

上海交通大学硕士学位论文

基于 OPNET 平台的航空电子系统
网络 AFDX 仿真

硕士研究生：高 旭

学 号：1122109014

导 师：付宇卓教授

申 请 学 位：工学硕士

学 科：计算机科学与技术

所 在 单 位：微纳电子学系

答 辩 日 期：2015 年 1 月

授予学位单位：上海交通大学

Dissertation Submitted to Shanghai Jiao Tong University
for the Degree of Master

Avionics System Network AFDX Simulation Based on OPNET Platform

| | |
|--------------------------------------|---------------------------------|
| Candidate: | Xu Gao |
| Student ID: | 1122109014 |
| Supervisor: | Prof. Yuzhuo Fu |
| Academic Degree Applied for: | Master of Engineering |
| Speciality: | Computer Science and Technology |
| Date of Defence: | Jan, 2015 |
| Degree-Confering-Institution: | Shanghai Jiao Tong University |

上海交通大学

学位论文原创性声明

本人郑重声明：所呈交的学位论文《基于 OPNET 平台的航空电子系统网络 AFDX 仿真》，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期： 年 月 日

上海交通大学

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权上海交通大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保密 ☐，在___年解密后适用本授权书。
本学位论文属于
不保密 ☐。

（请在以上方框内打“√”）

学位论文作者签名：

指导教师签名：

日期： 年 月 日

日期： 年 月 日

基于 OPNET 平台的航空电子系统网络 AFDX 仿真 摘要

随着航空技术的快速发展,航空电子系统逐渐成为航空系统中不可或缺的关键部分,航空电子系统由很多独立的航空电子系统组成。航空电子子系统之间需要传输的数据量不断增加,对航空电子系统传输总线的性能和安全性的需求也在不断提高。传统的航空电子系统传输总线标准 ARINC429、ARINC629 和 Mil-std-1553B 等,由于带宽小、速度慢和价格高的因素,已经无法满足现代航空电子系统的要求。在这种情况下,AFDX (Avionics Full Duplex Switched Ethernet) 网络标准应运而生。AFDX 网络基于以太网标准,传输速率最高可达到 1000Mb/s,同时使用全双工链路进行数据传输,减少了链路冲突。目前,AFDX 已应用于空客公司最新的 A380 飞机上。

虽然 AFDX 网络提供了高带宽和全双工的传输方式,由于使用虚链路的方式进行网络传输,各条虚链路在调度时刻仍然可能产生冲突,各条虚链路的时延抖动和关键链路的端到端时延无法得到保证。本文对 AFDX 网络标准进行了深入的研究,分析了 AFDX 网络中时延和时延抖动形成的机制,并针对传统调度方式的不足,提出了一种新的重整形调度算法 RSP 方法。RSP 方法通过实时速率和历史信息动态调整数据帧发送的先后顺序,可降低网络的时延抖动并保障关键链路的优先传输,并设计了相应的实验方案在 OPNET 网络仿真平台上进行了验证。

在 OPNET 网络仿真平台上,我们对 AFDX 网络进行建模,建立了 AFDX 网络的网络层模型和航空电子子系统、端系统以及交换机的节点层模型和进程层模型。并搭建仿真网络对 AFDX 网络进行仿真,选取平均时延和时延抖动作为衡量指标,仿真得到传统调度方法与 RSP 调度方法对网络性能的影响。实验证明,RSP 方法能够降低整体链路的时延抖动并保证关键链路的优先传输,对于实际航空电子通信

网络的设计有重要的参考意义。

关键词：AFDX 网络，RSP 调度方法，端到端时延，时延抖动

AVIONICS NETWORK AFDX SIMULATION BASED ON OPNET PLATFORM

ABSTRACT

With the rapid development of aviation technology, avionics system is becoming an indispensable key part of the aviation system. Avionics system is composed of many separated avionics sub-system. The amount of data transferred between the avionics subsystems is growing and the demand for bus transport avionics system performance and safety is also rising. Traditional avionics data bus ARINC429, ARINC629 and Mil-std-1553B, etc., due to the factor of low bandwidth and high price, has been unable to meet the demands of modern avionics systems. In this case, AFDX (Avionics Full Duplex Switched Ethernet) protocol has emerged. AFDX protocol is based on Ethernet standard, the transmission rate can reach up to 1000Mb / s. While the use of full-duplex data transmission link, conflicts on the link reduces. Currently, AFDX has been applied to the Airbus A380.

Although AFDX protocol provides high bandwidth and full-duplex transmission, each VLs may still conflict in scheduling time. Delay jitter and end to end delay on critical links can not be guaranteed. Through in-depth analysis of the mechanism of delay and delay jitter formation, we propose a new scheduling method RSP(re-shaping method) which can reduce delay jitter and protect high priority transmission link from been preempted. Compared with traditional scheduling method, RSP adjusts sending order of frame based on historical information. In the mean time, we design accordingly experiments to verify our theory on OPNET platform.

On the OPNET simulation platform, we build node model and process model of avionics sub-system、end-system and switch. And then

set up the simulation network and choose average delay and delay jitter as evaluation indicator of network performance. The simulation results shows that RSP method can reduce delay jitter and protect high priority transmission link from been preempted. It has important significance for actual avionics communication network design.

KEY WORDS: AFDX network, RSP scheduling method, end to end delay, delay jitter

目 录

| | |
|------------------------------|-----|
| 基于 OPNET 平台的航空电子系统网络 AFDX 仿真 | I |
| 摘 要 | I |
| ABSTRACT | III |
| 第一章 绪论 | 1 |
| 1.1 引言 | 1 |
| 1.2 国内外研究现状 | 3 |
| 1.2.1 国外研究现状 | 3 |
| 1.2.2 国内研究现状 | 3 |
| 1.3 论文研究意义和研究内容 | 4 |
| 1.3.1 研究意义 | 4 |
| 1.3.2 主要内容 | 4 |
| 1.4 论文结构 | 5 |
| 第二章 AFDX 网络与 AFDX 网络性能分析 | 7 |
| 2.1 全双工交换式以太网 AFDX | 7 |
| 2.2 虚拟链路 | 9 |
| 2.3 AFDX 协议栈 | 11 |
| 2.4 网络时延和时延抖动 | 13 |
| 2.4.1 网络时延 | 13 |
| 2.4.2 时延抖动 | 16 |
| 2.5 冗余网络 | 17 |
| 2.6 本章小结 | 18 |
| 第三章 重整形调度算法 | 20 |
| 3.1 传统调度算法 | 20 |
| 3.2 重整形算法 | 21 |
| 3.2.1 常用整形与调度算法 | 21 |
| 3.2.2 AFDX 网络中的重整形调度算法 | 22 |
| 3.2.3 算法执行流程 | 24 |
| 3.3 算法分析 | 26 |
| 3.4 本章小结 | 31 |

| | |
|----------------------------|----|
| 第四章 基于 OPNET 平台的 AFDX 网络建模 | 32 |
| 4.1 OPNET 网络仿真平台 | 32 |
| 4.1.1 网络仿真技术 | 32 |
| 4.1.2 Modeler 概述 | 34 |
| 4.1.3 Modeler 仿真原理 | 35 |
| 4.1.4 Modeler 建模方法 | 37 |
| 4.1.5 OPNET 通信机制 | 38 |
| 4.2 网络层建模 | 39 |
| 4.3 节点层模型 | 40 |
| 4.3.1 航空电子子系统模型 | 40 |
| 4.3.2 端系统模型 | 40 |
| 4.3.3 交换机模型 | 42 |
| 4.4 进程层建模 | 43 |
| 4.4.1 协议栈的进程层建模 | 43 |
| 4.4.2 虚拟链路的进程层建模 | 44 |
| 4.4.3 交换机的进程层建模 | 46 |
| 4.5 本章小结 | 46 |
| 第五章 仿真与结果分析 | 48 |
| 5.1 仿真网络 | 48 |
| 5.2 仿真参数设置 | 49 |
| 5.3 网络仿真与数据采集 | 50 |
| 5.3.1 单一链路工作 | 50 |
| 5.3.2 多链路同等优先级工作 | 52 |
| 5.3.3 多链路不同优先级工作 | 54 |
| 5.4 本章小结 | 58 |
| 第六章 结论与展望 | 60 |
| 参 考 文 献 | 62 |
| 致 谢 | 65 |
| 攻读硕士学位期间已发表或录用的论文 | 67 |

图 录

| | |
|--------------------------------|----|
| 图 2-1 航空电子系统 | 7 |
| 图 2-2 AFDX 网络系统架构 | 8 |
| 图 2-3 流量规整 | 10 |
| 图 2-4 流量调度 | 10 |
| 图 2-5 UDP 报头格式 | 11 |
| 图 2-6 IP 报头格式 | 12 |
| 图 2-7 IP 单播地址格式 | 12 |
| 图 2-8 IP 多播地址格式 | 13 |
| 图 2-9 AFDX 帧格式 | 13 |
| 图 2-10 AFDX 网络时延构成 | 14 |
| 图 2-11 AFDX 网络时延测量点 | 15 |
| 图 2-12 冗余网络 | 17 |
| 图 2-13 接收端处理过程 | 18 |
| 图 3-1 FIFO 调度方式 | 20 |
| 图 3-2 RSP 算法流程图 | 24 |
| 图 3-3 示例网络的拓扑结构 | 29 |
| 图 3-4 不同调度算法下的调度示意图 | 30 |
| 图 4-1 OPNET MODELER 仿真步骤 | 35 |
| 图 4-2 离散时间仿真流程图 | 36 |
| 图 4-3 网络层拓扑 | 39 |
| 图 4-4 航空电子子系统模型 | 40 |
| 图 4-5 端系统模型 | 41 |
| 图 4-6 交换机模型 | 42 |
| 图 4-7 UDP/IP/MAC 层进程模型 | 44 |
| 图 4-8 虚拟链路配置 | 45 |
| 图 4-9 虚拟链路进程模型 | 45 |
| 图 4-10 交换机配置表 | 46 |
| 图 4-11 交换机进程模型 | 46 |

| | |
|----------------------------------|----|
| 图 5-1 仿真网络拓扑结构 | 48 |
| 图 5-2 OPNET 参数设置 | 50 |
| 图 5-3 单链路工作时数据帧平均时延 | 51 |
| 图 5-4 FIFO 方法下时延变化情况 | 52 |
| 图 5-5 RM/EDF/RSP 方法下时延变化情况 | 53 |
| 图 5-6 FIFO 方法下时延变化情况 | 54 |
| 图 5-7 RM 方法下时延变化情况 | 55 |
| 图 5-8 EDF 方法下时延变化情况 | 55 |
| 图 5-9 RSP 方法下时延变化情况 | 55 |
| 图 5-10 不同调度方法下最大时延 | 55 |
| 图 5-11 不同调度方法下时延抖动 | 55 |

表 录

| | |
|----------------------------|----|
| 表 1-1 机载数据总线比较 | 2 |
| 表 3-1 示例网络的历史信息 | 29 |
| 表 3-2 不同调度方法下的调度间隔 | 31 |
| 表 5-1 虚拟链路参数配置表 | 49 |
| 表 5-2 虚拟链路配置表 | 52 |
| 表 5-3 平均时延量化表（单位:ms） | 53 |
| 表 5-4 虚拟链路配置表 | 54 |

第一章 绪论

1.1 引言

自 1903 年莱特兄弟发明了人类历史上第一架飞机“飞行者”号以来，在一个多世纪的发展历程中，航空技术逐渐成熟，并与电子、机械、测控、材料等技术紧密结合，为人们提供了更加便捷、舒适、安全的空中旅行服务。其中，航空电子系统无疑是现代航空技术的灵魂，航空电子系统承担了导航、通信、监测和管理等职责，控制着飞机的方方面面，航空电子系统的正常、稳定工作是飞机能够正常飞行的重要保障。

随着人们对于航空服务需求的不断增加和航空电子系统的不断发展，新一代航空技术必须要解决航空电子子系统之间数据类型复杂、数据传输量大、实时性要求严格、突发性强的问题。而传统的航空总线由于传输带宽低、造价高等因素已经无法胜任现代航空系统的需求。传统的航空总线标准主要有 ARINC429^[1]和 Mil-std-1553B^[2]，这几种机载数据总线经过严格的设计和测试，满足了当时航空系统的需求，但是随着数据量的进一步增加，其缺点逐渐显露。ARINC429 总线协议是美国电子工程委员会于 1977 年 7 月提出并获得批准使用，ARINC429 仅有 100KB/s 的数据传输量，同时由于采用点对点的拓扑结构，各个子系统之间相互传输数据需要多条数据总线，线缆的数量和重量大大增加，布线难度也非常复杂。Mil-std-1553B 是美国军方提出的指令/响应型传输协议，最高速度可达到 1MB/s，由于使用总线控制器对整个总线通信进行控制，集中式的总线调度影响了总线的灵活性，不适合子系统之间的大量数据传输。

在这样的背景下，空客公司在其最新的 A380 飞机上开发了一种全新的全双工交换式以太网传输标准 AFDX^{[3][4]}（Avionics Full Duplex Switch Ethernet），并收录于 ARINC664 第七部分中。AFDX 基于传统以太网模型，使用全双工链路进行数据传输，并采用虚拟链路（virtual link）对航空电子子系统间的通信链路进行逻辑隔离。AFDX 有别于传统的机载航空总线，它并不是一种总线标准，而是一种网络协议，使用 TCP/IP 协议栈进行数据包处理。在 ARINC429 中，发送机可连接的接收机数目为 20，而在 AFDX 中，最高可达到 1024 个端系统。同时，ARINC429 使用单工方式进行数据传输，若需要双向通信则需要两条物理链路，而 AFDX 是全双工链路，因此大大的减少了电缆的数量。AFDX 中没有总线控制器，采用星

型拓扑结构，避免了 Mil-std-1553B 总线中集中控制的问题。AFDX 标准可达到 100Mb/s 甚至更高的数据传输率，是传统航空总线的 100 倍以上，极大满足了现代飞机对于复杂航空电子子系统间数据通信和乘客对于娱乐设施的要求。

表 1-1 机载数据总线比较
Table1-1 Comparison of Avionics Data Bus

| 总线类型 | ARINC429 | Mil-std-1553B | AFDX |
|------|--------------|---------------|--------------|
| 传输速率 | 100kb/s | 2Mb/s | 100Mb/s |
| 网络拓扑 | 点对点 | 总线型 | 星型 |
| 通信方式 | 单工广播式 | 半双工 | 全双工交换式 |
| 应用 | A310, 波音 737 | F16, F18 | A380, 波音 787 |
| 节点规模 | 20 | 31 | 1024 |
| 布线难度 | 复杂 | 中等 | 简单 |
| 价格成本 | 较低 | 较高 | 较低 |

在网络通信中，链路的端到端时延（delay）和时延抖动（delay jitter）是衡量网络性能的重要参数。网络时延描述了数据包从发送机发出到接收机接收之间经历的时间，时延抖动则描述了网络时延的抖动情况。在数据传输时，网络时延是不可避免的，时延主要由以下几部分组成：传输时延、传播时延、调度时延、处理时延。其中，传输时延描述了从数据产生到从发送端口发送出去的时延，传播时延描述了数据包在链路上传输的时延，处理时延描述了数据包在交换机中进行存储转发的时延，调度时延则描述了发送端口接收数据包并将其发送出去经过的时延。传输时延、传播时延、处理时延一般均为固定时延，而调度时延是可变时延，取决于当前网络负载情况。时延抖动描述了数据包延迟的变化情况，时延抖动越小则相应的网络稳定性更高。如果网络发生拥塞，则响应的时延抖动会发生剧烈变化。对于对实时性要求高的任务而言，时延抖动必须限制在一定的范围内，从而保证网络的 QoS。

在航空电子系统中，存在多种类型的数据，有与飞机控制相关的传感器数据，也有乘客为满足娱乐需求而产生的多媒体数据，他们对于时延和时延抖动的要求不尽相同。这些数据中，控制相关的数据和报警信息影响飞机对于控制指令的响应和对紧急情况反应，是航空系统的关键数据，应该具有更高的传输优先级，其端到端时延和时延抖动需要低于特定的延迟上界。传统的 FIFO 调度方式下，数据包按照到达的先后次序进行调度，当网络中出现突发流量的情况下可能会造成关键任务传输时间长、时延抖动大的问题。因此，研究新的调度算法对于保证飞机中关键数据的及时有效传输具有重要的实际意义。

在本文中，我们使用 OPNET 平台进行网络仿真。OPNET 网络仿真平台是当前业界处于领先水平的网络技术仿真环境，可以用于设计和研究通信设备、协议，对其进行建模仿真。OPNET 平台提供强大的集成环境，减轻了编码和数据分析的工作量，被广泛的应用于各大公司、大学和研究机构中。

1.2 国内外研究现状

1.2.1 国外研究现状

自 2005 年 AFDX 网络协议作为新一代航空总线标准被收录入 ARINC664 part7 后，各国研究机构、大学和公司都对 AFDX 网络协议进行了广泛深入的研究。目前，国外对于 AFDX 的研究主要集中为以下两个方向：1、建立网络模型，运用网络演算（network calculus, NC）^[5]方法对时延上界进行尽可能精确的估算；2、通过轨迹追踪技术分析网络中数据包的传输过程，分析网络时延的构成，计算相应的时延上界。

在时延上界估算方向，Marc Boyer^[25]等人在使用 FIFO 策略的基础上运用 NC 理论进行端到端时延上界的分析，通过对聚合流量和独立流量两种情形的分析，得到了更精确的时延上界估计。Ridouard^[27]等人将随机网络演算方法（stochastic network calculus）应用于 AFDX 网络，在调度时使用静态优先级策略，得到了异构数据流相应的端到端时延上界并比较了与不考虑链路优先级时得到的时延上界的不同，相对于不考虑优先级的情况，在异构流的情况下高优先级的链路拥有更小的时延上界。在网络性能优化方向，Henri Bauer^{[23][29]}等人应用轨迹追踪策略对数据包在网络中传输的过程进行追踪，通过对 AFDX 传输过程进行分析，得到最差情况的场景，分析最差情景下的包传输过程，得到精确的时延上界。这些研究主要集中于对 AFDX 网络时延上界的理论分析，缺乏对网络性能的优化。

1.2.2 国内研究现状

随着 AFDX 协议被收录入 ARINC664 标准第 7 部分，国内各大学、研究机构也紧跟国外步伐，开展了对 AFDX 标准协议的研究，研究主要集中于以下几个方面：AFDX 网络时延上界的精确计算，AFDX 网络的实现，调度算法与调度策略以及 AFDX 网络实时性、稳定性的研究。在 AFDX 网络时延上界计算方面，杨云、熊华刚^{[6][16]}分析了数据包在 AFDX 网络中的处理过程，运用网络演算方法得到了数据帧在数据接收、数据交换、数据过滤和端口存储转发过程中的延迟。在 AFDX

网络实现方面,姜丽云^[45]、宋东^[43]等人对 AFDX 关键技术进行研究并对 AFDX 网络建构、端系统和交换机进行建模仿真。在调度算法和调度策略方面,陈昕^[22]等人提出了一种静态优先级队列调度算法 AVLSP 方法,有效地降低了关键任务的时延抖动。在 AFDX 网络实时性、稳定性研究方面,陈文刚^[38]等人提出了一种自适应优先调度算法,较好地满足了航空电子数据的传输实时性要求。国内的研究成果较国外更偏向于工程实现,包括很多具有重要意义的研究成果,但是缺乏对 AFDX 网络中时延抖动的优化。

1.3 论文研究意义和研究内容

1.3.1 研究意义

目前,大型飞机已经成为提高国家自主创新能力和增强国家核心竞争力的重要标志。而机载航空总线在飞机系统中起到枢纽的重要作用,AFDX 网络标准作为新一代航空电子系统网络标准,已经成功应用于空客公司最新的 A380 飞机和波音公司的波音 787 飞机。我国最新自主设计、研制的大型国产客机 C919 目前也已经全面进入结构总装攻坚阶段。对航空电子系统关键部分航空数据总线的研究也具有重要的意义。现代航空系统数据类型复杂、数据传输量大、数据实时性要求严格,对航空数据总线的要求非常高。本文正是在这样的环境下开展对新一代航空数据总线 AFDX 的研究,通过在 OPNET 网络仿真平台对 AFDX 网络进行建模,对其关键特性进行分析,并设计新的重整形调度算法取代传统的调度方式,达到降低时延抖动并保障关键数据优先传输的目的。

1.3.2 主要内容

(1)AFDX 网络标准研究与性能分析

本文对 AFDX 网络标准进行了深入的研究,主要包括 AFDX 网络各部分组成与功能,AFDX 网络中虚拟链路机制和流量管理机制,AFDX 网络的冗余机制等。AFDX 网络由航空电子子系统、端系统、交换机和传输链路组成,并基于传统的 Ethernet 802.3/IP/UDP 协议栈对数据进行封装。同时,AFDX 网络中使用虚拟链路(virtual link)对各条数据流进行逻辑隔离。AFDX 网络中使用 BAG(Bandwidth Allocation Gap)对流量进行控制,限制了同一链路相邻两个帧之间的最小间隔,为了保证数据通信的稳定性和正确性,AFDX 使用双冗余网络进行通信,同一条逻辑链路将通过两条不同的物理链路进行传输,在接收端进行冗余管理和差错控

