飞机数据网络
第 7 部分
航空电子全双工交换式
以太网（AFDX）网络

AIRCRAFT DATA NETWORK

PART 7

AVIONICS FULL DUPLEX SWITCHED

ETHERNET (AFDX) NETWORK

ARINC 664 part 7 规范

（翻译稿）

ARINC SPECIFICATION 664P7

PUBLISHED: June 27, 2005

目录

1 .0 引言

1.1 本文档的目的

定义本文档的用意在于定义一个确定性网络：航空电子全双工交换式以太网（Avioics Full Duplex Switched Ethernet, AFDX™）。AFDX™是空中客车（Airbus）公司拥有的一个商标，它被允许在文档中使用。本文档还在AFDX语境下突出航空电子系统附加的性能需求。

本规范使：

- 系统集成者使用 AFDX 设计航空关键系统;
- 装置设计者设计出可以与 AFDX 互操作的设备。

1.2 范围

文档列出这些需求是为了规定可互操作的功能元素，它们是：

- 数据链路层：MAC, VL 寻址概念
- 网络层：IP, ICMP
- 传输层：UDP, TCP （可选）
- 网络应用层：Sampling, Queuing, SAP, TFTP 和 SNMP

本文档将被限于对上面列出的规范化协议的描述。

注释

系统设计者可能逐项地加入其他的协议（例如在网络应用层加入FTP协议），但并不保证AFDX兼容的LRU（line replaceable unit，外场可替换单元）实现这个附加的协议。

这里没有定义物理层，但它可以采用在ARINC 664 Part 2中定义的任何解决方案。

这意味着任何AFDX兼容的LRU（包括交换机），只要遵循形状和装配的需求（由该系统具体决定，未在本文档中定义），可以与任何AFDX网络连接。

1.3 文档的组织

1.3.1 ARINC 664 规范，飞机数据网络

ARINC 664规范定义安装于飞机上的以太网数据网络。它被发展为多个部分，列举如下：

Part 1 – 系统概念和概况

Part 2 – 以太网物理层和数据链路层

Part 3 – 基于因特网的协议与服务

Part 4 – 基于因特网的地址分配与编号

Part 5 – 网络互连服务和功能元件

Part 6 – 保留

Part 7 – 航空电子全双工交换式以太网（AFDX）网络

Part 8 – 上层服务

1.4 相关的文档

1.4.1 本文档与其它 ARINC 标准的关系

当使用本规范提供的能力开发航空电子系统和子系统标准时，它们应该通过引用合并本规范中的条款。对于本规范的引用应该确保是本规范最近的版本。与本规范有关系的其他ARINC文档被列出如下：

ARINC Specification 653: 航空电子应用软件标准接口（Avionics Application Software Standard Interface）

ARINC Specification 615A: 使用以太网接口的软件数据加载器（Software Data Loader using Ethernet Interfaces）（应为ARINC Report 615A——译者注）

ARINC Specification 665: 可加载的软件标准（Loadable Software Standards）（应为ARINC Report 615A——译者注）

1.4.2 与工业标准的关系

IEEE 802.3标准2000年版，被本规范作为一个完整的部分考虑，并认为是需要阅读的。在本文档中，当引用到这个标准，标题简化为 “IEEE 802.3”。

1.4.3 RTCA 和 EUROCAE 文档

RTCA和EUROCAE文档发展适用于航空电子装置、系统和处理过程的最小操作性能标准（Minumum Operational Performance Standards, MOPS）。下列RTCA和EUROCAE文档的最新版本适合于本规范：

RTCA DO-160/EUROCAE ED-14：在本文档中是机载装置的环境条件和测试过程，当引用这个标准，标题简化为“DO-160”。

RTCA DO-254：机载电子硬件的设计保证指南（Design Assurance Guidance for Airborne Electronic Hardware）。

RTCA DO-178B：机载电子软件的设计保证指南（Design Assurance Guidance for Airborne Electronic Software）。

1.5 文档的优先性

本规范基于IEEE Std 802.3。本规范的理念是只在以下情况下定义对于IEEE 802.3的条款的改变，这些情况是：航空环境或使用者的意图与IEEE802.3的规定冲突；或者是必须通过对实现限定可用的选项以消除模糊的状况。本规范的内容被限定于描述这些变化和选项的限制，在本规范与可应用的ISO或IEEE标准发生冲突的时候，本规范应该具有优先性。

1.6 调整许可

本标准的实施将要遇到所有可应用的调整需求。迫切要求制造商得到对于这样的调整的所有可能的信息。这些信息不包含在本规范中，也不能从ARINC公司得到。

2 .0 概述

飞机数据网络已在本标准其它部分被描述，它被描述为应用IP寻址和相应的传输协议的IEEE 802.3以太网的一个定制版本。第7部分描述的是这种网络的一个子集，在这一部分中，服务质量（quality of service, QoS），其中包括及时地传送，是极为重要的。

AFDX网络是一种定制网络的一种特殊的情况。确定性网络可以与某个更宽范围的定制网络通信，并且可以推广到经过路由器或网关与某个兼容的网络通信。图2-1描绘了这种网络层次。对于更多的信息——关于如何使网络服务到网络协议映射的具体情况，参见附录C。

通常，控制系统，特别是航空电子系统，依赖于以实时的方式从源端到接收器传送完整的和最新的数据。对于安全关键性系统，“实时”通信链路是必需的。

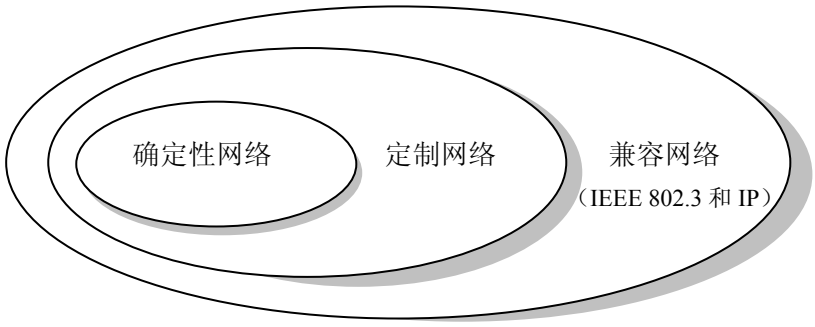


图 2-1 网络层次结构

2.1 比较模型

作为定时问题之间的比较，以太网与传统的飞机总线的比较是有帮助的。在下面的例子中，如图2-2所示，假设顺序传送的消息不发生出错/重传。

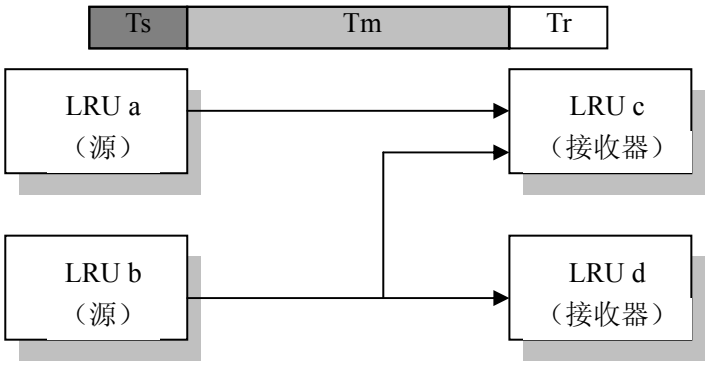


图 2-2 ARINC 429 总线

在这个例子中，参数如下：

Ts = 通过源端系统（end system, ES）的传送时延

Tm = 消息定时（长度÷带宽）

Tr = 通过接收器端系统的传送时延

以及总时延，L，即：

L=Ts+Tm+Tr

因为带宽是固定的，没有碰撞，并且端系统的时延是常量，从发送器通过网络到接收器传输一条消息所消耗的时间是常量。其他接收器端系统能够作为同一条消息的信宿，但不会影响这个定时情况，而第二个连接到LRU c的源将被一个独立的端系统有效地接收。

这种点到点系统几乎具有理想的确定性。传送一条消息的时间可以被计算出来，并且是常数。带宽的增加应该导致消息定时的减小。可靠性可以通过分析和现场的测试而被确定，并且可以采用冗余的系统。

因为每个端到端（end-to-end）链路与其他任意链路独立，没有冲突造成的延迟，在某个端系统中的故障将不影响与之通信的无故障的端系统。

2.2 交换式以太网网络

在多终端系统中，点到点的布线是主要的开销。以太网网络具有显著的优点，并且通过模仿点到点的连接，可以提供对于确定性网络的一个合适的模型。星形拓扑结构的交换式以太网的每个网段提供与图2-2所示的ARINC 429例子中同样的连接方式。

通过这样的节点，延迟时间不是仅被硬件延迟所固定；而是一个可变的量，称为抖动（jitter），它是由于与网络中其他的数据发生资源争用而产生的。通常根据累积的时延（包括硬件延迟和抖动效应）和链路带宽来对网络进行分析。以太网定时组件如图2-3所示。

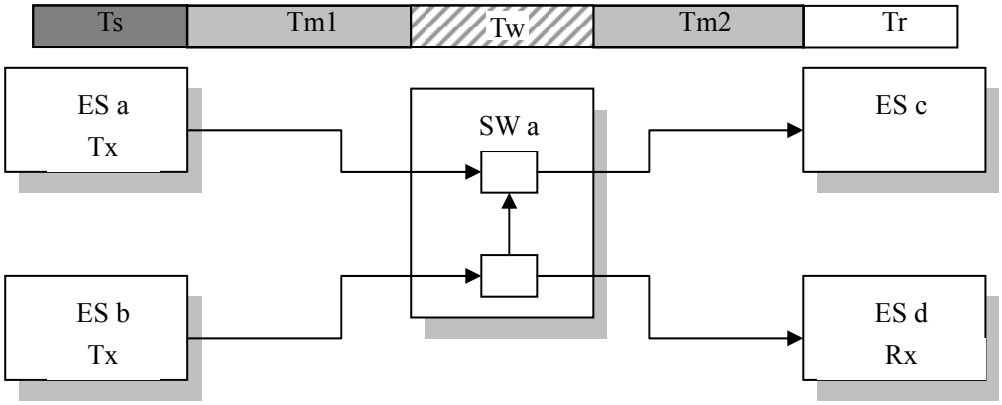


图 2-3 以太网定时

在这个简单的例子中，两个端系统发送器（ES a和ES b）向ES c提供数据，帧通过交换机SW a传送。在交换机上，同时接收从ES a和ES b到达的帧，顺序地将两种帧转发到ES c，最大抖动依赖于消息的长度：

Tj = (8×M)/Nbw+Tmin_gap （假设帧的长度相同）

式中：

Tj = 抖动时间

M = 以字节表示的帧的长度

Nbw = 介质带宽，以比特/秒（bit/s）为单位

Tmin_gap = 最小帧间间隔，以秒为单位

最前面的一个帧的时延:

La = Ts + Tm1 + Tsw + (8×M)/Nbw + Tm2 + Tr

式中:

Tsw = 交换机的硬件时延，以秒为单位

Tm*i* = 消息定时（长度/带宽）

被延迟的帧的时延：

Lb = La + Tj

这样，Tw是在交换机内的总时延，以秒为单位（Tsw + 输出缓冲时延），则Tw，对于某个输出端口，依赖于这个端口的流量。在异步网络中，对于任意特定的数据流，Tw将是时变的。

注释

交换机中的争用和存储转发能力导致对帧整个地进行缓存的需求。

这样，帧的延迟，(8×M)/Nbw，在Lb的表达式中出现了两次。

注释

这里只是简单提到抖动的概念。读者应该参考第3章和第4章对于AFDX网络中抖动的详细定义。

在正常的操作中，系统对两条数据链路（a-c）和(b-c)保持确定性，带有包含这种抖动的时延。在ES a或ES b发生一次故障的事件下，为了保持最大的系统完整性，服务质量（QoS）的定义可包括从非故障节点继续地传输数据。这意味着交换机具有过滤特性，这些特性在商用交换单元中是不常见的；这种故障隔离的方法能够确保在两个无故障的系统之间进行确定性的通信。

在一个典型的以太网网络中，分析中假设发送端系统是不同步的，并且帧传输符合某种随机到达分布。这意味着是一种泊松（Possion）发射分布，并且尽管带宽的平均值是已知的，在任意短的时间周期内，数据量的带宽是没有限制的。所以，通过网络的每条连接上的时延分布是一种没有受到限制的分布。这是网络中时延的概率描述的结果。此外，在典型的以太网中没有已知的方法限制来自任一个端系统的数据（在任意短的时间间隔内服从泊松分布）的实际的到达分布。所以，对端系统的带宽故障效应的限制是不可能的。

在AFDX网络中，分析中依然假设发送端系统是不同步的，但是帧的发送遵从一种有界的到达分布。这种有界的到达分布意味着确切的带宽规整（regulated）流量控制。所以，通过网络每条连接的时延分布是一个有界的分布。因此可得出一个可以计算出来的在网络中的最大时延，而不是一个概率性的时延。使用确切的带宽规整流量控制，使得在任意短的时间间隔内限制带宽利用率成为可能。所以，限制端系统的带宽故障效应是可能的。

为了继续模仿ARINC 429的工作情况，每个点到点布线的数据链路能够被确定性以太网网络中“虚拟链路”（Virtual Link, VL）所代替。每条VL的特性可以通过分析而确定，设定确切的带宽规整流量控制，如图2-4所示。这是通过对每条VL同时规定带宽和帧发送间隔的界限来实现的。

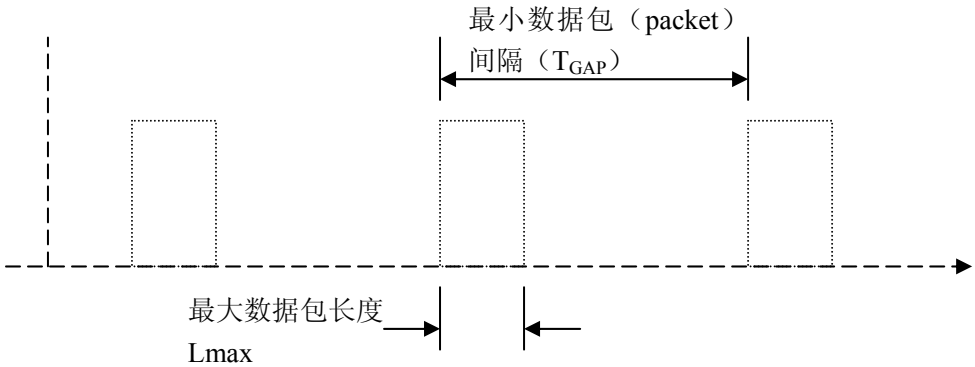


图 2-4 带宽调节流量控制

分配给这个VL的最大带宽是：$Lmax/T_{GAP}$。

在任意时间，通过加大的帧间隔或使用更小的帧来减少VL使用的带宽。然而，通过限定每条VL的特性，就能够分析整个网络的确定性属性。

2.3 可扩展性

网络拓扑结构的选择也将影响系统设计的可扩展性。对于AFDX，采用层次化的星形拓扑是由于这种形式易于扩展。

在任何实际的网络中，网络边界的扩充带来定时开销的增长，例如：一个单交换机的端口数目的增加。

2.4 次序完整性

在一般的定制网络中不提供保证措施以维持穿过网络的帧的次序完整性。

航空电子网络经常包含一些数据，它们的次序很重要。在帧之间存在某种关联时，应该保持次序完整性。出于这个原因，AFDX要求在给定的虚拟链路中保持次序完整性。

2.5 故障性能

在以太网网络拓扑中，出于对性能分析的考虑，不可能隔离地对待每条数据链路。需要某种机制以确保当一条数据链路出现故障时，余下的网络仍具有确定性。在AFDX的相关的功能块中，提供这种机制。

2.6 交换

在商用交换机中，通过采用“学习和老化”算法建立连接路径。为新的或是“老化”的数据源建立路径将导致不可预知的时延。

2.7 系统性能

只有给定网络拓扑结构，带有异步端点网络的确定性特性才能被评估。对于每条数据流，最大时延作为一种性能保证需求能够被推导出来。

3 .0 端系统规范

3.1 引言

端系统（End System, ES）的主要功能是提供一些服务，它们保证到某个分区（partition）软件的安全可靠的数据互换。

服务质量（Quality of Service, QoS）提供一种方法进行流量分类，并且确保特定类别的流量将总是以授权给它们的服务等级通过网络，而不用考虑争用的要求。

对于飞机网络（Aircraft Network），不需要区别对待几种类型或流量等级。每个网络传输请求必须得到服务，而不考虑数据类型；最大的网络传输延迟（也被称为“端到端时延”）必须得到保证。因此，飞机网络所需的仅有的服务等级就是保证服务。

保证服务提供一种稳固的、数学可证明的帧端到端传输延迟的上界。所以，对有界延迟的保证意味着在链路层次保证一定量的带宽。

这样，保证服务提供具有上界的延迟和固定的带宽，在一个发送节点与一个或多个接收节点之间得到一条逻辑上的开放性连接。属于同一个连接的帧定义为一条“流”（flow）。

概括起来，保证服务引起如下特性：

- 保证带宽和有界的时延
- 某条流量的特定延迟抖动（在同一条流量中任意两个帧之间的端到端传输延迟的变化量）不是固定的，这是因为它依赖于给定时间点的全局的网络流量。尽管如此，延迟抖动的界限能够通过数学计算得到。

图3-1给出了端系统通信栈的描述。

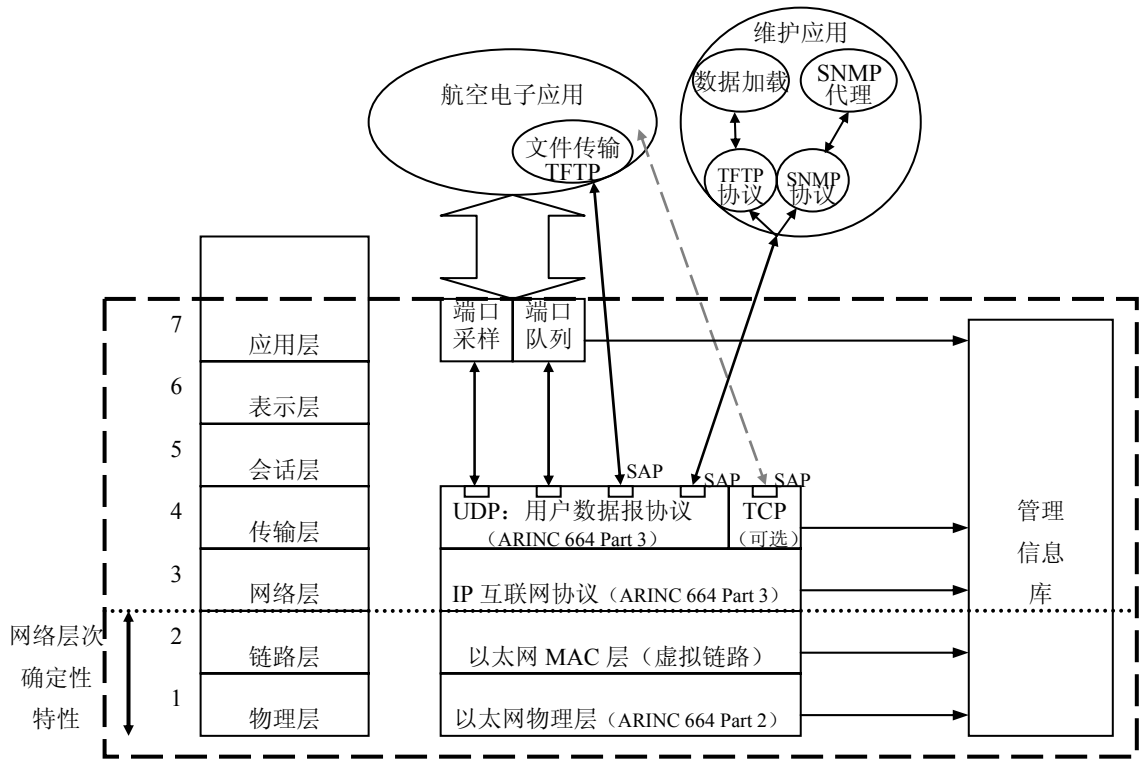


图 3-1 – 端系统协议层

3.1.1 ES 识别标记

使用最大正好是16个二进制位的编码用以识别ES，该编码仅对于系统集成者（system integration）是可见的。在附录B给出了一个使用12位编码作为ES识别标记的例子。

3.2 介质访问控制（MAC）层的互操作性与确定性

3.2.1 虚拟链路

“虚拟链路”（Virtual Link，VL）概念的描述如图3-2所示，它在本文档中被广泛地应用。

一个端系统可以被设计为仅接收VL而不发送VL，或者与之相反；这样，一个ES能够没有发起或接收的VL。通过VL进行端系统之间以太网帧的互换。在航空电子网络中任意一个VL都只有唯一的一个源端系统。

一条虚拟链路就是一个概念化的通信对象，具有如下的属性：

- 虚拟链路定义了一个逻辑上的单向连接，从一个源到一个或多个目的端系统，如图 3-2 所示。

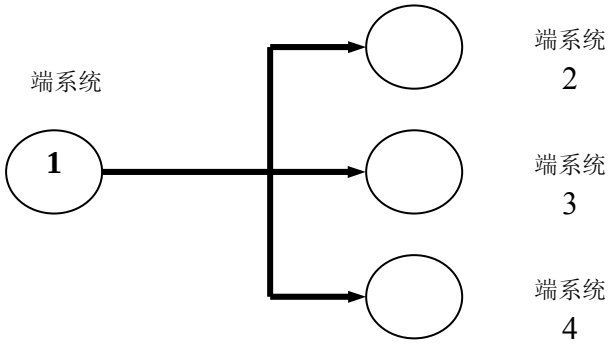


图 3-2 一个虚拟链路等于一个路径

- 每个虚链路都被指定一个最大带宽，该带宽由系统集成者分配。

在ES支持的虚拟链路中，ES应该利用可用带宽提供逻辑隔离。不论某个分区试图在一条VL上得到怎样的带宽利用率，其它任何的VL的可用带宽不受影响。

对每条虚拟链路，对于发送和接收（的次序完整性），端系统应该保持分区发出的数据的次序。

注释

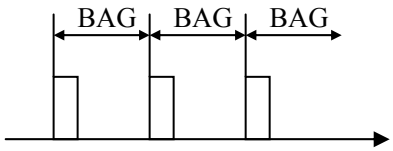
虚拟链路的处理是通过一种流量控制机制获得的，这种机制将属于这个ES的不同的数据源的数据流规进行规整，这种机制在网络层次提供的分区管理（partitioning）。

对于每条虚拟链路，不论其他虚拟链路是如何使用带宽的，端系统的通信协议栈应该保证它所分配的带宽，目的在于在网络层次上保持分区之间的隔离。一条虚拟链路不应被两个或两个以上的源分区所共享。

3.2.2 流/流量控制

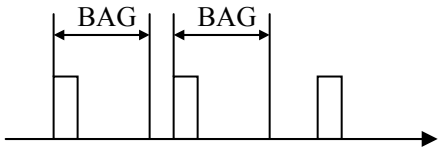
在每个端系统的输出端，与某条特定的虚拟链路相关联的帧的流量用两个参数来描述：带宽分配间隔（Bandwidth Allocation Gap，GAP）和抖动（Jitter）。

如果经过调度器的帧没有抖动（见3.2.3节：调度），BAG反映了在同一个VL中两个相邻的帧的起始二进制位之间的最小时间间隔，如图3-3和图3-4所示。



(jitter=0)

图 3-3 – VL 中最大带宽的数据流的 BAG



(jitter=0)

图 3-4 – VL 中非最大带宽的数据流的 BAG

注释

当一个VL就绪（eligible）但没有数据发送的时候，帧不会被发送。

为了保证每条VL的BAG，帧的流量被规整，如图3-5所示。

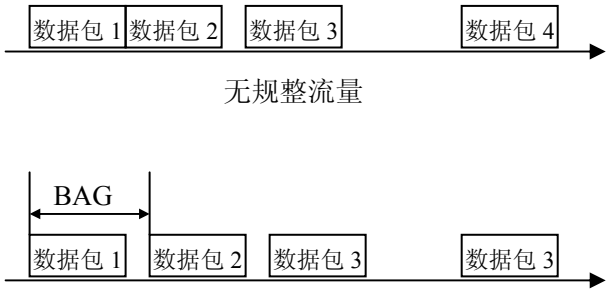


图 3-5 – 虚拟链路流的规整

3.2.3 调度

如果一个发送端系统有多条虚拟链路，调度器对来自于规整器的不同流量进行多路复用，如图3-6所示。

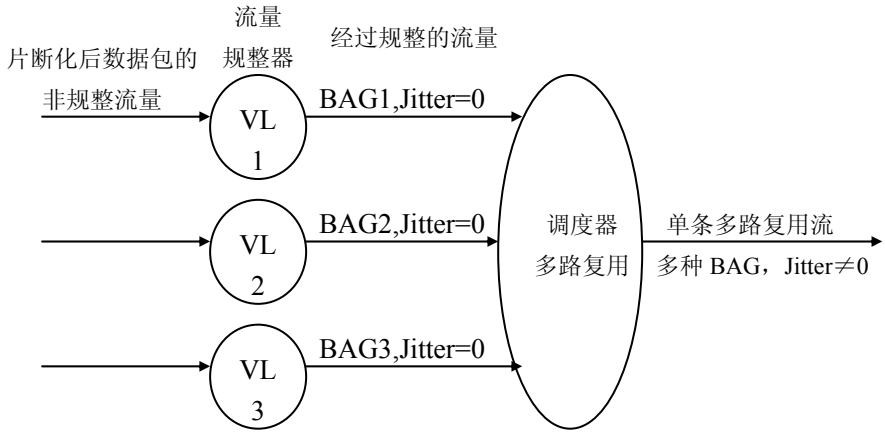


图 3-6 – 经过调度的流量控制机制的模型

在调度器的输出端，对于给定的某个虚拟链路，帧能够在某个有界的时间间隔中出现。这个时间间隔被定义为最大允许抖动（maximum admissable jitter）。该抖动是由调度算法引起的，而不是由流量本身造成的，如图3-7所示。

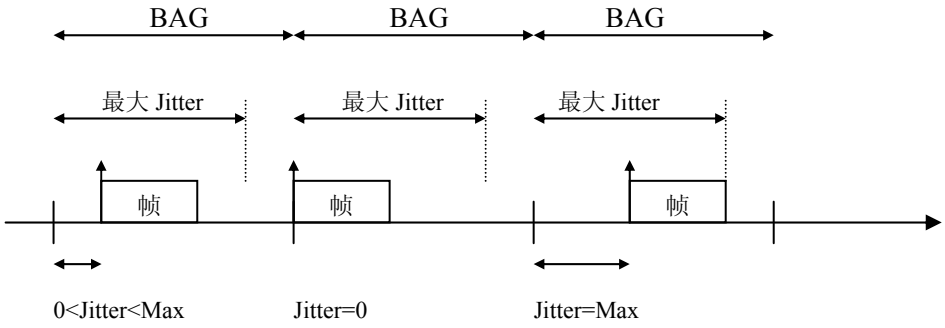


图 3-7 – 对于最大带宽的数据流的抖动效应

端系统应该以每个VL为基本单元对发送的数据进行规整，因为这种流量整形功能（流量特性的确切认识）是确定性分析的基础。

以每个VL为基本单元，流量规整器（流量整形功能）应该对流进行整形，使得在每个BAG间隔中（以毫秒为单位），发送的帧的数目不会多于一个。

注释

流量整形功能的目的是通过将帧分隔开，用以限制虚拟链路上瞬时的帧速率。规整器负责按照BAG控制分给虚拟链路的带宽。

每个VL的最大可使用带宽由它的BAG和被允许的Lmax（最大VL帧大小）所决定。最大可使用带宽=Lmax/BAG，以Kbytes每秒为单位。

端系统在发送和接收端都应该能够容纳直到1518字节的VL帧。

对于每个VL，端系统应该通过端系统配置表获得一个BAG值。

注释

使用一个配置表文件配置端系统。这种配置表的详细的内容超出了本文档的范围。

ES的流量整形功能应该能够在1ms到128ms的范围内控制BAG的值。这些值应该满足如下的公式：$BAG=2^k$（单位：ms），(k的取值范围是0到7)。

注释

如果一个分区以低于128ms的频度发出数据，就应该用128作为BAG的值。为了简化ES的设计，BAG的值被限于2的幂次。

3.2.4 端系统性能

系统集成者的主要目标是能够以一种确定性的方式使用一个AFDX端系统。通过创造一种ES性能的度量方法，AFDX减轻了系统集成者对于认证工作的负担，它为系统集成者提供一套灵活的解决方案和明确定义的约束条件。

3.2.4.1 时延

发送时延被定义为下列两个测量点之间的持续时间，如图3-8所示。

- 开始点 – 主机分区数据的最后的一个 bit 对于端系统的通信服务可用；
- 结束点 – 相应的以太网帧的最后一个 bit 被发送到物理介质上。

对于技术时延（technological latency）的测量在缓冲存储器为空的时候进行，没有冲突的接入源，也没有IP片断化，如图3-8所示。

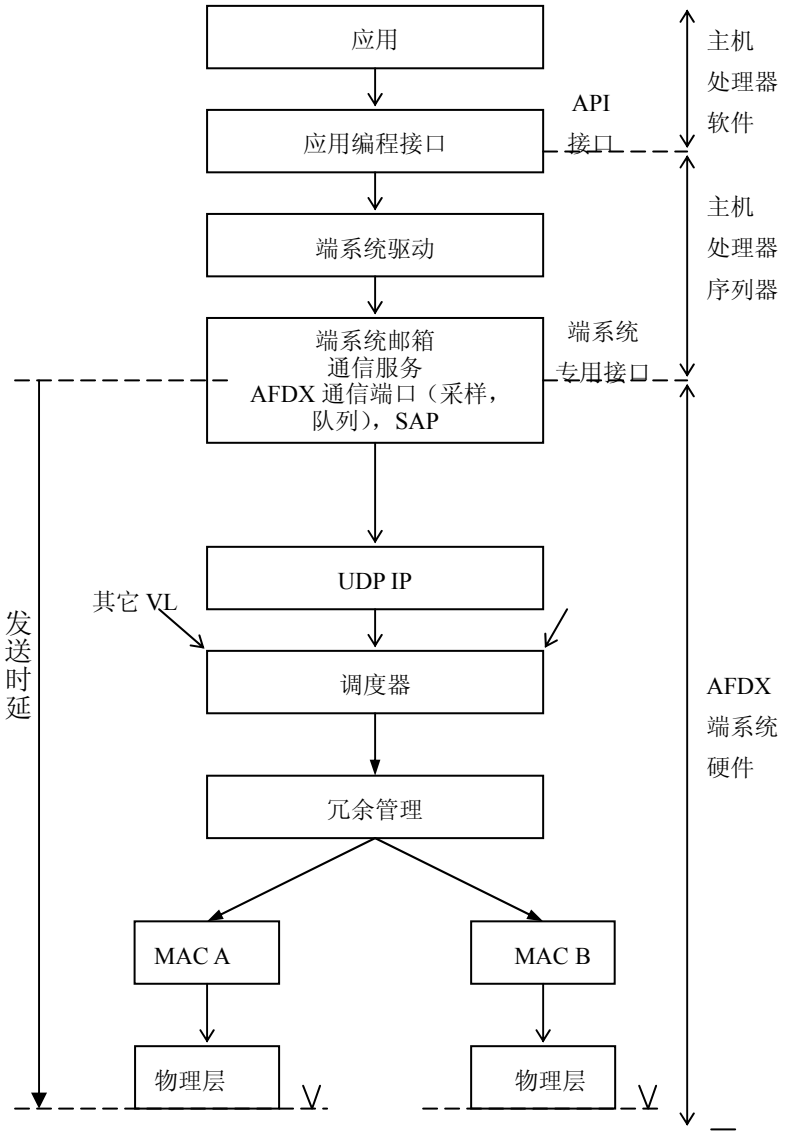


图 3-8 – Tx 性能测量点

端系统发送时的技术时延应该有界，它要小于150 μs + 帧延迟（frame delay）。

注释

设定ES的总时延包含技术时延（独立于流量负载）和配置时延（依赖于配置和流量负载）。

技术时延被定义为在没有其他任务处理的时候，端系统接受和处理应用数据，并开始发送所需的时间。

被加入的“帧延迟”包含将帧送达到物理层花费的时间。

接收时延被定义为下列两个测量点之间的持续时间，如图3-9所示。

- 起始点 – 以太网帧的最后一个 bit 在物理介质连接装置（attachment）上被接收到；
- 结束点 – 相关的数据得最后一个 bit 对于端系统主机分区可用。

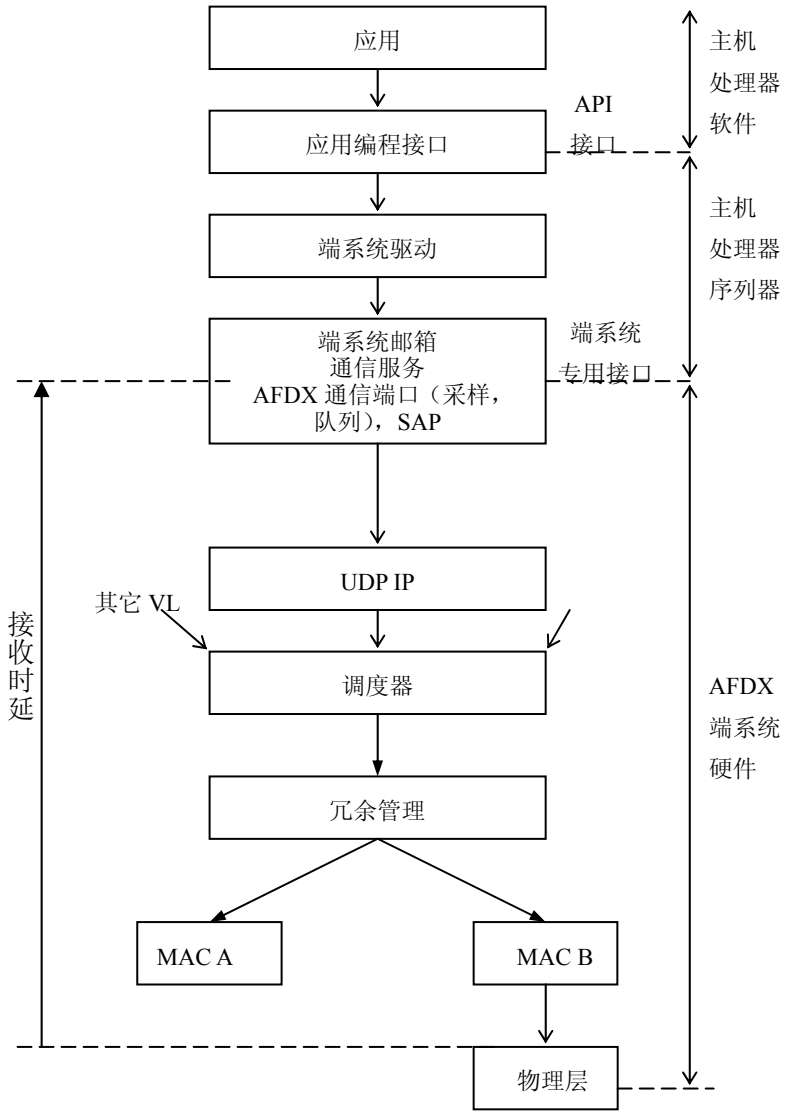


图 3-9 – Rx 性能测量点

端系统接收时的技术时延应该有界并小于150 μs。

3.2.4.2 MAC 约束

为避免在流量突发中丢失到达的帧，并且能够在传输中安插IFG（Inter-frame gap，帧间间隔），端系统的MAC层应该能够 ：

- 以介质中帧的全速率处理接收的帧；并且以介质中帧的全速率使合适的（被选中的）可用帧。
- 一帧紧接着一帧地传输帧。

最短的帧对应于每个物理连接装置（attachment）的最大的帧速率，即：

64字节（帧）+12字节（IFG）+7字节（前导字）+ 1字节（SFD，起始定界符）=84字节以100Mbit/s的速率传输。

等价于每帧6.72 μs的持续时间（大约每秒148800帧）。

注释

对于发送这个需求应能有所放松。然而，设计者应该非常仔细地考虑发送中对于最大抖动的兼容性的影响。

根据处理带有最小帧间间隔（12字节）的最短帧（64字节）的处理能力，这个需求更加苛刻。

3.2.4.3 抖动

在发送中，端系统输出端口的每个VL的最大允许抖动应该服从下列两个联立的公式：

$$\begin{cases} \max\_jitter \leq 40\mu s + \dfrac{\sum_{i\in\{\text{VL的集合}\}} (20 \text{ bytes} + L_{\max,i} \text{ bytes}) \times 8 \text{ bits/bytes}}{Nbw \text{ bits/s}} \\ \max\_jitter \leq 500\mu s \end{cases}$$

