC. In multiphase mode, the sender performs transmissions and retransmissions alternately on the basis of completely selective acknowledgement during a communication. In continuous mode, transmissions and retransmissions can be mixed during a communication. Critical problems of NBDT are that the huge memory is implemented by secondary device, and they do not consider the reliability of protocol. In another variation of HDLC, The motivation was to increase the number of I-frames in transit without a significant investment. Brodd and Donnan [8] proposed that frequent acknowledgements could accomplish this design goal, resulting in the IBM check-point mode variation. This check-point mode was based on periodic acknowledgement. Again in their paper, they do not consider the protocol reliability, buffering, and numbering.
For high throughput efficiency, Forward Error Correction (FEC) schemes have also been studied [9, 10, 11, 12]. We note, however, that FEC schemes have rarely been considered solely as error control schemes for DLC protocols. On the other hand, the combination of ARQ and FEC have been proposed to offer high reliability and improved performance in environments with high error rate [13, 14, 15]. Examples of combined schemes are Type-I, Type-II and memory ARQ. Their motivation is that the relatively low throughput of ARQ schemes is caused by retransmissions and is dependent upon the number of retransmissions required for the successful delivery of an I-frame. In Type-I, both the error detecting code and the information are encapsulated by an FEC code to lower the probability of retransmission. In Type-II, the sender alternately sends the original frame and its error correcting code at retransmission time. In memory ARQ, the original information is sent as the first transmission but its retransmissions contain carefully structured incremental information to eliminate corruption in the original transmission [14].

We also note that these researchers primarily focused on the coding technique appropriate to specific characteristics of an environment. For example, Paul [10] proposed the use of an interleaving code to translate the burst error due to mispointing of laser antenna into a random error.

In this paper we discuss an ARQ based data link control (DLC) protocol for intersatellite communications using point-to-point laser links through free space, called “LAMS-DLC”. The target network which we consider forms a unique environment which is composed of highly mobile low altitude multiple satellites (LAMS) 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 [16, 10, 8]. LAMS networks also have a large retargeting overhead which occupies a significant portion of the link lifetime [17]. Traditionally delay, queue length (buffer size) and throughput are the main performance characteristics of a data transmission system when using ARQ. It is also well known that there is a tradeoff point between high user throughput and low user delay in end-to-end data transmission [18]. We note, however, that both the throughput efficiency and the queueing delay associated with FIFO-based ARQ schemes are linearly dependent on the link error rate, the link distance, the data rate and the DLC protocol scheme used.
In considering the DLCs so far we notice that they assume reliable packet transfer as a constraint in their design [2, 19]. In networks with small propagation delays and fixed links, this constraint does not significantly affect the network performance [3]. 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. Conventional DLCs in a LAMS network exhibit significantly lower throughput efficiency due to these LAMS characteristics [7, 8]. As with other ARQ protocols our design objectives for LAMS-DLC remain the same, however, we have tailored the protocol to the LAMS environment. 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 LAMS-DLC should be designed to minimize the impact of idle time due to link initialization and link (re)synchronization, i.e. LAMS-DLC should be designed to maximize the throughput efficiency during the short time period available for data delivery. Our second observation is that a LAMS network environment requires 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) [7]. Therefore in LAMS-DLC we reduce the queueing delay by relaxing reliability constraints in so far as they do not cause serious problems. This is discussed further in Section 2.

We structure our paper as follows. In Section 2 we provide a detailed discussion of the following: characteristics of a LAMS network; a simple communication model for LAMS-DLC; the characteristics of traditional DLCs; and the problems of their application to a LAMS network and the technical aspects of LAMS-DLC. We also identify reasons why traditional DLC reliability considerations may be relaxed. In Section 3 we describe LAMS-DLC, a link-layer protocol appropriate for use in a LAMS network. In section 4 we derive performance measures for both LAMS-DLC and the selective repeat HDLC (SR-HDLC).
2 DLCP Background

Many highly reliable DLCPs are based on the ARQ scheme. Although FEC coding schemes can, in principal, provide the high reliability of ARQ schemes it is only at with considerable redundancy and encoding/decoding overhead. ARQ schemes also have the advantage of attaining this high reliability through simple operations. 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 delivery [2, 19]. Additional reliability constraints require error free procedures for link initialization, link failure detection, and resynchronization [19]. In this paper, we explain only those reliability constraints related to DLCP performance.

The reason that the popular DLCPs like HDLC or LAP-B require such reliability constraints is illustrated as follows. In the case of Normal Response Mode (NRM) of HDLC, a secondary node is assumed to have only limited capability for processing and buffering while the primary-secondary link is assumed to have a low error rate and short propagation delay. Therefore, it is necessary that the DLC procedure ensure strict reliability to prevent erroneous frames in the secondary. In this context, we note that GBN-HDLC is often preferred despite it's inferior performance when compared with SR [20]. In the case of Asynchronous Balanced Mode (ABM), HDLC provides data delivery to the higher layers assuming a connection-oriented services (Virtual Circuit). Should HDLC not provide strict reliability for the higher layers, those higher layers will require additional resources to provide this function. Note that in a LAMS network it will be difficult to provide a virtual circuit service, since high mobility implies that links will exist for short periods only.

Recall that improvements in the throughput of ARQ schemes depend on a reduction in the number of retransmissions and the productive use of idle time. Note that idle time is inevitable because of protocol synchronization [21]. In general ARQ schemes use Positive Acknowledgement (pos-ack) as well as Negative Acknowledgement (NAK) for notifying the sender of the status of previously transmitted I-frames. Thus pos-ack based ARQ schemes
increase the probability that an I-frame is retransmitted, denoted $P_R$. Consider, let $P_F$, $P_C$ be the probabilities that an I-frame and control frame are erroneous respectively. The retransmission probability of the pos-ack based ARQ scheme is at least $P_F + P_C - P_FP_C$. Using piggyback acknowledgments, $P_C = P_F$, therefore $P_R = 2P_F - P_F^2$. A more detailed analysis is given in Section 4. If only a NAK is used in the ARQ scheme then $P_R = P_F$. We realize, of course, that if only NAK is used and it is lost the DLCP will suffer I-frame loss. Therefore, in the design of LAMS-DLC we will incorporate a mechanism to avoid I-frame loss due to NAK losses.

