

QJ

中华人民共和国航天行业标准

FL 1640

QJ 20750—2018

空间数据系统文件传输协议

Space data system file delivery protocol

2018—01—18 发布

2018—05—01 实施

国家国防科技工业局 发布

目 次

前言	III
引言	IV
1 范围	1
2 规范性引用文件	1
3 术语和定义	1
4 缩略语	2
5 协议概述	3
5.1 一般要求	3
5.2 结构组成	3
5.3 协议特性	4
5.4 交互约定	7
6 业务定义	9
6.1 用户接口业务	9
6.2 原语概述	9
6.3 参数说明	10
6.4 底层通信系统业务	12
6.5 业务原语	12
7 协议规程	18
7.1 核心规程	18
7.2 扩展规程	30
8 协议数据单元格式	32
8.1 位序号约定	32
8.2 一般要求	33
8.3 文件指令 PDU	35
8.4 文件数据 PDU	38
8.5 TLV 参数	38
9 用户操作	41
9.1 预留消息格式	41
9.2 代理操作	41
9.3 目录操作	44
9.4 远程状态报告操作	45
9.5 远程挂起操作	47

9.6 远程恢复操作	48
9.7 存储与中继转发覆盖操作	49
10 业务类别	56
10.1 业务类别定义	56
10.2 类别 1——不可靠传输的功能	56
10.3 类别 2——可靠传输的功能	58
10.4 类别 3——通过一个或多个级联中转节点的不可靠传输的功能	60
10.5 类别 4——通过一个或多个级联中转节点的可靠传输	62
11 管理信息库	65
附录 A（资料性附录） 校验和计算示例	67

前 言

本标准的附录A为资料性附录。

本标准由中国航天科技集团有限公司提出。

本标准由中国航天标准化研究所归口。

本标准起草单位：北京空间飞行器总体设计部。

本标准主要起草人：何熊文、毛一岚、刘崇华、郭 坚、佟 玲、顾 明、阎 冬、刘伟伟、程博文、张翠涛、詹盼盼。

引 言

本标准参照了国际标准化组织发布的国际标准《空间数据与信息传输系统——CCSDS文件传输协议》(ISO 17355:2007),即空间数据咨询委员会(CCSDS)发布的推荐建议书CCSDS 727.0-B-4,技术内容一致,文字上做了简化处理。

空间数据系统文件传输协议

1 范围

本标准规定了空间数据系统文件传输协议（以下简称 CFDP）的业务定义、协议规程、协议数据单元格式、用户操作、业务类别和管理信息库等。

本标准适用于空间到地面、地面到空间、空间与空间之间的文件传输，可对航天器文件存储器进行上传、下载和控制等操作。

2 规范性引用文件

下列文件中的条款通过本标准的引用而成为本标准的条款。凡是注日期的引用文件，其随后所有的修改单（不包含勘误的内容）或修订版均不适用于本标准，然而，鼓励根据本标准达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件，其最新版本适用于本标准。

- GJB 727 航天测控系统术语与缩略语
- QJ 20743 空间数据系统术语和缩略语
- QJ 20746 遥测空间数据链路协议
- QJ 20747 遥控空间数据链路协议
- QJ 20748 高级在轨系统空间数据链路协议
- QJ 20749 空间数据系统空间包协议

3 术语和定义

GJB 727 以及 QJ 20743 确立的以及下列术语和定义适用于本标准。

3.1

协议实体 protocol entity

一个 CFDP 的具体实现实例，类似于一个互联网协议的主机。

3.2

元数据 metadata

在两个协议实体之间交换的除被传输文件内容之外的其他数据，包含与该文件或文件传输相关的信息。

3.3

文件传输单元 file delivery unit

在两个协议实体之间传输的文件及与该文件相关的元数据的组合。

3.4

协议数据单元 protocol data unit

在协议实体之间传输的独立的、有界的、能够自标识的数据，包含文件数据和文件指令两种类型。

3.5

事务 transaction

在两个协议实体之间进行的一次可唯一标识的 FDU 端对端传输，包含多个 PDU 的发送和接收。

3.6

文件存储器 filestore

用来存储文件的媒介。

3.7

偏移量 offset

在一个文件中某字节的偏移量定义为在其之前的文件的字节数。

3.8

文件传输进度 file delivery progress

一个 PDU 的文件数据内容首个字节在整个文件中的偏移量与该 PDU 文件数据内容长度之和。

3.9

发送进度 transmission progress

一个文件事务中发送端发出的所有文件数据 PDU 的文件传输进度最大值，为当前发送进度。

3.10

接收进度 reception progress

一个文件事务中接收端接收的所有文件数据 PDU 的文件传输进度最大值，为当前接收进度。

4 缩略语

下列缩略语适用于本标准。

ACK——Acknowledgement，肯定应答；

AOS——Advance Orbiting Systems，高级在轨系统；

APID——Application Process Identifier，应用过程标识；

CCSDS——Consultative Committee for Space Data Systems，空间数据系统咨询委员会；

CFDP——CCSDS File Delivery Protocol，CCSDS 文件传输协议；

EOF——End of File，文件结束；

FDU——File Delivery Unit，文件传输单元；

LV——Length-Value，长度-值；

MIB——Management Information Base，管理信息库；

MSB——Most Significant Bit，最高位；

MTU——Message to User，发往用户的消息；

NAK——Negative Acknowledgement，否定应答；

NCC——Network Control Center，网络控制中心；

OSI——Open Systems Interconnection，开放系统互连；

PDU——Protocol Data Unit，协议数据单元；

SAP——Service Access Point，业务访问点；

SDU——Service Data Unit, 业务数据单元;
 SFO——Store and Forward Overlay, 存储与中继转发;
 TC——Telecommand, 遥控;
 TCP——Transmission Control Protocol, 传输控制协议;
 TLV——Type-Length-Value, 类型-长度-值;
 TM——Telemetry, 遥测;
 UDP——User Datagram Protocol, 用户数据报协议;
 UT——Unitdata Transfer, 单元数据传输;
 UT-SAP——Unitdata Transfer Service Access Point, 单元数据传输业务访问点。

5 协议概述

5.1 一般要求

CFDP 可用于各种不同类型的任务环境, 从简单的单个低轨航天器, 到由多个地面设施、轨道器、着陆器及多条通信链路组成的复杂系统。在简单的模式下, 协议的核心规程提供了在单一链路上的文件传输能力, 而在包含多条链路和不同应用需求的任意网络中, 协议的扩展规程提供了“文件存储与中继转发”功能。协议与具体的数据存储实现技术无关, 对所传输的信息也不做约定。

CFDP 需要底层通信系统支持, 推荐使用 QJ 20749, 并与 QJ 20748、QJ 20746、QJ 20747 相配合。

5.2 结构组成

5.2.1 组成

CFDP 协议组成要素图如图 1 所示。

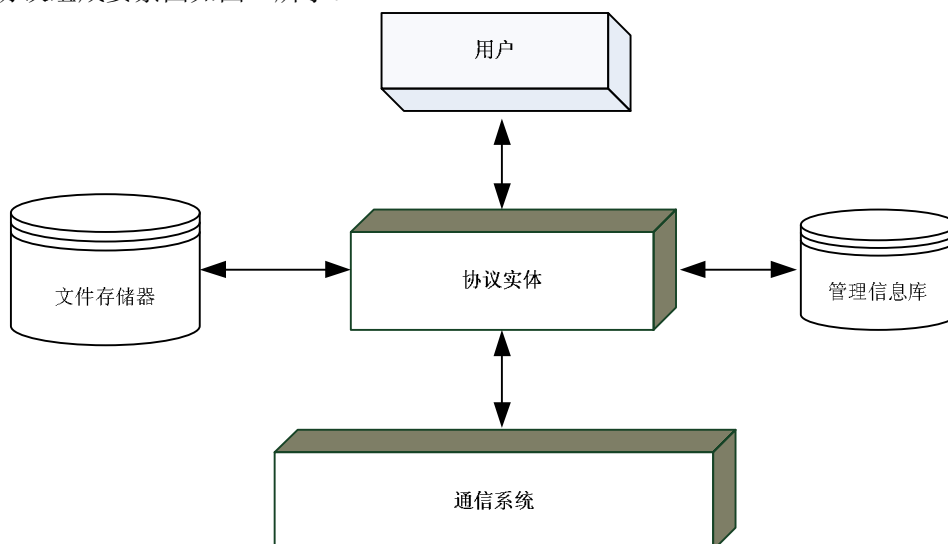


图 1 协议组成要素图

5.2.2 用户

用户是一个软件任务, 可以在有人干预或无人干预下运行。CFDP 根据用户的请求运行。用户使用第 6 章中定义的业务原语与协议进行交互操作。每个协议实体最多只有一个用户。在一些情况下可能无用户, 尤其是当一个实体总是作为扩展规程(进行存储与中继转发操作)的中转节点时, 则无需用户。

5.2.3 协议实体

协议实体由核心规程和可选的扩展规程组成。核心规程支持文件直接传输，用于单跳网络环境。扩展规程支持在有路由能力的多跳网络上进行间断或直接传输。协议只呈现给用户一个单一的业务接口，扩展规程的运行取决于服务质量和网络的多跳传输能力。

5.2.4 文件存储器

CFDP 用于在存储器之间进行文件复制操作，并假定所有协议实体都具有访问本地存储器的能力，任何文件或文件存储器都可以用一个标准的表示方法“虚拟文件存储器”进行表述。“虚拟文件存储器”具有一个标准的属性集，用于管理文件分发过程，在具体实现时，必须映射到由软硬件构成的物理文件存储器上。

5.2.5 通信系统

协议假定有一个被称为“UT层”的底层通信系统。在一个给定协议地址空间内的所有协议实体都可访问该层。为了使协议的适用范围更广泛，对所需 UT 层提供的服务应尽可能简单，并可通过一个业务访问点提供给任何一个协议实体。

由于底层服务的物理实现具有多样性，所以 CFDP 没有用到寻址需求以外的底层协议。寻址需求是指在底层通信系统范围内，根据管理信息库中的信息将协议实体的名字映射为“UT 地址”。

考虑到协议的相对独立性，对多个文件事务的支持、序列检查和差错检测均是由 CFDP 自身提供。PDU 的长度信息组织在协议 PDU 导头中，可独立于底层协议并且简化协议处理。

5.2.6 管理信息库

管理信息库 (MIB) 规定协议可供配置的管理参数默认值，管理参数包括本地配置参数以及远程配置参数。管理参数中的部分参数还用于在本地用户与本地的协议实体之间，以及本地协议实体与远端协议实体之间传递，以进行文件传输。MIB 的规定详见第 11 章。

5.3 协议特性

5.3.1 特性说明

协议支持将文件从一个文件存储器移动到另一个文件存储器，这两个文件存储器各自处于单独的数据系统中，通常由空间链路连接。每一个文件都关联了元数据。元数据的大小可变，包含了与文件或文件传输相关的信息，如文件名、文件存储请求或 MTU 等。文件和所有相关元数据的组合被称为一个 FDU，但也允许某个 FDU 只包含元数据。

FDU 传输由协议实体来实现，而每一个实体对应访问一个文件存储器。在两个协议实体之间、端到端传输单个 FDU 称为一个事务。启动事务将触发发送端和接收端两个实体间的文件复制操作。

两个协议实体之间的协议规程示意图如图 2 所示。在文件复制操作中，发起方协议实体为源端，接收方协议实体为目的端。当源端与目的端无法直接连接时，扩展规程自动执行多个文件复制操作，分别为：源端和首个中转节点间的复制操作、必要时在中转节点之间的复制操作、最后的中转节点与目的端之间的复制操作。扩展规程中每一次操作都与核心规程定义的文件复制操作相同。

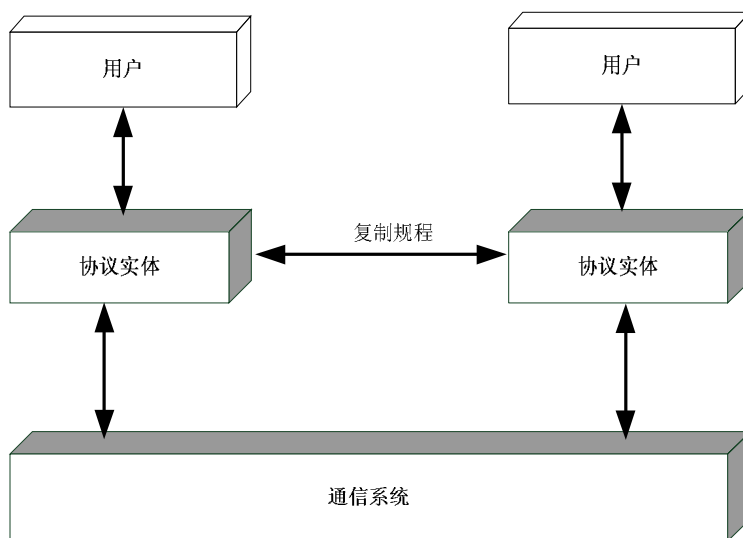


图 2 CFDP 协议规程示意图

操作选取“无应答模式”还是“应答模式”决定了传输的可靠性。在无应答模式下，数据传输失败不会报告给发送方，因此未能正确接收的数据不会被重传，无法保证文件接收完整性。在应答模式下，接收方会告知发送方未正确接收的文件段或辅助数据信息。这些未能正确接收的数据将被重传，以保证文件传输的完整性。

每个事务对应一个从源端到目的端的 FDU 复制过程。在该过程中，可能执行一些辅助的操作，如取消、挂起、恢复和报告。

当满足下列条件之一时，本次事务被终止：

- a) 文件已成功传输；
- b) 发送或接收实体禁止该事务；
- c) 检测到故障后取消事务；
- d) 源端或目的端 CFDP 用户已取消事务。

5.3.2 用户操作

用户操作指使用由本地协议实体提供的业务来触发远端协议实体启动附加事务。用户操作通过使用协议中的 MTU 功能向远端用户发出“命令”，从而使远端用户的本地实体启动一个事务。

CFDP 定义了五种标准的用户操作：

- a) 代理操作：用于发起从远端协议实体传送文件给其他用户的操作，接收文件的用户可以是代理操作的发起方本身（在这种情况下，即代理操作实现了文件“获取”功能）或某个第三方协议实体用户；在代理操作中传输的 FDU 通常包含一个文件，但也可仅包含元数据，如只含有给远程协议用户的文件存储指令或 MTU；
- b) 目录操作：用于请求传输远端用户本地存储器中指定目录的内容列表；
- c) 远程状态报告操作：用于向远端实体请求某个事务的状态报告；
- d) 远程挂起操作：用于向远端实体请求指定事务的挂起；
- e) 远程恢复操作：用于向远端实体请求指定事务的恢复。

用户操作的详细说明见第 9 章。

5.3.3 寻址

在由协议实体组成的网络中，每个协议实体被分配一个唯一的名称，称为“实体标识”，该标识需包含在每个PDU中的导头中，使得FDU能通过一系列的中转传输。由于每个协议实体都访问确定的文件存储器，所以一个协议实体的身份也同时指定了其对应的文件存储器。每个协议实体可根据本地MIB中的信息进行地址查找，完成协议实体名称和UT地址名称之间的转换。UT地址可以是互联网地址、射频设备缓存、APID、虚拟信道号或其他类型的实现形式。

5.3.4 存储与中继转发

当运行扩展规程时，某个PDU的发送方和（或）接收方可以是一个“中转”节点实体。当PDU的源端与最终目的端之间没有直接可达的连接，而只有通过中转实体间接连接时，需要使用扩展规程。此时节点实体可直接连接PDU的最终目的端或者下一个中转节点实体，在这样的一系列节点实体组成的链中，最后一个中转节点实体必须直接与最终目的端相连。

实现中要解决节点实体如何以及何时确定向其下一个目标转发PDU的问题。一般来说，期望尽可能快地转发每个PDU，而不是等到整个FDU已经完全从前一站被传输到本站后才开始向后传输，这种方法可以减少完整数据端到端传输所需要的总时间。然而，在实践中，协议实体之间的无线链路往往是不连续的，导致不能实现直接转发。在这种情况下，中转节点必须在一些持久介质中存储PDU，作为所传文件的中间副本，直至其转发完成。

因此，整个端到端的事务可能包括多个在相邻实体间连续执行的核心规程，其中一些是由扩展规程发起的，而不是本事务的用户发起。在此情况下，扩展规程起到了类似用户的角色和作用。但这种区别并不会影响到核心规程的运行。

当一个文件复制操作中的发送方是该文件的源实体，但接收方是一个中转节点实体，而不是文件的目的地实体时，发送方实体收到中转节点的“文件已成功接收通知”后，会发出一个TRANSFER-CONSIGNED.indication原语（而不是一个TRANSACTION-FINISHED.indication原语）给用户程序。这将通知发送方用户程序，文件已转移到第一个中转节点托管。如果发送方实体为支持重发，将原始文件作为重发缓存而保护，使其不能被删除或修改，则在收到此通知后即可解除保护。

如前所述，当文件复制操作的接收方是一个中转节点实体时，它可以选择是否等待整个FDU传输完成再开始将文件复制到目的端（或下一中转节点）。立即（渐进）转发具有最小化延迟的效果，使得目的端可以尽快获取文件的一部分。

当建立连接后，存储的PDU按照何种顺序进行发送是具体实现相关的问题。可使用可选的“流标签”选项（对应原语参数中的flow label），关联被传输的数据，以辅助协议实体确定如何和何时发出数据。

5.3.5 重传策略

根据任务需要和传输能力，CFDP支持可选的服务质量，包括无应答选项（不保证文件传输的完整性，不纠正错误，只检测错误并丢弃出错数据）、有重传恢复策略的应答选项。在应答选项下，接收方可以设置几个子选项，如设定NAK的发出时间，可选择立即发出或延迟发出（即NAK暂存起来直到预定的传输完成时刻后再发出）。重传策略具体详见第7章。

5.3.6 虚拟文件存储器

虚拟文件存储器的概念提供了从协议的文件存储指令到实际文件存储操作的映射。在具体实现中，这种映射操作由本地程序解决。协议允许使用元数据PDU附加的TLV格式文件存储指令。

为实现互操作，本标准假设虚拟文件存储器最少具备以下功能：

- a) 创建文件；
- b) 删除文件；
- c) 重命名文件；
- d) 追加文件；
- e) 替换文件；
- f) 创建目录；
- g) 删除目录；
- h) 列出目录。

在某些情况下，能够识别文件的记录边界是对协议实现有利的。如果这个选项被使用，文件存储器必须有能力区分此类文件和那些被处理为字节流的文件。

5.4 交互约定

5.4.1 复制操作

复制操作如图 3 所示。（本标准中所有图中虚线均表示可选项。）

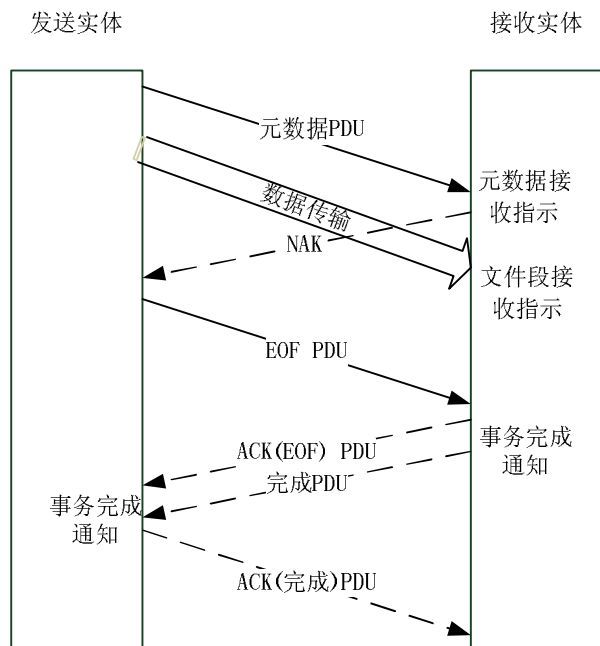


图 3 复制操作过程与事件顺序

复制操作开始于元数据的传输，其后跟随着数据（文件段）传输，最后以一个 EOF PDU 指示文件结束。接收实体收到某个文件段可选择使用业务原语应答，而收到元数据必须强制通知用户。

对于可靠事务，需添加许多功能。NAK 用于请求对丢失数据的重传。ACK（EOF）用于表示已收到 EOF PDU。由于丢失数据的 NAK 可能处于 EOF 之后，为了使发送端获取传输状态，当所有文件数据均已接收完成且成功组装后，接收端需要产生完成 PDU。完成 PDU 的传输由对完成 PDU 的 ACK 加以保证。

当文件传输任务完成后，发送实体和接收实体发出事务完成通知。事务完成通知用于告知发送应用程序文件传输任务已经完成，是一个抽象的内部事件，并不通知用户。

5.4.2 提交请求过程

用户发起一个 PUT.request 到源实体启动事务提交请求。该 PUT.request 触发 FDU 传输过程，以及第一次(可能也是仅有的一次)复制操作。该 PUT.request 还触发实体对该次事务生成一个“事务 ID”，上报给发送应用程序，使得本次事务能够在此后过程中被应用程序清楚识别。完成传输时源实体将触发一个 TRANSACTION-FINISHED.indication 通知，此操作对于目的实体是可选的。提交请求如图 4 所示。