制。评价 AFDX 网络性能的指标主要有端到端时延和时延抖动，端到端时延描述了 AFDX 网络的实时性，由于不同链路对实时性的要求不一，因此在设计中需要分别考虑，时延抖动则描述了 AFDX 网络的稳定性，

(2)RSP 调度方法研究与实现

传统 AFDX 网络使用 FIFO 方式进行调度，FIFO 方式是一种公平、简单的调度方法，但是由于包到达次序的不确定性，FIFO 方式可能会导致链路时延抖动大，关键链路等待时间长的问题。为了解决这个问题，我们提出了一种基于重整形思想的调度算法，该算法将链路分为三类，赋予不同链路不同的静态优先级，飞行控制信息与警报信息具有最高优先级，乘客为满足娱乐需求产生的多媒体数据具有一般优先级，备份数据和文件传输等具有最低优先级。在系统中，设定一个调整周期，系统运行之后，统计一个调整周期内各条链路产生的包数量，同一优先级的链路之间根据调整周期内产生的包数量决定相互之间的优先次序，从而实现动态优先级。

(3) OPNET^[46]平台搭建与仿真

介绍 OPNET 仿真平台的仿真原理和建模方法，在 OPNET 平台之上搭建仿真网络，搭建网络层模型、节点层模型、进程层模型以及进程间的状态机模型。在仿真网络中指定各条链路注入速率，完成符合 AFDX 协议规范的包格式封装和解封装，完成冗余网络搭建和冗余管理、差错控制。在节点层中，预先指定调度器需要执行的调度策略。随后运行仿真，比较各种情况下不同调度算法对网络性能的影响。

1.4 论文结构

本文针对新一代航空电子系统网络协议 AFDX 进行仿真，并对 AFDX 网络中的调度策略进行了深入的研究。论文共分为 6 章，章节安排如下：

第一章 绪论。介绍本文的研究背景，国内外研究现状，论文研究意义和主要内容以及论文结构。

第二章 AFDX 协议与 AFDX 网络性能分析。主要介绍 AFDX 协议标准，包括 AFDX 协议中的流控机制、调度机制、冗余管理机制、差错控制机制，以及对 AFDX 网络中端到端时延和时延抖动进行分析，提取出影响端到端时延和时延抖动的因素。

第三章 重整形调度算法。介绍传统 AFDX 网络中使用的 FIFO 调度方式并提出一种新的基于重整形思想的调度方法 RSP 方法。分析比较 FIFO 方式和 RSP 方

法的执行过程，在理论方面比较两者的优点和缺点，并通过一个示例网络分析对比两者对网络性能的影响。

第四章 基于 OPNET 平台的 AFDX 网络建模。介绍 OPNET 仿真平台的仿真原理和建模方法，并在 OPNET 平台上对 AFDX 网络进行建模，建模主要针对端系统和交换机。建模层次包括网络层建模、节点层建模和进程层建模，以及对进程之间的状态机进行建模。在节点中，内置两种调度方法，通过特定变量指示调用何种调度方法。

第五章 仿真结果与分析。配置不同情况下的仿真参数，运行仿真，比较不同情况下不同调度算法的端到端时延、时延抖动的变化情况，并分析得到的结果，验证本文提出的新的基于重整形思想的调度方法。

第六章 结论与展望。对全文进行总结，并提出下一步工作的展望。

第二章 AFDX 网络与 AFDX 网络性能分析

在本章中,将介绍 AFDX 网络相关的背景知识和 AFDX 网络性能的评价指标,包括 AFDX 各组成部分介绍以及网络内使用的各种机制。

2.1 全双工交换式以太网 AFDX

航空电子系统 (Avionics system) 诞生于 20 世纪 70 年代,经历了漫长的发展道路,伴随着电子工业的发展逐步走向一体化。早期的航空电子系统采用分立式结构,系统由很多单个独立的子系统构成,每个子系统都依赖于驾驶员的操作,驾驶员从子系统中获得消息,保持对整个系统的控制。由于过分依赖于驾驶员,分立式结构逐渐被混合式结构取代。混合式结构是向综合式结构过渡的一种结构形态,具体表现为部分子系统之间的综合,例如计算机、高度表、空速表、温度传感器等的组合。各分系统之间通过广播式航空数据传输总线 (ARINC429) 连接。随着系统集成度的进一步提高,综合式结构逐渐取代混合式结构。综合式结构是美国 DAIS (“宝石柱”) 计划的主要成果,它通过 Mil-std-1553B 数据总线将大多数航空电子子系统连接起来,从而实现信息的统一调度。电子技术逐渐成为飞机关键部位的核心技术,如飞行控制系统等。航空电子子系统一般包括飞机控制相关子系统、信息处理相关子系统和外部扩展相关子系统,各个子系统之间通过航空数据总线进行连接,完整的航空电子系统结构如图 2-1 所示。

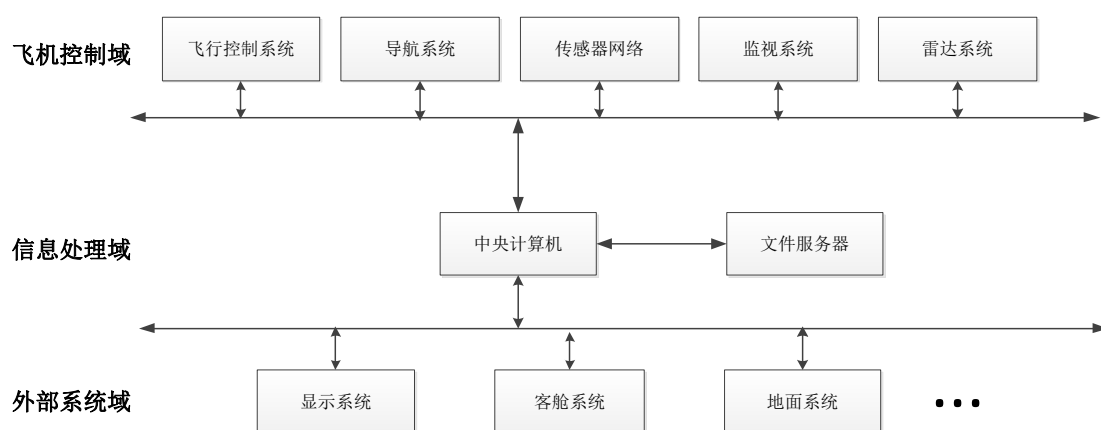


图 2-1 航空电子系统
Fig.2-1 Avionics system

现在，航空电子系统已经进入新一代更高程度的综合化结构时期。随着航空电子技术的不断发展，航空电子设备逐渐增加，系统的集成度更高，各个子系统之间需要传输的数据大量增加，人们对于航空服务的要求也在不断增加，乘客产生的多媒体数据也成为航空电子系统设计必须考虑的因素。在这样的背景下，空中客车（Air Bus）公司开始着手新一代航空电子系统传输网络的研究，提出了新一代的机载数据总线标准 AFDX。

AFDX（Avionics Full Duplex Switch Ethernet）标准定义于 ARINC664 part7，是新一代的机载航空总线标准，其全称为航空电子全双工交换式以太网。AFDX 网络基于商用以太网 IEEE 802.3 标准，同时对传统以太网进行改进使其成为适合航空电子系统数据传输的全双工确定性交换式以太网。AFDX 相对于传统的机载航空总线 ARINC429 和 Mil-std-1553B 来说，不仅提高了数据的传输率而且大量减轻了飞机上的电缆重量，提高了网络的扩展性和稳定性，由于采用了商用以太网的成熟技术，使 AFDX 网络更易于维护。

AFDX 网络的主要组成部分包括航空电子子系统、端系统、交换机和传输链路。网络系统架构如图 2-2 所示。

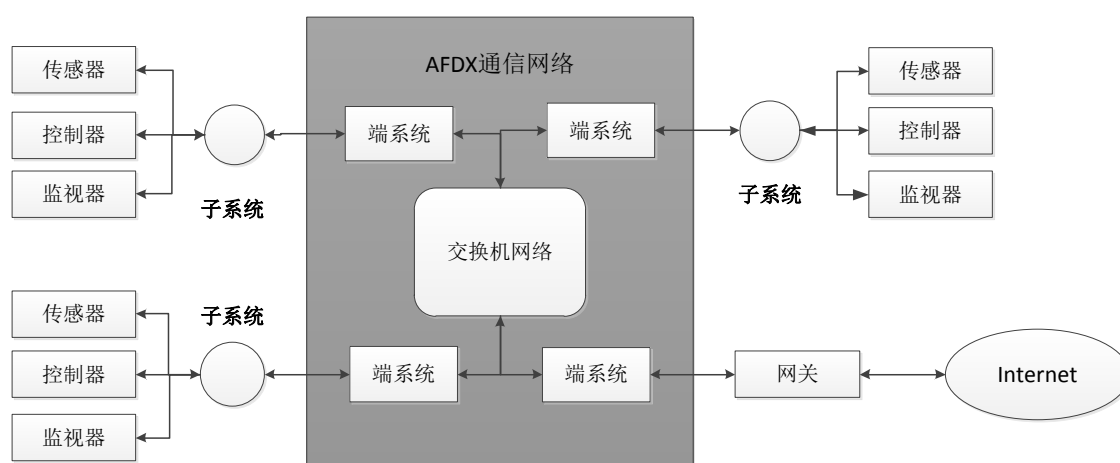


图 2-2 AFDX 网络系统架构
Fig.2-2 AFDX Network Architecture

（1）航空电子子系统。航空电子子系统是飞机上能够完成独立功能的小系统，包括通信系统、导航系统、显示系统、监测系统、飞行控制系统、雷达系统和传感器网络等。这些子系统通过航空数据总线或机载网络连接在一起，共同构成航空电子系统。

（2）端系统。端系统是连接航空电子子系统和航空电子系统通信网络的接口，

包括发送端系统和接收端系统。发送端系统接收航空电子子系统产生的数据，并对其进行符合 AFDX 各层协议规范的封装，然后按照虚拟链路编号将各个数据包放入对应的虚拟链路缓存中，在需要进行冗余管理的网络中，需要将数据包进行复制，放入对应输出端口的发送缓存中。在发送时刻，将数据包发送到传输链路。而接收端系统则接收来自通信网络的数据包，进行冗余管理和差错校验，将正确的数据包逐层向上递交，在经过各层协议处理，去掉协议包头之后，将数据递交给需要接收该数据的航空电子子系统。

(3) 交换机。AFDX 网络中，一个或多个交换机将各个端系统连接起来，和传输网络一起构成航空电子系统通信网络。交换机主要完成过滤与管制、交换、监视等功能。交换机的存在也是 AFDX 网络与传统航空数据总线 ARINC429 和 Mil-std-1553B 最大的区别，表明航空数据总线已由点对点连接和总线型转变为交换式。

(4) 通信链路。通信链路连接 AFDX 网络中的端系统和交换机，采用全双工方式通信，速率可达到 100Mb/s（最高可达 1000Mb/s）。

2.2 虚拟链路

虚拟链路（virtual link, VL）是一个抽象的通信对象，具有如下的属性：

- VL 定义了一个逻辑上的单向连接，数据帧从一个源端系统发出，经过 AFDX 通信网络到达一个或多个目的端系统。
- 每条 VL 都被指定一个最大带宽，在系统运行开始前分配。
- 对每条逻辑链路，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。

每条虚拟链路都被指定一个最大带宽，与某条虚拟链路相关联的流量用以下两个参数来描述：带宽分配间隔（Bandwidth Allocation Gap, BAG）和抖动（Jitter）。BAG 反映了同一条虚拟链路中两个相邻的帧的起始位之间的最小时间间隔。各条虚拟链路的 BAG 存储在端系统配置表中，BAG 的取值范围应该在 1ms 到 128ms 的范围内，且为 2 的整数幂，如下所示：

$$BAG = 2^k (0 \leq k \leq 7) \quad (2-1)$$

虚拟链路 VL_i 的 BAG 和链路最大包长度 L_{max} 决定了链路的最大带宽 B_{max} ，最大带宽的计算公式如下：

$$B_{max, i} = L_{max} \times 8 / BAG \quad (2-2)$$

如果某条链路的 BAG 为 8ms，最大包长度 L_{max} 为 1024Byte，则该链路的最

大带宽为 1,024,000bit/s。

为了保证每条虚拟链路的 BAG，需要对每条链路进行流量规整。如图 2-3 所示：

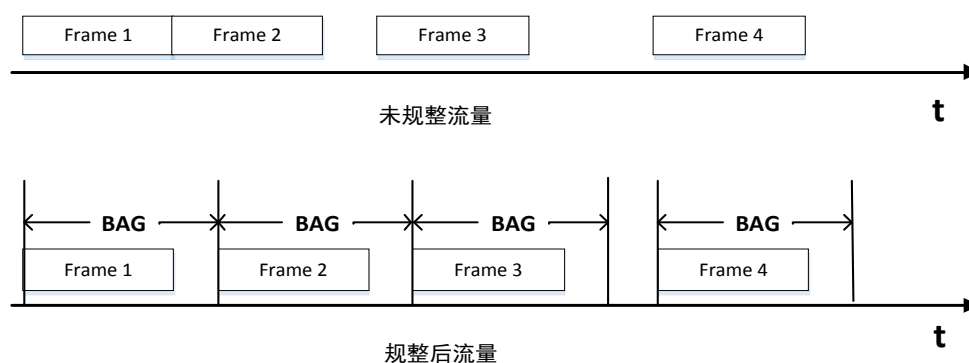


图 2-3 流量规整
Fig.2-3 Traffic Shaping

如果一个端系统中存在多条虚拟链路，则调度器对各条经过规整的虚拟链路进行调度，实现多路复用。各条虚拟链路的数据帧分别经过流量规整后得到固定间隔的数据帧流，这些数据帧流是不存在 jitter 的，后经过调度器处理，将存在冲突的数据帧安排相应的次序，引入 jitter，随后同一条物理链路上进行传输，如图 2-4 所示：

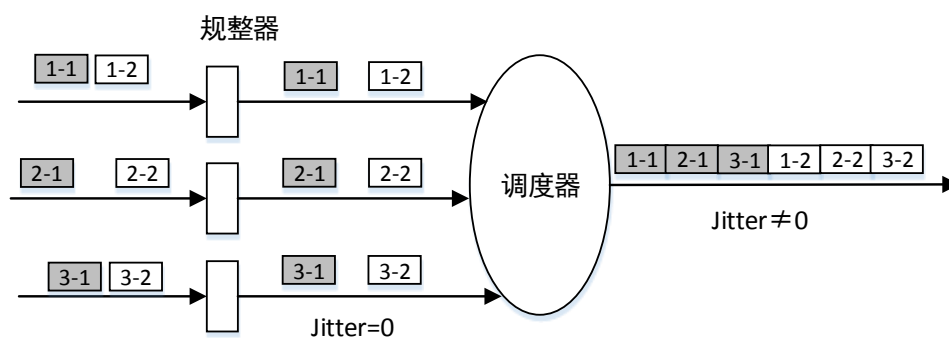


图 2-4 流量调度
Fig.2-4 Traffic Scheduling

对于给定的虚拟链路，帧需要在一个有界的时间间隔内出现，这个时间间隔被称为最大允许抖动（maximum admissible jitter）。端系统的最大允许抖动需满足公式 2-3：

$$\begin{cases} \max_jitter \leq 40\mu s + \frac{\sum_i^n (20 + L_{max,i}) \times 8}{c} \\ \max_jitter < 500\mu s \end{cases} \quad (2-3)$$

其中, c 为物理链路带宽, n 为端系统中虚拟链路数量, $L_{max,i}$ 为第 i 条虚拟链路的最大包长度。

在 AFDX 网络端系统的设计中加入带宽分配间隔 BAG 和抖动 jitter 是为了将链路的流量限制在一定范围内, 同时, 保证帧不会在超过最大允许抖动的时刻传输, 从而保障了 AFDX 网络的确定性。

2.3 AFDX 协议栈

AFDX 网络中, 采用 Ethernet 802.3/IP/UDP 协议栈完成数据封装、指定源端地址和目的端地址、进行数据分组、添加对应包头。AFDX 网络的包头格式在传统 TCP/IP 包头格式的基础上进行了小幅度改动, 使其符合 AFDX 网络的要求。

AFDX 网络中端到端通信的标识方式为: 源 UDP 端口号+源 IP 地址+目的 MAC 地址 (虚拟链路号)+目的 IP 地址+目的 UDP 端口号。每个数据包都由一个这样的五元组唯一指定了一条通信路径。

AFDX 网络中 UDP 封装格式与传统以太网相同。包头包括 4 个域: 源端口号、目的端口号、数据包长度、校验和。包头格式如图 2-5 所示:

| | |
|----------|----------|
| 16位源端口号 | 16位目的端口号 |
| 16位数据包长度 | 16位校验和 |
| 报文 | |

图 2-5 UDP 报头格式
Fig.2-5 UDP Header Format

AFDX 网络中, 网络层完成路由选择和标识通信主机的功能。IPv4 的帧结构如图 2-6 所示:

| | | | | |
|-----------------|--------------|--------------|----------------|---------------|
| 版本 4bit | 头部长度 4bit | 服务类型 8bit | 总长度 16bit | |
| 片段标识 16bit | | | 控制符 3bit | 片段偏移 13bit |
| 生存时间 8bit | | 协议 Bit | 头部校验和 16bit | |
| 源IP地址 32bit | | | | |
| 目的IP地址 32bit | | | | |
| 报文（1~1479B） | | | | |

图 2-6 IP 报头格式
Fig.2-6 IP Header Format

其中：

- IP 源地址被用以标识与端系统相关联的发送分区。
- IP 目的地址用以向一个或多个目的端系统转发 IP 数据包。
- IP 地址应为 A 类地址，且为私有网络单播地址（前 8 个 bit 应为“0000 1010”）。

IP 单播地址格式如图 2-7 所示：

| IP单播地址格式（32bit） | | | | |
|-----------------|--------------|-----------------------------------|----------------------------|------|
| A类 1bit | 私有地址 7bit | 用户自定义标识（User_defined_ID） 16bit | 分区标识（Partition_ID） 8bit | |
| 0 | 000 1010 | xxxx xx xx xx xx xx | 空余域 3bit | 5bit |

图 2-7 IP 单播地址格式
Fig.2 -7 IP Unicast Address Format

AFDX 网络中 IP 报头头部的 IP 源地址应该是 IP 单播地址，用来标识发送器。

用户自定义标识是一个单独的 16bit 的位域，被用来给每个 IP 可寻址的主机分配一个唯一的且有意义的 IP 地址。

分区标识由两个位域构成：

空余域的 3bit 一般设为 0，后面的 5bit 可能被用来标识具有超过 32 个分区情况下的分区。

AFDX 网络中数据帧头部的目的 IP 地址应该是：

- 单播地址，地址格式与图 2-7 相同。
- 多播地址，地址格式与图 2-8 相同。

| IP多播地址格式（32bit） | | |
|-----------------|------------------------------|----------------|
| 4bit | 28bit | |
| D类地址 1110 | 多播地址标识符 | |
| | 固定域（12bit） 0000 1110 0000 | 虚拟链路标识符（16bit） |

图 2-8 IP 多播地址格式
Fig.2-8 IP Unicast Address Format

AFDX 网络中，为实现冗余网络，在帧中加入了一个顺序号字段，AFDX 的帧结构如图 2-9 所示：

| AFDX帧格式 | | | | | | | |
|-------------|------------|----------------|----------------|----------------|------------------------|-------------|-----------|
| 6Byte | 6Byte | 2Byte | 46-1500Byte | | | | 4Byte |
| 目的MAC 地址 | 源MAC 地址 | 0x0800 IPv4 | IP包头 20Byte | UDP包头 8Byte | AFDX有效载荷 1-1471Byte | SN 1Byte | 帧校验序 列 |

图 2-9 AFDX 帧格式
Fig.2-9 AFDX Frame Format

在实际传输中，数据帧前还会加 7Byte 的前导字和 1Byte 的起始定界符，共同构成 8Byte 的前导码。前导码的格式为：AAAA AAAA AAAA AB(16 进制)。前导字的主要作用是进行同步。在数据帧传输完成后，不立刻传输下一个数据帧，而是增加 12Byte 的帧间间隔，防止接收端来不及接收，造成错误。