2.1 Characteristics of LAMS network

Reference to a satellite network typically implies the following properties: one or more satellites placed in geosynchronous orbits; satellites used as communications relays; point-to-point uplinks, with a common channel shared by several earth stations using a combination of FDMA and TDMA; broadcast down-links; and long propagation delays (in the order of 270ms). In these cases the emphasis is placed on contention-based ARQ schemes to provide a reliable data stream [10]. A conventional satellite network with such attributes has a relatively low total channel capacity because of the small number of geosynchronous satellites and the many earth stations demanding their services. Thus, in geosynchronous satellite networks, link capacity is the performance bottleneck. In these networks a major concern is link efficiency. Specifically, the goal is to minimize the waiting time for the channel which in turn yields high link efficiency [22].

In the case of a LAMS network we assume that many satellites are in low earth orbits and communicate with one another as described in [16]. With altitudes of only 1000km propagation delays are in the order of 10 to 100 ms [10]. These new systems employ point-to-point laser communications, which satisfy precise targetting, low power, light weight and small size constraints (so-called SWAP — size, weight and power — constraints [10]). Such systems have wide potential in defense and space applications and are characterized as follows: [23].

1. Multiple satellites in a low altitude orbit functioning as store-and-forward DCE.
2. Point-to-point laser communication with very high bandwidth (300 Mbps - 1 Gbps).

3. Long distance communication links (2,000 km - 10,000 km with relatively high BER ($10^{-5}$ - $10^{-7}$)).

4. Limited communication links per satellites due to SWAP.

5. High mobility.

The laser link channel is characterized by: 1) random errors resulting from optical noise sources — quantum noise, preamplifier thermal noise, dark current noise, detect excess noise, and optical background noise; and 2) burst errors from beam mispointing and subsequent tracking loss [10]. In fact the bit error rate (BER) of point-to-point laser links is very high. Burst errors in particular seriously impede reliable communication. As a result of high BER and given the SWAP constraints it is essential that some form of FEC technique be an integral component for error control in any proposed DLC. Paul et. al. [10] have proposed an interleaving coding technique to convert the burst error into random errors. This makes the codec design easier and reduces its complexity. Their codec was designed using convolutional codes and provides a minimal BER of $10^{-7}$ for satellite laser-links [10]. In the design of FEC schemes for links with high data rates, one of the most important issues is to reduce the complexity of the coding scheme. Thus it is unlikely that a simple CODEC will correct all burst errors. Therefore, the error control scheme of LAMS-DLC provides an ARQ mechanism.

2.2 Intersatellite Link Model

In our link model, we make the following assumptions:

1. Every satellite has an identical architecture with respect to intersatellite communication. This implies that each satellite can anticipate the capabilities (system complexity, buffer size etc.) of the others. Therefore speed mismatch etc. is not an issue.

2. All links operate in a full-duplex mode.
3. Incoming I-frames destined for other nodes are received by the sender and are stored in its sending buffer. The sender forwards these packets whenever the link is available.

4. Two different FEC schemes are used for frame transmissions. In LAMS-DLC control frames contain cumulative error information, therefore one FEC scheme is used for I-frames and another more powerful FEC is used to transmit control frames. Thus we do not permit piggybacking.

5. The delay due to FEC encoding/decoding is an integral part of the transmission media and is ignored throughout the remainder of this paper.

6. Buffer requirements consider a single link only, i.e. they are confined to the sending buffer of the sender and the receiving buffer of the receiver.

7. We do not consider flow control in this paper although it is a necessary component of the protocol. In fact, we briefly discuss a simple flow control scheme during our description of LAMS-DLC, although this is not the theme of this paper.

8. All parameters in the analysis of LAMS-DLC are deterministic, e.g. processing a frame is deterministic. This prevents unstable states as are found in M/M/1 queueing models.

9. Finally, the loss of frames are treated as a detectable error and we assume that no undetectable errors (CRC-violation).

2.3 Effects of Reliability against DLCP's performance

We have commented that reliability constraints adversely affect the performance of ARQ DLCPs. Furthermore, we note that this same constraint also adversely affects the performance of the end-to-end communication as well. Consider the relationship between the different reliability constraints associated with DLCPs mentioned earlier, i.e. in-sequence($I$), no-duplication($D$) and no-loss($L$). Violations of these constraints are denoted $\bar{I}$, $\bar{D}$ and $\bar{L}$ respectively. We define "in-sequence" using these relationships as follows: $I$ implies $D$ and also implies $L$. The converse relationships do not hold. Similarly, $\bar{D}$ implies $\bar{I}$ and $\bar{L}$ implies $\bar{I}$ but not conversely. Thus the effects of $\bar{I}$
is likely to be minimal. That is, $\bar{I}$ implies nothing, therefore by relaxing the in-sequence requirement in the reliability constraints ($I$) we obtain flexibility without violating $D$ or $L$. Let now consider the effects of the in-sequence requirement on the performance of DLCPs.

The in-sequence requirement forces DLCPs in a subnet to either discard or to store out-of-sequenced frames for later acknowledgement [2]. Consider the performance of GBN and SR in high traffic. With the former protocol, an I-frame loss implies the loss of all I-frames immediately following it. We note that if an I-frame loss occurs, there will follow a waste time while I-frames are discarded. We call the maximum number of in-transit frames at the time the link frame length, defined as $(D_{\text{link}} \cdot T_{\text{data}})/(V \cdot L_{\text{frame}})$, where $D_{\text{link}}$ is the link distance, $T_{\text{data}}$ is the data rate, $V$ is the velocity of light, and $L_{\text{frame}}$ is the average frame length. In a network with a large $D_{\text{link}}$ and $T_{\text{data}}$, GBN DLCPs will clearly discard many uncorrupted I frames. In the latter protocol, each intermediate node is required to have sufficient buffer space to hold I-frames until the erroneous I-frame is correctly received. We call the mean time period that a subnet node holds an I-frame as the "holding time". In the SR ARQ scheme, consider the case in which the sender sends a group of I-frames ($G$) and the first I-frame of the group is corrupted, thus the following I-frames of the group are stored. If those I-frames are of the same size, the buffer will have to hold at least $(G - 1)$ I-frames. Given that the flow control scheme is based on a window of size $W$, the buffer size will therefore be at least the window size $W$. So, for SR ARQ the buffer size requirement is defined by flow control, not by error control.