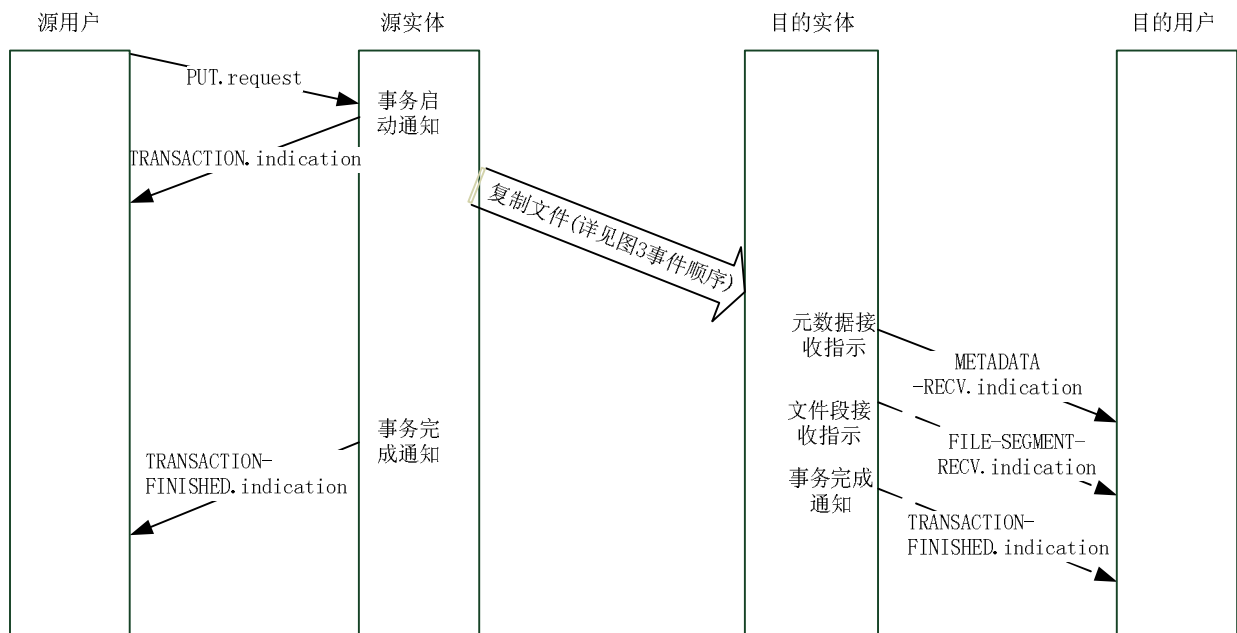


图 4 提交请求过程与事件顺序

5.4.3 文件存储转发过程

源实体和目的实体有可能无法直接连接，在这种情况下，可能需要在源实体、目的实体以及一个或多个中转实体之间进行一系列点对点的（发送方对接收方）的 PDU 交互操作以完成事务，此过程如图 5 所示。

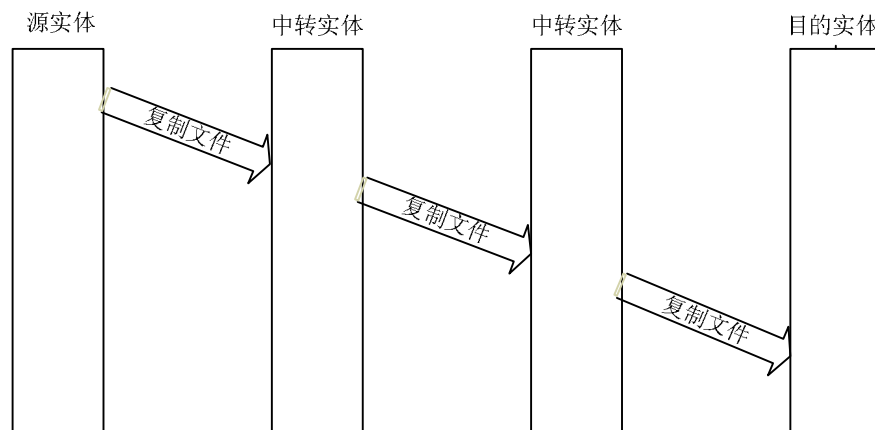


图 5 文件存储转发过程与事件顺序

5.4.4 代理提交请求过程

当需要传输的源文件不在本地文件存储器，提出此请求的用户必须先发起一个提交请求 PUT.request，提交请求包含一个给文件所在地远端协议实体的用户（即“代理用户”）的消息，此消息

请求由代理用户发起 PUT.request，并告知代理用户传输的目的端。其后，代理用户发起 PUT.request，启动第二个事务，传输的目的端可能就是提出初始 PUT.request 请求的用户本身（在这种情况下，代理操作即实现了本地用户从远程“获取”文件的功能）或某个第三方协议实体用户。在第二个事务完成后，代理用户发起第三个事务，通知原请求用户代理操作已经完成。代理提交请求过程如图 6 所示。

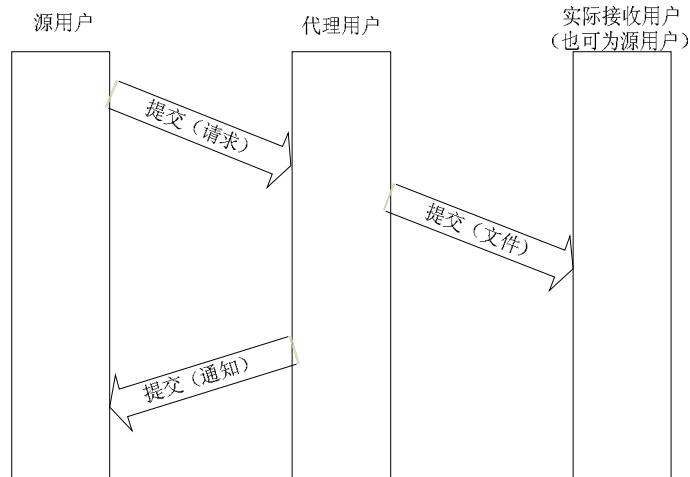


图 6 代理提交请求过程与事件顺序

6 业务定义

6.1 用户接口业务

协议提供给用户使用的业务包括：

- a) 发起文件存储器之间的文件传输；
- b) 启动远程文件存储器的文件存储操作；
- c) 接收与当前事务操作相关的事件；
- d) 查询当前事务状态信息；
- e) 发送或接收与当前事务相关的消息；
- f) 挂起、恢复或取消当前事务的传输。

协议实体应支持任意数量、处于发送或接收不同阶段的事务同时进行。实现上必须能够支持在前一个事务完成之前接收 PUT.request 原语并启动一个新的事务。事务的完成包括源实体初始发送的文件数据已经部分或全部被目的实体接收，且两个实体间任何所需的应答及重传数据均被接收。因此支持并行事务并不意味着必须在一个或多个事务正在初始传输时即开始新事务的初始传输（但也不排除这种情况）。

6.2 原语概述

6.2.1 CFDP 接收用户的以下请求原语：

- a) PUT.request;
- b) REPORT.request;
- c) CANCEL.request;
- d) SUSPEND.request;
- e) RESUME.request。

6.2.2 CFDP 向用户提交以下指示原语：

- a) TRANSACTION.indication;
- b) EOF-SENT.indication;
- c) TRANSACTION-FINISHED.indication;
- d) METADATA-RECV.indication;
- e) FILE-SEGMENT-RECV.indication;
- f) REPORT.indication;
- g) SUSPENDED.indication;
- h) RESUMED.indication;
- i) FAULT.indication;
- j) ABANDONED.indication;
- k) EOF-RECV.indication。

6.2.3 在使用扩展规程时，本业务还使用以下指示原语：TRANSFER-CONSIGNED.indication。

6.3 参数说明

6.3.1 “destination CFDP entity ID” 参数用于唯一标识接收 FDU 的目的协议实体。

6.3.2 “source CFDP entity ID” 参数用于唯一标识发送 FDU 的源协议实体。

6.3.3 “source file name” 参数规定如下：

- a) 必须包含需要被复制的文件在源实体文件存储器的完整路径名；
- b) 在提交的 FDU 中只包含元数据时，如 MTU 或单独的文件存储请求时，省略本参数。

6.3.4 “destination file name” 参数规定如下：

- a) 必须包含需要被复制的文件在目的实体文件存储器存放的完整路径名；
- b) 在提交的 FDU 中只包含元数据时，如 MTU 或单独的文件存储请求时，省略本参数。

6.3.5 “segmentation control” 参数规定如下：

- a) 应指明被传输的文件是否被分段为字节序列或是变长记录序列；
- b) 当本地或远程文件名被省略时本参数也被省略。

6.3.6 “transaction ID” 参数用于唯一标识一个 FDU 传输实例，应包含源实体 ID 和该实体确立的事务序列号。

注1：在任何情况下，源实体所在的协议地址空间中，任意一个 transaction ID 都是唯一的。

注2：为适应不同的任务要求，实体 ID 和事务序列号的长度可变。需制定一个支持不同参数长度的互操作方案。

6.3.7 “condition code” 参数用于在某些传输状态变化时提供附加信息。

6.3.8 “status report” 参数指明文件事务的状态，其格式和范围由具体实现决定，应当包括以下信息：

- a) 事务是否已经结束、取消、挂起或正在进行中；
- b) 已知的被接收实体成功接收的 FDU 长度；
- c) 已知的被发送实体发出的 FDU 长度。

6.3.9 “fault handler overrides” 参数规定如下：

- a) 如果选取该故障处置选项，将在检测到一种或多种故障状态时采取相应措施；每种故障处置包括对应的一类故障状态及处理此情况的处置措施，措施可为：

- 1) 取消文件事务;
 - 2) 挂起事务;
 - 3) 忽略故障但提交 FAULT.indication 原语给本地用户;
 - 4) 放弃事务并提交 ABANDONED.indication 原语给本地用户。
- b) 故障处置措施对应的故障状态可见第 8 章表中状态码的子集, 在列出的状态中“无错误”、“收到 SUSPEND.request 请求”和“收到 CANCEL.request 请求”并不属于故障。
- c) 对于服务请求参数中故障处置未覆盖的故障类型, 应按照 MIB 中设置的缺省方式处置。
- 6.3.10** 若实现上选取了可选的“transmission mode”参数, 则取代缺省传输模式, 其值可设为“应答”或“无应答”。
- 6.3.11** 若实现上选取了可选的“Message to User”参数, 该参数将在事务起始时即被传输给目的端用户。这些参数在“代理操作”、“远程状态报告操作”和“目录操作”相关章节中定义, 用于 CFDP 事务的远程启动。
- 6.3.12** “filestore requests”参数规定如下:
- a) 若选取了可选的“filestore requests”参数, 请求将在事务起始时即被传输给目的实体, 并在全部数据传输完成后由目的实体执行。
 - b) 每个“filestore requests”参数应包含以下子参数:
 - 1) 操作;
 - 2) 文件名 1;
 - 3) 文件名 2 (供“操作”子参数的某类操作使用)。
 - c) “操作”子参数有以下值:
 - 1) 创建文件;
 - 2) 删除文件;
 - 3) 重命名文件 (使用文件名 2);
 - 4) 追加文件 (使用文件名 2);
 - 5) 替换文件 (使用文件名 2);
 - 6) 创建目录;
 - 7) 删除目录;
 - 8) “拒绝”文件 (类似删除文件; 但若文件不存在时不会导致操作失败);
 - 9) “拒绝”目录 (类似删除目录; 但若目录不存在时不会导致操作失败)。
 - d) 如果子参数“文件名 1”或“文件名 2”的值与“目标文件名”相同, 则文件存储请求中的文件即为本次最新传输的文件。
 - e) 当同一事务中存在多个“filestore requests”时, 源实体应当维护这些请求的顺序, 并按顺序发给目的实体, 目的实体也按此顺序执行。
- 6.3.13** “filestore responses”参数表明指定的文件存储请求执行成功还是失败, 对应每个文件存储请求, 文件存储应答附加以下内容:
- a) 若请求失败, 附加说明;
 - b) 若请求成功, 附加相关的、对应格式的详细操作报告。

6.3.14 “filestore responses”参数不得用于传送目录列表信息。目录列表信息的传输通过目录操作事务实现。

6.3.15 “flow label”参数为可选项，用于支持优先级和抢占方案。该参数指示了 PDU 应当被传输的顺序，但如何使用该参数属于具体实现问题。

6.3.16 “offset”参数表示从文件起始位置的偏移量。

6.3.17 “length”参数表示接收的文件数据的字节数。

6.3.18 “progress”参数报告了当前文件发送或接收的进度。

6.3.19 各原语参数的用法在 6.5 节中说明。

6.4 底层通信系统业务

用于访问 UT 层业务的原语和参数为：

a) UNITDATA.request (UT_SDU, UT Address);

b) UNITDATA.indication (UT_SDU, UT Address)。

注1：为使 CFDP 通用，假定底层业务尽可能简单。本标准中 UT 业务数据单元 (UT_SDU) 是指 UT 层业务及其传送的分隔数据单元。虽然 UT 层在概念上为单一业务，然而在具体实现中，CFDP 可能在不同的时间、甚至同时使用几种不同的物理底层，这取决于与之通信的远端实体。在纯 CCSDS 网络中，唯一可用的业务是 CCSDS 路径或包业务，并且 UT_SDU 将作为 CCSDS 包的数据域。

注2：UT 地址参数的格式和内容由底层业务的寻址能力和规则决定。MIB 中的信息必须能够支持协议实体名称与 UT 地址之间的转换。

注3：协议运作在一个概念性的单一 UT 业务接入点上。实际上，UT 层并不强制必需通过单一的物理业务接入点提供所有业务。若同时使用了不同的物理底层通信系统，则需要不同的物理业务接入点。在某些特定的实现中，甚至可能对每个远端实体均维持不同的物理业务接入点，以使得 CFDP 能够在任何时候进行直接通信。

注4：上面列出的必需的 UT 层业务原语并不意味着该 UT 层不能支持有利于 CFDP 实现的其他额外服务。可能有价值的附加原语如下：

a) 当 UT 层准备接受另一个传输给指定 UT Address 的 PDU，提交一个指示（这个指示可用来实现数据流控制；该指示需要包含目标 UT Address，因为该 UT 层可能会同时向多个 UT Address 传输）；

b) 当 UT 层完成了 UT_SDU（内容为一个 CFDP PDU）的传输和 ACK 确认时，提交一个指示（该指示可以用来启动 ACK 定时器）。

注5：假定最低质量的底层业务为：

a) 交付 UT_SDU 中可能出现错误；

b) 不完整，某些 UT_SDU 丢失；

c) 按顺序。例如，UT_SDU 按照它们发送的顺序被传输（按顺序到达是“立即”、“异步”、“提示”这三种段丢失检测模式和保活机制设计的特定要求）。

CFDP 可以在可能发生数据错误和数据丢失的网络上运行，并且可以容忍缺少渐进段丢失检测规程，甚至数据到达失序。然而，该协议不是用于在经常发生此类情况的网络条件下工作。如果依赖 CFDP 解决问题，将导致性能严重下降。

6.5 业务原语

6.5.1 PUT.request 原语

6.5.1.1 功能

PUT.request 原语用于提出将文件从源文件存储器传输到目标文件存储器的文件提交请求。

6.5.1.2 原语形式

```
PUT.request(destination CFDP entity ID,
            [source file name],
            [destination file name],
            [segmentation control],
            [fault handler overrides],
            [flow label],
            [transmission mode],
            [messages to user],
            [filestore requests])
```

注：方括号内的参数为可选参数，下同。

6.5.1.3 发起时机

由源用户根据需要发起。

6.5.1.4 接收响应

收到本原语时，协议实体将启动源实体的提交规程。

6.5.2 CANCEL.request 原语

6.5.2.1 功能

CANCEL.request 原语用于请求取消当前事务。

6.5.2.2 原语形式

```
CANCEL.request (transaction ID)
```

6.5.2.3 发起时机

由源用户在事务进行中根据需要发起。

6.5.2.4 接收响应

收到本原语时，协议实体将生成一个取消通知。

6.5.3 SUSPEND.request 原语

6.5.3.1 功能

SUSPEND.request 原语用于请求挂起当前事务。

6.5.3.2 原语形式

```
SUSPEND.request (transaction ID)
```

6.5.3.3 发起时机

由任意用户在事务进行中根据需要发起，但对于采用无应答模式的事务，只能由事务的源端用户发起。

6.5.3.4 接收响应

收到本原语时，协议实体将生成一个挂起通知。

6.5.4 RESUME.request 原语

6.5.4.1 功能

RESUME.request 原语用于请求恢复当前已经被挂起的事务。

6.5.4.2 原语形式

RESUME.request (transaction ID)

6.5.4.3 发起时机

由任意用户根据需要在事务进行中发起，但对于采用无应答模式的事务，只能由事务的源端用户发起。

6.5.4.4 接收响应

收到本原语时，协议实体对处于指定的挂起状态的事务启动恢复程序，如果事务没有被挂起，则不作任何动作。

6.5.4.5 说明

应用程序开发者可能会发现，对于某些情况，程序最好能够支持事务的自动恢复，例如当应用程序可以确定当前事务挂起仅是由于该事务不活动造成的（即挂起是“不活跃”故障的处置结果）。当应用程序收到实体上报的与本事务各 PDU 接收相关的指示时，应用程序可推断出该事务已终止不活跃状态，即可自动恢复该事务。

6.5.5 REPORT.request 原语

6.5.5.1 功能

REPORT.request 原语用于请求关于当前事务状态的报告。

6.5.5.2 原语形式

REPORT.request (transaction ID)

6.5.5.3 发起时机

由任意用户在事务进行中根据需要发起。

6.5.5.4 接收响应

收到本原语时，协议实体将向用户产生一个 REPORT.indication 原语。

6.5.6 TRANSACTION.indication 原语

6.5.6.1 功能

TRANSACTION.indication 原语用于协议实体将启动事务的事务 ID 上报给源用户。

6.5.6.2 原语形式

TRANSACTION.indication (transaction ID)

6.5.6.3 发起时机

由协议实体在接收到 PUT.request 原语时产生。

6.5.6.4 接收响应

收到本原语时的处理由用户自定义。

6.5.6.5 说明

transaction ID 参数由协议实体返给用户，为源端用户提供一个该事务的唯一标识。

6.5.7 EOF-SENT.indication 原语

6.5.7.1 功能

[source file name],
[destination file name],
[messages to user])

6.5.9.3 发起时机

由目的端协议实体在接收到元数据 PDU 时产生。

6.5.9.4 接收响应

收到本原语时的处理由目的端用户自定义。

6.5.10 FILE-SEGMENT-RECV.indication 原语

6.5.10.1 功能

FILE-SEGMENT-RECV.indication 原语用于目的端协议实体向目的端用户指示接收到了单个文件的数据段（非元数据）。

6.5.10.2 原语形式

FILE-SEGMENT-RECV.indication(transaction ID,
offset,
length)

6.5.10.3 发起时机

由目的端协议实体在接收到文件数据 PDU 时产生。

6.5.10.4 接收响应

收到本原语时的处理由目的端用户自定义。

6.5.10.5 说明

是否产生本原语是可选的。offset 是指接收到的文件数据段的起始字节距离整个文件起始字节的字节数。

6.5.11 SUSPENDED.indication 原语

6.5.11.1 功能

SUSPENDED.indication 原语用于向用户指示事务已经挂起。

6.5.11.2 原语形式

SUSPENDED.indication(transaction ID,
condition code)

6.5.11.3 发起时机

由协议实体在文件传输程序产生挂起通知时生成。

6.5.11.4 接收响应

收到本原语时的处理由用户自定义。

6.5.12 RESUMED.indication 原语

6.5.12.1 功能

RESUMED.indication 原语用于向用户指示挂起的事务已经被恢复。

6.5.12.2 原语形式

RESUMED.indication(transaction ID,

progress)

6.5.12.3 发起时机

由协议实体在应答 RESUME.request 原语时生成。

6.5.12.4 接收响应

收到本原语时的处理由用户自定义。

6.5.12.5 说明

progress 参数用于表示协议实体在产生此指示的时刻，对于该事务文件的传输进度或者接收进度。

6.5.13 REPORT.indication 原语

6.5.13.1 功能

REPORT.indication 原语用于向用户报告事务的状态。

6.5.13.2 原语形式

REPORT.indication(transaction ID,
status report)

6.5.13.3 发起时机

由协议实体在收到 REPORT.request 原语时生成。

6.5.13.4 接收响应

收到本原语时的处理由用户自定义。

6.5.13.5 说明

status report 的格式和范围由具体实现决定。

6.5.14 FAULT.indication 原语

6.5.14.1 功能

FAULT.indication 原语用于协议实体向用户报告发生了故障状态，且对应这种故障状态的故障处置是“忽略”。

6.5.14.2 原语形式

FAULT.indication(transaction ID,
condition code,
progress)

6.5.14.3 发起时机

由协议实体在出现故障时，且对应这种故障状态的故障处置是“忽略”时生成。

6.5.14.4 接收响应

收到本原语时的处理由用户自定义。

6.5.14.5 说明

progress 参数用于表示协议实体在产生此指示的时刻，对于该事务文件的传输进度或者接收进度。

6.5.15 ABANDONED.indication 原语

6.5.15.1 功能

ABANDONED.indication 原语用于向用户报告发生了故障状态，且对应这种故障状态的故障处置是“放弃”。

6.5.15.2 原语形式

ABANDONED.indication(transaction ID,
condition code,
progress)