注意：max_jitter以微秒为单位（μs）；Nbw是介质带宽，以bit/s为单位，Lmax以字节为单位；40μs是典型的最小固定技术时延抖动。

根据该公式，端系统若具有较少的VL并且其中待处理的帧是短帧，则最大允许的抖动将较低。在所有的情况下，抖动被限制在500μs的界限内，以限制对整个网络确定性的影响。

注释

对于重负载的ES（发送中），发送中的优化调度可能要应对第二个公式。系统集成者有责任确定，对于选定的端系统配置和实现，500μs的界限不被超过。

这些值是展示AFDX确定性的基础，可以被用于评价端系统的性能限制。由于有限的处理能力，一个非优化的ES（考虑ES）将具有带宽限制。

为了对发送中端系统的允许时延进行数学化处理，两种限制的情况被定义。

对于第一种情况，设定主机分区在一个给定的VL上传输，具有均匀间隔的数据（没有突发），并且没有数据需要被片断化。在这条虚拟链路上端系统没有其它数据需要处理，对于这个给定的VLi，的总的允许延迟为：

$MAX\_Latency_i \leq BAG_i + Max\_jitter + Technological\_Latency\_in\_transmission$

当主机分区发送突发数据或者长消息需要被分片处理（fragmentation）的时候采用第二种情况。在这种情况下，如果端系统在这条虚拟链路上有其他数据需要处理，要传输的缓存数据将被延迟。对于这个给定的发送VLi，如果第(p-1)帧正在处理，对于编号为p的帧的最大延迟应该被下面的公式所限定。

$MAX\_Latency_i\ (frame\_p) \leq p*BAG_i + Max\_jitter + Technological\_Latency\_in\_transmission$

根据VL的带宽限制（流量整形）传输将被延迟。

注释

考虑配置时延，一些实现会得到优化的解决方案。在所有情况下，上面提到的值应该被遵守。

3.2.5 MAC 寻址

3.2.5.1 MAC 目的地址

虚拟链路应该仅被MAC目的地址识别，如图3-10，并且AFDX帧的MAC源地址应该是单播地址，用来识别物理的以太网接口。

在AFDX帧中的MAC目的地址应该是组（Group）地址和本地管理（Locally Administered）地址，并且应该与下面的格式兼容。

48 bit	
固定域 32 bit	虚拟链路标识符 16 bit
xxxx xx11 xxxx xxxx xxxx xxxx xxxx xxxx	

图 3-10 – MAC 多播地址格式

每个 ES 应该从系统的集成者得到“固定域”和“虚拟链路标识符”（Virtual Link Identifier）的值。这些取值不在 ARINC 664 中规定。

在任意给定的AFDX网络中，每个ES的MAC地址固定域应该是相同的。第一个字节的最低二进制位表示组地址（总是为 1）。

为了使用标准的以太网帧，应该用 MAC 组地址从端系统到（多个）端系统发送帧。

第一个字节的次低二进制位表明这是本地管理地址（总是为 1）。

注释

尽管在 MAC 层，只有组地址可以使用，单播通信能够被看作在 IP 层采用 IP 单播目的地址。

3.2.5.2 MAC 源地址

MAC源地址应该是与IEEE 802.3兼容的独立的和本地管理的地址。地址的结构由下面的段落具体说明。

以太网 MAC 控制器标识符 （48 bit）			
固定域 24 bit	User_Defined_ID 16 bit	Interface_ID 3 bit	固定域 5 bit
0000 0010 0000 0000 0000 0000	nnnn nnnn nnnn nnnn	mmm	0 0000

图 3-11- MAC 源地址格式

注释

协议没有推荐特定的源 MAC 地址构造算法。所以，对于 AFDX 端系统来说，有必要具备一种方法，用以确定它们所在的网络中所使用的地址构造算法。例如，管脚编程可以被用来作为一种方式来指明采用何种地址构造规则。

如图3-11所示，固定域被设置为“0000 0010 0000 0000 0000 0000”。

第一个字节的最低二进制位表明这是个体地址（等于0）。

第一个字节的次低二进制位表明这是局部管理地址（等于1）。

User_Defined_ID（用户定义标识符）是一个单独的16-bit域。系统集成者应该合理地使用它，用以为每个在网络上IP可寻址的主机给定一个独一无二的并且有含义的IP地址。

Interface_ID（接口标识符），如图3-12定义，表明以太网MAC控制器连接到哪个AFDX的冗余网络。

Interface_ID	含义
0 0 0	未用
0 0 1	以太网 MAC 控制器连接到网络 A
0 1 0	以太网 MAC 控制器连接到网络 B
0 1 1	未用
1 0 0	未用
1 0 1	未用
1 1 0	未用
1 1 1	未用

图 3-12 – Interface_ID （接口标识符）定义

3.2.6 冗余概念

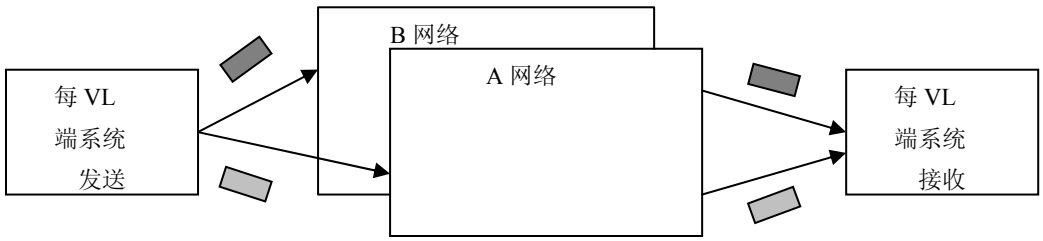


图 3-13 – 网络冗余概念

端系统之间的通信覆盖在多个独立且冗余的网络，这样，对于任何的（单件的）网络组件的失效，例如：一段链路或一台交换机的失效，数据流可以得到保护。这样作的效果是保护两个端系统之间的通信，抵御全网络范围内的组件失效。

图3-13展示了网络冗余的基本概念。冗余的方案是以每条虚拟链路为基础的。一个发送端系统和一个接受端系统通过一个特定的虚拟链路以如下的方式通信：

一个使用发送端系统的分区准备好一些数据并使它通过通信协议栈。这时一个顺序号（Sequence Number）域被加入到每个帧，并且该顺序号在前后相继的两个帧中是递增的。顺序号的加入使接收功能部件能够在将帧传送到接收分区之前重新构造出一条单一的、有次序的帧流，而不包含冗余复制的帧。在这种方法中，分区没有意识到下层的网络冗余，并且能够在通信栈和利用网络服务的分区之间建立一个简单的接口。

在默认的模式中，每个帧被发出并且同时通过两个网络。在接收的时候，在通信栈（低于IP层）中的算法采用“先到有效者胜出”的策略。这意味着从其中一个网络中得到带有下一个有效顺序号的第一个帧将被接受，并向上通过通信栈到达接收分区。但带有这个顺序号的第二个帧被收到时，它将被简单地丢弃。

如图3-14给出的帧流，RM（Redundancy Management，冗余管理）被置于IC（Integrity Checking，完整性检查）之后。在网络操作没有故障的情况下，IC仅仅是将它从网络中收到的帧传递到RM。AFDX冗余管理的功能仅仅是去除那些冗余复制的帧。

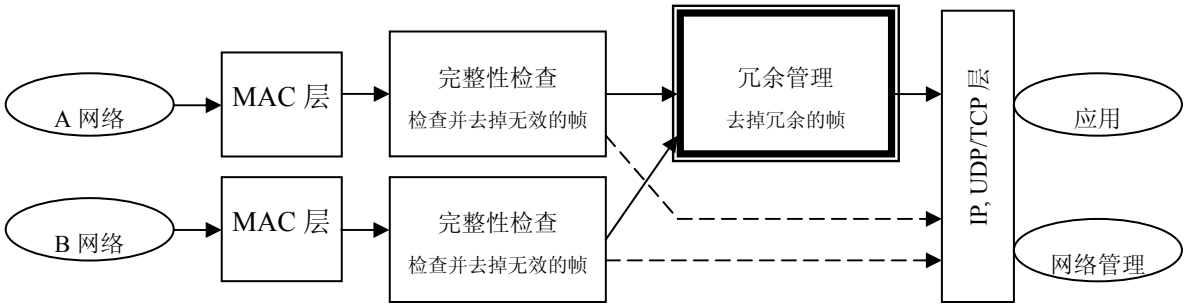


图 3-14 – 端系统地完整性检查和冗余管理

期望的行为如图3-15到图3-18所示。其中“RMA”行是指由冗余管理算法（Redundancy Management Algorithm, RMA）发送到分区的帧。

例1：非正常发送帧

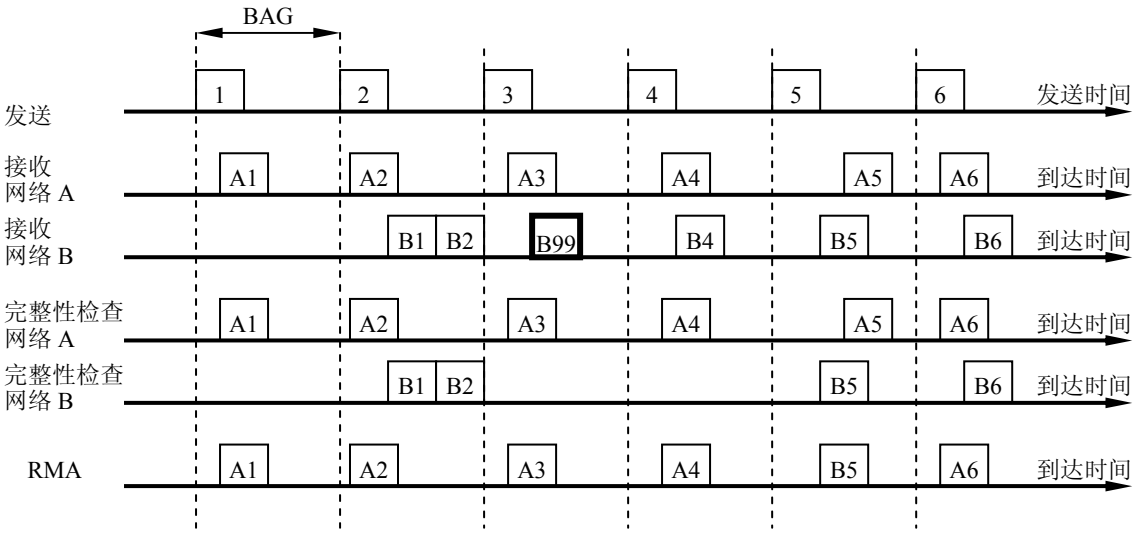


图 3-15 – 网络 B 发送一个非正常帧

冗余管理结果：非正常帧不被转发到分区。

例2：一个帧丢失

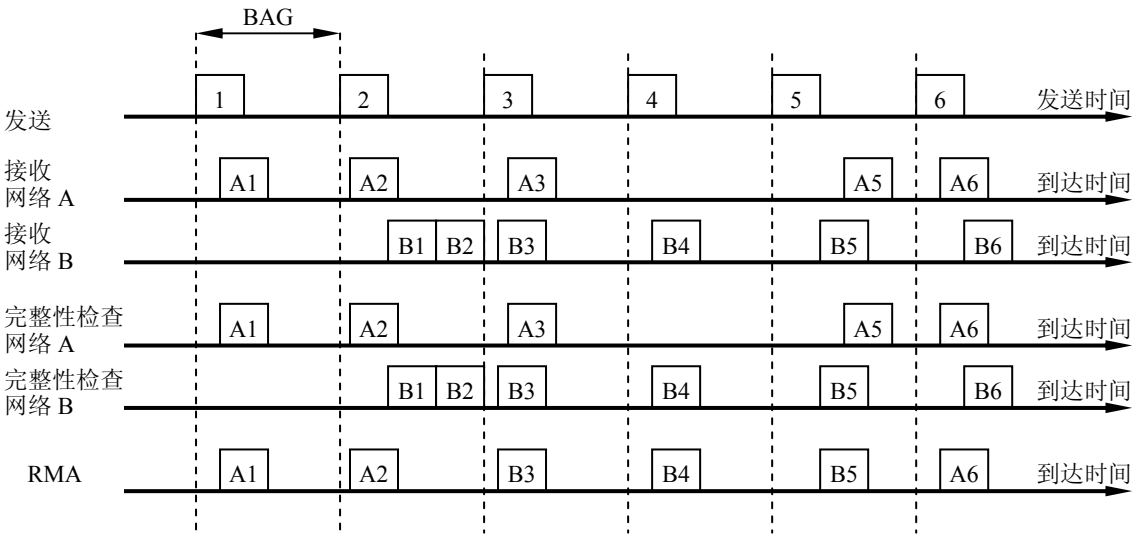
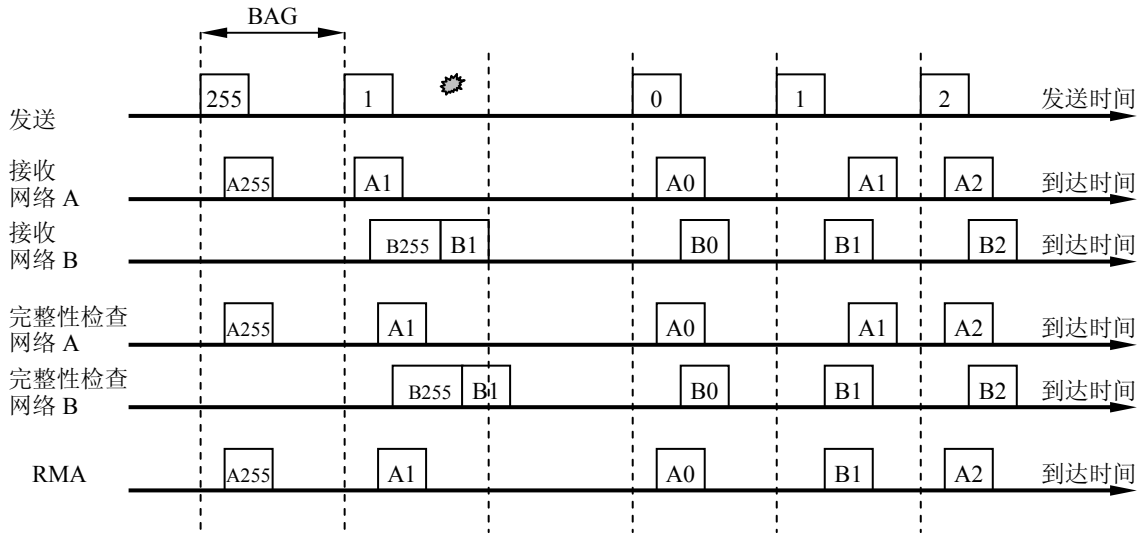


图 3-16 –网络 A 中一个帧丢失

因为一次比特错误（Bit Error），帧“A4”丢失。
冗余管理结果：在网络B中到达的帧被接收。

例3：发送器复位



：发送装置的复位

图 3-17 – 发送端系统复位

没有帧丢失。

例4：无意义交换（Babbling Switch）（卡塞帧）

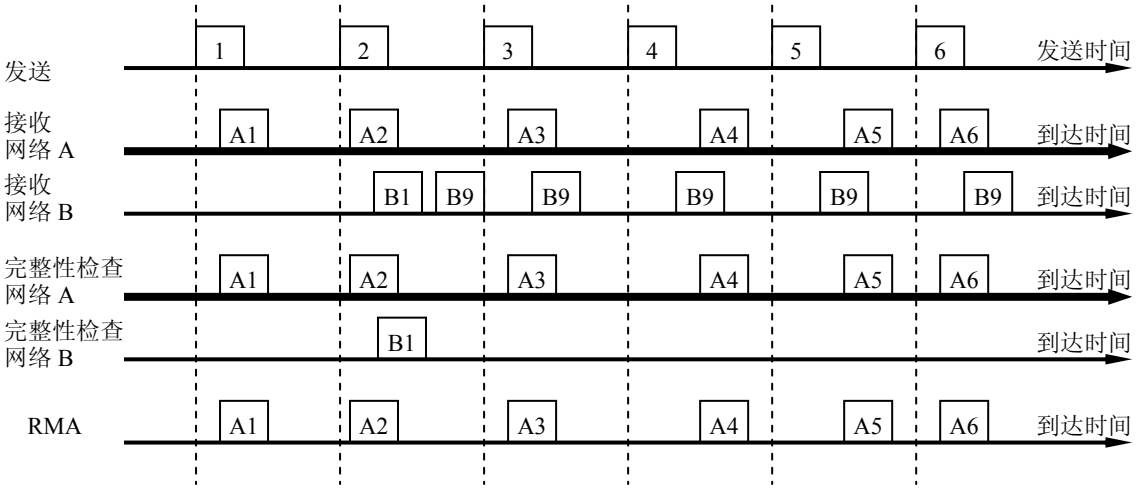


图 3-18 – 网络 B 的无意义帧

在例4中，冗余管理结果是：因为完整性检查，帧没有被转发到IP层。

3.2.6.1 顺序号与发送端系统

在AFDX网络中，对于每条VL，端系统应该对每个发送帧加入一个序列号。

帧顺序号的长度应该是8位比特，取值范围是0到255。

注释

顺序号的取值范围应该足够大，以满足在正常的操作下检测冗余帧的要求，但还要紧凑短小。例如：在最坏情况下BAG=1ms，SkewMax（最大偏斜时间）=5ms，在两个接收帧之间最大的SN（顺序号）偏移是：

$$Int\left[\frac{SkewMax}{BAG}\right]+2=7$$ ， 这个值远低于需要考虑计数回卷的128。

对于每条VL，顺序号的初始值应该被设置为0。顺序号总是在发送ES复位后被赋以这个初始值。

对于同一条VL上的每个相继传输的帧，帧顺序号应该递增加一，并且在值递增到255之后回卷到1。

注释

递增加１使检测丢失的帧成为可能。回卷到1决定顺序号的最大范围，而作为复位条件，保留SN=0。这样，使整性检查（见3.2.6.2节）得到完善。

帧顺序号应该作为MAC帧的有效载荷的一部分被放置在MAC帧的CRC域之前，如图3-19所示。

为了简化接收端系统的算法，帧的冗余副本应该被几乎同时地传输，它们之间的最大时间差不应超过0.5ms。

注释

对于每条VL，系统集成者可以自主决定是否采用冗余机制。如果对任意的VL，冗余设置被关闭，应该仔细地评估这种选择对系统的完整性和可用性的影响。

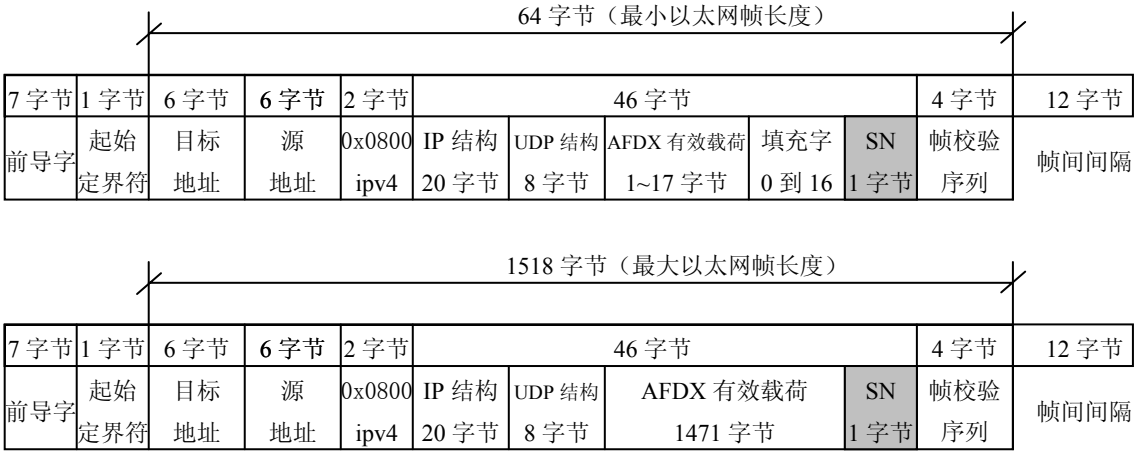


图 3-19 – 在最小和最大长度的帧中顺序号的位置

3.2.6.2 顺序标号与接收端系统

3.2.6.2.1 完整性检查

在没有故障的网络操作下，完整性检查仅简单地将它接收的帧传递给冗余管理过程，对于每个冗余网络是独立的。如果（基于顺序号）发现故障，完整性检查执行去除无效帧的任务，并且在执行此操作的时候通知网络管理机构。参照3.2.6节“冗余概念”中关于顺序号的用法。

在每个（冗余）网络，完整性检查功能对顺序号处于如下区间中的每个帧进行检查，该区间是：

$[PSN"+"1, PSN"+"2]$；

其中，“前序列号”（Previous Sequence Number, PSN）是这个VL接收到（但不一定被转发）的前一个帧的顺序号。

运算符“+”将顺序号的回卷考虑在内。所以，例如：如果PSN=254，则PSN”+”1=255，并且PSN”+”2=1。

注释

这个功能增加了完整性的强健性，例如：通过消除了卡塞帧或单个非正常帧，并减少无意义交换的影响。丢失单个的帧被认为是正常的事件，因为位错误概率（Bit Error Rate）不为0。

完整性检查也应该在下面的特殊情况下将如下的帧作为有效帧接收：

- 接收顺序号（Received Sequence Number, RSN）等于 0；
- 在任何的接收 ES 复位之后，第一次被收到的帧。

没有满足这些规则的帧被丢弃。

注释

这些特殊的情况改进了周期数据的完整性。否则，在发送ES和接收ES复位之后，会发生系统性的帧丢失。

只有在发送装置的一次复位之后，才发送顺序号为0的帧。

可通过更改配置表关闭两个（冗余）网络中VL的完整性检查功能。停用完整性检查允许接收机接收从网络A和网络B到达的所有帧。

3.2.6.2.2 冗余管理

冗余管理设定网络正处于适当的工作状况，特别是设定网络的确定性属性已被验证。

RM的配置一般基于SkewMax参数，即：在收到的两个（互为）冗余的帧之间的最大时间。这个值依赖于网络拓扑（帧跨越交换机的数目），并且应由系统集成者提供。对于每条VL，配置SkewMax的值（以ms为单位）。

定义：

- 冗余 VL 意味着同样的帧通过两个网络 A 和网络 B 发送；
- 非冗余 VL 意味着（可能是不同的）帧通过网络 A 或 B 发送。

基于每条VL，ES应该能够接收：

- 一条冗余的 VL 并将冗余数据之一传送到分区（RM 功能被激活）；
- 一条冗余的 VL，传送所有两份冗余数据（RM 功能未被激活）；
- 来自任一个接口的一条非冗余 VL，从它传输数据到分区（在这种情况，RM 的功能可以是被激活的，也可以不是）。

当冗余管理功能是被激活的，应该传送先到的冗余帧。

在通信中涉及到的任何装置的复位（传输端系统，接收端系统或AFDX交换机）应该不影响这种属性。“先到者胜出”的基本原则在发生单次的AFDX交换丢失的情况下使网络具有可用性。

注释

硬件复位时间被设为大于SkewMax。这保护RM算法避免（被如下情形）打乱：在复位之前帧被ES发送到网络，而在这次复位之后发送的帧到达ES。

冗余管理算法应只使用一个帧的RSN作为拒绝或接收的准则。完整性检查是一个与之分离的任务，即使没用到冗余它也被执行。

对于在接收器中的每个VL，冗余管理（RM）功能应该保证帧以递增的RSN次序转发。这还将应用于复位和偶然丢失帧的情况。如果在接到一个帧后超过最大偏斜（skew max）时间，则无论下一帧的顺序号是什么，它都将被接收。

冗余管理功能必须仅按照顺序对帧进行转发，但是不需要重新排序。所以，在一些情况下，其中一个网络上的某个帧的丢失也能够导致它的拷贝的丢失。

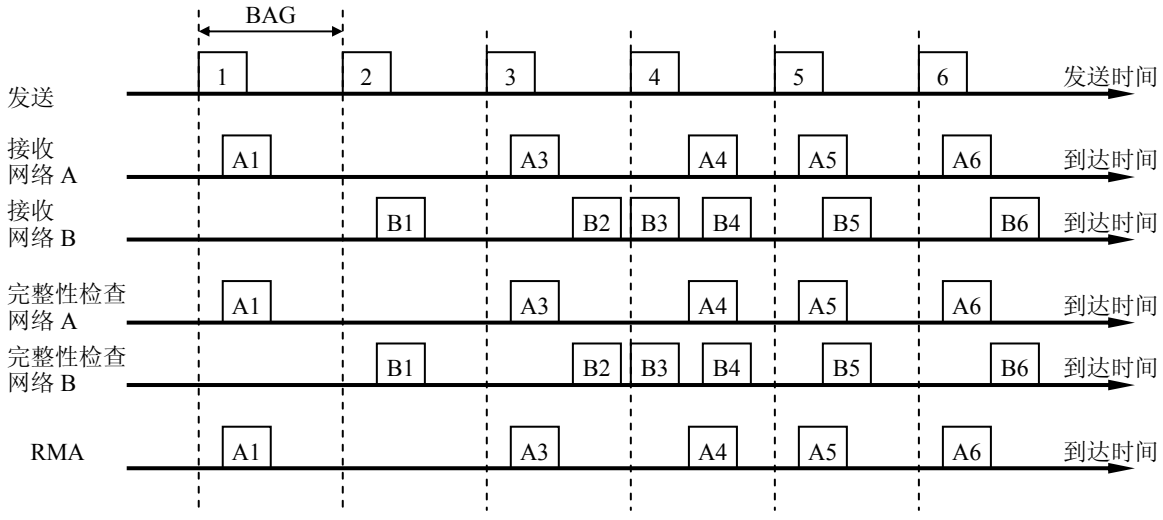


图 3-20 – 一个帧的丢失

例如，在图3-20中，帧“A2”在网络A中丢失，并且在这个网络中，帧“A3”在丢失帧的拷贝“B2”（在网络B）之前到达。在这种情况下，拷贝B2将不被转发到分区，尽管它作为第一个被收到的SN=2的帧。

3.3 IP 层和 IP 层以上的互操作性

注释

在分区端口（应用端口）和ES的端口之间没有标准的映射关系。这些需求将被写入特定装置的规范说明中。

尽管如此，AFDX通信端口（Comport）或服务访问点（Service Access Point, SAP）端口有两个数据消费者可能导致数据的丢失[1]，如图3-21。

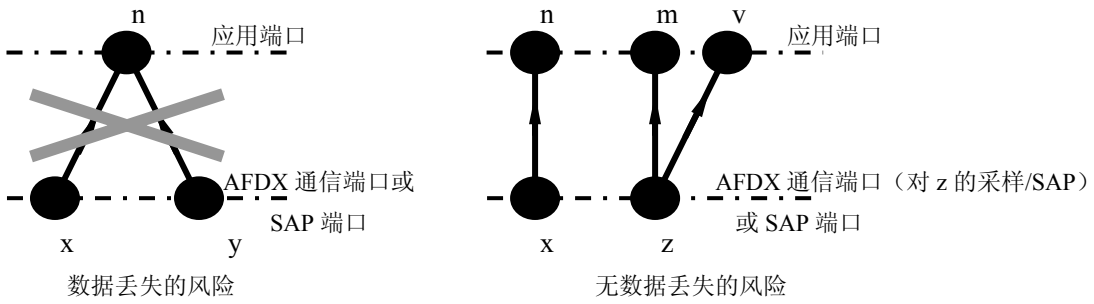


图 3-21 – 接收中共享 AFDX 通信端口

对应一个发送中的AFDX通信端口或SAP端口，存在两个数据源也可能导致数据的丢失，如图3-22所示。

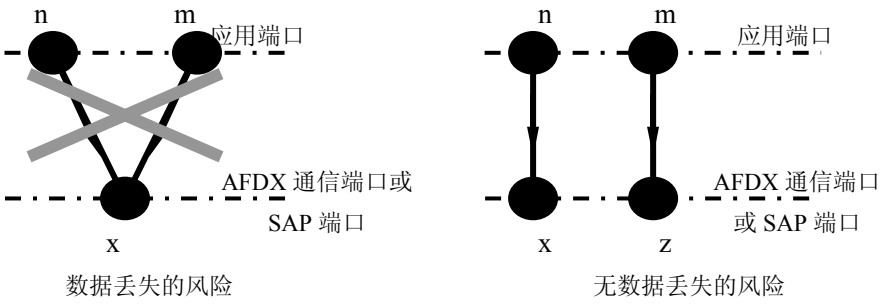


图 3-22 – 发送，端口无共享

用户端口可以被配置为发送或接收，但不能既是发送又是接收。

不能采用图3-21和3-22中打叉的例子。

3.3.1 航空电子服务

从航空电子分区的角度，ES通过两种类型的端口提供不同的数据传输模式：

[1] 此处的描述似与图 3-21 中“数据丢失的风险”的情况矛盾——译者注。

1. 通信端口：采样或队列模式（参见：ARINC 653）；
2. SAP端口：用来进行TFTP（简单文件传输协议）的传输，以及与兼容网络通信。

图3-23描述了具有两个分区（参见：ARINC 653对于分区的定义）和一个端系统的装置。每个分区具有一个IP地址。为了与分区通信，端系统使用两种端口类型：通信端口和SAP。

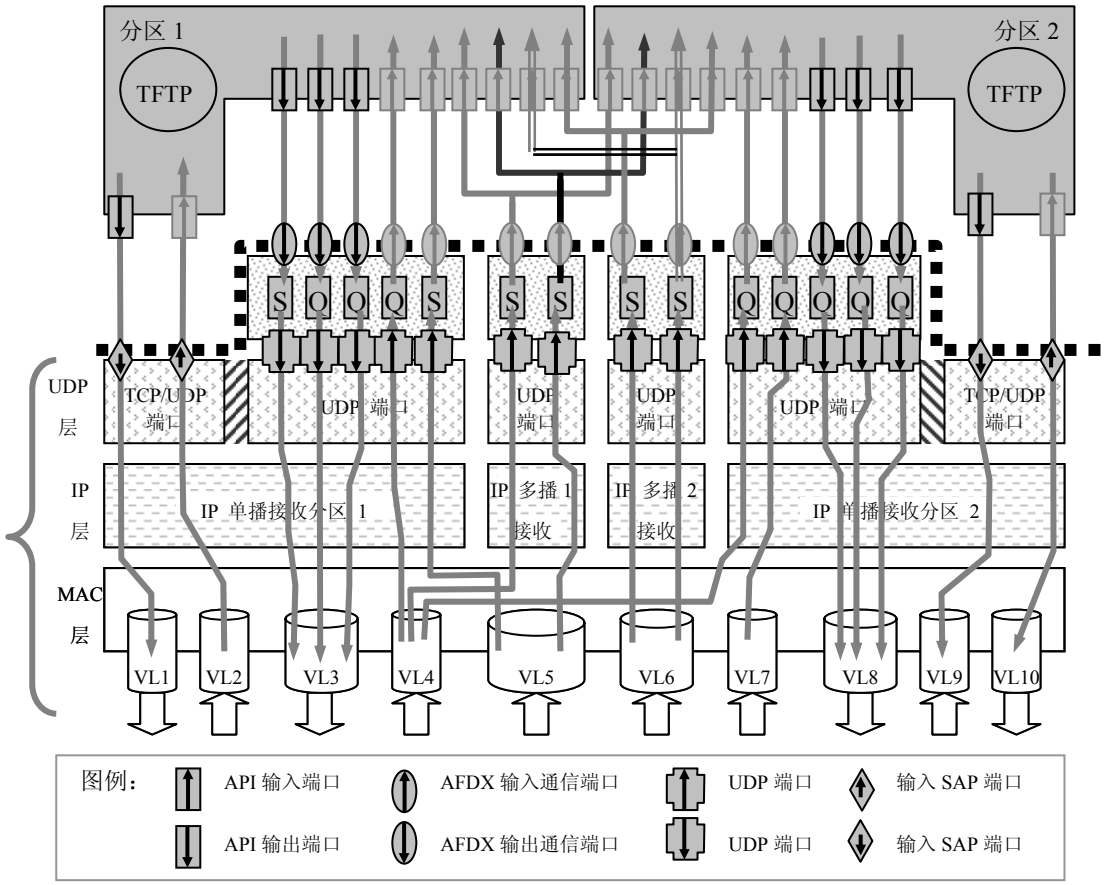


图 3-23 –分区和端系统之间的接口

3.3.1.1 通信端口

ES通过通信端口提供两种类型的服务：采样（sampling）和队列（queuing）。由于UDP相对效率较高，这两种服务均采用UDP通信。

3.3.1.1.1 航空电子采样服务

ES应按ARINC 653的要求提供采样服务（见ARINC 653的2.3.5.6.1节）。

3.3.1.1.1.1 发送

采样服务不能使用 IP 分片操作（IP fragment），这样每条采样消息的长度都应当小于或等于所对应的 VL 的有效载荷的要求。

采样服务应当基于多播，并且是从一个源到一个或多个目的的单向通信。

采样服务是简单、无连接、无应答的。它没有在传输数据中加入错误控制，并且除了 VL 流量控制之外无须额外的流量控制。这种传输方式类似于传统 ARINC 429 链路所提供的服务方式。

3.3.1.1.1.2 接收

最新的一条消息存储在一个特定的采样端口中，该消息应当能够被几个分区读取。（即：几个应用程序可能会预订该采样端口）。

每个采样端口都应当对应一个刷新（freshness）指示。该刷新指示对于每个读取该消息的分区都应是可见的。

3.3.1.1.2 航空电子队列服务

端系统应该按照ARINC 653（见ARINC 653的2.3.5.6.2节）的定义为航空电子分区提供队列服务。

队列服务是简单、无连接、无应答的。

队列服务应该能够管理同一个队列通信端口中不同长度的消息。

为了保证数据的顺序，队列服务应该在发送和接收时以先入先出（FIFO）原则管理消息。

每一个队列服务的实例应该能够管理长达8k字节的应用数据（这时需要IP分片操作）。

注释

无应答无连接队列服务之所以被多种通信任务所接受，在于在冗余的AFDX网络中帧丢失的概率低。

3.3.1.1.2.1 发送

采用分片操作的的时候，各片断应该按照次序被发送到AFDX网络中。

如果在发送过程中发生缓存溢出的情况，ES应当给源分区一个错误指示，并且该帧将被丢弃，如图3-24所示。

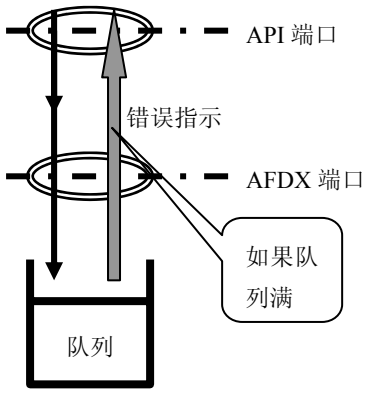


图 3-24 – 发送缓冲区溢出的错误指示

3.3.1.1.2.2 接收

在存在分片操作的情况下，直到整个消息被重新组装完毕，数据才能出现于分区的队列服务FIFO中。

如果在接收时发生缓存溢出，应该发送一条错误（提示）消息到接收分区，这时（到达的）帧将被丢弃。

3.3.1.2 SAP 端口

3.3.1.2.1 为兼容网络提供服务

端系统能够作为一个服务访问点（Service Access Point, SAP），并具有如下的特性：

- SAP 端口能够被用来在 AFDX 网络内通信；
- 作为 ES 设计的一部分，通过网关或路由器接入兼容网络；

- ES 应当提供可与兼容网络通信的 UDP 服务；
- 每个 UDP 服务访问点的实例都应该能够处理 8k 字节的数据报；
- 作为一种可选项，通过被合理配置的 SAP 端口，能够使用 TCP 直接接入 IP 层。