Suppose, however, that we relax the in-sequence requirement, what happens within subnet nodes? In the subnet, a message is partitioned into multiple I-frames, the receiver no longer has to store the out-of-sequence I-frames, instead it forwards them to the next node. To provide a reliable message delivery for its users the destination node now has responsibility to provide sequencing. Notice that relaxing the in-sequence requirement (i.e. $\bar{I}$) is distinct from relaxing the no-loss requirement (i.e. $\bar{L}$) 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. Given that the expected total delay of an I-frame between the source and the destination
is bounded, the overheads due to the buffer requirement and the additional processing power, is easily computed.

All ARQ schemes require a numbering mechanism with which the sender or receiver can uniquely identify each I-frame. This mechanism must satisfy the condition that at an arbitrary time, all unacknowledged I-frames may be uniquely identified. In fact unique numbering is accomplished by cyclically reusing sequence numbers [2]. In DLCs using windowing schemes, the number of unique sequence numbers equals the window size $W$, which we call the "numbering size". If a DLC wishes to send I-frames continuously, the numbering size depends on the larger of the buffer size and the link frame length. In general, however, since the buffer size depends on the link frame length, we see that the numbering size is proportional to the link frame length. Let us calculate the numbering size required for continuous operation. Let $H_{\text{frame}}$ be the maximum frame holding time allowed in a DLC. This is proportional to the link distance. The numbering size must therefore be greater or equal to $H_{\text{frame}}/\bar{L}_{\text{frame}}$, where $\bar{L}_{\text{frame}}$ is the mean I-frame length. In the case of HDLC, $H_{\text{frame}}$ is unbounded because the in-sequence constraint requires the use of the same sequence number for successive retransmissions of the same I-frame, i.e. each I-frame is identified with one number. Therefore $H_{\text{frame}}$ is the period from when the I-frame was sent to the receipt of its pos-ack. This is usually unbounded. In the case of LAMS-DLC periodic responses and a relaxation of the in-sequence constraint allows $H_{\text{frame}}$ to be bounded. Therefore the numbering size of LAMS-DLC is bounded with respect to continuous operation. With a large link frame length, the SR ARQ scheme is likely to require long numbering size for optimal frame length. The overhead in short frames is significant, which causes performance degradation.

Protocol synchronization is another important reliability constraint. For correct communication each end of the link maintains state variables which must be kept consistent containing the context information, e.g. association, message numbers, credits, delays etc. [21]. This requirement is called "Protocol Synchronization" [21]. In general the contexts evolve asynchronously due to transmission delays and response time in communication link. It is occasionally necessary to make sure that both contexts are simultaneously in a well-defined, e.g. at link initialization, resetting check-pointing, closing etc. [21]. Thus, during communication, inconsistencies of the context information

10
occur. We call the duration of such an inconsistency as the inconsistency gap ($I_g$). Unfortunately the inconsistent gap of the pos-ack based ARQ scheme will be unbounded if the transmission media is noisy. To see this, notice that in a pos-ack protocol an I-frame could be repeatedly corrupted, i.e. since the pos-ack based ARQ scheme is event-based, the sender can do nothing about an I-frame if the corresponding pos-ack does not arrive. Similarly, the receiver is locked if no new frames or control commands arrive. To overcome these problems, event-based protocols require some synchronization period to ensure the two ends are consistent, called the "resolving time period". Additionally "time-out" recovery is used to prevent the sender from waiting for an event indefinitely. An unexpected link failure will result in an inconsistency between the source and destination for a period of at least $I_g$ prior to the failure. At a failure, each end must correctly determine which I-frames should be kept and which ones may be removed to avoid duplicates or frame loss. Correctness is ensured by including check pointings during regular communication.

In the case of HDLC, the ends exchange RR control command (Check Point) every window size. Early in the window, both DLCs will have the consistent state variables (context information). As communication progresses they become inconsistent. In SREJ recovery of HDLC, if a SREJ is lost, the sender resends the corresponding frame after the timeout period has expired. However, the sender may not assume reception of that frame until its ACK is received. Should such an event occur repeatedly, the inconsistency gap of SR-HDLC would be unbounded. Thus in the worst case, a link failure may waste the total duration for a window. On the other hand, the periodic responses in LAMS-DLC guarantee that the inconsistency gap will not exceed the expected normal response time plus $C_{depth} I_{cp}$ about each frame, called as the maximum holding time of a frame.

3 LAMS-DLC

We now discuss LAMS-DLC, a DLC protocol designed for the LAMS network discussed in Section 2. There are several aspects to DLC protocol design, however, in this paper we consider error control and flow control only. In particular we examine the main features and elements of procedure.
3.1 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 acknowledgement, although it does 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.

3.2 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 Acknowledge- ment. That is, the protocol specifies that should the receiver detects 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. If the sender does not receive any response within a pre-defined duration (Cumulation Depth), the sender 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 it may lead to I-frame duplication if the link failure is not recoverable during the link lifetime. A more recent version of LAMS-DLC guarantees zero duplication as well as zero loss, however the analysis for this model has yet to be completed. 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. Error recovery of LAMS-DLC is divided into two different re-
covery methods like HDLC [20]: 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 acknowledgement. If there have been erroneous I-frames in the previous pre-defined 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 be retransmitted already. We call this recovery operation “Checkpoint 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 [20]. 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 can not 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 putting all the sequence numbers of previously erroneous I-frames during resolving period prior to the Request-NAK (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). 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 $C_{\text{depth}} \cdot W_{\text{cp}}$. 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 $C_{\text{depth}} \cdot W_{\text{cp}}$, 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 since $P_{C_{\text{depth}}} < \epsilon$ where $\epsilon$ is vanishingly small $^1$. Once the sender determines a link failure has occured it stops transmitting I-frames and informs the network layer.

