Expected File-Delivery Time of Deferred NAK ARQ in CCSDS File-Delivery Protocol

Daniel C. Lee, *Member, IEEE,* and Wonseok Baek, *Student Member, IEEE*

***Abstract*—We analyze an automatic repeat-request (ARQ) scheme of the Consultative Committee for Space Data Systems file-delivery protocol for the single-hop file-transfer operation. With regard to performance measures, this paper is mainly concerned with the time taken to transfer a file (file-delivery time) and throughput efficiency. We discuss the ARQ timer-setting rule that minimizes the expected file-delivery time under the constraint that the throughput efficiency is maximized. Then, for that timer-setting rule, we derive the expected file-delivery time.**

***Index Terms*—Automatic repeat-request (ARQ), delay, internetworking, protocol, throughput.**

I. Introduction

In RECENT years, the Consultative Committee for Space Data Systems (CCSDS) has made considerable efforts to provide flexible and efficient transfer of various types of data in a wide variety of mission configurations, from relatively low Earth-orbiting spacecraft to complex arrangements of deep-space orbiters and landers supported by multiple transmission links. In many mission scenarios, space networking faces extremely long propagation delays, intermittent link connectivity, limited bandwidth, and limited power budgets. In response to these factors and the need to automate the communication among spacecraft, the CCSDS File-Delivery Protocol (CFDP) has been developed [1]–[3]. The aforementioned mission environments make the conventional automatic repeat-request (ARQ) schemes impractical [4]. The most salient feature of ARQ schemes used in the CFDP, in comparison with conventional ARQ schemes, is that an acknowledgment (ACK) is not issued for most protocol data units (PDUs). For those PDUs, only negative acknowledgment (NAKs) are issued, which happens if the receiver perceives an anomaly in PDU delivery. ACKs are only used for ancillary data PDUs such as end-of-file (EOF) and finished (FIN) PDUs, which are used for closing the file-transfer operation.

In the CFDP, the file transfer is called a "transaction," and the sender assigns a transaction ID for each file-transfer operation. The transaction ID, along with the source ID and other information, is contained in the header of each PDU. The sender informs the receiver of the start of the file transfer by transmitting the meta-data PDU, which contains information such as the source and destination IDs, the file name, the file size, etc. Like most PDUs in the CFDP, there is no ACK for the meta-data PDU, and the sender is allowed to transmit file-data PDUs (PDUs carrying the actual content of the file) after transmitting the meta-data PDU. In other words, there is no handshaking for initiating a "transaction." The receiver detects the failure in delivering a file-data PDU or the meta-data PDU by noticing missing elements in the sequence of PDUs correctly received. Each file-data PDU has a field that specifies the starting byte number and ending byte number of the file data carried by the PDU, so the receiver can detect missing PDUs by observing the ending byte numbers and the starting byte numbers of the correctly received PDUs. If the meta-data PDU is lost in the first trial, the receiver will detect that the meta-data PDU is missing, because the new transaction ID in the header of that received PDU will indicate that the new transaction has begun. The receiver reacts to the missing PDU by sending NAK messages. Each NAK message contains the list of PDUs requested by the receiver for retransmission. Upon receiving a NAK, the sender retransmits the PDUs requested. When the sender runs out of the file-data PDUs to send, the sender sends an EOF PDU, thus initiating the closure of the file transfer.

After receiving the EOF PDU, the receiver acknowledges it with an ACK(EOF) and waits until the meta-data PDU and all of the file-data PDUs are received before it initiates the closure of the transaction. All data are eventually received because of the NAK mechanisms, and the receiver can notice the reception of all data from the file-size information contained in the meta-data PDU and the EOF PDU. Then, the receiver sends a FIN PDU. After receiving the FIN PDU, the sender acknowledges it with an ACK(FIN) and closes the transaction. When the ACK(FIN) is successfully delivered back to the receiver, the receiver also closes the transaction, at which point, the transaction is closed at both entities. For EOF and FIN PDUs, there are ACKs and retransmission timer mechanisms, so their exchange is reliable. According to the CFDP, the receiver and sender must both transmit an ACK message in response to each EOF/FIN PDU, even after closing the transaction, in order to prevent possible anomalies in closing the transaction (e.g., the one described in [5]).

Depending upon mission requirements and transmission capability, four selectable ARQ schemes are offered by the CFDP. These schemes (immediate NAK, deferred NAK, asynchronous NAK, and prompt NAK modes) share a common mechanism for initiating and closing the file-transfer operation, but they differ

Paper approved by Mounir Hamdi, the Editor for Network Architecture of the IEEE Communications Society. Manuscript received November 1, 2002; revised November 16, 2003. This work was supported by the Jet Propulsion Laboratory, Pasadena, CA, under Contract 1214036.

The authors are with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089 USA (e-mail: dclee@usc.edu).

Digital Object Identifier 10.1109/TCOMM.2004.833017

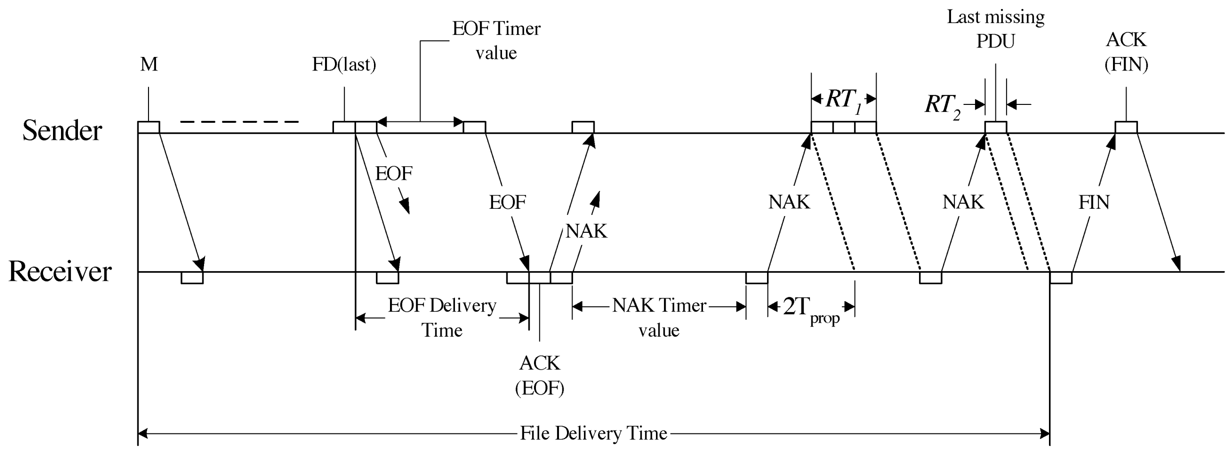

Fig. 1. Deferred NAK mode. M stands for the meta-data PDU, and $\mathrm{FD}(k)$ stands for the kth file-data PDU. The protocol definition in [3] defines the meaning and format of these PDUs in detail. T_{prop} stands for one-way propagation delay, and RT_k stands for the duration of the kth retransmission spurt. ("Transmission spurt" refers to consecutive transmission of PDUs back to back.).

in their times of issuing NAK messages and in their lists of PDUs requested for retransmission. In this paper, we consider only the deferred NAK mode.