6.5.15.3 发起时机

由协议实体在出现故障时，且对应这种故障状态的故障处置是“放弃”时生成。

6.5.15.4 接收响应

收到本原语时的处理由用户自定义。

6.5.15.5 说明

progress 参数用于表示协议实体在产生此指示的时刻，对于该事务文件的发送进度或者接收进度。

6.5.16 TRANSFER-CONSIGNED.indication 原语

6.5.16.1 功能

当且仅当扩展规程运行时，TRANSFER-CONSIGNED.indication 原语用于向源端用户报告第一个中转节点已经完成了文件的接收。

6.5.16.2 原语形式

TRANSFER-CONSIGNED.indication(transaction ID)

6.5.16.3 发起时机

由源端实体在接收到一个“终端系统状态”为“0”的完成 PDU 时生成。

6.5.16.4 接收响应

本原语授权源端用户可以修改或删除协议实体此前在非易失性存储空间中保存的重发缓冲器中的文件。

6.5.16.5 说明

本原语用于实现扩展规程时，在存储空间受限的环境中支持“缓存共享”。

6.5.17 EOF-RECV.indication 原语

6.5.17.1 功能

EOF-RECV.indication 原语用于目的端协议实体向目的端用户报告事务的 EOF PDU 被接收到。

6.5.17.2 原语形式

EOF-RECV.indication(transaction ID)

6.5.17.3 发起时机

由目的端协议实体在接收到事务的 EOF PDU 时生成。

6.5.17.4 接收响应

收到本原语时的处理由目的端用户自定义。

6.5.17.5 说明

是否实现本原语是可选的。

7 协议规程

7.1 核心规程

7.1.1 CRC 规程

7.1.1.1 PDU 发送实体的 CRC 规程

若选用 CRC，PDU 发送实体应该设置 CRC 标识为“1”状态，并且为每一个即将发出的 PDU 计算并插入 CRC。

7.1.1.2 PDU 接收实体的 CRC 规程

如果在到达的 PDU 中 CRC 标识为“1”，PDU 接收实体应计算 PDU 的 CRC，并且将 CRC 验证错误的 PDU 丢弃。

7.1.1.3 CRC 校验规程

7.1.1.3.1 CRC 运算的算法应该是符合 QJ 20747 中规定的 CRC 算法。

7.1.1.3.2 CRC 的值应该放置于 PDU 数据域的最后，并且它的长度应该计入 PDU 数据域的总长度中。CRC 算法的应用对象是从 PDU 导头部的第一个字节到 CRC 前 PDU 数据域的最后一个字节的全部数据。

7.1.2 校验和规程

校验和的长度为 32 位，并且它的运算应该遵循以下方法：

- a) 初始化时应该全部被设置为“0”；
- b) 从文件的起始位置起，以 4 字节为单元进行累加和运算；
- c) 每一个 4 字节单元由文件中对应的字节组成，将文件中的偏移是 4 的整数倍的字节填入 4 字节数据的第一个字节，将相邻的 3 个字节分别填入 4 字节数据的后 3 个字节；
- d) 求和运算的结果应该放入校验和的对应位置中，如果该结果在校验和规定的长度中溢出，则将进位的部分丢弃。

注1：对于偏移量不是 4 的整数倍的文件数据 PDU，为了能够在文件中填入对齐后的校验和，需要在偏移量为 Q 的字节前填入 N 个“0”，其中 $N = Q \bmod 4$ ，然后进行计算。

注2：对于数据长度为 M 个字节（该数据首个字节在文件中的偏移是 4 的整数倍），且 M 小于 4 的数据序列，需要在后面补充一些数据使其长度达到 4，并完成校验和。添加的数据为 4-M 个字节的“0”。

7.1.3 提交规程

当接收到了一个 PUT.request 原语，协议实体将会进行以下操作：

- a) “事务启动通知”规程。
- b) “文件复制”规程：
 - 1) 从 PUT.request 原语中获取故障处置信息（对应原语参数中的 fault handle overrides）；
 - 2) 从 PUT.request 原语获取 MTU 或文件存储请求（对应原语参数中的 filestore requests）；
 - 3) 当源文件名（对应原语参数中的 source file name）或者目的文件名（对应原语参数中的 destination file name）缺失，指示只传输元数据；
 - 4) 传输模式（应答模式或者无应答模式）由 PUT.request 原语中的传输模式参数（对应原语参数中的 transmission mode）控制，否则采用 MIB 中现有的默认设置。

7.1.4 事务启动通知规程

事务 ID 规定如下：

- a) 由源协议实体在响应 PUT.request 时生成；

b) 在后续所有过程中，均应用此事务 ID 表示该事务。

两个事务在同一时刻不能拥有相同的事务 ID，相同的事务 ID 将导致协议运行错误。源协议实体应该通过 TRANSACTION.indication 原语将事务 ID 通知协议用户。

7.1.5 PDU 转发规程

利用底层 UT 业务的寻址能力，每一个 PDU 将被发往合适的协议实体。利用 PDU 中目的实体 ID，可以从 MIB 中获得 PDU 目的地的远程 UT-SAP。

基于 PDU 的类型，PDU 所发往的协议实体 ID 将会是以下几种情况之一：

- a) 对于一个 ACK PDU，该 ID 是发送 PDU 并要求应答的协议实体 ID；
- b) 对于一个文件数据、元数据、EOF，或者提示 PDU，该 ID 是实施文件复制操作的接收实体 ID；
- c) 对于一个 NAK、完成 PDU 或者保活 PDU，该 ID 是实施文件复制操作的发送实体 ID。

注：当仅实现核心规程时，文件的发送方和接收方就是在 PDU 导头域标明的源实体和目的实体。当应用扩展规程时，文件的发送方和接收方则需要根据 PDU 中的实体 ID 识别是否与它本身相关。

当 PDU 的目的实体 ID 与它发往的协议实体的实体 ID 不一致时，该 PDU 应该被丢弃。除非实体实现了扩展规程，则这些 PDU 在扩展规程中处理。

7.1.6 文件复制规程

7.1.6.1 通用规程

7.1.6.1.1 说明

在本标准后续章节，EOF PDU 或者完成 PDU 的具体类型应该包括括号中的特殊状态码（如“无错误”），或者一般状态码“取消”，表示 PDU 包含的除了“无错误”和“挂起接收请求”的一般场景编码。

ACK PDU 的类型，即括号中的被应答的 PDU 类型，包括 EOF 或者完成。

当只有核心规程实现时，文件的发送方往往是事务的源实体，文件的接收方往往是事务的目的实体。

7.1.6.1.2 发送端文件复制规程

发送端文件复制规程如下：

- a) 如果事务为无应答模式，在发送端应使用无应答模式规程；如果事务是有应答模式，在发送端应使用有应答模式规程。
- b) 文件复制规程开始时，协议发送端向接收端发送一个元数据 PDU。
- c) 元数据 PDU 应该包括：
 - 1) 文件中是否包含分段传输时需考虑的数据记录边界的指示；
 - 2) 文件的大小（若发送的是有记录边界的文件则填写，否则填 0）；
 - 3) 文件的源和目的名（路径名）；
 - 4) 可选的错误处理、到用户的消息、文件存储请求、流标签。

注：要确保事务所有相关的元数据都包含在一个独立的元数据 PDU 中。

- d) 对于那些不只是发送元数据的事务，文件复制开始时，协议发送端将从发送文件存储中取出文件，然后将它在文件数据 PDU 中发送。

- e) 从分段位置的文件起始处，每一个数据分段都应该在文件数据 PDU 中发送，其中包括偏移量。
- f) 如果分段控制业务参数请求考虑数据记录边界，则分段的首字节应为一个文件数据记录的首字节；多个完整的数据记录可能组成一个独立的分段，但是没有任何一个数据记录可以扩展到多个分段中。

注：文件中数据记录边界如何被检测的方法是一个本地实现的问题。在协议实体间的互操作中应用这种方法不是必须的。（但是，这种方法可能在源和目的协议用户应用中的正确操作时需要。）

- g) 如果分段控制业务参数被忽略，则默认没有数据记录边界。

注：分段算法不在本标准中进行规定。

- h) 如果分段控制业务参数要求数据记录边界被考虑，但是文件中却没有可用的数据记录结构，则会提示无效文件结构错误。

注：文件中监测数据记录结构的算法不在本标准中进行规定。这是与发送端相关的部署问题，与接收端无关。

- i) 当元数据 PDU 和所有文件数据 PDU 发出时（当（A）发送的文件是有界限的并且最后 1 个字节已经发出，或者（B）发送的文件是无界限的并且文件的发送已经结束，或者（C）没有文件要发送），执行下列操作：

- 1) 协议发送实体应该发出一个 EOF（无错误）PDU；
- 2) 基于管理信息库中的设置，协议发送实体可发出一个可选的 EOF-SENT.indication 原语；
- 3) EOF（无错误）PDU 应该包含原始文件的校验和以及文件的长度。

- j) 流标签为可选项，可以支持优先级以及抢占机制。

注：流标签的使用是特定的实现问题。

7.1.6.1.3 接收端文件复制规程

接收端文件复制规程如下：

- a) 如果事务为无应答模式，在接收端应使用无应答模式规程；如果事务是有应答模式，在接收端应使用有应答模式规程。
- b) 协议接收端应该存储事务接收到的第一个 PDU 中的传输模式值，并在后续事务的处理中使用；如果协议接收端不能在该模式下处理，则会提示无效传输模式错误。
- c) 协议接收端应该存储包含在元数据 PDU 中的错误处理信息、文件大小、流标签，以及文件名等信息，并在后续事务的处理中使用。
- d) 所有重复的元数据 PDU 应该被丢弃。
- e) 对于不只是传输元数据的事务，如果协议接收端发现文件存储器无法接收该文件，则提示文件存储拒绝错误。

注：文件存储的能力问题是与实现相关的问题。

- f) 如果接收方就是事务的目的地，则目的实体端接收到元数据 PDU 时，应该发出 METADATA-RECV.indication 原语：

- 1) 如果元数据 PDU 中包含一个或多个用户消息，它们应该作为 METADATA-RECV.indication 原语的参数传递；
- 2) 如果元数据 PDU 包含文件存储请求，应该在事务文件结束后存储起来为后续执行做准备。

- g) 当接收端收到了文件数据 PDU 后, 应该使用文件数据 PDU 中偏移量和分段大小信息将这些文件数据组装成为一个文件:
 - 1) 任何重复的数据应该被丢弃;
 - 2) 如果元数据 PDU 中的分段控制标识表示的是数据记录边界需要被考虑时, 接收到的文件数据 PDU 数据记录的对齐会被用于数据记录的重构;
 - 3) 如果接收到的文件数据 PDU 的偏移和分段大小超过了之前接收到的第一个 EOF(无错误) PDU 中的文件大小标识, 则会提示文件大小错误。
- h) 事务元数据、所有的文件数据, 以及 EOF (无错误) PDU 在最初时就被接收端接收:
 - 1) 为传输的文件计算校验和;
 - 2) 比较接收到的文件中包含的校验和与计算得到的校验和;
 - 3) 如果校验和的比较验证是正确的, 则可以认为本次文件传输结束;
 - 4) 如果校验和的比较验证是错误的, 则提示文件校验和错误。
- i) 在最初收到 EOF (无错误) PDU 时, PDU 中包含的文件大小会与事务接收过程中得到的文件大小进行比较, 如果传输过程中得到的文件大小超过了 PDU 中包含的文件大小, 则会提示文件大小错误。
- j) 流标签为可选项, 可以支持优先级以及抢占机制。

注: 流标签的使用是具体实现问题。

7.1.6.2 接收端可选文件复制规程

如果协议接收端正是事务的目的地, 则它在接收到一个文件数据 PDU 后, 会可选择的发出 FILE-SEGMENT-RECV.indication 原语, 初始接收到 EOF PDU 时, 会可选择的发出 EOF-RECV.indication 原语。

7.1.6.3 无应答模式规程

7.1.6.3.1 发送端无应答模式规程

协议发送端发送 EOF (无错误) PDU 后, 会发出文件发送完成通知 (已完成)。

7.1.6.3.2 接收端无应答模式规程

当接收到 EOF (无错误) PDU 时, 进行如下操作:

- a) 如果文件已经成功结束, 则会发出文件发送完成通知 (已完成);
- b) 否则将会启动一个与事务相关的检查计时器; 该计时器将根据具体情况设定超时时间, 并且会为每一个独立的事务设置超时时间; 当定时器超时, 接收实体会检查文件接收是否已经完成; 如果已经完成, 则会发出文件发送完成通知 (已完成); 否则提示已达检查上限错误, 将定时器复位。

7.1.6.4 有应答模式规程

7.1.6.4.1 发送端有应答模式规程

发送端有应答模式规程如下:

- a) 协议发送端会重传所有 NAK PDU 中请求的元数据 PDU，以及在 NAK PDU 分段请求的起始和结束偏移中定义的数据文件范围。
注：在文件复制规程中的重传缓冲区可能是一个私有的或者是一个公开的文件。这是一个具体的实现问题，缓冲区的释放同理。
- b) 重传数据的分段不应受到原始分段传输及 NAK PDU 中起始和结束偏移的限制，但仍受分段控制参数（表明是否有记录边界）规定的影响。
- c) 接收到终端系统状态标识为“1”的完成（无错误）PDU，协议发送实体会发出文件发送完成（完成）通知。
- d) 对终端系统状态标识符为“0”的完成（无错误）PDU 的处理将在扩展规程中定义。
- e) ACK 规程应该应用于 EOF（无错误）PDU 和完成（无错误）PDU，分别使用 ACK（EOF）PDU 和 ACK（完成）PDU 作为预期的应答。

7.1.6.4.2 接收端有应答模式规程

如果下列条件为真，则可以发现文件数据的缺失：

- a) 接收到的文件数据第 1 个分段的第 1 个字节的偏移量不为 0；
- b) 文件数据中某个中间分段的最后 1 个字节的偏移量与其下一个邻近段的第一个字节偏移量的偏差大于 1；
- c) 文件数据最后 1 个分段的最后 1 个字节的偏移量与 EOF（无错误）PDU 中指示的文件长度的偏差大于 1。

对于未接收到元数据 PDU 的情况，元数据丢失的检测可以通过接收到的事务中第 1 个非元数据 PDU（文件数据或者 EOF（无错误））来实现。

与正在运用的分段丢失检测规程相一致，协议接收实体将使用以下方式发出 NAK PDU：

- a) 每一个 NAK PDU 能够识别与其相关的文件数据的子集，如：NAK PDU 的范围；NAK PDU 的范围由文件中的两个偏移量来表示，代表范围的起始和结束。
- b) 除了范围之外，每个 NAK PDU 应该包含 0 个或者多个分段请求；NAK PDU 中的分段请求应该能够表明文件数据范围内还未收到的所有分段的起始偏移量和结束偏移量，也能够识别丢失的元数据。
- c) 分段请求的起始偏移量和结束偏移量不需要与文件最初分段相关联。
- d) 当发现分段不连续时，分段丢失检测规程将会产生 NAK 序列：
 - 1) 每个 NAK 序列都与文件数据的一些子集相关，如：NAK 序列的范围。
 - 2) NAK 序列范围的起始就是同一个事务的上一个 NAK 序列范围的结束，如果此前无 NAK 序列，则为 0。
 - 3) 如果已经接收到事务的 EOF（无错误）PDU，NAK 序列范围的结束值就是文件的完整长度；否则应为触发产生 NAK 序列时的当前接收进度。
 - 4) 每个 NAK 序列应该包含尽可能多的需要的 NAK PDU，需要考虑 UT_SDU 内容最大长度限制，包含在 NAK 序列范围内的为了对丢失的文件数据和元数据请求重传而必须发送的全部分段请求。

注：一般情况下，每个 NAK 序列只包含 1 个 NAK PDU。

- 5) 序列中第 1 个 NAK PDU 的范围起始值就是 NAK 序列的范围起始值，序列中每个 NAK PDU 的范围起始值就是上一个 NAK PDU 的范围结束值。序列中最后一个 NAK PDU 范围结束值就是 NAK 序列的范围结束值，序列中每个 NAK PDU 中的范围结束值就是 NAK PDU 最后分段请求的结束偏移量。

当文件传输结束时，如前所述，实体应该：

- a) 发出文件传输完成通知（完成）；
- b) 如果接收实体就是事务的目的地，发出完成（无错误）PDU，将其中终端系统状态标识置为“1”，所有的文件存储应答作为参数包含在完成（无错误）PDU 中。

ACK 规程应该应用于 EOF（无错误）PDU 和任意的完成（无错误）PDU 中，并且应该分别使用 ACK（EOF）PDU 和 ACK（完成）PDU 作为预期应答。

7.1.6.4.3 接收端渐进段丢失检测规程

多个渐进段丢失检测规程并不是互斥的。渐进段丢失检测规程的选择是一个本地化实现问题。对于一个给定的事务，在接收到 EOF（无错误）PDU 后，渐进段丢失检测规程就停止工作了。共有如下三种模式：

- a) 立即模式；在立即模式下，在接收到 EOF（无错误）PDU 之前检查到每个文件数据缺失或者元数据丢失时，协议实体都会立即发出一个 NAK 序列；
- b) 提示模式；在提示模式下，在接收到 EOF（无错误）PDU 之前接收到提示 PDU，协议接收实体会发出 NAK 序列；如果文件数据和元数据没有丢失，NAK 序列会包含一个不含任何分段请求的 NAK PDU；
- c) 异步模式；在异步模式下，在接收到 EOF（无错误）PDU 之前，为响应与实现相关外部事件的应答，协议接收实体会发出 NAK 序列；如果文件数据和元数据没有丢失，NAK 序列会包含一个不含任何分段请求的 NAK PDU。

注：外部事件的实例包含对于不连续可见航天器的通信窗口开启和接收端协议实体定时器触发的信号。

7.1.6.4.4 发送端渐进段丢失检测规程

对一个与实现相关的外部事件的应答，协议发送实体将发出一个提示（NAK）PDU。

7.1.6.4.5 接收端延时段丢失检测规程

在接收到一个事务的 EOF（无错误）PDU，接收实体应决定事务的文件数据和元数据是否需要接收，方式如下：

- a) 如果检测到任何的文件数据缺失或者元数据丢失，并且之前发出的 NAK PDU 却没有包含与之相关的任何分段请求，那么协议接收实体将会发出一个 NAK 序列；
- b) 事务相关的 NAK 计时器将会启动；该计时器应该设置一个与具体实现相关的超时时间；当计时器超时，接收实体应该决定事务的文件数据和元数据是否需要接收；如果需要接收，则接收实体应该发出一个 NAK 序列，它的范围从 0 开始，一直扩展到整个文件长度，定时器也同时复位。

注：由于 NAK 定时器的超时时间是一个完全与具体实现相关的，因此在每一次数据完整的处理中该超时时间并不要求完全相同。事实上，超时时间每一次都可能会有变化。在实际实现时，对文件中接收数据完整性的重复检查以及可能的 NAK 序列的发送，可能发生在接收到 EOF（无错误）PDU 后的任意时刻。

7.1.6.4.6 NAK 限制

NAK 次数限制是与具体实现相关的，对于一个事务来说，若达到了与实现相关的某个限制条件，则应提示已达 NAK 限制错误。

7.1.6.5 保活规程

7.1.6.5.1 一般要求

保活机制用于发送实体监视接收实体的文件数据接收进度，可在对 EOF（无错误）PDU 进行确认之前检测到问题。

7.1.6.5.2 接收端保活规程

在所有的有应答模式中，协议接收实体会周期性地向发送实体发送保活 PDU，通知本实体到目前为止的事务接收进度。

在接收到提示（保活）PDU 时，应该发送保活 PDU 进行响应。

在接收到 EOF（无错误）后，应终止保活 PDU 的传输。

7.1.6.5.3 发送端保活规程

在协议发送端，如果保活 PDU 报告的接收进度与本实体的传输进度的差异超过了预设的阈值，则协议发送实体将会可选择的提示保活已达阈值错误。

在对某些与具体实现相关事件的响应时，发送实体将会产生一个提示（保活）PDU 而使接收实体发送保活 PDU。

7.1.6.6 取消应答规程

7.1.6.6.1 接收端取消应答规程

协议接收实体在收到 EOF（取消）PDU 后，将会发出传输已完成（取消）通知。

如果当前是有应答模式，ACK 规程将发送对 EOF（取消）PDU 的 ACK（EOF）PDU 作为响应。

7.1.6.6.2 发送端取消应答规程

协议发送实体在接收到完成（取消）PDU 后，将会发出传输已完成（取消）通知。

如果当前是有应答模式，ACK 规程将发送对完成（取消）PDU 的 ACK（完成）PDU 作为响应。

7.1.6.7 恢复规程

7.1.6.7.1 一般要求

在接收到协议用户发出的 RESUME.request 原语后，将会应用恢复规程，方式如下：

- a) 当与其相关的事务当前并没有挂起时，RESUME.request 原语将被忽略；
- b) 如果如 RESUME.request 原语相关的事务不仅是挂起状态，而且是冻结状态，则认为该事务不是挂起状态，当前唯一适用的操作是发出 RESUME.indication 原语。