3.3 Advantages

The most important advantage of Checkpoint Recovery is cumulative NAKs which guarantee no-loss of I-frames even though NAK-based protocols could lose I-frames due to a control command loss. Cumulative NAK also reduces the mean I-frame holding time of the sender since the increment of the I-frame holding time caused by a NAK loss, is just $W_{\text{cp}}$. In contrast, assuming SR

$^1$For example given a BER of $10^{-7}$, $P_{C_{\text{depth}}} \leq 10^{-10}$
ARQ scheme were used, the NAK (or ACK) loss would introduce a throughput loss since the increment of the holding time is at least a round trip time (as is the case of HDLC). During a burst error, many I-frames will be corrupted, and so too will the NAKs triggered by these erroneous I-frames (because of the duration of burst errors). In a LAMS network these burst error will be quite common and are likely to simulate link failure. Subnet nodes are therefore expected to resynchronize frequently resulting in poor line utilization. Again, with cumulative NAKs we avoid this performance degradation provided that \( C_{\text{depth}} \cdot W_{\text{cp}} > \bar{L}_{\text{burst}} \) where \( \bar{L}_{\text{burst}} \) is the burst error length.

Removal of the in-sequence constraint also offers additional flexibility and robustness to LAMS-DLC. Since LAMS-DLC permits out-of-sequence I-frames, the receiving buffer need not hold successful I-frames for resequencing. LAMS networks assumes that such link failures will not occur. Thus, the holding time of a frame in the receiving buffer is dependent only on the processing complexity of DLC. After processing the I-frame, the I-frame is moved to the sending buffer of the next hop. Therefore, the buffer problem associated with DLC is restricted to only the sending buffer. As a result, removing the in-sequence constraint allows DLC layer to operate continuously.

With respect to sequence numbering, LAMS-DLC permits a bounded number size. Recall from Section 2.3 that the numbering size is equal to \( H_{\text{frame}}/\bar{L}_{\text{frame}} \). In LAMS-DLC, \( H_{\text{frame}} \) can be substituted into the Resolving Period which is bounded by

\[
R + \frac{1}{2} W_{\text{cp}} + C_{\text{depth}} W_{\text{cp}}
\]

This is permissible since, if there is no response after the resolving period for the first transmission of an I-frame has passed, the sender stops sending I-frames, otherwise LAMS-DLC assigns a new sequence number to the I-frame at retransmission. Thus, the numbering size guarantees to identify any unacknowledged I-frame uniquely. In this context, the inconsistency gap of LAMS-DLC is also bounded to the resolving period. In other words, when an unexpected unrecoverable link failure occurs, the sender/receiver
can locate the range of suspicious I-frames and can recover I-frames without loss or duplication. In this paper we do not deal with this issues of DLC.

3.4 Flow Control

In LAMS-DLC, the receiving buffer is assumed uncongested because the transparent size for the receiving buffer is bounded. In a homogeneous network, however, flow control is required owing to the congestion of the sending buffer which could be filled by frames sent from higher layers. To inhibit such internal incoming frames may give rise to dead-lock. Therefore, it is necessary to provide a mechanism to block external incoming frames. In LAMS-DLC, we propose a simple flow control mechanism. While overflow in the receiving buffer is anticipated by the receiver, the receiver sets the Stop-Go-bit of Check-Point Commands to 1. However, if the receiver judges itself as having sufficient buffer space to accept all the incoming I-frames, it simply sets Stop-Go-bit to 0. Whenever the sender receives a Check-Point command with Stop-Go-bit set to 1, the sender decreases the sending rate of I-frames by some predefined value. If the sender keeps detecting Stop-Go-bit set to 1 during a predefined time, the sender repeatedly decreases the sending rate of I-frames. However, if the sender detects Stop-Go-bit set to 0, the sender increases the sending rate. If necessary, the receiver discards the overflowing I-frames while sending control with the Stop-Go-bit set to 1. The objective of this mechanism is to minimize the losses due congestion, and the wasted time due to delayed responses. While this algorithm is sufficient for our purpose. In the analysis of LAMS-DLC, the impact of flow control will not be considered since the frequency at which this mechanism is invoked, is expected to be low. The transparent size of the receiving buffer allows most channel traffic to be accommodated.

LAMS-DLC also allows the sending buffer to be controlled. Recall from Section 2.3 that in LAMS-DLC, the maximum holding time of a I-frame in the sender buffer is bounded. This means that the sending buffer has a transparent size and is controlled efficiently. If we decrease the check point interval, that holding time will be decreased. In the other word, the sending buffer is under control. So LAMS-DLC is said to have the buffer control as well as the flow control. Although buffer control and flow control are closely related they are actually distinct in terms of their objectives. Flow
control is typically associated with the receiving buffer of the receiver. That is, flow control is used to prevent congestion of the receiving buffer. Buffer control in LAMS-DLC, however, is associated with the sending buffer of the sender. Buffer control attempts to minimize the holding time of the sender to reduce the buffer requirement. Traditional DLC protocols prevent the sender from transmitting I-frames until the flow control command (e.g. RR in HDLC) indicates it may do so. However, buffer control of LAMS-DLC does not affect the sender’s sending rate, thus the sender may transmit all new I-frames in its sending buffer while the link is available. This mechanism works because the receiving buffer does not hold successfully arrived I-frames. That is, from the sender’s perspective the receiving buffer is infinite, while from the receiver’s perspective a receiving buffer is unnecessary. Recall that each node of LAMS-DLC is assumed to have an identical communication architecture with respect to hardware/software. Thus the sender is aware of the receiver’s communication limitations. Provided the receiving buffer requirement is transparent to the DLC’s operation we may safely assume that the receiving buffer size is infinite or unnecessary. In LAMS-DLC the receiving buffer size is transparent. See Section 4 for details.