为了与兼容网络通信，一个发送 ES 要具有指定目的地址的能力，即：IP 地址和端口号。出于这个目的，当 ES 接到来自兼容网络的请求的时候，要求这些地址是可用的。

3.3.1.2.2 SAP 端口错误管理

接收器监视着SAP端口服务质量的降级。如果接收器出现缓冲区溢出，应该发出一条错误（提示）消息到接收分区，该（到达的）帧将被丢弃。

3.3.1.2.3 文件传输服务

应该使用简单文件传输协议（Trivial File Transfer Protocol, TFTP）传输文件。

TFTP的规范在许多RFC中定义，如表3-1所示：

表 3-1 – RFC 对于 TFTP 的定义

RFC 号	标题	种类
783	TFTP 协议（版本 2）	标准，更新为 RFC 1350
1123	Internet 主机应用与支持的需求	标准
1350	TFTP 协议（版本 2）	标准
2347	TFTP 可选扩展	标准们的发展路线，更新为 RFC 1350
2348	TFTP 块大小的选项	标准们的发展路线，更新为 RFC 1350
2349	TFTP 超时和传输块大小的选项	标准们的发展路线，更新为 RFC 1350
1785	TFTP 选项协商分析	信息，更新为 RFC 1350

文件传输服务的每个实例能够管理长达8k字节的数据块。

3.3.1.3 子虚拟链路

一个VL能够由多个子虚拟链路（Sub-VL）构成，如图3-25到图3-27所示；在这种情况下，VL仅由这些Sub-VL组成。

每个Sub-VL具有一个专用的FIFO（先入先出队列），并且这些Sub-VL FIFO被主FIFO队列（VL的FIFO队列）基于轮询（round-robin）的方式读出。轮询功能是基于MAC帧的，这样IP分片操作（如果有的话）在加载Sub-VL的FIFO前就应该已经被执行。

注释

Sub-VL的实现是可选择的特性，对网络的确定性没有影响。它可以被用于优化VL的带宽利用率。

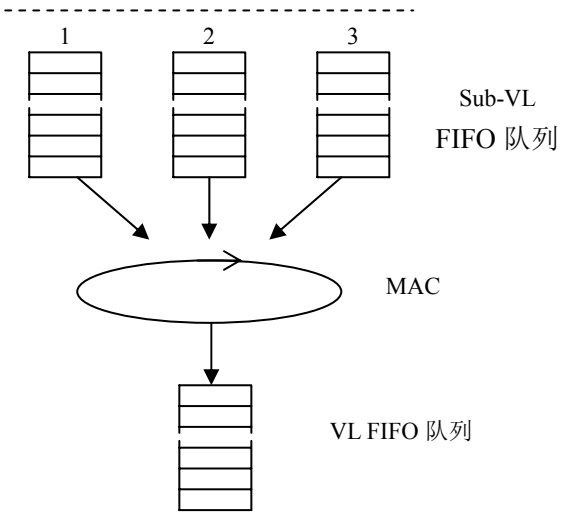


图 3-25 – Sub-VL 的 FIFO 队列

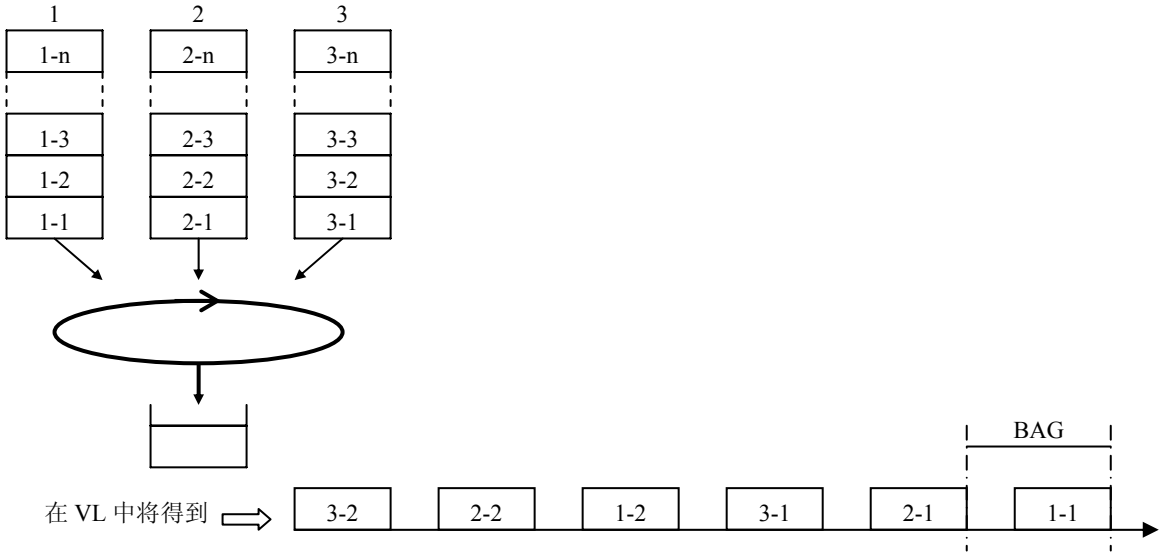


图 3-26 – VL 上的流量的第一个例子

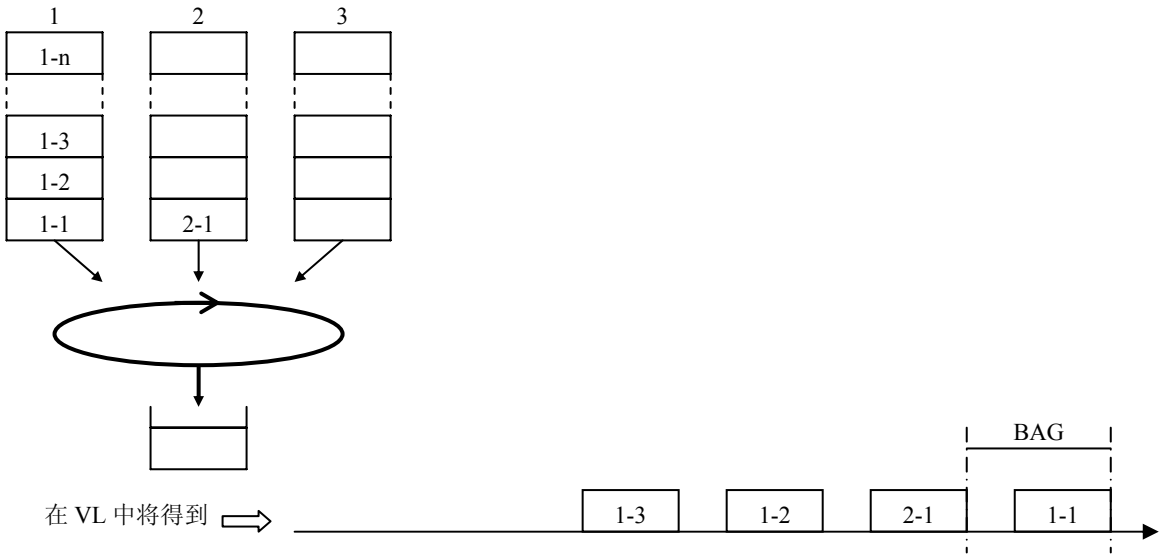


图 3-27 – VL 上的流量的第二个例子

一个VL的FIFO队列应该能够管理最多4条Sub-VL的FIFO队列。

每个Sub-VL的FIFO队列应该按照顺序轮询的方式被读取，这样如果任何Sub-VL的FIFO具有流量，每隔BAG，一个帧将被发送到主VL上。一旦一个帧被发送，轮询序列就被暂停，直到这个BAG间隔结束，（接着）序列从下一个Sub-VL的FIFO重新开始。

一个Sub-VL的FIFO队列应该仅被一个VL的FIFO队列读出。IP分片操作（如果需要）应该在IP层被执行。这将避免诸如短的采样消息被长的队列消息而延迟的情况。在出现IP分片的情况下，轮询继续进行；这样，如果从某个Sub-VL取出一个片断，随后（仍然继续轮询操作）从下一个Sub-VL进行取帧或片断的操作。

Sub-VL是位于IP层之下的。

3.3.2 简单文件传输协议的例子

该例子描述利用TFTP从LRU1发送一个文件到LRU2，如图3-28所示。在这里，两个VL被定义：VL1和VL2。

VL1：LRU1到LRU2，VL2：LRU2到LRU1。

在初始化阶段：

1. 传输由LRU1初始化，LRU1从源端口45000到目的端口69发出一个请求，在LRU2上端口69（即：端口号为69——译者注）是专用于TFTP的端口。
2. LRU2激活了一个TFTP会话，这个会话响应（LRU1的）请求，发送一条消息到LRU1的45000端口。这条消息指明这次传输所选定的接收端口（端口47000）。
3. 此时，连接被建立。LRU1能够以数据包发送文件。从LRU1到LRU2的通信使用源端口45000和目的端口47000。
4. 每个数据包的应答被LRU2发送，使用源端口47000和目的端口45000。

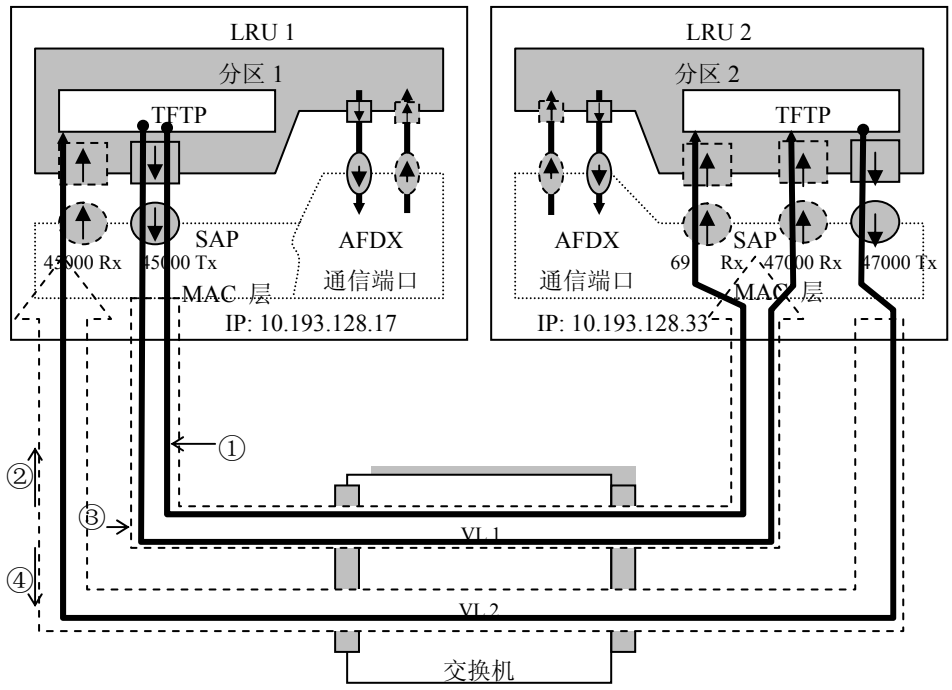


图 3-28 – 在 AFDX 网络中 TFTP 通信的例子

3.3.3 ES 通信协议栈

图3-29展示了ES通信协议栈。

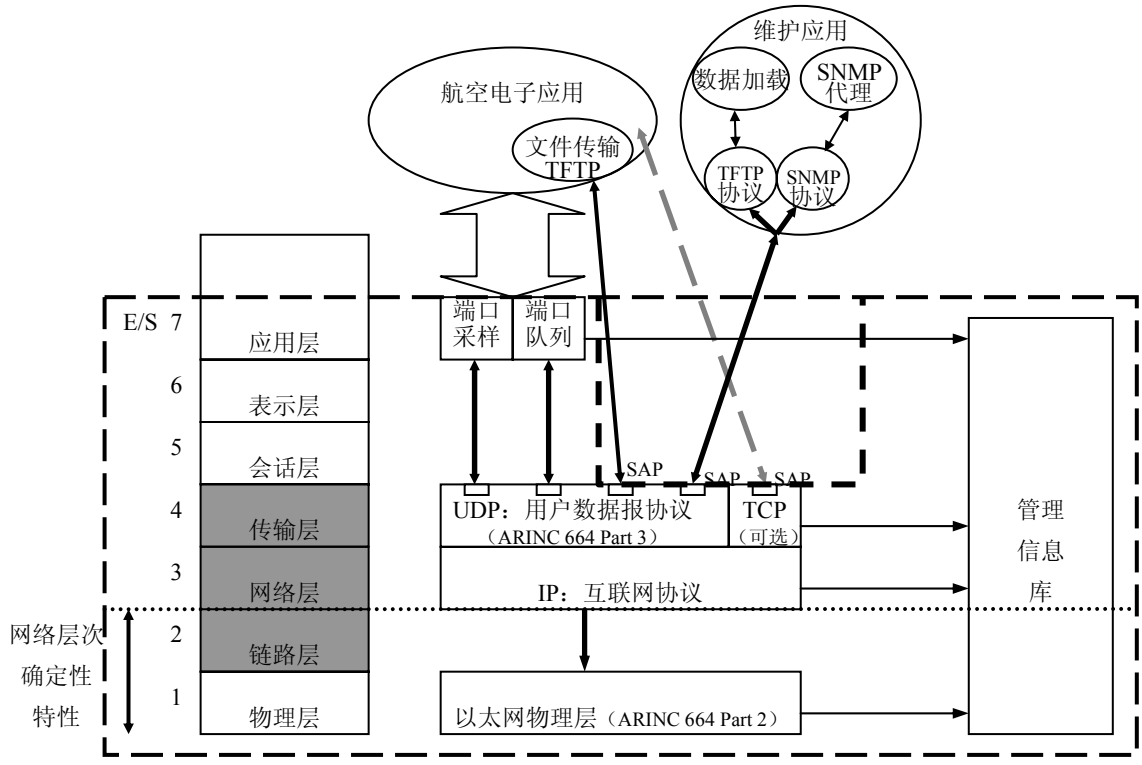


图 3-29 – ES 的协议栈

3.3.3.1 ES 的 MAC 协议定制

ES的数据链路层应基于使用IEEE 802.3标准定义的全双工以太网链路。

ES生成的每个以太网帧应与IEEE 802.3兼容。

即使在物理层故障的情况下，所有的输出接口将持续传输。

注释

这是为了防止长时间链路失效后（例如：在一个交换机复位或中间传输的物理层失效），发送被缓冲的旧帧。它也可能有助于避免从交换机到端系统以及交换机之间故障的传播。

最大的AFDX帧长度应该基于每个VL被指定。

在接收端，如果AFDX帧格式、帧校验序列（Frame Check Sequence, FCS）和CRC校验（Cyclic Redundancy Check，循环冗余校验）是有效的，该帧（不包括前导字和起始定界符域）应该被转发到上层。

3.3.3.2 ES 的 IP 协议定制

数据包结构的版本应该是IPv4。

IPv4的帧结构应该与图3-30相一致。

4 bit	4 bit	8 bit	16 bit	16 bit	3 bit	13 bit	8 bit	8 bit	16 bit	32 bit	32 bit	1-1473 byte
版本	IHL 头部长度	服务类型	总长度	片断标识	控制标志	片断偏移量	生存时间	协议	头部校验和	IP 源地址	IP 目的地址	IP 有效载荷

图 3-30 – IPv4 数据包结构

一般地，在IPv4数据包结构中，总长度域的取值范围应该是从21到1500字节。在AFDX中，由于存在顺序号（Sequence Number, 见3.2.6.2.2，冗余管理功能），范围是从21到1499。

注释

总长度域不将顺序号考虑在内。

在端系统中实现IP/IMCP UDP TCP的附加的条款，请参考附件2（Attachment 2）。

3.4 网络级别的互操作

3.4.1 编址

3.4.1.1 引言

在AFDX网络中数据流通过接收ES的UDP/TCP目的端口，IP目的地址，MAC目的地址和物理以太网连接被唯一地识别。

执行基于帧的过滤，接收端系统仅向通信端口或SAP端口转发有效帧。有效帧由分析目的地址（TCP/UDP, IP, MAC）和物理连接来确定。

3.4.1.2 无分片的 AFDX 帧结构

图3-31展示最小最大帧情况下的AFDX帧结构。

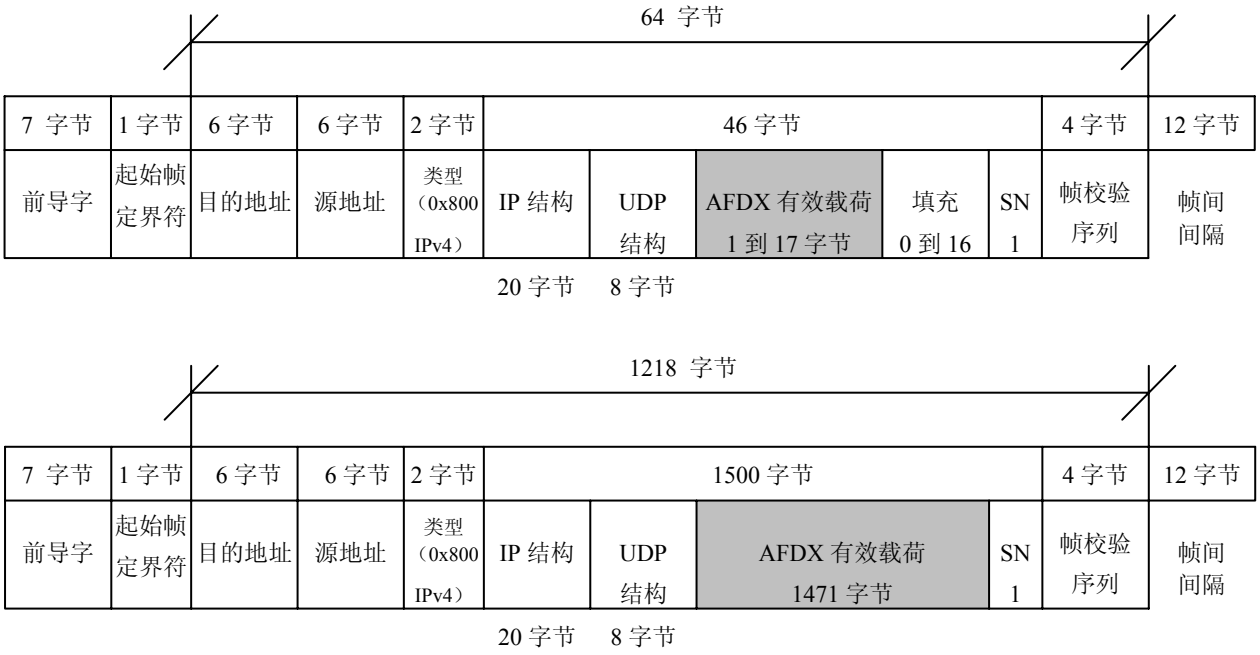


图 3-31 – AFDX 帧的结构

一个寻址规则的例子如图 3-32 所示。

在图 3-32 中，端系统 1 具有三条虚拟链路：VL1，VL2 和 VL3。

端系统 1 的分区 1 接入一条虚拟链路 VL1。

端系统 1 的分区 2 接入两条虚拟链路 VL2 和 VL3。

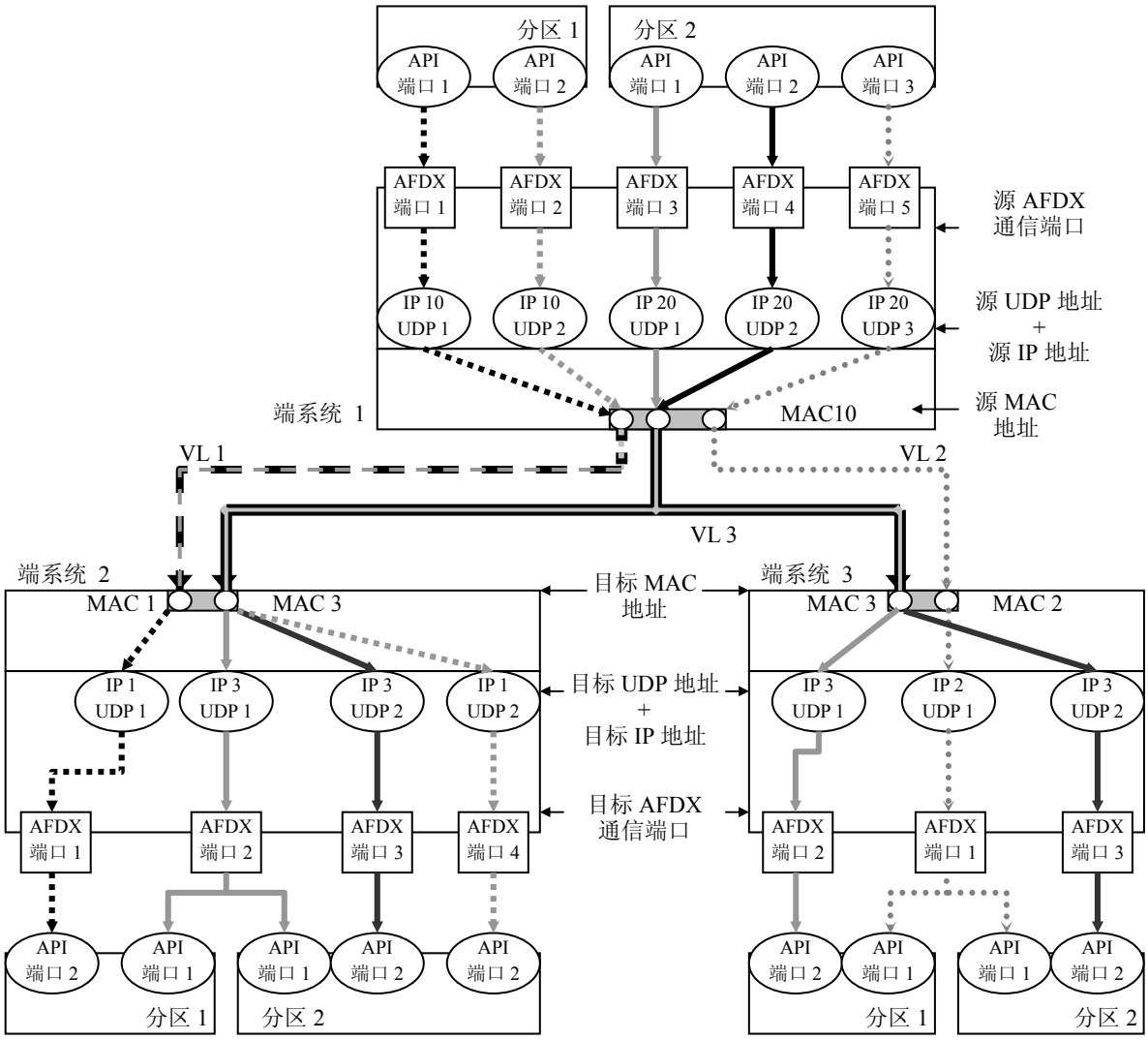


图 3-32 – 寻址的例子

下面的表格是每个 ES 的地址表。

端系统 1 的发送表

AFDX 源通信端口	源分区	源 UDP	源 IP	源 MAC	目的 UDP	目的 IP	目的 MAC
AFDX 端口 1	分区 1	UDP1	IP10	MAC10	UDP1	IP1	MAC1(VL1)
AFDX 端口 2	分区 1	UDP2	IP10	MAC10	UDP2	IP1	MAC1(VL1)
AFDX 端口 3	分区 2	UDP1	IP20	MAC10	UDP1	IP3	MAC3(VL3)
AFDX 端口 4	分区 2	UDP2	IP20	MAC10	UDP2	IP3	MAC3(VL3)
AFDX 端口 5	分区 2	UDP3	IP20	MAC10	UDP1	IP2	MAC2(VL2)

端系统 2 的接收表

AFDX 源通信端口	源分区	源 UDP	源 IP	源 MAC	目的 UDP	目的 IP	目的 MAC
AFDX 端口 1	分区 1	UDP1	IP10	MAC10	UDP1	IP1	MAC1(VL1)
AFDX 端口 4	分区 2	UDP2	IP10	MAC10	UDP2	IP1	MAC1(VL1)
AFDX 端口 2	分区 1 和分区 2	UDP1	IP20	MAC10	UDP1	IP3	MAC3(VL3)
AFDX 端口 3	分区 2	UDP2	IP20	MAC10	UDP2	IP3	MAC3(VL3)

端系统 3 的接收表

AFDX 源通信端口	源分区	源 UDP	源 IP	源 MAC	目的 UDP	目的 IP	目的 MAC
AFDX 端口 2	分区 1	UDP1	IP20	MAC10	UDP1	IP3	MAC3(VL3)
AFDX 端口 3	分区 2	UDP2	IP20	MAC10	UDP2	IP3	MAC3(VL3)
AFDX 端口 1	分区 1 和分区 2	UDP3	IP20	MAC10	UDP1	IP2	MAC2(VL2)

在接收器中，仅根据 MAC, IP 目的地址和 UDP 目的地址进行解多路复用。
图 3-33 展示了这个例子的物理拓扑结构。

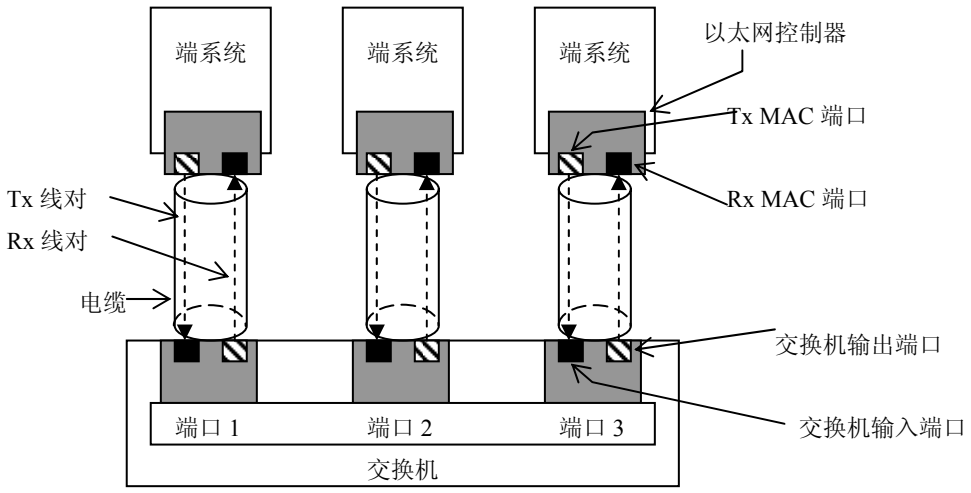


图 3-33 – 物理拓扑的例子

交换机转发由表 3-2 定义。

表 3-2 交换转发

输入端口	接收到的帧的 MAC 目的域	输出端口
1	MAC1 (VL1)	2
1	MAC2 (VL2)	3
1	MAC3 (VL3)	2 和 3

注释

（在 AFDX 网络中）MAC 地址应该被理解为潜在的单播或多播的以太网地址。

3.4.1.3 端到端（End-to-end）通信的标识

每个帧中层对层的对等（peer-to-peer）通信的标识方式是：源UDP端口+源IP地址+目的MAC地址(VL标识)+目的IP地址+目的UDP端口。

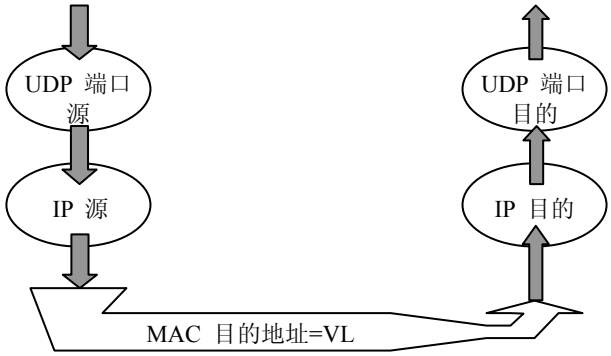


图 3-34 – 消息标识概念

对于一个源IP地址，应该有多个源UDP/TCP端口。对于一个目的IP地址，应该有多个目的UDP/TCP端口。

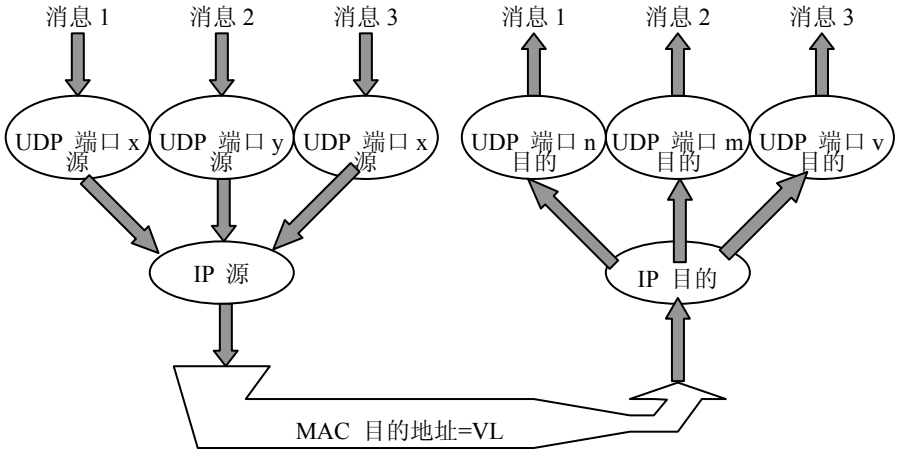


图 3-35 – 在一条虚拟链路上唯一的消息标识

在图3-35中，3个消息由3个五元组识别。

消息1 => 源UDP端口x + 源IP + 目的MAC + 目的IP + 目的UDP端口n；

消息2 => 源UDP端口y + 源IP + 目的MAC + 目的IP + 目的UDP端口m；

消息3 => 源UDP端口z + 源IP + 目的MAC + 目的IP + 目的UDP端口v。

3.4.1.3.1 AFDX 内部通信

在 AFDX 网络内部的端到端通信能够被认为是 AFDX 内部通信（Intra-AFDX）。

AFDX 内部通信的主要特征是对于每个消息，寻址是静态定义的。

对于单向的通信：

- AFDX 通信端口通过使用 UDP 端口进行定义，这样的端口能够是发送器或接收器（的端口）。
- AFDX 通信端口特性由采样和队列服务决定。

对于双向通信：

- 用方可能由 TFTP 构成（或其他未来发展的在 UDP/TCP 之上的协议），有两种可能：
  1. 利用 SAP 端口。这些访问点被连接到 UDP 或 TCP 端口，并且每个 SAP 能够作为一个发送器或接收器。为了获得一次双向通信，应该使用两个 SAP（例如：SAP 30000 Tx 和 SAP 30000 Rx）。
     在这个情况下，两个五元组将被确定，分别对应通信的每个方向。
     建议采用与互联网协议完全兼容的 SAP 端口：如端口号 69 被用于 TFTP。
  2. 使用传统的 AFDX 通信端口。双向通信，在一个单 ES 上应该需要两个 AFDX 通信端口：一个发送器和一个接收器（例如：AFDX 通信端口 15000Tx 和 AFDX 通信端口 15000Rx）。

对于双向通信，端口应该在队列模式下使用。

3.4.1.3.2 AFDX 与外网通信

本节描述在AFDX网络与兼容网络之间的通信。通信的两种模式被定义如下：

单向通信：

单向通信将总是从一个发送ES到一个兼容的网络，并且由一个UDP端口链路和传统的AFDX通信端口构成。这意味着ES配置表将包含目的IP和端口号。这样为了寻址将会存在一个静态定义的五元组。

双向通信：

可以利用TFTP，SNMP，615A协议或者其他基于UDP/TCP的协议（有待将来开发）。

使用SAP端口，并且像对于Intra-AFDX通信一样，每个SAP或者作发送器，或者作接收器。应该用两个SAP获得一个双向通信通道。

接收SAP能够将兼容网络中源的IP地址和UDP/TCP端口标识传递给分区。

发送SAP能够将兼容网络中目的的IP地址和UDP/TCP端口标识传递给分区。

3.4.1.4 IP 寻址格式

注释

这里供识别的 IP 地址范围可能与 ARINC 664 Part 4 中的寻址范围相冲突。只要这些地址只被用于 AFDX（封闭的）网络，就不会出现问题。

另一方面，如果 AFDX ES 将帧发送到这个封闭的网络之外（到另外的网络域（Domain)），就应该适用 ARINC664 part4。为了达到这一点，系统设计者可能要使用 IP 地址翻译机制。

3.4.1.4.1 IP 源地址

IP源地址应该被用以标识与端系统相关联的发送分区。

IP目的地址应该被某个端系统用来向一个或多个目的端系统转发IP数据包。

IP地址应是在A类地址，并且为私有网络单播地址（前8个bit应该是”0000 1010”）。

IP单播地址格式如图3-36所示。

User_Defined_ID（用户定义标识）是一个单独的16-bit的位域。它应该按系统集成者所期望的合适（方案），被用来给网络中每个IP可寻址的主机赋以一个唯一的并且有意义的IP地址。

IP 单播地址格式（源或单播目的）32 bit				
A类 1 bit	私有地址IP 7 bit	User_Defined_ID（用户定义标识） 16 bit	Partition_ID（分区标识）	
“0”	“000 1010”	“nnnn nnnn nnnn nnnn”	空余域 3 bit	5 bit

图 3-36 – IP 单播地址格式

Partition_ID（分区标识）由两个位域组成：

空余域（Spare field）一般不用并设为0，如图3-37。这些二进制位可能被用来在系统具有超过32个分区的情况下作为分区的标识。

空余域	含义
0 0 0	无特殊含义

图 3-37 – 空余域定义

AFDX帧中IP头部的源地址应该是IP单播地址，用来识别发送器。

3.4.1.4.2 目的 IP 地址

AFDX帧中头部的目的IP地址应该是：

- 要么是 IP 单播地址，用来识别目标订阅者；
- 要么是 IP 多播地址，与图 3-38 所示的格式兼容。

IP 寻址格式 32 bit		
4 bit	28 bit	
D类 “1110”	IP多播标识符	
	固定域 12 bit = “0000 1110 0000”	虚拟链路标识符 16 bit

图 3-38 – IP 多播地址格式

3.4.1.5 AFDX 通信端口，SAP 和 UDP/TCP 寻址格式

对于AFDX内部通信，在端系统和分区之间存在两种接口：AFDX通信端口和SAP端口。

AFDX通信端口，如图3-39所示，具有如下特性：

- 单向接入：发送（Tx）或接收（Rx）；
- 采样和队列模式：采样和队列两种模式仅在接收时具有意义；
- 在发送时，在一个“AFDX 通信端口”和五元组（源 UDP 端口, 源 IP 地址, 目的 MAC 地址，目的 IP 地址，UDP 目的端口）之间仅只有一条链路。AFDX 通信端口属于唯一分区。
- 在接收时，在一个“AFDX 通信端口”和五元组（源 UDP 端口, 源 IP 地址, 目的 MAC 地址，目的 IP 地址，UDP 目的端口）之间仅只有一条链路。“AFDX 通信端口”可以被多个不同的分区访问。