7.1.6.7.2 发送端恢复规程

在接收到 RESUME.request 原语后，协议发送实体应该进行如下操作：

- a) 恢复文件分段的传输；
- b) 发出 RESUME.indication 原语。

如果当前是有应答模式：

- a) 任何挂起的提示 PDU 都应该被恢复；
- b) 该实体之前应用 ACK 规程发出的 PDU 都应该被恢复。

7.1.6.7.3 接收端恢复规程

在接收到 RESUME.request 原语后，协议接收实体应该进行如下操作：

- a) 恢复 NAK PDU 的传输；
- b) 恢复挂起的保活 PDU 的传输；
- c) 发出 RESUME.indication 原语。

该实体之前应用 ACK 规程发出的 PDU 都应该被恢复。

7.1.6.8 报告规程

在文件复制规程过程中接收到 REPORT.request 原语时，协议实体应发出 REPORT.indication 原语来说明当前传输事件的进度。

注1：由于与其他协议实体没有互操作，因此没有与该规程相关的协议。

注2：REPORT.indication 原语可以不依赖于 REPORT.request 原语异步发出。

7.1.7 ACK 规程

7.1.7.1 PDU 发送端 ACK 规程

当 ACK 规程应用于一个 PDU 时，进行如下操作：

- a) 在发出 PDU 的同时，协议发送实体应该启动一个定时器，并为必要的重传保存发出的 PDU；
- b) 如果在定时器超时之前没有收到预期应答，协议发送实体应该重发原始 PDU；
- c) 协议发送实体应该保存重传次数的计数；
- d) 如果超过当前阈值，协议发送实体会提示未收到 ACK 次数达到上限错误；
- e) 协议发送实体在接收到预期应答后会释放保存的 PDU。

7.1.7.2 PDU 接收端 ACK 规程

在应用了 ACK 规程后，当接收到一个 PDU，将立即发出预期应答。

注1：协议实体发出预期应答为了确认收到了 PDU，预期应答的发出并不意味着协议接收实体已执行了相关的其他动作。例如，ACK (EOF (取消)) PDU 的产生意味着 EOF (取消) PDU 被接收，而不是相应的事务被取消。

注2：协议接收实体在接收到应用了 ACK 规程的 PDU 后，会发出预期应答。预期应答的目的是在发送端关闭 PDU 重传计时器，而不论事务状态是未定义状态、活跃状态、终止状态还是未识别状态。

7.1.8 错误处理规程

当一个错误被检测到或是被提示后，都应该有相应的错误处理操作。

在 MIB 中应该为每一种类型的错误设置一种默认的处理，但是在特定的事务中会有一种或多种错误类型的特殊的错误处理方法。如果有这样的特殊错误处理方法，应该使用相应的错误类型取代默认的错误处理方法。

注：在接收到事务的元数据之前，协议接收实体不知道应该使用哪一个错误处理操作，所以只能使用默认的错误处理操作。在提交请求中指定的错误处理操作并不能保证接收实体就能够按照指定的操作进行处理。

错误处理的引入将导致以下结果：

- a) 发出取消通知；
- b) 发出挂起通知，触发条件为事务使用“有确认”的传输模式发送，或者在事务源端提示错误状况；
- c) 协议不处理，例如：错误状况将被忽略，有可能导致不可预期的协议行为，并产生带错误标

识码的 FAULT.indication 原语；

- d) 结束正在进行的事务，同时发送包含错误标识码的 ABANDONED.indication 原语。
即使没有相应的错误处理，所有的错误也应该记录下来。

7.1.9 文件存储规程

文件存储规程如下：

- a) 只有当相关联的文件复制规程正确结束，才能够执行文件存储请求；
注：当文件复制规程过程中出现任何类型的错误，则不执行相关联的文件存储请求，也不会发送文件存储应答。
- b) 文件存储请求在元数据 PDU 的指令参数域中发送，与 PUT 原语中的顺序一致；
- c) 文件存储的执行是强制的；文件存储请求的执行顺序与元数据 PDU 中的指令参数域中的顺序一致；
- d) 文件存储请求的执行会产生文件存储应答参数；如果当前是有应答模式，文件存储应答参数将通过完成（无错误）PDU 发送给请求的发起方；
- e) 如果某个文件存储请求未被执行，则后续同一个元数据 PDU 中的文件存储请求都不会被执行；对于每个后续未被执行的文件存储请求，应该在文件存储应答参数中返回未执行状态码；
- f) 文件存储请求的异常不会提示任何类型的错误。

7.1.10 不活跃监控规程

不活跃监控规程如下：

- a) 对于一个特定的事务，如果在特定的时间段内（事务不活跃）未接收 PDU，应该提示不活跃传输错误；
- b) 不会应用于无应答传输模式中；
- c) 对于不活跃传输错误的默认处理操作为取消该事务。

7.1.11 内部规程

7.1.11.1 完成通知规程

7.1.11.1.1 概述

注1：在以下部分，完成通知将包括两种情况，当括号中特征为“完成”时，表示规程成功结束，“取消”则表示规程取消。当特征名在上下文中是相关的，当与特征无关时，可以不写括号中的特征名。

注2：针对一个事务的传输完成（取消）通知可能会在对同一个事务的传输完成（完成）通知之后分发。这可能由以下情况导致。例如接收实体对完成（无错误）PDU 相应的 ACK PDU 接收的重复错误，会导致发出达到活跃应答阈值错误提示。完成通知进程的标准将延续这种方式。在接收端，当传输完成（取消）通知发出后，接收到的文件将发送给用户进行正常处理。

7.1.11.1.2 发送端完成通知处理规程

针对文件复制规程的完成通知，协议发送实体将会进行以下操作：

- a) 清除文件重发缓冲器中未发的部分；
- b) 停止文件分段和元数据的发送。

如果发送使用的是有应答模式，则：

- a) 所有提示 PDU 的发送都应该被终止；
- b) 被该实体之前发出的应用 ACK 规程的 PDU 应该被终止。

在任何情况下，执行如下操作：

- a) 如果发送实体就是事务的源端，将发出 TRANSACTION-FINISHED.indication 原语表示传输的完成状态。
- b) 如果同时满足以下 4 个条件，则通过 TRANSACTION-FINISHED.indication 原语发送文件存储应答参数：
 - 1) 发送实体就是事务的源端；
 - 2) 文件使用有应答模式传输；
 - 3) 完成通知中的规程配置为“完成”；
 - 4) 事务完成时到达的完成 PDU 包含着文件存储应答参数。

7.1.11.1.3 接收端完成通知处理规程

如果使用有应答模式，执行如下操作：

- a) 当发生超时或是其他类型的事件，终止 NAK PDU 的传输；
- b) 终止保活 PDU 的传输；
- c) 终止该实体之前发出的应用了 ACK 规程的 PDU。

在任何情况下，执行如下操作：

- a) 如果发送实体就是事务的目的端，完成通知中的设置为“完成”，协议接收实体会执行提交规程中传输的任意文件存储请求；
- b) 如果完成通知中的特征为“取消”，并且接收实体就是事务的目的端，则未发送完成的数据根据 MIB 中的设置选项丢弃或者保存；

注：当一个包含传输完成（取消）通知的事务之前包含的是传输完成（完成）通知时，则不需要丢弃或者保存不完整的数据。

- c) 如果接收实体就是事务的目的端，它将根据配置发出 TRANSACTION-FINISHED.indication 原语，来表明事务已完成；
- d) 如果接收实体就是事务的目的端，TRANSACTION-FINISHED.indication 原语有文件存储应答或者状态报告参数时，这些参数应该在其中进行传递。

7.1.11.2 取消通知规程

7.1.11.2.1 一般要求

在文件复制规程中的任意时刻，协议发送端和接收端都可能会发出取消通知。

注：在接收到错误提示或者协议用户端发出的 CANCEL.request 原语时，作为应答，应该发出取消通知。

7.1.11.2.2 发送端取消通知规程

针对文件复制规程的取消通知，协议发送实体应该进行如下操作：

- a) 发出传输完成（取消）通知；
- b) 当接收到 CANCEL.request 原语或者由错误状态引发的取消通知，会发出 EOF（取消）PDU，来表明事务终止的原因；EOF（取消）PDU 中的文件大小域应该包括事务当前的传输进度；PDU 中的校验和应该是到目前为止所发送的所有文件 PDU 的校验和。

如果在有应答模式下发送，则进行如下操作：

- a) 对 EOF（取消）PDU 应用 ACK 规程时，其预期响应应为 ACK（EOF）PDU，其条件码与 EOF

(取消) PDU 一致;

- b) 在发出 EOF (取消) PDU 并且在接收到预期应答之前, 如果 ACK 规程无效, 任何接收到的 PDU 都会被忽略。

在 EOF (取消) PDU 中提示的错误, 会导致事务终止。

7.1.11.2.3 接收端取消通知规程

针对文件复制规程的取消通知, 协议接收实体会发出完成 (取消) 通知。

如果在有应答模式下接收, 则进行如下操作:

- a) 协议接收实体发出完成 (取消) PDU, 表明事务终止的原因: 接收到 CANCEL.request 原语, 或者导致产生取消通知的错误码;
- b) 对完成 (取消) PDU 应用 ACK 规程时, 其预期响应应为 ACK (完成) PDU, 其条件码与完成 (取消) PDU 一致;
- c) 在发出完成 (取消) PDU 并且在接收到预期应答之前, 如果 ACK 规程无效, 任何接收到的 PDU 都会被忽略;
- d) 在该 PDU 中提示的错误会导致事务的终止。

7.1.11.3 挂起通知规程

7.1.11.3.1 一般要求

在文件复制规程中的任意时刻, 协议发送端和接收端都可能会发出挂起通知。

在接收到错误提示或者协议用户端发出的 SUSPEND.request 原语时, 作为应答, 应该发出挂起通知。

如果相关联的事务已经被挂起, 或者被协议接收实体针对无应答模式发送的事务发出的挂起通知, 则应忽略该挂起通知。

下面涉及的发送端和接收端事务挂起的效果是全面的, 包含了所有可能的情况。在某些情况下, 尽管事务正处于挂起状态, 当接收到 EOF 和完成 PDU 时, 应该进行与 ACK 规程和文件复制规程相一致的应答和处理。

7.1.11.3.2 发送端挂起通知规程

当文件复制规程中出现挂起通知时, 协议发送实体将会进行如下操作:

- a) 挂起正在进行的文件分段传输;
- b) 保存事务的状态。

如果当前是有应答模式, 则进行如下操作:

- a) 任何提示 PDU 的传输都应被挂起;
- b) 非活动状态的定时器应被挂起;
- c) 该实体之前发出的应用了 ACK 规程的 PDU 应被挂起。

如果发送实体就是事务的源端, 那么该实体应发出 SUSPENDED.indication 原语。

7.1.11.3.3 接收端挂起通知规程

当提交规程中出现挂起通知时, 协议接收实体将进行如下操作:

- a) 挂起 NAK PDU 的传输;
- b) 挂起任意保活 PDU 的传输;

- c) 挂起非活跃状态的定时器；
- d) 挂起完成（无错误）PDU 的传输；
- e) 挂起该实体之前发出的应用了 ACK 规程的 PDU；
- f) 保存事务状态。

7.1.12 链路状态改变规程

7.1.12.1 一般要求

链路状态改变规程应该应用于 MIB 在 PDU 发送的起始和结束时刻的通知，或者从特定远端协议实体接收 PDU 的时刻。PDU 发送和结束时刻与具体的实现相关。

7.1.12.2 传输时机规程

当向某个特定远端协议实体传输的传输时机结束，协议实体是发送实体并且特定的远端实体是接收实体时，该协议实体应该将全部事务的传输”冻结”。

事务传输的冻结应该与在发送端进行事务的挂起过程具有相同的作用，只是没有发出 SUSPENDED.indication 原语，并且该事务不能认为挂起。

在出现发往特定远端协议实体的传输时机时，并且协议实体本身是发送实体及特定远端实体是接收实体，该发送实体对所有事务的传输进行”解冻”。

事务传输的解冻应该与发送端事务恢复操作具有相同的作用，只是没有发出 RESUME.indication 原语，并且对挂起的事务不起作用。

7.1.12.3 接收时机规程

当从某个特定远端协议实体的接收时机结束，协议实体是接收实体并且特定的远端实体是发送实体时，如果当前不是无应答模式，则该协议实体应该将全部事务的接收过程冻结。

事务接收的冻结应该与在接收端进行事务的挂起过程具有相同的作用，只是没有发出 SUSPENDED.indication 原语，并且该事务不能认为挂起。

在出现发往特定远端协议实体的接收时机时，并且协议实体是接收实体及特定远端实体是发送实体的情况，如果当前不是无应答模式，该接收实体对所有事务的传输进行“解冻”。

事务接收的解冻应该与接收端事务恢复操作具有相同的作用，只是没有发出 RESUME.indication 原语，并且对挂起的事务不起作用。

7.2 扩展规程

7.2.1 一般要求

由于扩展规程是对核心规程的补充并依靠核心规程实现，协议对扩展规程的实现应该同时支持所有核心规程。

特殊情况下，文件复制规程中涉及的中转节点实体应该具有与源实体和目的实体相同的取消、挂起和恢复事务的能力。协议实体在任意时刻作为中转节点实体时，都具有扩展操作的能力：能够接收在文件复制规程过程中发出的服务应答，并且能够提交任何协议服务必要的请求。

7.2.2 文件数据中继规程

7.2.2.1 一般要求

当文件复制规程的接收实体不是文件的目的端时，以下的额外规程将被执行。

PDU 被中转节点中继的传输模式与其被中转节点接收的传输模式相同。

需要注意的是，对于同一个事务，中转节点将同时作为在首次文件复制规程的过程中的接收端和二次文件复制规程过程中的发送端。

7.2.2.2 接收端无应答模式下的文件数据中继规程

协议接收实体在接收到元数据、文件数据或者 EOF（无错误）PDU 时，会中继 PDU 的内容至目的实体或者其他中转节点。

7.2.2.3 发送端有应答模式下的文件数据中继规程

协议发送实体在接收到包含终端系统状态标识为“0”的完成（无错误）PDU，将会发出委托通知。

注：事务在委托通知后仍处于未完成状态，它仍然容易被取消。

7.2.2.4 接收端有应答模式下的文件数据中继规程

对于一个全新的事务接收的第一个 PDU，协议接收实体会作为发送方发起二次文件复制规程，过程与发送方一样。

注1：二次文件复制规程的接收方是文件的目的端或者是其他中转节点。接收实体在规程发起阶段就被指定，根据的是规程开始阶段的 PDU 导头部的目的实体标识符。

注2：文件复制过程中中转节点实现的分段算法并不要求与该文件之前或者后续的发送节点中实现的算法完全一致。这意味着，同一个事务在不同的文件复制过程中文件数据 PDU 的大小将会不同。

注3：在使用了扩展规程的有应答模式下，协议对于数据达到顺序性要求的特征将不再起作用。这些特征包括保活算法和渐进段丢失检测方法。在有应答模式下，重传可能会导致中转节点转发的文件数据分段的顺序与源端发出的顺序不同。

在接收到每个元数据 PDU 或者是文件数据 PDU 后，接收实体应该将 PDU 的内容中继到二次文件复制规程的接收方。

当中转节点实体完成文件传输时，该实体将 EOF（无错误）PDU 中继到二次文件复制规程的接收方，并且会发出包含终端系统状态标识为“0”的完成（无错误）PDU 给文件复制规程的原始发送实体。

7.2.3 完成（无错误）PDU 中继规程

当任意协议实体接收到包含终端系统状态标识为“1”的完成（无错误）PDU，如果该实体不是该事务的源端协议实体，则执行以下在发送端核心规程有应答模式之外的操作：

- a) 协议接收实体将会把 PDU 中继到源端实体或者是其他中转节点；
- b) ACK 规程应该应用于中继完成（无错误）PDU。

7.2.4 取消传播规程

7.2.4.1 当既不是文件源端或是目的端的协议实体接收到 EOF（取消）PDU 时，会执行以下操作：

- a) 该实体发出取消完成通知；
- b) 如果当前是有应答模式，应该对 EOF（取消）PDU 应用 ACK 规程，其预期响应是产生一个 ACK（EOF）PDU；
- c) 协议接收实体应该向文件的目的端中继 EOF（取消）PDU；
- d) 如果当前是有应答模式，应该对中继 EOF（取消）PDU 应用 ACK 规程，其预期响应是产生一个 ACK（EOF）PDU。

注：所有的核心完成（取消）通知规程都应该应用，协议接收实体应该同时作为发送实体和接收实体。

7.2.4.2 当一个协议实体接收到一个完成（取消）PDU 时，若该实体不是文件的源端或目的端，则

会执行以下操作：

- a) 该实体发出完成（取消）通知；
- b) 应该对完成（取消）PDU 应用 ACK 规程，其预期响应是产生一个 ACK（完成）PDU；
- c) 协议接收实体将完成（取消）PDU 中继到文件的源端；
- d) 应该对中继完成（取消）PDU 应用 ACK 规程，其预期响应是产生一个 ACK（完成）PDU。

注：所有的核心完成（取消）通知规程都应该应用，协议接收实体应该同时作为发送实体和接收实体。

7.2.5 延时传输规程

对于无法立即转发的 PDU，协议实体应有延迟传输的能力，将 PDU 发送给目的端或者其他中转节点。

注1：协议实体如何确定 PDU 何时可以发出是与具体实现相关的。

注2：与事务相关的流标签的值将会决定 PDU 传输的顺序，但是流标签的使用方式则是与具体实现相关的。

7.2.6 内部扩展规程

7.2.6.1 委托通知规程

当出现文件复制规程的委托通知时，协议发送实体将会进行如下操作：

- a) 释放文件重传缓冲区中所有未释放的部分；
- b) 停止文件分段和元数据的传输；
- c) 停止提示 PDU 的传输；
- d) 停止该实体之前发出 PDU 的 ACK 规程的应用。

若发送实体就是传输的源端，则协议实体会发出 TRANSFER-CONSIGNED.indication 原语。

7.2.6.2 取消通知规程

当协议实体产生文件复制规程的取消通知，并且该协议实体既不是文件源端也不是目的端的，协议接收实体应该被认为既是发送实体也是接收实体，并且应用所有的核心取消通知规程。

注：EOF（取消）PDU 的接收实体是文件的目的端或者是其他中转节点。完成（取消）PDU 的接收实体是文件的目的端或者是其他中转节点。

7.2.6.3 挂起通知规程

当在既不是文件源端也不是接收端的协议实体发生文件复制规程的挂起通知，协议接收实体应该被认为既是发送实体也是接收实体，并且应用所有的核心挂起通知规程。

7.2.7 扩展恢复规程

当协议实体接收到 RESUME.request 原语时，并且该实体既不是文件源端也不是目的端的协议实体，协议接收实体应该被认为既是发送实体也是接收实体，并且应用所有的核心恢复规程。

8 协议数据单元格式

8.1 位序号约定

位序号的约定如图 7 所示。一个 N 位数据域中：第一个被传送的位（图 7 中最左位）称为位 0，亦即最高有效位，接着是位 1，……直至位 N-1。

本标准提到的字节（Byte）都是 8bit。

数据结构中字节的标号开始于“0”。所有备用字段固定填“0”。

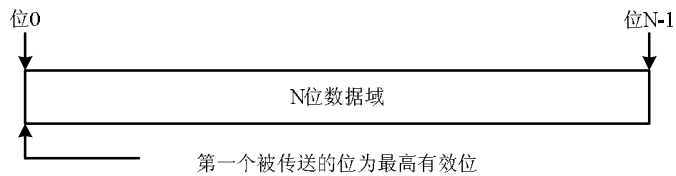


图7 位序号约定

8.2 一般要求

8.2.1 PDU 组成

PDU 由固定格式的 PDU 导头、固定格式或可变格式的 PDU 数据域组成，如图 8 所示。

PDU导头														PDU数据域	
版本	PDU 类型	方向	传输模式	CRC 标识	保留	PDU数据域长度	保留	实体 ID 长度	保留	事务序列号长度	源实体 ID	事务序列号	目的实体 ID	数据域	CRC域 (可选)
3bit	1bit	1bit	1bit	1bit	1bit	16bit	1bit	3bit	1bit	3bit	可变	可变	可变	可变	16bit

图8 PDU组成

PDU 导头的格式见表 1，部分“可选域”的有无由其标识位指定。PDU 导头的传输遵照表 1 的顺序。每个 PDU 通过“源实体 ID”和“事务序列号”与一个特定的事务关联。

表 1 固定的 PDU 导头字段

字段	长度 (bit)	取值	注释
版本	3	000	第 1 个版本
PDU 类型	1	0: 文件指令 1: 文件数据	—
方向	1	0: 发往文件接收方 1: 发往文件发起方	表示 PDU 传输方向
传输模式	1	0: 有应答 1: 无应答	—
CRC 标识	1	0: 无 CRC 域 1: 有 CRC 域	—
保留	1	置 0	—
PDU 数据域长度	16	—	为 PDU 数据域的字节数
保留	1	置 0	—
实体 ID 长度	3	—	实体 ID 的字节数减 1，例如“0”表示实体 ID 长度为 1 字节。在 PDU 导头中适用于所有实体 ID
保留	1	置 0	—
事务序列号长度	3	—	事务序列号的字节数减 1，例如“0”表示事务序列号长度为 1 字节
源实体 ID	可变	—	发起事务的实体的唯一标识，类型为无符号二进制整型
事务序列号	可变	—	在实体发起的所有事务中，唯一地标识一个事务，类型为无符号二进制整型
目的实体 ID	可变	—	传输元数据和文件数据的最终目的地实体的唯一标识，类型为无符号二进制整型

