IET Communications

Research Article

Double retransmission deferred negative acknowledgement in Consultative Committee for Space Data Systems File Delivery Protocol for space communications

ISSN 1751-8628
Received on 6th December 2014
Revised on 21st July 2015
Accepted on 2nd September 2015
doi: 10.1049/iet-com.2014.1184
www.ietdl.org

Haitao Wu[1,2], *Qinyu Zhang*[1], *Zhihua Yang*[1] ✉, *Jian Jiao*[1], *Yunhe Li*[2], *Shushi Gu*[1]

[1]*Department of Electronics and Information Engineering, Harbin Institute of Technology Shenzhen Graduate School, Shenzhen, People's Republic of China*
[2]*Faculty of Electronic Information and Mechanical Electrical Engineering, Zhaoqing University, Zhaoqing, People's Republic of China*
✉ *E-mail: yangzhihua@hitsz.edu.cn*

Abstract: To improve the reliability of file transfer and shorten file transfer time in space communication, this study aims to provide an improved strategy for deferred negative acknowledgement (NAK) in Consultative Committee for Space Data Systems File Delivery Protocol (CFDP). Based on a theoretical analysis of the recommended deferred NAK, the authors propose a double retransmission deferred NAK strategy instead to guarantee the reliability of file transfer; the file transfer time is reduced significantly using fewer retransmission spurts. They make the performance comparisons of the recommended deferred NAK in CFDP with the authors' proposed strategy under several typical scenarios. Numerical and simulation results show the effectiveness of the proposed strategy.

Nomenclature

K, N_{retra}	number of times a retransmitted PDU is repeated
N	total number of PDUs in a transaction
L_{PDU}	length of PDU
P_e	bit error rate of link
P_{ef}	error or loss probability of PDU
P_{er}	error probability of delivering NAK
T_{prop}	one-way propagation delay
T_{PDU}	transmission time of file data PDU or meta-data PDU
RT	transmission time of the PDUs in recovery stage
σ	random variable that obeys uniform distribution in the interval [0, 3]
θ	random variable that obeys uniform distribution in the interval [0, 1]
S	random variable, transmission number of a PDU in retransmission stage

1 Introduction

Over the past few decades, the Consultative Committee for Space Data Systems (CCSDS) has standardised a set of communication protocols for deep space exploration, and other space-based satellite networks. The objective of space communication is to realise scientific missions to the Moon, Mars, and other space exploration by providing communication infrastructure among planets, moons, satellites, spacecraft, crewed vehicles, rovers, landers, sensors and so on. Space communication is very specific to a particular mission or operation, and ensuring reliable communications is one of the primary requirements of space missions [1].

Ordinarily there are two kinds of reliable transfer mechanisms: retransmissions and redundant coding schemes. The former is implemented through explicit retransmission request or expiration of a timer set at the sender side. The latter may adopt packet-oriented erasure correcting codes to enable automatic recovery of the missing data on the receiver side. Furthermore, coding solutions applying the concept of a digital fountain such as LT, Tornado, and Raptor, deserve close attention, and were initially considered within the CCSDS standardisation process. In spite of the virtues they offer, they still require the constant availability of a feedback channel for signaling the completion of the decoding procedure to the sender, which would continuously transmit new redundancy symbols, wasting power and link bandwidth [2].

As is widely known, CCSDS File Delivery Protocol (CFDP) [3] is a permanent part of the deep space communication protocol. It offers end-to-end services for file transfer to and from onboard mass memories and gets reliable transfer of files by following a file transfer protocol-like paradigm. It adopts an Automatic Repeat reQuest (ARQ) mechanism to guarantee reliability, using negative acknowledgement (NAK) to replace the acknowledgement (ACK) that is widely used in terrestrial network. Being one of the reliable services in CFDP that scientific communities mostly focus on, the deferred NAK mode is thoroughly considered and analysed in this paper.

The remaining of this paper is organised as follows. Section 2 discusses some related work with regard to evaluating and optimising the performance of CFDP. The algorithmic description, theoretical analysis, and simulations of deferred NAK in recommended CFDP are explained in Section 3. An improved strategy for deferred NAK, namely the double retransmission (DR) strategy, is provided in Section 4. In Section 5, comparisons and simulations of the recommended deferred NAK and our strategy are performed under three typical scenarios. Finally, conclusions are drawn in Section 6.

2 Related work

Some work had been done already to evaluate and optimise the CFDP performance. A theoretical analysis of the expected file transfer time in deferred NAK CFDP over a direct link had been performed [4]. All parameter values were set under the restriction that the throughput performance was never decreased and that file transfer time was considered as a whole, thus blurring the meaning of 'defer' as well as the boundary between the first sending and recovery stage. In addition, Baek and Lee [5] presented a theoretical analysis of the expected file transfer time in immediate NAK CFDP. A mathematical analysis of performance evaluation of CFDP for latency and storage requirement were derived from Gao and SeGuí [6]. De Cola devised a mechanism to extend the CFDP features, named as CFDP-UE-RT (repeated transmission), transmitting the same protocol data unit (PDU) for $N-1$ times consecutively; N was the number of CFDP PDUs that transferred to the underlying layer. Although it increased the probability of data delivery, it cost large amounts of bandwidth [7]. The relationships between the packet loss rate, packet size, throughput, and the number of repeated transmissions were discussed only under the cislunar scenario, whereas the file transfer time was not mentioned. Papastergiou *et al.* [8] advanced the deep-space transport protocol (DS-TP), which transmitted each packet twice, importing some delay between the original transmission and the retransmission. DS-TP could obtain two times faster than conventional protocols such as TCP, SCPS-TP, or Saratoga protocol, and thus improved file transfer time, but it induced higher overhead.