- 发送和接收的路径由配置所固定，如图 3-39 所示。

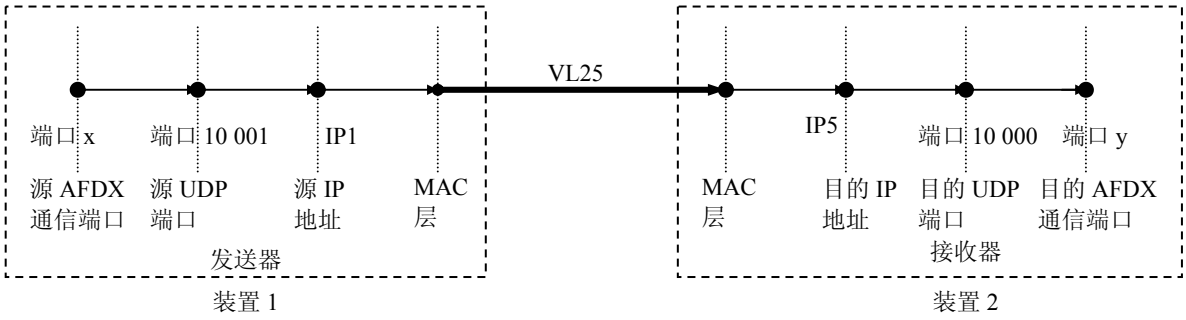


图 3-39 – AFDX 通信端口

SAP 端口（如图 3-40 所示）具有如下特点：

- 一个 SAP 端口被映射到一个 UDP（或者根据将来的扩展，映射到 TCP）端口，术语“SAP”是为了与“AFDX 通信端口”相区别
- 单向接入：发送（Tx）或者接收（Rx）。
- 两个 SAP 端口可以组成一个双向通信接入，例如：端口 500 Tx 和端口 500 Rx。
- 发送中，SAP 端口使用固定配置的元组（源 UDP 端口, 源 IP 地址, 源 MAC 地址，目的 MAC 地址（VL 标识）），目的 IP 地址和目的 UDP（或 TCP）端口由分区提供。
- 接收中，SAP 仅与一个“目的端口+目的 IP 地址+目的 MAC 地址（VL 标识）+源 MAC 地址”的通信相连。源 IP 地址和源 UDP（或 TCP）端口由 ES 派发到分区。

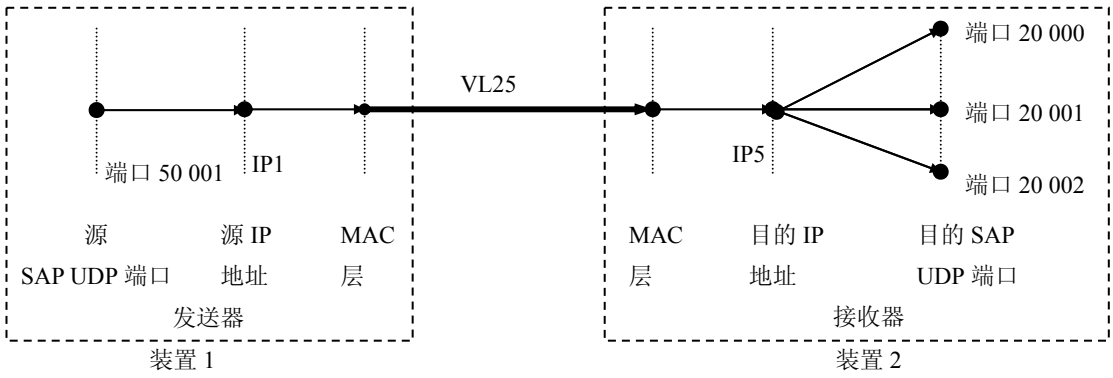


图 3-40 – 使用 UDP 的 SAP 端口

3.4.1.5.1 AFDX 通信端口

发送中，一个AFDX通信端口应该仅与单独的一组地址相链接：源UDP端口，源IP地址，VL（目的MAC地址），目的IP地址和目的UDP端口。

接收中，一个AFDX通信端口应该仅与单独的一组地址相链接（唯一的目的UDP端口， 目的IP地址，VL（目的MAC地址）），并且，如果冗余管理关闭，还应与以太网物理接口相连。

3.4.1.5.2 SAP 端口

发送中，一个SAP端口应该仅与单独的一组地址相链接（源UDP端口， 源IP地址和VL（目的MAC地址））。

接收中，一个SAP端口应该仅与单独的一组地址相链接（目的UDP/TCP端口，目的IP地址和VL（目的MAC地址））。

应该以UDP/TCP端口号识别服务访问点。

接收中，SAP应该使源IP地址和源UDP/TCP端口对于接收分区可用。

发送中，SAP应该允许分区指定目的分区的IP地址和UDP/TCP端口。

3.4.1.5.3 SAP 和 AFDX 通信端口号的分配

端口号的分配在**ARINC 664 Specification part 4** “飞机数据网络（ADN），Part 4 – 基于Internet的地址结构和号码指派”中定义。图3-41给出这种分配范围，以及对于AFDX和选择方法。

端口范围 （以十进制值表达）	ARINC 664 分配范围	AFDX 分配范围
0-1,023	ICANN[2]监管， “周知”端口号	ICANN 监管， “周知”端口号
1,024-16,383	ICANN 注册， ARINC 664 指派	由网络管理员指派
16,384-32,767	ICANN 注册， 系统集成者或用户定义	
32,768-65,535	ICANN 注册， 建议作为临时端口指派	

图 3-41 – SAP 和 AFDX 端口号的分配

每一个IP单播（地址）或多播（地址），端口分配范围的重新划分如图3-42所示。

端口类型	通信类型	端口范围	注释
AFDX 通信端口	AFDX ⇔ AFDX AFDX ⇔ 兼容网络	1,024 – 65,635	用于采样和队列通信
SAP	AFDX ⇔ AFDX AFDX ⇔ 兼容网络	0-1,023	用于标准的通信：例如，端口 69 打开一个 TFTP，数据加载（ARINC 615A），SNMP 等。
	AFDX ⇔ AFDX AFDX ⇔ 兼容网络	1,024 – 65,635	用于双向通信：具体的 TFTP 传输等

图 3-42 – 对于 IP 单播或多播的端口分配范围

注释

端口号重复使用: 仅有的技术限制在于——两个相同的端口号不能用在同一个 VL 中。但是，也许由系统设计者可以添加一些规则，使得 UDP 端口号在网络中唯一，或者在另一方面，在双向通信中强制使用同一个端口号。

[2] ICANN—Internet Corporation for Assigned Names and Numbers，互联网名称与地址分配机构。

4 .0 交换机规范

4.1 基本概念

交换机包含五个相互作用的功能块（block），如图4-1所示。

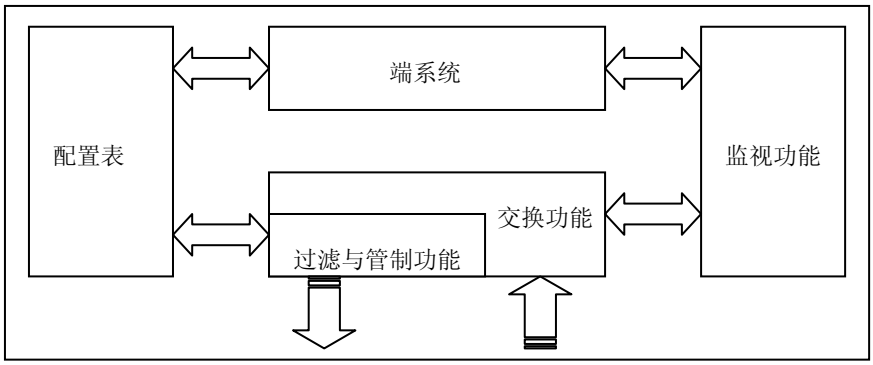


图 4-1 AFDX 交换机的主要功能模块

所有到达交换机的帧，在过滤与管制（Filtering & Policing）功能阶段都要在不同的步骤中被过滤，这些步骤中采用的规则涉及帧的完整性，帧长度，流量预算，以及可接受的目的（地址）。

交换功能（Switching function）执行交换的核心工作。经过过滤与管制功能处理的帧被转发到合适的物理输出端口，通过这些端口它们再次离开交换机。

这些功能都由静态配置表中的配置数据控制。

ES 功能模块提供了外部设备与交换机通信的方法（将接收的帧传送给交换机，并允许交换机向外发送帧）。这主要是用于数据加载以及监视功能的实现。

所有的操作都被监视功能（Monitoring function）块监视，该模块记录事件（日志），诸如：某个帧的到达，或者一次 CRC 校验失败；并且还创建关于内部状态的统计量。因为交换机是网络的一部分，监视功能与网络管理功能（Network Management Function）通信，通信内容是操作信息和有关于健康状态的信息。

4.1.1 过滤与管制功能概述

4.1.1.1 管制与过滤参数

在本章节中假设编号为*i*的虚拟链路VLi具有如下的特性：

- VLi 的 BAG（带宽分配间隔）是 BAGi（以秒为单位）。
- VLi 进入一个特定的交换机的时延抖动（Jitter，代表一个时间窗口（的范围），落在该窗口中的帧被确认）是 $J_{i,\mathrm{switch}}$（以秒为单位）。
- VLi 的最大帧长度[3]被计作 $S_{\max,i}$（以字节为单位），它是一个可配置参数，取值范围是[84,1538]。
- VLi 的最小帧长度被计作 $S_{\min,i}$（以字节为单位），它是一个可配置参数，取值范围是[84,1538]。$S_{\min,i}$ 应该小于或等于 $S_{\max,i}$。

帧长度的取值如图 4-2。实际物理线路上的以太网帧还应当在 MAC 帧的基础上加上 12 字节的帧间间隙（IFG）、7 字节的前导字、1 字节的起始标识（SFD）：总长度为 64 到 1518 字节。

[3] 原文中对于帧长度变量的字母标号和角标前后不一致，译文中将它们统一为 $L_{\max,i}$，$L_{\min,i}$ 或 $S_{\max,i}$，$S_{\min,i}$（线路帧），且用大写字母表示——译者注。

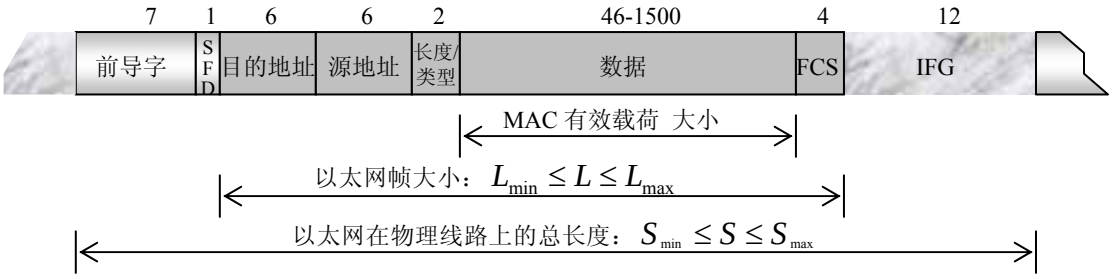


图 4-2 – 帧大小的数值

4.1.1.2 帧过滤

过滤机制使交换机仅仅向选定的目的传送有效的帧。一旦帧到达交换机，就将被检查。并且监视帧头中给定域的内容（例如：目的地址域，帧校验序列域等）和帧本身的构造，如下：

- 帧大小：帧是否长于或短于封装所允许的范围；
- 帧完整性：嵌入在帧中的 FCS 是否与接收的内容的计算值匹配；
- 帧路径：根据到达的帧中的目的地址域（在 AFDX 网络中，是虚拟链路标识符），该域所请求的目的地址是否允许访问。

如果帧的属性与配置参数不符合，帧被过滤，即：被丢弃，并且一个或多个MIB实体被更新。MIB实体的定义，以及更新条件和处理的方法在相关的规范文档中被规定。

作为过滤功能的一部分，对帧进行如下的检查：

- 目标地址的有效；（与一条有效 VL 相关的以太网地址，包括固定的位域）
- 在目的端口（根据交换机的配置表），该 VL 将被有效地接收；
- 帧校验序列有效；
- 以太网帧长度（L）是 8-bit 的整数倍；（数据对齐）
- 以太网帧长度（L）在 [64，1518] 字节的范围内；
- 以太网帧长度（L）小于或等于 L_{max}；
- 在采用基于字节的流量管制方法的情况下，以太网帧（线路上的）长度（S）大于或等于 S_{min}。

注释

对于基于字节的流量管制算法的确定性的证明，基于 Smin 参数。尽管在给定的一条 VL 中任何小于 S_{min} 的帧必须被丢弃，为了保证该算法总是有效的，特别是在 ES 功能紊乱的情况下，这个最小值必须被考虑。

在AFDX应用的情况下，（以太网帧中的）长度/类型域被用来作为类型域，帧长度的一致性检查不包含这个域。

在航空环境下，“有效帧”的定义可能比商业环境下的定义更加严格。在网络的语境下，表示“强制执行（有效性检查及其后续操作）”的术语是“管制”（policing）。

4.1.1.3 流量管制

本节描述了一种基于目的地址执行流量管制算法的模型。目的地址域包含用以在某个AFDX环境下识别虚拟链路的信息。虚拟链路定义的流具有一些确切的流量属性，这些属性包括：一组接收者，两帧之间的最小允许间隔。其通信流量必须以隔离的方法进行维护，以使相应的属性得到保证。为了在本规范中与商用产品文档中采用同样的术语，“目的地址”被作为术语“虚拟链路”或“VL”的同义词来使用。

抖动现象依赖于虚拟链路和交换机的属性，它是到达一个特定交换机的所有虚拟链路流量的函数。如果一个虚拟链路跨越多个交换机，实际的抖动在每个中间的交换机可能是不同的。但是，对于任意的交换机，时延和抖动的最大值具有一个上界。

流量管制模型是从端系统的角度描述的，因为端系统是进入交换机的主要流量的生成器，如图4-3。而基于模型对交换机进行描述则是一种以交换机为中心的视角，以这种视角观察的属性，是从交换机的某种合适的实现中提取的。

流量管制可能以两种不同的算法实现，根据用以证明确定性的数学方法的不同，分为基于字节的流量管制和基于帧的流量管制。

- 基于字节的流量管制以“比特每秒”为单位表示带宽使用率，过滤输出 VL；
- 基于帧的流量管制以“帧每秒”为单位表示的带宽使用率，过滤输出 VL。

交换机要么可能实现两种算法中的一种，在基于字节（Byte-based）和基于帧（Frame-based）的算法中选择其一，要么两个都实现。算法的选择将影响证明网络可调度性（schedulability）的方法。

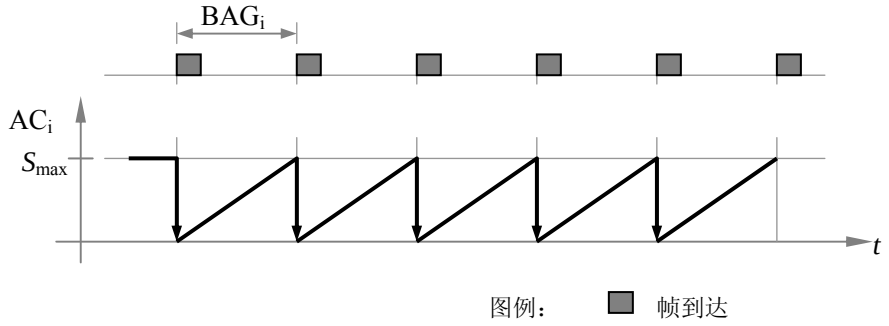


图 4-3 没有抖动的流量的例子

管制算法的描述如下：

初始化，VLi的帐户ACcount（也记作ACi ，以字节为单位）被设置为：$S_{\max,i}\cdot\left[1+\frac{J_{i,\text{switch}}}{\text{BAG}_i}\right]$；

ACi 随着时间的推进，按照$\left(\frac{S_{\max,i}}{\text{BAG}_i}\right)$的比率增长，但最高不能超过$S_{\max,i}\cdot\left[1+\frac{J_{i,\text{switch}}}{\text{BAG}_i}\right]$。

- 每当 VLi 中的一个帧到达交换机的时候，都要检查 ACi。
- 考虑以太网线路上的接收帧的总长度：S=以太网帧长度（L）+20 个字节；这 20 个字节代表 帧间间隔 IFG+前导字 Preambe+帧起始定界符 SFD。

注释

如果 $J_{i,\text{switch}}$=0，则 ACi 被初始化设置为 $S_{\max,i}$（如图 4-3）。

存在两种被管理的帐户：Byte ACi（基于字节的过滤）和 Frame ACi（基于帧的过滤）

对于基于字节的过滤：

• 如果 Byte ACi 大于 S，则帧被接受并且从帐户 ACi 的值中取出 S ；

• 如果 Byte ACi 小于 S，该帧被丢弃，相应的 MIB（Management Information Base，管理信息库）实体被刷新，Byte ACi 的值不变。

对于基于帧的过滤：

- 如果 Frame ACi 大于 $S_{max,i}$，则帧被接受并且从帐户 ACi 的值中取出 $S_{max,i}$ ；
- 如果 Frame ACi 小于 $S_{max,i}$ ，该帧被丢弃，相应的 MIB 实体被刷新，Frame ACi 的值不变。

该流量过滤规则在文献中被称为“令牌桶”（token bucket）算法。

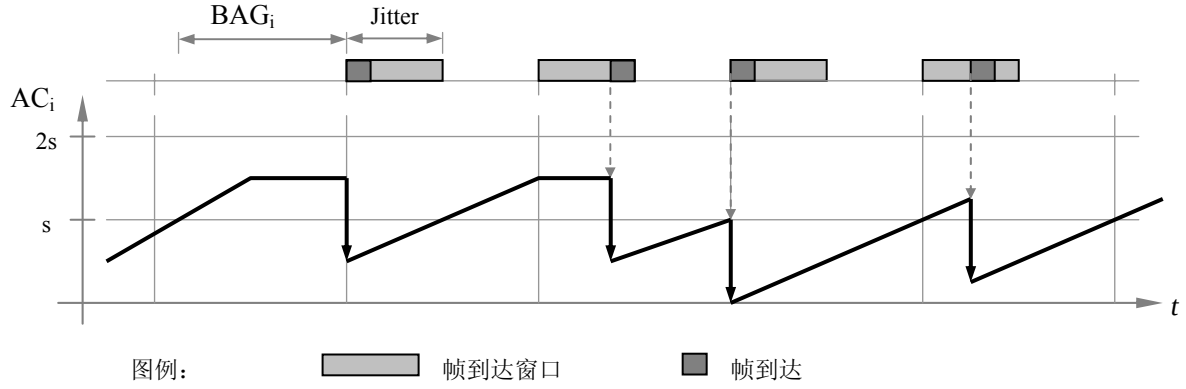


图 4-4 抖动为 BAG/2 情况下的流量的例子

流量过滤机制应该在交换机上实现，以保证网络的故障抑制功能。为了不让某个失效的端系统干扰网络，流量中的任何与网络配置不适合的帧应该必须被丢弃。

4.2 过滤与管制功能

4.2.1 帧过滤

应使预算帐户（Byte ACi和Frame ACi）不被无效的帧（即那些不被转发的帧）消耗。被管制机制丢弃的帧不应计入预算。

注释

假使基于流量的管制过程为丢弃的帧消耗预算，通过交换机的帧必须等待的最小时间段就要由前一个被丢弃的帧算起，而不再是由前一个被（交换机）发出的帧算起。这样，如果帧序列中的帧与帧之间的间隔（除了流量控制所必需的专用时间间隔之外——译者注）没有多余的空隙，该序列将被“有效地”阻塞，而不是将它的流量限制到设定的预算量。注意：管制的目的是将流量限制到设定的预算量，而不是为了在不符合预算的情况下阻塞全部的数据流。

为了保证整个网络的稳健性，只转发没有被破坏的帧是重要的。所以，对于到达的流量，交换机应该按照IEEE Std 802.3使用帧校验序列CRC（循环冗余校验）域对每个帧进行检查。对于不满足该项检查的帧，交换机应该予以丢弃。

交换机应该丢弃帧的长度（L）大于1518字节或小于64字节的输入帧。

交换机应该丢弃帧的长度不是8-bit的整数倍的输入帧（对齐错误）。

如果输入帧的以太网线路总长度大于对应的VL允许的最大长度值S_{max} ，交换机应该将它丢弃，因为它将消耗超过分配带宽的通信资源。

在使用基于字节的管制的情况下，如果输入帧的以太网线路总长度小于对应的VL允许的最小长度值，交换机也应该将它丢弃。

交换机应该丢弃目的MAC地址中32-bit的固定地址错误的帧。

交换机应该丢弃虚拟链路标识（VL ID）与交换机输入端口不对应的输入帧。

4.2.2 流量管制

管制算法基于对每个VL的帐户的管理。

交换机应该用VL的标识符从配置表中得到过滤与管制的相应信息。每个收到的帧都按这些存储在配置中的信息来过滤和管制。通过VL ID访问相关的参数。

MAC目的地址和ACi之间的两种类型的关系应该被支持：

- 一个 ACi 对应一个唯一的 MAC 目的地址（（单条）VLi）。
- 一个 ACi 对应一组（几个）MAC 目的地址（多条 VL）：帐户共享。

如果一个VL使用共享的帐户，它只能对应唯一的一个帐户，并且使用这个帐户的一部分。

注释

分区可以得益于这种帐户共享能力，特别是如果帧的丢失可以被接受。该能力会在VL各自的隔离间取得妥协，并削减带宽保证。

出于安全的原因，最好将一个目的地址固定地联系一个帐户。然而，采用帐户共享特性是基于网络管理者和/或应用设计人员的决定。

例如，几个LRU可能对一些VL使用帐户共享，用来从LRU到数据加载器（Data Loader）的数据加载。所有的LRU同时进行数据加载是极为不可能的。这样，定义一个从每一个LRU到数据加载器的VL会浪费带宽。而且，ARINC 615A中含有的恢复机制允许有限制的帧的丢失。

帐户共享仅是用与交换机中的功能，不影响LRU-ES的配置或功能。

配置表中有目的MAC地址与ACi，BAG，Jitter，S_{max}之间的关联，在使用基于字节的流量管制的情况下，可能还包括与S_{min}的关联。

流量管制应该基于（如下）参数：BAG, Jitter, S_{max}，S_{min}（在使用基于字节的流量管制的情况下）。它应该机械地执行在“过滤与管制功能概述”（4.1.1节）中所述的算法。

对于每一个VL或一组共享同一个帐户的多个VL，流量管制功能应该根据配制表核准一个BAG值。

交换机的流量管制功能至少应该使BAG的值在从1ms到128ms的范围内可配制。

对于每一个VL或一组共享同一个帐户的多个VL，流量管制功能应该根据配制表核准一个最大的Jitter值。

交换机的流量管制功能至少应该使最大可允许的Jitter值在从0到10毫秒的范围内可配制。

这意味着ACi至少具有一个为 $S_{\max,i} \cdot \left[1 + \frac{10}{BAG_i}\right]$（BAG的单位取毫秒）的最大值。

流量管制功能应该至少能够处理帧长度（L）在[64~1518]字节之间的以太网帧。

注释

这个与以太网帧大小（L）有关的（取值）范围：不包括 IFG+前导字+SFD 的20个字节的长度。

应该使每条VL对应一个BAG值，每条VL对应一个Jitter值，每条VL对应一个最大帧长度值。

当一组VL共享一个ACi，该组中的所有的VL应该具有相同的BAG，以及相同的最大和最小的以太网线路总长度（S_{max}，S_{min}）的值。

当一组VL共享一个ACi，流量管制功能应该采用该组中所有的VL中最大的Jitter值作为这个ACi的Jitter值。

流量应该在小于或等于100μs的时间解析度下被管制。

注释

这提供了一个考虑到管制功能的时间颗粒度的界限；这样，对于不是很准确的实现的影响，（起到）限制作用。

流量被管制时，时间解析度应该具有$\pm 10^{-4}$的相对冗余量。

注释

这具体规定了对于被配制的时间解析度值，交换机管制频度的准确度。

注释

上面的两个参数不影响互操作性。从这个角度来看，它们可以被忽略。然而，设计者应该考虑这些交换机的两个参数，因为它们对网络的整体性能（延迟，确定性）有影响。

4.3 （空缺）

4.4 交换功能

这一节对交换机的交换功能进行说明。

属于某个虚拟链路的输入和输出的帧的次序应该被交换机保持。

注释

AFDX用户期望以帧发送时同样的次序对它们进行接收。

交换机不应该修改输入帧的帧校验序列。因为交换机作为网络元件应该被嵌入在帧结构中的帧完整性机制所覆盖。在任何环境下不可以在交换机中被重新生成CRC值，包括在重传时。

对于每个帧，目的地址域的内容应该被用于从配置表得到合适的一个或多个目的端口，它们是帧必须被转发到的端口。

如果一个输出端口由于缓冲区拥塞不能接收某个帧，对于这个端口这个帧应该被丢弃。

注释

这是交换引擎中最关键的功能之一。无论在给定的输出端口发生什么情况，必须保证交换机引擎持续地循环工作。

在一个端口链路失效的情况下，即将被转发到该端口，或是被保存在这个端口的缓冲区中的帧将被丢弃。

输出端口不应传输比“最大延迟”（max delay）更晚的帧，“最大延迟”参数在配置表中基于每个输出端口被定义。

在一个给定的端口，一个帧的最大延迟参数被定义为在下列前后两个事件之间经过的最长的时间。这两个事件是：

- 某个帧的最后的一个二进制位到达某个交换机的输入端口；
- 这个帧的最后的一个二进制位从这个交换机的给定的输出端口离开

交换机应该具有在任意端口接收一个帧并将它转发到任意端口组合的能力（包括刚才的接收端口）。例如，两个分区在同一个主机上，它们之间的数据交换也能够在AFDX网络上进行，这样可以提高数据交换的可观测性和可移植性。

交换机应该具有基于目的MAC地址的流量优先级机制，分为高优先级和低优先级两类流量。优先级应该在配置表中基于虚拟链路定义。

对于每个输出端口，优先级被设置为高的帧应该先于优先级被设置为低的帧发出。

在被发送的过程中的低优先级的帧不应由于高优先级的帧的到达而被抢占。

4.5 交换机 ES 功能

4.5.1 概述

除了与网络冗余有关的内容之外，交换机的端系统应该符合本文档 3.0 章节的要求。

4.5.2 寻址策略

该端系统应该使用它自有的 MAC 单播地址作为它发出帧的源 MAC 地址。

4.6 监视功能

AFDX 监视功能基于：

- 在每一个 AFDX 组件（装置，网络用户和交换机）上实现的 MIB 库（Management Information Base，管理信息库），用以存储关于这些组件的信息；
- 在每一个 AFDX 组件（装置，网络用户和交换机）上实现 SNMP 代理，使用 SNMP 与网络管理功能（Network Management Function）通信；
- 网络管理功能从所有组件收集信息，实现信息的关联，用来检测/定位失效，并用来分析网络性能。

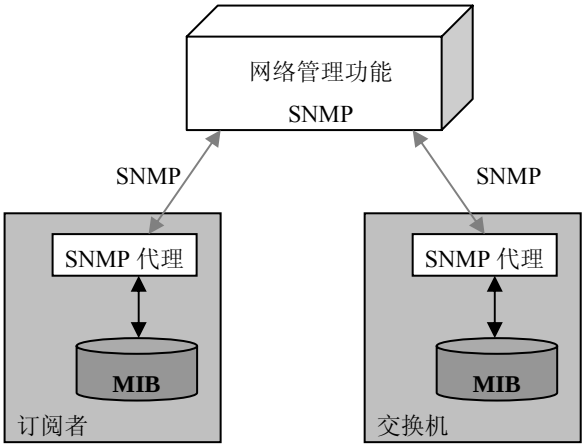


图 4-5 – 航空电子数据通信网络监视概况

交换机包含一个 AFDX 端系统，它是一个 AFDX 订阅者。该交换机应实现一个 MIB 库和一个 SNMP 代理。另外，交换机的 MIB 库包含一个唯一的与交换机功能有关的 MIB 对象。

故障检测应该可以修改由交换机管理的 MIB 库中的状态对象。

故障/健康 MIB 变量应该每 100ms（如同其他的 MIB 变量）更新一次。这将保证当故障/健康标志被发出后，将总是反映交换机的当前状态。

注释

通过 SNMP 请求得到的数据应该反映交换机的当前状态。

交换机应该带有一个故障/健康标志。当至少一个使用中的以太网端口处于故障中的时候，故障/健康标志应该被设置为“故障”。故障/健康标志的输出应该至少每隔 100ms 更新一次。

注释

这个故障/健康标志能够被发出到飞机告警系统（Flight Warning System）。这些故障/健康标志的实现（例如：离散量输出）应该在一个 ARINC 规范中定义。

4.7 配置文件

4.7.1 引言

每个交换机，如图4-6所示，拥有至少2个配置文件：默认配置文件和OPS（操作模式）配置文件。根据操作模式，合适的文件以一种互斥的模式被选定。在一个网络中对于所有交换机，所有文件必须是一样的。这样的文件的特征是：

- 默认配置文件 Default_Configuration_File：对应于驻留配置，在交换机为空的时候使用（现场可加载软件还没有被加载），或者在交换机正在被加载时使用（在这两种情况下，交换机在数据加载（Dataloading mode, DL）模式）。
- OPS 配置文件 OPS_Configuration_File：对应于可加载配置，当在交换机操作模式下使用（Operational mode, OPS）。

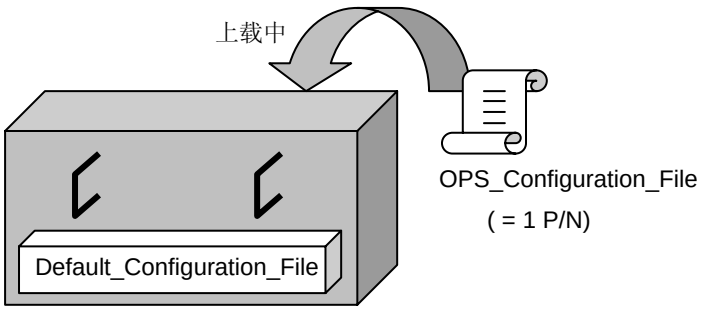


图 4-6 – 配置文件的概况

注释

另外的配置文件也可以被定义（例如：内场文件（shop file），用于维护和产品测试）。

每一个 OPS_Configuration_File 包含几个表格。交换机将根据交换位置提取相关的表格。交换机位置将根据某种能够在某个 ARINC 标准中定义的方法被识别。在这种方法中，从某个交换机出发，或是到某个交换机的编址是唯一的。交换机使用存储在配置表（Configuration Tables）中的参数用以实现：

- 过滤与管制功能、
- 交换功能、
- 端系统功能。

在下面的段落中，假设根据 12 支管脚识别交换机的位置。这种约定并不是来自于规范化的考虑，

而只是作为一个例子。然而，如果实现中决定使用这样 12 支管脚，则下面的定义应被采用。

4.7.2 Default_Configuration_Table

Default_Configuration_Table 与交换机的硬件 P/N（Hardware P/N，硬件部件号）是不可分离的，并且只能在内场改变，与硬件 P/N 的修改有关。

Default_Configuration_Table 用来定义交换机的端系统的默认行为：默认接收和默认发送。这个表格应该是常驻的。

4.7.2.1 默认物理端口

Port #1 是与交换机的端系统通信的默认外部物理端口。默认端口的速率被设置为 100Mbps，无速率自协商。默认端口允许交换机的端系统接收和发送帧，甚至在它还没有加载它的可加载的配置之前，也允许这样的操作。

4.7.2.2 默认接收配置

依赖于在飞机网络中的位置，通过管脚编程获得，交换机默认地订制（subscrible）唯一的接收 VL，下文中称为 $VL_{(0,\ position)}$。

$VL_{(0,\ position)}$的含义为：

- 下标中“0”代表它是一个接收 VL，
- “position”表示 VL 依赖于交换机在飞机中的位置。

交换机的 Default_Configuration_Table 中关于接收的部分应该在非易失存储器中,至少包含下列信息（默认接收 VL 的特性）：

- VL 标识（目的 MAC 地址）：$VL_{(0,\ position)}$；
- 最大允许的以太网线路总长度（S_{max}）；
- 带宽分配间隔。

默认接收 VL 必须具有与交换机自有的端系统通信的能力，即使交换机没有用 OPS_Configuration_File 加载（至少能够进行数据加载）。

这个默认的 VL 被用来与网络管理功能和数据加载器（Data-Loader）通信。帧的长度与以太网线路总长度有关（包括所有协议开销，IFG，前导字和 SFD）。

交换机的端系统功能将仅考虑目标 IP 地址域与它自己的 IP 相同的帧（其 IP 地址由管脚 P_1 到 P_{12} 的编程得到，参见 4.10.2.1 节）。根据配置文件中定义的值，交换机的管制功能在默认的 VL 上被激活。

4.7.2.3 默认发送配置

每一个交换机有一个唯一的发送 VL，下文中记为 $VL_{(1,\ position)}$。VL 的标识依赖于交换机在飞机中的位置，如 Dafult_Configuration_File 中的定义，并在默认数据加载 VL（Data Load VL）一节中被详细说明。

$VL_{(1,\ position)}$的含义为：

- 下标中“1”代表它是一个发送 VL，
- “position”表示 VL 依赖于交换机在飞机中的位置。

交换机的 Default_Configuration_Table 中关于发送的部分应该在非易失存储器中至少包含下列信息（默认发送 VL 的特性）：

- VL 标识（MAC 目的地址）：$VL_{(1,\ position)}$，遵守图 4-7 所示的特性；
- 最大允许的以太网线路总长度（S_{max}）；
- 带宽分配间隔。

0	0	0	0	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}

图 4-7 – VL 标识符的特性

其中 P_1-P_{12} 对应于管脚编程值：1=GROUND（接地）， 0=OPEN（悬空），

基于这个 12 位二进制数的值，计算出 $VL_{(1, position)}$的 VL ID ，它对应着；

- 带宽分配间隔；
- 最大允许的以太网线路总长度（S_{max}）。

默认发送 VL 将允许交换机发送信息，甚至在没有加载它的 OPS_Configuration_File 的情况下（至少能够进行数据加载的应答）。这个默认的 VL 被用来与网络管理功能和数据加载器通信。帧的长度与以太网线路总长度有关（包括所有协议开销，IFG，前导字和 SFD）。

当交换机的端系统发送帧时，在 AFDX 帧中的源 IP 和源 MAC 地址域中将是自身的交换机 IP 和 MAC 址，由 P_1 到 P_{12} 的管脚编程确定。（参见 4.10.2.1 节）

4.7.3 现场可加载配置表：OPS_Configuration_File

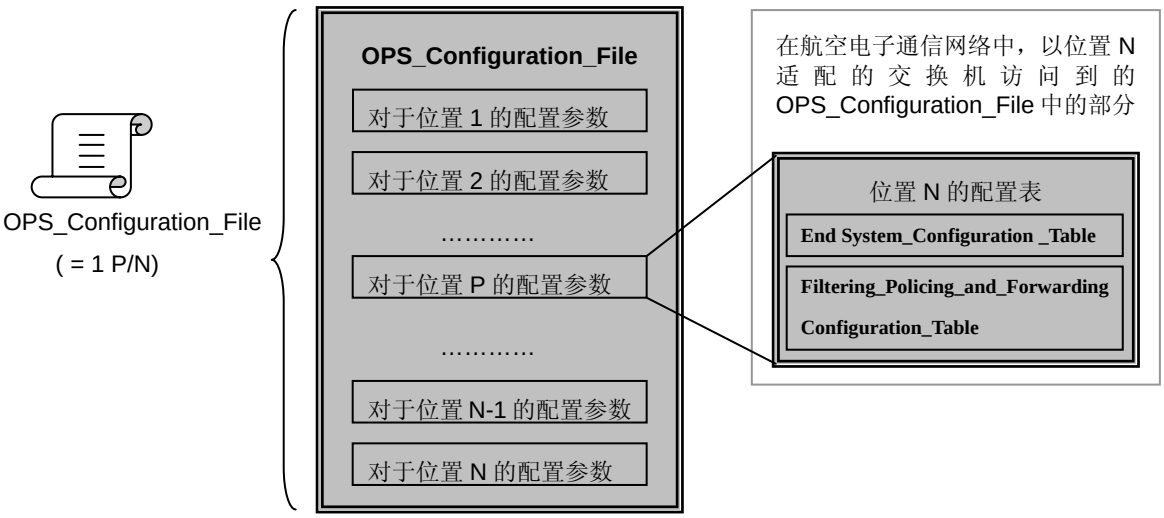


图 4-8 – OPS_Configuration_File 概况

在航空电子数据通信网络（Avionics Data Communications Network）中的所有交换机都加载同样的 OPS_Configuration_File（对于同样的 P/N）。OPS_Configuration_File，如图 4-8 所示，包含所有交换机的操作配置。根据当前的位置，由管脚编程得到，交换机仅访问的仅访问 OPS_Configuration_File 中指定的表。