In the deferred NAK mode, the receiver defers issuance of NAKs until it correctly receives the EOF PDU from the sender. The receiver keeps the record of all missing PDUs until the EOF PDU is successfully delivered. After receiving the EOF PDU, the receiver issues an ACK(EOF) and issues a NAK that requests retransmission of all missing PDUs, if any. Upon receiving a NAK, the sender immediately retransmits all PDUs that the NAK requests. At the end of each transmission of a NAK, the receiver sets a NAK timer, and when the NAK timer expires, the receiver again examines the record of missing PDUs. If missing PDUs still remain, the receiver issues another NAK and again starts a NAK timer. This process continues until the receiver receives all necessary PDUs that contain the whole file content and the meta-data PDU. After receiving all necessary PDUs, the receiver issues a FIN PDU, and upon receiving the FIN PDU, the sender issues an ACK(FIN) and closes the transaction. The delivery of the FIN PDU is guaranteed in the same way as the EOF PDU. The receiver closes the transaction when the ACK(FIN) is successfully delivered to the receiver. See Fig. 1 for an illustration of the operation of the deferred NAK mode.

In this paper, we present modeling and analysis of the CFDP deferred NAK mode. We consider the single-hop file-transfer operation. With regard to performance measures, the paper is mainly concerned with the time taken to transfer a file (expected file-delivery time) and throughput efficiency resulting from the protocol specification. In Section II, we discuss the ARQ timer-setting rule that minimizes the expected file-delivery time under the constraint that the throughput efficiency is maximized. Then, for that timer-setting rule, we derive an expression for the expected file-delivery time and discuss how to compute it numerically. In Section III, on the basis of our derivation, we numerically present how the constrained optimal expected file-delivery time varies with parameters such as the channel quality, PDU size, file size, etc.

II. Analysis of Deferred NAK Mode

A. Preliminaries

We define the "file-delivery time" to be the time from the beginning of the transmission (first bit of the meta-data PDU) until the first instant when all file data, meta data, and the EOF PDU have been successfully received by the receiver. We also define "EOF delivery time" as the length of the time interval between the instant immediately after the sender's transmission of the last file-data PDU and the instant of the receiver's receiving the last bit of error-free EOF PDU. As illustrated in Fig. 1, the file-delivery time consists of a number of transmission spurts (initial transmission of a file and several retransmission spurts after successful transmission of the EOF PDU), EOF delivery time, and time gaps between transmission spurts. "Transmission spurt" refers to consecutive transmissions of PDUs back to back. Note that our definition of file-delivery time does not include the time for the FIN-ACK(FIN) procedure. The reason that we define file-delivery time in this way is that the whole file is obtained by the receiver as soon as all file data, meta data, and the EOF PDU have been successfully received by the receiver. If one is interested in the duration of the entire transaction, the time to deliver the ACK(EOF) and the FIN-ACK(FIN) must to be included along with the file-delivery time.

We denote by N the total number of PDUs carrying the file data plus 1 (counting the meta data). For simplicity, we assume that these N PDUs have an identical length and an identical probability of failed delivery (PDU error or loss). We also assume that all NAKs have an identical length and an identical probability of failed delivery. Based on these assumptions, we derive the expected file-delivery time. For the analysis, we assume that PDU error events in forward and backward links are statistically independent, and that the transmission times of the file-data PDUs and the meta-data PDU are equal. Notations we use are specified in Table I.

As can be deduced from the protocol description in Section I (illustrated in Fig. 1), the expected file-delivery time depends upon certain parameter values that an implementer can freely

TABLE I
NOTATIONS

Symbol	Definition
P_{ef}	Prob. of PDU error in forward link
$P_{ef(EOF)}$	Prob. of error in delivering EOF PDU
P_{er}	Prob. of error in delivering NAK
T_{prop}	One-way propagation delay
T_{PDU}	Transmission time of meta data or file data PDU
T_{NAK}	Transmission time of NAK PDU
T_{EOF}	Transmission time of EOF PDU
$T_{ACK(EOF)}$	Transmission time of ACK(EOF) PDU

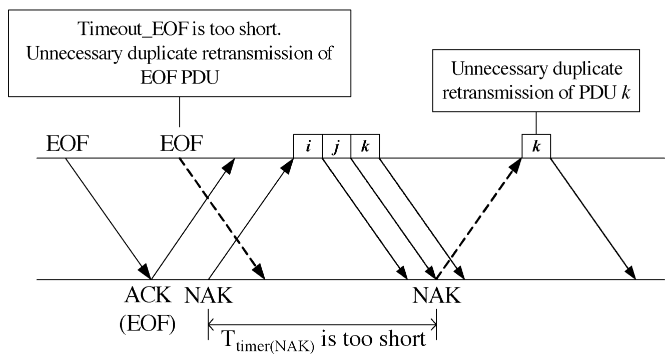

Fig. 2. Effect of timeout values on CFDP performance.

choose; for example, the EOF timer and the NAK timer. We assume that the parameter values are set to minimize the expected file-delivery time under the constraint that the throughput efficiency is never compromised. Under the environment of the long propagation delay, the throughput efficiency can be compromised in the form of unnecessary duplicate retransmission of an identical PDU. For example, if the timeout value of the EOF timer, which we refer to as timeout_EOF, is set too small, the sender retransmits the EOF PDU before receiving the ACK(EOF) because of timer expiration, even in the case in which the first EOF PDU and the ACK(EOF) are successfully delivered. Unnecessary duplicate retransmission of the file-data PDU can occur if the timeout value of the NAK timer is set too small, as illustrated in Fig. 2. In order to prevent unnecessary duplicate retransmission, and to minimize the expected file-delivery time, the timeout value of the EOF timer, timeout_EOF, should be $2T_{\text{prop}} + T_{\text{ACK(EOF)}}$, where T_{prop} denotes the one-way propagation delay between the sender and the receiver. Let RT_k denote the transmission time of the PDUs requested by the receiver for the kth retransmission spurt. (Fig. 1 illustrates RT_1 and RT_2.) Then, the timeout value of the NAK timer for the kth retransmission spurt, $T^k_{\text{timer(NAK)}}$, should be $2T_{\text{prop}} + RT_k$.