4 Performance Evaluation

In this section we evaluate the performance of LAMS-DLC. In particular we consider its effective throughput (throughput efficiency) and compare LAMS-DLC with SR-HDLC. In ARQ schemes, a retransmission increases the amount of time required for the successful delivery of an I-frame by at least as much as a round trip. If the retransmission is also corrupted, another retransmission will occur with the same delay, and so on. In ARQ schemes, therefore, there will be consecutive retransmissions associated with successful transmissions of each I-frames in the sending buffer, this is called the retransmission tail [24]. Thus, ARQ schemes show an unavoidable delay due to this retransmission tail. If we assume that no I-frame has arrived between the beginning of an I-frame transmission and the end of its retransmission tail, the total time period for successful I-frame delivery can be divided into two distinct periods: the transmission period for the initial transmission of the I-frame in the sending buffer; and the retransmission period for retransmissions of the corrupted I-frame. In the case of SR-HDLC, these two
distinct periods are repeated every time the window is exhausted. However, LAMS-DLC permits the transmission period of another group of I-frames to be overlapped with the retransmission period of the current group. Specifically, LAMS-DLC can send frames during the idle time period in SR-HDLC provided I-frames are available. This makes it difficult to compare the two DLC protocols using retransmission tails, however, since this length describes the performance of a DLC protocol we will attempt to accommodate these differences.

We need to clearly define these two distinct periods. The transmission period is defined as the time period from the start of the transmission of a new I-frames until the valid ACKs (or NAKs) arrive and the sender has finished processing it. Thus this transmission period may or may not contain retransmissions of erroneous I-frames. However, to define the succeeding period disjointly, the succeeding period may or may not contain the frame transmission time. We assume that each retransmission period is associated with only single I-frame. The retransmission period is defined as the time period from the end of the preceding period to the moment that either single I-frame is resolved or the sender decides to retransmit the I-frame. These definitions divide the total time period into two disjoint periods.

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:

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

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

So

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

On the other hand, the case with SR-HDLC is slightly different. First, SR-HDLC is based on both positive ack and negative acknowledgement ARQ. If an explicit positive ACK is lost or corrupted, the relevent I-frame must be retransmitted after its timeout has expired. Therefore, the sender must keep the erroneus I-frame in its buffer until that frame is delivered successfully. To inform the sender of new credits, the receiver must send an RR command after all I-frames have successfully arrived. This RR command is therefore the final positive acknowledgement for that window. In each retransmission period, the last retransmitted I-frame contains RR($p$) which means that the sender asks the receiver to send RR if the receiver received all the I-frames successfully. So, the loss of this RR command gives rise to another retransmission period. Again, this will adversely affect the performance of SR-HDLC in a LAMS network. These two cases are described as follows:
In a transmission period,

\[ PT_{R}^{HDL} = P_{R} \text{ of a I-frame in a transmission period} \]
\[ = \text{Prob[I-frame has an error] } \times \text{Prob[NAK has no error]} + \]
\[ \text{Prob[either ACK or NAK is lost]} \]
\[ = P_{F}(1 - P_{C}) + P_{C} \]
\[ = P_{F} + P_{C} - P_{F}P_{C} \]

In a retransmission period,

\[ PR_{R}^{HDL} = P_{R} \text{ of a retransmitted I-frame in a retransmission period} \]
\[ = \text{Prob[I-frame has an error]} + \]
\[ = \text{Prob[RRI is lost]} \times \text{Prob[I-frame has no error]} \]
\[ = P_{F} + P_{C}(1 - P_{F}) \]
\[ = P_{F} + P_{C} - P_{F}P_{C} \]

So

\[ \bar{s}_{HDL} = \frac{1}{1 - (P_{F} + P_{C} - P_{F}P_{C})} \]

The mean number \( \bar{s} \) of the periods for the successful transmission of an I-frame in LAMS-DLC and HDLC are summarized as follows:

\[ \bar{s}_{LAMS} = \frac{1}{1 - P_{F}} \]
\[ \bar{s}_{HDL} = \frac{1}{1 - (P_{F} + P_{C} - P_{F}P_{C})} \]

We now calculate the mean length of the transmission period and the retransmission period for each protocol, denoted by \( D_{\text{trans}} \) and \( D_{\text{retrn}} \) respectively.

We make the following assumptions to simplify the derivation: every I-frame is of the same size; \( N < W \); and the sender receives no 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

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

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 acknowledgement 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 \( D_{return} \). 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 \( I_{cp}/2 \). 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 \((\bar{n}_{cp} - 1) \cdot I_{cp}\).

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

21
One way propagation time \((R/2)\) +  
Transmission time of the cp-command \((t_c)\) +  
Processing time of the cp-command \((t_{proc})\)  

\[ = N \cdot t_f + t_c + t_{proc} + R + (\bar{n}_{cp} - \frac{1}{2}) \cdot I_{cp} \]

LAMS-DLC will typically show a slightly longer transmission period than the selective repeat ARQ without a delayed response. Let us consider 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. Recall the assumption that each retransmission period has only one I-frame transmission. Even if the cp-command reports errors it is reasonable to assume that the number of associated I-frames is on average one. In the case of LAMS-DLC, \(D_{retrn}\) is equivalent to \(D_{trans}\) except for the transmission time.