2.4 网络时延和时延抖动

2.4.1 网络时延

网络时延和时延抖动是衡量网络性能的重要参数，数据包从在发送端产生到被接收端接收所经历的时间称为时延，每个数据包在网络中进行传输都必须经历

一定的时间，时延是不可避免的。

在 AFDX 网络中，端到端时延的构成包括：发送时延、传播时延、交换时延和接收时延。时延构成如图 2-10 所示：

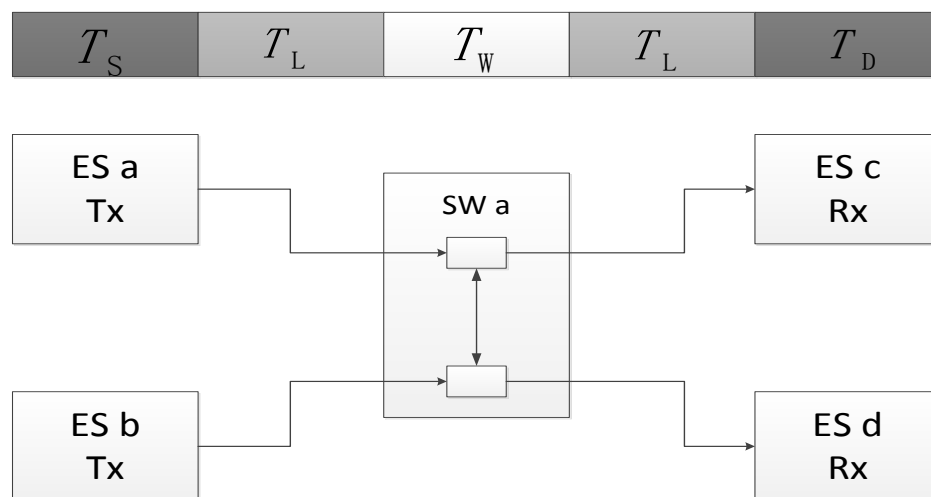


图 2-10 AFDX 网络时延构成
Fig.2-10 AFDX Delay Component

其中， T_S 为发送时延， $T_S = T_{S1} + T_{S2}$ ， T_{S1} 为技术时延， T_{S2} 为调度时延；

T_L 为链路传播时延，与信号传播速度和距离有关；

T_W 为交换机转发时延， $T_W = T_T + T_{SW}$ ， T_T 为交换机的技术时延， T_{SW} 为交换机转发时延；

T_D 为接收时延。

AFDX 网络中，发送时延被定义为下列两个测量点之间所经历的时间：

- 开始点：上层分区数据的最后 1bit 对于端系统的通信服务可用。
- 接收点：封装完成的以太网帧的最后 1bit 被发送到物理链路上。

接收时延被定义为下列两个测量点之间所经历的时间：

- 开始点：以太网帧的最后 1bit 在物理链路上被接收端接收到。
- 结束点：解封装得到的原始信息的最后 1bit 对于上层分区可用。

发送时延和接收时延测量点如图 2-11 所示：

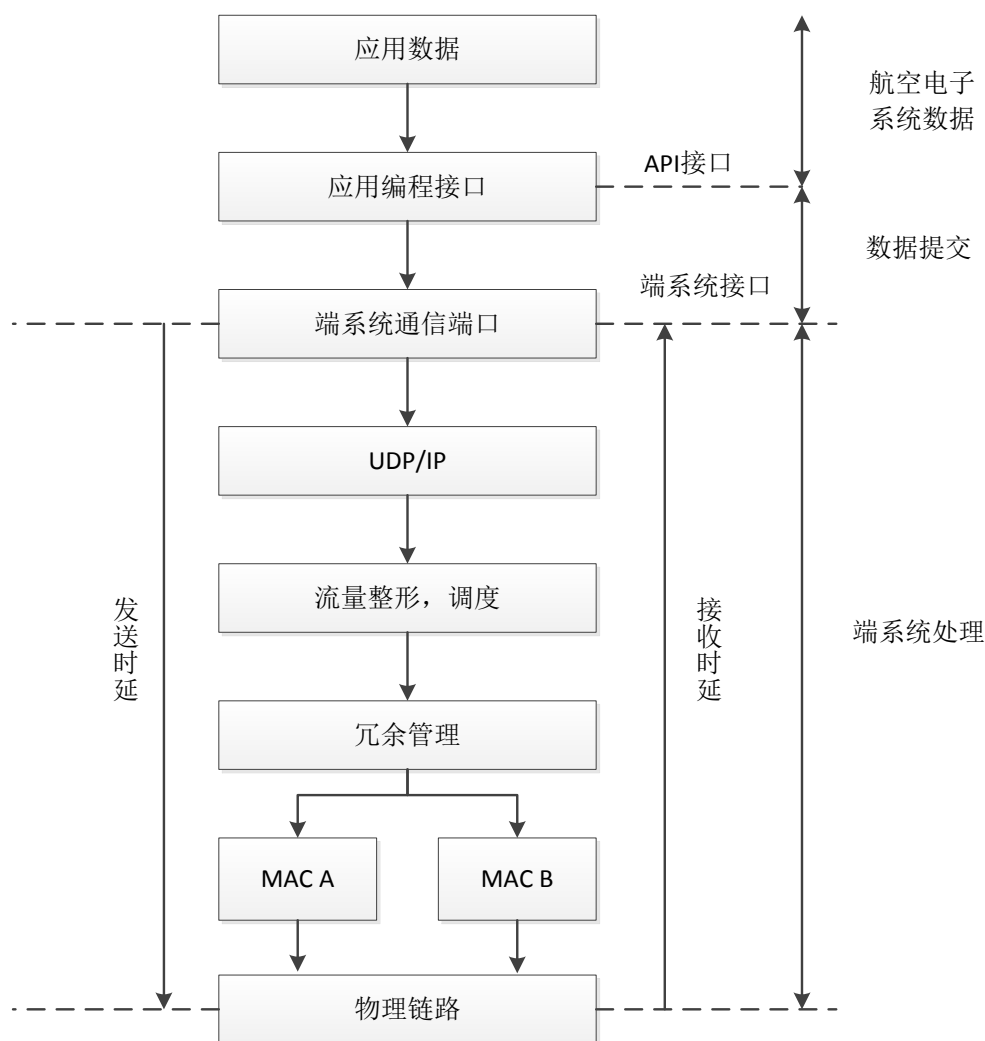


图 2-11 AFDX 网络时延测量点
Fig.2-11 AFDX Delay Detect point

一般情况下，时延包括两个部分：固定部分和可变部分。在 AFDX 网络中，发送时延的固定部分是指独立于流量负载的技术时延，技术时延被定义为在没有其他任务进行处理的时候，端系统接收和处理应用数据，最后发送到物理链路上所需的时间。发送时延的可变是指依赖于流量和配置的时延，主要在调度时刻形成，当多条虚拟链路同时处于可调度状态时，先得到调度的链路的可变时延小于后得到调度的链路。AFDX 网络标准中规定，发送时延的上界是 $150\mu\text{s} + \text{帧延迟}$ ，帧延迟指端系统将帧发送到物理链路所花费的时间。

传播时延主要由链路数据传输率和距离相关，传播时延=物理链路的长度/物理链路的信号传播速度。

交换机转发时延包括交换机技术时延和排队转发时延，其中，技术时延属于

固定时延，排队转发时延属于可变时延。技术时延包括将数据包接收、过滤、放入对应输出缓存所经历的时间。排队时延是指数据包存入交换机输出缓存到离开交换机所经历的时间。

端系统的接收时延一般包括接收的技术时延和进行冗余管理的时延。接收的技术时延是指没有其他任务进行处理的时候，端系统接收和处理应用数据，最后提交给上层应用所需的时间。进行冗余管理的时延是指接收来自两个冗余网络的帧，选择其中先到达的正确的帧进行接收。

2.4.2 时延抖动

时延抖动 (delay jitter)，指由于网络拥塞或排队等因素造成每个数据包到达接收端所经历的时间的变化情况。网络传输是一个复杂的过程，需要经历不同的链路和交换机，当数据离开发送端经过交换机交换转发后，在发送端经过流量整形形成的相邻数据包之间的均匀间隔被破坏了，有的数据包经历的排队时间短，有的数据包经历的排队时间长，从而形成不规则的时延变化曲线。时延变化越大，时延抖动也越大。

时延抖动描述了同一条链路的数据包端到端时延的变化情况。不同于网络时延，时延抖动是可以避免的，时延抖动越小，代表网络越稳定。时延抖动是衡量网络实时性和 QoS 的重要参数。对于一些多媒体应用，时延抖动的约束比网络时延的约束要更强，多媒体数据时延抖动大会造成声音或画面颤抖，影响用户体验。在多媒体数据传送过程中，相邻数据包到达的时间差会发生差异。在 RTP 协议中，当一个数据包发送时，发送端在 RTP 报文头部增加一个时间戳，当在目的端被接收时，目的端也同样增加一个时间戳；计算这两个时间戳就可以得到数据包的延迟时间。若在传输过程中，该延迟时间不同，则链路存在抖动。在视频应用中，抖动表现为图像闪烁；在语音服务中，抖动会导致声音不连续和不清楚。抖动的大小取决于数据包的延迟时间的差异程度。

AFDX 网络中，时延抖动是指同一条虚拟链路上的数据包端到端时延的变化情况。航空电子系统内包括大量的传感器，如温度传感器、高度传感器、压力传感器、迎角传感器等，构成传感器网络，传感器周期性采集数据，并发送到主控计算机，这些数据不仅需要低时延，同时需要低延迟抖动，若网络时延抖动过大，则可能出现紧急数据无法及时获取的情况。而且，现在越来越多的娱乐应用出现在航空电子系统中，这些多媒体数据对时延的要求相对小，但是需要保证低延迟抖动，从而保证声音和画面的流畅。可以看出，有效降低时延抖动也是设计

AFDX 网络时必须考虑的方面。

2.5 冗余网络

航空电子系统的工作环境相对恶劣，极有可能出现网络组件的失效，为了防止因单个网络组件的失效而引起的网络瘫痪，例如：一条物理链路或一台交换机的失效，数据流可以得到保护，AFDX 网络采用了双冗余网络的机制，端系统之间的通信在两个相互独立的冗余网络中进行，从而极大的降低了网络的失效率。

AFDX 中的冗余网络如图 2-12 所示：

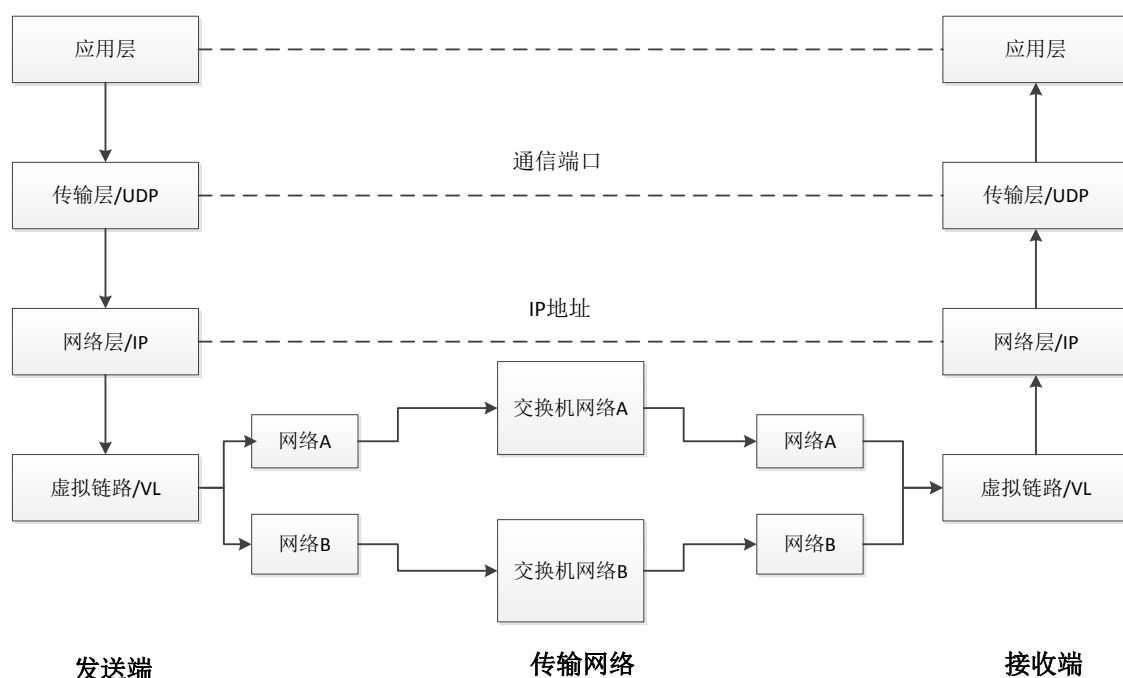


图 2-12 冗余网络
Fig.2-12 Redundant Network

AFDX 网络中使用冗余网络工作的机制如下：

发送端系统接收上层传来的数据，经过通信协议栈对其进行封装，每个帧都会被分配一个顺序号（sequence number），同一条虚拟链路上相邻帧的顺序号是递增的，封装完成之后，复制封装好的数据帧，将这两个数据帧加入两条冗余网络对应的输出缓存中，在图 2-7 中，为网络 A 和网络 B。随后两个具有相同顺序号的帧在网络 A 和网络 B 中独立传输，当到达接收端时，接收端需要在向上层递交数据之前重新构造出一条有次序的单一帧流，去除冗余复制的帧。上层分区无法感知到底层的冗余网络，即图 2-7 中的应用层、传输层、网络层只接收一个来自

底层的数据帧，而无需担心冗余。

在冗余网络中，每个帧被发出并且同时通过两个网络。接收端的冗余管理策略是“先到且正确的帧胜出”，在接收端维护一个序号记录当前接收到的最新序号，当物理链路传输来的帧经过完整性校验后，则接收该帧，并将接收的序号记录为新接收帧的序号。另一个网络中传送的相同序号的帧到达接收端时，将被丢弃。接收端处理过程如图 2-13 所示：

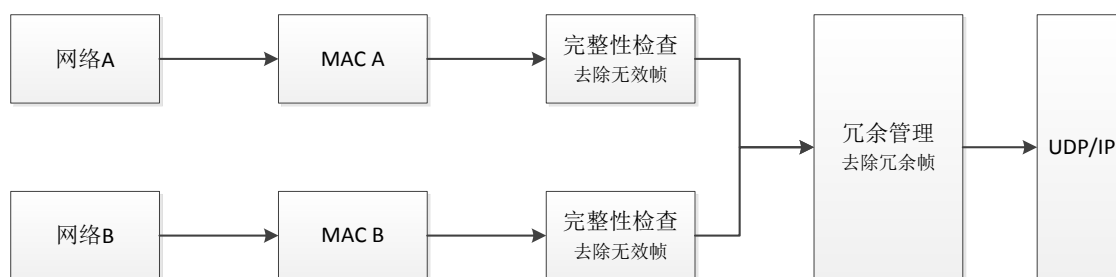


图 2-13 接收端处理过程
Fig.2-13 Process Procedure in Receiver

在 AFDX 网络中，端系统对每个发送帧都分配一个 8bit 的序号，取值范围为 0 到 255。对于每条虚拟链路，序号的初始值被设置为 0，当发送端复位之后，该值也被置为 0。对于同一条虚拟链路上相继传输的帧，帧的序号递增加 1，并且在递增到 255 之后回卷到 1。

在接收端系统的设计中，完整性检测在冗余管理之前进行，完整性检测完成差错校验和帧区间检查，去除无效帧，将正确的帧提交给冗余管理部件。接收端对每个帧进行检查，如果帧序号处于区间 $[SN + 1, SN + 2]$ 时，则接收该帧。其中，SN 为当前已接收到的最新的帧序号，期望接收的是序号为该序号加 1 或加 2 的帧。设置这个区间是为了提高网络的鲁棒性，因为 AFDX 网络的工作环境较为恶劣，丢失单个帧被认为是正常的事件，同时也消除了单个非正常帧对网络的影响。

冗余网络的加入增加了 AFDX 网络的稳定性和可用性，但是同时也增加了线缆的数量和交换机的数量，在实际应用中可选择是否采用冗余机制。

2.6 本章小结

本章介绍了 AFDX 网络的基本概念并从时延和时延抖动方面分析了 AFDX 网络的性能。

首先，介绍了航空电子系统的发展历程和 AFDX 网络的基本组成以及各部分组件的作用。其次，介绍了 AFDX 网络中的虚拟链路，虚拟链路是对特定数据流的逻辑表示，同时介绍了 AFDX 网络中流量整形和调度的相关知识。接下来，从时延和时延抖动角度分析了 AFDX 网络的性能。最后，本章对 AFDX 网络中的冗余网络进行了介绍，分析了冗余网络的必要性以及实现方式。

第三章 重整形调度算法

本章主要介绍 AFDX 网络中的调度算法，针对传统 FIFO、RM、EDF 调度算法存在的问题提出一种新的重整形算法，介绍了该算法的思想和实现过程并分析比较了重整形算法与传统调度算法的优缺点。

3.1 传统调度算法

网络通信中，调度器是规整流量，进行多路复用的部件。一个端系统或交换机中，一般存在多条数据流，这些数据流在逻辑上相互独立，物理上共享同一条物理链路进行传输。当同时有多条链路到达就绪状态时，调度器根据内置的调度算法选择其中一条链路的数据包进行发送。调度算法是调度器的核心，影响链路的时延和时延抖动。一个优秀的调度算法可以在不增加时延的情况下尽可能减小链路的时延抖动。

在 AFDX 网络中，由于每个端系统可挂载多个航空电子子系统，因此在同一个端系统中存在多条虚链路，每一条虚链路传输一种类型的数据，多条虚链路之间相互独立，系统通过调度器对多条虚链路进行多路复用。传统的调度算法是先进先出（First In First Out, FIFO）算法，FIFO 方式下，系统为每条物理链路维护一个缓存队列，存放待发送的数据包。每个数据包经过整形后到达输出端口，直接放入对应物理链路的缓存队列尾部，等待发送，早到达的数据包将在晚到达的数据包之前发送。

FIFO 方式严格按照帧到达的时间进行调度，如图 3-1 所示：

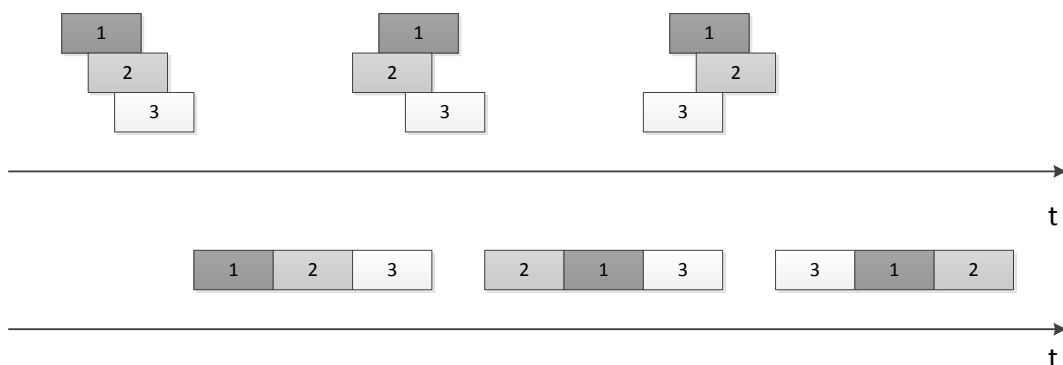


图 3-1 FIFO 调度方式

Fig.3-1 FIFO Scheduling Method

FIFO 方法是一种简单有效的调度方法，实现方式简单，可靠性高，并且充分保证了各条链路的公平性。但是，FIFO 方式会引起链路的时延抖动大、关键数据传输无法得到保证。特别是对于与飞行控制相关的关键数据，必须保证其实时性，因此我们提出了一种新的基于重整形思想的调度算法来降低传输链路的时延抖动，并保证关键任务的优先传输。

3.2 重整形算法

3.2.1 常用整形与调度算法

(1) 常用整形算法

在网络通信中，漏桶算法（Leaky Bucket）和令牌桶算法（Token Bucket）是常用的流量整形算法。

漏桶算法通过控制数据注入到网络的速率，平滑网络上的突发流量。从而对流量进行整形，保证网络不会因为某一段时间的突发流量形成拥塞，造成某些数据帧丢失。

令牌桶算法通过维护一个缓冲区，缓存待发送的数据帧，当网络中的令牌足够时，才转发缓存中的数据帧，每发送一个数据帧，令牌减少一定的数量。随着时间的推移，系统匀速的产生令牌。通过对令牌的控制，能够防止突发的大流量传输。

漏桶算法通过设定固定的流出速率来限制网络的流量，因此在网络负荷很低的情况下，漏桶算法对网络资源的使用率低。即使网络中流量很小（不出现拥塞），漏桶算法仍然按照固定速率对数据帧进行转发，为网络提供了非常稳定的流量。

令牌桶算法与漏桶算法不同，令牌桶算法能够在限制网络流量的同时允许一定程度的突发传输。当系统中，网络传输速率小于令牌生成速率时，系统会积攒一定数量的令牌，当出现突发流量时，这部分令牌将允许一定的突发流量。算法设计者可设置令牌的上限值，从而限制这部分突发流量的大小。

令牌桶算法和漏桶算法对所传输的数据不敏感，它们只能够对单条链路的通信流量进行控制，当网络中存在多条链路时，需要对每一条链路进行流量整形，然后通过调度器来选择要发送的数据帧。