B. Expected Value of EOF Delivery Time

Denoting by G_{EOF} the geometrically distributed random variable that counts the number of EOF PDU transmissions up to and including the first successful delivery, we can express the EOF delivery time as

$$(G_{\text{EOF}} - 1)(T_{\text{EOF}} + \text{timeout\_EOF}) + (T_{\text{prop}} + T_{\text{EOF}})$$

and its expected value as

$$\begin{aligned}\mathbb{E}\left[(G_{\text{EOF}} - 1)\right](T_{\text{EOF}} + \text{timeout\_EOF}) + T_{\text{prop}} + T_{\text{EOF}} \\ = \frac{P_{ef(\text{EOF})}}{1 - P_{ef(\text{EOF})}}(T_{\text{EOF}} + \text{timeout\_EOF}) \\ + T_{\text{prop}} + T_{\text{EOF}}. \qquad (1)\end{aligned}$$

For $\text{timeout\_EOF} = 2T_{\text{prop}} + T_{\text{ACK(EOF)}}$, the expected EOF delivery time is

$$\frac{P_{ef(\text{EOF})}\left(2T_{\text{prop}} + T_{\text{EOF}} + T_{\text{ACK(EOF)}}\right)}{1 - P_{ef(\text{EOF})}} + T_{\text{prop}} + T_{\text{EOF}}. \qquad (2)$$

C. Expected File-Delivery Time

We first define and analyze the random variable representing the number of transmission spurts in the transaction. We define random variable K_i to represent the number of transmissions of the ith PDU, up to and including its first successful transmission. Then, under our channel assumption, K_i has a geometric distribution. The transmission spurts will reoccur until all PDUs are delivered to the receiver, so the number of transmission spurts is $\max(K_1, K_2, \ldots, K_N)$. We define random variable M_N as $M_N = \max(K_1, K_2, \ldots, K_N)$, and note that $M_N - 1$ is the number of retransmission spurts.

Now we consider the time interval between the issuance of a NAK and the reception of the corresponding retransmissions. Once EOF PDU has been successfully received, the receiver issues the first NAK and sets the NAK timer. In this analysis, we assume that the timeout value of the NAK timer is set at the two times propagation delay plus retransmission time. Since the receiver knows the amount of missing data and the transmission rate of the link, the receiver can simply compute the transmission time of those missing PDUs. Let us first consider the expected time between issuance of the first NAK and "nominal reception of the last bit of the first retransmission spurt," by which we mean the time that the last bit of the first retransmission spurt is transmitted plus T_{prop}. In the case that all of the PDUs of that retransmission spurt are lost, there is no actual reception. Note that the timeout value of the NAK timer is set as

$$T^k_{\text{timer(NAK)}} = 2T_{\text{prop}} + RT_k, \quad k = 1, 2, \ldots \qquad (3)$$

Taking into account the case that a NAK is lost (with the result that the NAK timer expires), the expected time between issuance of the first NAK and nominal reception of the last bit of the first retransmission spurt is given as

$$\begin{aligned}\sum_{i=1}^{\infty} i\left[T_{\text{NAK}} + T^1_{\text{timer(NAK)}}\right] P_{er}^{i-1}(1 - P_{er}) \\ = \frac{T_{\text{NAK}} + T^1_{\text{timer(NAK)}}}{1 - P_{er}} \\ = \frac{T_{\text{NAK}} + 2T_{\text{prop}} + RT_1}{1 - P_{er}}. \qquad (4)\end{aligned}$$

Similar expressions follow for the time between issuance of the first NAK after the nominal reception of the nth retransmission spurt and the nominal reception of last bit of the $(n+1)$th retransmission spurt. Thus, the expected time interval between the issuance of the first NAK and reception of last bit of last retransmission spurt can be obtained as

$$\mathbb{E}\left(\sum_{k=1}^{M_N-1}\frac{2T_{\text{prop}}+T_{\text{NAK}}+RT_k}{1-P_{er}}\right)=\frac{[\mathbb{E}(M_N)-1](2T_{\text{prop}}+T_{\text{NAK}})}{1-P_{er}}+\frac{\mathbb{E}\left(\sum_{k=1}^{M_N-1}RT_k\right)}{1-P_{er}}. \quad (5)$$

Note that $\mathbb{E}(\sum_{k=1}^{M_N-1}RT_k)$ is the expected total time taken for transmission of meta and file-data PDUs until all of them have been successfully delivered, minus the time taken to transmit them for their first trials. Thus

$$\mathbb{E}\left(\sum_{k=1}^{M_N-1}RT_k\right)=\sum_{i=1}^{N}\mathbb{E}(K_i-1)\cdot T_{\text{PDU}}=(N\cdot T_{\text{PDU}})\left(\frac{1}{1-P_{ef}}-1\right). \quad (6)$$

Note that in deferred NAK mode, the receiver sends NAK only after receiving EOF PDU. Therefore, expected file-delivery time of a transaction, which includes the expected EOF delivery time, is given as

$$\begin{aligned}&T_{\text{prop}}+N\cdot T_{\text{PDU}}+\frac{[\mathbb{E}(M_N)-1](2T_{\text{prop}}+T_{\text{NAK}})}{1-P_{er}}+\frac{(N\cdot T_{\text{PDU}})\left(\frac{P_{ef}}{1-P_{ef}}\right)}{1-P_{er}}\\&\quad+\left[\frac{P_{ef(\text{EOF})}\left(2T_{\text{prop}}+T_{\text{EOF}}+T_{\text{ACK(EOF)}}\right)}{1-P_{ef(\text{EOF})}}+T_{\text{EOF}}\right]\\&=T_{\text{prop}}+\frac{[\mathbb{E}(M_N)-1](2T_{\text{prop}}+T_{\text{NAK}})}{1-P_{er}}+N\cdot T_{\text{PDU}}\left[1+\frac{P_{ef}}{(1-P_{er})(1-P_{ef})}\right]\\&\quad+\left[\frac{P_{ef(\text{EOF})}\left(2T_{\text{prop}}+T_{\text{EOF}}+T_{\text{ACK(EOF)}}\right)}{1-P_{ef(\text{EOF})}}+T_{\text{EOF}}\right].\end{aligned} \quad (7)$$