Recently, codec algorithms have been introduced in CFDP optimising solutions to improve the reliability of point-to-point transmission. In another work [7], De Cola also proposed erasure coding technologies combined with CFDP core procedure in deep space communication such as CFDP-UE-RSE (Reed Solomon encoding) and CFDP-UE-LDPC (low-density Parity check). The latter offered the best results thanks to the robust coding technique. In the case of 'hard link intermittence,' CFDP-UE-RSE offered encouraging results, while CFDP-UE-RT was promising when applied to 'almost clear sky' conditions. Packet layer coding algorithms applied with long erasure codes [2] could attain more satisfactory results than ARQ-based schemes in several aspects. However, the use of pure erasure coding schemes required careful optimisation of code rate and frame length. Deep-space file transfer protocol (DSFTP) [9] combined fountain coding with a decoding mechanism to overcome retransmission by using repeated decoding at the receiver. Nevertheless, when the bit error rate (BER) was lower than the decoding threshold, the performance degraded significantly. Packets interleaving CFDP [10] retained the retransmission process and achieved data error recovery by an interleaved coding mechanism, but it increased the complexity of the sender and introduced more overhead. Repeated sending file delivery protocol [11] depended on the repeated sending and real-time adaptation of BER, which was calculated according to the PDU loss probability of current link. The receiver sent back the estimated BER of the current link along with NAK, but the BER information would be obsolete due to the long one-way propagation time, causing the sender cannot reliably set the proper number of duplicate retransmissions.

Space communications have entered a new era in which a multi-hop architecture is exploited, presenting an increasing number of alternative communication paths. This can be traced back to the emergence of the Interplanetary Internet [12] and delay-tolerant networks (DTNs) [13]. A space-oriented file transfer protocol for DTN was provided in [14]. A DTN-oriented protocol design incorporating ensure codes within CFDP was presented in [15]. The delay-tolerant transport protocol (DTTP) [16] addressed reliable data transfer in stressed space communications. Being primarily a transport layer protocol, it satisfied the inherent architecture requirements of DTN in the absence of IP network infrastructure. It allowed for reliable, efficient data transfer, offering a number of application-oriented transmission strategies.

From the above work, we can conclude the following. On the one hand, CFDP was definitely designed for deep space and is still very valuable. Actually, NASA's DINET 3 project aimed mainly at upgrading the CFDP software on the deep impact spacecraft to run over DTN. Due to DTN stack is essentially a decomposition of CFDP, where the functional modules have been significantly enhanced: the acknowledged procedures of CFDP are implemented in Licklider transmission protocol (LTP) (except that file transfer may be used for other applications) and the extended procedures of CFDP are implemented in Bundle Protocol (BP) (which requires more powerful routing and flow control). What is left in CFDP when we remove these capabilities are file segmentation, transmission, and reassembly, which are also the critical core of CFDP – yet these parts of CFDP are not addressed in DTN architecture. So CFDP is a permanent part of deep space communication architecture. On the other hand, we should design protocol in hopes of simplifying the protocol stack, because the erasure coding technologies implanted into the data link or other layer make the protocols more complex and hard to control or manipulate over extremely long distances. Reliability in the event of a sustained outage must also be considered, because so many code blocks would be lost during the period of outage, causing the receiver or intermediate node difficulty to reconstruct the file data. At present, CFDP is one of the most important applications that the DTN architecture has to support. Space DTN is still in its incubation period, issues such as routing and congestion control have not been researched thoroughly, and the protocol is very much application specific.

Consequently, the reliability of file transfer is an open issue in simplifying the protocol design, to avoid introducing excessive complexity while keeping it easily scalable to DTN architecture. The purpose of this paper is to bring forward an improved strategy for recommended deferred NAK in CFDP. We design a DR strategy to guarantee the reliability of file transfer and make a trade-off between file transfer time and throughput reasonably. We conduct a theoretical analysis of deferred NAK and our proposed strategy and measure their performance under several typical scenarios.

3 Algorithmic descriptions and analysis of deferred NAK

3.1 Preliminaries

3.1.1 CFDP description: CFDP is mainly an application-layer protocol that includes transport-layer functionalities as well [1]. CFDP provides two kinds of file transfer services: core procedure and extended procedure [17]. The former offers basic point-to-point file transfer functions implemented across a direct single link. The latter is designed for end-to-end file transfer in more complex mission scenarios where no direct link is available between the source and the destination, and it supports multi-hop delivery across an arbitrary network involving multiple links [18]. It provides subsequent transmissions of files between intermediate nodes, which end up at the destination node (i.e. store-and-forward overlay) [14].

In CFDP, file transfer operation is named a *transaction*, and the sender assigns an ID number for each transaction. Each file is segmented into PDUs of variable length before transmission. The header length can range up to 24 bytes; the payload can contain up to 65, 536 bytes. For the first transmission attempt, a meta-data PDU is sent at the beginning, followed by the file data PDUs sent out in sequence with an end-of-file (EOF) PDU, which marks the end of the first transmission attempt and signals the receiver to respond. If missing or error-corrupted PDUs were detected, a NAK message is issued back to the sender, with a list of PDUs that need to be retransmitted. Upon reception of the NAK, the sender will retransmit the requested PDUs until all missing PDUs have been successfully received. Once all PDUs are correctly received, the receiver will send a Finished (FIN) message to the sender, signalling the completion of the transaction.

CFDP can work in either unreliable or reliable mode, according to mission requirements and transmission capability. The unreliable mode implements no mechanisms to ensure completeness of data delivery. The communication reliability, if necessary, may be ensured by proper mechanisms implemented within the underlying layers. When reliable service has been selected, the CFDP uses both NAK and ACK. NAKs are used to request retransmission of lost data, and ACKs are used for ensuring the receipt of EOF and FIN PDUs [17].