OPS_Configuration_File 中的每一个表包含：

- EndSystem_Configuration_Table（（交换机）端系统的配置表）
- Filtering_Policing_and_Forwarding_Configuration_Table（过滤管制与转发配置表）

OPS_Configuration_File 应该是一个现场可加载的软件文件，符合 ARINC 615A 和 ARINC 665 规范。

4.7.3.1 EndSystem_Configuration_Table （（交换机）端系统的配置表）

在 OPS 模式，交换机的端系统功能使用 EndSystem_Configuration_Table，表的选择与该交换机的当前位置有关，目的在于：

- 在接收端，检查接收帧的一致性；
- 在发送端，构建以太网帧并实现流量控制机制。

4.7.3.2 Filtering_Policing_and_Forwarding_Configuration_Table（过滤管制与转发配置表）

在操作模式（OPS），过滤与转发功能依赖于包含在 Filtering_Policing_and_Forwarding_Configuration_Table 中的参数，表的选择与交换机当前的位置有关。

Filtering_Policing_and_Forwarding_Configuration_Table 应包含下列参数。

对于每个 VL：

- 输入物理端口
- 输出物理端口列表
- MAC 目的地址（VL 标识符，VLi）
- 带宽分配间隔（BAGi）
- 最大允许抖动
- ACcount（ACi）[注明这个是否账户是否被共享]
- 最大允许以太网线路总长度（S_{max}）
- 最小允许以太网线路总长度（S_{min}）
- 优先级

对于每一个端口：

- 最大延迟
- 端口状态（ON/OFF（开/关））
- 物理端口的速度
- 对于低优先级 VL 的输出缓冲区大小
- 对于高优先级 VL 的输出缓冲区大小

这些参数是为了配置交换机的过滤、管制与转发功能所需的。

4.8 操作模式

4.8.1 概述

图 4-9 展示了交换机不同的模式以及这些模式之间互相转换的条件。

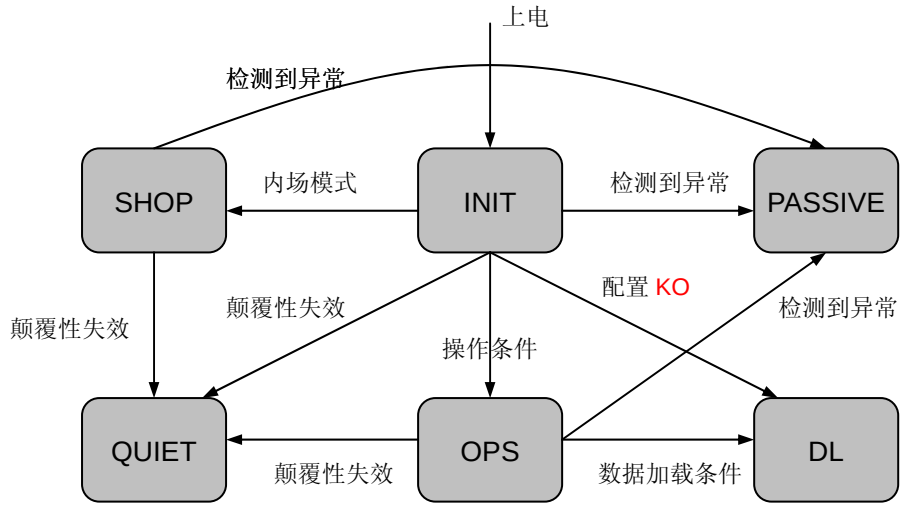


图 4-9 – 交换机操作模式

交换机在上电或复位之后进入 INIT 模式。

OPS 是交换机的操作模式。在这个模式下，依赖于加载的配置表（OPS_Configuration_File）中的信息，交换机执行操作功能：帧和流量过滤，交换功能、监视功能等。

DL 是交换机仅集中进行自身的数据加载时模式（特别是上载操作）。

可选的 SHOP 模式被定义，并能够被用来执行脱离飞机的调试。SHOP 的条件应该在一种 ARINC 700 规范中被定义。

在 INIT 模式时检测到异常将导致激活 PASSIVE 模式。在 PASSIVE 模式中，交换机仅提供与网络管理功能通信的能力，所有其它功能都被停止。

QUIET 模式：当发生颠覆性失效事件，以至于使交换机的行为发生问题，进入该模式。

复位动作允许从任何模式返回 INIT 模式。

注释

复位的能力是可选的，并且能够在某个 ARINC 标准中被定义，或是在系统集成者附加的规范中被定义。

4.8.2 INIT

INIT 是交换机上电或复位之后进入的默认模式。在这个模式下，交换机根据它的当时的情景准备并选择它的后续模式。

在一次手动复位动作之后，或者当电源中断的中断时间长于透明时间（Transparency Time）或电源维持（Power Hold Up）时间[4]，在紧接着这次中断的上电之后，交换机应该系统性地进入 INIT 模式。

注释

系统集成者或 ARINC 标准应该定义透明时间/电源维持时间的值。

4.8.2.1 初始化顺序

在 INIT MODE（INIT 模式），交换机执行的初始化顺序如图 4-10 所述。

[4] 维持时间，关机后电源保持延迟的时间——译者注。

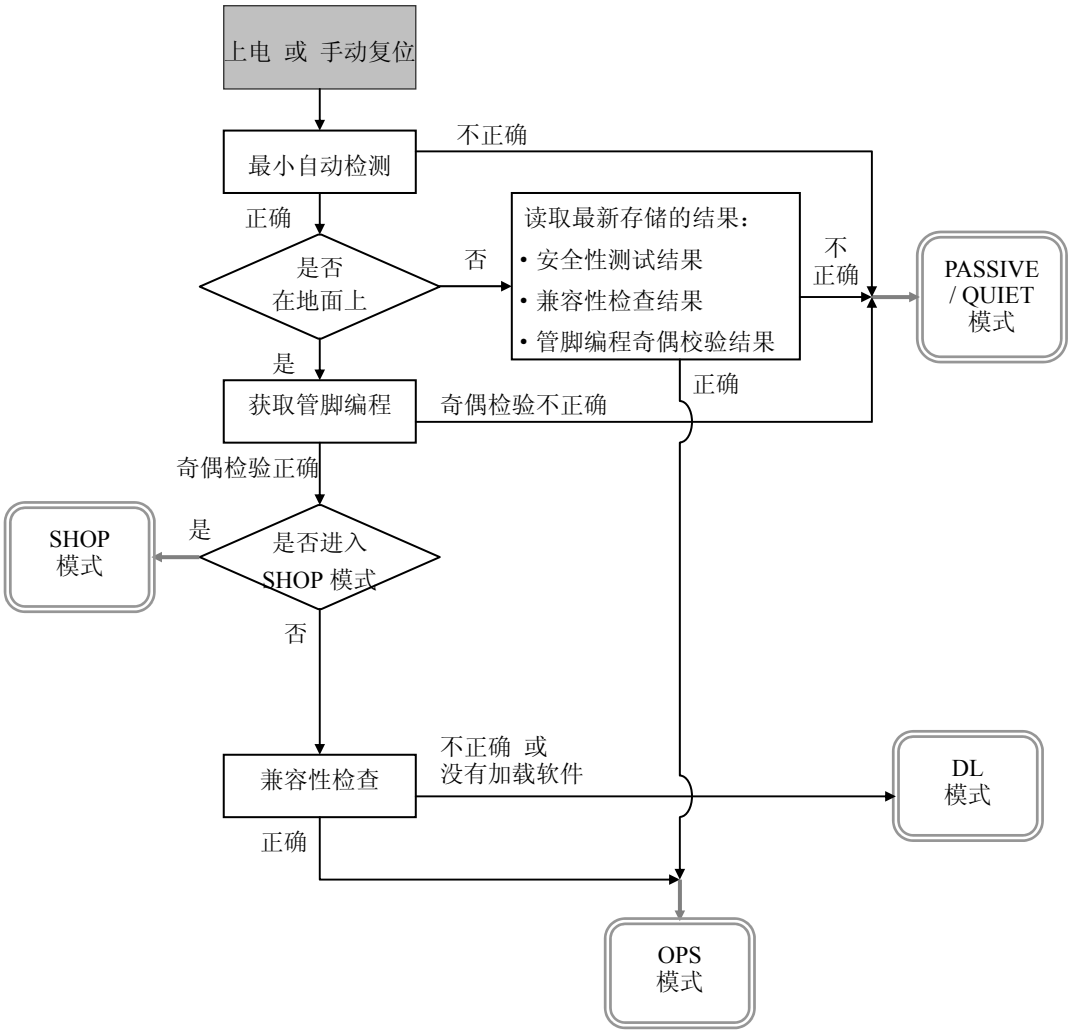


图 4-10 – 典型的交换机初始化顺序

交换机应该先于它们的“订户”（subscribers）（即：所连接的一些 LRU）进入可操作状态，以便用户在它们自己的初始化阶段可以通过 AFDX 网络通信。因此，从上电到进入全 OPS 模式（所有的 OPS 功能都运行起来）的初始化顺序过程执行的时间应该小于一个规定的初始化时间。

注释

系统的集成者或 ARINC 标准应该定义在 Switch_Ground_Condition 为有效和无效的条件下的这个初始化时间。

在 INIT 模式下，故障/健康指示应该停留在故障状态。

4.8.2.2 Ground_Condition

系统集成者或 ARINC 标准应该定义地面条件的值（信号的数目，逻辑条件，确认 等）。

4.8.3 OPS：操作模式

在初始化之后，如果兼容性检查正确并且内场条件无效，交换机进入 OPS 模式（操作模式）。

在 OPS 模式，交换机应该执行它的正常功能（帧过滤与流量管制，交换功能，端系统功能），专门依赖于在 OPS_Configuration_File 文件中被激活部分中所包含的数据。

在 OPS 模式，交换机端系统应该能够执行下列 ARINC 615A 数据加载操作：

- 信息操作[5]、
- FIND（Find Identification of Network Devices，寻找网络设备的标识）请求。

在 OPS 模式，故障/健康指示应该被设置为健康状态。

4.8.4 DL：数据加载模式

为了允许向空白交换机（这意味着没有已加载的软件，例如：为了替换一个失效的交换机而安装的一台空白交换机）上载数据。从 INIT 模式，空白的交换机进入 DL 模式，当且仅当满足下列条件：

- Swith_Ground_Condition 有效，
- 兼容性检查的结果不正确，或是没有已加载的软件。

从 OPS 模式进入 DL 模式，当且仅当使下列条件全部得到满足：

- Switch_Ground_Condition 有效，
- 交换机从数据加载器收到一个 ARINC 615A 上载操作初始化请求（[TH_Uploading_Initialization] 消息[6]），并且目的地址是该交换机自有的端系统。
- 头文件（header file）已经收到并被接受。

从 OPS 模式到 DL 模式的转换不应破坏 ARINC 615A 会话。

注释

模式转换对于 ARINC 615A 透明。

在 DL 模式，交换机应该能够完全执行下列 ARINC 615A 数据加载操作：

- 信息操作、
- 上载操作、
- FIND 请求。

上载是一个必须优先地排除其他操作的特殊操作。在上载操作期间，交换机仅只专用于这件操作。

在 DL 模式，交换机执行 ES 功能应该仅依赖于它的驻留配置表（Default_configuration_table）中包含的数据。

注释

这是为了允许空白交换机的上载（这意味着还没有已加载软件，例如：替换掉一个失效的交换机，安装一台空白交换机之后）。

在 DL 模式，故障/健康指示应该被设置为“健康”状态。

在 DL 模式结束的时候，交换机应该返回到 INIT 模式。DL 模式的结束由交换机内部确定，它与数据加载功能的结束（由 ARINC 615A 规定）有关。

4.8.5 SHOP（可选）

进入 SHOP 模式（内场模式）的条件依赖于实现。

---

[5] 信息操作：在地面维护操作中用来在目标硬件和可加载软件飞机部件的配置中获得信息（即：硬件和软件的标识和部件号）。参见 ARINC 615A-2 报告 5.4.2。

[6] 参见 ARINC 615A-2 报告 6.5.2。

在这种模式下，交换机的一些附加的功能可能会被激活。

ARINC 615A 操作（信息，上载，下载，FIND）可以在 SHOP 模式下可用。

4.8.6 PASSIVE

在 PASSIVE（被动）模式，交换机的功能应停止，交换机应保持静默（silent）。但是，与网络管理功能的通信能力应该被维持，除非导致被动状态的结果是源于管脚编程信号的奇偶校验错误。

注释

如果管脚编程不正确，交换机不知道自身的 MAC 地址，不能与网络管理功能交换信息。

默认的 VL（在本文中被称为 $VL_{(0, position)}$和 $VL_{(1, position)}$）用于与网络管理功能通信。

注释

这将是用于数据加载的同一个 VL。注意到为了在这个模式下使用 SNMP 和 ICMP 协议，在航空电子网络中，有必要由同一个 LRU 作为发起 SNMP、ICMP 和数据加载的请求源。

从 INIT 模式，只要下列任一个条件成立，交换机进入 PASSIVE 模式：

- 管脚编程奇偶校验失败，
- 安全性测试（如果有的话）的结果不正确。

注释

当任何一个上述的条件成立，交换机不应该进入 OPS 模式。同样，参见 4.10.1.2 节 “ 奇偶校验和处理 ”，对于管脚编程奇偶校验的定义。

当进入 PASSIVE 模式，交换机应该将故障/健康指示设为“故障”状态。

4.8.7 QUIET（寂静模式）

在 QUIET 模式，交换机的所有功能被停止，交换机不能发送帧，也不允许与交换管理功能（即：SNMP）通信。当进入到 QUIET 模式之后，交换机软件禁用所有端口。

在进入 QUIET 模式时，交换机将故障/健康指示设置为“故障”状态。

当发生任何种类的颠覆性失效事件，以至于使交换机的行为有问题（的条件下），进入该模式。下面是这些类型失效的例子：

- 管脚编程奇偶校验失败，
- 心跳（信号）（Heart Beat）监视测试失败[7]，
- 引导代码（Boot Code）CRC 校验失败，
- 处理器指令（Processor Instruction）测试失败，
- 地址/数据总线监视失败，
- RAM 存储器监视失败，
- Built-In-Self-Test（BIST，内建自测试）失败，
- 硬件注册测试失败，
- 默认 OPS 软件 CRC 校验失败，
- 默认配置文件 CRC 校验失败。

[7] 原文为“管脚编程 CRC 校验失败”，根据上下文更改——译者注。

这些类型的失效都表明一些严重的硬件问题。这些类型的失效具有潜在地导致交换机的部分区域产生不稳定的和不可预测的行为，该交换机不可被信任。寂静模式确保交换机将不转发帧，并将保持寂静（不试图生成消息）。

4.9 数据加载

4.9.1 数据加载的一般要求

应该使用 ARINC 615A 和 ARINC 665 报告作为将软件的配置表上载入交换机的过程与协议的指导。

4.9.2 配置识别

4.9.2.1 交换机配置的定义

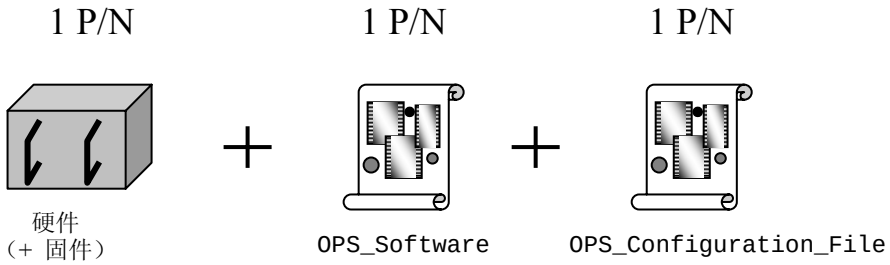


图 4-11 – 交换机配置定义

每个交换机至多由三个部件号（P/N）确定，如下所示：

• 它的硬件 P/N，包含驻留软件（固件）。驻留软件不能在飞机上加载。它只能在修理车间（内场）中经过特定的程序才能被改变。

• 两个现场可加载软件 P/N：

- OPS_Software：现场可加载操作软件，
- OPS_Configuration_File：交换机的现场可加载配置表。

交换机不应具有多于两个现场可加载软件（OPS_Software 和 OPS_Configuration_File）。这些现场可加载应该对于航空电子数据通信网络[8]中的所有交换机都是同样的。

注释

OPS_Software 和 OPS_Configuration_File 可以在一次加载中组合进行。

4.9.2.2 交换机上电配置识别

当交换机上电后，它的配置信息应该通过 ARINC 615A“信息操作”（Information Opertation）的方法访问。

4.9.3 数据加载器 IP 地址

交换机依靠读取数据加载器的源 IP 地址学习得到数据加载器的 IP 地址。

4.10 管脚编程

交换机硬件管脚编程应该被用来确定交换机在网络中的位置，并且与交换机的端系统的默认 MAC

[8] 此处原文为“A/C”，联系到上文（图 4-5 和 4.7.3 小节），认为这是“航空电子（A）数据通信（C）网络”的缩写——译者注。

和 IP 地址相关联。

4.10.1 管脚编程过程

4.10.1.1 读取管脚编程的恰当的时机

硬件编程管脚应该在交换机初始化的时候，并且仅在 Switch_Ground_Condition 有效的情况和安全性测试（如果有安全性测试）之前进行。

若 Switch_Ground_Condition 条件为假，硬件编程管脚不应被读取。交换机应该使用最后一次存储的数值。

注释

这保证在飞行中即使管脚编程出错不会有操作性的后果。

4.10.1.2 奇偶校验与处理

12 位编程管脚（P_1 到 P_{12}）应该具有奇偶校验二进制位。

为了在飞行条件下掉电时使用，当管脚被读取并且通过校验，编程管脚的值应该被存储在非易失存储器中。

4.10.2 管脚编程的列表

4.10.2.1 位置编码

交换机应该通过 12 位管脚编程（命名为 P_1 到 P_{12}）得到位置信息。在管脚编程中，GROUND（接地）值应该被编码为 1，OPEN（开路）值应该被编码为 0。

这样，遵守本文档的 4.5.2 节“寻址策略”（将被提供）。

4.11 性能特征

4.11.1 通用特性

交换机的过滤、管制与转发功能应该能够处理至少 4096 条 VL。

注释

这与 4096 个不同的目的 MAC 地址有关。

注释

系统集成者或 ARINC 标准应该定义交换机的物理端口的数目。

4.11.2 物理层特性

作为一个网络组件，交换机必须遵循这种网络的一些基本的特性。在这些特性中有物理链路速率，或者是介质访问的模式。下面列出了适用的需求。

每个端口的速率应该被配置于 10Mbps 或 100Mbps，全双工操作，无自适应速率调整。每个端口的速率应该在配置表中被定义。

物理层应该符合 ARINC 664 part 2 规范。

4.11.3 处理能力

给定一个确定性的配置（确定性配置被定义为没有任何输出端口饱和），交换机必须能够处理（即：接收，过滤，管制，中继，发送）以线速输入的所有的帧，不论这些帧的长度如何。

注释

这考虑到帧头部处理和帧交换（对应于短帧）的最大处理性能，以及充足的存储器（对于缓存长帧）。

前面的需求隐含地提出输入端口的公平处理：应该不存在某个输入端口具有高于其他输入端口的优先级的情况。

每个交换机输入端口的 MAC 层应能够以线速接收帧。

注释

线速的定义是：不考虑帧的大小，当以 12 个字节的最小的帧间间隔（Interframe Gap, IFG）接收帧的时候，能够得到的最大的帧速率。

这与最大的帧速率有关：64 字节（帧）+12 字节（IFG）+7 字节（前导字）+1 字节（起始定界符，SFD）=84 字节。当在 100Mbit/s 的速率下进行操作，这相当于 6.72μs 每帧（大约 148800 帧每秒）。

每个交换机端口应该能够处理以线速到达的任何有效长度的帧。

给定一个确定性配置，对于所有的不被过滤与管制功能拒绝的输入帧，交换机应该能够以线速对它们进行中继，而不论它们的长度。

注释

确定性配置被定义为没有任何输出端口饱和。

每个交换机的输出端口应该能够以线速发出帧。

交换机的技术时延（technological latency）应该小于 100μs。

注释

时延被定义为从接收到帧中最后的二进制位到发送这个帧的最后的二进制位之间所经历的时间。交换机时延[9]由三部分组成：交换机功能的技术时延，由于交换机加载的配置时延，以及帧在介质上传送所需的时间。

所以，技术时延是对“确定性”的网络配置有贡献的参数。

缓冲存储解决数据争用输出端口的问题。每个交换机的输出端口应该能够缓冲至少 512 个帧（在高和低优先级间平衡）。

注释

需求对每个输出端口设定一个特定的最小存储容量。其他缓冲区的约束（输入缓冲存储区的大小等）可以由交换机的设计得到。

[9] 这条注释并没有说明交换机的工作状态，按照常识，在工作状态交换机的时延应该包括技术时延（包含：帧过滤、解多路复用）、帧传送时延和输出多路复用排队延迟；故此时是否包括“交换机加载的配置时延”，暂存疑——译者注。

5 .0 系统问题

5.1 性能

性能被定义为 ES 能够操作的（性能指标上限的）最大的百分比。最大吞吐率（“线速度”）对应一帧接一帧传输的情况。

实际性能是通过测量处理 1ms 时间段内一帧接一帧紧接着突发传输的帧所必需的时间来计算。

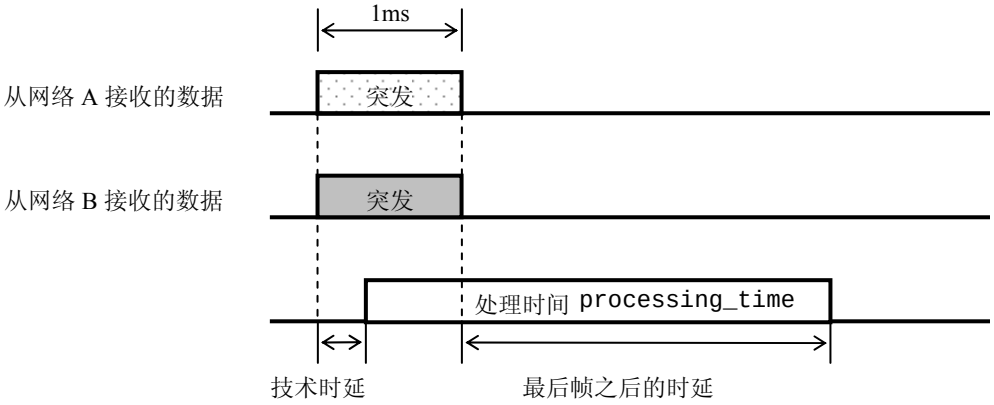


图 5-1 – 1ms 时间段内一帧接一帧传输的突发通信

冗余管理算法被关闭。假设来自网络 A 和网络 B 的数据属于不同的 VL，都被提供给相应的应用程序。

性能（占“线速度”的百分比）按照下面的式子定义：

$$\text{Performance}=100\times\frac{1}{\text{last\_frame\_latency}-\text{technological\_latency}+1}=100\times\frac{1}{\text{processing\_time}}$$

其中“last_frame_latency”和“technological_latency”以毫秒（ms）为单位表示。

其中，“last_frame_latency”对应被接收的 1ms 突发通信量的最后一帧，不论这个帧来自网络 B 还是网络 A。在接收过程中测量点与这些时延的定义相同。（见图 5-1）

接收性能可能依赖于几个参数，例如帧长度、队列或采样数据，ES 发送活动等。

端系统设计者应该提供 ES 发送和接收处理能力的信息。作为指导，应提供下列固定的参数：

发送和接收的容量：

- 端口数目、
- VL 数目、
- Sub-VL 数目、
- 帧长度、
- 每条 VL 的 IP 多播组大小；

发送速度：

- 时延、
- 帧速率；

接收速度：

- 时延、
- 流量剖面。

本节开始的部分已经介绍了测试接收性能的方法。

依赖于端系统的设计，性能特征可能随不同的实际配置而不同（如：VL 数目，是否对数据包进行分片操作）。

端系统设计者应该提供关于 ES 接收部分存储容量的信息。

附件 1 数据格式

1-1.0 引言

这份附件提供用于航空电子领域AFDX飞机数据网络的格式化消息的需求规格。这些消息都是在应用层（即：OSI模型的应用层）被创建和接收的，不包括对于应用层的协议的消息格式，例如TFTP和ARINC 611。这里出现的格式都是被限制于不基于协议的数据。

这份规格说明的用意在于依照消息在网络介质上出现的形式定义消息格式。一些具体实现可能使用软件中间层执行格式解析和翻译；这样可能导致应用任务采用超出本文规定的不同格式的消息和数据基本类型（即：数据原语，data primitives）。应用任务采用的实际格式，并不在本规格说明的范围之内。本规格说明只用以处理如下过程中的数据表示方法，即：作为传输层的有效载荷从数据的“生产者”产生，经过网络介质，向上到达数据的“消费者”——传输层，作为该层的有效载荷。

数据格式化方法的说明包括以下几个小节：

1. 引言
2. 基本数据元素的格式（布尔型，整型，浮点型，字符串型等）
3. AFDX消息结构的描述
4. 格式举例：飞机参数到网络中消息的分组（grouping）格式

1-1.1 以太网二进制位/字节的次序

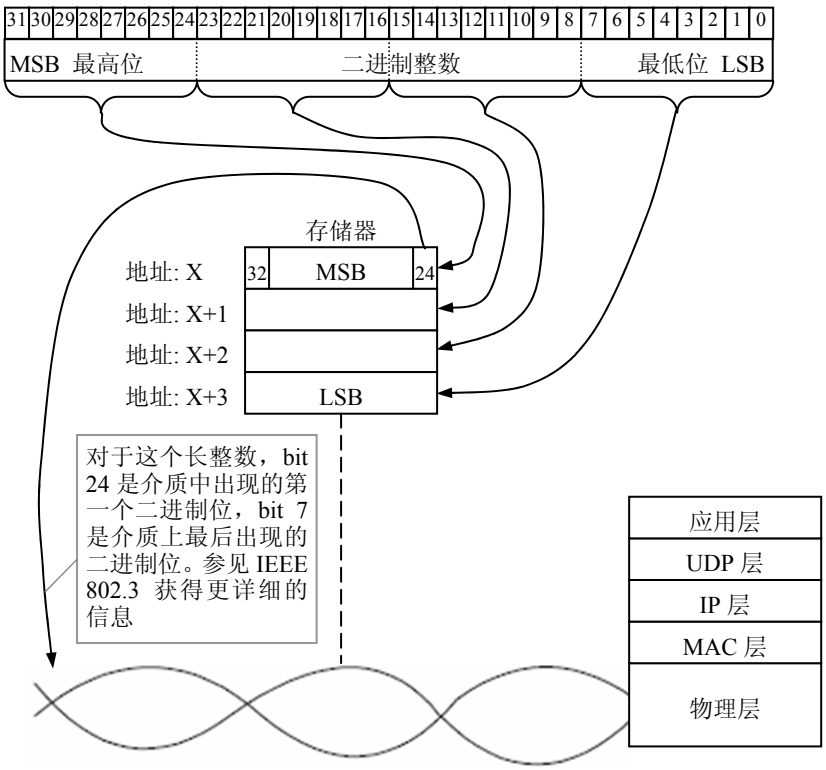


图 1-1.0 以太网在介质中二进制位和字节的次序符合大端（Big Endian）格式

飞机数据网络对于字节的次序，与互联网标准相同，规定整数的最高有效字节先被传输（即：大端（Big Endian）风格）。如果数据包被从一台机器传输到另一台机器，考虑其中依次相继的字节，对于这个数据包中一个二进制的整数，它的最高有效字节最靠近数据包的开始，它的最低有效字节最靠近数据包的结尾。

以太网是按照大端类型定义的，这意味着在存储器中数据的表示方法是：以较高的二进制位段数字和字节在先（即：低地址值），随后字节中包含较低的二进制位数字。图1-1.0说明在以太网接口的缓冲存储器中基本的数据结构是怎样被表示的。它还展示在介质中数据是怎样排列的。

1-1.2 抽象和传递句法

在数据层次结构最低的一级起始的是数据单元。数据单元，或是数据原语，是飞机数据网络中的可识别的最小的数据片段。数据单元是一个特定的值，被表达一段离散的信息的范围和解析度所定义。在讨论这些数据单元的格式之前，应当先介绍抽象和传递句法的概念。

1-1.2.1 抽象句法

抽象句法（abstract syntax）就是抽象于底层的计算平台与网络。这个抽象的表示法是应用程序使用的格式。一个抽象的句法将允许实现者以一类可以被人读懂的高级定义语言描述消息。这个定义将展示如何用一系列的数据原语组成每个消息，而每个数据将具有一个可以被接收的取值范围。对一个设计者来说，因为对于一个真实的消息存在近乎无限种可能的单元值的组合形式，试图将所有可能的消息显式地的列出来是不现实的。与此类似，以太网的消息长度之大以至于存在近乎无限种不同的可能出现的消息。幸运的是，一些可用的工具被用来帮助处理这个问题。存在高级定义语言帮助定义消息格式（定义在消息中有哪些数据原语，它们以何种次序在消息中出现，以及何种取值范围是可接受的）。这样一些语言具有翻译器（编译器），后者能够自动地生成表示这种消息格式的程序语言的数据结构。

1-1.2.2 传递句法

传递句法（transfer syntax），或者被称为物理句法（physical syntax），是消息通过网络时的格式。由抽象句法定义的数据原语的组合构成的完整消息，由应用程序发送出去。底层的平台（计算机和网络接口）可能将消息翻译成为传递句法。消息通过网络转发到一个或多个目的计算机，在那里另外的翻译工作被执行，使得消息被送入应用程序之前，从传递句法翻译到抽象句法。

注释

OSI 模型中的表示层（presentation layer，第 6 层）被用于执行这种类型的服务。一种可能的实现是在软件中包含属于这一层的一个薄层，执行消息的编码和解码。这可以作为在抽象句法和传递句法之间的翻译服务。一种备选的方案可以是与应用层程序连接的编\解码器的代码。这些代码可以使用工具创建，使得这项工作相当自动化。

系统设计者对传递句法的选择将取决于系统需求。对于一个系统，传递效率可能是最优先考虑的因素，编码的简易程度可能是另一个被最优先考虑的因素，也许可靠性是另外的被优先考虑的因素。对于一个系统，如果编码解码的简易程度具有关键的重要性，传递句法应该能够被选择为与抽象句法相同。

考虑一个枚举类型参数，例如，假定该枚举量作为某系统的模式指示器，其定义如图1-1.1所示。这个枚举类型参数应该能够作为一个抽象句法的定义。编码原则可以将其编码成32位的有符号的整数，属于一种被允许的传递句法数据原语之一。

处理器模式枚举元素规范定义		
标签：	值：	标记
	0	关（Off）
	1	暂停（Halt）
	2	无效自陷（Invalid Strapping）
	3	数据加载（Dataload）
	6	无效平台配置（Invalid Platform Configuration）
	7	软件确认（Software Validation）
	13	IBIT（内部自检）
	18	正常（Normal）

图 1-1.1 - 枚举型定义举例

注释

假设一个飞机上装备了一个网络和计算机系统，除了大气数据传感器（Air Data Sensor，ADS）之外，所有的 LRU/LRM 都装备了以太网接口。假设 ADS 是 ARINC 429 接口，提供系统一个标签值(label)，标记为空速。整个系统的传递句法，除了这个空速的消息之外，有可能与抽象句法是相同的。为了这个 32-bit 的标签值，这个系统要开发一个传输句法，在每一个接收端使用语法解析器代码将它翻译成抽象句法。使用这个空速参数的应用程序用到这个传递句法，所以，当在未来将基于 ARINC 429 的 ADS 升级成为基于以太网的 ADS 的时候，所必须作的所有工作是运行一些工具（去掉语法解析器代码），以及某些分区的数据加载。应用程序不需要更改，这种方法使应用程序成熟度更高（体现于改动较少）和可移植。

1-2 .0 原语数据元素

每一个数据元素是一个预定义的独立于网络的数据类型，基于一种中性的接口定义语言（Interface Definition Language, IDL）。这些数据类型具有周知的二进制位和字节次序，使得应用程序能够通过网络来发送和接收数据元素，而不用考虑系统中使用的主机处理器的原始的二进制位和字节顺序。表1-1列出[10]了当前在飞机数据网络中定义的数据元素类型。

表 1-1 – 原语数据类型

数据元素类型	长度（单位：bit）	取值范围
Singned_32_Integer （有符号长整型）	32	$-2^{31} \ldots 2^{31}-1$
Singned_64_Integer （有符号双倍长度整型）	64	$-2^{63} \ldots 2^{63}-1$
Float_32 （单精度浮点型）	32	$\pm(2\cdots\frac{1}{2^{22}})\cdot 2^{-126\cdots127}$
Float_64 （双精度浮点型）	64	$\pm(2\cdots\frac{1}{2^{51}})\cdot 2^{-1022\cdots1023}$
Boolean （布尔型）	32	N/A （不采用）
String （字符串）	根据数据元素的定义而不同（参见本文1-2.5有关段落）	N/A （不采用）
Opaque Data （非透明数据）	根据数据元素的定义而不同（参见本文1-2.6有关段落）	N/A （不采用）

[10] 原文中为“表 1-1.0”，参照本附件中的表的编号规则，改为“表 1-1”，下同——译者注。

注释

在 AFDX 网络里只允许使用表 1-1 中所列出来的数据原语。当工作环境要求必须使用除了这些列出的类型之外的数据结构的时候，应该将其放在非透明数据（Opaque Data）类型里面。然而，要尽可能避免使用非透明数据类型。AFDX 消息要尽可能由经过定义的原语组成。

这个数据原语的小集合使消息格式化简单。而保持格式化简单可以减少消息的复杂性，并减少需要编码和解码许多不同数据类型的软件数量。为了获得这种简单化，会丧失一些效率。将一个可以用8个二进制位表示的整数用32个二进制位存储并不是最有效的方法。但是将所有的整数存储为32个二进制位却对计算有利。相比于以前技术来说，以太网具有很宽的带宽，消息的长度能够相当长。

下面的段落将给出网络中出现的每个原语数据元素的格式。

注释

整数和浮点数据类型在 AFDX 网络消息中是从不缩放的，尺度因子总是 1，这不同于在 ARINC 429 中，参数都有各自不同的尺度因子，通常都不是 1。

1-2.1 有符号长整型——Signed_32

这个32位的有符号整数用了2的补码形式表示数据。

图 1-2.1 有符号 32 位整数表示法

注释

当将 8 位或 16 位的整数转换成标准的 32 位 2 的补码形式长整数型的时候，如果短整数是一个有符号的值，应该进行符号扩展；如果短整数是一个无符号数，应该进行零扩展。

1-2.2 有符号双倍长度整型——Signed_64

这个32位的有符号整数用了2的补码形式表示数据，如图1-2.2所示。

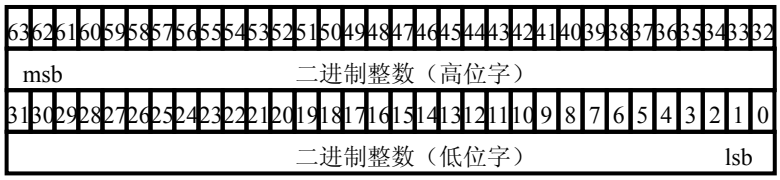


图 1-2.2 有符号 64 位整数表示法

1-2.3 浮点型

浮点型数字使用IEEE 754规定的格式。细节详见于该标准，这里给出一些方便应用的信息。

1-2.3.1 标准精度浮点型—— Float_32 - IEEE754

单精度浮点型数长度是32个二进制位，具有3个区域，它们是：

1. S：符号位－0代表正数，1代表负数；

2. E：指数区域－二进制数，以2为底的幂指数部分，8个二进制位；
3. F：尾数区域－ 有效数字的小数部分，二进制数，以23个二进制位表示。
下面的公式给出标准格式数字的实数取值：$(-1)^S \times (1.F) \times 2^E$；
下面的公式给出非标准格式数字的实数取值：$(-1)^S \times (0.F) \times 2^E$。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
S	指数								尾数																						