8.2.2 实体 ID

CFDP PDU 的“实体 ID”均为无符号二进制整型，其字节长度由 PDU 中“实体 ID 长度”字段指定。

8.2.3 事务序列号

CFDP PDU 的“事务序列号”均为无符号二进制整型，其字节长度由 PDU 中“事务序列号长度”

字段指定。协议实体响应 PUT.request 原语请求，并按顺序分配序列号。

8.2.4 PDU 类型

定义了两种 PDU 类型：

- a) 文件指令；
- b) 文件数据。

8.2.5 PDU 数据域

PDU 数据域包含多种参数，每种参数可以是下面四种格式中的一种：

- a) 固定格式数据；
- b) LV 格式；
- c) TLV 格式；
- d) 可变长文件数据。

为了优化协议的效率，本标准中对固定格式数据进行了明确定义。其在 PDU 中的位置和长度是固定的。

PDU 中与偏移量相关的字段是描述该字段的字节之前的字节数量，例如，文件首字节的偏移量是零。

不同 PDU 中的实体 ID、事务序列号的长度可能不同。两个不同长度的实体 ID 通过以下方式进行比较，将较短长度的实体 ID 补零，使之与较长长度的实体 ID 的长度一致，然后进行比较。在比较不同长度的事务序列号时也采用这种方式。

8.2.6 LV 格式

LV 格式由长度（L）和值（V）两个字段组成，详见表 2。

表 2 LV 对象格式

字段	字段大小（bit）	字段取值范围
长度	8	0 至 255
值	8*长度	—

LV 格式数据的长度可变，但是它在 PDU 中的位置是不变的。协议可以通过它的长度字段来解析。

8.2.7 TLV 格式

TLV 对象由类型（T）、长度（L）和值（V）三个字段组成，详见表 3。

表 3 TLV 对象格式

字段	字段大小（bit）	字段取值范围
类型	8	见 8.5 节
长度	8	0 至 255
值	8*长度	—

TLV 格式数据在 PDU 中的长度和位置均是可变的。协议可以通过它的类型字段和长度字段来解析。

在接下来的章节中，参数可以表示为固定参数（有明确的域长度）、LV 参数（用“LV”记号表示）或者 TLV 参数（用“TLV hh”记号表示，hh 是 TLV 对象的类型域的十六进制值）。本标准一共定义了 6 种 TLV 参数。

8.3 文件指令 PDU

8.3.1 一般要求

文件指令 PDU 如图 9 所示。其 PDU 数据域包括 8bit 的指令码和可变长度的指令参数域。

PDU 导头														PDU 数据域		
版本 (=0)	PDU 类型 (=0)	方向	传输模式	CRC 标识	保留	PDU 数据域长度	保留	实体 ID 长度	保留	事务序列号长度	源实体 ID	事务序列号	目的实体 ID	指令码	指令参数域	CRC 域 (可选)
3bit	1bit	1bit	1bit	1bit	1bit	16bit	1bit	3bit	1bit	3bit	可变	可变	可变	8bit	可变	16bit

图9 文件指令PDU组成

指令码的定义详见表 4，指令参数域包含多个参数，具体详见后续章节。指令参数域的参数有一个是 4bit 表示的状态码，详见表 5。

表 4 文件指令码

指令码	表示的指令
0x00	保留
0x01	保留
0x02	保留
0x03	保留
0x04	EOF PDU
0x05	完成 PDU
0x06	ACK PDU
0x07	元数据 PDU
0x08	NAK PDU
0x09	提示 PDU
0x0C	保活 PDU
0x0D-0xFF	保留

表 5 状态码

状态码 (二进制)	表示的状态
0000	无错误
0001	达到 ACK 次数上限
0010	达到保活次数上限
0011	错误的传输模式
0100	文件存储被拒绝
0101	文件校验和错误
0110	文件长度错误
0111	达到 NAK 次数上限
1000	检测到不活跃状态
1001	错误的文件结构
1010	达到检测次数上限
1011-1101	保留
1110	收到 SUSPEND.request 请求
1111	收到 CANCEL.request 请求

8.3.2 EOF PDU

带有 EOF PDU 的文件指令的指令参数域内容见表 6。

表 6 EOF PDU 指令参数域内容

参数	长度 (bit)	值	注释
状态码	4	见表 5	—
备用	4	—	—
文件校验和	32	—	参见校验和规程 7.1.2
文件长度	32	—	以字节为单位, 即使状态码不为“无错误”, 文件长度应仍为传输的文件数据字节的总和
故障定位	TLV	一个实体 ID TLV; 参见 8.5.6	如果状态码是“无错误”则省略, 否则, 在 TLV 中的实体 ID 是产生事务取消操作的实体 ID

8.3.3 完成 PDU

带有完成 PDU 的文件指令的指令参数域内容见表 7。

表 7 完成 PDU 指令参数域内容

参数	长度 (bit)	值	注释
状态码	4	见表 5	—
终端系统状态	1	0-中转节点产生 1-终端系统产生	当扩展规程实现时, “0” 在终端系统状态中是有意义的
传输码	1	0-数据完整 1-数据不完整	“数据完整”表示元数据、所有文件数据和 EOF 已经接收完成, 并且完成校验和的确认
文件状态	2	00-传输文件被故意丢弃 01-传输文件因存储拒绝被丢弃 10-传输文件已成功保留在存储区中 11-传输文件状态未报告	仅当事务包含了文件数据的传输时, 文件状态才有意义
文件存储应答	TLVs	见表 17	对于元数据 PDU 的每个文件存储请求 TLV, 必须包含文件存储应答 TLV
故障定位	TLV	一个实体 ID TLV, 参见 8.5.6	如果状态码是“无错误”则省略, 否则, 在 TLV 中的实体 ID 是产生事务取消操作的实体 ID

8.3.4 ACK PDU

带有 ACK PDU 的文件指令的参数域内容见表 8。“事务状态”参数的值定义包括(均是二进制表示):

- a) “00”, 未定义的; ACK PDU 所属的事务在这个实体中当前是不活跃的, CFDP 不会保留事务的历史; 该事务可能之前是活跃的、现在已经被终结, 或者是在这个实体中从来都是不活跃的;
- b) “01”, 活跃的; ACK PDU 所属的事务在这个实体中当前是活跃的;
- c) “10”, 终结的; ACK PDU 所属的事务在这个实体中当前是不活跃的, CFDP 不会保留事务的历史, 该事务之前是活跃的、现在已经被终结;
- d) “11”, 未识别的; ACK PDU 所属的事务在这个实体中当前是不活跃的, CFDP 不会保留事务的历史, 该事务在这个实体中从来都是不活跃的。

表 8 ACK PDU 指令参数域内容

参数	长度 (bit)	值	注释
指令码	4	见表 4, 只有 EOF 和完成 PDU 是被应答的	ACK PDU 的指令代码
指令子类型码	4	—	该值取决于指令码; 对于完成 PDU 的 ACK: 如果完成 PDU 是中转节点产生的, 该值为“0000”, 如果完成 PDU 是终端系统产生的, 该值为“0001”。(当扩展规程实现时, 这种区别是有意义的) 对于其他文件指令的正确应答, 该值为“0000”
状态码	4	见表 5	ACK PDU 的状态码
保留	2	—	—
事务状态	2	—	发出应答的实体的上下文的事务状态

8.3.5 元数据 PDU

带有元数据 PDU 的文件指令的指令参数域内容见表 9。

表 9 元数据 PDU 指令参数域内容

参数	长度 (bit)	值	注释
分段控制	1	0-考虑记录边界 1-不考虑记录边界	—
保留	7	—	全置零
文件长度	32	文件的长度	单位是字节; 对于无记录边界的文件, 被置为全零
源文件名	LV	—	当没有相关的文件时, 比如对于代理操作的消息, LV 的长度字段为零, LV 的值字段被忽略
目的文件名	LV	—	当没有相关的文件时, 比如对于代理操作的消息, LV 的长度字段为零, LV 的值字段被忽略
可选项	TLVS	—	文件存储请求; 给用户的消息; 故障处置选项; 流标签

8.3.6 NAK PDU

带有 NAK PDU 的文件指令的指令参数域内容见表 10。

表 10 NAK PDU 指令参数域内容

参数	长度 (bit)	值	注释
起始范围	32	见 7.1.6.4.2	—
结束范围	32	见 7.1.6.4.2	—
段请求 (*N)	64*N	见表 11	—

每个段请求见表 11。

表 11 段请求组成

参数	长度 (bit)	值	注释
起始偏移	32	数据-请求段的起始偏移量 元数据-00000000 (十六进制)	以字节为单位
结束偏移	32	数据-请求段末尾之后第一个字节偏移 元数据-00000000 (十六进制)	以字节为单位

8.3.7 提示 PDU

带有提示 PDU 的文件指令的指令参数域内容见表 12。

表 12 提示 PDU 指令参数域内容

参数	长度 (bit)	值	注释
要求响应的类型	1	0-NAK 1-保活	—
备用	7	—	—

8.3.8 保活 PDU

带有保活 PDU 的文件指令的参数域内容见表 13。

表 13 保活 PDU 内容

参数	长度 (bit)	值	注释
进度	32	00000000-ffffff(十六进制)	以字节为单位

8.4 文件数据 PDU

文件数据 PDU 组成如图 10 所示。

PDU 导头														PDU 数据域		
版本 (=0)	PDU 类型 (=1)	方向	传输模式	CRC 标识	保留	PDU 数据域长度	保留	实体 ID 长度	保留	事务序列号长度	源实体 ID	事务序列号	目的实体 ID	偏移量	文件数据	CRC 域 (可选)
3bit	1bit	1bit	1bit	1bit	1bit	16bit	1bit	3bit	1bit	3bit	可变	可变	可变	32bit	可变	16bit

图 10 文件数据 PDU 组成

文件数据 PDU 的数据域内容见表 14。

表 14 文件数据 PDU 内容

参数	长度 (bit)	值	注释
偏移量	32	—	以字节为单位
文件数据	可变	—	—

8.5 TLV 参数

8.5.1 文件存储请求 TLV

文件存储请求 TLV 的类型是 0x00。文件存储请求 TLV 的值见表 15。

表 15 文件存储请求 TLV 内容

参数	长度 (bit)	值	注释
操作码	4	见表	—
备用	4	—	—
第一个文件名	LV	—	—
第二个文件名	LV	—	仅对部分操作码，该字段才有效

操作码和参数见表 16。

表 16 文件存储请求 TLV 操作码

操作码（二进制）	功能	是否含有第二个文件名
“0000”	创建文件	否
“0010”	删除文件	否
“0010”	重命名文件	是
“0011”	追加文件	是
“0100”	替换文件	是
“0101”	创建目录	否
“0110”	删除目录	否
“0111”	拒绝文件（如果存在则删除）	否
“1000”	拒绝目录（如果存在则删除）	否

8.5.2 文件存储应答 TLV

文件存储应答 TLV 的类型为 0x01。文件存储应答 TLV 的值见表 17。文件存储应答状态码见表 18。

表 17 文件存储应答 TLV 内容

参数	长度（bit）	值	注释
操作码	4	—	—
状态码	4	参见表 18	—
第一个文件名	LV	—	—
第二个文件名	LV	—	仅对部分操作码，该字段才有效
文件存储消息	LV	—	如果没有文件存储消息参数，LV 中的长度字段为 0，数据字段被忽略

表 18 文件存储应答状态码

操作码（二进制）	功能	文件名的使用	状态码（二进制）
“0000”	创建	第一个文件名-将要创建的文件名	0000-成功 0001-不允许创建 1111-未执行
“0001”	删除	第一个文件名-将要删除的文件名	0000-成功 0001-文件不存在 0010-不允许删除 1111-未执行
“0010”	重命名	第一个文件名-旧文件名 第二个文件名-新文件名	0000-成功 0001-旧文件名不存在 0010-新文件名已经存在 0011-不允许重命名 1111-未执行
“0011”	追加	第一个文件名-将成为新文件的第一部分的文件名，也作为新文件的文件名 第二个文件名-将成为新文件的第二部分的文件名	0000-成功 0001-文件名 1 不存在 0002-文件名 2 不存在 0011-不允许追加 1111-未执行

表 18 (续)

操作码 (二进制)	功能	文件名的使用	状态码 (二进制)
“0011”	追加	第一个文件名-将成为新文件的第一部分的文件名, 也作为新文件的文件名 第二个文件名-将成为新文件的第二部分的文件名	0000-成功 0001-文件名 1 不存在 0002-文件名 2 不存在 0011-不允许追加 1111-未执行
“0100”	替换	第一个文件名-文件内容将要被替换的文件名 第二个文件名-将要替换第一个文件名的内容的文件名	0000-成功 0001-文件名 1 不存在 0002-文件名 2 不存在 0011-不允许替换 1111-未执行
“0101”	创建目录	第一个文件名-将要创建的目录名称	0000-成功 0001-不允许创建目录 1111-未执行
“0110”	删除目录	第一个文件名-将要删除的目录名称	0000-成功 0001-文件不存在 0010-不允许删除 1111-未执行
“0111”	拒绝文件	第一个文件名-将要被删除的文件名	0000-成功 0010-不允许删除 1111-未执行
“1000”	拒绝目录	第一个文件名-将要被删除的目录名	0000-成功 0010-不允许删除 1111-未执行

8.5.3 给用户的消息 TLV

给用户的消息 TLV 类型是 0x02, 该消息的值的格式未定义。

为了更好地进行交互, 每个消息很有必要包含一些标识。标准 CFDP 代理和目录操作消息 (见第 9 章) 的消息 ID 将 ASCII 码 “cfdp” 放在每个消息的前 4 字节。

8.5.4 故障处置选项 TLV

故障处置选项 TLV 的类型是 0x04, 它的值只有 1 字节, 编码见表 19。

表 19 故障处置选项域编码

参数	长度 (bit)	值	注释
状态码	4	见表 5	这些状态码对于“无错误”、“收到 SUSPEND.request 请求”和“收到 CANCEL.request 请求”不适用, 因为这些状态码不是错误。
处理代码	4	0000-用于扩展 0001-产生取消通告 0010-产生挂起通告 0011-忽略错误 0100-放弃事务 0101-1111-保留	—

8.5.5 流标签 TLV

流标签 TLV 的类型是 0x05, 值的格式未定义。

8.5.6 实体 ID TLV

实体 ID TLV 的类型是 0x06, 值是一个实体 ID, 见 8.2。

9 用户操作

9.1 预留消息格式

用户操作可通过在不同 CFDP 用户之间进行预留的 CFDP 消息交换完成。这些消息作为 MTU TLV 格式中的值，通过 CFDP 元数据传输。

每个消息都由“预留消息头”和随后的消息内容组成，消息头格式见表 20，消息内容的格式随着消息类型变化。

表 20 预留消息头

区域	长度 (bit)	取值	注释
消息标识符	32	ASCII 码的“cfdp”	用于从所有用户消息中识别出“预留消息”
消息类型	8	详见后面表格	

“预留消息”的消息内容中要包含的项目见第 7 章协议规程。

消息内容域可以为固定格式、可变格式或者 LV 格式，见 8.2 描述。

“源端事务 ID 消息”对于所有用户操作都一致，其消息类型为 0x0A，内容格式见表 21。

表 21 源端事务 ID 消息

区域	长度 (bit)	取值	注释
保留位	1	置“0”	—
实体 ID 长度	3	—	源实体 ID 的字节数减 1，例如“0”表示源实体 ID 长度为 1 字节
保留位	1	置“0”	—
事务序列号长度	3	—	事务序列号的字节数减 1，例如“0”表示事务序列号长度为 1 字节
源实体 ID	可变	—	事务发起实体的唯一标识 无符号二进制整型
事务序列号	可变	—	从所有本实体发起的事务中，唯一标识某事务。 无符号二进制整型

9.2 代理操作

9.2.1 一般要求

“代理操作”是指用户（操作“发起方”）使用协议服务驱动一些远端协议实体用户（操作“应答方”）发起文件传输至其他实体用户（操作“受益方”）。代理操作的受益方可以是操作的发起方（在这种情况下，代理操作即实现文件“获取”功能）也可以是其他第三方协议实体用户。

为实现互操作，用户必须实现并遵循以下代理操作规范。

9.2.2 代理操作消息类型

适用于“代理操作”的预留消息的消息类型取值见表 22。

表 22 代理操作消息类型

消息类型	含义
0x00	代理提交请求
0x01	代理 MTU
0x02	代理文件存储请求
0x03	代理故障处置选项
0x04	代理传输模式
0x05	代理流标签
0x06	代理分段控制
0x07	代理提交应答
0x08	代理文件存储应答
0x09	代理取消提交

9.2.3 启动代理操作

9.2.3.1 一般要求

需要启动一个代理操作时，用户需使用 PUT.request 原语请求发送一个 FDU，其元数据包含一个或多个“预留消息”，消息详细定义见 9.2.3.2 到 9.2.3.8，相应 PUT.request 原语参数见 6.5.1。

作为 FDU 的目的地，远端用户（应答方）转而使用这些消息内容，制定另一个 PUT.request 原语以完成所需的远程文件传输。

由协议实体 TRANSACTION.indication 原语为原始的 PUT.request 请求分配一个事务 ID，报告给发起方用户，并包含在事务 PDU 导头中发送给远程代理用户，其代理操作的结果报告中应引用该事务 ID。

9.2.3.2 代理提交请求

代理提交请求消息是需强制实现的必选项，消息格式见表 23。

表 23 代理提交请求消息

区域	长度 (bit)	取值	注释
目的实体 ID	可变	LV	受益方协议实体的 ID
源文件名	可变	LV	若该参数缺省则长度为 0
目标文件名	可变	LV	若该参数缺省则长度为 0

9.2.3.3 代理 MTU

可选择包含一个或多个“代理 MTU”消息，消息格式见表 24。

表 24 代理 MTU 消息

区域	长度 (bit)	取值	注释
消息内容	可变	LV	见 8.5.3

9.2.3.4 代理文件存储请求

可选择包含一个或多个“代理文件存储请求”消息，消息格式见表 25。

表 25 代理文件存储请求消息

区域	长度 (bit)	取值	注释
长度 (Byte)	8	请求的字节长度	—
请求	8*长度	单个文件存储请求，参数值见表 15	—

9.2.3.5 代理故障处置选项

可选择包含一个或多个“代理故障处置选项”消息，消息格式见表 26。

表 26 代理故障处置选项消息

区域	长度 (bit)	取值	注释
故障处置选项	8	见 8.5.4	—

9.2.3.6 代理传输模式

“代理传输模式”为可选实现项，消息格式见表 27。

表 27 代理传输模式消息

区域	长度 (bit)	取值	注释
缺省	7	全“0”	—
传输模式	1	见表 1	—

9.2.3.7 代理流标签

“代理流标签”为可选实现项，消息格式见表 28。

表 28 代理流标签消息

区域	长度 (bit)	取值	注释
流标签	可变	LV	见 8.5.5

9.2.3.8 代理分段控制

“代理分段控制”为可选实现项，消息格式见表 29。

表 29 代理分段控制消息

区域	长度 (bit)	取值	注释
缺省	7	全“0”	—
分段控制	1	见表 9	—

9.2.4 代理提交请求的应答

当收到“代理提交请求”消息，应答方用户需要使用 PUT.request 原语申请发送一个 FDU，包括以下参数：

- a) 目的实体 ID，在“代理提交请求”消息中已指定；
- b) 源文件名，若在“代理提交请求”消息中有指定的源文件名；
- c) 目标文件名，若在“代理提交请求”消息中有指定的目标文件名；
- d) 分段控制，若接收到同一事务的 METADATA-RECV.indication 中“代理分段控制消息”有指定的分段控制；
- e) 传输模式，若接收到同一事务的 METADATA-RECV.indication 中“代理传输模式消息”有指定的传输模式；
- f) 流标签，若接收到同一事务的 METADATA-RECV.indication 中“代理流标签消息”有指定的流标签；
- g) 接收到同一事务的 METADATA-RECV.indication 的各“代理故障处置选项”消息中的一个；
- h) 接收到同一事务的 METADATA-RECV.indication 的各“代理 MTU”消息中的一个；
- i) 接收到同一事务的 METADATA-RECV.indication 的各“代理文件存储请求”消息中的一个；

j) 一个消息类型为“源端事务 ID”的预留消息，引用其“事务 ID”启动代理提交请求。

9.2.5 代理提交请求的结果报告

9.2.5.1 一般要求

一旦收到一个 TRANSACTION-FINISHED.indication 原语，说明已经完成了相应代理提交请求启动的事务，作为被请求的应答方用户需使用 PUT.request 原语启动一个新的事务，请求发送一个 FDU 给代理操作的发起方，其元数据中包含每一个消息类型为“代理传输应答”（见 9.2.5.2）的预留消息和源端事务 ID，以及多个（可以没有）消息类型为“代理文件存储应答”（见 9.2.5.3）的预留消息。