There are four selectable options associated with the issuance of NAKs: *immediate*, *deferred*, *prompted*, and *asynchronous*. In the

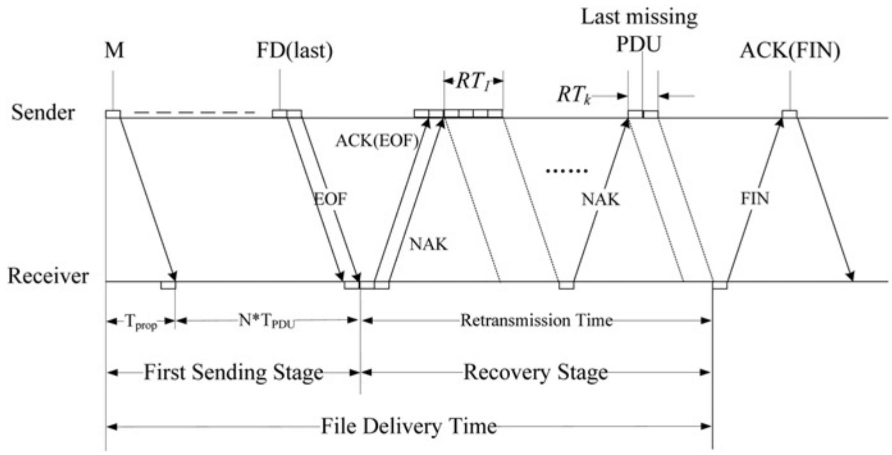

Fig. 1 *Transfer procedure of deferred NAK mode*

first case, once missing PDUs are detected, a NAK is issued in order to trigger the recovery phase at the sender side as soon as possible. When CFDP is configured to run in deferred mode, the detection of missing PDUs is performed only after the initial reception of the EOF (no error) PDU. As far as prompted and asynchronous modes are concerned, the detection of missing PDUs depends on external events, such as explicit (asynchronous mode) or periodic (prompted mode) requests by the sender. The recovery phase is managed by means of NAK timers, necessary to reissue NAK notifications if PDUs are still missing after initial retransmission rounds [7]. Here, we consider deferred NAK thoroughly in this paper.

3.1.2 Deferred NAK mode: In this mode, the receiver entity defers the issuance of NAK that contains information about all missing PDUs. After the EOF is received correctly, the receiver replies an ACK (EOF) and then a NAK (if needed). Upon the reception of the NAK, the sender promptly retransmits every PDU that the NAK demands. After each issuance of a NAK, a NAK timer is initiated by the receiver; when the timer expires, the receiver checks again the list of missing PDUs. If there are remaining missing PDUs, another NAK is issued and again a NAK timer is started by the receiver. This kind of process recurs until the receiver gets the last missing PDU. When all PDUs are received correctly, a FIN PDU is issued by the receiver. After successful reception of the FIN PDU, an ACK (FIN) is issued and then the transaction is closed by the sender. When receiving the ACK (FIN) successfully, the receiver subsequently closes the whole transaction. The transfer procedure of recommended deferred NAK is illustrated in Fig. 1. The total file delivery time contains two stages: first sending stage and recovery stage.

3.2 Model and theoretical analysis of deferred NAK

File transfer time is always a key criterion of evaluating protocol performance. Here, we define *file transfer time* to be the interval from the start of the transfer process to the instant when all data PDUs, meta-data PDU, and EOF PDU have been delivered to the receiver successfully [4]. As depicted in Fig. 1, we chronologically divide the total file transfer time into two separated stages: first sending stage and recovery stage. The *first sending stage* starts from the sender's transfer of meta-data PDU that initiates the transaction and finishes after the errorless reception of EOF at the receiver entity. The *recovery stage* begins with the deliver immediately following transfer of ACK (EOF) and ends at the instant when all missing PDUs have been correctly received. It should be noted that our definition of file transfer time excludes the procedure of FIN-ACK (FIN).

For the theoretical analysis combined with the actual situations, we consider the transport layer channel to be an erasure channel and make some assumptions as follows:

- The link is full duplex.
- Length of all file data PDUs and meta-data PDU is identical.
- Error or loss probability for each PDU is identical.
- PDU error events in uplink and downlink are statistically independent.

As the length of EOF, ACK (EOF), and NAK is 72 bits, 16 bits and $64(N_R+1)$ bits, respectively, in which N_R is the number of PDUs to be retransmitted in NAK, the transmission time is negligible compared to the file data PDUs. As a result, the error or loss probability for them is smaller than that of file data PDUs. For simplicity, we assume that the error or loss probability of EOF and ACK (EOF) is negligible; that is we assume that CFDP control messages such as EOF and ACK (EOF) always arrive correctly, even though the channel is not perfect. The error or loss probability of a NAK is two orders of magnitude less than the error or loss probability of file data PDUs.

We denote N the total PDU number in a transaction, including one meta-data PDU and $N-1$ file data PDUs. Thus the total file transfer time roughly contains three parts: one-way propagation delay, time for transmitting N PDUs, and retransmission time during the recovery stage, as depicted in Fig. 1.

Before the theoretical analysis of recommended deferred NAK, we declare the setting rules on both two timers of EOF and NAK. The rules are set to minimise the expected file transfer time on condition that throughput performance is never decreased. Considering CFDP over a direct single-hop link, for the sake of preventing redundant retransmission, the proper time-out value for EOF timer should be $2T_{\text{prop}}+\sigma$. That is, this setting value for EOF timer should better double the one-way propagation time ($2T_{\text{prop}}$), plus a random variable σ to allow for small degrees of unaccounted delay associated with the transmission. The time-out value of the NAK timer set upon issuance of the NAK that causes the kth retransmission spurt in the recovery stage should be $2T_{\text{prop}}+RT_k+\theta$, where RT_k denotes the transmission time of the PDUs requested by the receiver for the kth retransmission spurt in the recovery stage, and θ is a random variable allowing for small degrees of queuing delay associated with the retransmission.

Under the above assumptions and description, the file transfer time during the first sending stage should be $NT_{\text{PDU}}+T_{\text{prop}}+\sigma$. Now let us focus on analysis of the recovery stage. We denote random variable S_i as the transmission number of ith PDUs, until its first successful reception during the period of recovery stage. Under our channel assumption, S_i has a geometric distribution, and S_i is equal or greater than zero because some possible PDUs have been successfully transmitted to the receiver after the first sending stage. The retransmission spurts will repeat until all data PDUs have been transferred to the receiver correctly, so max (S_1, S_2, …, S_N) is the number of retransmission spurts. Thus, we define random variable $S_M=\max(S_1, S_2, \ldots, S_N)$.