To complete the analysis, we need to obtain $\mathbb{E}(M_N)$. We first provide the following proposition, which is somewhat illuminating.[1]

Proposition 1:

$$\frac{\sum_{k=1}^{N}\frac{1}{k}}{-\ln(P_{ef})}\le\mathbb{E}(M_N)<\frac{\sum_{k=1}^{N}\frac{1}{k}}{-\ln(P_{ef})}+1$$

Proof: See the Appendix.

Proposition 1 indicates that $\mathbb{E}(M_N)$ increases in logarithmic order with N. The expected file-delivery time in (7) has a term that increases linearly with N and a term that has the factor $\mathbb{E}(M_N)$. For very long propagation delay, the multiplicative factor $\mathbb{E}(M_N)$ is much larger than that of the term linear of N, which is on the order of the PDU transmission time. In such an environment, as the number of PDUs in the file (N) increases, the expected file-delivery time is initially dominated by the term logarithmically growing with N, and the order of growth later becomes linear with a small multiplicative factor for large values of N. For a small propagation delay (relative to the PDU transmission time, T_{PDU}), the order of growth is always dominated by the term linear of N. *Proposition 1* provides a good idea of the expected file-delivery time's order of growth with N, but the difference between the bounds in *Proposition 1* is 1.0. This can be considered loose, especially for application to the case of a long propagation delay. Thus, we now discuss numerical evaluation of $\mathbb{E}(M_N)$. We have

$$\begin{aligned}\mathbb{E}(M_N)&=\sum_{m=1}^{\infty}P(M_N\ge m)\\&=\sum_{m=1}^{\infty}[1-P(M_N<m)]\\&=1+\sum_{m=2}^{\infty}[1-P(M_N<m)]\\&=1+\sum_{m=2}^{\infty}\left[1-\prod_{i=1}^{N}P(K_i<m)\right]\\&=1+\sum_{m=2}^{\infty}\left[1-\left(1-P_{ef}^{m-1}\right)^N\right]\\&=1+\sum_{m=1}^{\infty}\left[1-\left(1-P_{ef}^{m}\right)^N\right].\end{aligned} \quad (8)$$

Note that $\mathbb{E}(M_N)$ can be expressed as a finite summation as follows:

$$\mathbb{E}(M_N)=1+\sum_{k=1}^{N}\binom{N}{k}\frac{P_{ef}^k}{1-P_{ef}^k}(-1)^{k+1}. \quad (9)$$

Thus, in theory, we can compute the exact value of $\mathbb{E}(M_N)$ in a finite number of computational operations. However, we face difficulties in numerical evaluation for a large value of N. Term $\binom{N}{k}(P_{ef}^k/(1-P_{ef}^k))(-1)^{k+1}$ of the summation in (9) can have a very large factor $\binom{N}{k}$ and a very small P_{ef}^k. Thus, the evaluation of a term can be numerically difficult. Truncating the summation by omitting the terms that are difficult to compute does not give a good idea of how accurate such an approximation is. In addition, the terms in the summation could be both positive and negative, so such truncation does not give an upper or lower bound either. In fact, from (8), we can use finite summation $1+\sum_{m=1}^{s^*}[1-(1-P_{ef}^m)^N]$ as both an approximation and a lower bound. As we increase the number of additions s^*, the evaluation becomes more accurate. The numerical inaccuracy (the remainder) can be expressed as follows:

$$\begin{aligned}R_{s^*}&\triangleq\sum_{m=s^*+1}^{\infty}\left[1-\left(1-P_{ef}^m\right)^N\right]\\&=\sum_{m=s^*+1}^{\infty}P_{ef}^m\left\{1+\left(1-P_{ef}^m\right)+\left(1-P_{ef}^m\right)^2+\cdots+\left(1-P_{ef}^m\right)^{N-1}\right\}.\end{aligned} \quad (10)$$

We can guarantee the error percentage of numerical evaluation $1+\sum_{m=1}^{s^*}[1-(1-P_{ef}^m)^N]$ by obtaining an upper bound

[1] We derived these bounds with simple engineering mathematics, as presented in the Appendix. Other mathematically interesting properties of M_N can be found in [6].

on R_{s^*}. By using a generalized Bernoulli's inequality [8, p. 69], term $(1-P_{ef}^m)^n$, $n=1,2,\ldots,N-1$ in (10) can be bounded above by

$$\frac{1-P_{ef}^m}{1+(n-1)P_{ef}^m}.$$

Thus, we can obtain the following upper bound of R_{s^*}:

$$R_{s^*} \leq \sum_{m=s^*+1}^{\infty} P_{ef}^m \left\{ 1 + \left(1-P_{ef}^m\right) + \frac{1-P_{ef}^m}{1+P_{ef}^m} + \frac{1-P_{ef}^m}{1+2P_{ef}^m} + \cdots + \frac{1-P_{ef}^m}{1+(N-2)P_{ef}^m} \right\}. \tag{11}$$

Individual terms in the right-hand side of (11) can be bounded above by using the following relations:

$$\sum_{m=s^*+1}^{\infty} \frac{P_{ef}^m}{1+nP_{ef}^m} \leq \int_{s^*+1}^{\infty} \frac{P_{ef}^x}{1+nP_{ef}^x} dx + \frac{P_{ef}^{s^*+1}}{1+nP_{ef}^{s^*+1}} \tag{12}$$

$$\sum_{m=s^*+1}^{\infty} \frac{P_{ef}^{2m}}{1+nP_{ef}^m} \geq \int_{s^*+1}^{\infty} \frac{P_{ef}^{2x}}{1+nP_{ef}^x} dx. \tag{13}$$