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

\[ = t_f + t_c + t_{proc} + R + (\bar{n}_{cp} - \frac{1}{2}) \cdot I_{cp} \]

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

\[ D_{low}^{LAMS}(N) = D_{trans}^{LAMS}(N) + (\bar{s}_{LAMS} - 1)D_{retrn}^{LAMS} \]

\[ = (N + \bar{s}_{LAMS} - 1)t_f + \bar{s}_{LAMS}(R + t_c + t_{proc}) + \bar{s}(\bar{n}_{cp} - \frac{1}{2})I_{cp} \]

22
\[ \approx N t_f + s_{\text{LAMS}} R + s_{\text{LAMS}} (\bar{n}_{cp} - \frac{1}{2}) I_{cp} \]

Analyzing SR-HDLC, we use the same assumptions regarding uniformity and retransmission period. Before the analysis we briefly explain the error recovery function of HDLC: First, SREJ (the selective negative ack) is used only for I-frames transmitted in the transmission period. Thus, both positive and negative acknowledgement are used only in the transmission period. Second, error recovery in the retransmission period is based on timeout recovery. Thus, if the sender does not receive an RR command during a retransmission period it sends the relevant I-frames again once the timeout has expired. In the final retransmission period the sender will receive the RR command. We call this final retransmission period the Resolve period. In the case of SR-HDLC, the behavior of the retransmission period is expected to differ from that of the transmission period. To express these different behaviors, we define the following terminologies. Transmission delay is the mean duration between the end of transmission of the I-frames in a transmission period and the start of the succeeding retransmission period, this is denoted by \( d_{\text{trans}} \). Retransmission delay is likewise defined except that it applies during the retransmission period, and is denoted by \( d_{\text{retrn}} \). Resolve delay is the same as the retransmission delay except that it is applied to the last period only. These two delays are expressed as follows:

\[
\begin{align*}
  d_{\text{trans}} & = P_C t_{\text{out}} + (1 - P_C) (R + 2t_{\text{proc}} + t_c) \\
  d_{\text{retrn}} & = t_{\text{out}} \\
  d_{\text{resol}} & = R + 2t_{\text{proc}} + t_c
\end{align*}
\]

These represent the delays that the sender suffers after transmitting in each period. In general, \( t_{\text{out}} \) should be much greater than max round trip time in a dynamically changing network. In [] it is suggested that \( t_{\text{out}} = R_t + n \sqrt{\text{var}(R_t)} \) where \( R_t \) is distance as a function of time and \( n \) is a constant of 1 or 2. In a LAMS network, \( \text{var}(R_t) \) is likely to be large because of high mobility. Therefore we use \( t_{\text{out}} = R + \alpha \), where \( R \) is the mean round trip time on \( R_t \) and \( \alpha \) is constant. If \( R_{\text{min}} \leq R_t \leq R_{\text{max}} \) where \( R_{\text{min}} \) and \( R_{\text{max}} \) are the min/max distance, then \( R = \frac{1}{2} (R_{\text{min}} + R_{\text{max}}) \) and \( t_{\text{out}} \geq R_{\text{max}} \). Thus \( \alpha \geq R_{\text{max}} - R \).
Let us calculate the mean lengths of transmission, retransmission and resolve period denoted by $D_{trans}^{HDLC}$, $D_{retrn}^{HDLC}$ and $D_{resol}^{HDLC}$ respectively.

$$D_{trans}^{HDLC}(W) = \text{Transmission time of } W \text{ I-frames}(t_f W) +$$

Transmission delay($d_{trans}$)

$$= W \cdot t_f + (1 - P_C) (R + 2t_{proc} + t_c) + P_C (R + \alpha)$$

$$D_{retrn}^{HDLC} = \text{Transmission time of one I-frame}(t_f) +$$

Prob[This period is Resolve one] $\times$ Resolve delay($d_{resol}$) +

Prob[This period is not Resolve one] $\times$ Retransmission delay($d_{retrn}$)

$$= t_f + (1 - P_F) (1 - P_C) d_{resol} + (1 - (1 - P_F)(1 - P_C)) d_{retrn}$$

$$= t_f + R + \alpha (1 - P_F - P_C + P_F P_C) + (P_F + P_C - P_F P_C) (2t_{proc} + t_c)$$

Thus, the total mean time for the safe delivery of $W$ I-frames in SR-HDLC is:

$$D_{low}^{HDLC}(W) = D_{trans}^{HDLC}(W) + (\bar{s}_{HDLC} - 1) D_{retrn}^{HDLC}$$

$$= (W + \bar{s}_{HDLC} - 1) t_f + \bar{s}_{HDLC} R +$$

$$((\bar{s}_{HDLC} - 1) (1 - P_F - P_C + P_F P_C) - P_C) \alpha +$$

$$((\bar{s}_{HDLC} - 1) (P_F + P_C - P_F P_C) + 1 - P_C) (2t_{proc} + t_c)$$

$$\approx W t_f + \bar{s}_{HDLC} R + ((\bar{s}_{HDLC} - 1) (1 - P_F - P_C + P_F P_C) - P_C) \alpha$$

$D_{low}^{HDLC}(W)$ is the mean time needed for the delivery of I-frames in a window of size $W$. So in low traffic given an infinite buffer, the mean total times of SR-HDLC and LAMS-DLC for $N$ I-frames are:

$$D_{low}^{LAMS}(N) \approx N t_f + \bar{s}_{LAMS} R + \bar{s}_{LAMS} (\bar{n}_{cp} - \frac{1}{2}) I_{cp}$$

$$D_{low}^{HDLC}(N) \approx N t_f + \bar{s}_{HDLC} R + ((\bar{s}_{HDLC} - 1) (1 - P_F - P_C + P_F P_C) - P_C) \alpha$$

Now 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 $\alpha$ is small. However, it is likely that $\alpha \gg \bar{n}_{cp}$ in a highly changing network and $\bar{s}_{HDLC} > \bar{s}_{LAMS}$ in high error environment. Therefore we expect that $D_{low}^{HDLC}(N)$ will be greater than $D_{low}^{LAMS}(N)$ in a LAMS network even if we assume 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 for the removal rate is clearly $1/t_f$. 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.