In a similar way, we consider the minimum setting value for NAK timer is $2T_{prop} + RT_k + \theta$. As in practical situations, the sender could not retransmit the missing PDUs ordered by the NAK timely. For instance, the sender may have to delay the requested retransmission because preference must be given to the previously queued outbound data. However, it is difficult to estimate the queuing delay, so adding a random variable θ to the NAK timer is a simple approach under such an operational environment. So the expected retransmission time during the first retransmission spurt can be obtained as

$$\sum_{i=1}^{\infty}\left[i(2T_{prop} + RT_1 + \theta)\right]P_{er}^{i-1}(1 - P_{er}) = \frac{2T_{prop} + RT_1 + \theta}{1 - P_{er}} \quad (1)$$

So the expected time during the whole recovery stage is given as

$$E\left(\sum_{k=1}^{S_M}\frac{2T_{prop} + RT_k + \theta}{1 - P_{er}}\right) = \frac{E(S_M)\cdot(2T_{prop} + \theta)}{1 - P_{er}} + \frac{E\left(\sum_{k=1}^{S_M} RT_k\right)}{1 - P_{er}} \quad (2)$$

For the calculation of $E(S_M)$, we have

$$\begin{aligned} E(S_M) &= \sum_{m=1}^{\infty} P(S_M \geq m) \\ &= \sum_{m=1}^{\infty}\left[1 - P(S_M < m)\right] \\ &= \sum_{m=1}^{\infty}\left[1 - \prod_{i=1}^{N} P(S_i < m)\right] \\ &= \sum_{m=1}^{\infty}\left[1 - \left(1 - P_{ef}^{m}\right)^N\right] \end{aligned} \quad (3)$$

To complete the analysis, we need to obtain $E(S_M)$, in the premise of ensuring the certain accuracy of $E(S_M)$, we used finite summation (m sums from 1 to 100) as both an approximation and a lower bound to numerically compute them in the simulations.

Note that $E\left(\sum_{k=1}^{S_M} RT_k\right)$ is the expected time needed for transfer of the missing PDUs until all of them have been successfully received. Thus we have

$$E\left(\sum_{k=1}^{S_M} RT_k\right) = \sum_{i=1}^{N} E(S_i)T_{PDU} = N\cdot T_{PDU}\cdot\left(\frac{P_{ef}}{1 - P_{ef}}\right) \quad (4)$$

Therefore, considering together the first sending stage and the recovery stage, the expected file transfer time of a single transaction can be given as

$$T_{RE} = N\cdot T_{PDU} + T_{prop} + \sigma + \frac{E(S_M)\cdot(2T_{prop} + \theta)}{1 - P_{er}} + \frac{N\cdot P_{ef}\cdot T_{PDU}}{(1 - P_{er})\cdot(1 - P_{ef})} \quad (5)$$

3.3 Simulations and analysis

In this subsection, a large number of simulation experiments were made for the mathematical expressions derived in the previous subsection. Throughout our whole simulation experiments, in order to characterise the different operative conditions in space communications, the considered region of BER without forward error correction is between 10^{-4} and 10^{-8}, which is frequently faced in space communication [4, 19–21]. The simulation results of expected file transfer time of deferred NAK under different conditions are depicted in Figs. 2 and 3 using MATLAB. It is worth noting that the astronomical unit, equivalent to time (a.u., 1 a.u. = 480 s), is used. These two figures illustrate how the expected file transfer time is affected by variables such as the PDU error rate, one-way propagation delay, PDU transmission time, and the number of PDUs. As the actual value of θ depends on the level of performance desired and the practical PDU scheduling scheme adopted by the sender, which is in accordance with specific implementation (beyond the scope of this paper). For the simplicity and convenience of experimental simulations, we assume θ obeys uniform distribution in the interval [0, 1]. We also assume that P_{er} is less than P_{ef} (about two orders of magnitude) for the reason elaborated in Section 3.2. From Figs. 2 and 3, obviously, we can conclude that with the continuous increase of P_{ef}, the number of PDUs, the PDU transmission time, the expected file transfer time climbs up gradually.

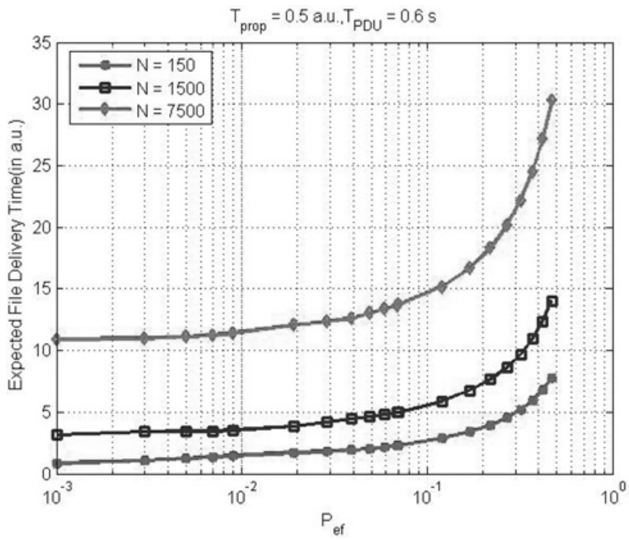

Fig. 2 *Expected file transfer time against P_{ef} under different PDUs number*

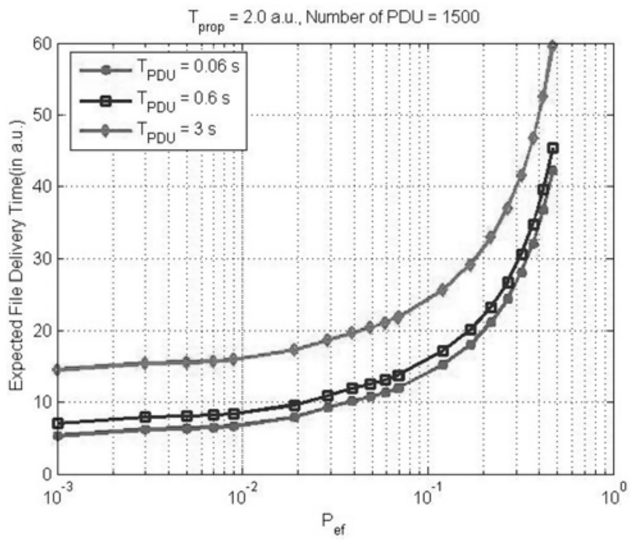

Fig. 3 *Expected file transfer time against P_{ef} under different transmission time of PDUs*