图 1-2.3.1 32 位浮点数表示法

1-2.3.2 双精度浮点型——Float_64－IEEE754

双精度浮点型数长度是64个二进制位，具有3个区域，它们是：
1. S：符号位－0代表正数，1代表负数；
2. E：指数区域－二进制数，以2为底的幂指数部分，11个二进制位；
3. F：尾数区域－ 有效数字小数部分，二进制数，以52个二进制位表示。
下面的公式给出标准格式数字的实数取值：$(-1)^S \times (1.F) \times 2^E$ ；
下面的公式给出非标准格式数字的实数取值：$(-1)^S \times (0.F) \times 2^E$ 。

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
S	msb				指数						lsb	msb							尾数												
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
尾数																														lsb	

图 1-2.3.2 64 位浮点数表示法

1-2.4 布尔型

1-2.4.1 标准布尔型

布尔数是在一个32-bit的域中用单个的二进制位表示。如果用这个32-bit的位域只代表一个布尔量，则使用bit 0（最低位），如果bit 0是逻辑“1”，则该布尔量被称为“真”（或者代表 是、激活，开 等含义），如果bit 0是逻辑“0”，布尔量的值被称为“假”（或者代表否、去活、关 等含义）。

1-2.4.2 逐位打包布尔型（Bin-Wise Packed Boolean）

布尔量可能被打包成32-bit的结构（如图1-2.4.2所示）。每一位代表各自的布尔量实体，一个单独的结构可以代表多至32个布尔实体，如果需要表示第33个布尔量，则一个新的32-bit结构被加入来保存第33位布尔量，布尔量在一个结构里面被分配从最低有效位往上填充至最高有效位，在32-bit结构中没有用到的高位端必须是全0。

bit 31	bit 30	bit 29	bit 28	— — — -	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	0	0	第 29 个布尔量	— — — -	第 7 个布尔量	第 6 个布尔量	第 5 个布尔量	第 4 个布尔量	第 3 个布尔量	第 2 个布尔量	第 1 个布尔量

图 1-2.4.2 逐位打包布尔数（29 个布尔数）

1-2.5 字符串

字符串被定义为是*n*个ASCII码字符的序列（如图1-2.5.1所示）。每个字节是一个ASCII码字符，它们被从1到*n*的编号，在字符串结构的前2字节是一个16-bit的无符号整数，这个值是字符串的长度（字符串里字符的个数）。

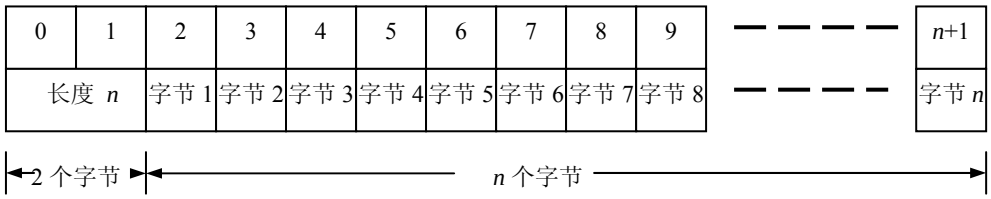


图 1-2.5.1 字符串数据结构格式

字符串结构是固定尺寸的，数据结构由3个部分组成：长度域，数据域和填充域。长度域中的值定义数据区占据的字节数，填充区在剩余的地方填充0以填充满整个结构，见图1-2.5.2。

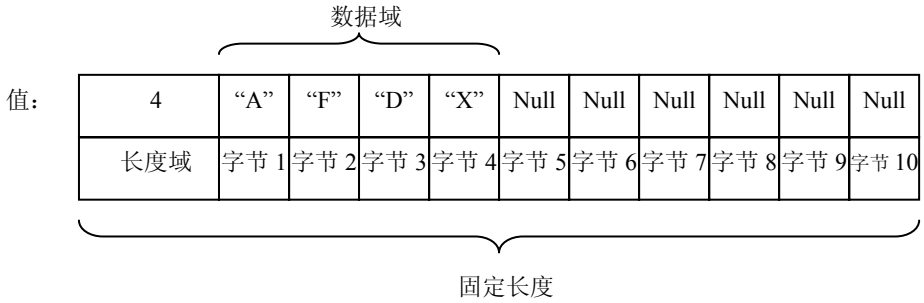


图 1-2.5.2 字符串数据结构

1-2.6 非透明（Opaque）数据

对于表1-1中没有列出或没有给出的数据类型或数据结构，用非透明数据原语来存储具有这些类型或结构的数据。数据格式可能与航空电子无关，数据域可能是可选的，或是可变的。可能这种数据是不可描述的。这类数据被称之为“Opaque（非透明）”。在一个非透明数据结构中所有用不到的字节都应该被填充为二进制0。

1-2.6.1 固定长度非透明数据

固定长度非透明数据被定义n字节的序列，字节从1到*n*排号。参见图1-2.6.1.1。

图 1-2.6.1.1 固定长度 Opaque 数据结构

1-2.6.2　可变长度非透明数据

可变长度非透明数据的前面定义了一个半字（16 bit）长度的区域，后面接着*n*个字节的序列，从1到*n*排号。如图1-2.6.2.1。长度区域被编码为16-bit无符号整数。在序列中的字节*m*（序列中任意一个字节）总是跟在字节*m*－1后，而序列中第1个字节总是跟在长度域的后面。

图 1-2.6.2.1 可变长度非透明数据结构

这个结构允许在不知道数据细节的情况下对其进行存储。尽管数据元素可能不会用完所有被定义的存储区域，长度域可以告知存储在结构内的数据元素的长度。长度*n*代表的意思是数据元素的真实长度，而不一定是存储区域的最大长度。

可变长度非透明数据结构的大小是固定，数据结构包括3个部分：长度域，数据域，填充域。长度域的数字定义数据区占用的字节数，填充区在剩余的地方填充二进制0，以填满整个结构大小。参见图1-2.6.2.2。

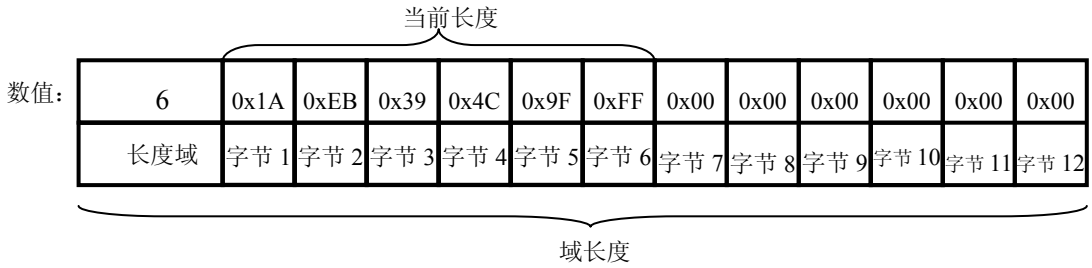


图 1-2.6.2.2 可变长度非透明数据

1-3 .0　消息结构

这个部分涉及消息的结构，以OSI模型的视角，这是应用层的内容。这些消息是网络发布的有效载荷。

1-3.1　隐式和显式消息/端口号

能够将消息结构定义为隐式（Implicit）或显式（Explicit）。显式消息结构包括格式信息，它允许消息的接收者对消息解码。格式信息是通信开销，它可能包括标识符（用以指出在消息中被编码的是什么样的数据结构），并且/或者包括长度参数以指出结构的长度。

隐式消息并没有相关的格式开销。消息中只包括数据参数，并没有解释消息内容的识别方法，隐式消息更有效率地利用网络带宽。

这里专门（exclusively）定义的消息结构是隐式的，应用程序通过接收数据的端口号得知消息的结构。这种实用的作法与Internet中广泛接受的概念“周知服务”（WKS）是相符的。例如，作为周知服

务的简单文件传输协议（Trivial File Transfer Protocol, TFTP）在端口69上可用。在综合化过程中，飞机网络里的每一个消息格式都被指派端口号和WKS名，消息通过它的周知服务名被引用。

在典型的AFDX网络实现中使用两种类型的端口号：

1. 传输层（TCP或UDP）端口号、

2. ARINC 653通信端口（comport）。

ARINC 653通信端口的定义不在本规范的范围之内，所以，这里除了典型地说明它们的值被任意选择，并且出于简易的考虑可以被一对一映射到传输层的端口号之外，不作其他的讨论。

传输层UDP和TCP端口号被Internet编号分配机构（Internet Assigned Numbers Authority，IANA）所管理分配。这些编号被分为三组：

1. 指派端口号：0－1023；

2. 注册端口号：1024－49151；

3. 动态/私有端口号：49152－65535。

AFDX网络是静态配置的封闭网络，所以，（按照Internet的管理进行）传输层端口分配并不重要，可以由系统集成者使用从1到65535的任意值作为端口号。对于那些存在于网络中的服务（例如：端口69对应TFTP，端口80对于HTTP）可以使用周知的服务端口号，这样可以减少混淆，也可以利用熟悉商业网络的团队成员的经验。

因为AFDX网络的封闭性，在飞机上与其他网络通信是通过上层互连设备。这个装置可以是第三层网关路由器。如果使用网关，它将提供地址翻译，直到传输层（含传输层）的寻址。这就意味着网关将与AFDX网络使用预定义的端口号通信（从网关到AFDX，或从AFDX到网关）。这些预定义的号码能够被仔细地选择以确保相邻的网络间没有传输层寻址冲突。

考虑网络到网络（network-to-network）的通信可能在三层路由器中实现的情况，并且在这种情况下将不存在传输层翻译，系统综合者应该考虑至少避免使用范围是0-1023的传输层端口。对于完全保证避免与相邻网络寻址的的冲突，第二组端口号（从1024到49151）也应该被避免使用。

1-3.2 数据对齐

如果数据被适当对齐，计算机将更有效率地存储和检索数据，数据对齐（data alignment）涉及数据如何在物理存储器里存储。如果数据是对齐的，能够节省可观的处理时间。如果数据元素在消息中对齐，当消息存储在应用程序的存储空间内，应用程序将易于确保它们是对齐的。

如果在存储器中数据元素的地址可以被数据元素的长度（以字节为单位）整除，则称这个数据元素是对齐的。例如，对于4字节（32-bit）标量值，如果它的地址是4字节的倍数，则被称为是对齐的；对于8字节（64-bit）标量值，如果它的地址是8字节的倍数，则被称为是对齐的；以此类推。对于那些可变长度的数据原语，如一个字符串，因为它的长度可能非常长，期望它的地址被它的长度除尽应该说是不可接受的。在这种类似的情况下，应该利用填充字将这个数据（的起始点）放置在2-byte的边界（即：地址可以被2整除）。

在消息开始处的保留字（Reversed word）应该被对齐到32-bit的边界。在整个消息中的功能状态集（Functional Status Sets）也应该与32-bit的边界对齐。当消息被传递到接收的应用程序，该消息应该被放置到内存中的缓冲区，这样保留字是对齐的。这确保整个消息也是对齐的。

数据对齐的运用导致对填充字段的不时之需。考虑一个存储在0x0008－0x000B地址区域的4字节标量的例子（如图1-3.2.1）。如果一个8字节的标量在它后面存储，将不从0x000C开始存储，因为这将不能得到适当的对齐。它必须从0x0010开始存储；而不使用0x000C到0x000F的存储区域，或者将它们填充。地址0x0010可以被8除尽。这个讨论的例子假设传输层有效载荷（即：消息）的第一个字节是“字节0”。

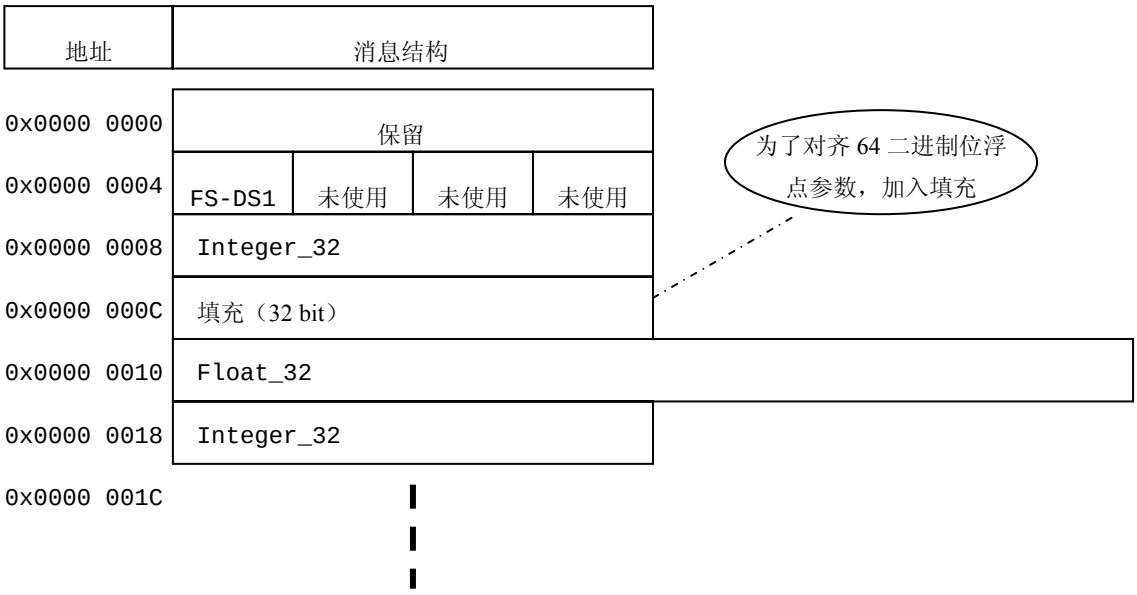


图 1-3.2.1　　数据对齐示例

在图1-3.2.1定义了一个示例消息。它的第4部分是一个32-bit 的填充域。填充是为了数据对齐而被放置到这个结构中，以保证下一个区域（双精度浮点类型，64-bit）所在的地址可以被它的长度整除。

1-3.3 备用和填充

在消息中的备用域是一个当前不被使用的区域，但可能被保留供未来所用。系统集成者可能选择在消息中添加备用域，作为一种手段以备在未来改变的时候控制转换成本。备用域可以在未来被用来发送数据参数，而不会对不使用它们的应用程序产生任何影响。

填充是消息中未被使用的区域，这是特定的计算机或通信需求的结果。最普遍的填充应用是为了数据对齐。

从系统角度，备用域和填充都是消息中未被使用的区域。如果必须在消息中加入新参数，两者都可以被使用。

1-3.4 功能数据集

功能数据集（Functional Data Sets ，FDS）是一种将消息中数据原语聚合成组的办法（数据原语的定义见本文档的1-2.0章节）。数据原语代表飞机的参数或其它数据。所有基于非协议的数据被格式化成为功能数据集以便在飞机数据网络（ADN）中传输。功能数据集（FDS）由两种类型的域构成：

1. 功能状态集（Functional Status Sets, FS）、
2. 数据集（Data Sets, DS）。

FDS的结构如图1-3.4.1所示。功能状态集和数据集组合起来形成功能数据集。

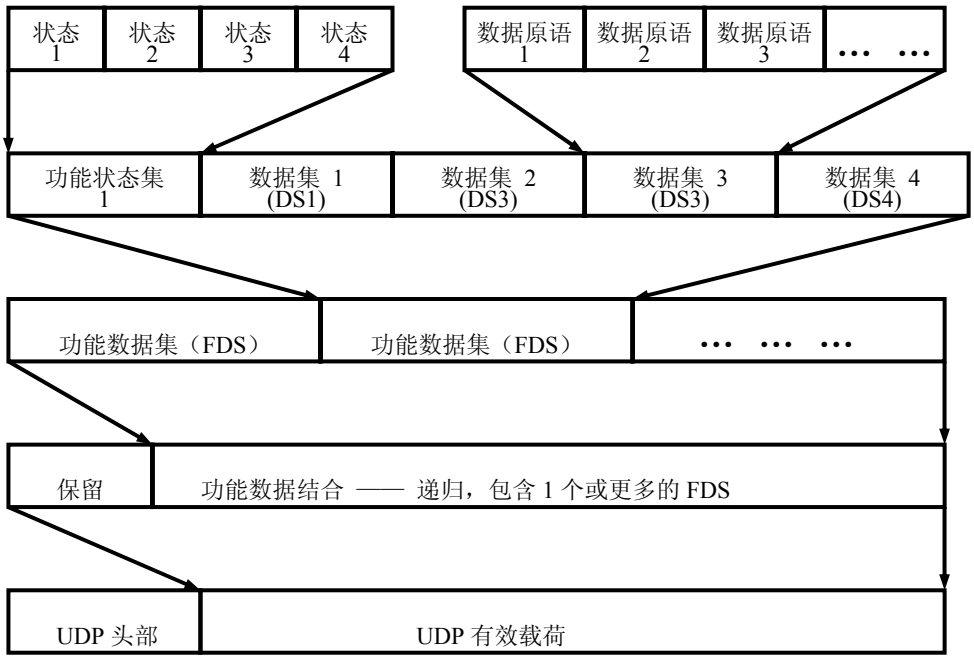


图 1-3.4.1 功能数据单元（FDS）结构

1-3.4.1 功能状态集

功能状态集（Functional Status Set，FSS）是一个由4个8比特状态域组成的32-bit区域（见图1-3.4.2）。第一个状态区域被用来表示第一个数据集（DS）的健康情况和状态。如果在这个FDS中没有其他的DS，余下的24个二进制位必须是0。如果使用其他的DS，每个余下的状态域被用来表示它们的健康情况和状态，直到4个DS全被使用。如果希望用到超过4个的DS，则有一个新的FSS被加到入FDS中，其中跟着1到4个DS，以此类推。FDS的数目没有限制（因此，对于DS的数目也是这样），它们可以被放置到一个单个的消息中，除非受到底层的传输机制（即：有效载荷长度）的限制。

一个DS具有一个或者多个数据原语。在一个DS中可能使用的数据原语的数目没有限制，除非受到底层的传输机制的限制。重要的是，在任意给定的数据集中，所有的数据原语被一个状态域表示。这意味着如果其中一个数据原语无效，全部的数据集不得不被标记为无效。由于这个原因，将源于同一种装置（例如：一个传感器）的数据原语分成一组是很有好处的。如果数据集中的某一条无效，它们可能由于一个装置的故障而全部无效。

考虑一个多模式无线电（Multi-Mode Radio，MMR）的例子，MMR可能具有一个以太网的接口，以允许所有的无线电能够在网络上通信（如：仪表着陆系统ILS，DME系统 等）。每一个无线电接收机的一些数据有可能被包含在一个以太网消息里，但是应该在不同的DS中被分组，以便它们能被不同的FS所表示。如果ILS故障，ILS DS就会被标记为ND（no data，无数据）；而同时如果DME的功能还是正常的，DME DS被标记为NO （normal operation，正常操作）。

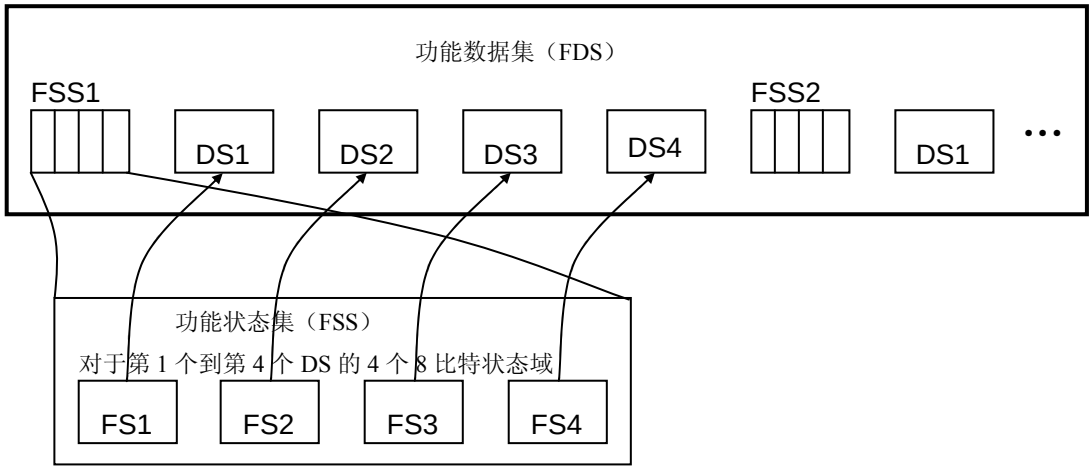


图 1-3.4.2　功能状态集结构

航空电子系统必须知道接收数据的状态。功能状态（FS）字节为每一个FDS中的数据集提供这种指示。FS是一个枚举类型的字节，枚举量的定义如表1-2所示，参见表1-3给出的枚举量的二进制和十进制的表示。

表 1-2 –功能状态枚举量

条件	定义
ND（无数据）	在数据集中没有有效数据，这应该包括失效（Fail）、警告（Warn）和其他一些使内容无意义的条件。
NO（正常操作）	有效数据，正常操作环境
FT（功能测试）	装置测试条件
NCD（非计算数据）	无效数据，装置在正常操作环境，但不能进行可靠数据计算

状态信息是网络中的数据生成提供的。如果DS中的数据原语是不可用的（即使是其中一个数据原语不可用），ND（无数据）状态指示应该被使用。当为一个DS计算功能状态的时候，可能有多于一个的状态存在。在这种情况下，最高优先权的状态就应该被编码入状态域。优先权1为最高优先权，优先权4为最低。例如，一个DS可能由10个飞机参数组成，假设这些参数中的第九个是NO（正常操作），而第十个为NCD（非计算数据）。在这种情况下，这个DS的FS应该被设为NCD，因为NCD具有较高的优先权。

这里的功能状态定义与ARINC 429的符号状态矩阵（SSM）状态定义具有合理的映射关系。映射一个ARINC 429 的SSM “错误警告”条件到ARINC 664使用ND（无数据）状态。如果接收方读到的功能状态值不是（十进制）0，3，12或48（参见表1-3），它将丢弃在相关的DS中数据。

表 1-3 –功能状态枚举量的值

状态	2 进制数	10 进制数	优先级
ND（无数据）	00000000b	0	1
NO（正常操作）	00000011b	3	4
FT（功能测试）	00001100b	12	3
NCD（非计算数据）	00110000b	48	2

1-3.4.2　数据集

数据集域是一组（可以是一个，也可以是多个）数据原语，这些数据原语组成一个数据集（DS）。

如果有未来发展的需要，或者是具有将来添加参数的风险，也能够在数据集（DS）的结尾部分包含一个数据备用域。使用备用域能够在不得不在DS中添加数据的时候通过减小对应用开发者的影响而减少转换的花费。设置数据备用是系统设计者的责任。应该被仔细地使用以节省网络带宽。备用数据域只能被放置在DS的结尾。只要填充域的大小不被改变，该填充域能够被未来的数据使用。参见图1-3.4.2.1的DS的例子。

在数据集域的数据编码的定义如在本附件的1-2.0章节所示。数据集是一组数据原语结构。在数据集中数据原语一个挨着一个放置，在两个原语之间除了为了对齐的填充区域之外没有未被使用的字节。备用数据域如果不被使用，应该以二进制0填充。

注释

在图 1-3.4.2.1 的消息的填充区是在消息中间放置大的数据原语的结果。当把最大的数据原语放在消息的最前端时，填充的需要能够被减小。然而，这样的填充域能够被用作备用区域，所以具有填充区的数值有一些潜在价值。

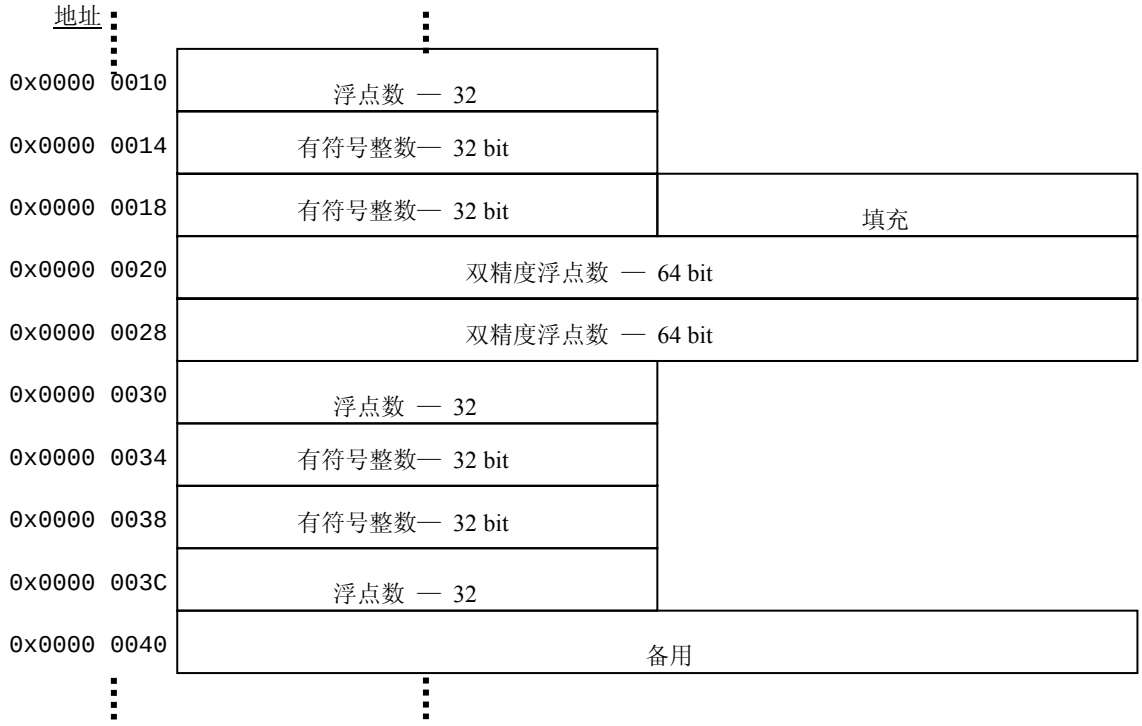


图 1-3.4.2.1 数据集的例子

1-3.5 整体消息结构

现在我们需要将所有这些数据结构放在一起组成一个完整的消息。一个完整消息的格式如图1-3.5.1所示。

整体消息结构的第一个区域是32-bit的数据域，即“未来使用预留”区域。这个域后面跟着一个或多个FDS，如前面的章节所述，一个FDS由一个FS和DS组成。32-bit的保留域应该被填充二进制0。这个区域不被使用，它被包含在消息中是出于对未来发展的预留。

消息中最后的区域是一个可选的整体备用（Global Spare）域，由系统设计者慎重使用，这个域可能具有足够大的空间，足以容纳一个新的FDS，这个被考虑到的FDS应该很可能在未来成为必要的消息内容。在早期为了发展的需要而留下备用域，具有将今后几年内的改变带来的影响减小到最小的潜力。如果消息中包含这个域，它应该全部被二进制0填充。

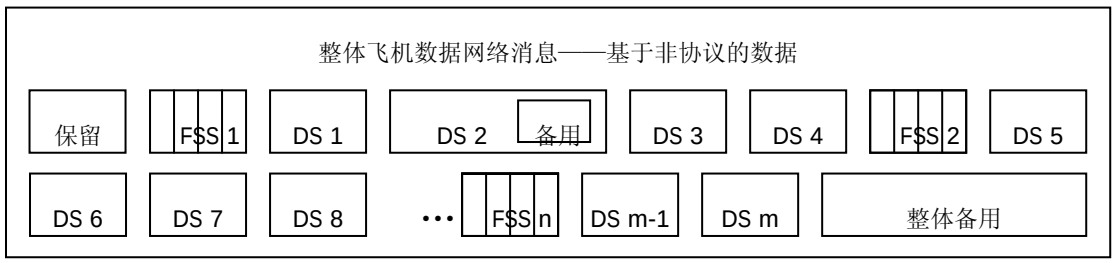


图 1-3.5.1 整体飞机数据网络（ADN）消息结构

图1-3.5.1展示了一个完整的飞机数据网络的消息，包含32-bit的保留域，后面跟着“*n*”个功能数据集。第1个功能数据集具有功能状态集，后面跟着4个数据集从DS1到DS4。其中2号数据集在其尾部具有一些备用存储空间。第2个功能数据集以功能状态集开始，后面跟着4个数据集从DS5到DS8。这个模式一直持续到FDSn。这个功能数据集以功能状态集开始，后面跟着2个数据集DSm-1到DSm。整体备用域接在FDS最后。它可能为未来的DS（作为FDSn的一部分）提供空间，或者这些空间能够包含2个附加的FDSn的DS和另外的完整的FDS。

1-3.6 消息设计的指导方针

当需要修改一个已被定义的而且在使用中的消息结构的时候，必须小心仔细。为了使改变的花费尽可能地低，对消息的修改应该遵循的方式是不改动数据集在消息中的位置。如果在数据结构中数据集的位置偶然地被改变，则作为这条消息的生产者和消费者的软件将则会发生不必要的影响。下面的措施能够帮助避免这种在消息结构的改变方面付出高昂的代价：

1. 总是在每个数据集的末尾定义一个备用区域。
2. 总是在每个消息的末尾定义一个整体备用区域。
3. 注意那些对齐所用到的填充区可以用作备用区域，当需要的时候使用它们添加参数。
4. 当从一个数据集中移除一个参数时，将其替换成一个填充区域。不要将消息的剩余的部分上移，不要去掉该参数所填占的区域。
5. 通过避免存在的消息参数位置的移动，使改变对消息的影响最小化，将新参数加入到填充域或备用域。

整体备用区可以被用作在这个消息的末端增加一个完整的新数据集 ，或者为消息中定义的最后一个数据集添加参数。选择上述哪一项方法是由功能状态驱使的。如果新参数被存在的FS所覆盖，将其放在最后的数据集里。如果新参数由一个不同的FS表示，则创建一个新的数据集。

1-4 .0 FDS示例定义

这个部分举例表明在功能数据集中使用AFDX数据原语是如何构造消息的。例如，如何进行从ARINC 429到AFDX的格式化，参见附录B。

1-4.1 AFDX消息结构定义

以太网提供的消息长度明显大于ARINC 429。这允许以与ARINC 429不同的格式表示数据，数据元素能够具有更高的精度。在以太网中最小尺寸的数据包具有18字节的有效载荷。如果在AFDX网络中被发送的消息小于18个字节长度，在消息出现在介质上之前，它将被加上填充字直到18字节。发送18字节的有效载荷并不是一种有效率的网络使用方法，因为与数据包的开销相比载荷较小。当数据包的尺寸达到几百个字节长度，网络开始变得更有效率。

需要根据数据原语的产生方式对它们分组。例如，如果一个传感器以相同的速率发送几个不同的ARINC 429标签值（label），当该传感器配备到一个AFDX网络接口的时候，这些数据元素很可能是被分为一组，一起构成单个的消息。

下面的消息的定义用到经验规则，该规则显示消息是如何由一个功能状态集和一个数据集构成，而数据集如何由一些数据原语构成。当可能的时候，消息结构中的一行通常是一个数据原语。方框表示一个数据原语，它被用来表示一组字节，最高位字节在左边，最低位字节在右边。同样，在字节里面的二进制位也是最高位（msb）在左边，最低位（lsb）在右边。

1-4.2 消息格式举例

表1-4是一个消息格式的例子。它演示如何在消息中应用填充域和备用域。在这个例子中，填充域是为了保持数据元素的对齐。

在一些情况下，布尔型数据原语可能被用来表示离散量。在表格1-4中的示例的消息具有一个打包的布尔量，处于地址偏移量0x00C0，表1-4展示这个被用作位置离散量的布尔量中每个二进制位的定义。图1-4.2.1显示这个布尔数据原语的细节，即它是如何在存储区被实际地表示的。打包的布尔量是右端对齐的，它们从右端开始，向左填入数据位，并且可能在左端留下备用位。

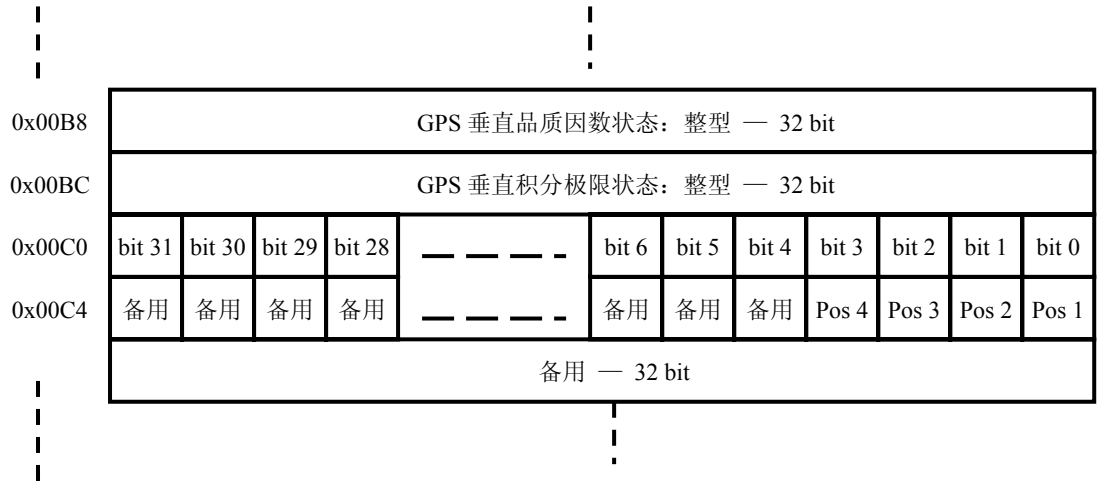


图 1-4.2.1 表 1-4 中消息的布尔量的细节

表 1-4 消息格式举例

FDS 名	飞机参数	数据原语	地址	速率 (Hz)	WKS
GPS PVT 数据	保留	保留字	0x0000	1	C_GPS_Data
	FSS	函数状态字	0x0004		
DS1	GPS 传感器模式	32 位整数型 Integer_32	0x0008		
	GPS 时间标号的 UTC	32 位浮点型 Float_32	0x000C		
	GPS 时间标号的 UTC（精度/小数）	64 位浮点型 Float_64	0x0010		
	GPS 日期日	32 位整数型 Integer_32	0x0018		
	GPS 日期月	32 位整数型 Integer_32	0x001C		
	GPS 日期年	32 位整数型 Integer_32	0x0020		
	GPS 海拔	32 位浮点型 Float_32	0x0024		
	GPS 地速	32 位浮点型 Float_32	0x0028		
	GPS 垂直速度	32 位浮点型 Float_32	0x002C		
	GPS 南北真实速度	32 位浮点型 Float_32	0x0030		
	GPS 东西真实速度	32 位浮点型 Float_32	0x0034		
	GPS 真实轨迹角	32 位浮点型 Float_32	0x0038		
	填充区	32 位填充型 Pad - 32	0x003C		
	GPS 当前位置－纬度	64 位浮点型 Float_64	0x0040		
	GPS 当前位置－经度	64 位浮点型 Float_64	0x0048		
	GPS 水平精度因子	32 位浮点型 Float_32	0x0050		
	GPS 水平性能参数	32 位浮点型 Float_32	0x0054		
	GPS 水平不确定界限	32 位浮点型 Float_32	0x0058		
	GPS 水平完整界限	32 位浮点型 Float_32	0x005C		
	GPS 接收机自我完好监视检测卫星错误	32 位整数型 Integer_32	0x0060		
	备用区	32 位备用型 Pad-32	0x0064		
	GPS 垂直精度因子	32 位浮点型 Float_32	0x0068		
	GPS 垂直性能参数	32 位浮点型 Float_32	0x006C		
	GPS 垂直完整性界限	32 位浮点型 Float_32	0x0070		
	GPS 传感器模式状态	32 位整数型 Integer_32	0x0074		
	GPS 时间状态 UTC 状态参数	32 位整数型 Integer_32	0x0078		
	GPS 时间状态 UTC（完好/小数）状态参数	32 位整数型 Integer_32	0x007C		
	GPS 时间状态参数	32 位整数型 Integer_32	0x0080		
	GPS 高度状态参数	32 位整数型 Integer_32	0x0084		
	GPS 地速状态参数	32 位整数型 Integer_32	0x0088		
	GPS 垂直速度参数	32 位整数型 Integer_32	0x008C		