To compare LAMS-DLC and SR-HDLC we shall compute their transparent buffer sizes. From the perspective of a DLC ther 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 the receiving buffer of the receiver need be considered. Behavior of the receiving/sending buffer in the sender/receiver respectively are not determined by the communicating DLC protocols but rather by the activities of other links in a node. We propose the following model for analysis of buffer size. The sending buffer of the sender is filled by the network layer. While flow control limits incoming frames over a link in low traffic a disparity between incoming and outgoing frames may give rise to the same situation as high traffic to the DLC protocol. For example, if the incoming frame rate is much greater than the outgoing frame rate and most of incoming frames are destined for the link, the average rate coming into the sending buffer associated with the link can easily exceed $1/t_f$. Therefore it is not inpractical that we assume the incoming rate into the sending buffer is always $1/t_f$. In this model, the receiver sends eligible I-frames to the network layer as soon as they are resolved.

Let us evaluate the transparent buffer size for LAMS-DLC. 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 \( D_{low}^{LAMS}(N) \) the mean total time for \( N \) frames. Thus we use a different approach to derive the mean holding time of the sending buffer.

Let

\[
H_{\text{frame}}^{LAMS} = \text{The mean holding time of a I-frame}
\]

\[
H_{\text{succ}}^{LAMS} = \text{The mean holding time of a successful I-frame}
\]

\[
H_{\text{fail}}^{LAMS} = \text{The mean holding time of an erroneous I-frame.}
\]

We can express \( H_{\text{frame}}^{LAMS} \) as follows:

\[
H_{\text{frame}}^{LAMS} = (1 - P_F) \cdot H_{\text{succ}}^{LAMS} + P_F \cdot H_{\text{fail}}^{LAMS}
\]

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

\[
H_{\text{succ}}^{LAMS} = D_{\text{trans}}^{LAMS}(1)
\]

\[
= R + t_f + t_c + t_{\text{proc}} + (\bar{n}_{cp} - \frac{1}{2})I_{cp}
\]

In the case of \( H_{\text{fail}}^{LAMS} \), we can express it with \( H_{\text{frame}}^{LAMS} \) and \( H_{\text{succ}}^{LAMS} \) recursively as follows: an erroneous I-frame is first retransmitted \( D_{\text{trans}}^{LAMS}(1) \) after transmission of the I-frame. After the sender retransmits that I-frame, the mean holding time of the new I-frame is again \( H_{\text{frame}}^{LAMS} \) 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:

\[
H_{\text{fail}}^{LAMS} = H_{\text{succ}}^{LAMS} + H_{\text{frame}}^{LAMS}
\]

\[
= R + t_f + t_c + t_{\text{proc}} + (\bar{n}_{cp} + \frac{1}{2})I_{cp} + H_{\text{frame}}^{LAMS}
\]

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

\[
H_{\text{frame}}^{LAMS} = (1 - P_F) \cdot H_{\text{succ}}^{LAMS} + P_F \cdot H_{\text{fail}}^{LAMS}
\]
So

\[
H_{frame}^{LAMS} = \frac{1}{1 - P_F} \cdot H_{nuoc}^{LAMS} = \frac{1}{1 - P_F} \cdot (R + t_f + t_c + t_{proc} + (\bar{n}_{cp} - \frac{1}{2}) I_{cp}) \approx \bar{s}_{LAMS} \cdot (R + (\bar{n}_{cp} - \frac{1}{2}) I_{cp})
\]

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. The destination node has the responsibility to solve duplication and ordering of incoming I-frames. 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 \( H_{frame}^{LAMS} \) because all frames are unresolved and these frames must be kept in the sending buffer. But after time period \( H_{frame}^{LAMS} \), the removed rate from the sending buffer will to be approximately \( 1/t_f \). The the sending buffer is now stable in the deterministic model (although in the M/M/1 queuing 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 \( H_{frame}^{LAMS} \). So, the mean buffer size \( B_{LAMS} \) for transparent operation of LAMS-DLC, (i.e. the mean buffer size) is:

\[
B_{LAMS} = \text{size}(\text{the sending buffer}) + \text{size}(\text{the receiving buffer}) = \frac{1}{t_f} H_{frame}^{LAMS} + \frac{t_{proc}}{t_f} = \frac{1}{t_f} \bar{s}_{LAMS} \cdot (R + t_f + t_c + t_{proc} + (\bar{n}_{cp} - \frac{1}{2}) I_{cp}) + \frac{t_{proc}}{t_f} \approx \frac{1}{t_f} \bar{s}_{LAMS} \cdot (R + (\bar{n}_{cp} - \frac{1}{2}) I_{cp})
\]

In the case of SR-HDLC, the mean holding time of the sender can be calculated the same way as LAMS-DLC, however the transparent buffer size
for SR-HDLC is computed differently. In fact, we note there is no transparent buffer size for SR-HDLC. Consider, regardless of the window size of SR-HDLC, a time period for resolving that window is always needed. During this period frames can not be sent by the sender. Assume that new frames arrive in the sending buffer at the same rate as frames depart. When the sender has consumed the window and is waiting for a RR command from the receiver, new frames will continue to arrive at the sending buffer. So, the sending buffer must hold these new frames until the next window is active, i.e. during $D_{low}^{HDL}$ $W$ frames are removed but $D_{low}^{HDL}/t_f + W$ arrived. If the same situation occurs in the next window the sending buffer will be still longer. In fact the sending buffer increases continuously. As a result, we observe that there is no transparent sending buffer size in SR-HDLC. On the other hand the receiver must maintain a buffer at least the same size as the window since the receiver can not send even valid frames to the packet layer until all preceding frames become valid.