IET Commun., 2016, Vol. 10, Iss. 3, pp. 245–252

4 Improved strategy for deferred NAK

4.1 Benefit of increasing repeated times

Relative to the delay level of milliseconds in near-earth communication, delay of minutes, or even hours between interplanetary links dominates the file transfer time. The high BER of deep space requires multiple retransmissions of file data, thus enlarging the file delivery delay substantially. Thereby, increasing the success rate of the retransmission is an important way to shorten the delay. Sending one PDU multiple times consecutively would improve the success rate of retransmission. Suppose L_{PDU} is the length of the PDU, P_{e} is the BER of the link, and P_{ef} is the error or loss probability of the PDU; the relationship between them is

$$P_{\text{ef}} = 1 - (1 - P_{\text{e}})^{L_{\text{PDU}}} \tag{6}$$

Under the condition that the error probability of PDU is statistically independent during the whole file transfer process, P_{ef} is rewritten as below

$$P_{\text{ef}} = \left[1 - (1 - P_{\text{e}})^{L_{\text{PDU}}}\right]^{N_{\text{retra}}} \tag{7}$$

N_{retra} is the number of repeated times of a retransmitted PDU. Through the experiment, we can draw the conclusion that increasing N_{retra} can decrease P_{ef} significantly under certain limits.

4.2 Improved strategy description

According to the deferred NAK mode defined in the CFDP recommendation [3], the receiving entity saves all information about missing data until the EOF PDU is received. It then issues a NAK (if needed) to request retransmission of the missing data one time. As mentioned before, some PDUs have already been successfully delivered to the receiver entity before retransmission starts. Therefore, we only have to consider the treatment of missing PDUs to improve reliability in order to reduce the total file transfer time. The operation of our improved strategy is as same as the operation of recommended deferred NAK, except during the recovery stage, upon receiving a NAK, the sender entity promptly retransmits each missing PDU K times consecutively.

4.3 Improved model and theoretical analysis

We denote random variable S_i^K as the transmission number of the ith PDUs, until its first successful reception during the K-retransmission-based period. Under the aforementioned channel assumption, S_i^K still obeys a geometric distribution, and also $S_i^K \geq 0$. The spurts of retransmission will recur until all requested PDUs have been successfully transferred to the receiver; hence we define random variable $S_M^K = \max(S_1^K, S_2^K, \ldots, S_N^K)$, where $\max(S_1^K, S_2^K, \ldots, S_N^K)$ is the spurt number during the retransmission stage.

Recalling the setting rules of NAK timer, the expected retransmission time during the first retransmission spurt can be obtained as (1). So the expected file transfer time during the whole retransmission stage can be deduced as follows

$$E\left(\sum_{k=1}^{S_M^K} \frac{2T_{\text{prop}} + RT_k + \theta}{1 - P_{\text{er}}}\right) = \frac{E(S_M^K)\cdot(2T_{\text{prop}} + \theta)}{1 - P_{\text{er}}} + \frac{E\left(\sum_{k=1}^{S_M^K} RT_k\right)}{1 - P_{\text{er}}} \tag{8}$$

where

$$\begin{aligned} E(S_M^K) &= \sum_{m=1}^{\infty} P(S_M^K \geq m) \\ &= \sum_{m=1}^{\infty}\left[1 - P(S_M^K < m)\right] \\ &= \sum_{m=1}^{\infty}\left[1 - \prod_{i=1}^{N} P(S_i^K < m)\right] \\ &= \sum_{m=1}^{\infty}\left[1 - \left(1 - P_{\text{ef}}^{K(m-1)+1}\right)^N\right] \end{aligned} \tag{9}$$

Mind that $E\left(\sum_{k=1}^{S_M^K} RT_k\right)$ is the total expected time needed for transfer of all retransmitted PDUs until they are successfully delivered to the receiver. Thus we get

$$E\left(\sum_{k=1}^{S_M^K} RT_k\right) = \sum_{i=1}^{N} E(S_i^K) T_{\text{PDU}} = (K\cdot N\cdot T_{\text{PDU}})\left(\frac{P_{\text{ef}}}{1 - P_{\text{ef}}^K}\right) \tag{10}$$

So the number of PDUs delivered in the K-retransmission-based stage is

$$K\cdot N \frac{P_{\text{ef}}}{1 - P_{\text{ef}}^K} \tag{11}$$

Thus, the total expected file transfer time of a transaction can be given as

$$\begin{aligned} T_{KR} = N\cdot T_{\text{PDU}} + T_{\text{prop}} + \sigma + \frac{E(S_M^K)\cdot(2T_{\text{prop}} + \theta)}{1 - P_{\text{er}}} \\ + \frac{K\cdot N\cdot P_{\text{ef}}\cdot T_{\text{PDU}}}{(1 - P_{\text{er}})\cdot(1 - P_{\text{ef}}^K)} \end{aligned} \tag{12}$$

In general, we have a clear understanding of K-retransmission-based deferred NAK. We will validate the K-retransmission-based scheme in the following section and determine the desired value of K for retransmission.

4.4 Simulations and validation

This section includes both analytical and experimental evaluations of our improved strategy. We develop a theoretical model expressed in terms of the decrement of file transfer time by one extra retransmitted PDU to validate the performance of K-retransmission-based strategy.