若因任何原因 PUT.request 不能实现，应答方用户应停止这个代理操作的后续动作。

9.2.5.2 代理传输应答

“代理传输应答”消息是需强制实现的必选项，消息格式见表 30。

表 30 代理传输应答 Put Response 消息

区域	长度 (bit)	取值	注释
状态码	4	见表 5	—
缺省	1	置“0”	—
传输码	1	见表 7	—
文件状态	2	见表 7	—

9.2.5.3 代理文件存储应答

如果 TRANSACTION-FINISHED.indication 要求有文件存储应答，那么每一个应答需包括一个代理文件存储应答消息，格式见表 31。

表 31 代理文件存储应答 Filestore Response 消息

区域	长度 (bit)	取值	注释
长度	8	应答的字节长度	—
应答	8*长度	单个协议文件存储应答的内容（见表 17 定义）	—

9.2.6 取消代理提交请求

9.2.6.1 一般要求

当需要取消一个代理操作时，代理操作发起方需要使用 PUT.request 原语请求发送一个 FDU，其元数据包含每一个消息类型为“代理传输取消”（见 9.2.6.2）的预留消息和源端事务 ID。

9.2.6.2 取消代理提交

取消代理提交消息是需强制实现的必选项，该消息没有内容。

9.2.7 取消代理提交的应答

当收到一个取消代理提交消息，应答方用户需使用 CANCEL.request 原语请求取消由相应源端事务 ID 消息所唯一标识的事务。需根据 9.2.5 提交执行的结果报告。

9.3 目录操作

9.3.1 一般要求

为实现互操作，用户必须实现并遵循以下协议目录操作规范。

9.3.2 目录操作消息类型

各个用于“目录操作”的预留消息的消息类型取值见表 32。

表 32 目录操作消息类型

消息类型（十六进制）	判读
10	目录列表请求 Directory Listing Request
11	目录列表应答 Directory Listing Response

9.3.3 启动目录列表

9.3.3.1 一般要求

当需要启动一个目录列表操作，用户需使用 PUT.request 原语请求发送一个 FDU，其元数据包含每一个消息类型为“目录列表请求”（见 9.3.3.2）的预留消息。

9.3.3.2 目录列表请求

目录列表请求消息格式见表 33。

表 33 目录列表请求消息

区域	长度（bit）	取值	注释
目录名	可变	LV	—
目录文件名	可变	LV	给请求方用户的本地存储的文件名称和路径，应答方用户需按此放置目录列表文件

9.3.4 目录列表请求的应答

9.3.4.1 一般要求

当收到一个“目录列表请求”，用户需使用 PUT.request 原语请求发送一个 FDU 给目录列表请求消息的发起方实体。该 FDU 的文件数据是一个目录列表文件，该 FDU 的元数据包含每一个消息类型为“目录列表应答”（见 9.3.4.2）的预留消息和源端事务 ID。

如果远端用户无法提供目录列表文件，其应答的 FDU 则不含有文件数据，由目录列表应答消息中的“列表应答码”指示文件列表请求不成功。

9.3.4.2 目录列表应答

目录列表应答消息格式见表 34。

表 34 目录列表应答消息

区域	长度（bit）	取值（十六进制）	注释
列表应答码	8	00~7F: 成功 80~FF: 不成功	—
目录名	可变	LV	目录列表请求中提出的目录名
目录文件名	可变	LV	目录列表请求中提出的，给请求方用户的，本地存储的列表文件名称和路径

9.3.5 取回目录列表

当收到一个预留消息的消息类型是“目录列表应答”时，请求方用户可从消息文件中取回列表。

9.4 远程状态报告操作

9.4.1 一般要求

为实现互操作，用户必须实现并遵循以下协议远程状态报告规范。

9.4.2 远程状态报告操作消息类型

“远程状态报告”消息的消息类型取值见表 35。

表 35 远程状态报告操作消息类型

消息类型（十六进制）	判读
20	远程状态报告请求 Remote Status Report Request
21	远程状态报告应答 Remote Status Report Response

9.4.3 启动远程状态报告

9.4.3.1 一般要求

当需要启动一个远程状态报告操作，用户需使用 PUT.request 原语请求发送一个 FDU，其元数据包含的预留消息定义见 9.4.3.2。

9.4.3.2 远程状态报告请求

远程状态报告请求消息格式见表 36。

表 36 远程状态报告请求消息

区域	长度（bit）	取值（十六进制）	注释
保留位	1	置“0”	—
实体 ID 长度	3	—	源实体 ID 的字节数减 1，例如“0”表示源实体 ID 长度为 1 字节
保留位	1	置“0”	—
事务序列号长度	3	—	事务序列号的字节数减 1，例如“0”表示源实体 ID 长度为 1 字节
源实体 ID	可变	—	发起传输、请求远程报告的实体的唯一标识
事务序列号	可变	—	从所有本实体发起的事务中，唯一标识某事务；无符号二进制整型
报告文件名	可变	LV	给请求方用户的本地存储的文件名称和路径，应答方用户需按此放置状态报告文件

9.4.4 远程状态报告请求的应答

9.4.4.1 一般要求

当收到一个“远程状态报告请求”，远端用户需使用 PUT.request 原语请求发送一个 FDU 给请求的发起方实体。该 FDU 的文件数据是一个指定事务的状态报告，该 FDU 的元数据包含每一个消息类型为“远程状态报告应答”（见 9.4.4.2）的预留消息和源端事务 ID。

如果远端用户无法提供状态报告文件，其应答的 FDU 则不含有文件数据，由远程状态报告应答消息中的“报告应答码”指示远程状态报告请求不成功。

如果远端用户可以提供状态报告文件，状态报告的性质见 6.3.8 描述。

9.4.4.2 远程状态报告应答

远程状态报告应答消息格式见表 37。

表 37 远程状态报告应答消息

区域	长度（bit）	取值（十六进制）	注释
传输状态	2	与 ACK PDU 定义的事务状态一致，见表 8	—
缺省	5	全“0”	—
报告应答码	1	0=不成功 1=成功	—

表 37 (续)

区域	长度 (bit)	取值 (十六进制)	注释
缺省	1	置“0”	本字段与后面 5 个字段一起确定所报告的事务, 参数取值于“远程状态报告请求”
实体 ID 长度	3	—	—
保留位	1	置“0”	—
事务序列号长度	3	—	—
源实体 ID	可变	—	无符号二进制整型
事务序列号	可变	—	无符号二进制整型

9.4.5 取回远程状态报告

当收到的协议消息包含有一个远程状态报告应答消息, 则请求方用户将收到一个指定文件的远程状态报告。

9.5 远程挂起操作

9.5.1 一般要求

为实现互操作, 用户必须实现并遵循以下协议远程挂起操作规范。

9.5.2 远程挂起操作消息类型

“远程挂起 MTU 消息”的消息类型取值见表 38。

表 38 远程挂起操作消息类型

消息类型 (十六进制)	判读
30	远程挂起请求 Remote Suspend Request
31	远程挂起应答 Remote Suspend Response

9.5.3 启动远程挂起

9.5.3.1 一般要求

当需要启动一个远程挂起操作, 用户需使用 PUT.request 原语请求发送一个 FDU, 其元数据包含的预留消息定义见 9.5.3.2。

9.5.3.2 远程挂起请求

远程挂起请求消息格式见表 39。

表 39 远程挂起请求消息

区域	长度 (bit)	取值	注释
保留位	1	置“0”	—
实体 ID 长度	3	—	源实体 ID 的字节数减 1, 例如“0”表示源实体 ID 长度为 1 字节
保留位	1	置“0”	—
事务序列号长度	3	—	事务序列号的字节数减 1, 例如“0”表示源实体 ID 长度为 1 字节
源实体 ID	可变	—	需挂起的事务的发起方实体的唯一标识
事务序列号	可变	—	从所有本实体发起的事务中, 唯一标识某事务; 无符号二进制整型

9.5.4 远程挂起请求的应答

9.5.4.1 一般要求

当收到一个“远程挂起请求”，用户需使用 SUSPEND.request 原语确认可挂起的指定事务，再使用 PUT.request 原语请求发送一个 FDU 给挂起请求消息的发起方实体。该 FDU 的元数据包含每一个消息类型为“远程挂起应答”（见 9.5.4.2）的预留消息和源端事务 ID。

有可能在同时间有多种驱动挂起同一事务，如，一个协议实体收到一个远程挂起请求，要求挂起一个已经被本地故障所挂起的事务。用户应用需要管理这些挂起命令队列，使被挂起的这个事务保持挂起状态，直至所有的挂起命令均被恢复命令所撤销。若事务由于检测到故障导致挂起，需保持挂起状态直至用户应用确定这个故障状态不再存在。同时，一个由协议实体发起远程挂起请求而挂起的事务，需保持挂起直至收到来自同一用户的远程恢复请求。操作方式由具体实现决定。

9.5.4.2 远程挂起应答

远程挂起应答消息格式见表 40。

表 40 远程挂起应答消息

区域	长度 (bit)	取值 (十六进制)	注释
挂起指示	1	“0”——未挂起 “1”——挂起	—
传输状态	2	与 ACK PDU 定义的事务状态一致，见表 8	—
缺省	5	全“0”	—
缺省	1	置“0”	本字段与后面 5 个字段一起确定被请求挂起的事务，参数取值于“挂起请求”
实体 ID 长度	3	—	—
缺省	1	置“0”	—
事务序列号长度	3	—	—
源实体 ID	可变	—	无符号二进制整型
事务序列号	可变	—	无符号二进制整型

9.6 远程恢复操作

9.6.1 一般要求

为实现互操作，用户必须实现并遵循以下协议远程恢复操作规范。

9.6.2 远程恢复操作消息类型

“远程恢复 MTU 消息”的消息类型取值见表 41。

表 41 远程状态报告操作消息类型

消息类型 (十六进制)	判读
38	远程恢复请求 Remote Resume Request
39	远程恢复应答 Remote Resume Response

9.6.3 启动远程恢复

9.6.3.1 一般要求

当需要启动一个远程恢复操作，用户需使用 PUT.request 原语请求发送一个 FDU，其元数据包含的预留消息定义见 9.6.3.2。

9.6.3.2 远程恢复请求

远程恢复请求消息格式见表 42。

表 42 远程恢复请求消息

区域	长度 (bit)	取值 (十六进制)	注释
保留位	1	置“0”	—
实体 ID 长度	3	—	源实体 ID 的字节数减 1, 例如“0”表示源实体 ID 长度为 1 字节
保留位	1	置“0”	—
事务序列号长度	3	—	事务序列号的字节数减 1, 例如“0”表示源实体 ID 长度为 1 字节
源实体 ID	可变	—	需恢复的事务的发起方实体的唯一标识
事务序列号	可变	—	从所有本实体发起的事务中, 唯一标识某事务; 无符号二进制整型

9.6.4 远程恢复请求的应答

9.6.4.1 一般要求

当收到一个远程恢复请求消息, 用户需使用 RESUME.request 原语确认可恢复的指定事务, 再使用 PUT.request 原语请求发送一个 FDU 给挂起请求消息的发起方实体。该 FDU 的元数据包含每一个消息类型为“远程恢复应答”(见 9.6.4.2)的预留消息和源端事务 ID。

注: 挂起命令队列的管理需求见 9.5.4.1。

9.6.4.2 远程恢复应答

远程恢复应答消息格式见表 43。

表 43 远程恢复应答消息

区域	长度 (bit)	取值 (十六进制)	注释
挂起指示	1	“0”——未挂起 “1”——挂起	由于可能同时间有多种驱动同时挂起事务, 因此对事务成功执行远程恢复请求, 并不一定会改变其挂起状态
传输状态	2	与 ACK PDU 定义的事务状态一致, 见表 8	—
缺省	5	全“0”	—
缺省	1	置“0”	本字段与后面 5 个字段一起确定被请求挂起的事务, 参数取值于“挂起请求”
实体 ID 长度	3	—	—
缺省	1	置“0”	—
事务序列号长度	3	—	—
源实体 ID	可变	—	无符号二进制整型
事务序列号	可变	—	无符号二进制整型

9.7 存储与中继转发覆盖操作

9.7.1 一般要求

SFO 系统是一种用于“在始终无法直接通信的 CFDP 协议实体的用户间传送文件”的可选机制, 该机制的实现不依赖于扩展规程。

每个传输文件以逐跳传输的方式在中转节点用户(相当于扩展规程中的中转节点协议实体)中接

收、存储与中继转发，直至到达代理用户（其协议实体可以与目的端用户直接通信）。被传输的文件及所有关联的元数据被封装在一个“SFO 传送 FDU”中传送至每一个节点，SFO 传送 FDU 的元数据还包含一个“SFO 请求 MTU”，标识了传送的特征。

代理用户将收到的 SFO 传送 FDU，内容包括原始文件和从 SFO 传送事件自身元数据中提取出的原始关联元数据，通过常规的单跳协议传输发送给目的端用户，称作 SFO 最终传送事务。这个操作类似代理文件传输的“应答”，只是发送给目的端用户 SFO 请求信息外还包括所有的原始元数据。

路由或中继转发中的传送错误、成功（可选）情况，可使用“SFO 传送报告 FDU”向源端用户反馈报告。可能的话，传送失败、成功（可选）情况也向最终目的端用户报告。

SFO 操作的图示见图 11。SFO 传送报告 FDU 既可能是代理的中转实体向目的完成传送后产生，也可能是任一节点路由或中继传送错误或成功（可选）报告（即“追踪”进程）。报告既送给发起方源用户也送给最终目的实体，同样报告本身也根据需要进行转发。

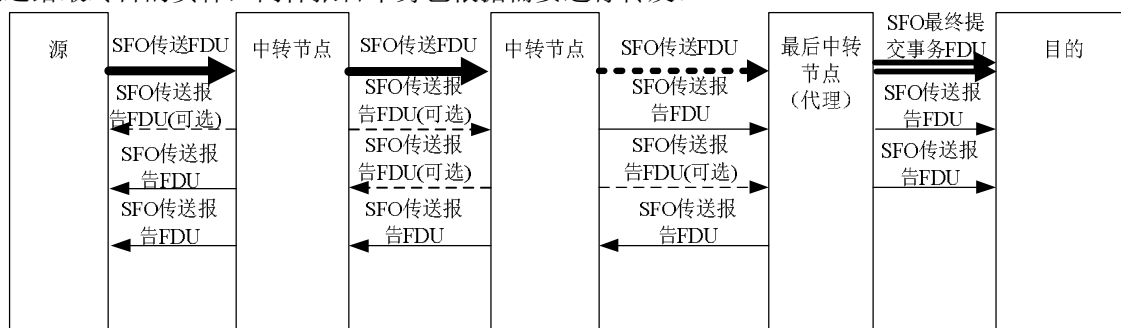


图11 SFO操作

为实现互操作，用户必须实现并遵循 9.7.2 至 9.7.7 的协议存储与中继转发规范。

9.7.2 SFO 消息类型

各个用于 SFO 操作的预留消息的消息类型取值见表 44。

表 44 SFO 消息类型

消息类型（十六进制）	判读
40	SFO 请求 Request
41	SFO MTU
42	SFO 流标签 Flow Label
43	SFO 故障处置选项 fault handler override
44	SFO 文件存储请求 Filestore Request
45	SFO 报告 Report
46	SFO 文件存储应答 Filestore Response

9.7.3 路由

SFO 路由步骤如下：

- a) 用户应用需可以访问“所有关联的已知协议实体 ID 及其可以直接接收文件的所有其他实体”的基础信息；
- b) 用户应用需对该基础信息进行计算，得到与最终目的端用户之间最短可能路径的路由表；路由表是一组协议实体的表单，起始为用户应用自身的本地实体，结束于最终目的端用户的本地

实体；表单内每个实体均可通过协议从表单中前一个实体直接接收文件。

9.7.4 启动 SFO 传送

9.7.4.1 一般要求

为了启动 SFO 传送文件和关联元数据，用户需调用 SFO 路由程序（见 9.7.3）以确定本地协议实体到最终目的端用户的路径。

若计算出的路由表为空表单，则传送无法实现，启动 SFO 传送的尝试直接失败。

若计算出的路由表只包含本地协议实体，则用户应用就是文件最终传送目标，启动 SFO 传送的尝试直接失败。

若计算出的路由表只包含本地协议实体和最终用户，则用户应用可直接使用 PUT.request 原语请求直接发送文件和元数据给最终目标，不需进一步启动 SFO 功能。

若计算出的路由表除了本地协议实体和最终用户以外，还包含一个或者多个其他协议实体（中转节点），则用户应用需使用 PUT.request 原语请求发送一个“SFO 传送 FDU”给路由表中第一个节点。SFO 传送 FDU 的文件数据为给最终用户的文件，元数据包括一个 SFO 请求消息（见 9.7.4.3 至 9.7.4.6 定义），消息内容从与文件的关联元数据中获取。“目标文件名”在 PUT.request 原语中规定格式为：

SFOsourceEntityID_SFOrequestLabel.sfo

其中 SFOsourceEntityID 是 ASCII 码表示的 SFO 请求消息中给出的源实体 ID，省略起始的“省略”，同样 SFOrequestLabel 也由 SFO 请求消息给出。

注 1：文件名省略路径名，是为了避免与中转节点文件系统中的路径命名规则相冲突。

注 2：用户应用使用 SFO 除了保存传送至本地的文件，还应在本地文件存储中保留一定空间供存储需转发的临时文件。在具体实现上，任何从 SFO 转发过程收到的文件都需注意空间分配避免引起溢出。

请求方用户应用在该 PUT.request 原语中判断选取需要的传输模式、分段控制、故障处置选项和流标签，约束协议传输结果。

注：传输完成后的任意时间，用户应用可选择删除其传输文件的本地副本。

9.7.4.2 SFO 请求

SFO 请求消息是需强制实现的必选项，消息格式见表 45。

表 45 SFO 请求消息

区域	长度 (bit)	取值 (十六进制)	注释
追踪控制标识	2	0=不需追踪 1=只向源追踪 2=只向目标追踪 3=向双方向追踪	控制传送节点的转发成功报告
传输模式	1	见本标准前文规定	指定由代理节点传送至最终目标的传输模式
分段控制	1	见本标准前文规定	指定由代理节点传送至最终目标的分段控制
缺省	4	全 0	—

表 45 (续)

区域	长度 (bit)	取值 (十六进制)	注释
之前节点计数	8	由发起方源用户置 0, 由路由各节点逐一增加	表明请求被节点实体接收、转发的次数
SFO 请求标签	可变	LV	格式由发起方源用户特定实现来规定, 可用于解释发起方用户的传送目的。 可打印的 ASCII 码文本
源实体 ID	可变	LV	发起方源用户的协议实体 ID。 无符号二进制整型
目的实体 ID	可变	LV	最终目的端用户的协议实体 ID。 无符号二进制整型
源文件名	可变	LV	若该参数缺省则长度为 0
目标文件名	可变	LV	若该参数缺省则长度为 0

9.7.4.3 SFO MTU

可选择包含一个或多个“SFO MTU”消息, 消息格式见表 46。

表 46 SFO MTU 消息

区域	长度 (bit)	取值	注释
消息内容	可变	LV	封装 MTU 文本给最终目的端用户

9.7.4.4 SFO 文件存储请求 Filestore Request

可选择包含一个或多个“SFO 文件存储请求”消息, 消息格式见表 47。

表 47 SFO 文件存储请求消息

区域	长度 (bit)	取值	注释
长度 (Byte)	8	请求的字节长度	
请求	8*长度	单个协议文件存储请求, 见本标准前文规定	封装文件存储请求给最终目的端用户的本地协议实体执行

9.7.4.5 SFO 故障处置选项 Fault Handler Override

可选择包含一个或多个“SFO 故障处置选项”消息, 消息格式见表 48。

表 48 SFO 故障处置选项消息

区域	长度 (bit)	取值	注释
故障处置选项	8	见本标准前文规定	用于由代理至最终目标的传送

9.7.4.6 SFO 流标签 Flow Label

“SFO 流标签”为可选实现项, 消息格式见表 49。

表 49 SFO 流标签 Flow Label 消息

区域	长度 (bit)	取值	注释
流标签	可变	LV, 见本标准前文规定	用于由代理至最终目标的传送

9.7.5 SFO 传送文件数据单元的分程传送

9.7.5.1 当接收到一个 TRANSACTION-FINISHED.indication, 表明成功接收了一个文件数据单元, 其元数据包含一个 SFO 请求消息。用户调用 SFO 路由程序 (见 9.7.3) 确定从本地协议实体到最终目的实体的路由表。

9.7.5.2 若计算出的路由表为空表单，则传送无法实现。用户需使用 SFO 报告程序（见 9.7.6）向发起方源用户应用报告路由失败。

9.7.5.3 若计算出的路由表只包含本地协议实体，则用户应用就是文件最终传送目标，不需进一步启动 SFO 功能。

9.7.5.4 若计算出的路由表除了本地协议实体和最终用户以外，还包含一个或者多个其他节点，则确立了一个分程事务。

9.7.5.5 若 SFO 请求消息的“之前节点计数”结果数量超过最大可能范围（255），则该分程传送不被接受。用户需使用 SFO 报告程序（见 9.7.6）向发起方源用户应用报告该请求的点数超最大范围。

9.7.5.6 若文件数据单元的元数据中所指定的目标文件名与前面 9.7.4.1 规定的格式不符，用户需使用 SFO 报告程序（见 9.7.6）向发起方源用户应用报告目标文件名有误。

9.7.5.7 此外，用户应用需使用 PUT.request 原语请求发送接收到的文件（若有的话）和所有收到的 SFO 请求、SFO 流标签、SFO 故障处置选项、SFO MTU 以及 SFO 文件存储请求消息给路由表中的第一个节点。在 PUT.request 原语中指定的目标文件名，即为接受到文件数据单元元数据中的目标文件名。请求方用户应用需判断选取需要的传输模式、分段控制、故障处置选项和流标签，约束协议传输结果。