In very high traffic, the buffer sizes for continuous operations of SR-HDLC and LAMS-DLC are:

$$B_{LAMS} \approx \frac{1}{t_f}^{\tilde{\bar{n}}}_{LAMS}(R + \hat{n}_{cp} - \frac{1}{2}I_{cp})$$

$$B_{HDLC} = \infty$$

In SR-HDLC, the window size is related to the frame sequence number which is also associated with the frame length, the window size is limited to $W \approx M/2$ for $M = 2^l$ where $l$ is the maximum bit length of sequence number field. We note that usually in SR-HDLC the transmission period and retransmission period are distinct. In the case of LAMS-DLC, however, erroneous I-frames will be retransmitted along with I-frame transmission. SR-HDLC is able to do this only by assuming a large buffer. Also, in high traffic the number of retransmissions will increase transmission period in the following analysis. We include this observation in this analysis. After link setup in LAMS-DLC, only new I-frames are transmitted. After $H_{frame}^{LAMS}$, new and retransmitted I-frames will co-exist. Let $N_{total}(N)$ be the total number of I-frames sent during a transmission period for $N$ new frames, including the retransmitted I-frames, and $h = H_{frame}^{LAMS}/t_f$, which means the length of
the mean holding time in frame. We can divide a transmission period into multiple subperiods which correspond to the mean holding times plus the remaining period. Let \( n = \lfloor \frac{N_{\text{total}}}{h} \rfloor \) and \( r = N_{\text{total}} \mod h \). Also let \( N_i \) be the number of new I-frames in each subperiod for \( 1 \leq i \leq n + 1 \).

\[
\begin{align*}
N_1 &= h \\
N_2 + N_1 P_R &= h \\
N_3 + N_2 P_R + N_1 P_R &= h \\
&\quad \ldots \\
N_n + N_{n-1} P_R + \cdots + N_1 P_R^{n-1} &= h \\
N_{n+1} + r P_R + r P_R^2 + \cdots + r P_R^n &= r
\end{align*}
\]

where \( \sum_{i=1}^{n+1} N_i = N \).

So

\[
N_{\text{total}}(N) = N_1 + (N_2 + N_1 P_R) + (N_3 + N_2 P_R + N_1 P_R^2) + \cdots + (N_n + N_{n-1} P_R + \cdots + N_1 P_R^{n-1}) + r
\]

\[
= \sum_{i=1}^{n+1} \sum_{j=1}^{i+1} N_j P_R^{n-i} + r
\]

We can approximate \( n \) by \( \lfloor \frac{N}{h} \rfloor \) and \( r \) by \( N \mod h \). In SR-HDLC, each sub-period is the expected normal response time \( (d_{\text{normal}} = R + t_f + 2t_{\text{proc}} + t_o) \). Therefore, in high traffic \( N > h \) in LAMS-DLC while \( N > W \) and \( W t_f > d_{\text{normal}} \) in SR-HDLC. If we let \( m = \lfloor \frac{N}{W} \rfloor \), \( r_w = N \mod W \), \( N_{\text{win}}^{\text{HDLC}} = N_{\text{total}}(W) \) and \( N_{\text{LAMS}}^{\text{HDLC}} = N_{\text{total}}(N) \), the mean total times of SR-HDLC and LAMS-DLC required for successful transmission of \( N \) frames are:

\[
\begin{align*}
D_{\text{high}}^{\text{HDLC}}(N) &= m D_{\text{low}}^{\text{HDLC}}(N_{\text{win}}^{\text{HDLC}}) + D_{\text{low}}^{\text{HDLC}}(r_w) \\
D_{\text{high}}^{\text{LAMS}}(N) &= D_{\text{low}}^{\text{LAMS}}(N_{\text{LAMS}}^{\text{LAMS}})
\end{align*}
\]

As a consequence, in high traffic, LAMS-DLC shows the following throughput efficiency \( \eta_{\text{LAMS}} \) with the transparent buffer size \( B_{\text{LAMS}} \):

\[
\eta_{\text{LAMS}} = \frac{N}{D_{\text{high}}^{\text{LAMS}}(N)}
\]

29
\[ = \frac{N}{N_{\text{total}} t_f + \bar{s}_{\text{LAMS}} R + \delta_{\text{LAMS}}} \]

where \( B_{\text{LAMS}} = \frac{1}{I_f} \bar{s}_{\text{LAMS}} \left( R + \left(\bar{n}_{cp} - \frac{1}{2}\right) I_{cp} \right) \) and \( \delta_{\text{LAMS}} = \bar{s}_{\text{LAMS}} \left( \bar{n}_{cp} - \frac{1}{2} \right) I_{cp} \)

In SR-HDLC, if \( W = B_{\text{LAMS}} \) and \( N_{\text{HDLC}} = m N_{\text{win}} + \tau_w \), the throughput efficiency \( \eta_{\text{HDLC}} \) with the buffer size \( B_{\text{HDLC}} = 2B_{\text{LAMS}} \) is:

\[
\eta_{\text{HDLC}} = \frac{N}{D_{\text{high}}^{\text{HDLC}}(N)}
\]

\[
= \frac{N}{m D_{\text{low}}^{\text{HDLC}}(N_{\text{total}}^{\text{HDLC}}) + D_{\text{low}}^{\text{HDLC}}(\tau_w)}
\]

\[
= \frac{N}{N_{\text{total}}^{\text{HDLC}} t_f + (m + 1) \bar{s}_{\text{HDLC}} R + (m + 1) \delta_{\text{HDLC}}}
\]

where \( \delta_{\text{HDLC}} = \left( \left( \bar{s}_{\text{HDLC}} - 1 \right) \left( 1 - P_F - P_C + P_F P_C \right) - P_C \right) \alpha \) and \( \alpha = t_{\text{out}} - R \).

As we see from the above two equations as the channel traffic increases, the throughput efficiency of LAMS-DLC will be much better that that of SR-HDLC. Provided that we provide the transparent buffer size for LAMS-DLC, LAMS-DLC will almost show the increasing throughput efficiency as the channel traffic \( (N) \) increases, because \( \frac{N_{\text{LAMS}} R + \delta_{\text{LAMS}}}{N} \) is likely to be decreased. In low channel traffic, the round trip time \( (R) \) of HDLC is significantly larger in LAMS networks. Therefore, HDLC is expected to have large buffer size and window size.
References


design. Joint sponsorship, COMSAT Laboratories, Clarkburg, Maryland 20871, 1989.