Under the aforementioned analysis, we have constructed our typical simulation scenarios. We consider the operation of file-transfer over a direct link. The typical configurations of parameters are listed in Table 1.

First, we denote T_{first} as the time spent in the first sending stage and $T_{K-\text{retra}}$ as the time spent in K-retransmission-based stage, while T_{sd} corresponds to the retransmission time needed in recommended deferred NAK. Then we denote N_{first} as the number of PDUs delivered to the receiver during the first sending stage; obviously, $N_{\text{first}} = N$. $N_{K-\text{retra}}$ is defined as the number of PDUs

Table 1 Typical configuration of simulation parameters

Items	Scenario A GEO	Scenario B Earth to Moon	Scenario C Earth to Mars
file size		2 MB	
N		1000	
P_{ef}	0.001–0.5		
T_{prop}	0.12 s	1.352 s	1.5625 a.u. (750 s)
T_{PDU}	0.008 s	0.008 s	0.8 s

requested in the K-retransmission-based stage, while N_{sd} corresponds to the number of PDUs needed to be retransmitted in recommended deferred NAK.

Now we define the model as follows

$$R(K, sd) = -\frac{(T_{\text{first}} + T_{K-\text{retra}}) - (T_{\text{first}} + T_{sd})}{(N + N_{K-\text{retra}}) - (N + N_{sd})} = -\frac{T_{K-\text{retra}} - T_{sd}}{N_{K-\text{retra}} - N_{sd}} \tag{13}$$

Here, $R(K, sd)$ is a ratio that refers to the decrement of expected file transfer time by one extra retransmitted PDU delivered in the K-retransmission-based strategy, compared to the recommended deferred NAK mode. Without loss of generality, we only consider Earth to Mars scenario to assess its general performance. Tests are performed to show how the performance changes as the packet error probability rises from 0.001 to 0.5 and retransmission times K grow from 2 to 6. The experimental results are shown in Fig. 4.

It is clear that $R(K, sd)$ decreases with the increasing retransmission times. We see that the values of $R(2, sd)$ under different P_{ef} performs better than other cases. If the number of repeated transmissions is further increased, from 3 to 6, when P_{ef} ranging from 0.1 to 0.5, the curves will tend to be flat. Best results are gained when K is equal to 2, with the maximum decrement of file transfer time for the same PDUs number. More precisely, the contribution to lower the file transfer time at the cost of power is maximised in the case of $K=2$, compared to $K=1$ in the recommended deferred NAK mode.

In the light of the above-mentioned, we determine that the desired K is equal to 2, and name this improved strategy DR deferred NAK. As observed in Fig. 4, it is straightforward that the contribution becomes smaller and smaller with the gradually increasing K.

5 Performance comparisons and analysis

In this section, we derive the numerical expression of DR deferred NAK, and make a validation compared by numerical analysis and random simulation. We go through several measurements to compare DR deferred NAK with recommended deferred NAK.

5.1 Numerical analysis and random simulation

Jointly considering (8)–(12) in Section 4.3, the total expected file transfer time of a transaction based on DR deferred NAK strategy can be easily rewritten as

$$T_{\text{DR}} = T_{\text{prop}} + \sigma + N \cdot T_{\text{PDU}} + \frac{E(S_M^2) \cdot (2T_{\text{prop}} + \theta)}{1 - P_{\text{er}}} + \frac{2N \cdot P_{\text{ef}} \cdot T_{\text{PDU}}}{(1 - P_{\text{er}}) \cdot (1 - P_{\text{ef}}^2)} \tag{14}$$

where

$$E(S_M^2) = \sum_{m=1}^{\infty} \left[1 - (1 - P_{\text{ef}}^{2m-1})^N\right] \tag{15}$$

We have verified the mathematical expression of (14) with intensive random simulation under Scenario C. As shown in Fig. 5, the simulation results closely match the mathematically derived results through a large number of experiments. Nevertheless, the random simulation process not only spends more run time, but also needs more programming work and skill. The considered range of P_{ef} without any error correction measures is between 0.001 and 0.5, which is frequently faced in space communication.

5.2 Deferred NAK against DR deferred NAK

Under the simulation parameters presented in Table 1, we implement several experiments to compare DR deferred NAK with recommended deferred NAK under three typical scenarios. The comparison results based on the numerical analysis are shown in Fig. 6. As observed, we notice that the file transfer time ascends along with the increase of P_{ef} gradually. Although the DR deferred NAK mode makes great contribution for the GEO scenario with the change of P_{ef}, it remarkably suits the latter two scenarios, especially when P_{ef} slowly becomes larger. When P_{ef} is less than 0.01, our proposed algorithm almost matches the recommended deferred NAK, it mainly because few packets are lost in this case. However, when P_{ef} is larger than 0.01, our proposed algorithm shows its significance and more gain can be obtained. Obtained through the results, we can see DR deferred NAK outperforms the recommended deferred NAK distinctly. For one thing, the longer the distance is, the more advantage DR deferred NAK offers; for another, the higher the P_{ef} is, the more contributions DR deferred NAK provides. It is the DR deferred NAK strategy of reducing the number of retransmissions that shortens the total file transfer time of file transfer noticeably. Just as discussed in [4] (in Section 2.3, proposition 1), $E(S_M^2)$ increases in logarithmic order with N. The expected file transfer time expressed in (14) has two terms that