Let $y = 1 + nP_{ef}^x$, then we have

$$\int_{s^*+1}^{\infty} \frac{P_{ef}^x}{1+nP_{ef}^x} dx = \int_{1+nP_{ef}^{s^*+1}}^{1} \frac{1}{(n \ln P_{ef})y} dy = \frac{-\ln\left(1+nP_{ef}^{s^*+1}\right)}{n \ln P_{ef}} \tag{14}$$

$$\int_{s^*+1}^{\infty} \frac{P_{ef}^{2x}}{1+nP_{ef}^x} dx = \int_{1+nP_{ef}^{s^*+1}}^{1} \frac{y-1}{(n^2 \ln P_{ef})y} dy = -\frac{P_{ef}^{s^*+1}}{n \ln P_{ef}} + \frac{\ln\left(1+nP_{ef}^{s^*+1}\right)}{n^2 \ln P_{ef}}. \tag{15}$$

From (11)–(15), we have

$$\begin{aligned} R_{s^*} \leq & \sum_{m=s^*+1}^{\infty} \left(2P_{ef}^m - P_{ef}^{2m}\right) + \sum_{n=1}^{N-2} \sum_{m=s^*+1}^{\infty} \frac{P_{ef}^m}{1+nP_{ef}^m} \\ & - \sum_{n=1}^{N-2} \sum_{m=s^*+1}^{\infty} \frac{P_{ef}^{2m}}{1+nP_{ef}^m} \\ \leq & \frac{2P_{ef}^{s^*+1}}{1-P_{ef}} - \frac{P_{ef}^{2(s^*+1)}}{1-P_{ef}^2} \\ & + \sum_{n=1}^{N-2} \left[\frac{-\ln\left(1+nP_{ef}^{s^*+1}\right)}{n \ln P_{ef}} + \frac{P_{ef}^{s^*+1}}{1+nP_{ef}^{s^*+1}} \right] \\ & + \sum_{n=1}^{N-2} \left[\frac{P_{ef}^{s^*+1}}{n \ln P_{ef}} - \frac{\ln\left(1+nP_{ef}^{s^*+1}\right)}{n^2 \ln P_{ef}} \right]. \end{aligned} \tag{16}$$

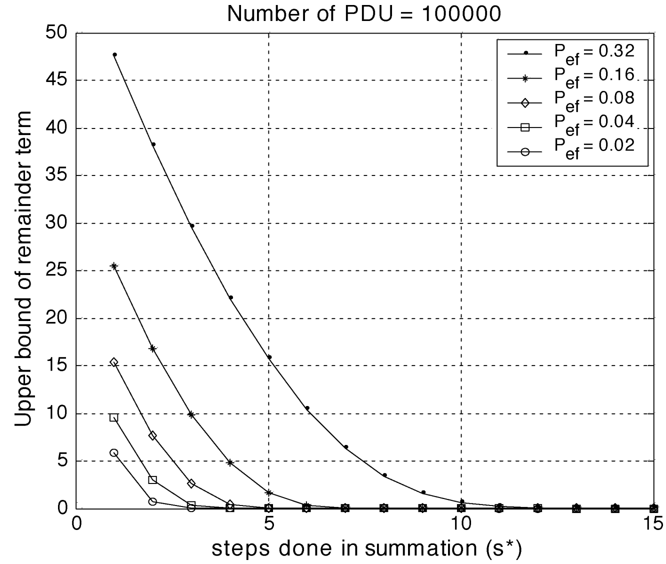

Fig. 3. Upper bound of R_{s^*}.

For a desired level of accuracy, one can use (16) to determine an appropriate value of s^* for computing $\mathbb{E}(M_N)$. Fig. 3 indicates that this upper bound decays very rapidly as we increase s^*. Thus, one can numerically compute $\mathbb{E}(M_N)$ with a fairly small number of additions and guarantee a small error percentage. In order to further simplify the decision of a proper value of s^* for numerical computation, one can obtain an upper bound of the right-hand side of (16). For example, using the well-known inequality $\ln(1+x) \leq x$, we derive

$$\begin{aligned} R_{s^*} \leq & \frac{2P_{ef}^{s^*+1}}{1-P_{ef}} - \frac{P_{ef}^{2(s^*+1)}}{1-P_{ef}^2} + \sum_{n=1}^{N-2} \left[\frac{-P_{ef}^{s^*+1}}{\ln P_{ef}} + \frac{P_{ef}^{s^*+1}}{1+nP_{ef}^{s^*+1}} \right] \\ \leq & \frac{2P_{ef}^{s^*+1}}{1-P_{ef}} - \frac{P_{ef}^{2(s^*+1)}}{1-P_{ef}^2} + \sum_{n=1}^{N-2} \left[\frac{-P_{ef}^{s^*+1}}{\ln P_{ef}} + P_{ef}^{s^*+1} \right] \\ = & \frac{2P_{ef}^{s^*+1}}{1-P_{ef}} - \frac{P_{ef}^{2(s^*+1)}}{1-P_{ef}^2} \\ & + (N-2)\left(1 - \frac{1}{\ln P_{ef}}\right) P_{ef}^{s^*+1}. \end{aligned} \tag{17}$$

The upper bound in (17), in addition, explicitly shows that the error term decays at least exponentially fast as s^* increases.

III. Expected File-Delivery Time: Numerical Presentation

The mathematical expression derived in previous sections for the expected file-delivery time in deferred NAK mode is numerically presented in Figs. 4–6. Note that the astronomical unit (a.u., 1 a.u. = 480 s) is used. Figs. 4–6 illustrate how the expected file-delivery time in deferred NAK mode is affected by variables such as the PDU error rate, the number of PDUs in the file, the PDU transmission time, etc. In these figures, we assumed that T_{NAK}, T_{EOF}, and $T_{\mathrm{ACK(EOF)}}$ are two orders of magnitude less than T_{PDU}, because of the small sizes of the NAK, EOF, and ACK(EOF) PDUs. We also assumed that