(2) 常用调度算法

为了确保实时处理系统的正确性和安全性，特别是航空航天、电信、国防等领域对于系统的实时性有着强烈的需求，人们在实时任务调度理论和实时通信等

领域进行了大量的研究并取得了显著的成果。

实时任务调度理论是实时处理技术的核心，每个实时任务都有响应的时限要求，在进行任务调度时，需要判断任务能否在其截止时间内完成。由于实时系统的侧重点不同，调度方式也分为多种类型。根据任务实时性要求的重要程度，分为硬实时调度和软实时调度。根据调度算法和可调度性判定在任务运行之前还是运行期间进行，可分为静态调度和动态调度。根据被调度任务是否可被抢占，分为抢占式调度和非抢占式调度。根据任务请求的周期分为周期任务调度和非周期任务调度。不同调度方式具有不同的优缺点，适用于不同的场景。

目前，应用较多的实时调度算法主要包括：速率单调（Rate Monotonic, RM）算法和最早截止时间优先（Earliest Deadline First, EDF）算法。

RM 算法是一种适用于周期性任务调度的静态优先级调度算法。它根据任务的周期来分配优先级，周期越小，则优先级越高。RM 算法基于建立在一系列理想假设基础上的理想调度模型，在实际应用中，需要对这些假设进行一定修改。

RM 调度算法理想的假设模型包括以下假设：

- 1、所有的任务都是周期性的，必须在限定时间内完成。
- 2、任务的作业需在任务的下一个作业到来之前完成。
- 3、每个任务之间相互独立，不存在前后次序的约束。
- 4、任务的运行时间不变，指任务的每个作业需要的处理时间相同。
- 5、非周期性任务在特殊情况下发生，如系统初始化或系统紧急处理程序。
- 6、其他假设，包括任务切换时间忽略不计、可抢占调度等等。

EDF 算法是一种动态优先级任务调度算法。它根据当前任务的截止周期来为任务动态分配优先级，任务的截止时间越小，则任务的优先级越高，相反，任务的截止时间越长，则任务的优先级越低。在 EDF 调度算法中，每个任务执行完毕后，选择就绪任务中优先级最高的任务进行执行。

RM 算法输出静态算法，算法的额外开销小，可预测性好。但是由于静态调度算法一旦做出调度决定后，在整个运行期间就无法再进行更改，因此调度的灵活性不如动态调度算法。EDF 算法是动态调度算法，根据任务的实时情况进行调度选择，增强了系统的灵活性，能够应对突发情况。但是，EDF 算法的运行开销较大且可预测性差。

3.2.2 AFDX 网络中的重整形调度算法

AFDX 网络中，通过整形器对各条虚拟链路的流量进行控制。由于在链路中

加入了带宽分配间隔 (BAG), 因此同一条链路相邻两个帧之间至少会相隔一个 BAG 进行传输。可以看出 AFDX 网络中的流量整形器是对漏桶算法的一种改进, 漏桶算法以固定速率控制网络的流量, 控制流量的单位是每秒流出的 Byte 数, AFDX 网络中, 每条虚拟链路的数据帧虽然不是以固定速率发送帧, 但是规定了数据帧的最大速率, 控制流量的单位是每秒流出的数据帧的数目。同时, 在网络中加入了最大允许抖动 (jitter) 从而保证数据帧不会在输出缓存中等待太久, 防止数据帧“饿死”。

在进行流量整形, 给每条链路分配 BAG 和 Jitter 后, AFDX 网络通过调度器对多条虚拟链路进行复用。传统的 FIFO 方式在调度时刻按照帧到达的先后顺序进行调度, 无法保障关键链路的实时性传输, 同时由于网络传输的不确定性, 不同时刻不同链路上的数据帧到达的先后顺序可能有差异, 从而导致链路的时延出现变化, 形成时延抖动。

针对这个问题, 我们设计了一种新的调度算法: 重整形调度算法 (Re-shaping Method)。重整形算法基于优先级调度策略, 通过对链路的传输次序进行动态规划, 保证在一定的时间内, 链路的传输次序不会改变。

重整形调度算法中, 我们首先对航空电子系统中的数据进行分类, 赋予不同的优先级。飞行控制数据、导航数据、传感器数据、警报信息等数据由于与飞机的正常飞行和安全相关, 属于关键数据, 具有最高的优先级。乘客产生的多媒体数据, 需要一定的实时性, 将次于关键数据进行传输。一些备份数据和文件数据, 对实时性的要求低, 将分配最低的优先级。重整形算法为每条链路设置一个输出缓存, 以优先队列的方式进行组织,

其次, 重整形算法中, 对同一类优先级的链路将根据速率大小动态指定传输次序, 流量大的链路被认为紧急度更高, 如警报数据, 当传感器检测到异常情况, 将在一段时间内连续多次发出警报数据, 这些数据需要优先传输。系统中维护一个变量, 记录一个调整周期长度, 在调整周期内统计每条链路发送的数据帧的数目和最大帧长度。当调整周期结束时, 按照周期内统计的各条虚拟链路的数据帧数目和最大帧长度, 指定下一个调整周期内数据帧发送的先后次序。流量大的链路将在同一类优先级的链路中优先得到传输, 此处流量的单位是数据帧/秒, 这是因为紧急数据一般数据帧数量多, 但单个帧的长度短。

最后, 对于不规整的流量, 重整形算法采用抢占式调度, 当紧急数据帧到达输出端口, 且输出端口的缓存不为空时, 直接将该数据帧加入输出缓存的头部。

3.2.3 算法执行流程

系统初始化阶段：设置调整周期Period，对系统中的每条虚拟链路维护以下三个向量记录每条虚拟链路的历史信息：

- VL_NUM：记录一个调整周期内该链路发送的包数目；
- VL_Lmax：记录一个调整周期内该链路发送的包的最大长度；
- Send_order：设置下个调整周期内各条链路的发送次序。

算法流程图如图 3-2 所示：

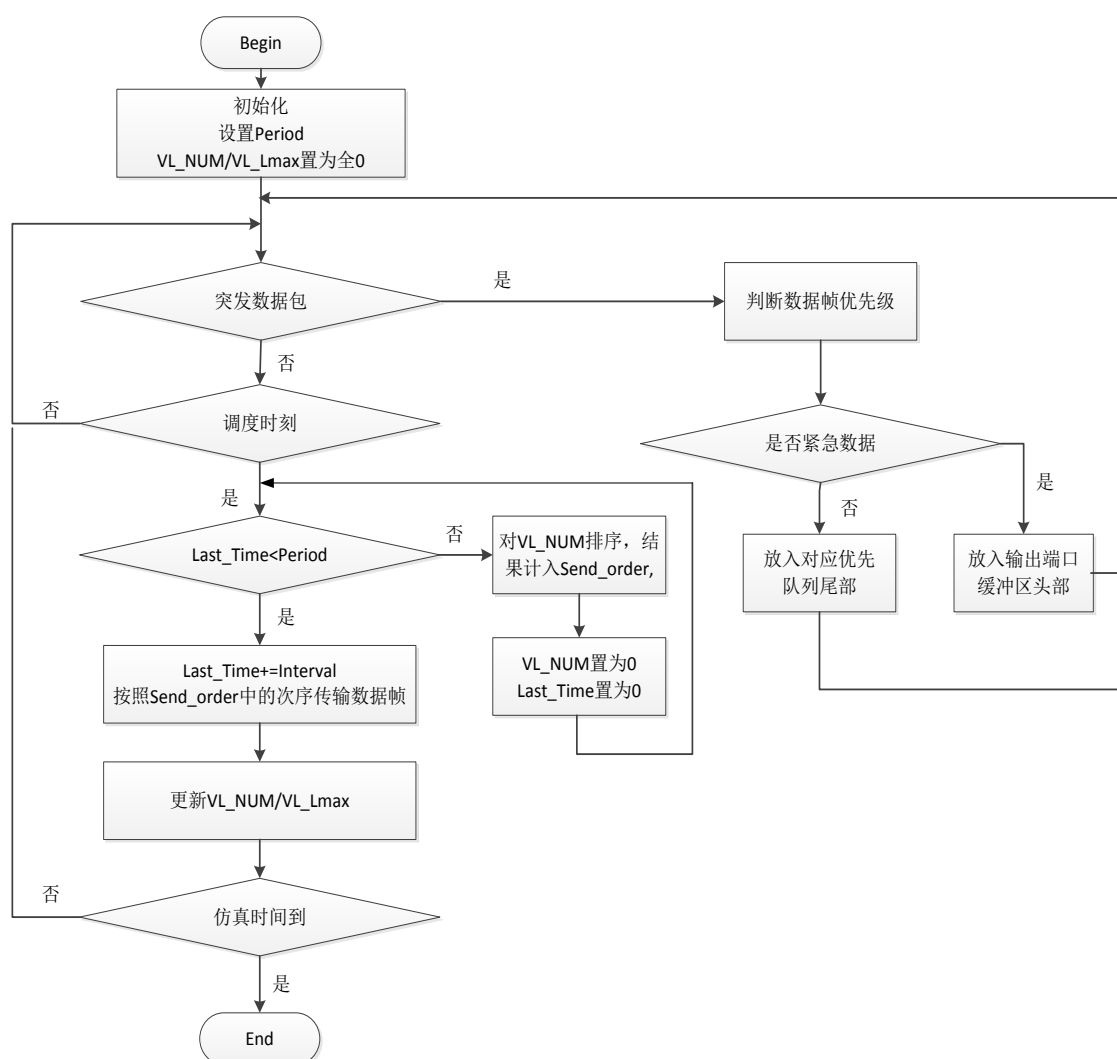


图 3-2 RSP 算法流程图
Fig.3-2 The flow chart of RSP method

系统运行时：当系统处于调整周期内时，每当一个数据帧到达输出端口，放入对应链路的输出缓存。在调度时刻，调度器按照 *Send_order* 内的次序将准备好的帧发送出去，每发送一个数据帧该链路对应的 *VL_NUM* 向量递增一次。

调度周期结束时刻，系统首先按照优先级对各条链路进行排序，接下来根据 *VL_NUM* 中记录的历史信息对同一优先级内的各条链路进行排序，将排序结果存入 *Send_order* 向量内。

RSP 算法的伪代码描述如下：

```

1.  Init:
2.       $t=0$ 
3.      for  $i=1$  to  $n$ 
4.          do  $N[i]=0$ ;  $L[i]=0$ ;  $s[i]=0$ 
5.  Run:
6.      If  $t < P$ 
7.           $t += Interval$ 
8.      else {
9.          for  $j=1$  to  $n$ 
10.             do if  $VL[i].priority > VL[i-1].priority$ 
11.                  $swap(s[i], s[i-1])$ 
12.             for  $k=1$  to  $m$ 
13.                 for  $a=1$  to  $v[k]$ 
14.                     do if  $N[a] > N[a-1]$ 
15.                          $swap(s[i], s[i-1])$ 
16.              $s[0]=0$ ;
17.             for  $k=1$  to  $n$ 
18.                  $s[k] = s[k-1] + L[k-1]/C$ 
19.             }
20.             If  $pk.arrival()$ 
21.                  $t1 = shaping()$ 
22.                  $t2 = t1 + s[ID(pk)]$ 
23.                  $send(pk, t2)$ 

```

其中, t 为持续时间, n 为虚拟链路个数, N 为 *VL_NUM* 向量, L 为 *VL_Lmax*

向量, s 为 Send_order 向量。

3.3 算法分析

航空电子系统中, 包含多种类型的数据, 这些数据包括具有不同的时延和时延抖动的要求, 如何能够有效的减小时延, 降低时延抖动是设计航空数据总线的首要目标, 特别是对于一些关键链路, 实时性要求较高, 且对抖动的容忍度低, 因此需要调度器对它们进行优先调度。**RSP** 方法相对于传统的调度方式, 能够有效地降低链路的时延抖动, 并保证关键链路的优先传输, 下面我们将从不同情况分析 **RSP** 调度方法的性能。

(1)、负载最轻:

当系统中只有一条链路进行数据传输时, 将是链路负载最轻的情况, 此时链路的时延和时延抖动性能如公式 3-1, 3-2 所示:

$$\text{send_delay}_{\text{traditional}} = \text{send_delay}_{\text{RSP}} = \text{TECHNICAL_DELAY} \quad (3-1)$$

$$\text{delay_jitter}_{\text{traditional}} = \text{delay_jitter}_{\text{RSP}} = 0 \quad (3-2)$$

可以看出, 当系统中只有一条链路进行数据传输时, 不存在链路之间的冲突情况, 调度器只需要对单一链路进行调度, 传统方式和 **RSP** 方式下的时延都等于系统的技术时延, 系统的时延抖动近似为 0。

(2)、负载最重:

当系统中所有链路都进行数据传输, 且传输数据帧的帧长是系统允许的最大长度时, 将是链路负载最重的情况。

传统 **FIFO/RM/EDF** 方式和 **RSP** 方式下的时延范围如公式 3-3 到 3-6 所示:

$$0 \leq \text{send_delay}_{\text{FIFO}} \leq \frac{\sum_{i=1}^N L \max_i}{C} \quad (3-3)$$

$$\frac{\sum_{j=1}^a L \min_j}{C} \leq \text{send_delay}_{\text{RM}_i} \leq \frac{\sum_{j=1}^b L \max_j}{C} \quad (3-4)$$

$$0 \leq \text{send_delay}_{\text{EDF}} \leq \frac{\sum_{i=1}^N L \max_i}{C} \quad (3-5)$$

$$\frac{\sum_{j=1}^c L \min_j}{C} \leq \text{send_delay}_{\text{RSP}_i} \leq \frac{\sum_{j=1}^d L \max_j}{C} \quad (3-6)$$

FIFO/RM/EDF/ RSP 方式下的时延抖动范围如公式 3-7 到 3-10 所示：

$$0 \leq \text{delay_jitter}_{\text{FIFO}} \leq \frac{\sum_{i=1}^N L \max_i}{C} \quad (3-7)$$

$$0 \leq \text{delay_jitter}_{\text{RM}_i} \leq \frac{\sum_{j=1}^b L \max_j - \sum_{j=1}^a L \min_j}{C} \quad (3-8)$$

$$0 \leq \text{delay_jitter}_{\text{EDF}} \leq \frac{\sum_{i=1}^N L \max_i}{C} \quad (3-9)$$

$$0 \leq \text{delay_jitter}_{\text{RSP}_i} \leq \frac{\sum_{j=1}^d L \max_j - \sum_{j=1}^c L \min_j}{C} \quad (3-10)$$

式中， N 为系统中虚拟链路的数目；

$L \max_i$ 为第 i 条链路的最大帧长度；

$L \min_i$ 为第 i 条链路的最小帧长度；

a 为调度周期小于当前链路调度周期的链路总数；

b 为调度周期小于当前链路调度周期的链路数和突发链路数的总和。

c 为优先级大于第 i 条链路优先级的链路总数；

d 为优先级大于等于第 i 条链路优先级的链路总数；

C 为传输链路的数据传输率。

从上面的式子中可以看出，在负载最重的情况下，非最低优先级的链路在 RM/RSP 调度方法下比在 FIFO/EDF 调度方法下的时延上界要小，这是由于 FIFO/EDF 方式考虑的是系统内的全部虚拟链路，而 RM/RSP 方法需要考虑的仅仅是优先级大于等于当前链路优先级的链路总数，因此高优先级的链路在传输时不会被其他低优先级的链路抢占。

同时，EDF/FIFO 方式下，每条虚拟链路的时延抖动范围大于 RM/RSP 下链路的时延抖动范围，EDF/FIFO 方式下，时延抖动范围的上界与时延上界相同，而 RSP 方式下，时延抖动范围为最大时延和最小时延的差值，最大时延和最小时延出现的场景描述如下：

最大时延：大于等于当前链路优先级的链路全部以最大帧长进行传输，即

$$\text{send_delay}_{j,\max} = \sum_{j=1}^b L \max_j。$$

最小时延：大于当前链路优先级的链路全部以最小帧长进行传输，即

$$send_delay_{j,\min} = \sum_{j=1}^a L \min_j。$$

(3)一般情况：

当系统中部分虚拟链路工作时，即属于一般情况，负载最重的情况是一般情况的特例，可由负载最重情况下的链路性能推导出一般情况下链路的时延和时延抖动性能。

FIFO、RM、EDF、RSP 方式下时延的变化范围如公式 3-11 到 3-14 所示：

$$0 \leq send_delay_{FIFO} \leq \frac{\sum_{i=1}^{N1} L \max_i}{C} \quad (3-11)$$

$$\frac{\sum_{j=1}^a L \min_j}{C} \leq send_delay_{RM_i} \leq \frac{\sum_{j=1}^b L \max_j}{C} \quad (3-12)$$

$$0 \leq send_delay_{EDF} \leq \frac{\sum_{i=1}^{N1} L \max_i}{C} \quad (3-13)$$

$$\frac{\sum_{j=1}^{c1} L \min_j}{C} \leq send_delay_{RSP_i} \leq \frac{\sum_{j=1}^{d1} L \max_j}{C} \quad (3-14)$$

FIFO 和 RSP 方式下时延抖动的变化范围如公式 3-15 到 3-18 所示：

$$0 \leq delay_jitter_{FIFO} \leq \frac{\sum_{i=1}^{N1} L \max_i}{C} \quad (3-15)$$

$$0 \leq delay_jitter_{RM_i} \leq \frac{\sum_{j=1}^b L \max_j - \sum_{j=1}^a L \min_j}{C} \quad (3-16)$$

$$0 \leq delay_jitter_{EDF} \leq \frac{\sum_{i=1}^{N1} L \max_i}{C} \quad (3-17)$$

$$0 \leq delay_jitter_{RSP_i} \leq \frac{\sum_{j=1}^d L \max_j - \sum_{j=1}^c L \min_j}{C} \quad (3-18)$$

式中， $N1$ 为系统中工作的虚拟链路的数目；

$L \max_i$ 为第 i 条链路的最大帧长度;

$L \min_i$ 为第 i 条链路的最小帧长度;

a 为调度周期小于当前链路调度周期的链路总数;

b 为调度周期小于当前链路调度周期的链路数和突发链路数的总和。

c1 为优先级大于第 i 条链路优先级的链路总数;

d1 为优先级大于等于第 i 条链路优先级的链路总数;

C 为传输链路的数据传输率。

可以看出,在一般情况下,对于 RM 调度方法,为了保证任务的周期性,每个数据包都以最大帧长安排发送时间,因此和最差情况下的时延和时延抖动相同。非最低优先级的链路在 RSP 调度方法下比在 EDF/FIFO 调度方法下的时延上界要小,同时每条链路的时延抖动范围在 RSP 调度方法下也比 EDF/FIFO 方式更小。

下面我们将以一个简单的示例网络来说明两种调度方法的差异:在示例网络中,包含三条虚拟链路(VL1~VL3),其中 VL1 和 VL2 具有高优先级,VL3 优先级低。

示例网络的拓扑结构如下图所示:

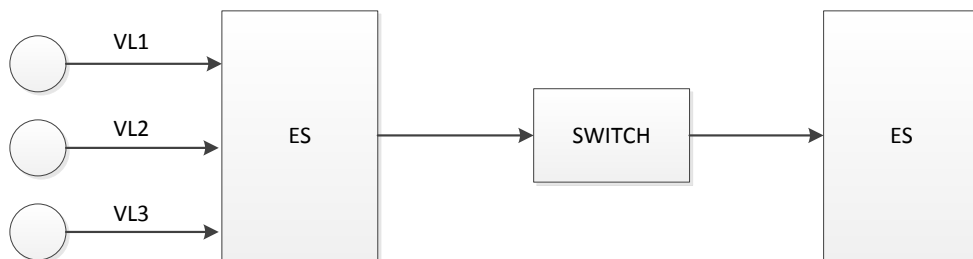


图 3-3 示例网络的拓扑结构
Fig.3-3 The Topology of Example Network

三条链路的历史信息如表 3-1 所示:

表 3-1 示例网络的历史信息
Table 3-1 Historical Information of Example Network

| VL | Historical Information | | | | |
|-----|------------------------|---------|--------|---------|------------|
| | Priority | BAG(ms) | VL_NUM | VL_Lmax | Send_order |
| VL1 | 2 | 4 | 100 | 1500 | 2 |
| VL2 | 2 | 2 | 200 | 1200 | 1 |
| VL3 | 1 | 2 | 300 | 1800 | 3 |

我们模拟几种数据帧可能出现的顺序,分析和比较这几种情况下 FIFO 调度方式和 RSP 调度方式的差异,并选择 VL2 作为要分析的目标链路。在示例网络中模拟了 4 种不同的帧到达序列,调度时刻间隔为 8ms,确保前后两次调度之间不存在冲突。在 FIFO 调度方法下,数据帧按照到达的时刻先后进行调度,由于到达序列的不同,从而调度的次序也不同。在 RSP 方法下,按照各条虚拟链路的历史信息进行数据帧的调度,由于 VL1 和 VL2 拥有更高的优先级,因此在调度时 VL1、VL2 将优先于 VL3 进行调度,同时,VL2 在上个调整周期内发送的包的数量(VL_NUM 值)比 VL1 大,因此当 VL2 和 VL1 产生冲突时,VL2 的数据帧将优先传输,因此可以得到三者发送的优先次序为 VL2>VL1>VL3。

不同调度方法下的调度示意图如图 3-4 所示:

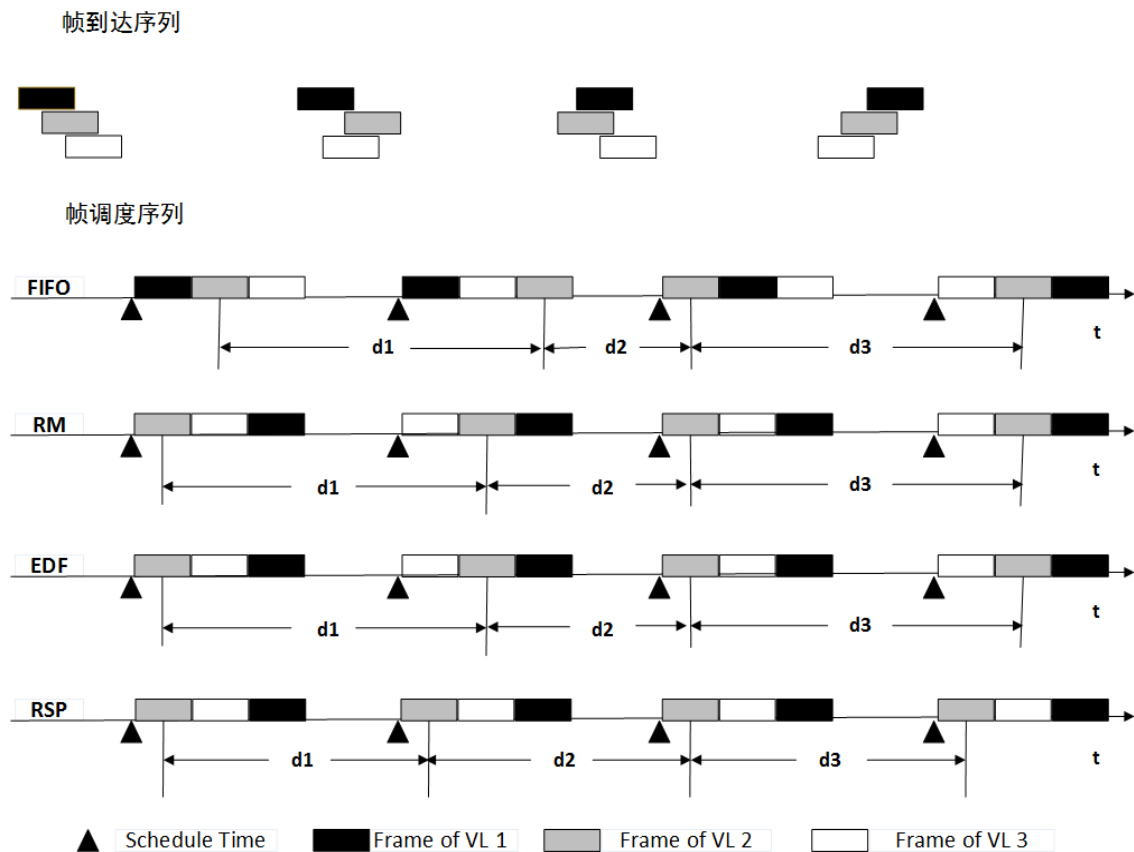


图 3-4 不同调度算法下的调度示意图

Fig.3-4 Schematic view of different scheduling method

其中, d1-d3 记录了 VL2 的 4 个数据帧被调度之间的时间间隔。

不同调度方法下 VL2 的调度间隔如表 3-2 所示:

表 3-2 不同调度方法下的调度间隔
Table 3-2 Transmission Interval of VL2

| Method | Transmission Interval | | |
|--------|-----------------------|-------|-------|
| | d1 | d2 | d3 |
| FIFO | 8.144 | 7.736 | 8.144 |
| RM | 8.144 | 7.856 | 8.144 |
| EDF | 8.144 | 7.856 | 8.144 |
| RSP | 8 | 8 | 8 |

从上面的示意图和表 3-2 中可以看出，FIFO 方式下，VL2 的数据帧被调度的时刻由链路的帧到达序列决定，由于网络传输的复杂性，帧到达序列不可预测，从而造成 VL2 的时延抖动大且不可预测。

RM 和 EDF 调度方式下，由于 VL2、VL3 的调度周期相同，因此在调度时刻可能出现 VL2 和 VL3 的帧的先后次序不固定的问题，转发次序将由帧到达序列决定，也可能产生一定的时延抖动。

RSP 方式下，根据优先级和历史信息的值，我们可以看出 VL2 的优先级高且在上个调整周期内发送的数据帧最多，VL1 的优先级高但发送的数据帧数目少于 VL2，VL3 的优先级低。因此，当调度时刻出现链路冲突时，VL2 将优先被调度，VL1 其次，VL3 最后。三条链路的先后次序相对固定，从而时延也相对稳定，时延抖动较小。

3.4 本章小结

本章主要介绍了 AFDX 网络中的传统调度算法 FIFO、RM、EDF 方法，针对传统调度方法存在的问题。提出了一种基于重整形思想的调度算法 RSP，介绍了 RSP 方法的原理、执行流程和步骤。并从理论角度分析了传统调度方法和 RSP 方法的时延以及时延抖动的差异。最后，通过一个简单的示例网络验证了 RSP 调度方法的有效性。

第四章 基于 OPNET 平台的 AFDX 网络建模

本章主要通过 OPNET 平台完成 AFDX 网络的建模，主要包括 OPNET Modeler 的介绍和 AFDX 网络模型的建立，包括网络层模型、节点层模型、进程层模型和相关的配置模型。

4.1 OPNET 网络仿真平台

4.1.1 网络仿真技术

在网络通信迅速发展的今天，人们对于网络的需求越来越多，对于网络性能的要求也在不断提高，网络研究人员不仅要不断开发新的网络协议、设计新的通信算法；同时也要不断研究如何有效利用现有资源，使网络的性能最高。在这两方面的研究过程中我们都需要对网络方案进行分析和验证。进行网络技术分析和验证一般有以下几种方法：

(1) 分析方法：分析方法的主要过程是通过设置限定条件和合理假设来建立网络的数学模型。随后通过对该数学模型的迭代求解，来完成对实际网络的分析和验证。

(2) 实验方法：实验方法是指搭建实际的网络来进行网络性能的分析 and 验证。因此在实验方法中，不仅需要软件资源，同时也需要硬件资源。

(3) 仿真方法：使用网络仿真平台建立所要研究的网络系统的模型，设定仿真条件，在计算机上运行仿真，并分析运行的输出结果。

在前期的网络研究中，由于技术和资源的限制，我们只能够使用分析方法对网络进行分析。分析方法受限于限定条件与假设，同时由于建立数学模型的难度较高，因此无法运用于复杂网络的设计和验证中。

分析方法能够对网络进行纯理论的分析，但是无法考虑实际网络运行中各种各样其他因素的影响，往往得到的分析结果与实际出入很大。而实验方法填补了分析方法的不足，实验方法通过使用实际的设备搭建实际的网络来对网络性能进行分析，实验方法得到的结果更接近于真实情况。

但是由于实验方法每更换一次实验方法都需要对网络进行更改甚至重新搭建，因此需要耗费大量的时间和软硬件资源，实施的周期长，成本也高，逐渐无法满足现代网络研究的需要。而随着软件技术的迅速发展，网络研究人员能够通过软件平台搭建需要研究的网络，网络研究也正式进入了仿真方法时代，仿真通过软

件对网络中的设备、协议、进程等进行建模，模拟实际网络的行为。能够用相对较少的费用和时间了解网络在不同条件下的各项特性，获取进行网络研究所需的丰富有效的数据。

仿真方法主要依赖于网络仿真平台，通过使用网络仿真平台提供的模型库、函数库来搭建所要研究的网络。仿真方法无需使用实际的设备，因此能够大大缩短网络的建设周期，节省人力物力成本。同时，仿真方法能够在一段时间内对网络进行多次仿真，采集不同情况下网络的性能数据，因此可以在实际网络的建设中，提供科学、客观、可靠的评估。能够缩短网络建设的周期，规避不必要的风险。

目前，使用较广泛的网络仿真软件包括 OPNET 网络仿真平台和 NS2 仿真模拟平台。这两个网络仿真平台都具有强大的功能，下面我们将从不同方面对它们进行分析和比较：

在软件功能方面，OPNET 平台提供的功能较为完备，用户不仅可以对注入模型，分布模型进行设置，也可以使用不同厂家提供的网络通信设备和通信链路来搭建自己的网络。OPNET 平台覆盖了绝大多数现在市场上的主流网络设备和通信协议，用户无需担心找不到所需的设备。同时，OPNET 提供了功能强大且完备的模型库和数据采集与分析工具，用户可以方便地选择库中已有的网络拓扑结构和通信设备来搭建网络，也可以直接使用平台内置的数据分析工具对采集的数据进行分析处理，生成相应的报告或报表。而在 NS2 中没有提供这样的模型库，用户只能通过脚本来建立实际网络的逻辑结构，同时 NS2 中对结果的采集和分析也需要借助其他软件的辅助。

在操作易用性方面，OPNET 平台提供了简单方便的图形化操作界面，可以使用非常少的操作就能够得到比较详细和真实的仿真结果，从仿真网络搭建、模块编译、数据注入到运行仿真、数据采集、结果分析、报表生成，OPNET 平台都提供了非常便捷的操作。对于一些常规网络，用户甚至无需编写代码，只需要选择相应设备和链路，指定需要采集的参数就可以完成一次网络仿真。而 NS2 则需要通过编写脚本和代码来搭建仿真网络，用这种方式建立复杂的网络会变的非常困难。

可以看出，OPNET 网络仿真平台提供的功能强大完备，操作简便易用，因此在我们的建模和仿真过程中，将选择 OPNET 平台作为我们的网络仿真平台，在其上建立 AFDX 网络的模型并搭建仿真网络，运行仿真，最后采集数据进行结果分析。

4.1.2 Modeler 概述

OPNET 网络仿真平台最早是由麻省理工学院(MIT)信息决策实验室受美国军方的委托开发出来的。作为目前应用最广泛、功能最完备的网络仿真平台, OPNET 为许多网络通信领域的专业人士提供了基于软件层面的预测解决方案。

1987 年以来, OPNET 网络仿真平台迅速稳步发展, 目前已经在通信、国防以及其他网络领域被广泛认可和采用。OPNET 网络仿真平台能够提供综合化的网络规划与设计信息, 成千上万的学校、组织、研究机构使用 OPNET 仿真平台来优化网络性能、提高通信网络和应用数据的可用性。

OPNET Modeler 作为 OPNET 软件的主体是当前使用最广泛的网络技术仿真开发平台。它为开发人员提供了网络建模、仿真以及性能分析的集成环境, 极大程度减轻了研究人员编码以及分析的工作量。无论是研究新的通信设备、通信协议, 还是优化现有网络结构、网络应用, Modeler 都提供了简单灵活、功能强大的设计工具, 帮助用户以最小的代价完成网络规划设计。

OPNET Modeler 所能应用的领域包括端到端网络结构设计 (End to End Network Architecture Design)、系统级网络仿真 (System Level Simulation for Network Devices)、新协议开发和优化 (Protocol Development and Optimization)、网络和业务层配合以达到最好的性能 (Network Application Optimization and Deployment Analysis)。例如, 在 TCP/IP 协议栈的开发中, 从 IPv4 网络升级为 IPv6, 采用什么样的技术效果比较好; 新协议的开发和优化中, 如目前流行的 4G 无线通信协议。在系统级网络仿真中, 设计新的调度算法或路由算法, 分析如何使系统达到要求的 QoS; 在网络和业务优化方面, 可以分析新引进的网络或者业务对整个网络的影响, 以及网络对业务的需求等, 实际中业务和网络是对矛盾体, 通过 Modeler 模拟可以寻找业务和网络之间能达到的最好指标。

OPNET Modeler 为网络仿真提供全面综合的集成开发环境和性能评测方法, Modeler 仿真主要包括以下步骤: 模型建立、运行仿真与数据采集、结果分析。

模型建立: 利用 OPNET Modeler 提供的模型库和模型编辑器建立实际系统的模型。OPNET Modeler 采用层次化的建模方法, 包括网络层 (Network) 建模、节点 (Node) 层建模和进程 (process) 层建模。进程层模型模拟节点内各个模块的状态和行为, 并通过有限状态机模拟进程状态的转变; 节点层模型将各个进程模型连接起来形成网络设备; 网络层模型将这些网络设备连接起来形成所要研究的网络。不同的网络层模型可以模拟不同的网络场景, 不同的网络场景之间可以相互比较, 从而确定出什么样的设计才是最优的。

运行仿真与数据采集：设置仿真的运行条件和参数，指定需要采集的数据。然后运行仿真，采集仿真结果。

结果分析：仿真结束后，采集的数据组织并保存在数据文件中，可以使用 Modeler 本身提供的数据分析功能或者使用第三方数据处理软件（如 Matlab 等）进行数据分析和处理。

这几个步骤按顺序不断迭代，从而逐步优化网络的性能，迭代流程图如图 4-1 所示：

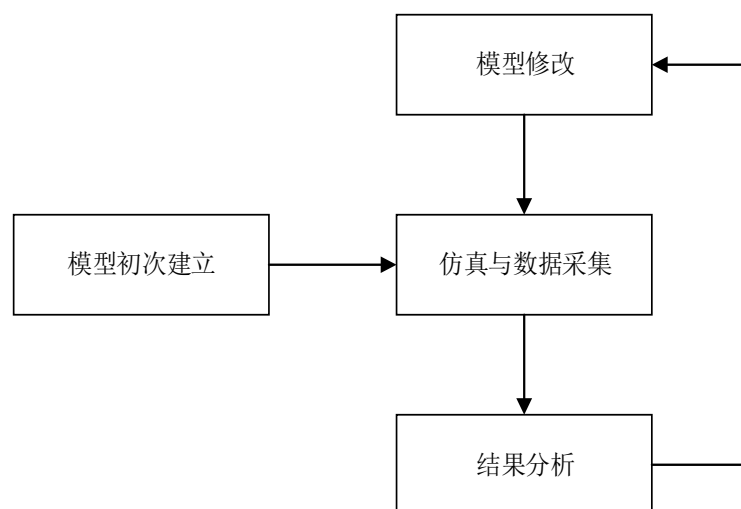


图 4-1 OPNET Modeler 仿真步骤
Fig.4-1 OPNET Modeler Simulation Procedure

4.1.3 Modeler 仿真原理

OPNET Modeler 采用离散事件驱动（Discrete Event Driven）的仿真原理，其中，“事件”是指网络状态的变化，当网络中有“事件”发生时，Modeler 推进仿真向前进行。网络状态描述了当前网络中节点和进程的信息。网络状态的变化使用进程模型中的有限状态机进行描述，只有当网络状态发生变化时，仿真器才工作。因此，与时间驱动的仿真模型相比，离散事件驱动的计算量少，仿真运行的时间快，且不会遗漏网络的状态信息。在 OPNET Modeler 中，维护一个离散事件驱动的时间调度器（Event Scheduler）和事件列表，事件列表中记录了所有事件发生和结束的时间。时间调度器按照事件发生的时间顺序对事件列表中的事件排序，然后依次取出事件列表中的事件进行执行，当事件执行完毕后，将事件从事件列表中清除。当新事件加入事件列表时，按照新事件的开始事件将其插入到事件列

表中正确的位置上。离散事件仿真流程如图 4-2 所示：

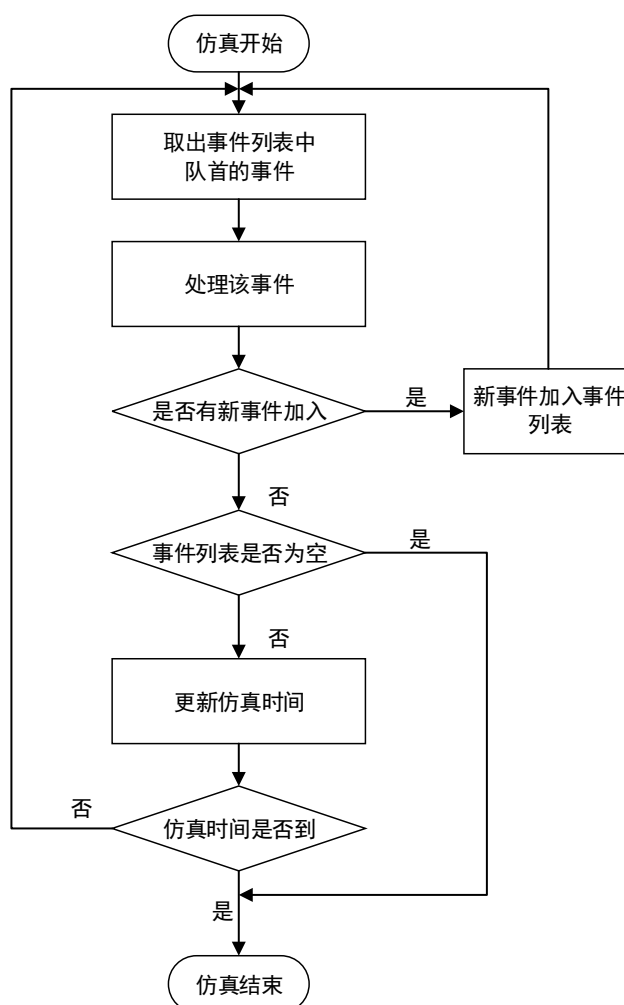


图 4-2 离散时间仿真流程图
Fig.4-2 DES Simulation Flow Chart

在仿真进行时，各个节点之间和节点中各个进程之间以传递数据包的方式进行通信，并以中断方式传递事件的相关信息。离散事件驱动的仿真机制，可以保证通信的并发性和顺序性。通过进程间的状态机转换描述网络状态的变化，每出现一个事件中断时就会触发一个描述网络行为或系统行为的进程模型的运行，因此仿真核心可以仿真通信网络中任何时间、任何情况下的网络状态^[33]，同时也可以避免对无状态变化时间内的仿真，因此相对于传统的实验方法能够更快的得到网络的性能参数。

在离散时间驱动中，事件执行必成后更新的时间是仿真时间(Simulation time)，

仿真时间不同于逝去时间(Elapsed time)。仿真时间描述了当前仿真执行的进度，是网络中的抽象时间，它随着仿真事件的执行而逐渐推进。而逝去时间则描述了仿真程序运行的时间，是真实时间。在进程层模型中，仿真时间的推进可以通过执行中断调度事件列表中的事件来完成，在 OPNET 中提供了多种类型的中断，如自中断、周期中断等。每一个中断都对应一个事件，因此可以利用中断完成事件的调度，例如执行自中断调用语句 `op_intrpt_schedule_self(op_sim_time()+ T, 中断序号)`后，下一个事件将在当前仿真时间推进 T 秒后执行。

4.1.4 Modeler 建模方法

OPNET Modeler 采用层次化和模块化的建模方式。Modeler 的三层模型分别为：网络层、节点层和进程层。层次化设计能够将复杂的网络分解为不同的层次结构，每一层中使用模块化的建模方式，每个模块完成小的任务，每层包含多个这样的小模块，组合在一起完成该层的功能。

网络层建模主要完成对网络设备和通信链路的设置和连接，作为 Modeler 三层模型的最高层，网络层描述了整个网络的行为，包括多少个通信节点，多少条通信链路等。当网络层模型很复杂时，也可细分为各个子网，各个子网连接起来形成整个网络。

节点层建模主要完成网络设备的功能，每个节点层模型都由多个小的节点模块构成，每个节点模块实现该节点功能的一个方面，例如：数据采样、数据包封装、差错校验等，多个节点模块组合在一起构成功能完整的网络设备。在节点层建模中，各个节点模块之间通过包流(packet_streams)或数据线(statistic wires)来连接，包流传递的是当前节点要处理的数据，可以对其中的数据包进行封装或拆分，统计线则完成两个子模块之间特定数据的传输，该数据和要处理的数据无关，只是完成节点模块间状态信息的交换。在 OPNET Modeler 中，节点模块一般包括数据流、处理器和收发机，数据流完成节点内部数据包的接收、存储和转发；处理器完成对数据包的处理，包括封装、校验等；而收发机则完成从物理链路接收数据包和发送数据包到物理链路的功能。

进程层建模主要描述节点模型内各个节点模块的功能和行为，进程层模型不仅可以模拟通信算法、网络协议的执行，同时还可以实现流量控制、差错校验等功能。作为 Modeler 三层模型的最底层，进程层模型描述了网络通信的细节，每个进程模型都由不同的进程状态和状态间的转换构成。进程状态描述了当前网络所处的实际状态，进程在同一时刻只能处于其中一个状态中，这些状态的组合构

成了进程状态集，而所有状态间的转换则构成了网络的行为集合。进程状态分为两类，forced 状态和非 unforced 状态，forced 状态描述了进程的一个暂态，当触发该状态的条件满足后，进程进入 forced 状态，执行完 forced 状态的动作后，当前进程立刻离开 forced 状态，进入 unforced 状态。unforced 状态描述了进程的常态，当进程进入 unforced 状态后，执行完进入代码后需要等待新的触发条件满足后，才会离开 unforced 状态。在进程模型中，每个进程都至少有一个初始状态，初始状态是进程被访问的第一个状态，可以是 forced 状态，也可以是 unforced 状态。