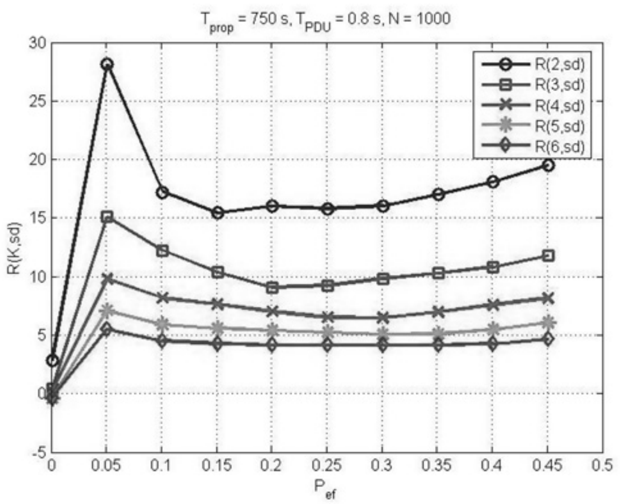

Fig. 4 *Validation of K-retransmission-based deferred NAK under Earth to Mars scenario: file size = 2 MB, transmission rate = 20 kb/s, and one way propagation delay = 750 s*

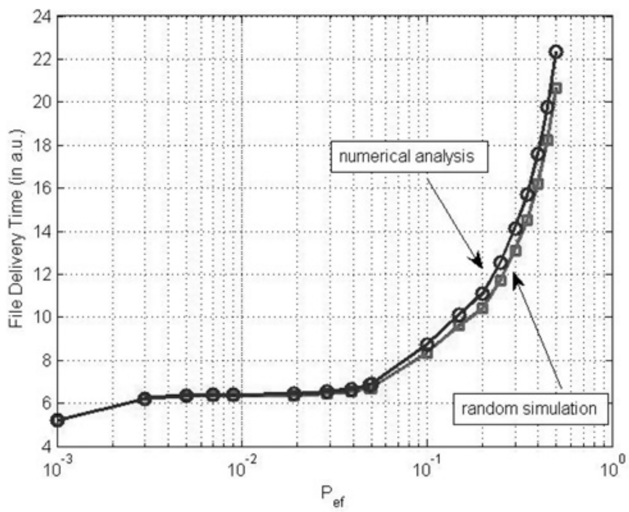

Fig. 5 *DR deferred NAK: analytic and simulation under Earth to Mars scenario: file size = 2 MB, transmission rate = 20 kb/s, and one way propagation delay = 750 s*

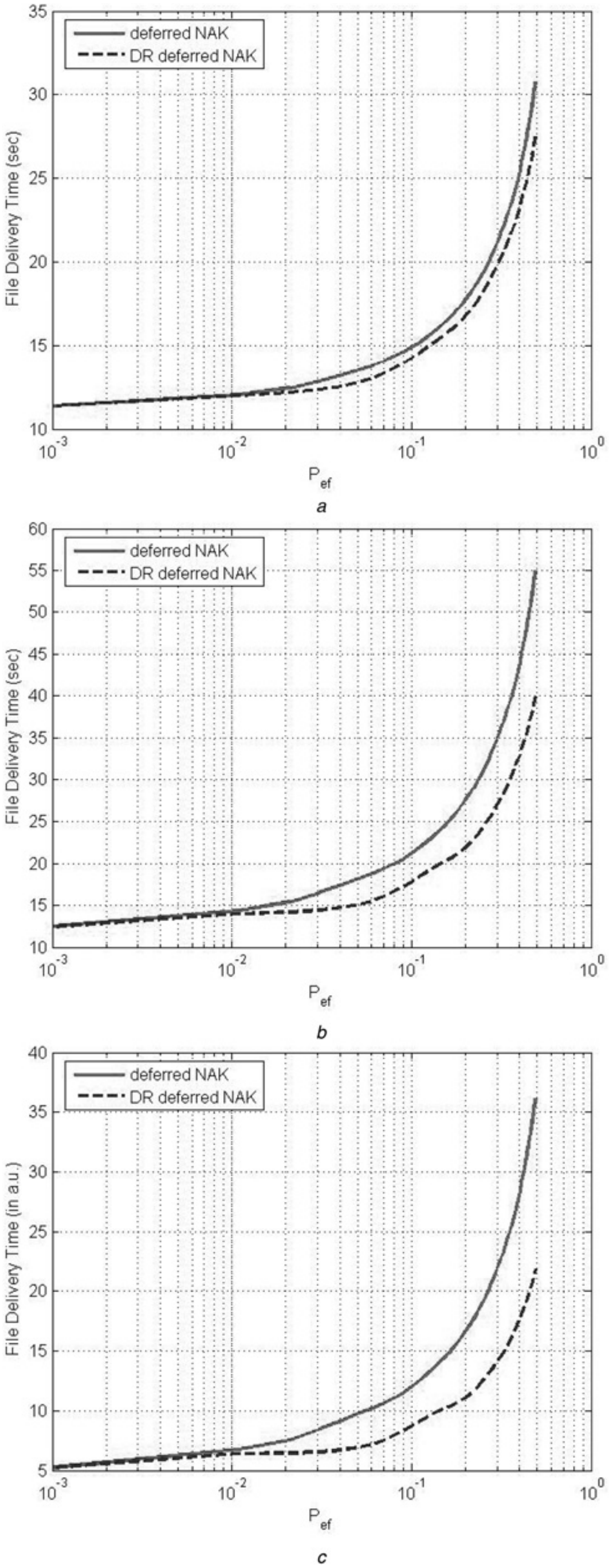

Fig. 6 *Comparison between deferred NAK and DR deferred NAK*

a GEO
b Earth to Moon
c Earth to Mars

increase with N and another one term that has the factor $E(S_M^2)$. For long-haul propagation delay, the product of multiplicative factor $E(S_M^2)$ and one-way propagation delay is much bigger than that of PDU number and the PDU transmission time. In this case, with the increase of PDU number, the expected file transfer time is initially ruled by the term growing with N logarithmically, and later the growth order becomes linear for larger N. Once N is fixed, the expected file transfer time is completely dominated by the multiplicative factor and long one-way propagation delay. In addition, the growth order is always ruled by N when the one-way propagation delay and the PDU transmission time are almost the same length.

Last but not least, as previously mentioned, the DR deferred NAK is superior to the recommended deferred NAK. It especially suits scenarios characterised by extremely long propagation delay, higher packet error rate, or both – common in deep space communication.