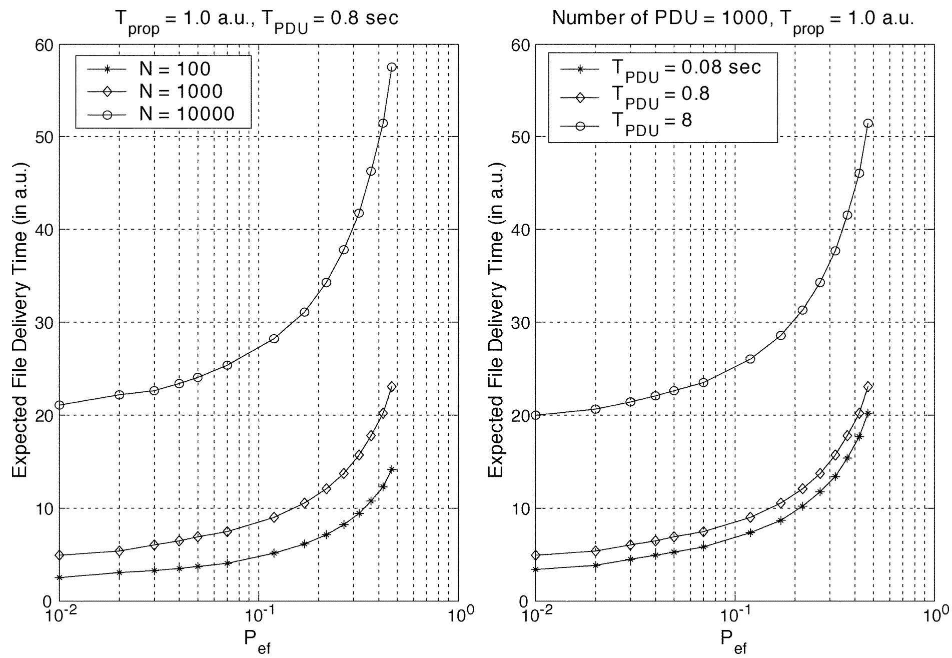

Fig. 4. Expected file-delivery time versus P_{ef}.

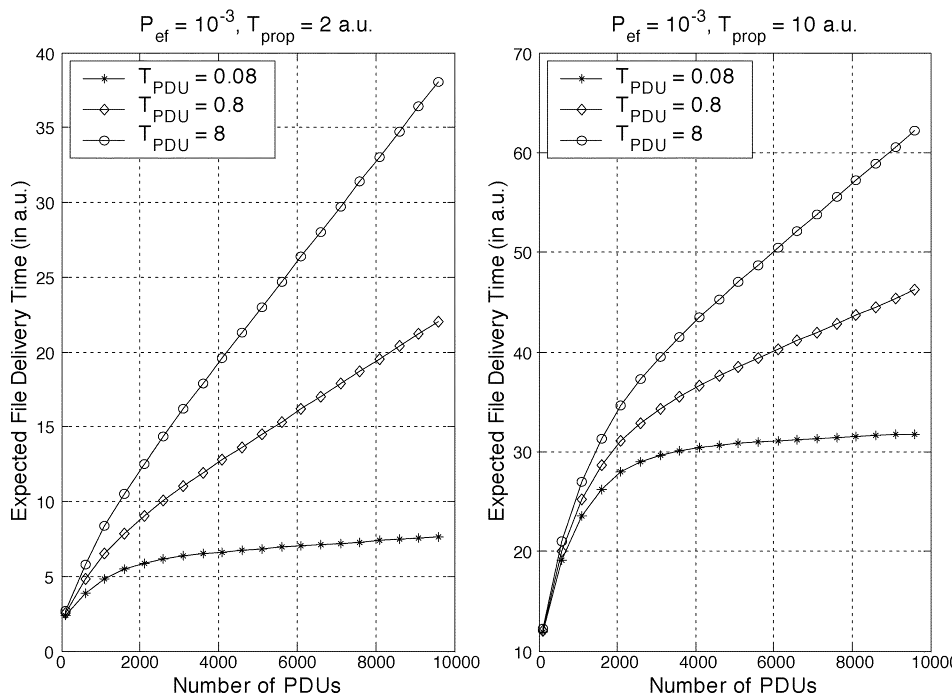

Fig. 5. Expected file-delivery time versus number of PDUs.

$P_{ef(\mathrm{EOF})}$ and P_{er} are two orders of magnitude less than P_{ef} for the same reason.

In Fig. 7, we compare the numerical evaluation of (7) and the results of random simulation. In this figure, we set $s^* = 20$ to compute $\mathbb{E}(M_N)$ numerically. The figure illustrates how the expected file-delivery time is affected by the bit-error rate (BER) of the link. The considered region of BER without forward error correction (FEC) is between 10^{-5} and 10^{-7}, because achievable BERs without FEC range between 10^{-5} and 10^{-7} in typical space communications. The simulation results and the mathematically derived results closely match, as can be observed from the figure. However, the random simulation took much more computational time and required much more programming effort.

IV. Conclusion

We derived the expression for the minimum expected file-delivery time of the CFDP deferred NAK mode under the constraint that the throughput efficiency is maximized, in the sense that there is no unnecessary duplicate retransmission. For the purpose of gaining simple performance intuition, in determining

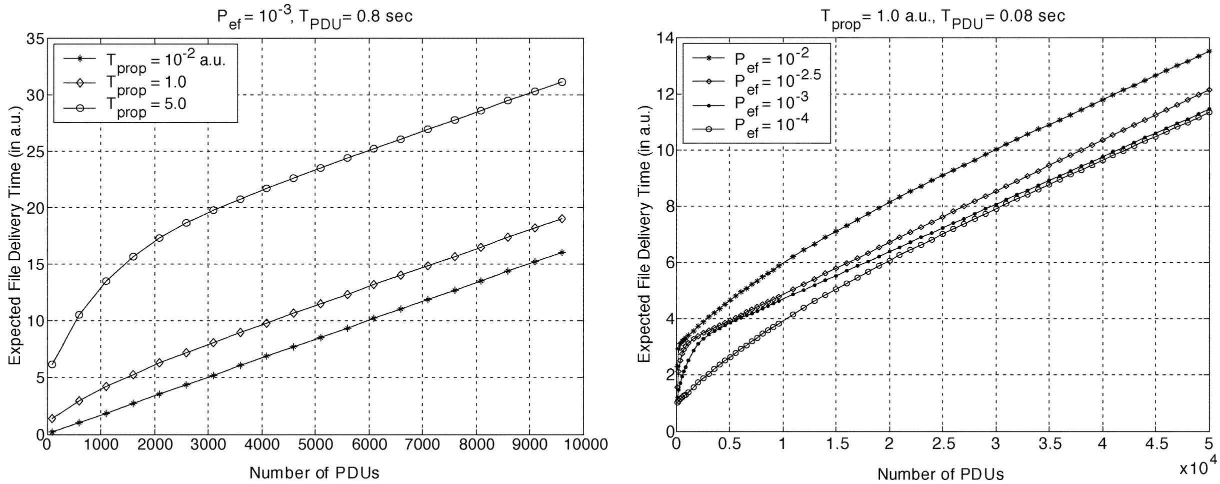

Fig. 6. Expected file-delivery time versus number of PDUs.