4.1.5 OPNET 通信机制

OPNET Modeler 中提供以下几种方式完成不同层次、不同对象之间的消息传递，主要包括：基于包的通信、中断方式、基于接口控制信息（ICI）进行通信和基于统计线的通信：

1、基于包的通信：数据包是网络中实际传输的数据经过封装、校验等处理后形成的数据载体。OPNET Modeler 采用基于包的通信方式来模拟实际网络中端到端节点之间数据的流动。包也是 OPNET Modeler 为支持基于信息源的通信而定义的一种数据结构。在 OPNET Modeler 中可以编辑、生成任何标准或自定义的包格式。节点也可以对包进行动态创建、修改、封装、拆分。数据包的组成主要包括头部结构和载荷信息。在头部结构中，每个数据域都有固定的格式，并以名字字段或索引号进行标识，不同的数据域存储不同的信息，包括包的长度、源端口/目的端口、校验和等。在载荷信息中，存储实际需要传输的数据。头部结构一般为定长，为载荷信息则一般为变长。

2、中断方式：OPNET Modeler 提供多种类型的中断，包括周期性中断和自中断等，仿真核心通过引发中断来控制仿真的执行。在 OPNET Modeler 中，常见的中断包括 begsim_intrpt、endsim_intrpt、failure_intrpt、regular_intrpt、stat_intrpt。这些中断的触发条件、触发场景各不相同，每个中断都由一个中断号来标识，在仿真过程中，中断是不可缺少的重要工具。

3、基于接口控制信息（ICI）进行通信：在 OPNET Modeler 中，ICI 可以用来完成不同节点模块之间的信息的传递，在 ICI 中存储了与事件相关的相关数据和控制信息。在仿真过程中，每个 ICI 都与一个特定的事件进行绑定，每触发一次该事件，都可以更新一次存储在 ICI 中的相关信息。不同节点模块之间通过 ICI 来完成交互，ICI 的数据结构更简单，因此使用 ICI 方式进行通信比使用数据包方便，适用范围也更广。在仿真过程中，通过使用 op_ici_create()函数来完成 ICI 的

创建,使用 `op_ici_attr_set()`函数来完成 ICI 中相关信息的设置,使用 `op_ici_install()`函数来完成 ICI 与事件的绑定。

4、基于统计线的通信:OPNET Modeler 中提供了一个简单的接口,该接口使用节点域中的统计线来完成模块间少量特定类型的信息传输,被传输的信息被称为统计量。在网络通信中经常会出现节点模块之间仅需传输少量单一类型的信息的情况,对于这种情况而言,利用数据包来进行通信显得过于复杂,此时使用统计线来进行通信可以有效的节省资源。同时,统计线还可以用来动态获取其他节点模块的状态,可以完成对其他节点模块状态的监视。此外,统计线还可以作为信号量,用于实现不同模块进程的底层信令机制。

4.2 网络层建模

AFDX 网络是全双工交换式以太网,不同于传统以太网,AFDX 网络使用全双工链路进行数据传输,网络拓扑为星型。网络中包含的网络设备包括航空电子子系统、端系统、交换机和传输链路。

一个示例网络的网络层拓扑如图 4-3 所示:

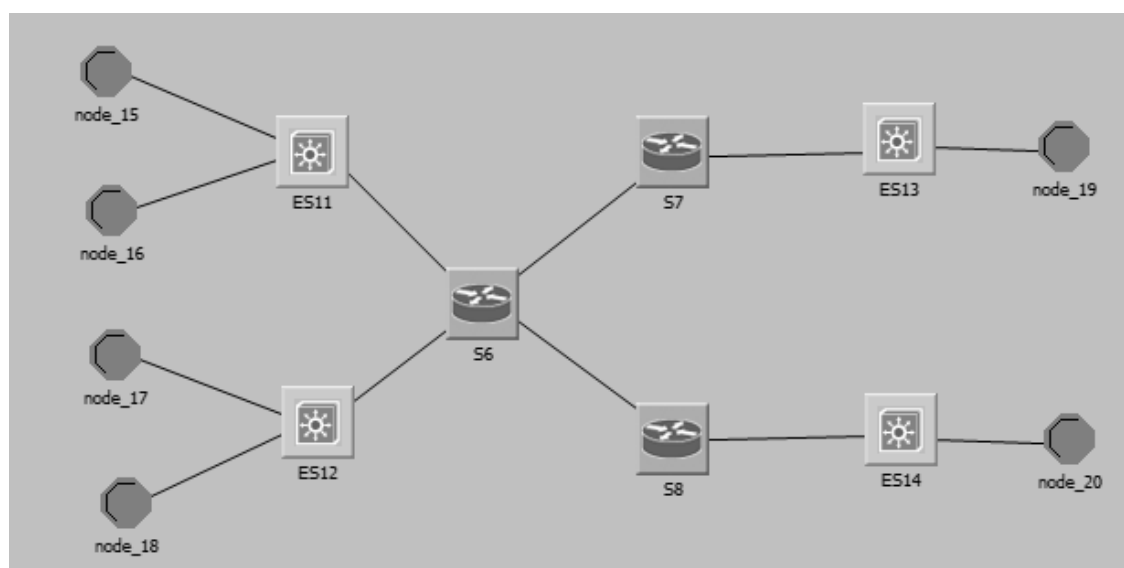


图 4-3 网络层拓扑

Fig.4-3 Network Level Topology

示例网络中,包含 6 个航空电子子系统,4 个端系统,3 个交换机和 12 条传输链路。这些网络设备可通过不同连接方式组织在一起,构成 AFDX 网络。星型的拓扑结构为网络提供了很好的扩展性和灵活性。

4.3 节点层模型

在 OPNET 仿真中，每个网络设备都是一个节点，节点完成一个网络设备完成的所有工作，节点由多个节点模块构成，每个节点模块完成节点功能的一部分。AFDX 网络中包含航空电子子系统、端系统、交换机和传输链路四种网络设备，本文将分别对他们的 OPNET 模型进行说明。

4.3.1 航空电子子系统模型

在 AFDX 网络中，航空电子子系统产生应用数据，并将数据传送给端系统进行传输。由于在网络仿真中，我们并不关心产生的数据类型，并假设每个航空电子子系统只产生一种类型的数据，因此可以建立一个简单的数据发生器和数据接收器作为航空电子子系统的模型，并以节点编号区分各个子系统。模型如图 4-4 所示：

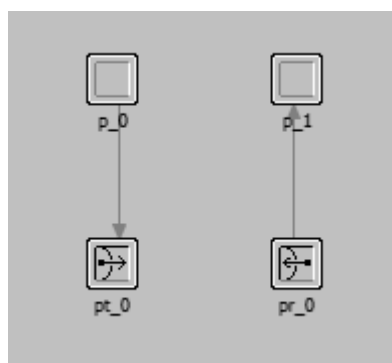


图 4-4 航空电子子系统模型
Fig.4-4 Avionics Sub-system Model

其中，p_0 模拟数据发生器，使用 OPNETModeler 提供的标准发包模型 simple_source，p_1 模拟数据接收器，使用标准的销毁模型 sink。pt_0 是发送器，pr_0 是接收器。

4.3.2 端系统模型

端系统是 AFDX 网络中上层应用与通信网络之间的接口，每个端系统使用一个 16 位的二进制编码来表示。端系统主要完成的工作包括数据接收、协议栈处理、流量整形与调度、数据帧发送。

AFDX 网络基于 IEEE 802.3 标准，采用 TCP/IP 协议栈进行数据封装，传输层

使用 UDP 协议，网络层使用 IP 协议，端系统模型如图 4-5 所示：

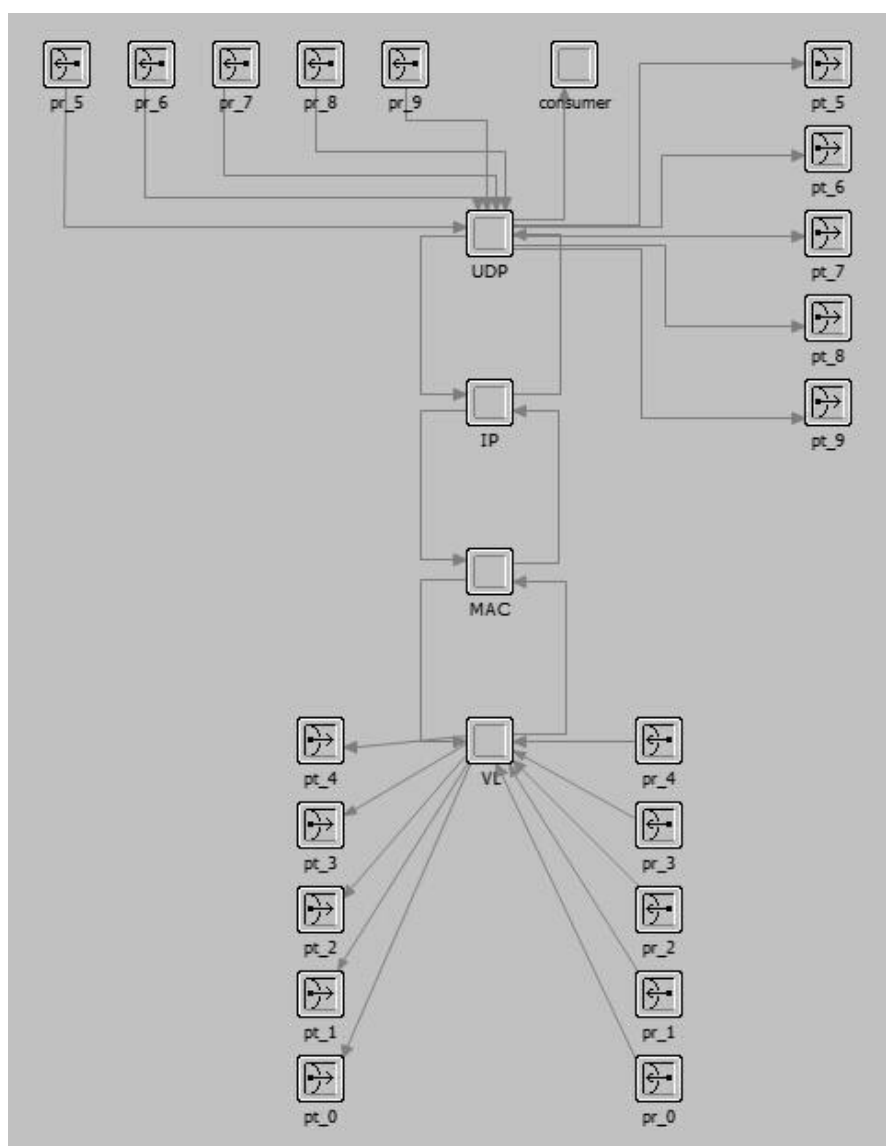


图 4-5 端系统模型
Fig.4-5 End-system Model

在端系统模型中，UDP 模块模拟传输层行为；指定链路的 UDP 端口号，包括源端口号和目的端口号，为数据包添加 UDP 包头。

IP 模块模拟网络层行为；指定链路的源 IP 地址和目的 IP 地址，若数据包过大，则对数据包进行分组处理，并添加 IP 包头。

MAC 模块模拟逻辑链路层行为；将上网络层提供的 IP 数据包封装成符合 AFDX 帧格式的数据帧。

VL 模块模拟介质访问层的行为；模拟虚拟链路的行为，对虚拟链路进行流量整形和调度处理，还包括完整性检测和冗余管理。

consumer 模块模拟消费者行为；使用 OPNET 提供的标准销毁模型 sink。

pt0~9 模拟发送机行为；

pr0~9 模拟接收机行为。

4.3.3 交换机模型

交换机是构成 AFDX 通信网络的核心，所有交换机形成交换机网络，完成存储转发功能。AFDX 网络中交换机模型如图 4-6 所示：

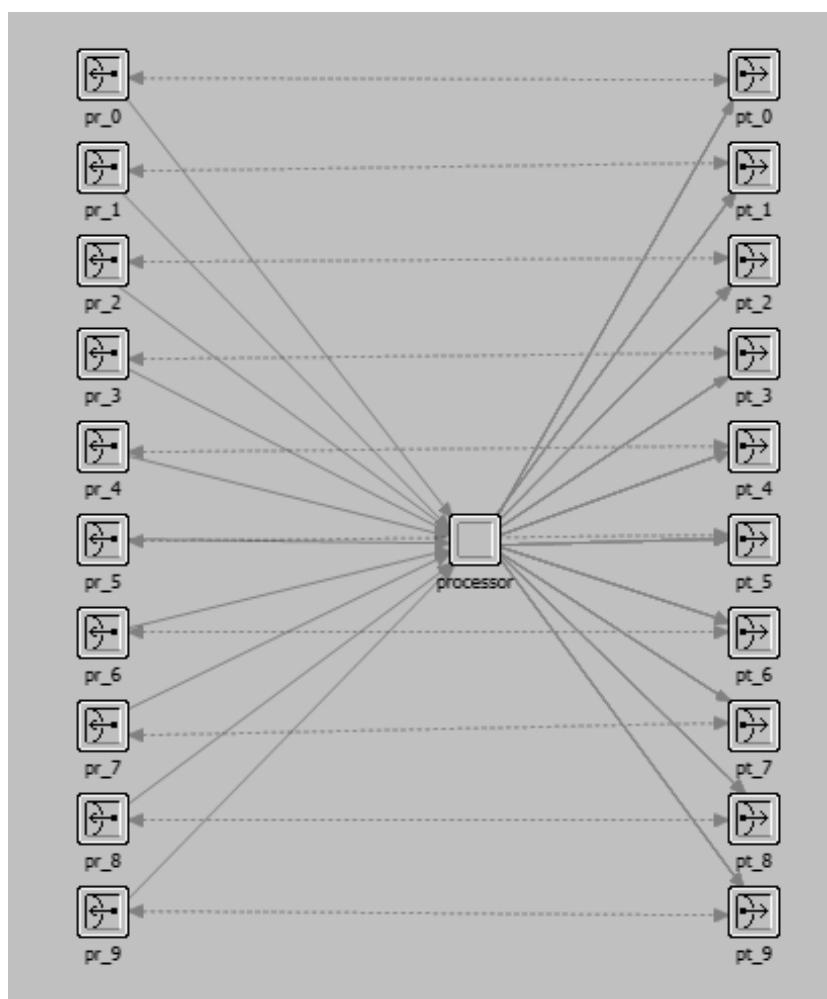


图 4-6 交换机模型

Fig.4-6 Switch Model

在交换机模型中，processor 完成交换机的主要功能，包括完整性校验、数据包过滤、流量管制和交换转发等。

pt0~9 模拟发送机行为;
pr0~9 模拟接收机行为。

4.4 进程层建模

OPNET 建模中进程描述了网络所处的状态,刻画节点模型里处理机和队列模型的行为。进程可以处于两种状态: forced 和 unforced 状态。forced 状态描述了进程的一个暂态,当触发该状态的条件满足后,进程进入 forced 状态,执行完 forced 状态的动作后,当前进程立刻离开 forced 状态,进入 unforced 状态。unforced 状态描述了进程的常态,当进程进入 unforced 状态后,执行完进入代码后需要等待新的触发条件满足后,才会离开 unforced 状态。在进程模型中,每个进程都至少有一个初始状态,初始状态是进程被访问的第一个状态,可以是 forced 状态,也可以是 unforced 状态。

4.4.1 协议栈的进程层建模

AFDX 网络中,采用 Ethernet 802.3/IP/UDP 协议栈来完成对数据包的封装、指定源端地址和目的端地址、进行数据分组、添加对应包头、进行流量控制和差错控制。AFDX 网络的包头格式在传统 TCP/IP 包头格式的基础上进行了小幅度改动,使其符合 AFDX 网络的要求。

AFDX 网络中端到端通信的标识方式为:源 UDP 端口号+源 IP 地址+目的 MAC 地址(虚拟链路号)+目的 IP 地址+目的 UDP 端口号。每个数据包都由一个这样的五元组唯一指定了一条通信路径。

例如,消息元组(SU,SI,DM,DI,DU)标识了这样一个消息,该消息的源 UDP 端口为 SU,源 IP 地址为 SI,目的 MAC 地址(虚拟链路号)为 DM,目的 IP 地址为 DI,目的 UDP 端口号为 DU。

在发送中,一个 AFDX 通信端口仅与单独的一组地址相连接:源 UDP 端口,源 IP 端口,VL(目的 MAC 地址),目的 IP 地址和目的 UDP 地址。

在接收时,一个 AFDX 通信端口仅与单独的一组地址相连接:唯一的源 UDP 端口,目的 IP 地址,VL(目的 MAC 地址),并且,如果冗余管理关闭,还应当与以太网的物理接口相连接。

在 OPNET 建模中,对 TCP/IP 协议栈进行建模,端系统内 UDP/IP/MAC 层的进程模型如图 4-7 所示:

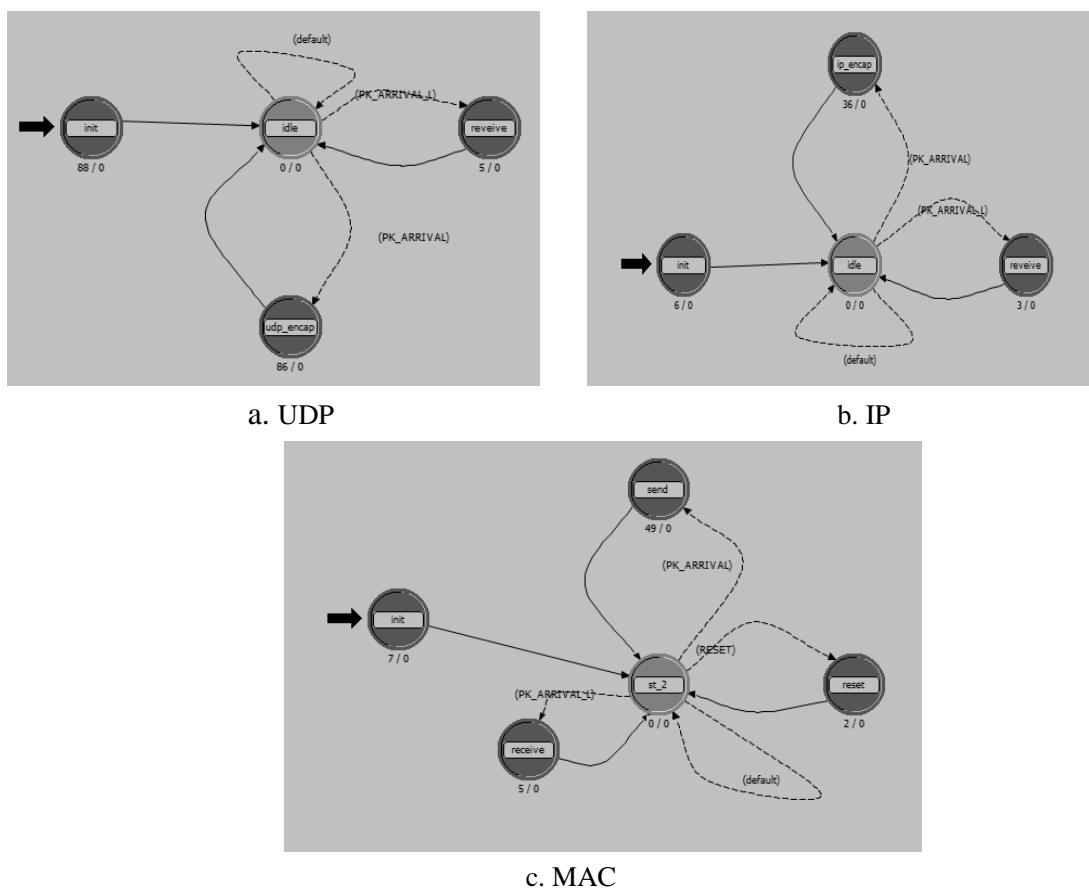


图 4-7 UDP/IP/MAC 层进程模型
Fig.4-7 UDP/IP/MAC Process Model

4.4.2 虚拟链路的进程层建模

在 AFDX 网络通信中，虚拟链路指定了一条预先分配好的链路，使用静态路由的方式进行数据帧的转发。链路配置通过以下几个表完成：端系统发送表，虚拟链路配置表，接口配置表。

各个表完成的功能如下：

- 端系统发送表，对端系统内所有连接进行标识，指明每条连接的源分区标号，源 UDP 端口号，源 IP 地址，源 MAC 地址目的 UDP 端口号，目的 IP 地址，目的 MAC 地址和虚拟链路编号。
- 虚拟链路配置表，记录端系统内所有虚拟链路的配置参数，包括最大帧间距（BAG），最大帧长 Lmax，虚链路端口，最小帧长 Lmin，链路优先级和最大允许抖动 Max_jitter。
- 接口配置表，记录每条虚拟链路的端口，包括两个冗余网络 A、B 的输入

端口和输出端口。

这些配置表之间的关系如图 4-8 所示：

端系统发送表

| | | | | | | | | |
|------|-----|--------|-------|--------|---------|--------|---------|-------|
| 连接序号 | 源分区 | 源UDP端口 | 源IP地址 | 源MAC地址 | 目的UDP端口 | 目的IP地址 | 目的MAC地址 | VL_ID |
|------|-----|--------|-------|--------|---------|--------|---------|-------|