9.7.5.8 若分程传送在除了“无错误”以外任何情况下终止，则用户需使用 SFO 报告程序（见 9.7.6）向发起方源用户以及最终目的端用户应用报告分程传送失败。

注：该报告无条件执行，与接受到的 SFO 请求消息中“跟踪控制标识”的取值无关。

9.7.5.9 若分程传送在“无错误”情况下终止，倘若 SFO 请求消息中“跟踪控制标识”不为零，用户应用需使用 SFO 报告程序（见 9.7.6）向发起方源用户或最终目的端用户（根据“跟踪控制标识”的取值）报告分程传送成功。

注：传输完成后的任意时间，用户应用可选择删除其传输文件的本地副本。

9.7.5.10 若计算出的路由表只包含本地协议实体和最终用户，则用户应用可视作 SFO 传送操作的代理节点，需启动一个“SFO 最终传送事件”，使用 PUT.request 原语请求发送一个 FDU，包含以下参数：

- a) 目的实体 ID，在“SFO 提交请求”消息中已指定；
- b) 源文件名，若在“SFO 提交请求”消息中有指定的源文件名；
- c) 目标文件名，若在“SFO 提交请求”消息中有指定的目标文件名；
- d) 分段控制，在“SFO 提交请求”消息中已指定；
- e) 传输模式，在“SFO 提交请求”消息中已指定；
- f) 流标签，若接收到同一事务的 METADATA-RECV.indication 原语中“SFO 流标签消息”中有指定的流标签；
- g) 接收到同一事务的 METADATA-RECV.indication 原语中各“SFO 故障处置选项消息”中的一个；
- h) 接收到同一事务的 METADATA-RECV.indication 原语中各“SFO MTU”消息中的一个；
- i) 接收到同一事务的 METADATA-RECV.indication 原语中各“SFO 文件存储请求消息”中的一个；
- j) 整个 SFO 请求消息即一个附加的 MTU，是一类预留消息，定义见 9.7.4.2。

当传输结束，代理用户应用需使用 SFO 报告程序（见 9.7.6）向发起方源用户应用报告最终文件传

送的成功或失败结果。

注：传输完成后的任意时间，用户应用可选择删除其传输文件的本地副本。

9.7.6 SFO 报告要求

9.7.6.1 一般要求

当需要报告 SFO 文件传送结果时，用户应用需要调用 SFO 路由程序（见 9.7.3）确定从本地协议实体到源端发起用户和（或）最终目的端用户的路由表。

若计算出向源端的路由表为空表单，则无法向源端传送，不需进一步启动 SFO 功能。

若计算出向源端的路由表只包含本地协议实体，则用户应用就是文件的发送源，不用再向源端报告，故不需进一步启动 SFO 功能。

若计算出向源端的路由表包括一个以上的实体，用户应用需使用 PUT.request 原语请求发送一个“SFO 传送报告 FDU”，包括以下参数：

- a) 目的实体 ID，设为路由表中第二个实体的 ID（本地协议实体之后第一个）；
- b) 无文件、源文件名或者目标文件名；
- c) 一个单独的 SFO 报告消息，定义见 9.7.6.2；
- d) 多个（可以没有）SFO 文件存储应答消息，定义见 9.7.6.3。

当传输完成，或因任何原因 PUT.request 不能实现，用户应用需停止这个 SFO 传送的后续动作。

若计算出向目的端的路由表为空表单，则无法向目的端传送，不需进一步启动 SFO 功能。

若计算出向目的端的路由表只包含本地协议实体，则用户应用就是文件的最终目标，不用再向目的端报告，故不需进一步启动 SFO 功能。

若计算出向目的端的路由表包括一个以上的实体，用户应用需使用 PUT.request 原语请求发送一个“SFO 传送报告 FDU”，包括以下参数：

- a) 目的实体 ID，设为路由表中第二个实体的 ID（本地协议实体之后第一个）；
- b) 无文件、源文件名或者目标文件名；
- c) 一个单独的 SFO 报告消息，定义见 9.7.6.2；
- d) 多个（可以没有）SFO 文件存储应答消息，定义见 9.7.6.3。

当传输完成，或因任何原因 PUT.request 不能实现，用户应用需停止这个 SFO 传送的后续动作。

9.7.6.2 SFO 报告

SFO 报告消息是每个 SFO 传送报告中的必选项，消息格式见表 50。如果 SFO 传送报告是一个 SFO 最终传送事件的结果，则该消息的“状态码”、“传输码”和“文件状态”需从 TRANSACTION-FINISHED.indication 原语的完成状态中获取，否则该三个字段的值不确定。

表 50 SFO 报告消息

区域	长度 (bit)	取值 (十六进制)	注释
SFO 请求标签	可变	LV	从该 SFO 传送的 SFO 请求消息中拷贝“请求标签”
源实体 ID	可变	LV	该 SFO 传送的发起方源端 CDFP 实体。 无符号二进制整型
目的实体 ID	可变	LV	该 SFO 传送的最终目标 CDFP 实体。 无符号二进制整型
报告实体 ID	可变	LV	发出该 SFO 传送报告的 CDFP 实体的 ID。 无符号二进制整型

表 50 (续)

区域	长度 (bit)	取值 (十六进制)	注释
之前节点计数	8	由报告用户应用置 0, 由路由各节点逐一增加	表明报告被节点实体接收、转发的次数
SFO 请求的之前节点计数	8	从该 SFO 传送的 SFO 请求消息中拷贝“之前节点计数”	—
报告码	8	1、2、3、4、5、6 或 7	1: 最终文件传送完成; 2: 分段传送失败; 3: 路由失败, 无法分段传送; 4: 分段传送成功; 5: 最终文件传送失败; 6: 节点数超上限; 7: 目标文件名有误, 无法分段传送
状态码	4	见本标准前文规定	—
区域	长度 (bit)	取值 (十六进制)	注释
方向	1	0: 报告去往源端用户应用; 1: 报告去往目的端用户应用	—
传输码	1	见本标准前文规定	—
文件状态	2	见本标准前文规定	—

9.7.6.3 SFO 文件存储请求

若 SFO 传送报告是一个 SFO 最终文件传送事件的结果, 并且若 TRANSACTION-FINISHED.indication 的传输完成信号中包括一个或多个文件存储应答, 则相应每个应答都包含一个 SFO 文件存储应答消息, 消息格式见表 51。

表 51 SFO 文件存储应答消息

区域	长度 (bit)	取值 (十六进制)	注释
长度	8	应答的字节长度	—
应答	8*长度	单个协议文件存储应答的内容, 见本标准前文规定	封装了一个由本地协议实体产生到最终目的端用户的文件存储请求

9.7.7 SFO 传送报告的分程传送

9.7.7.1 当接收到一个 TRANSACTION-FINISHED.indication, 表明成功接收了一个文件数据单元, 其元数据包含一个 SFO 报告消息。用户调用 SFO 路由程序 (见 9.7.3) 确定从本地协议实体到报告发送方向 (在 SFO 报告消息的“方向”中指定) 的用户应用的路由表。

9.7.7.2 若计算出的路由表为空表单, 则传送无法实现。用户应用不需对该 SFO 传送做进一步操作。

9.7.7.3 若计算出的路由表只包含本地协议实体, 则用户应用就是文件最终传送目标, 不需进一步启动 SFO 功能。

9.7.7.4 若计算出的路由表包含一个以上的节点, 则确立了一个分程事务。

9.7.7.5 若 SFO 请求消息的“之前节点计数”结果数量超过最大可能范围 (255), 则该分程传送不被接受, 不需进一步启动 SFO 功能。

9.7.7.6 此外, 用户应用需使用 PUT.request 原语请求将接收到的 SFO 报告和所有收到的 SFO 文件存储应答消息, 发送至路由表中第二个实体 (本地协议实体之后第一个)。当传输完成, 或因任何原因 PUT.request 不能兑现, 用户应用需停止这个 SFO 传送的后续动作。

10 业务类别

10.1 业务类别定义

标准定义的业务类别具体如下：

- a) 类别 1——不可靠传输；
- b) 类别 2——可靠传输；
- c) 类别 3——通过一个或多个级联中转节点的不可靠传输；
- d) 类别 4——通过一个或多个级联中转节点的可靠传输。

注：下面各节对适用于每一种业务类别的功能和规程进行了描述，这些功能和规程的详细描述详见第 7 章。同时为了应用这些业务类别，还需对 5.3 描述的协议特性，各业务类别规程中指定的相关章节，以及规程中引用的 PDU（第 8 章）等内容进行参考。

10.2 类别 1——不可靠传输的功能

10.2.1 类别 1——源端规程

类别 1 提供了从源端到目的端的定界或不定界数据文件的不可靠传输操作。针对类别 1 的功能，源端协议实体所使用的规程具体如表 52 所述。

表 52 类别 1 源端规程

规程	章节	备注
PDU 发送实体中 CRC 规程	7.1.1.1	—
CRC 验证规程	7.1.1.3	—
校验和规程	7.1.2	—
提交规程	7.1.3	—
事务启动通知规程	7.1.4	—
PDU 转发规程	7.1.5	仅适用于元数据、文件数据以及 EOF PDU
发送端实体文件复制规程	7.1.6.1.2	—
发送端无应答模式规程	7.1.6.3.1	—
恢复规程	7.1.6.7	适用于无应答模式下的发送实体的规程
报告规程	7.1.6.8	—
错误处理规程	7.1.8	基本的互操作行为为发布取消通知
文件存储规程	7.1.9	仅限适用于发送实体的规程
内部规程	7.1.11	适用于无应答模式下的发送实体的规程
传输时机规程	7.1.12.2	—

10.2.2 类别 1——目的端规程

针对类别 1 的功能，目的端协议实体使用的规程具体如表 53 所示。

表 53 类别 1 目的端规程

规程	对应章节	备注
PDU 接收实体的 CRC 规程	7.1.1.2	—
CRC 验证规程	7.1.1.3	—
校验和规程	7.1.2	—
PDU 转发规程	7.1.5	适用于丢弃的错误传输 PDU
接收端实体文件复制规程	7.1.6.1.3	—
接收端无应答模式规程	7.1.6.3.2	—
接收端取消应答规程	7.1.6.6.1	不需要肯定应答
报告规程	7.1.6.8	—
错误处理规程	7.1.8	基本的互操作行为为丢弃事务
文件存储规程	7.1.9	仅限适用于接收实体的规程
不活跃监控规程	7.1.10	—
内部规程	7.1.11	适用于转发的不完整数据以及无应答模式中接收实体的规程

10.2.3 类别 1 事件图

类别 1 事件图如图 12 所示。

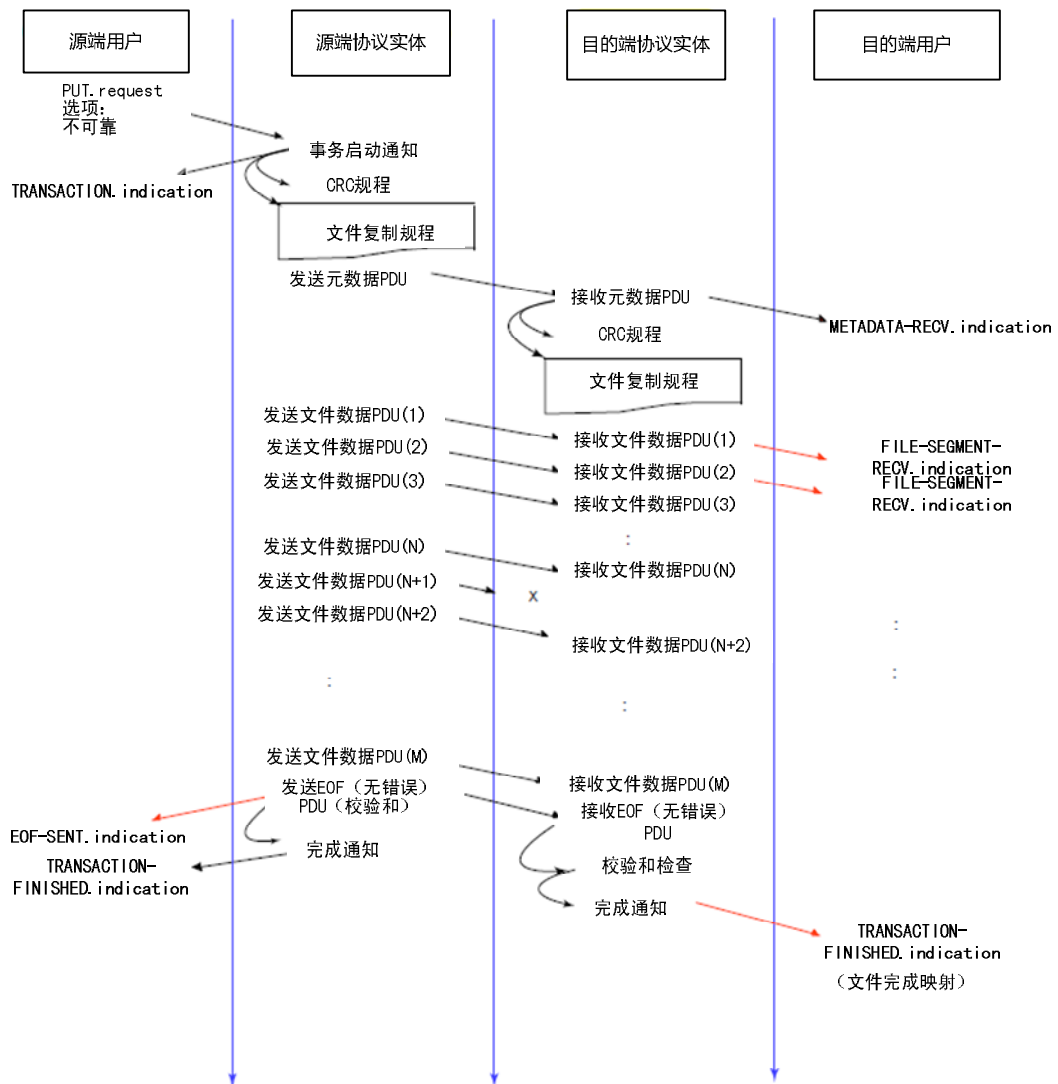


图 12 类别 1 过程示意图

10.3 类别 2——可靠传输的功能

10.3.1 类别 2——源端规程

类别 2 为从源端到目的端的定界或不定界数据文件提供可靠的传输操作。针对类别 2 的功能，源端协议实体使用的规程具体见表 54。

表 54 类别 2 源端规程

规程	章节	备注
CRC 规程	7.1.1	—
校验和规程	7.1.2	—
提交规程	7.1.3	—
事务启动通知规程	7.1.4	—
PDU 转发规程	7.1.5	—
发送端文件复制规程	7.1.6.1.2	—
发送端有应答模式规程	7.1.6.4.1	—
发送端渐进段丢失检测规程	7.1.6.4.4	—
发送端保活规程	7.1.6.5.3	—
发送端取消应答规程	7.1.6.6.2	—
恢复规程	7.1.6.7	仅限适用于发送实体中的规程
报告规程	7.1.6.8	—
PDU 发送端 ACK 规程	7.1.7.1	—
PDU 接收端 ACK 规程	7.1.7.2	—
错误处理规程	7.1.8	仅需发布取消通知
文件存储规程	7.1.9	仅限适用于发送实体中规程
不活跃监控规程	7.1.10	—
内部规程	7.1.11	仅限适用于发送实体中规程
链路状态改变规程	7.1.12	—

10.3.2 类别 2——目的端规程

针对类别 2 的功能，目的端协议实体使用的规程具体见表 55。

表 55 类别 2 目的端规程

规程	章节	备注
CRC 规程	7.1.1	—
校验和规程	7.1.2	—
PDU 转发规程	7.1.5	—
接收端文件复制规程	7.1.6.1.2	—
接收端有应答模式规程	7.1.6.4.2	—
接收端渐进段丢失检测规程	7.1.6.4.3	互操作中仅需要延迟模式
接收端保活规程	7.1.6.5.2	—
接收端取消应答规程	7.1.6.6.1	需要肯定的确认
恢复规程	7.1.6.7	仅限适用于接收实体的规程
报告规程	7.1.6.8	—

表 55 (续)

规程	章节	备注
PDU 发送端 ACK 规程	7.1.7.1	—
PDU 接收端 ACK 规程	7.1.7.2	—
错误处理规程	7.1.8	仅需发布取消通知
文件存储规程	7.1.9	仅限适用于接收实体的规程
不活跃监控规程	7.1.10	—
内部规程	7.1.11	适用于转发的不完整数据以及无应答模式中接收实体的规程
链路状态改变规程	7.1.12	—

10.3.3 类别 2 事件图

类别 2 事件图如图 13 所示。

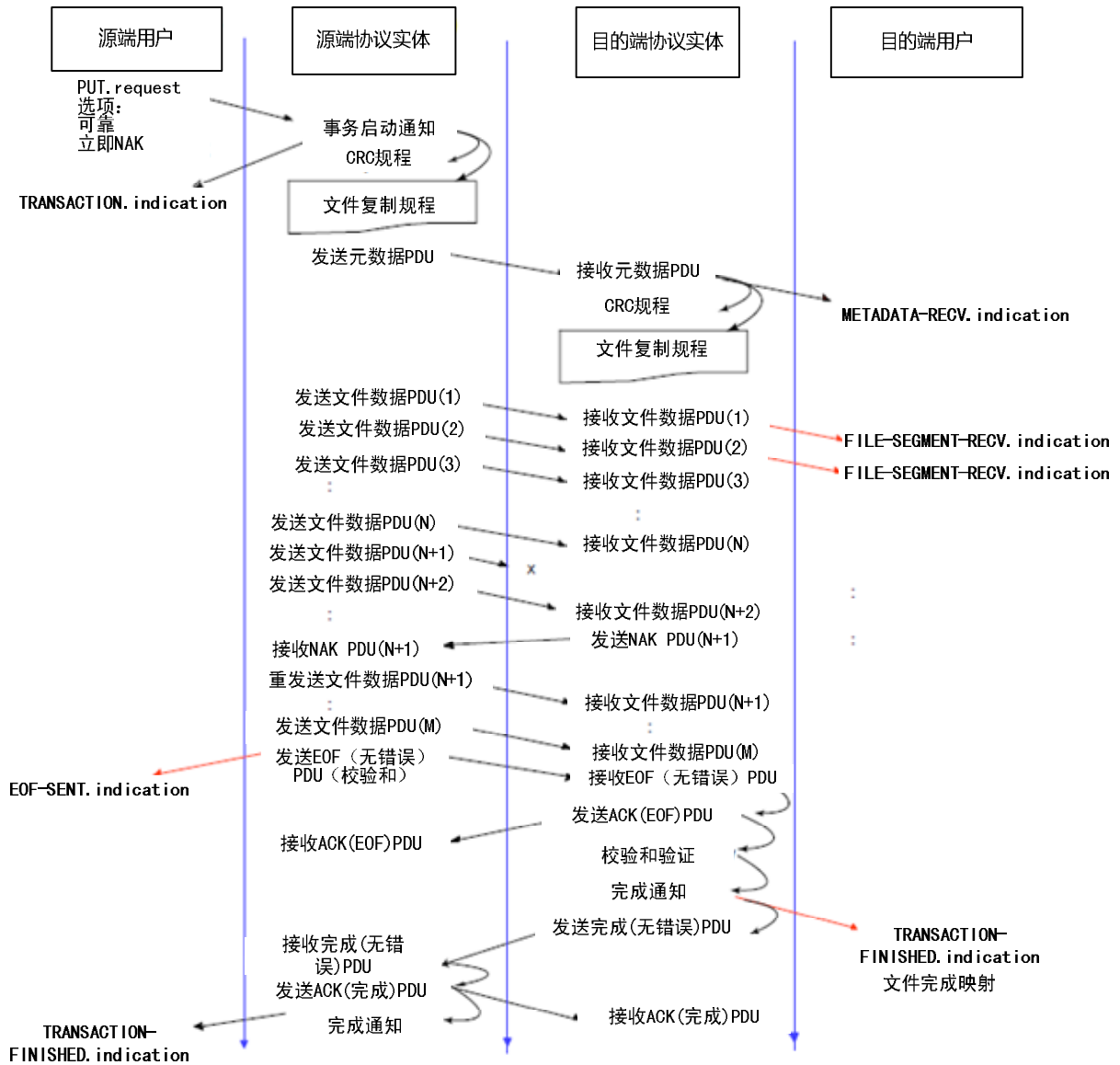


图 13 类别2过程示意图

10.4 类别 3——通过一个或多个级联中转节点的不可靠传输的功能

10.4.1 类别 3——源端规程

类别 3 为经过级联的一个或多个中转节点传递从源端到目的端的定界或不定界数据文件提供不可靠传输。针对类别 3 功能，源端协议实体使用的规程具体见表 56。

表 56 类别 3 源端规程

规程	章节	备注
PDU 发送实体中 CRC 规程	7.1.1.1	—
CRC 验证规程	7.1.1.3	—
校验和规程	7.1.2	—
提交规程	7.1.3	—
事务启动通知规程	7.1.4	—
PDU 转发规程	7.1.5	仅适用于元数据、文件数据以及 EOP 等 PDU
发送端文件复制规程	7.1.6.1.1	—
发送端无应答模式规程	7.1.6.3.1	—
恢复规程	7.1.6.7	适用于无应答模式下发送实体的规程
报告规程	7.1.6.8	—
错误处理规程	7.1.8	仅需发布取消通知
文件存储规程	7.1.9	仅限适用于发送实体的规程
内部规程	7.1.11	仅限适用于无应答模式下发送实体的规程
传输时机规程	7.1.12.2	—