6 Conclusion and future work

This paper addressed the model and optimisation for the deferred NAK mode in CFDP. An improved strategy for the deferred NAK mode in CFDP is provided and compared with the recommended deferred NAK under three typical scenarios. A large number of numerical and simulation results show that the proposed strategy is superior to the recommended deferred NAK, and is especially suitable for the scenarios characterised by extremely long propagation delay, higher packet error rate, which commonly faced in deep space environment.

At present, CFDP is one of the most important applications that the DTN architecture has to support. CFDP can run over BP/LTP, with LTP providing the retransmission-based reliability. That's why the open source distribution of Interplanetary Overlay Network (ION) contains fully conformant implementations of both CFDP and Asynchronous Message Service (AMS) besides the core DTN protocols. Needless to say, CFDP is very valuable and definitely worth using in deep space communication now and in the future.

Future work will emphasise on the protocol hierarchy that ION software offers. We will also tackle a deeper analysis of BP custody transfer and LTP retransmission to realise the reliability of file transfer over extremely long distance.

7 Acknowledgments

This work was supported by the National Natural Sciences Foundation of China (NSFC) under grant nos. 61032003 and 61102083.

8 References

1 De Cola, T., Paolini, E., Liva, G., *et al.*: 'Reliability options for data communications in the future deep-space missions', *Proc. IEEE*, 2011, **99**, (11), pp. 2056–2074
2 De Cola, T., Marchese, M.: 'Reliable data delivery over deep space networks: benefits of long erasure codes over ARQ strategies', *IEEE Wirel. Commun.*, 2010, **17**, (2), pp. 57–65
3 Consultative Committee for Space Data Systems: 'CCSDS file-delivery protocol (CFDP), recommendation for space data system standards'. CCSDS 727.0-B-4, Blue Book, NASA, Washington, D.C., January 2007
4 Lee, C.D., Baek, W.: 'Expected file-delivery time of deferred NAK ARQ in CCSDS file-delivery protocol', *IEEE Trans. Commun.*, 2004, **52**, (8), pp. 1408–1416
5 Baek, W., Lee, C.D.: 'Analysis of CCSDS file-delivery protocol: immediate NAK mode', *IEEE Trans. Aerosp. Electron. Syst.*, 2005, **41**, (2), pp. 503–524
6 Gao, J.L., SeGuí, J.S.: 'Performance evaluation of the CCSDS file delivery protocol – latency and storage requirement'. Aerospace Conf., Edinburgh, Scotland, May 2005, pp. 1300–1312
7 De Cola, T., Ernst, H., Marchese, M.: 'Performance analysis of CCSDS File Delivery Protocol and erasure coding techniques in deep space environments', *Comput. Netw.*, 2007, **51**, (14), pp. 4032–4049
8 Papastergiou, G., Psaras, I., Tsaoussidis, V.: 'Deep-space transport protocol: a novel transport scheme for space DTNs', *Comput. Commun.*, 2009, **32**, (16), pp. 1757–1767
9 Zhu, W.L., Zhai, L.J.: 'DSFTP: deep-space file transfer protocol', *J. Astronaut.*, 2011, **32**, (3), pp. 640–644 (in Chinese)
10 Jiao, J., Guol, Q., Zhang, Q.: 'Packets interleaving CCSDS file delivery protocol in deep space communication', *Aerosp. Electron. Syst. Mag.*, 2011, **26**, (2), pp. 5–11

11 Wang, Y., Li, J., Chen, G., *et al*.: 'Repeated sending file delivery protocol in satellite networking communication', *J. Syst. Eng. Electron.*, 2012, **23**, (6), pp. 815–823

12 Cerf, V., Burleigh, S., Hooke, A., *et al*.: 'Interplanetary Internet (IPN): architectural definition'. Internet draft, draft-irtf-ipnrg-arch-00. txt, May 2001

13 Burleigh, S., Hooke, A., Torgerson, L., *et al*.: 'Delay-tolerant networking: an approach to interplanetary Internet', *IEEE Commun.*, 2003, **41**, (6), pp. 128–136

14 Koutsogiannis, E., Papastergiou, G., Tsaoussidis, V., *et al*.: 'Evaluation of CCSDS file delivery protocol over Delay Tolerant Networks'. Int. Workshop on Satellite and Space Communications, New York, USA, September 2009, pp. 381–384

15 De Cola, T.: 'A protocol design for incorporating erasure codes within CCSDS: The case of DTN protocol architecture'. 5th Advanced Satellite Multimedia Systems Conf. and the 11th Signal Processing for Space Communications Workshop, Cagliari, Italy, September 2010, pp. 68–73

16 Samaras, C.V., Tsaoussidis, V., Peccia, N.: 'DTTP: a delay-tolerant transport protocol for space internetworks'. 2nd ERCIM Workshop on eMobility, Tampere, Finland, May 2008, pp. 3–14

17 Consultative Committee for Space Data Systems: 'CCSDS File Delivery Protocol (CFDP) – Part 1: Introduction and overview'. CCSDS 720.1-G-3, Green Book, NASA, Washington, D.C., April 2007

18 Wang, R., Shrestha, B.L., Wu, X., *et al*.: 'Unreliable CCSDS file delivery protocol (CFDP) over cislunar communication links', *IEEE Trans. Aerosp. Electron. Syst.*, 2010, **46**, (1), pp. 147–169

19 Schnurr, R., Rash, J., Hogie, K., *et al*.: 'NASA/GSFC space internet: extending internet technology into space'. NASA/GSFC Space Internet – NRO Technical Seminar, NASA Goddard Space Flight Center, May 2001

20 Wang, R., Taleb, T., Jamalipour, A., *et al*.: 'Protocols for reliable data transport in space internet', *IEEE Commun. Surveys Tutor.*, 2009, **11**, (2), pp. 21–32

21 Bezirgiannidis, N., Burleigh, S., Tsaoussidis, V.: 'Delivery time estimation for space bundles', *IEEE Trans. Aerosp. Electron. Syst.*, 2013, **49**, (3), pp. 1897–1910