虚拟链路配置表

| | | | | | | |
|-------|-----|----------|------|----------|-----|------------------|
| VL_ID | BAG | 最大帧长Lmax | 端口标识 | 最小帧长Lmin | 优先级 | 最大允许抖动Max_jitter |
|-------|-----|----------|------|----------|-----|------------------|

接口配置表

| | | | | |
|------|---------|---------|---------|---------|
| 端口标识 | 网络A输出端口 | 网络B输出端口 | 网络A输入端口 | 网络B输入端口 |
|------|---------|---------|---------|---------|

图 4-8 虚拟链路配置
Fig.4-8 Configuration of Virtual Link

虚拟链路的进程层模型如图 4-9 所示：

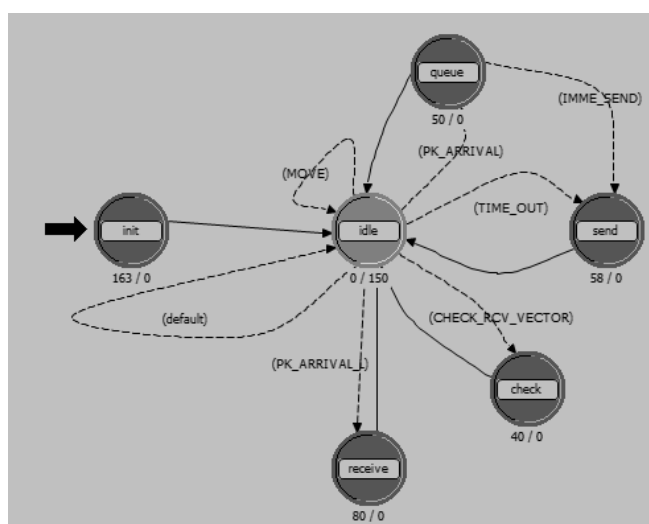


图 4-9 虚拟链路进程模型
Fig.4-9 Process Model of Virtual Link

其中，Init 模块完成相关初始化工作；queue 模块完成流量整形，send 模块完成发送功能，receive 模块完成接收功能，check 模块完成冗余管理。

4.4.3 交换机的进程层建模

在交换机中，需要实现数据帧的过滤、流量管制和数据帧的转发，其中，交换转发表是交换机模型中数据帧转发的依据，转发端口表记录多播地址的转发端口，交换转发表和转发端口表的格式如图 4-10 所示：

交换转发表

| | | | | | |
|-------|-------|--------|------|------|-------|
| VL_ID | 源IP地址 | 目的IP地址 | 传输类型 | 转发端口 | 转发端口表 |
|-------|-------|--------|------|------|-------|

转发端口表

| | | | | | |
|-------|-----|-----|-----|-----|-----|
| 转发端口表 | 端口数 | 端口1 | 端口2 | ... | 端口n |
|-------|-----|-----|-----|-----|-----|

图 4-10 交换机配置表
Fig.4-15 Configuration of Switch

交换机的进程层模型如图 4-11 所示：

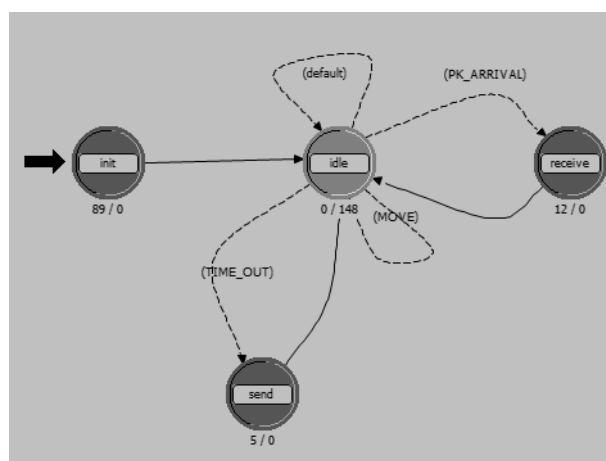


图 4-11 交换机进程模型
Fig.4-11 Process Model of Switch

4.5 本章小结

本章主要介绍了在 OPNET 平台对 AFDX 网络进行建模的方法，首先介绍了 OPNET 网络仿真平台：介绍了 OPNET Modeler 仿真平台和 OPNET 的离散事件仿

真机制、层次建模机制以及通信机制。其次详细说明了 AFDX 网络在 OPNET 平台上的建模过程。包括网络层建模、节点层建模和进程层建模。

网络层建模：描述 AFDX 网络设备间的连接方式。

节点层建模：对航空电子子系统、端系统和交换机进行建模。

进程层建模：建立 AFDX 网络的协议栈进程建模，虚拟链路进程模型和交换机进程模型以及他们对应的配置信息。

第五章 仿真与结果分析

本章主要完成网络仿真和结果分析，包括仿真网络的配置，仿真参数的设置，数据的采集和结果的分析。

5.1 仿真网络

为了对 AFDX 网络进行仿真，我们建立了如下的仿真网络：网络中包含 14 个航空电子子系统，每个子系统产生一种类型的数据，并对应一条虚拟链路，编号为 V1-V14。包括 5 个端系统，编号为 E1-E5。包括 4 个交换机，编号为 S1-S4。仿真网络的拓扑结构如图 5-1 所示：

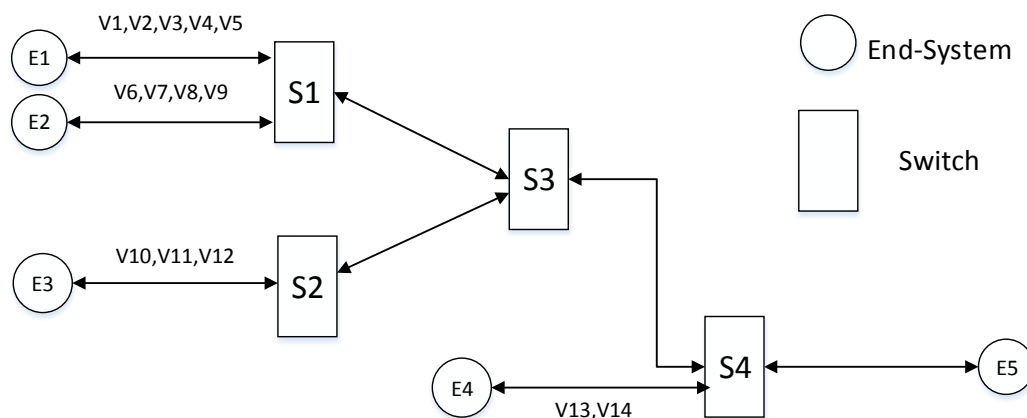


图 5-1 仿真网络拓扑结构
Fig.5-1 Simulation Network Architecture

我们选取 VL1-VL5 作为考察链路，因为 VL1-VL5 属于同一个端系统，且都通过交换机 S1 进行数据帧转发，因此可以仿真当链路发生冲突时，调度算法的性能优劣。

对于航空电子系统而言，网络的实时性和稳定性是保障飞机正常飞行和乘客安全的重要因素，也是评价机载航空总线性能的主要指标。我们选择的性能评价指标主要有以下几个：

1、数据包平均时延：描述了数据包从源端发送到目的端接收的时间间隔。平均时延反映了网络的实时性，时延越小，链路的实时性越高。在实际的网络中，关键任务的实时性需要得到保障。

2、时延抖动：时延抖动描述了时延的抖动情况。时延抖动越小，网络的稳定性越好，在实际的网络中，对连续性要求高的网络要求时延抖动小。

5.2 仿真参数设置

在仿真网络中，我们对网络中的 14 条链路进行配置。仿真的时间设置为 1 分钟，其中数据发生器在第 10 秒开始产生数据。设置三级优先级，各级优先级仿真的数据特点如下：

- 优先级为 3 的链路模拟紧急数据，如警报信息等，具有突发性，在一段时间内发送多个包，要求链路时延低。
- 优先级为 2 的链路模拟传感器数据，监测数据等，这些数据一般具有周期性，连续性，要求网络的时延小，但实时性要求低于优先级为 3 的链路。
- 优先级为 1 的链路模拟用户多媒体数据，文件传输等，这些数据的数据量大，链路数量多，对时延的要求低，希望时延抖动尽量小。

虚拟链路配置参数如表 5-1 所示：

表 5-1 虚拟链路参数配置表
Table5-1 Configuration of Virtual Link

| VL_ID | BAG (ms) | Lmax (Byte) | Begin_t (s) | End_t (s) | Priority | port |
|-------|-------------|----------------|----------------|--------------|----------|------|
| 1 | 8 | 1500 | 10 | 60 | 1 | 0 |
| 2 | 16 | 1500 | 10 | 30 | 2 | 0 |
| 3 | 32 | 1500 | 10 | 60 | 1 | 0 |
| 4 | 16 | 1500 | 20 | 30 | 3 | 0 |
| 5 | 64 | 1500 | 10 | 60 | 1 | 0 |
| 6 | 32 | 1500 | 10 | 60 | 1 | 0 |
| 7 | 16 | 1500 | 40 | 50 | 3 | 0 |
| 8 | 8 | 1500 | 10 | 60 | 1 | 0 |
| 9 | 16 | 1500 | 10 | 60 | 1 | 0 |
| 10 | 16 | 1500 | 10 | 60 | 2 | 0 |
| 11 | 32 | 1500 | 10 | 60 | 1 | 0 |
| 12 | 16 | 1500 | 10 | 60 | 2 | 0 |
| 13 | 16 | 1500 | 10 | 10 | 3 | 0 |
| 14 | 32 | 1500 | 10 | 10 | 1 | 0 |

在虚拟链路设置中，VL1、VL3、VL5 具有相同优先级，可以仿真相同优先级链路间出现冲突的情况，VL2、VL4 具有高优先级，可以仿真不同级优先级链路间出现冲突的情况。

在仿真中，设置仿真的调整周期为 0.1s。

其他 OPNET 平台仿真参数设置如图 5-2 所示：

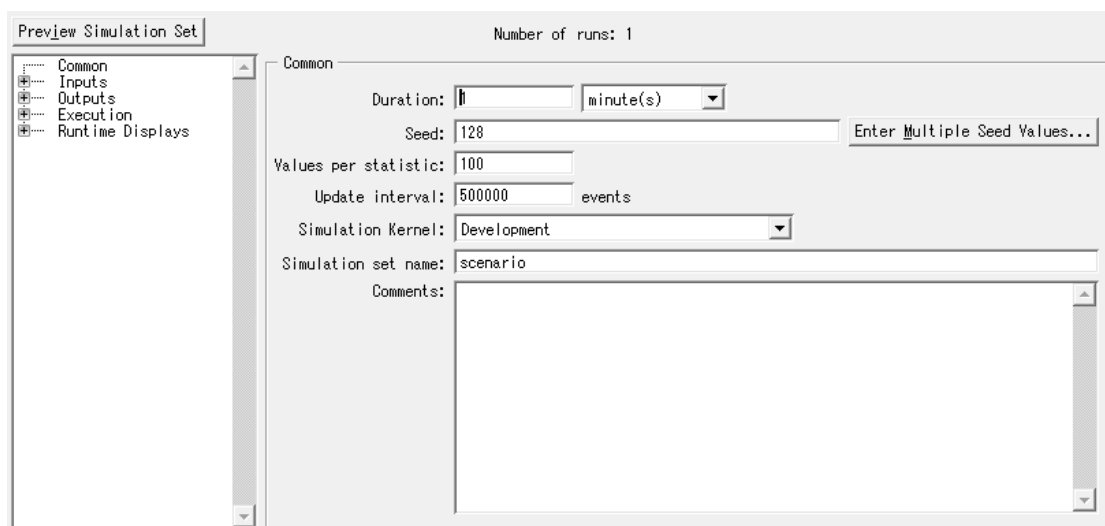


图 5-2 OPNET 参数设置
Fig.5-2 Parameter Setting on OPNET Platform

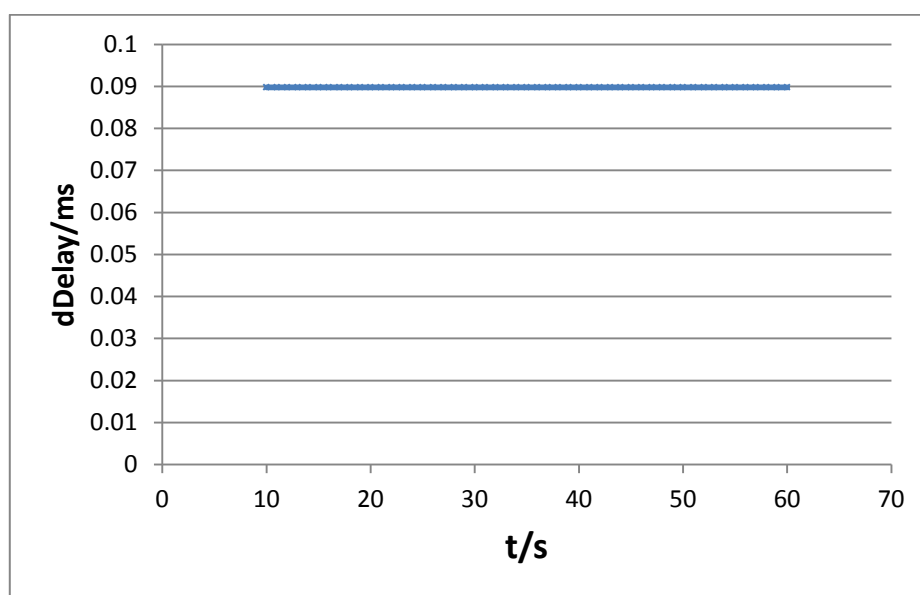
5.3 网络仿真与数据采集

为了能够尽量反映出实际 AFDX 网络的情况并验证 RSP 调度算法的性能，我们设置如下的仿真场景并逐一对他们进行仿真：

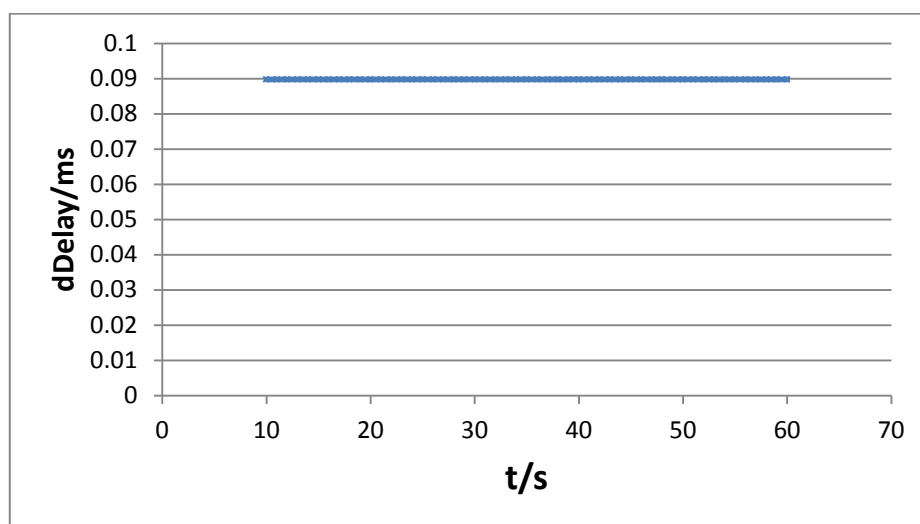
- 单链路工作：网络中只有单一链路工作，此情况为理想情况。
- 多链路相同优先级：网络中存在多条链路，但是各条链路的优先级相同，因此可以仿真存在调度冲突时，选择不同的调度算法对网络性能的影响。
- 多链路不同优先级：网络中存在多条链路，且各条链路的优先级不尽相同，可以仿真不同调度算法对关键链路性能的影响和对整体链路性能的影响。

5.3.1 单一链路工作

首先，我们对单一链路的网络进行仿真，假设网络中只有一条链路 VL1 在工作，则 VL1 的数据帧平均时延如图 5-2 所示：



a. FIFO/RM/EDF



b. RSP

图 5-3 单链路工作时数据帧平均时延

Fig.5-2 Average Delay of Frame when Single Link Work

可以看出，在只有单一链路工作的情况下，在传统方法下和 RSP 方法下的数据帧的平均时延大约为 0.08976ms，时延抖动几乎为 0，此时延即为网络的技术时延。

因此，在理想情况下，即 AFDX 网络中无冲突的时候，调度器选择可以发送的链路进行数据包发送，此时不同调度算法的选择对网络时延无影响，网络的时延只由技术时延决定，且时延抖动非常小。

5.3.2 多链路同等优先级工作

在单链路工作情况下，使用传统方式进行调度和使用 RSP 方法进行调度其结果没有明显差异。接下来我们仿真当系统中工作的链路均具有同等优先级时，各条链路的时延情况。

在上面的仿真网络中，VL1、VL3、VL5 具有同等优先级，因此我们选择 VL1、VL3、VL5 作为考察链路，分别使用 FIFO、RM、EDF、RSP 方式进行调度。链路的注入速率和最大帧间距 BAG 配置如下表所示：

表 5-2 虚拟链路配置表
Table5-2 Configuration of Virtual Link

| VL 编号 | VL1 | VL3 | VL5 |
|--------|-----------|--------|--------|
| 注入速率 | 102.4kB/s | 32kB/s | 16kB/s |
| 链路 BAG | 8ms | 32ms | 64ms |

在 FIFO 调度方式下，设置仿真时间为 120s，各条链路的平均时延如图 5-4 所示：

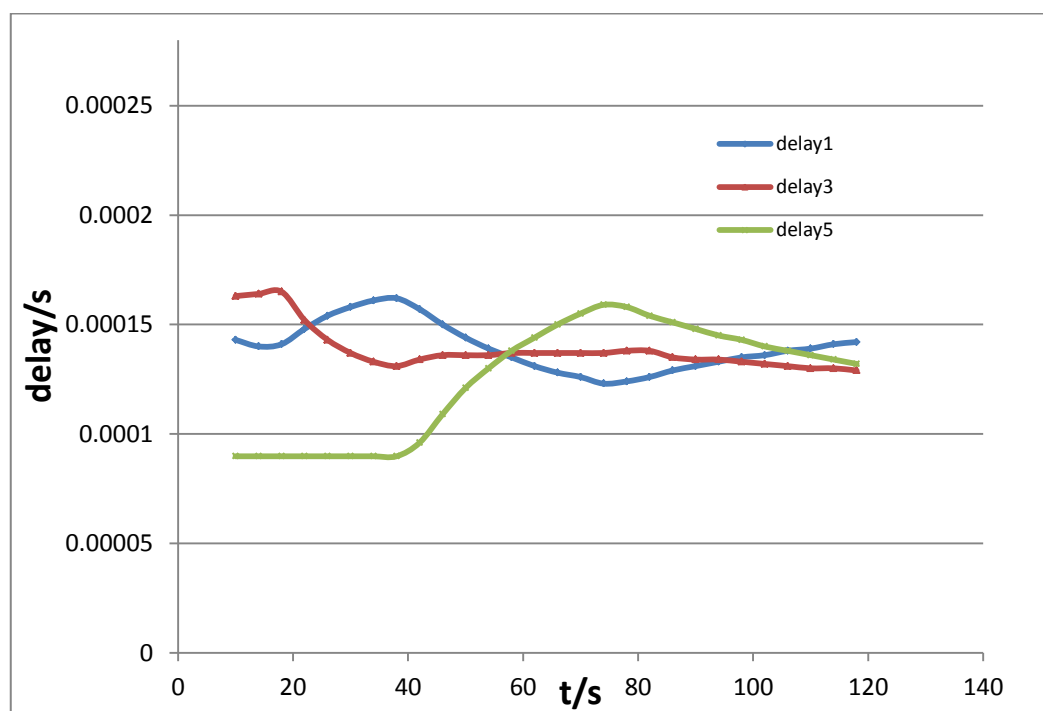


图 5-4 FIFO 方法下时延变化情况
Fig.5-4 Average Delay Using FIFO Method

在 RM、EDF、RSP 方法下，各条链路的平均时延如图 5-5 所示：

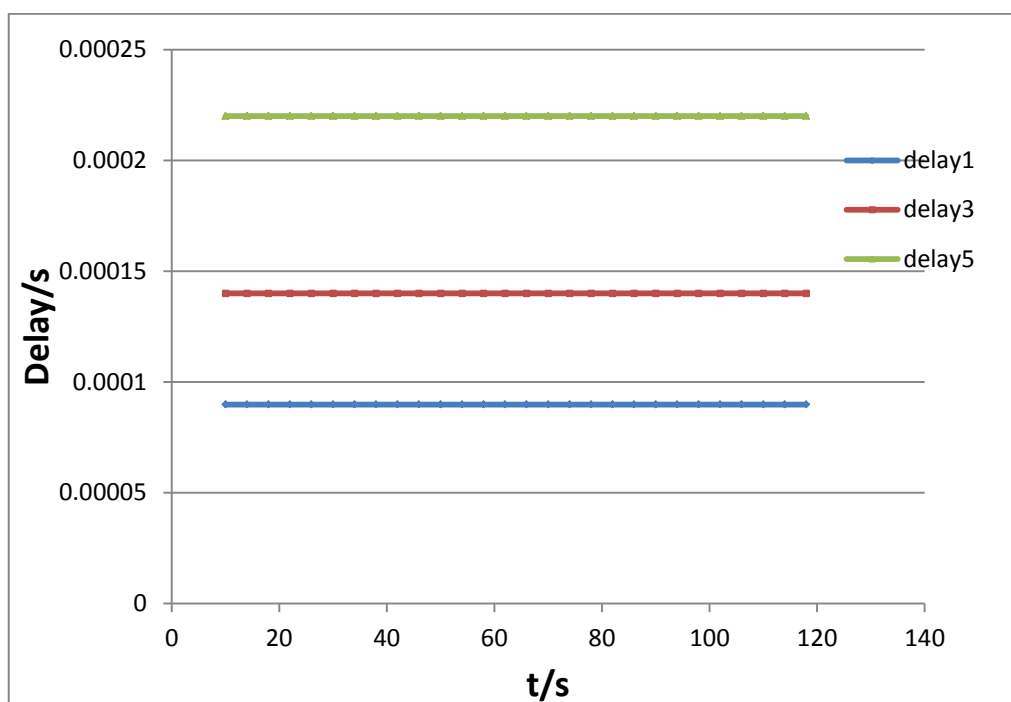


图 5-5 RM/EDF/RSP 方法下时延变化情况
Fig.5-5 Average Delay Using RM/EDF/RSP Method

各条链路的平均时延与抖动情况如表 5-3 所示：

表 5-3 平均时延量化表（单位:ms）
Table5-3 Quantitative Average Delay Table

| VL | FIFO | | | RM/EDF/RSP | | |
|-----|-------|-------|-------|------------|------|----|
| | 最大时延 | 最小时延 | 抖动 | 最大时延 | 最小时延 | 抖动 |
| VL1 | 0.162 | 0.123 | 0.039 | 0.09 | 0.09 | 0 |
| VL3 | 0.165 | 0.132 | 0.033 | 0.14 | 0.14 | 0 |
| VL5 | 0.159 | 0.09 | 0.069 | 0.22 | 0.22 | 0 |