10.4.2 类别 3——中转节点规程

针对类别 3 功能，中转节点协议实体使用的规程具体见表 57。

表 57 类别 3 中转节点规程

规程	章节	备注
CRC 规程	7.1.1	—
校验和规程	7.1.2	—
PDU 转发规程	7.1.5	仅适用于元数据、文件数据以及 EOF PDU
发送端文件复制规程	7.1.6.1.1	—
接收端文件复制规程	7.1.6.1.2	—
发送端无应答模式规程	7.1.6.3.1	—
接收端无应答模式规程	7.1.6.3.2	—
接收端取消应答规程	7.1.6.6.1	无需肯定确认
恢复规程	7.1.6.7	适用于无应答模式下发送实体的规程
报告规程	7.1.6.8	用户应用是可选的,但是如果存在用户应用则必须实现报告规程
错误处理规程	7.1.8	仅需放弃事务
不活跃监控规程	7.1.10	适用于无应答模式下的发送以及接收实体的规程
内部规程	7.1.11	—

表 57 (续)

规程	章节	备注
传输时机规程	7.1.12.2	—
接收端无应答模式下的文件数据中继规程	7.2.2.2	—
完成(无错)PDU 中转规程	7.2.3	—
取消传播规程	7.2.4	—
延迟传输规程	7.2.5	—
取消通知规程	7.2.6.2	—
挂起通知规程	7.2.6.3	—
扩展的恢复规程	7.2.7	—

10.4.3 类别 3——目的端规程

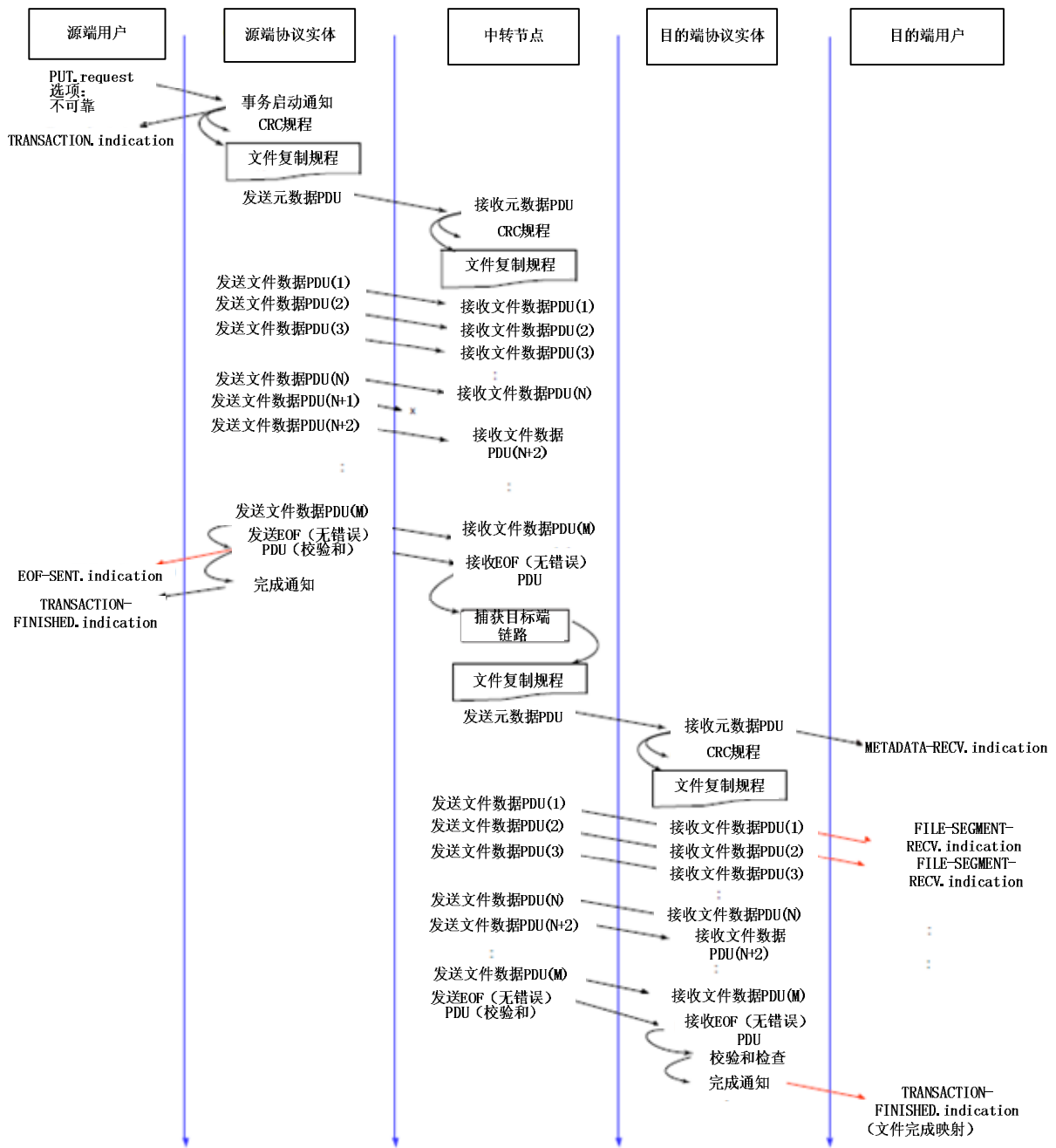
针对类别 3 功能，目的端协议实体使用的规程具体见表 58。

表 58 类别 3 目的端规程

规程	章节	备注
接收实体中的 CRC 规程	7.1.1.2	—
CRC 验证规程	7.1.1.3	—
校验和规程	7.1.2	—
PDU 转发规程	7.1.5	仅适用于丢弃的错误传输 PDU
接收端文件复制规程	7.1.6.1.3	—
接收端无应答模式规程	7.1.6.3.2	—
接收端取消应答规程	7.1.6.6.1	无需肯定的确认
报告规程	7.1.6.8	—
错误处理规程	7.1.8	仅需放弃事务
文件存储规程	7.1.9	仅限于适用于接收实体的规程
不活跃监控规程	7.1.10	—
内部规程	7.1.11	适用于转发的不完整数据以及无应答模式中接收实体的规程

10.4.4 类别 3 事件图

类别 3 事件图如图 14 所示。



注：中转节点使用挂起/恢复规程以及复制文件规程会和用户应用之间存在潜在的交互。当协议实体作为中转节点使用时，实体自身的功能即为用户应用，因此为了使图中的描述更为清晰，与用户应用之间的交互并没有在图中显示出来。潜在的交互仅限于业务类别 2 图中描述的 `EOF-SENT.indication`, `Suspend.indication`, and `RESUME.request` 原语。

图14 类别3过程示意图

10.5 类别 4——通过一个或多个级联中转节点的可靠传输

10.5.1 类别 4——源端规程

类别 4 提供有限或无限的数据文件通过一个或多个级联中转节点媒介从源端到目的端的可靠传输。对于类别 4 功能，源协议实体应使用表 59 中指定的规程。

表 59 类别 4 源端规程

规程	章节	备注
CRC 规程	7.1.1	—
校验和规程	7.1.2	—
提交规程	7.1.3	—
事务启动通知规程	7.1.4	—
PDU 转发规程	7.1.5	—
发送端文件复制规程	7.1.6.1.2	—
发送端有应答模式规程	7.1.6.4.1	—
发送端渐进段丢失检测规程	7.1.6.4.4	—
发送端保活规程	7.1.6.5.3	—
发送端取消应答规程	7.1.6.6.2	—
恢复规程	7.1.6.7	仅适用于发送实体可用的规程
报告规程	7.1.6.8	—
PDU 发送端 ACK 规程	7.1.7.1	—
PDU 接收端 ACK 规程	7.1.7.2	—
错误处理规程	7.1.8	仅撤销通知发布需要
文件存储规程	7.1.9	仅适用于发送实体可用的规程
不活跃监控规程	7.1.10	—
内部规程	7.1.11	仅适用于发送实体可用的规程
链路状态改变规程	7.1.12	—
发送端有应答模式下的文件数据中继规程	7.2.2.3	—
委托通知规程	7.2.6.1	—

10.5.2 类别 4——中转节点规程

对于第四类业务，中转节点协议实体应使用表 60 中指定的规程。

表 60 类别 4 中转节点规程

规程	章节	备注
CRC 规程	7.1.1	—
校验和规程	7.1.2	—
PDU 转发规程	7.1.5	—
发送端文件复制规程	7.1.6.1.1	—
接收端文件复制规程	7.1.6.1.3	—
发送端有应答模式规程	7.1.6.4.1	—
接收端有应答模式规程	7.1.6.4.2	—
接收端渐进段丢失检测规程	7.1.6.4.3	延迟模式仅在互操作时才需要
发送端渐进段丢失检测规程	7.1.6.4.4	—
保活规程	7.1.6.5	—
取消应答规程	7.1.6.6	需要肯定应答 不完整的数据转发
恢复规程	7.1.6.7	—

表 60 (续)

规程	章节	备注
报告规程	7.1.6.8	用户应用程序是可选的，但是当报告规程存在时，用户应用程序必须应用
PDU 发送端 ACK 规程	7.1.7.1	—
PDU 接收端 ACK 规程	7.1.7.2	—
错误处理规程	7.1.8	仅撤销通知发布需要
不活跃监控规程	7.1.10	—
内部规程	7.1.11	仅适用于发送实体可用的规程
链路状态变化规程	7.1.12	—
文件数据中继规程	7.2.2	—
完成（无错误）PDU 中转规程	7.2.3	—
取消传播规程	7.2.4	—
延时传输规程	7.2.5	—
内部扩展规程	7.2.6	—
扩展恢复规程	7.2.7	—

10.5.3 类别 4——目的端规程

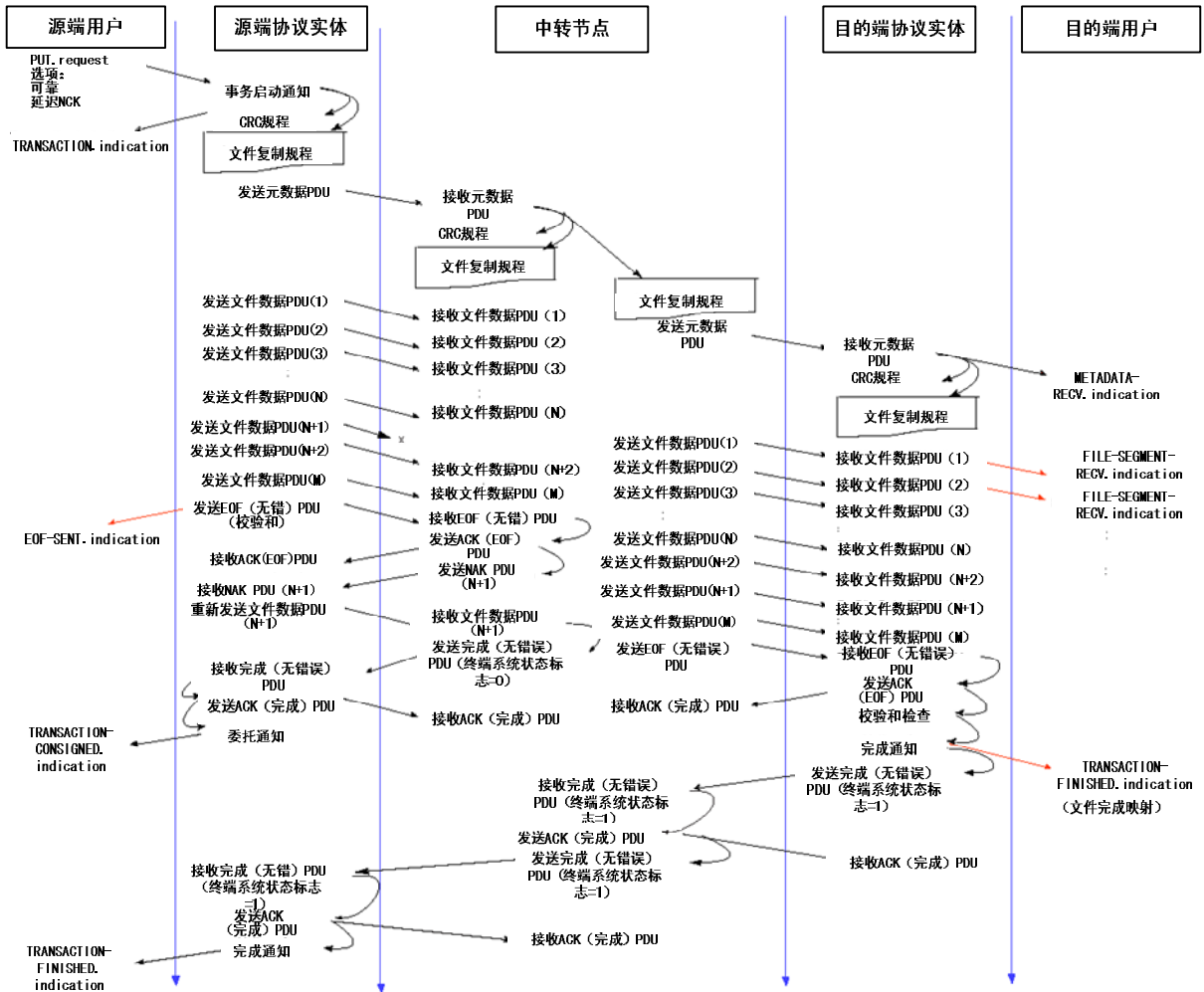
对于第四类业务规程，目的实体应使用表 61 规定的规程。

表 61 类别 4 目的端规程

规程	章节	备注
CRC 规程	7.1.1	—
校验和规程	7.1.2	—
PDU 转发规程	7.1.5	—
接收端文件复制规程	7.1.6.1.3	—
接收端有应答模式规程	7.1.6.4.2	—
接收端渐进段丢失检测规程	7.1.6.4.3	延迟模式仅在互操作时才需要
接收实体保活规程	7.1.6.5.2	—
接收端取消应答规程	7.1.6.6.1	需要肯定应答。 不完整的数据转发
恢复规程	7.1.6.7	仅适用于发送实体可用的规程
报告规程	7.1.6.8	—
PDU 发送端 ACK 规程	7.1.7.1	—
PDU 接收端 ACK 规程	7.1.7.2	—
错误处理规程	7.1.8	仅撤销通知发布需要
文件存储规程	7.1.9	仅适用于接收实体可用的规程
不活跃监控规程	7.1.10	—
内部规程	7.1.11	不完整的数据转发。 仅适用于接收实体可用的规程
链路状态变化规程	7.1.12	—
延时传输规程	7.2.5	—

10.5.4 类别4事件图

类别4事件图如图15所示。



注：中转节点发送实体使用挂起/恢复规程和复制文件规程必然需要与用户应用程序存在潜在交互过程。当一个协议实体作为一个中转节点，实体本身功能就是用户应用程序；因此为了清晰起见，图中并未画出与用户应用程序的交互过程。潜在的交互过程被限制在 EOF-SENT.indication, SUSPEND.indication 和 RESUME.request 原语，如类别2事件图所示。

图15 类别4过程示意图

11 管理信息库

每个协议实体由单独的管理信息库参数定义。参数的相关定义如下所述。

本地协议实体配置信息表详见表 62，该表列出了本标准中所有本地协议实体的管理信息库配置信息参数及其使用方式。

表 62 本地协议实体配置信息参数

参数	内容
是否发送 EOF-SENT.indication	类型为布尔型, 取值 1 或 0
是否发送 EOF-RCV.indication	类型为布尔型, 取值 1 或 0
是否发送 FILE-SEGMENT-RCV.indication	类型为布尔型, 取值 1 或 0
作为发送端实体时是否发送 TRANSACTION-FINISHED.indication	类型为布尔型, 取值 1 或 0
作为接收端实体时是否发送 SUSPENDED.indication	类型为布尔型, 取值 1 或 0
作为接收端实体时是否发送 RESUMED.indication	类型为布尔型, 取值 1 或 0
默认错误处理	对每种错误类型, 规定其默认的处理方式, 为 7.1.8 所示默认处理方式中的其中一种
路由信息 (仅扩展规程)	对系统中每个协议实体, 路由信息包含所有相邻的实体

表 63 列出了本标准中所有远端协议实体的管理信息库配置信息参数及其使用方式。远端协议实体使用参数与其直接交互的本地实体采用的参数一致。

表 63 远端协议实体配置信息参数

参数	内容
UT 地址	当传输给远端协议实体时采用该参数
单向光时	用于计算 NAK 以及正向 ACK 时间间隔
队列往返延时允许时间	用于计算 NAK 以及正向 ACK 时间间隔
异步模式 NAK 间隔	时间间隔或者 N/A
异步模式保活间隔	时间间隔或者 N/A
异步模式报告间隔	时间间隔或者 N/A
立即 NAK 模式使能	布尔型 (1、0)
提示模式传输间隔	时间间隔或者 N/A
默认传输模式	有应答或无应答模式
事务取消时对不完整接收文件的处置	丢弃或保留
是否需要 CRC	布尔型 (1、0)
最大文件段长度	字节
保活限值	字节或者 N/A
ACK 定时器超时限值	超限值
NAK 定时器超时限值	超限值
事务不活跃限值	时间限制值
传输时机开始时间	运行环境产生
传输时机结束时间	运行环境产生
接收时机开始时间	运行环境产生
接收时机结束时间	运行环境产生

附 录 A
(资料性附录)
校验和计算示例

本附录列出了文件校验和计算示例。为便于对示例部分的理解，计算算法规则如下：

- a) 规则 1：对于数据单元长度偏移量不是 4 的整数倍的文件，为计算校验和，需要将数据补齐为 4 字节整数倍长度。因此，对于偏移量为 Q 的数据需要插入 N 字节的 0，且 $N=Q\%4$ ；
- b) 规则 2：为计算包含 M 个字节且 M 小于 4 的序列的校验和，必须补 0 进行计算。补 0 的方式是在字节序列的最前面补 4-M 个 0。

假设一个文件包含 15 个字节：

00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e

若最大分割长度为 6，则文件将通过三个文件数据 PDU 发送，见表 A.1。

表 A.1 文件数据 PDU 内容

偏移量	长度	内容
0	6	00 01 02 03 04 05
6	6	06 07 08 09 0a 0b
12	3	0c 0d 0e

为计算校验和，将文件数据按字（4 字节）方式分割并累加，使得字节偏移量为 4 的整数倍。由此，第一个字由字节偏移量为 0 的字节置于字的高位：

00 ____

拷贝剩余 3 字节数据组成第一个字：

00 01 02 03

第 2 个字将字节偏移量为 4 的字节置于字的高位：

04 ____

拷贝剩余 3 字节数据组成第二个字：

04 05 06 07

第 3 个字将字节偏移量为 8 的字节置于字的高位：

08 ____

拷贝剩余 3 字节数据组成第三个字：

08 09 0a 0b

为计算最后剩余文件长度为 3 字节的剩余字，将字节偏移量为 12 的字节置于字的高位：

0c ____

拷贝剩余 2 字节数据组成第四个字：

0c 0d 0e _

最后添加 $(4-3)=1$ 个字节的 0 在第四个字的最末尾：

0c 0d 0e 00

校验和按照 4 个 32 位字计算：

00010203

+04050607
 +08090a0b
 +0c0d0e00

按照各个文件接收顺序计算校验和，假设文件接收顺序见表 A.2。

表 A.2 文件接收顺序

偏移量	长度	内容
0	6	00 01 02 03 04 05
12	3	0c 0d 0e
6	6	06 07 08 09 0a 0b

从接收到第一个文件数据单元中恢复出如下数据字：

00 01 02 03
 04 05 00 00

从第二个文件数据单元中恢复出如下数据字：

0c 0d 0e 00

第三个文件数据单元中采用略微不同的方式，按照规则 1 描述。由于数据单元长度为 6，并非 4 的整数倍，因此将数据单元按照 4 对齐，分成 2 个字。对于偏移量为 Q 的数据需要在第一个字节前插入 N 字节的 0，且 $N=Q\%4$ 。而 6 模 4 为 2，因此需要在数据单元的最前面插入 2 个 0，使得数据单元由：

06 07 08 09 0a 0b

变为

00 00 06 07 08 09 0a 0b

按照这种方式重新组织数据单元的数据：

00 00 06 07
 08 09 0a 0b

按照上述数据重新计算的校验和为：

00010203
 +04050000
 +0c0d0e00
 +00000607
 +08090a0b

由于 $0400000+00000607=04050607$ ，这正是前面例子中的第二个字。从这个例子可以看出，校验和计算与文件数据单元的接收顺序无关。

中华人民共和国航天行业标准

空间数据系统文件传输协议

QJ 20750—2018

*

中国航天标准化研究所出版

北京市丰台区小屯路 89 号

邮政编码：100071

中国航天标准化研究所

印务发行部印刷、发行

版权专有 不得翻印

*

2018 年 5 月出版

定价：148 元