续表 1-4

FDS 名	飞机参数	数据原语	地址	速率 (Hz)	WKS
	GPS 南北速度，真状态	32 位整数型 Integer_32	0x0090		
	GPS 东西速度，真状态	32 位整数型 Integer_32	0x0094		
	GPS 真实轨迹角参数	32 位整数型 Integer_32	0x0098		
	GPS 当前位置——纬度参数	32 位整数型 Integer_32	0x009C		
	GPS 当前位置——经度参数	32 位整数型 Integer_32	0x00A0		
	GPS 水平精度因子参数	32 位整数型 Integer_32	0x00A4		
	GPS 水平品质因子状态	32 位整数型 Integer_32	0x00A8		
	GPS 水平不确定界限参数	32 位整数型 Integer_32	0x00AC		
	GPS 水平完整性界限参数	32 位整数型 Integer_32	0x00B0		
	GPS 垂直精度因子参数	32 位整数型 Integer_32	0x00B4		
	GPS 垂直品质因子状态	32 位整数型 Integer_32	0x00B8		
	GPS 垂直完整性界限参数	32 位整数型 Integer_32	0x00BC		
	离散值	32 位布尔型 Boolean_32	0x00C0		
	位置 0	布尔型 Boolean	0x00C0:0		
	位置 1	布尔型 Boolean	0x00C0:1		
	位置 2	布尔型 Boolean	0x00C0:2		
	位置 3	布尔型 Boolean	0x00C0:3		
	备用	28 位备用型 Spare - 28	0x00C0:4-31		
	备用	32 位备用型 Spare - 32	0x00C4		
	备用	32 位备用型 Spare - 32	0x00C8		

附件 2　IP/ICMP，UDP 和 TCP 定制条款

表 2-1，表 2-2 和表 2-3 是 RFC1122 中相应表格的拷贝。这些表中列出了每个协议中可用的基本的服务和特性。对于每个协议，RFC-1122 基于每一个特征，提供一个关于需求、建议和选项的显式集合。这些规定都是针对商用以太网的。以下表格列出了针对 AFDX 的信息。

注释

对于 AFDX，不采用“可选项”（OPTION）一栏，取而代之的是“不适用”（NOT APPLICABLE），这是为了标识一些不能应用于 AFDX 端系统的需求。例如，IP 选项和路由器需求。

“不适用”应该被理解为“不许可”（MUST NOT，译为：“不许可”、“不得”）。

表 2-1- 传输层 TCP 需求概要（RFC-1122 第 4.2 节）

传输层 TCP 需求概要（RFC-1122 第 4.2 节）	INTERNET RFC1122	AFDX			注释
		必须	不适用	不许可	
推送（push）标志					
聚合（aggregate）或排队非推送（un-pushed）数据	4.2.2.2	X			
当应用层发出一系列的“发送”（SEND）呼叫但没有设置推送标志时，TCP 可以在内部聚合数据且不发送。同样，当收到一系列的段（Segment）而没有设置推送标志位（PSH）时，TCP 可以在内部排队数据而不将接收数据传输给应用层。					
发送者销毁连续的推送标志	4.2.2.2			X	
当发送方在对数据进行组包时，为发送最大的段，发送方应该销毁连续的推送标志位。					
发送呼叫能够指定推送操作	4.2.2.2	X			
TCP 可以在发送呼叫中实现推送标志…					
若不能，则发送者缓存操作未被明确定义的	4.2.2.2			X	
…这时发送 TCP：(1)不得无明确定义地缓存数据。					
若不能，则推送最后的段	4.2.2.2	X			
并且(2)必须在最后缓存的段中设置推送标志位（即：队列中不再有数据发送时）。					
通知接收方应用层推送操作	4.2.2.2	X			
传递一个已经收到的推送标志到应用层，此项现在是可选的。					
在可能的情况下发送最大长度的段	4.2.2.2	X			
然而，为改善性能，TCP 应该尽可能地发送最大长度的段。					
滑动窗口					

续表 2-1

传输层TCP需求概述（RFC-1122第4.2节）	INTERNET RFC1122	AFDX			注释
		必须	不适用	不许可	
作为无符号整数对待	4.2.2.3	X			
滑动窗口的大小必须作为无符号整数对待。					
作为 32-bit 的数字处理	4.2.2.3	X			
为了以后扩展，TCP 实现应把窗口大小作为 32 整数对待。					
从右边收缩窗口	4.2.2.16			X	
TCP 接受方不应该收缩窗口，也就是窗口由右至左收缩。					
抵御窗口收缩的强健性	4.2.2.16	X			
然而，发送 TCP 必须具备抵御窗口收缩的强健性；窗口收缩可能导致“可用窗口” 的大小（见 4.2.3.4 节）变成负值。					
接收方窗口无明确定义地关闭	4.2.2.17			X	
TCP 可以使提供给它的接收窗口无明确定义地关闭。					
发送方探测零窗口	4.2.2.17	X			
必须支持探测（被提供的）零窗口。					
重传超时（retransmission timeout, RTO）过后首先探测窗口（注 1）	4.2.2.17	X			
当一个零窗口持续一个重传超时间隔后，发送主机应该发送第一个零窗口探测					
指数回退	4.2.2.17	X			
发送主机应该指数增加连续的两次探测之间的时间间隔。					
允许窗口无明确定义地保持为零	4.2.2.17	X			
发送 TCP 必须允许连接保持开放。					
发送方超时 OK 零窗口连接	4.2.2.17			X	
发送 TCP 必须允许连接保持开放。					
紧急（urgent）数据					
指针指向最后八位组（octet）	4.2.2.4		X		
紧急指针指向最后的（LAST）八位组的数据序列号（不是 LAST+1）。					
任意长度的紧急数据序列	4.2.2.4		X		
TCP 必须支持任意长度的紧急数据序列					
异步通知应用层紧急数据	4.2.2.4		X		
无论何时接收到紧急指针，并且先前没有未处理的紧急数据，TCP 必须异步地通知应用层。					

续表 2-1

传输层 TCP 需求概述（RFC-1122 第 4.2 节）	INTERNET RFC1122	AFDX			注释
		必须	不适用	不许可	
应用层可以得知是否/有多少紧急数据在排队			X		
必须有一种方法使应用层能够得知连接上还有多少紧急数据未被读取，或至少确定是否还有紧急数据未被读取。					
TCP 可选项（options）					
在任何段中接收 TCP 可选项	4.2.2.5	X			
TCP 必须能够在任何段中接收 TCP 可选项。					
忽略未被支持的可选项	4.2.2.5	X			
假设可选项有一个长度域，TCP 必须无误地忽略它未实现的可选项。					
处理不合法的可选项长度	4.2.2.5	X			
TCP 必须准备处理不合法的可选项长度					
实现发送和接收最大段长度（maximum segment size, MSS）的可选项	4.2.2.6	X			
TCP 必须实现发送和接收 MSS 可选项。					
若不是默认的 536，则发送 MSS 可选项，	4.2.2.6	X			
当 TCP 收到的 MSS 不是默认的 536 时，应该在每个同步（SYN）段中发送一个 MSS 可选项…					
总是发送 MSS 的可选项	4.2.2.6			X	
…可以总是发送 MSS 的可选项。					
发送 MSS 的默认值是 536	4.2.2.6	X			
如果在连接建立时没有收到 MSS 可选项，TCP 必须设定已发送了一个 MSS 的默认值为 536。					
计算有效的发送段长度	4.2.2.6	X			
TCP 真正发送的段的最大长度，即“有效发送 MSS”，必须是发送 MSS 中较小的（反映远程主机可用重新组装的缓存大小）且是 IP 层所允许的最大长度。					
TCP 校验和					
发送方计算校验和	4.2.2.7	X			
发送方必须生成校验和					
接收方检查校验和	4.2.2.7	X			
接收法必须检查校验和					
打开连接					

续表 2-1

传输层 TCP 需求概述（RFC-1122 第 4.2 节）	INTERNET RFC1122	AFDX			注释
		必须	不适用	不许可	
支持同时开放尝试	4.2.2.10	X			
TCP 必须支持同时开放尝试。					
SYN-RCVD（已收到连接请求）状态记忆前一状态	4.2.2.11	X			
TCP 实现必须跟踪记忆 SYN-RCVD 状态，以确定是以被动打开（OPEN）还是主动打开方式到达该状态。					
被动开放呼叫与其他连接相干涉	4.2.2.18			X	
每个被动打开呼叫不得影响任何以前创建的连接记录。					
为相同的端口功能仿真侦听	4.2.2.18	X			
支持并发多用户使用的 TCP，必须提供一个打开呼叫，当应用程序的一个连接在一个端口上处于 SYN-SENT（连接请求已经发出，等待确认应答）或 SYN-RCVD 状态时，此呼叫将允许在相同端口上进行侦听。					
必要时为 SYN 询问源 IP 地址	4.2.3.7	X			
TCP 必须询问 IP 层以选择一个本地的 IP 地址。					
否则，使用连接的本地地址	4.2.3.7	X			
其他情况下，无论前一个段在此连接上是被发送或接收，TCP 必须使用与先前段使用的相同的本地地址。					
对于广播/多播 IP 地址的打开	4.2.3.10	X			有可能收到多播打开呼叫，在这种情况下，必须将其拒绝。
TCP 实现必须拒绝一个来自无效的远程 IP 地址（如：来自一个广播或多播地址）的本地打开呼叫，将其作为一次错误。					
静默地丢弃到广播/多播地址的段	4.2.3.10	X			
TCP 实现必须静默地丢弃地址为广播/多播地址的到达的 SYN 段。					
关闭连接					
RST（reset，复位）能携带数据	4.2.3.12	X			
TCP 应该允许接收的 RST 段携带数据。					
通知应用程序取消中断	4.2.2.13	X			
如果 TCP 连接被远程站点关闭，本地应用程序必须被告知是否为正常关闭还是被异常取消。					
半双工关闭连接	4.2.2.13			X	
主机可以实现“半双工”TCP 关闭顺序，这样已经呼叫关闭的应用程序不得继续从连接中读取数据。					

续表 2-1

传输层 TCP 需求概述（RFC-1122 第 4.2 节）	INTERNET RFC1122	AFDX			注释
		必须	不适用	不许可	
发送 RST 指示数据丢失	4.2.2.13	X			
当接收的数据还在 TCP 中，如果主机发出关闭呼叫，或关闭呼叫发出后新数据被接收，TCP 应该发送 RST 指示数据丢失。					
在 TIME_WAIT（等待超时，即：等待重发数据）状态持续 2×MSL（Maximum Segment Lifetime）秒	4.2.2.13	X			
当连接主动关闭时，它必须在 TIME_WAIT 状态逗留 2×MSL 秒。					
从 TIME_WAIT 状态接收 SYN	4.2.2.13			X	
可以从远程 TCP 接收一个新的 SYN 以直接从 TIME_WAIT 状态中重新打开连接。					
重传					
Jacobson 慢启动算法	4.2.2.15			X	
在 Jacobson[TCP:7]近期在网络拥塞和 TCP 重传稳定性方面的工作中创造了一种发送算法，这种算法把“慢启动”和“拥塞避免”相结合。TCP 必须实现这种算法。					
Jacobson 拥塞避免算法	4.2.2.15			X	
在 Jacobson[TCP:7]近期在网络拥塞和 TCP 重传稳定性方面的工作中创造了一种发送算法，这种算法把“慢启动”和“拥塞避免”相结合。TCP 必须实现这种算法。					
重传相同 IP 标识域	4.2.2.15			X	
若需重传的数据包与原数据包相同（意味着不仅数据边界不变，窗口和段头的确认域也不变），可以使用相同 IP 标识域					
Kam 算法	4.2.3.1			X	
为计算“重传超时”（RTO）主机 TCP 必须实现 Kam 算法和 Jacobson 算法。					
Jacobson 重传超时估计算法	4.2.3.1			X	
为计算“重传超时”（RTO）主机 TCP 必须实现 Kam 算法和 Jacobson 算法。					
指数（趋势的）回退	4.2.3.1			X	
对于相同的数据段的相继的 RTO 值，实现中必须包括“指数回退”。					
SYN 重传超时计算与数据相同	4.2.3.1	X			
SYN 段的重传应该与数据段的重传使用相同的算法。					
推荐使用的初始值和边界	4.2.3.1	X			

续表 2-1

传输层 TCP 需求概述（RFC-1122 第 4.2 节）	INTERNET RFC1122	AFDX			注释
		必须	不适用	不许可	
应该使用下面的值初始化一个新连接的估计参数。					
(a) RTT（往返时间，round-tips time）= 0 秒。					
(b) RTO = 3 秒。					
推荐使用的 RTO 上下边界在大型网络中是不适当的 。下边界应该使用以秒的分数度量，上边界应该使用 2× MSL 度量，即 240 秒。					
确认的生成					
对乱序的段排队	4.2.2.20			X	
TCP 应该能够对乱序的 TCP 段排队					
在发送 ACK（确认应答）之前处理所有已排队的数据	4.2.2.20	X			
通常，必须对接收的数据段进行处理，从而尽可能实现聚合的 ACK 数据段。					
为乱序的段发送 ACK	4.2.2.21			X	
当窗口内合法的段到达但不在窗口左边界的情况下，TCP 可以发出一个 ACK 应答 RCV.NXT。					
延迟的 ACK	4.2.3.2	X			
TCP 应该实现延迟的 ACK，但 ACK 不应被过分延迟。					
延迟小于 0.5 秒	4.2.3.2	X			
延迟必须小于 0.5 秒					
每隔一个全尺寸段的确认应答	4.2.3.2	X			
在全尺寸的段的流量中，至少应该每隔一个段发送一个 ACK。					
接收方 SWS 避免算法（注 4）	4.2.3.2	X			
TCP 必须在接收端包括“SWS 避免”算法。					
发送数据					
配置生存时间（TLL）	4.2.2.19		X		不在 AFDX 中使用，缺省值是 1。
用来发送 TCP 段的 TTL 值必须被配置。					
发送方 SWS 避免算法	4.2.3.6			X	
TCP 必须在发送方包括 SWS 避免算法。					
Nagle 算法	4.2.3.4			X	
为适应短数据段，TCP 应该实现 Nagle 算法					
应用程序能够将 Nagle 算法设置为无效。	4.2.3.4		X		

续表 2-1

传输层 TCP 需求概述（RFC-1122 第 4.2 节）	INTERNET RFC1122	AFDX			注释
		必须	不适用	不许可	
必须有一种方法使应用程序能够在一个个体的连接上置 Nagle 算法无效。					
连接失败					
R1 重传否定提示（注 5）	4.2.3.5	X			
当相同的段重传的次数达到或超过门限 R1 时，应发送否定提示到 IP 层，以触发网关诊断。					
关闭 R2 重传连接（注 6）	4.2.3.5	X			
当相同数据段重传的次数达到门限 R2（R2 大于 R1）时，关闭连接.					
应用程序能够设置 R2	4.2.3.5	X			
应用程序必须能够为某一连接设置 R2。					
当重传次数小于 R1 且大于等于 R2 时通知应用程序	4.2.3.5			X	
当重传次数介于 R1 与 R2 之间时，TCP 应该通知应用程序传输问题（除非是这样的通知已经被应用程序设置为无效）。					
推荐的 R1 和 R2 的值	4.2.3.5	X			
R1 应该至少是重传 3 次所用的时间，R2 应该至少是 100 秒。					
为 SYN 使用相同的机制	4.2.3.5	X			
SYN 重传的处理应与上面所描述的对数据重传的处理相同，包括对应用层的通知。					
对于 SYN 的 R2 至少 3 分钟	4.2.3.5			X	
对于 SYN 段的 R2 必须足够大（至少 3 分钟）以提供重传。					
发送“keep-alive”（保持活性）包。	4.2.3.6			X	AFDX 中连接是静态配置的，所以，“keep alive”数据包没有用处。
应用程序能够发送请求	4.2.3.6		X		
TCP 实现可以包括“keep-alive”。					
缺省设置是“关”（off）	4.2.3.6		X		
若包括“keep-alive”，缺省必须是“关”。					
仅当空闲的间隔发送	4.2.3.6		X		
“Keep-alive”包必须只能在连接上没有数据和确认包到达的间隔内被发送。					

续表 2-1

传输层 TCP 需求概述（RFC-1122 第 4.2 节）	INTERNET RFC1122	AFDX			注释
		必须	不适用	不许可	
间隔是可配置的	4.2.3.6		X		
间隔必须是可配置的。					
缺省值至少是 2 小时	4.2.3.6		X		
若实现了“keep-alive”机制，要像“死”连接一样，它不能为了响应任何特定的探测而解释失败。					
IP 可选项					
忽略 TCP 不理解的可选项	4.2.3.8	X			
当接收的可选项从 IP 层上传向 TCP 层时，TCP 必须忽略它不理解的选项。					
支持时间戳	4.2.3.8			X	
TCP 可以支持时间戳。					
支持路由记录	4.2.3.8			X	
TCP 可以支持路由记录可选项。					
源路由：					
应用程序可以具体规定源路由。	4.2.3.8			X	
当应用程序主动打开连接时，它必须能够具体指明源路由。					
在数据报（datagram）中覆盖源路由	4.2.3.8			X	
此路由信息必须优于数据段中的源路由信息。					
从源路由信息中建立返回路由	4.2.3.8			X	
当 TCP 连接被动地打开且到达的数据包带有完全 IP 源路由选项（包含返回路由），TCP 必须保存返回路由，并且在此连接上的所有段使用此路由。					
最迟的源路由覆盖	4.2.3.8			X	
如果最迟的段中具有不同的源路由，最迟的定义应该覆盖早前的定义。					
从 IP 接收 ICMP 信息	4.2.3.9			X	
发送目的不可达（0，1，5）到应用程序。	4.2.3.9		X		仅允许 ICMP“回送请求”（echo request）和“回送应答”（echo reply）
TCP 必须通过减慢连接上的发送来响应源抑制（source quench）。为源抑制推荐的处理过程是触发慢启动（slow start），如同发生了重传超时。目的不可达——代码 0,1,5。					

续表 2-1

传输层 TCP 需求概述（RFC-1122 第 4.2 节）	INTERNET RFC1122	AFDX			注释
		必须	不适用	不许可	
目的不可达(0, 1, 5)中止（abort）连接。	4.2.3.9		X		仅允许 ICMP“回送请求”和“回送应答”
因为不可达信息指示软错误状态，所以 TCP 不能中止连接。					
目的不可达（2, 4）中止连接	4.2.3.9		X		仅允许 ICMP“回送请求”和“回送应答”
目的不可达——代码 2，4；表示存在硬错误，所以应该中断连接。					
源抑制（source quench） => 慢启动	4.2.3.9		X		仅允许 ICMP“回送请求”和“回送应答”
TCP 必须通过减慢连接上的发送来响应源抑制（source quench）。为源抑制推荐的处理过程是触发慢启动（slow start），如同发生了重传超时。					
超过时间 => 告知应用程序不中止	4.2.3.9		X		仅允许 ICMP“回送请求”和“回送应答”
此时应该与目的不可达——代码 0, 1, 5 的处理方法相同。					
参数问题 => 告知应用程序不中止	4.2.3.9		X		仅允许 ICMP“回送请求”和“回送应答”
此时应该与目的不可达——代码 0, 1, 5 的处理方法相同。					
地址验证（validation）					
对无效的 IP 地址的打开呼叫	4.2.3.10	X			
TCP 实现必须拒绝一个来自无效的远程 IP 地址的本地打开呼叫，将其作为一次错误。					
拒绝来自无效 IP 地址的 SYN	4.2.3.10	X			
来自无效源地址的 SYN 必须被 TCP 层或 IP 层忽略。					
静默地丢弃目的是广播/多播地址的 SYN	4.2.3.10	X			
TCP 实现必须静默地丢弃目的是广播/多播的 SYN。					
TCP/应用层接口服务					

续表 2-1

传输层 TCP 需求概述（RFC-1122 第 4.2 节）	INTERNET RFC1122	AFDX			注释
		必须	不适用	不许可	
错误报告机制	4.2.4.1	X			
必须有一种机制向应用程序报告软 TCP 错误状态。					
应用程序能够使错误报告例程（Error Report Routions）无效	4.2.4.1			X	应用程序能够忽略错误报告。
不想接收这样的错误报告（ERROR_REPORT）的应用程序应能够置错误报告例程无效。					
应用程序能够为发送具体规定 TOS	4.2.4.2			X	在 AFDX 中不使用 TOS。
应用层必须能够为发送到连接上的段具体规定服务类型（type of service, TOS）。					
无更改传给给 IP	4.2.4.2		X		在 AFDX 中不使用 TOS。
当在连接上发送数据段时，TCP 应该把当前 TOS 的值无更改的传给 IP 层。					
应用程序在连接期间可以更改 TOS	4.2.4.2			X	
应用程序应该能够在连接期间更改 TOS。					
将接收的 TOS 向上传送给应用程序。	4.2.4.2		X		在 AFDX 中不使用 TOS。
TCP 可以将最近接收的 TOS 传送给应用程序。					
FLUSH（刷新）呼叫	4.2.4.2			X	
一些 TCP 实现已经包含 FLUSH 呼叫。					
在打开呼叫中可选的本地 IP 地址参数	4.2.4.4			X	
打开呼叫必须有一个可选参数。					
注：					
1 RTO：retransmit timeout, 重传超时					
2 MSS：maximum segment size, 最大段尺寸					
3 ISN：initial sequence number, 初始序号					
4 SWS：silly window syndrome, 糊涂窗口综合症					
5 R1：first retransmission threshold, 第一个重传门限					
6 R2：second retransmission threshold, 第二个重传门限					

表 2-2 - 传输层 UDP 需求概要（RFC-1122 章节 4.1）

传输层 UDP 需求概要（RFC-1122 第 4.1 节）	INTERNET RFC1122	AFDX			注释
		必须	不适用	不许可	
UDP 发送端口不可达	4.1.3.1		X		
若一个数据报到达，此数据报所发往的 UDP 端口没有侦听呼叫，则 UDP 应该发送一个 ICMP 端口不可达消息。					
UDP 中的 IP 可选项					
传送接收的 IP 可选项到应用层	4.1.3.2		X		在 AFDX 中不使用 IP 可选项。
UDP 必须传送它从 IP 层所收到的任何 IP 选项透明的传给应用层。					
应用层能够在发送中具体规定 IP 可选项	4.1.3.2		X		在 AFDX 中不使用 IP 可选项。
应用层必须能够在所发送的 UDP 数据报中具体规定 IP 选项。					
UDP 向上传送 IP 可选项到 IP 层	4.1.3.2		X		在 AFDX 中不使用 IP 可选项。
UDP 必须传送这些选项到 IP 层。					
传送 ICMP 消息到应用层	4.1.3.3	X			
UDP 必须传送它从 IP 收到的所有 ICMP 错误消息到应用层。					
UDP 校验和					
能够生成/检查校验和	4.1.3.4		X		AFDX 中不使用校验和
主机必须具有实现生成和检查校验和的功能。					
静默地丢弃校验和出错的数据报	4.1.3.4		X		AFDX 中不使用校验和
如果收到的UDP数据报的校验和非灵并且无效，则UDP静默地丢弃此数据报。					
发送方不生成校验和的选项	4.1.3.4		X		AFDX 中不使用校验和
应用程序可以选择并控制是否生成 UDP 校验和。					
默认生成校验和	4.1.3.4		X		AFDX 中不使用校验和
但在缺省的情况下必须是生成校验和。					
接收方可以选择是否接收无校验和的数据报	4.1.3.4		X		AFDX 中不使用校验和

续表 2-2

传输层 UDP 需求概要（RFC-1122 第 4.1 节）	INTERNET RFC1122	AFDX			注释
		必须	不适用	不许可	
应用程序可以选择和控制是否接收无校验和的数据报。					
UDP 多重宿主（multihoming）功能					
传送特定的目的地址到应用层	4.1.3.5			X	
当 UDP 数据报被接收时，它的特定目的地址必须被传送到应用层					
应用层能够具体规定本地 IP 地址	4.1.3.5			X	
应用程序必须能够具体规定发送 UDP 数据报所使用的 IP 源地址。					
应用层具体规定本地 IP 地址通配符（wildcard）	4.1.3.5			X	
或者不指定 IP 地址（在这种情况下，网络软件将选择合适的源地址）。					
通知应用层所使用的本地 IP 地址。	4.1.3.5			X	
应该存在一种方法通知应用层所使用的源地址（如：这样应用层接下来仅从相应的接口接收应答数据报）。					
来自错误源 IP 地址（的数据报）被 UDP/IP 静默地丢弃	4.1.3.6			X	
来自错误源 IP 地址（如：广播或多播地址）的 UDP 数据报被 UDP/IP 层丢弃					
仅发送有效的 IP 源地址（的数据报）	4.1.3.6			X	
当一主机发送 UDP 数据报时，源地址必须是本主机的 IP 地址之一。					
UDP 应用接口服务					
完全的 3.4 节的 IP 接口应用	4.1.4		X		这些参数是静态配置的，因此不适用。
UDP的应用接口必须提供3.4节描述的全部IP/传输接口的服务。因此，使用UDP的应用程序需要3.4节描述的以下功能：GET_SRCADDR()，GET_MAXSIZES()，ADVISE_DELIVPROB()，和RECV_ICMP()。例如：GET_MAXSIZE()能够被用来学习对于一个特殊的三元组 (接口, 远程主机, TOS) 有效的最大的“UDP最大数据报”的长度。					
当发送数据报时，能够具体规定 TTL，TOS 和 IP 可选项	4.1.4		X		这些参数是静态配置的，因此不适用。

续表 2-2

传输层 UDP 需求概要（RFC-1122 第 4.1 节）		INTERNET RFC1122	AFDX			注释
			必须	不适用	不许可	
	应用层程序必须能够为发送 UDP 数据报所需的设置 TTL、TOS 和 IP 可选项的值，这些值必须透明地传送给 IP 层。					
向上传送接收的 TOS 到应用层		4.1.4			X	
	UDP 可以将接收到 TOS 向上传送到应用层。					

表 2-3 – 传输层 IP 需求概要（RFC1122 第 3.5 节）

传输层 IP 需求概要（RFC-1122 第 3.5 节）	INTERNET RFC1122	AFDX			注释
		必须	不适用	不许可	
实现 IP	3.1	X			
主机软件的 Internet 层必须实现 IP。见 3.3.7 节，要求支持 IGMP 协议（Internet Group Management Protocol，Internet 组管理协议）的需求。					
实现 ICMP	3.1	X			
主机软件的 Internet 层必须实现 ICMP。					
在应用层操纵远程多重宿主（multihoming） 功能	3.1			X	
目前，远程多重宿主必须在应用层被操纵。					
支持本地多重宿主	3.1	X			
主机可以支持本地多重宿主。					
如果有能力转发数据报，则符合网关规范	3.1		X		
转发其他主机生成的数据报的任何主机即作为现行的网关，并且也必须符合在网关需求 RFC[INTRO:2]中安排的规范。					
嵌入式网关的配置开关	3.1		X		
包含嵌入式网关的 Internet 主机必须具有一个配置开关以停用网关功能…					
配置默认为“非网关”（non-gateway）	3.1		X		没有默认配置，只有一个配置。所以，不作改变。
…并且这个开关必须是默认为“非网关”。					
基于接口数目的自动配置	3.1				
如果主机具有多于一个接口，主机软件不得自动地进入网关模式。					
能够记录丢弃的数据报	3.1	X			在 MIB 中记录入日志
然而，为了诊断问题，主机应该提供对错误进行记录的能力（见 1.2.3 节），包括静默地丢弃的数据报的内容…					
以计数器记录	3.1	X			在 MIB 中记录入日志
…并且应该用一个统计量计数器上记录事件					
如果 IP 版本不为 IPv4 则静默地丢弃（数据报）	3.2.1.1	X			
一个版本号不是 IPv4 的数据报必须被静默地丢弃					
验证 IP 校验和，静默地丢弃坏的数据报	3.2.1.2	X			
主机必须验证每个接收的数据报的 IP 头部的校验和，并且静默地丢弃每一个具有错误的校验和的数据报。					
寻址：					

续表 2-3

传输层 IP 需求概要（RFC-1122 第 3.5 节）	INTERNET RFC1122	AFDX			注释
		必须	不适用	不许可	
子网寻址（RFC-950）	3.2.1.3	X			
主机必须支持对 IP[IP:3]的子网扩展					
源地址必须是主机的自有 IP 地址	3.2.1.3	X			
当主机发出任何数据报，IP 源地址必须是它的自有 IP 地址之一（但不是广播或多播地址）。					
静默地丢弃目的地址损坏的数据报	3.2.1.3	X			
一个输入的数据报包含被这一节的规则认为是无效的 IP 源地址，主机必须静默地丢弃这个数据报					
支持重新组装	3.2.1.4	X			
Internet 模型要求每个主机支持重新组装					
在同样的数据报中保留同样的 ID 域	3.2.1.5			X	RFC1122 3.2.1.5 节表示这是一个应用事项
当发送早先的数据报的同样的拷贝，主机可以有选择地保留在这个拷贝中同样的标识域。					
TOS（服务类型）：					
允许传输层设置 TOS	3.2.1.6			X	
IP 层必须提供传输层设置每个被发出的数据报的 TOS 域；默认是所有二进制位全为 0					
将接收到的 TOS 向上传递给传输层	3.2.1.6			X	
IP 层应该将接收到的 TOS 向上传递给传输层					
使用 RFC-795 链路层对 TOS 的映射	3.2.1.6			X	
包含在 RFC-795 中的特殊的链路层 TOS 映射不应该被实现。					
TTL（生存时间，Time-to-Live）：					
发送数据包（packet）时使 TTL 为 0	3.2.1.7			X	必须被设置为 1
主机不得发送一个带有生存时间（TTL）的值为 0 的数据报					
丢弃 TTL<2 的接收数据包	3.2.1.7			X	
主机不得仅仅因为一个正在接收的数据报的 TTL 小于 2 而将它丢弃					
允许传输层设置 TTL	3.2.1.7			X	
IP 层必须提供某种方法让传输层设置发送的每个数据报的 TTL 域					
固定 TTL 是可配置的	3.2.1.7			X	
当用到一个固定的 TTL 值，它必须是可配置的。					

续表 2-3

传输层 IP 需求概要（RFC-1122 第 3.5 节）	INTERNET RFC1122	AFDX			注释
		必须	不适用	不许可	
IP 可选项：					
允许传输层发送 IP 可选项	3.2.1.8			X	
必须有某种方法让传输层对包含在发送的 IP 数据报中的 IP 可选项进行规定。					
传送所有接收到的 IP 可选项到更高层	3.2.1.8			X	
所有在数据报中接收到的 IP 可选项（除了 NOP 或 END-OF-LIST）必须被传递到传输层（或者当数据报是 ICMP 消息的情况下至 ICMP 处理）					
IP 层静默地忽略未知的可选项	3.2.1.8	X			
IP 层和传输层必须逐个解释这些 IP 可选项，理解其中的一部分并且静默地忽略掉其他的部分。					
保密选项	3.2.1.8a			X	
在一些环境下，在每个数据报中要求保密选项。					
发送流识别选项	3.2.1.8a			X	
这个选项被废弃；不应该被发送…					
静默地忽略流识别选项	3.2.1.8b	X			
…并且，如果收到，它必须被静默地被忽略					
	3.2.1.8d		X		这条规定只应用于 ES。
发起和处理记录路由（Record Route）选项的实现是可选的。					
时间戳选项	3.2.1.8e		X		
发起和处理时间戳选项是可选的。					
源路由（Source Route）选项：					
发起和终止源路由选项	3.2.1.8c		X		无可选项
主机必须支持发起一个源路由，并且必须能够充当某个源路由的最终目的地。					
填补到传输层的带有完整源路由的数据报	3.2.1.8c		X		无可选项
如果主机接收到一段包含源路径已完成的消息（即，指针点数超过了最后字段），那么数据包就已经到达了它的最后目的地；收到的选项（记录路径）必须传给传输层（或者交给 ICMP 处理）					
建立正确的（非冗余的）返回路径	3.2.1.8c		X		无可选项
当一条返回源的路径被建立的时候，该路径必须正确地组织，即使所记录的路径包含了源主机。					
在 IP 头部中发送多个源路径选项	3.2.1.8c		X		无可选项

续表 2-3

传输层 IP 需求概要（RFC-1122 第 3.5 节）	INTERNET RFC1122	AFDX			注释
		必须	不适用	不许可	
一个 IP 头部中包含多个源路径选项，那么该数据包不得被发送。					
填补到传输层的带有完整源路由的数据报	3.2.1.8c		X		无可选项
Internet 控制消息协议（ICMP）					
静默地丢弃未知类型的 ICMP 消息	3.2.2	X			
如果接收到一个未知类型的 ICMP 消息，则必须静默地丢弃它。					
包含 8 字节以上的原始数据报	3.2.2			X	
包含的字节数目与接到的相等	3.2.2				
每个 ICMP 错误消息中包含 Internet 头部和并且至少 8 个字节的错误信息数据,多于 8 个字节也可以会发送。					
为传输协议解析 ICMP 错误	3.2.2		X		无 ICMP 错误
在其它情况下如果网络层需要向传输层发送一个 ICMP 错误消息，IP 协议号必须从原来的头部中提取出来，并选择合适的传输层协议实体来处理这个错误。					
发送 TOS=0 的 ICMP 错误消息	3.2.2		X		无 ICMP 错误
一个 ICMP 错误消息应该发送正常的 TOS 位 （例如：0）。					
发送 ICMP 消息作为：					
——ICMP 消息	3.2.2		X		无 ICMP 错误
——IP 广播或 IP 多播	3.2.2		X		无 ICMP 错误
——链路层广播	3.2.2		X		无 ICMP 错误
——非唯一源地址的数据报	3.2.2		X		无 ICMP 错误
在收到下列情况时，不得发送 ICMP 错误消息：					
* 一个 ICMP 错误消息，或者					
* IP 广播或 IP 多播的数据报，或者					
* 链路层广播数据报，或者					
* 不是起始片断（fragment），或者					
* 源地址不定以单个主机的数据报——例如：全 0 地址，回环（loopback）地址，广播地址，多播地址或类地址（Class address）。					
返回 ICMP 错误消息（当不被禁止的时候）	3.2.2		X		无 ICMP 错误
在实际情况中，除了上述禁止发送 ICMP 消息的时候，主机都必须在侦测到错误的时候返回 ICMP 错误数据报。					
目的不可达：					
生成目的不可达（代码 2/3）	3.2.2.1		X		无 ICMP 错误

续表 2-3

传输层 IP 需求概要（RFC-1122 第 3.5 节）	INTERNET RFC1122	AFDX			注释
		必须	不适用	不许可	
主机应该生成代码为 2 和 3 的目的不可达消息。					
向更高层传送目的不可达的 ICMP 消息	3.2.2.1		X		无 ICMP 错误
接收到的目的不可达消息必须向传输层报告。					
更高层协议响应不可达消息	3.2.2.1		X		无 ICMP 错误
传输层应该恰当地使用该信息。					
仅将目的不可达作为提示	3.2.2.1		X		无 ICMP 错误
接收到的目的不可达消息如果代码是 0（网络），1（主机）或者 5（源路由错误）可能是因为暂时的路由故障，并且只能将此当作不可达的提示而不是证明。					
重定向：					
主机发送重定向	3.2.2.2		X		没有来自 ES 的 ICMP 重定向
主机应该发送 ICMP 重定向消息；重定向只能由网关发送。					
在接收到重定向消息时缓存更新路由	3.2.2.2		X		没有来自 ES 的 ICMP 重定向
主机接收到一个重定向消息必须根据该消息来更新它的路由信息。					
处理来自主机和网络的重定向消息	3.2.2.2		X		没有来自 ES 的 ICMP 重定向
每个主机必须准备接受主机和网络的重定向消息并且按照 3.3.1.2 节描述的那样处理它们。					
丢弃非法重定向消息	3.2.2.2		X		没有来自 ES 的 ICMP 重定向
如果指定的新的网关地址不在重定向消息所到达的统一个网络（子网）中，或者重定向消息的源地址不是当前指定目的地的第一跳网关，则重定向消息应该被静默地丢弃。					
源抑制（source quench）：					
在缓存溢出的时候发送原抑制消息	3.2.2.3			X	
当主机接近或者已经到达了输入缓冲区或其他资源的临界值迫使主机丢弃数据报时，主机可以发出一个源抑制消息。					
向更高层传送源抑制消息	3.2.2.3		X		只有 ICMP 的“回送请求”（echo request）和“回送应答”(echo reply)被允许