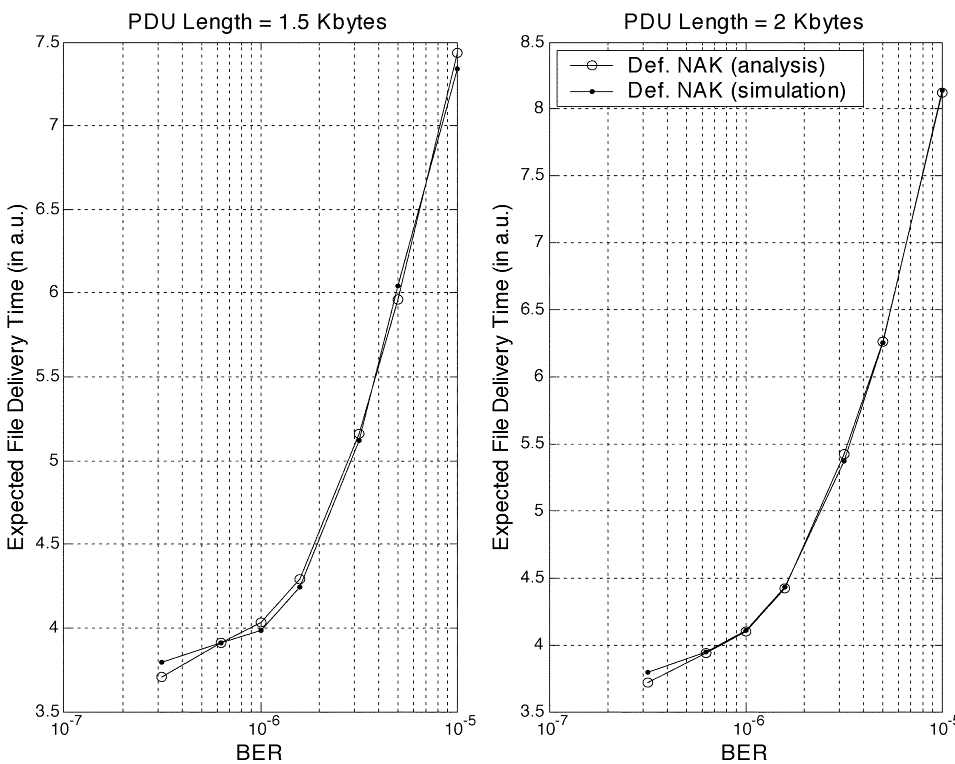

Fig. 7. Deferred NAK: analytic and simulation results. Expected file-delivery time of deferred NAK mode versus BER. File size $=$ 1 MB, transmission rate $=$ 20 Kb/s in both directions, and propagation delay $= 480$ s.

the NAK timer-setting rule, we assumed that the sender can start retransmission immediately after receiving a NAK PDU. (Recall that we set the timeout value of the NAK timer for the kth retransmission spurt as $T^k_{\mathrm{timer(NAK)}} = 2T_{\mathrm{prop}} + RT_k$ in Section II.) In real operations, the sender may not be able to start retransmission of the PDUs requested by the NAK. For example, if the sender is performing multiple outgoing transactions concurrently (multiplexed transactions), the sender may have to delay the requested retransmissions in a particular transaction, because of previously queued outbound data belonging to another transaction that must be transmitted before the newly requested PDUs are retransmitted. This queuing delay is difficult to estimate. However, a simple way of improving throughput efficiency in such an operational environment is to add a constant value z to the NAK timer, namely, to use

$$T^k_{\mathrm{timer(NAK)}} = 2T_{\mathrm{prop}} + RT_k + z, \quad k = 1, 2, \ldots \qquad (18)$$

The actual value of z to be used depends upon the level of throughput efficiency desired and the PDU scheduling scheme of the sender. (The scheduling scheme at the sender is specific to the implementation, and beyond this paper's scope.) If the timeout value of NAK timer (18) is used in place of the timeout value of NAK timer (3), the expected time between the issuance of the first NAK and the nominal reception of the last bit of the last transmission spurt in (5) is replaced by

$$\mathbb{E}\left(\sum_{k=1}^{M_N-1} \frac{2T_{\text{prop}} + T_{\text{NAK}} + RT_k + z}{1 - P_{er}}\right) = \frac{[\mathbb{E}(M_N) - 1]\,(2T_{\text{prop}} + T_{\text{NAK}} + z)}{1 - P_{er}} + \frac{\mathbb{E}\left(\sum_{k=1}^{M_N-1} RT_k\right)}{1 - P_{er}} \tag{19}$$

and thus, the expected file-delivery time is

$$T_{\text{prop}} + \frac{[\mathbb{E}(M_N) - 1]\,(2T_{\text{prop}} + T_{\text{NAK}} + z)}{1 - P_{er}} + N \cdot T_{\text{PDU}} \left[1 + \frac{P_{ef}}{(1 - P_{er})(1 - P_{ef})}\right] + \left[T_{\text{EOF}} + \frac{P_{ef(\text{EOF})}\left(2T_{\text{prop}} + T_{\text{EOF}} + T_{\text{ACK(EOF)}}\right)}{1 - P_{ef(\text{EOF})}}\right] \tag{20}$$

in place of (7).

Note that the expected file-delivery time depends upon several variables, e.g., file size, PDU size, the propagation delay, etc. With the results of our mathematical derivation in (7) and (20), we can generate numerical values for the expected file-delivery time quickly, without computationally intensive random simulation, for a range of different environmental and design variables.

Finally, we note that the CFDP or its variant may be useful beyond space applications, although the CFDP has been standardized by CCSDS for use in space networking. For example, the feature of no ACK message for the file-data PDUs (i.e., NAK only) may also be useful for secure communication, in which the receiver's emission should be small in order to hide its presence or location.