续表 2-3

传输层 IP 需求概要（RFC-1122 第 3.5 节）	INTERNET RFC1122	AFDX			注释
		必须	不适用	不许可	
当源抑制消息到达时，IP 层将其必须上报给传输层（或 ICMP 处理）。					
更高层对源抑制消息的响应	3.2.2.3		X		只有 ICMP 的“回送请求”和“回送应答”被允许
传输层或应用层应该实现针对任意一个可以向同一目的地发送一系列数据报并可以保存足够的状态信息以满足其可用性的协议应答源抑制消息的机制。					
超过时间（time exceeded）消息；向更高层传送	3.2.2.4		X		只有 ICMP 的“回送请求”和“回送应答”被允许
接收到的超过时间消息必须上传给传输层。					
参数问题：					只有 ICMP 的“回送请求”和“回送应答”被允许
主机应答生成参数问题消息，一个接收到的参数问题消息必须传送到传输层，并被报告给用户。					
发送参数问题消息	3.2.2.5		X		只有 ICMP 的“回送请求”和“回送应答”被允许
把参数问题消息传送给更高层	3.2.2.5		X		只有 ICMP 的“回送请求”和“回送应答”被允许
向用户报告参数问题	3.2.2.5		X		只有 ICMP 的“回送请求”和“回送应答”被允许
ICMP 回送请求（echo request）或回送应答（echo reply）					
回送服务器和回送客户端	3.2.2.6	X			
每个主机必须实现 ICMP 回送服务器功能，即为特定的目的发送回送请求并接收回送应答。					
回送客户端	3.2.2.6	X			
主机应当为发送回送请求和接收回送应该当实现一个应用层接口，该接口用于诊断。					
丢弃发送向广播地址的回送请求	3.2.2.6	X			
发送给一个 IP 广播或多播地址的 ICMP 回送请求可以被丢弃					
丢弃发送向多播地址的回送请求	3.2.2.6	X			

续表 2-3

传输层 IP 需求概要（RFC-1122 第 3.5 节）	INTERNET RFC1122	AFDX			注释
		必须	不适用	不许可	
发送给一个 IP 广播或多播地址的 ICMP 回送请求可能被丢弃					
使用特定目的地址来标识回送应答的源	3.2.2.6	X			
在 ICMP 回送应答中的 IP 源地址必须与应答的回送请求消息中的目的地址一致（参见 3.2.1.3 节）。					
回送应答消息中的数据需要与请求中一致	3.2.2.6	X			
ICMP 回送请求中的数据必须被完全包含在对应的响应消息中。					
向更高层传送回送应答消息	3.2.2.6	X			
回送应答消息必须传送到 ICMP 用户接口，除非相应的请求消息由 IP 层产生。					
反射记录路由，时间戳选项	3.2.2.6		X		只有 ICMP 的”回送请求”和“回送应答”被允许
如果接收到一个记录路由或时间戳选项的 ICMP 回送请求，相应的项目应该被更新加入当前主机，并且该项目将被包含在相应的回送应答消息的 IP 头中，并且不被截断。因此，路由信息将包含整个往返路径（round trip）。					
反向和反射源路由选项	3.2.2.6		X		只有 ICMP 的”回送请求”和“回送应答”被允许
如果在一个 ICMP 回送请求中有源路由选项，则返回的路径必须反向并作为回送应答消息的源路由选项。					
ICMP 信息请求和响应	3.2.2.7		X		只有 ICMP 的”回送请求”和“回送应答”被允许
主机不应该实现这些消息。					
ICMP 时间戳和时间戳响应：	3.2.2.8		X		只有 ICMP 的”回送请求”和“回送应答”被允许
主机可以实现时间戳和时间戳应答消息。					
使延迟可变性（variability）最小化	3.2.2.8		X		只有 ICMP 的”回送请求”和“回送应答”被允许
如果该功能被实现，则它应该可以从设计上保证最小的时延。					

续表 2-3

传输层 IP 需求概要（RFC-1122 第 3.5 节）	INTERNET RFC1122	AFDX			注释
		必须	不适用	不许可	
静默地丢弃多播时间戳	3.2.2.8		X		只有 ICMP 的”回送请求”和“回送应答”被允许
一个发送向多播或组播 IP 地址的 ICMP 时间戳请求消息可以被静默地丢弃。					
静默地丢弃广播时间戳	3.2.2.8		X		只有 ICMP 的”回送请求”和“回送应答”被允许
一个发送向多播或组播 IP 地址的 ICMP 时间戳请求消息可以被静默地丢弃。					
使用特定的目的地址作为时间戳应答的源	3.2.2.8		X		只有 ICMP 的”回送请求”和“回送应答”被允许
时间戳应答消息的源地址必须使用相应的时间戳请求消息的目的地址。					
反射记录路由，时间戳选项	3.2.2.6		X		只有 ICMP 的”回送请求”和“回送应答”被允许
如果在一个 ICMP 回送请求中收到源路由选项，则返回路由必须被反向并且作为时间戳应答消息的源路由选项使用。					
反向和反射源路由选项	3.2.2.8		X		只有 ICMP 的”回送请求”和“回送应答”被允许
如果在一个 ICMP 回送请求中有源路由选项，则返回的路径必须反向并作为回送应答消息的源路由选项。					
向更高层传送时间戳应答消息	3.2.2.8		X		只有 ICMP 的”回送请求”和“回送应答”被允许
到达的时间戳应答消息必须被传送到 ICMP 用户接口。					
遵从“标准值”的规则	3.2.2.8		X		只有 ICMP 的”回送请求”和“回送应答”被允许
时间戳（“标准值”）的推荐的记录形式是从世界时（Universal Time）的午夜开始以毫秒计算。					
ICMP 地址掩码请求和回答					

续表 2-3

传输层 IP 需求概要（RFC-1122 第 3.5 节）	INTERNET RFC1122	AFDX			注释
		必须	不适用	不许可	
地址掩码源可配置	3.2.2.9		X		只有 ICMP 的”回送请求”和“回送应答”被允许
用于一台特定的主机上的方法的选择必须是可以配置的。					
支持地址掩码的静态配置	3.2.2.9		X		只有 ICMP 的”回送请求”和“回送应答”被允许
(1) 静态配置信息…					
启动时动态获得地址掩码	3.2.2.9		X		只有 ICMP 的”回送请求”和“回送应答”被允许
(2) 利用系统初始化过程的副作用动态获得地址掩码…					
通过 ICMP 地址掩码请求/应答获得地址掩码	3.2.2.9		X		只有 ICMP 的”回送请求”和“回送应答”被允许
(3) 发送 ICMP 地址掩码请求并且接收 ICMP 地址掩码应答					
如果没有得到应答则重新发送地址掩码请求	3.2.2.9		X		只有 ICMP 的”回送请求”和“回送应答”被允许
当使用方法(3)时，如果没有立即收到地址掩码响应则将在一小段时间内再次发送地址掩码请求，					
如果没有得到应答则设定为默认掩码	3.2.2.9		X		只有 ICMP 的”回送请求”和“回送应答”被允许
当采用方法(3)时，直到收到一个地址掩码应答前，主机都应该设定一个适合于 IP 地址的掩码。					
只在收到第一个应答时更新地址掩码	3.2.2.9		X		只有 ICMP 的”回送请求”和“回送应答”被允许
当采用方法(3)时，必须用第一个收到的地址掩码应答消息设置与特定的本地 IP 地址有关的掩码。					
合理地检查地址掩码	3.2.2.9		X		只有 ICMP 的”回送请求”和“回送应答”被允许
主机应该在任何地址掩码启用时进行一些合理的检查。					

续表 2-3

传输层 IP 需求概要（RFC-1122 第 3.5 节）	INTERNET RFC1122	AFDX			注释
		必须	不适用	不许可	
发送非授权的地址掩码应答消息	3.2.2.9		X		只有 ICMP 的”回送请求”和“回送应答”被允许
系统不得发送地址掩码响应消息，除非它获得授权作为地址掩码代理（agent）。					
显式地被配置为代理	3.2.2.9		X		只有 ICMP 的”回送请求”和“回送应答”被允许
一个被授权的代理可能是一个主机也可能是一个网关，但是它必须被显式地配置为地址掩码代理。					
静态配置→地址掩码授权标志	3.2.2.9		X		只有 ICMP 的”回送请求”和“回送应答”被允许
使用静态配置地址掩码，应该有一个额外的配置标记来确定主机是否是这个掩码的授权代理。					
初始化时的广播地址掩码应答	3.2.2.9		X		只有 ICMP 的”回送请求”和“回送应答”被允许
如果主机配置成一个代理，那么在初始化时，主机必须的通过合适的接口来广播一个地址掩码应答。					
路由越界数据报					
使用地址掩码来决定本地/远程	3.3.1.1		X		网关的 IP 地址(如果有的话）都在配置表中规定
为了确定目的节点是否与网络连接，必须使用接下来的算法					
在连接的网络上没有网关的操作	3.3.1.1	X			
主机 IP 层必须能够在最小网络环境，尤其是无网关的情况下，正确操作					
保持下一跳网关的“路由缓存”（route cache）	3.3.1.2		X		AFDX（的路由）是静态定义的
为了高效地路由一系列数据报，源主机必须具有路由缓存来保存对下一跳网关的映射					
同等地对待主机和网络的重定向	3.3.1.2		X		只有 ICMP 的”回送请求”和“回送应答”被允许

续表 2-3

传输层 IP 需求概要（RFC-1122 第 3.5 节）	INTERNET RFC1122	AFDX			注释
		必须	不适用	不许可	
因为通常不知道目的地址的子网掩码，因此网络重定向消息应该被当作主机重定向消息处理。					
如果没有缓存入口（entry），则使用默认网关	3.3.1.2		X		AFDX（的路由）是静态定义的
IP 层必须从它的默认网关列表中选择一个。					
支持多个默认网关	3.3.1.2		X		AFDX（的路由）是静态定义的
IP 层必须支持多个默认网关。					
提供静态路由表	3.3.1.2		X		AFDX（的路由）是静态定义的
作为一个附加的特性，IP 层主机可以实现“静态路由”表					
标志（flag）：路径由于重定向而可以被覆盖	3.3.1.2		X		只有 ICMP 的”回送请求”和“回送应答”被允许
每个静态路由可以包含一个标志，用来表示它是否能被 ICMP 重定向覆盖。					
主机端关键的路由项，非网络地址	3.3.1.3		X		AFDX（的路由）是静态定义的，无默认网关
每个路由缓存入口需要包含以下几个域：					
(1) 本地 IP 地址（针对多宿主（multihoming）主机）					
(2) 目的 IP 地址					
(3) 服务类型					
(4) 下一跳网关 IP 地址					
在路由缓存中包含 TOS	3.3.1.3		X		AFDX（的路由）是静态定义的，无默认网关
应该包含 TOS 字段。					
能够侦测下一跳网关的失效	3.3.1.4		X		AFDX（的路由）是静态定义的，无默认网关
IP 层必须能够侦测路由表中下一跳网关的失效，并且选择另一个网关					
假设路由永远完好	3.3.1.4		X		AFDX（的路由）是静态定义的，无默认网关

续表 2-3

传输层 IP 需求概要（RFC-1122 第 3.5 节）	INTERNET RFC1122	AFDX			注释
		必须	不适用	不许可	
一个特定的网关不应该在缺少功能的肯定指示（positive indication）时未被定义地使用。					
不断地 Ping 网关	3.3.1.4		X		AFDX（的路由）是静态定义的，无默认网关
特别地，主机不得简单地通过不断 Ping 网关来确定第一跳网关的状态					
仅流量发出时 Ping 网关	3.3.1.4		X		AFDX（的路由）是静态定义的，无默认网关
只有在数据发向网关之后才能使用 Ping…					
只有在没有肯定指示的情况下才能 Ping	3.3.1.4		X		AFDX（的路由）是静态定义的，无默认网关
…并且，此时没有其他的肯定指示表明该网关的功能正常。					
从更高或更低的层获得提示	3.3.1.4		X		AFDX（的路由）是静态定义的，无默认网关
为了避免 Ping 网关，Internet 层之上或之下的层应当能够对于路由缓存入口的状态给出“提示”，以给出可用的关于网关的肯定的（网关完好）和否定的（网关死机）的信息。					
从失效的默认网关切换到其他网关	3.3.1.5		X		AFDX（的路由）是静态定义的，无默认网关
如果失效的网关不是目前默认的网关，IP 层可以立即切换到默认网关。如果目前使用的是默认网关，并且发生了失效，则 IP 层必须选择另一个网关作为默认网关。					
使用手动的方法来进入配置信息	3.3.1.6		X		AFDX（的路由）是静态定义的，无默认网关
必须提供某种手动进入配置数据的方法					
片断重组（reassembly）与分片操作（fragmentation）：					
能够重新组装接收的数据报	3.3.2	X			
IP 层必须实现 IP 数据报的重组。					

续表 2-3

传输层 IP 需求概要（RFC-1122 第 3.5 节）	INTERNET RFC1122	AFDX			注释
		必须	不适用	不许可	
数据报（的 EMTU_R）至少 576 字节	3.3.2	X			
EMTU_R 是可配置或者是未被定义的（注 7）	3.3.2	X			由系统集成者定义
用 EMTU_R 来设计定义可重组的数据报的接收有效最大传输单位（“Effective MTU to receive”，其中 MTU 为“Maximum Transfer Unit”）；某些情况下也被称为“重组缓存大小”。EMTU_R 必须大于或等于 576，必须是可配置的或者是未被定义的，并且还应该大于或等于所连接网络的 MTU。					
传输层能够学习 MMS_R（能够被接收和重组的最大消息长度，见注 8）	3.3.2		X		由系统集成者定义
必须存在让传输层通过学习获知 MMS_R 的机制					
重组超时的时候发出“时间超过”ICMP	3.3.2		X		只有 ICMP 的”回送请求”和“回送应答”被允许
如果发生超时，则部分重组的数据报必须被丢弃，并且发出一个“ICMP 时间超过”消息到源主机（如果片断 0 已经被收到）。					
设定重组超时的值	3.3.2		X		重组过程不使用超时
TCP 规范[TCP:1]设定 MSL 为 2 分钟。这样设定一个合理的重组超时值的上限。					
向更高层传输 MMS_S（接收有效最大传输单位，见注 8）	3.3.3			X	
主机必须实现某种机制有以使传输层获知 MMS_S。					
输出数据包（packet）的本地分片	3.3.3	X			
可选地，IP 层可以有意地采用某种机制对输出的数据报进行分片。					
否则，不能发送大于 MMS_S 的包	3.3.3			X	
不进行本地分片的主机必须保证传输层（针对 TCP）或者应用层（针对 UDP）能从 IP 层获知 MMS_S，并且不发送长度超过 MMS_S 的数据报。					
向网外的目的发送的 MTU 的最大长度为 576	3.3.3			X	由系统集成者定义
只要目的地址不在连接的网络中，在缺乏整个路径中最小的 MTU 的切实的信息的情况下，IP 层应该使 EMTU_S<=576。否则，使用所连接的网络的 EMTU_S。					

续表 2-3

传输层 IP 需求概要（RFC-1122 第 3.5 节）	INTERNET RFC1122	AFDX			注释
		必须	不适用	不许可	
所有子网 MTU（All-Subnets-MTU）配置标志	3.3.3			X	
一个主机的 IP 层实现可以具有一个“所有子网 MTU 配置标志”（ALL-Subnets-MTU），用来表示所连接的网络中的 MTU 将被用于在同一个网络中的不同子网的目的节点，但对于其他网络则不是这样。					
多宿主（multihoming）：					
对于特定的目的地址以同样的地址作为应答	3.3.4.2	X			
(1) 如果数据报是对一个接收到的数据报的响应，响应的源地址就应该是请求的特定的目的地址。					
允许应用程序选择本地 IP 地址	3.3.4.2	X			
(2) 应用程序必须能够显式地具体规定用于初始化连接或请求得源地址。					
静默地丢弃“错误”（wrong）接口的数据报	3.3.4.2	X			
(A) 如果数据报的目的地址与接收它所在的物理接口不符，主机可以静默地丢弃接收到的数据报。					
仅通过正确的接口发送数据报	3.3.4.2	X			
(B) 主机可以要求它自己只通过与数据报的 IP 地址相符的物理接口发送（非源路由）IP 数据报。					
源路由转发					
以源路由选项转发数据报	3.3.5			X	
按照下面给出的约束，主机可以在源路由中扮演中间跳步（intermediate hop）的角色，并向下一个特定的跳步（hop）转发源路由数据报。					
遵从相关的网关规则	3.3.5		X		此规则仅能用于一个 ES
然而，执行类似于网关的功能，主机必须遵从对于一个网关转发源路由数据报的所有相关的规则。					
按照网关规则更新 TTL	3.3.5		X		此规则仅能用于一个 ES
TTL 域必须递减并且对于网关数据段可能被丢弃。					
能够产生 ICMP 错误代码 4 和 5	3.3.5		X		只有 ICMP 的”回送请求”和“回送应答”被允许
主机必须能够产生带有下列代码的目的不可达消息：					
4 （需要分片但是 DF 置位）当一个源路由数据报不能被分片以适应目标网络					

续表 2-3

传输层 IP 需求概要（RFC-1122 第 3.5 节）	INTERNET RFC1122	AFDX			注释
		必须	不适用	不许可	
5 （源路由失效）当一个源路由数据报不能被转发，例如：因为路由问题或者因为严格源路由的下一跳不在所连接的网内。					
IP 源地址不是本地主机	3.3.5		X		此规则仅能用于一个 ES
被转发的源路由数据报可以（并且在正常情况下将会）有一个不是转发主机地址之一的源地址。					
更新时间戳，记录路由选项	3.3.5		X		不使用这些选项
转发一个包含路由记录选项的源路由数据报的主机必须更新这个选项（如果它还有空间）。					
转发一个包含时间戳选项的源路由数据报的主机必须按照这个选项的相关规则将当时的时间戳添加到该选项。					
对于非本地源路由的可配置的开关	3.3.5		X		此规则仅能用于一个 ES
支持非本地源路由的主机必须具有一个可配置的开关以禁用转发。					
默认为“关”（OFF）…	3.3.5		X		此规则仅能用于一个 ES
这个开关的默认值为“关”。					
对于非本地源陆游满足网关访问规则	3.3.5		X		此规则仅能用于一个 ES
主机必须满足约束非本地源路由转发的可配置管制过滤器的所有需求。					
如果不转发，则发送目的不可达的 ICMP（代码 5）	3.3.5		X		只有 ICMP 的”回送请求”和“回送应答”被允许
如果主机收到了不完整的源路由数据报，但是由于某种原因并没有进行转发，则主机应该应答一个目的不可达的 ICMP（代码 5，源路由失败）消息，除非这个数据本身就是一个 ICMP 错误消息。					
广播：					
广播地址	3.2.1.3			X	
3.2.1.3 节中定义了四种标准的 IP 广播地址形式：					
* 有限广播：{-1, -1}					
* 直接广播：{网络号, -1}					
* 子网直接广播：{网络号, 子网号, -1}					

续表 2-3

传输层 IP 需求概要（RFC-1122 第 3.5 节）	INTERNET RFC1122	AFDX			注释
		必须	不适用	不许可	
* 所有子网直接广播：{网络号, -1, -1}					
主机必须能够识别这些输入数据报中目的地址的任何形式。					
接收 0 或 1 两种广播格式	3.3.6			X	
存在一类非标准的广播地址形式，是用 0 替代了-1。所有主机应该能够识别且接收任何使用这种非标准的广播地址作为目的地址的输入数据报。					
配置可选项以发送 0 或 1 广播	3.3.6			X	
针对每个物理接口，主机可以随意地通过可配置选项选择使用 0 或-1 的广播地址形式来发送数据。					
默认广播为 1 格式	3.3.6			X	
该选项应该使用默认的广播地址形式（-1）。					
识别所有广播地址格式	3.3.6			X	
一个主机必须识别收到的数据报中使用的任何一种目的地址。					
在链路层广播中使用 IP 广播/多播地址	3.3.6			X	
当主机向链路层广播地址发送数据报，那么 IP 目的地址必须要是合法的 IP 广播地址或者 IP 多播地址。					
静默地丢弃仅链路层（link-layer-only）广播的数据报	3.3.6			X	
主机应该丢弃通过链路层广播（第 2.4 节）接收但没有指定 IP 多播或广播目的地址的数据报。					
在所连接的网络中使用有限广播地址	3.3.6			X	
主机应该使用有限广播地址来向所连接的网络广播。					
多播：					
支持本地 IP 多播（RFC-1112）	3.3.7	X			
主机应该支持在所有连接的网络中进行本地 IP 多播；对于这些网络，已经规定了从 D 类 IP 地址到链路层地址的映射。					
支持 IGMP（RFC-1112）	3.3.7			X	
主机应该支持本地 IP 多播包括发送多播数据报、加入多播组并接收多播数据报，还包括离开多播组。这意味着除了 IGMP 协议本身之外的所有协议都是可选的。					
在启动的时候加入全主机组	3.3.7			X	
如果 IGMP 没有实现，那么主机在 IP 初始化时应该加入全主机组（224.0.0.1）并在 IP 层活动的时候一直是这个组的成员。					

续表 2-3

传输层 IP 需求概要（RFC-1122 第 3.5 节）	INTERNET RFC1122	AFDX			注释
		必须	不适用	不许可	
更高层能获知接口多播能力	3.3.7			X	
主机应该提供某种方法以使更高层协议或者应用层确定主机所连接的网络支持的 IP 多播寻址情况。					
接口：					
允许传输层使用所有 IP 机制	3.4			X	
IP 层和传输层之间的接口必须允许对 IP 层所有机制的完全访问，包括可选项，服务类型和生存时间。					
向传输层传送接口标识	3.4			X	
传输层必须具有设置这些接口参数的机制，提供从应用程序传输这些参数的途径；上述两种功能或者任具其一，或者两种兼备。					
向传输层传送所有 IP 可选项	3.4			X	
参数可选项包括接收到的数据报中的所有的 IP 可选项；并且这些可选项被传送到传输层。					
传输层能发送某些 ICMP 消息	3.4			X	
传输层必须能发送某些 ICMP 消息。					
向传输层发送特定的 ICMP 消息	3.4			X	
IP 层必须向合适的传输层处理例程传送某些 ICMP 消息。					
包含 IP 头部+ICMP 消息中的原始数据（8 字节或更长）	3.4			X	
对于一个 ICMP 错误消息，上传的数据必须包括原来的 IP 头部和 ICMP 消息所包含的所有原始消息的字节。					
超常规功能	3.4		X		
注：					
7 EMTU_R：接收到的最大有效传输单元。见 RFC-1122 3.3.2 节。					
8 MMS_R （maximum receive transport size）：能够被接收和重组的最大“消息”长度。见 RFC-1122 3.3.2 节。 MMS_S （maximum send transport size）： 能够被发送的最大“消息”长度。见 RFC-1122 3.3.3 节。					例如：MMS_S = EMTU_S - <IP 头部长度>。
注释：					
为了减少混淆，TCP 轮廓定义中使用“不适用”（NOT APPLICABLE）而不是“不”（NO）。					

附录 A 一个端系统标识的例子

在网络中，一个或多个端系统的主机可以借助以下内容识别：

- 域 ID (功能分组)
- 边 ID
- 位置 ID

域 ID 指出了设备所属的域（功能分组）。
边 ID 指出了设备在域内所在的边。
位置 ID 指出了设备相对于域中的边所在的位置。
域 ID、边 ID 和位置 ID 组成 12 位的用户 ID，作为 IP 地址和 MAC 地址的一部分，结构如图 A-1 所示：

以太网 MAC 控制器识别（48 位）						
固定域	用户定义的 16 位 ID				接口	固定域
24 位	固定域 4 位	域 ID	边 ID	位置 ID	ID	5 位
“0000 0010 0000 0000 0000 0000”	“0000”	4 位	3 位	5 位	3 位	“00000”

IP 单播寻址格式（源或者单播目的）32 位							
A 类	私有 IP 地址	用户定义的 16 位 ID				分区 ID	
1 位	7 位	固定域 4 位	域 ID	边 ID	位置 ID	8 位	
“0”	“0001010”	“0000”	4 位	3 位	5 位	“000”	5 位

图 A-1 MAC 和 IP 地址的结构

域 ID 编码用 4 位二进制位，0000 和 1111 是禁止使用的，因此，域 ID 有 14 种可能的选择。
边 ID 编码用 3 位二进制位，000 和 111 是禁止使用的。
位置 ID 编码用 5 位，00000 和 11111 是禁止使用的
对于每一个主机设备，域 ID、边 ID 和位置 ID 都会被详细说明。

附录 B ARINC 429 的 AFDX 格式指南

飞机上有时会用到ARINC 429设备。AFDX网络必须具备一种手段，从这些设备传输数据，或者将数据传输到这些设备。这一节将提供将ARINC 429的标签值（label）打包封装在AFDX消息里的消息定义的例子。

在AFDX网络上放置ARINC429 的数据的首选方法是将单个的参数转换成（本文档的附件1）表1-1中允许的AFDX数据元素之一，所有工作都在AFDX和ARINC429相交汇的网关上进行。这有助于保证尽可能地将网络上所有的数据一致地表示。由于某些原因，当这种情况不可能时，如果使用下面的方法，至少能够获得一定级别的通用性。

给出两种普通的方法：

1. 把每个标号放入固定长度32-bit的非透明数据原语，
2. 把一些不同数量的标签值放入可变长度的非透明结构中。

方法1：

图B-1.1是一个将一个单独的标签值放入消息的例子。这个结构应该被用来容纳单个的标签值信息。注意FS（functional status 功能状态）代表的是将ARINC 429数据引入AFDX网络的网关的状态，ARINC 429设备的状态则在这个标签值内得到体现。

图 B-1.1 单个标号消息的消息格式

图B-1.2和B-1.3展示一个消息格式中包含多个标签值消息的例子。ARINC 429标签值通过集中器设备（concentrator device）放置在网络上，这些集中器设备包括：

- RDC（远程数据集中器）——集总离散量/ARINC429 等
- IOC（输入输出集中器）——集总模拟量/离散量/ARINC429 等
- RIU（无线电接口单元）—— 从ARINC 429到无线电设备的通信以及从无线电设备到ARINC 429的通信
- P5 （座舱顶部控制面板）—— 使用ARINC 429与控制面板通信

图 B-1.2 使用一个数据集（DS）的多标签值消息结构

图 B-1.3 使用多个数据集（DS）的多标签值消息结构

当这些设备将ARINC 429标签值封装到以太网帧的时候，生成ARINC 429标签值的设备的状态在各个单个的帧中以符号状态矩阵（SSM）域的形式被标识。在这种方法中，每个ARINC 429设备的状态是在帧中被标识。FS被用来标识集中器的状态。

如果所有的标号被一个单线程的设备集中到单个以太网帧，应该使用单个的DS，并且DS的状态被单个的FS所反映。如果集中器设备被设计为某种方式，它的一部分可能以某种形式失效，进而导致一些标签值无效，但一个完整的合法以太网帧仍旧被生成并带有一些有效的标号，应该考虑多于一个的DS的方式。这种方式下每个DS被它自己的FS所代表。这允许即使在故障和一些无效数据的压力下，一些完好的数据被接收并被使用。

方法2

这种方法假设网关在一些类型的调度表下运行，并且周期性地发送带有ARINC 429标签值的消息。标签值可以按照不同的速率到达网关，这导致在任何给定的周期不同数量的标号被集中到AFDX网络。这个方法将标签值表提供给数据的消费者，就如同这个消费者被直接连接到ARINC 429设备一样。这些标签值按照次序被网关接收，这样网关和AFDX网络在本质上是透明的。因为任何一个从网关上发送的消息可能拥有可变数量的标签值，所以要有一个可变数量的数据结构（参见图B-1.4）。可变长度的非透明结构恰好被提供来作这样的工作。系统设计者必须指出在两次传输之间可能到达网关的标签值的最大和最小数量。从最大数量，确定可变长度非透明结构的最大尺寸。随后，这个结构被静态的配置到系统中。

当标签值到达网关，它们被一个接着一个地放在非透明结构中，在长度域和填充域的字节的后面。当到了传输的时间，网关将计算出静态长度中有多少完好数据的字节数，并且能够把这个数字加载到这个结构的长度域。非透明域中未被使用的部分全部被填充为二进制0。然后消息被发送，具有完整的静态结构，而不仅仅是用到的部分。

接收到这个结构的数据消费者能够读取长度域，而且计算这里有多少个标签值将被处理。当既有的代码将被重用，用以提供对ARINC 429设备的服务的情况下，很有可能已经存在对每个标签值进行解析并确定数据内容的代码。

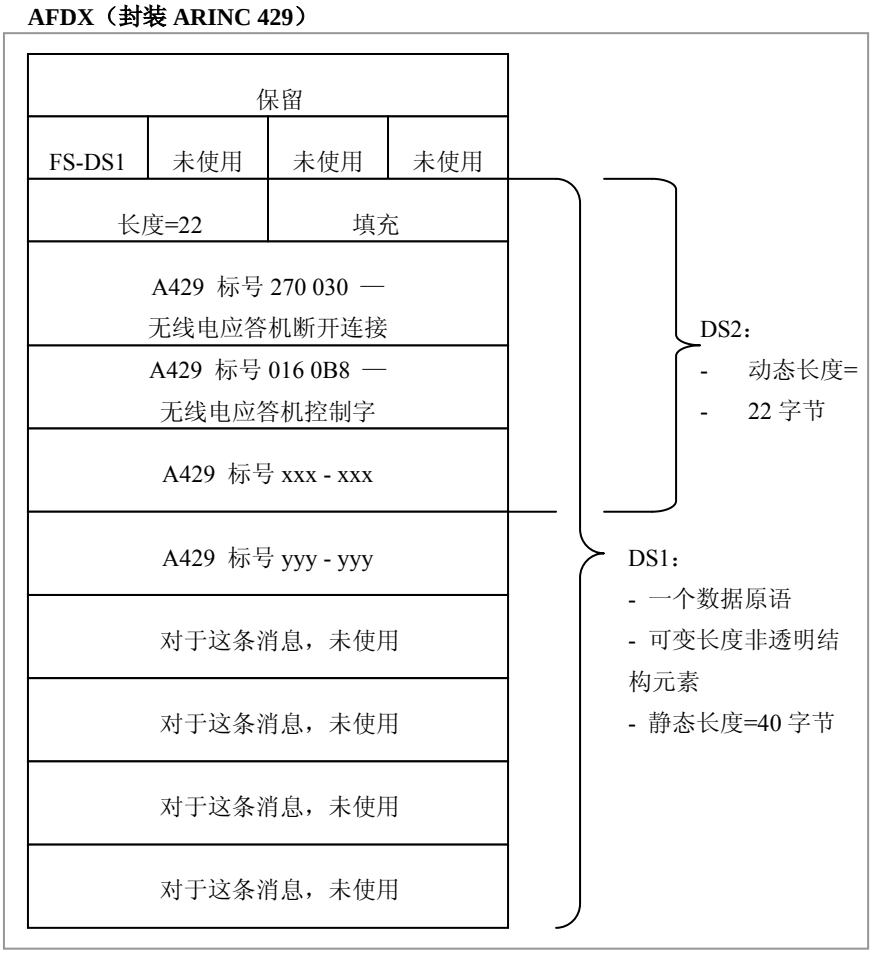


图 B-1.4　对于 ARINC 429 使用可变长度非透明结构的消息格式

发送完整的，静态定义的结构，有助于在系统中保持较低的时延抖动。这样使得每个消息具有同样的长度，尽管消息中的数据并不都是有效的。根据消息可能的最大的尺寸，带宽已经被分配给这样的数据流，并且对于任何其他的数据流是不可用的。所以这样不会发生浪费。

附录 C 网络术语

应用数据（Application Data）

应用数据是通过接口的数据单元，所谓接口是指一个应用程序和在端系统上实现的通信端口或SAP端口之间的接口。应用数据仅包含这个应用程序发送或接收的数据。应用数据被封装在UDP数据报或TCP段中被发送。

术语“应用消息”（Application Message）有时也可以被用来代替“应用数据”。

段（Segment）

段是通过TCP层和IP层之间的接口的数据单元。一个段包含一个TCP头部，接着一个应用消息。段被封装在IP数码报中被发送。

消息（Message）

在本文档中，消息是在TCP或UDP级别上传输的单元。一个消息包含一个传输协议的头部，接着是应用数据。为了通过AFDX端到端地传送，消息必须被封装在一个IP数据报中。

IP 数据报（IP Datagram）

IP数据报是在IP层端到端传输的单元。一个IP数据报包含一个IP头部，接着是传输层数据，即：带有一个IP头部，接着一个UDP数据报或一个段。

片断（Fragment）

数据包的最大尺寸被限制在网络的MTU（最大传输单元）。当一个UDP数据报过长而不能装进一个数据包中的时候，它被打断成为几乎是任意数目的数据片，分别被放置在片断中。

数据包（Packet，又被称为“分组”）

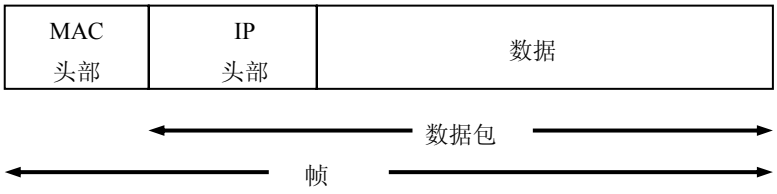
数据包是通过IP层和数据链路层（主要是MAC层）的数据单元。它包括一个IP头部和数据。数据包可以是一个完整的IP数据报，或者是一个片断。

帧（Frame）

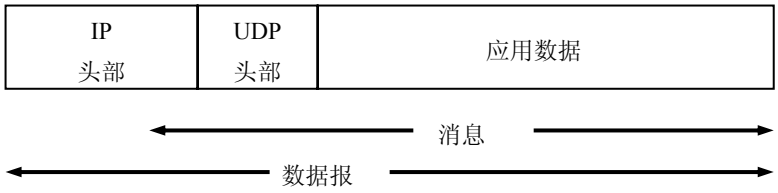
帧是在以太网层的传输单位，包含MAC头部，接着是一个数据包。

术语帧、数据包、数据报、消息、段和应用数据如下面的原理示意图所示：

在有连接的网络中传输：



在IP片断化之前或在IP重新组装之后（对于UDP和TCP）



附件 D 服务到协议的映射

服务到协议的映射

表 D-1 表示将网络服务映射到网络协议的过程。该表描述了提供某类服务所需要使用的协议。表的横行对应于服务，纵列对应于协议。在行列交叉点上，用“X”来表示了 AFDX 网络中网络服务与协议的对应关系。

表 D-1 - 服务到协议的映射

		协议																	
		VIP	IEEE 802.3	IPV4	UDP	TCP	ARP	RARP	ICMP	IGMP	FTP	DNS	DL-TFTP	TFTP	BOOTP	DHCP	HTTP	Telnet	SNMP
服务	物理层与数据链路层		X																
	广播																		
	多播		X	X															
	网络层			X					X										
	数据传输（简单的）				X														
	数据传输（复杂的）																		
	将 IP 地址映射为 MAC 地址（802.3）		X																
	将主机名映射为 IP 地址																		
	网络诊断																		X
	网络配置		X	X	X				X				X						X
	路由																		
	文件传输													X					
	远程用户进入																		
	网络管理																		X
	简单的远程数据存取																		
	ARINC615A 数据加载		X	X	X								X						
	注意：表中列出的大多数协议在 AFDX 都有具体的使用描述。详见 ARINC664 规范 part 7 的主体部分。																		