APPENDIX
PROOF OF *PROPOSITION 1*

K_i for each i has geometric distribution

$$P(K_i = k) = P_{ef}^{k-1}(1 - P_{ef}), \quad k = 1, 2, 3, \ldots \tag{21}$$

We take the approach of approximating $M_N \equiv \max(K_1, K_2, \ldots, K_N)$ by $\max(X_1, X_2, \ldots, X_N)$, where X_i for each i has exponential distribution

$$f_X(x) = \lambda \exp(-\lambda x), \quad x \geq 0. \tag{22}$$

A. $\max(X_1, X_2, \ldots, X_N)$

Recall that the expected time until any arrival among n independent Poisson processes, each with arrival rate λ, is $1/(n\lambda)$ [7]. Consider N independent Poisson processes, each of which terminates after the first arrival. Then, the time until the Nth arrival (the last arrival) is $\max(X_1, X_2, \ldots, X_N)$. Therefore, we see that

$$\mathbb{E}\left[\max(X_1, X_2, \ldots, X_N)\right] = \frac{1}{N\lambda} + \frac{1}{(N-1)\lambda} + \cdots + \frac{1}{2\lambda} + \frac{1}{\lambda} = \frac{1}{\lambda}\sum_{k=1}^{N}\frac{1}{k}. \tag{23}$$

B. $M_N \equiv \max(K_1, K_2, \ldots, K_N)$

Consider mapping d

$$d(x) \equiv \left\lceil \frac{\lambda x}{-\ln P_{ef}} \right\rceil \tag{24}$$

where we denote $\lceil y \rceil \equiv \min\{n \in Z |\, y \leq n\}$. Then, random variables $d(X_1), d(X_2), \ldots, d(X_N)$ are statistically independent, and have a geometric distribution identical to that of random variable K_i. Moreover, for each realization of random variables $X_1, X_2, \ldots, X_N$, we have

$$\max\left\{d(X_1), d(X_2), \ldots, d(X_N)\right\} = d\left(\max(X_1, X_2, \ldots, X_N)\right). \tag{25}$$

Therefore, we have

$$\mathbb{E}(M_N) = \mathbb{E}\left[\max(K_1, K_2, \ldots, K_N)\right] = \mathbb{E}\left[\max\left\{d(X_1), d(X_2), \ldots, d(X_N)\right\}\right] = \mathbb{E}\left[d\left(\max(X_1, X_2, \ldots, X_N)\right)\right]. \tag{26}$$

From (24), we have

$$\frac{\lambda \max(X_1, X_2, \ldots, X_N)}{-\ln P_{ef}} \leq d\left(\max(X_1, X_2, \ldots, X_N)\right) < \frac{\lambda \max(X_1, X_2, \ldots, X_N)}{-\ln P_{ef}} + 1. \tag{27}$$

Therefore, from (23), (26), and (27), we have

$$\frac{\sum_{k=1}^{N}\frac{1}{k}}{-\ln P_{ef}} = \frac{\lambda \mathbb{E}\left[\max(X_1, X_2, \ldots, X_N)\right]}{-\ln P_{ef}} \leq \mathbb{E}(M_N) < \frac{\lambda \mathbb{E}\left[\max(X_1, X_2, \ldots, X_N)\right]}{-\ln P_{ef}} + 1 = \frac{\sum_{k=1}^{N}\frac{1}{k}}{-\ln P_{ef}} + 1. \tag{28}$$

■

REFERENCES

[1] *CCSDS File Delivery Protocol (CFDP), Part 1: Introduction and Overview*, CCSDS 720.1-G-1, Jan. 2002.

[2] *CCSDS File Delivery Protocol (CFDP), Part 2: Implementers Guide*, CCSDS 720.2-G-1, Jan. 2002.

[3] *CCSDS File Delivery Protocol (CFDP), Recommendation for Space Data System Standards*, CCSDS 727.0-B-2, Oct. 2002.

[4] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, "Delay-tolerant networking: An approach to interplanetary internet," *IEEE Commun. Mag.*, pp. 128–136, June 2003.

[5] W. Baek and D. C. Lee. Caution for initializing and closing CFDP transaction. presented at *Proc. Int. Telemetering Conf.* [CD-ROM]

[6] H. Prodinger, "Combinatorics of geometrically distributed random variables: Left-to-right maxima," *Discr. Math.*, vol. 153, pp. 253–270, 1996.

[7] A. W. Drake, *Fundamentals of Applied Probability Theory*. New York: McGraw-Hill, 1967.

[8] D. S. Mitrinović, J. E. Pečarić, and A. M. Fink, *Classical and New Inequalities in Analysis*. Dordrecht, The Netherlands: Kluwer, 1993.

Daniel C. Lee (S'91–M'92) received the B.S. degree in electrical engineering (with honors) and the B.S. degree in mathematics, both in 1985, from the University of Maryland at College Park, and the S.M. degree in 1987 and the Ph.D. degree in 1992, both from the Massachusetts Institute of Technology, Cambridge, in electrical engineering and computer science.

From 1993 to 1998, he devoted his research to the systems engineering of networks and communication systems at the U.S. Naval Research Laboratory (NRL), Washington, DC. At the Center for Computational Science of NRL, he participated in the development of an object-oriented protocol software framework, CASiNO. At the Naval Space Center of NRL, he developed a proxy agent for managing the ICEbox network, a U.S. government information-dissemination system. In 1998, he joined the Electrical Engineering Department, University of Southern California, Los Angeles, where his main research interests have been quality of service and resource-allocation issues in communication systems and networks. Applications of his research include wireless communications and networking, sensor networks, optical networks, and internet multimedia.

Dr. Lee's honors include the Alan Berman Research Publication Award from NRL in 1995, the Navy's Outstanding Performance Award at NRL in 1995, and the Frederick C. Hennie III Teaching Award from MIT in 1989.

Wonseok Baek (S'98) received the B.S. degree in electronics engineering from Seoul National University, Seoul, Korea, in 1995, and the M.S. degree in electrical engineering from Polytechnic University, Brooklyn, NY. He is currently working toward the Ph.D. degree at the Department of Electrical Engineering, Communication Science Institute, University of Southern California, Los Angeles.

His research interests are cross-layer design issues in wireless networks.