从图 5-3 中，可以看出，各条链路的时延变化没有规律，随着仿真时间的推进，VL1、VL3、VL5 的平均时延趋于相同，可以证明 FIFO 方式是一种公平的调度方式。但是 FIFO 方式下各条链路的时延变化大，即各条链路的时延抖动较大。

从图 5-4 中可以看出，在 RM 方式下，系统根据链路的周期大小静态安排各条链路的优先级，由于 VL1 的转发周期即 BAG 值最小，因此 VL1 的优先级最高。VL3 次之，VL5 最低。在 EDF 方式下，链路的截止时间定义为 BAG-已等待时间，

由于 VL1 的 BAG 最小，因此 VL1 拥有最小的截止时间，在转发时，拥有最高的优先级。在 RSP 方式下，按照各条链路在上个调整周期内发送的数据帧的数量决定链路的发送次序，由于 VL1 具有最大的注入速率，因此在调度时，VL1 具有相对靠前的发送次序，VL3 次之，VL5 的发送次序最靠后。

5.3.3 多链路不同优先级工作

在实际 AFDX 网络中，一般都是多条链路同时工作，这里我们仿真所有链路同时工作的情形。并设置各条链路的注入速率如表 5-2 所示：

表 5-4 虚拟链路配置表
Table5-2 Configuration of Virtual Link

| VL 编号 | VL1 | VL2 | VL3 | VL4 | VL5 |
|--------|-----------|-----------|--------|-----------|--------|
| 注入速率 | 102.4kB/s | 51.2 kB/s | 32KB/s | 51.2 kB/s | 16kB/s |
| 链路 BAG | 8ms | 16ms | 32ms | 16ms | 64ms |

在 FIFO 方式下，VL1-VL5 的时延情况如图 5-6 所示：

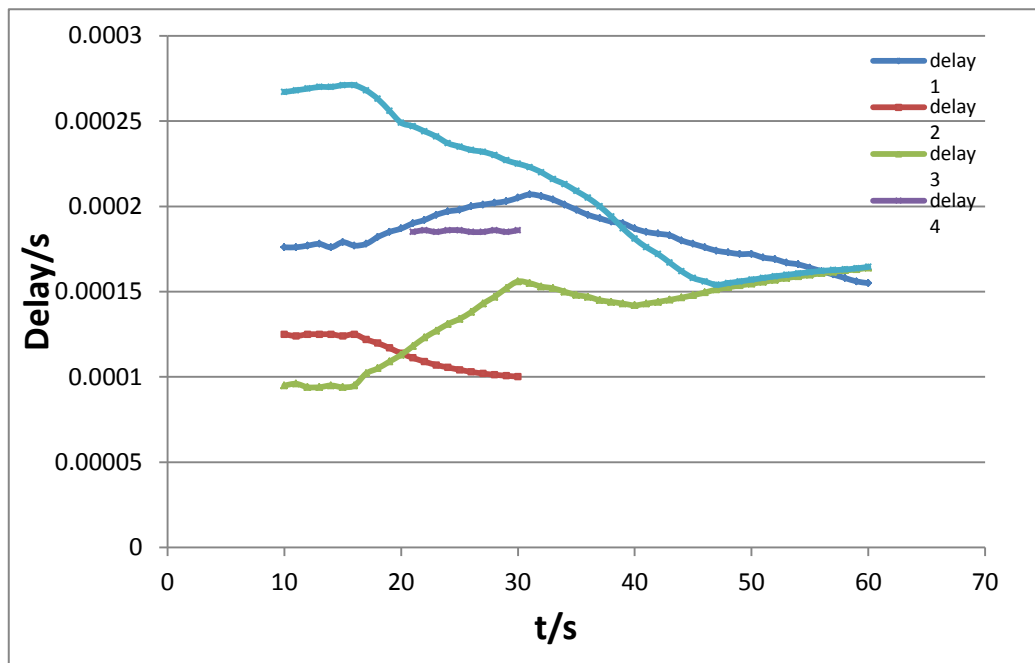


图 5-6 FIFO 方法下时延变化情况
Fig.5-6 Average Delay Using FIFO Method

从图中可以看出，在 FIFO 方式下，关键链路 VL4 的时延仍然是比较大的，

在 VL4 进行传输时，会被 VL2 和 VL3 抢占。VL4 的平均时延大概为 0.186ms，高于上面计算的技术时延约 0.1ms。链路 VL2 在调度初期会被 VL3 抢占，而 VL2 的优先级大于 VL3，破坏了 VL2 的实时性。同时，VL1-VL3 链路的时延变化范围大。链路的时延抖动较严重。

在 RM 方式下，VL1-VL5 的时延情况如图 5-7 所示：

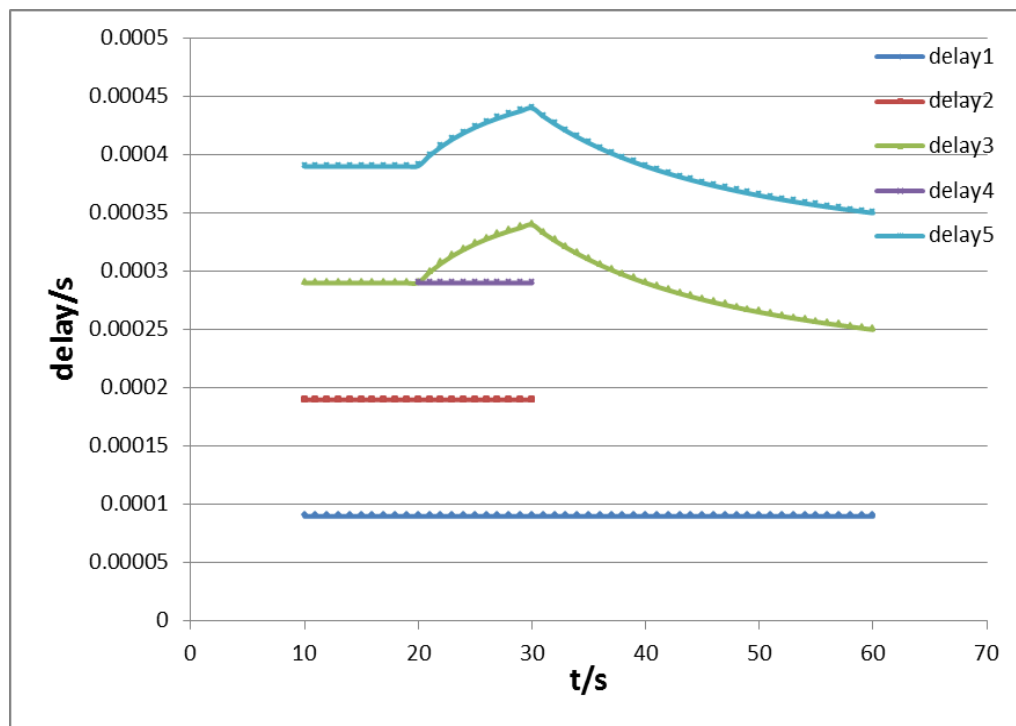


图 5-7 RM 方法下时延变化情况
Fig.5-7 Average Delay Using RM Method

从图中可以看出，在 RM 调度方法下，系统根据链路的周期值即 BAG 来静态决定链路的优先级，由于 VL1 的 BAG 最小，因此在调度时刻，VL1 的数据帧将拥有最大的优先级，VL2 与 VL4 的 BAG 值相同，但是 VL2 在起始时刻就有数据帧发送，因此 VL2 的优先级高于 VL4，VL3 的优先级低于 VL4 但高于 VL5。该优先级在随后的仿真过程中，不再改变。仿真进行到 20s 时，VL4 有数据帧发送的时候，将抢占 VL3 和 VL5 的发送，因此会造成 VL3、VL5 在 20s 之后的时延上升。仿真进行到 30s 时，VL2、VL4 停止发送数据，因此 VL3、VL5 的发送时间下降。

在 EDF 方法下，VL1-VL5 的时延情况如图 5-8 所示：

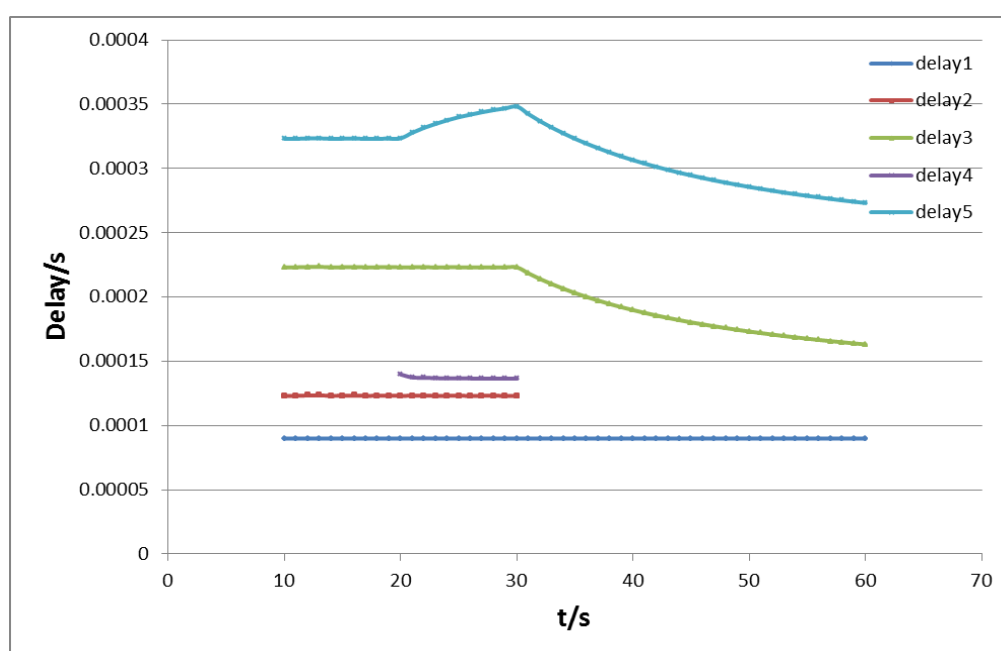


图 5-8 EDF 方法下时延变化情况
Fig.5-8 Average Delay Using EDF Method

在 RSP 方法下，VL1-VL5 的时延情况如图 5-9 所示：

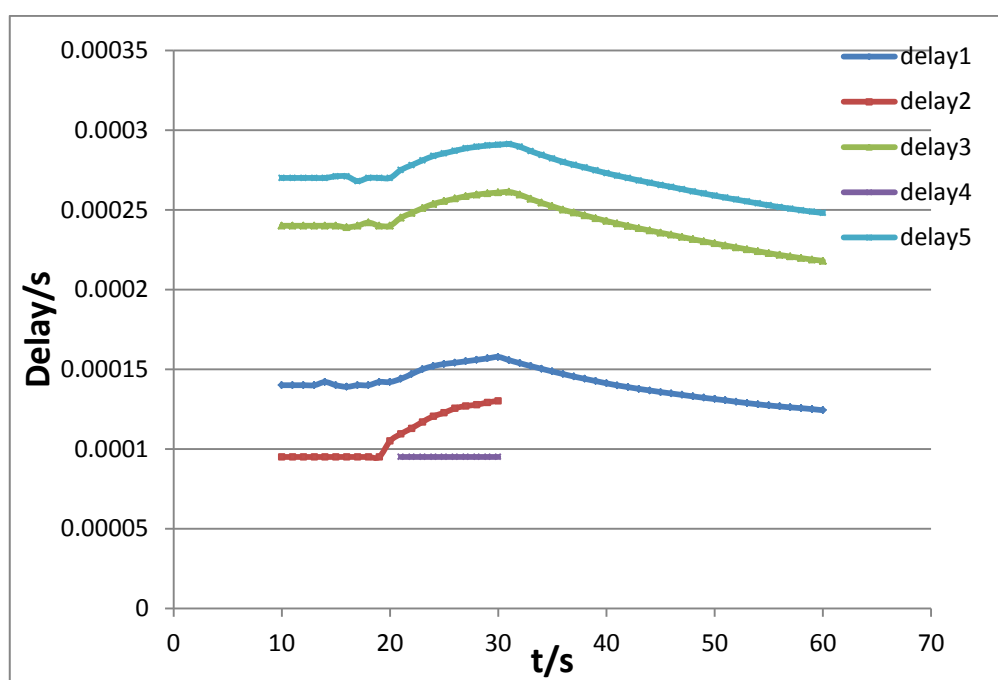


图 5-9 RSP 方法下时延变化情况
Fig.5-9 Average Delay Using RSP Method

在 EDF 方式下，链路的优先级由链路的截止时间动态决定，一般情况下，拥有较小的 BAG 的链路其截止时间较短，因此在调度时刻 VL1 拥有高优先级，而对于 VL2 和 VL4 而言，他们的 BAG 相同，因此两者的时延也几乎相同。对于 VL3 和 VL5 而言，由于 BAG 值较大，因此在调度时刻往往被其他链路抢占，因此时延较大。

在 RSP 方式下，当处于 20s-30s 之间时，VL4 有数据进行传输时，将具有最大的优先级，且时延抖动几乎为 0。当时间处于 10s-20s 时，VL2 有数据传输而 VL4 无数据传输时，此时 VL2 具有最大的优先级，因此 VL2 将优先于其他链路进行传输。在 30s-60s 时间段内，仅有 VL1、VL3、VL5 进行数据传输，由于 VL1、VL3、VL5 的链路优先级相同，因此各条链路之间的传输顺序将由链路的流量动态决定，VL1 的注入速率最高，VL3 次之，VL5 最低，因此当出现链路冲突时，优先传输 VL1 的帧，其次传输 VL3 的帧，最后传输 VL5 的帧。

对于航空电子系统而言，最大时延往往意味着系统可能出现的最差情况，如果最大时延能够满足系统的要求，则可以认为该算法符合系统的要求，图 5-10 展示了不同调度算法下不同链路的最大时延情况：

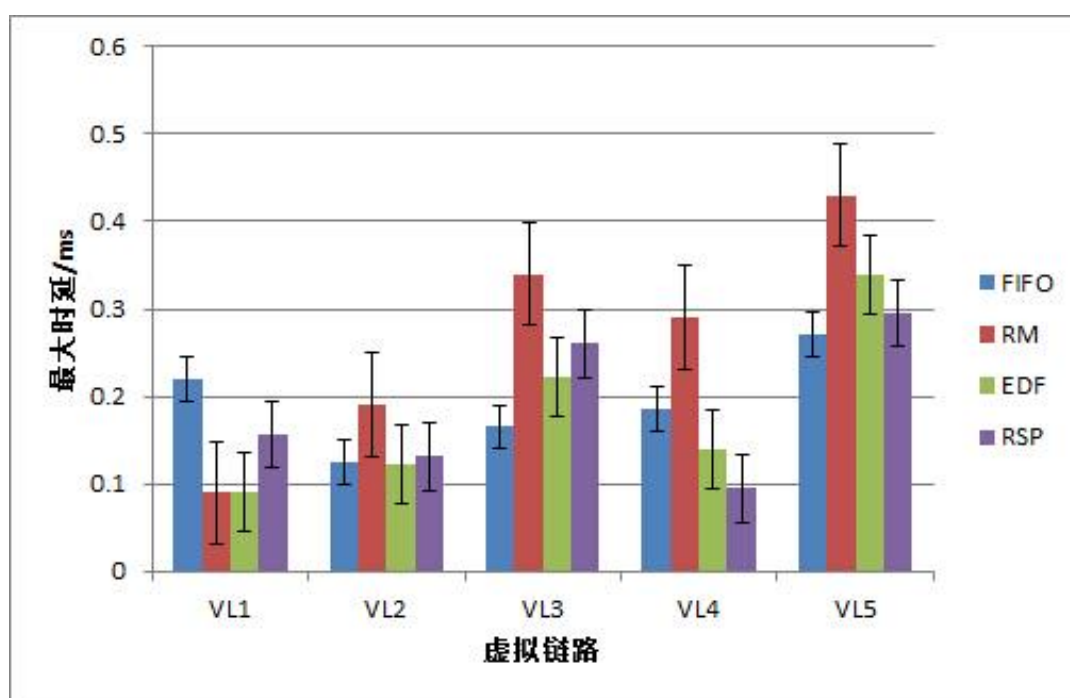


图 5-10 不同调度方法下最大时延
Fig.5-10 Maximum Delay Using Different Method

从图 5-9 中可以看出，对于紧急数据 VL4 而言，RSP 方法相对于其他方法有

效的降低了链路的最大时延，从而保证了关键链路的高优先传输。对于链路 VL2 而言，RSP 方法和 RM 以及 EDF 方法下的最大时延几乎相同，RSP 方法也保证了 VL2 的性能。

对于 VL1、VL3、VL5 而言，用户更关心的性能主要是时延的抖动，对时延的要求小，图 5-11 描述了不同调度方法下各条链路的时延抖动情况：

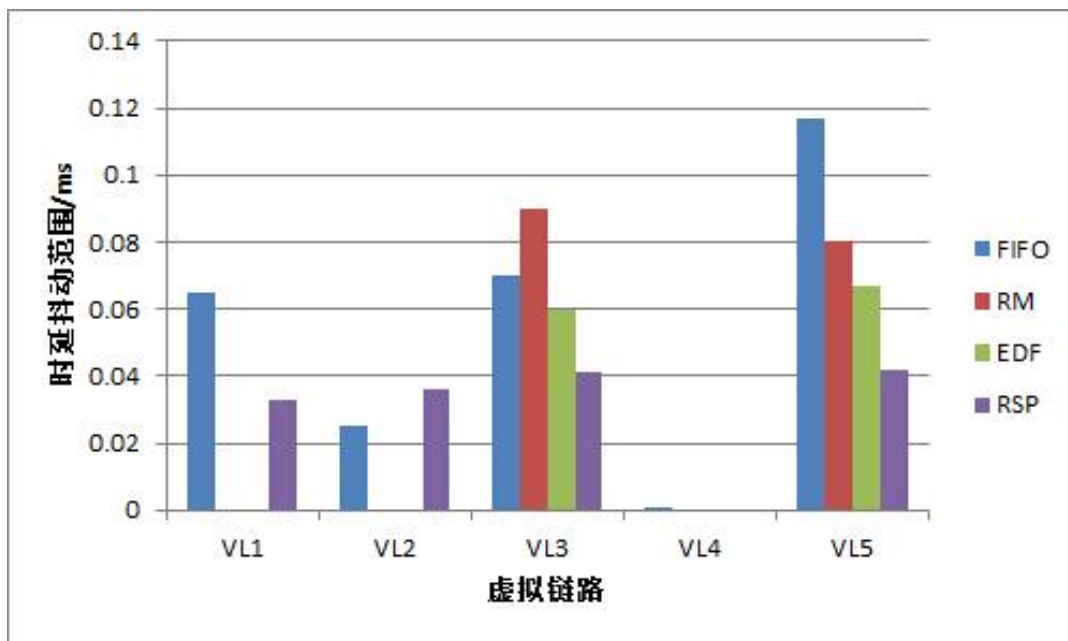


图 5-11 不同调度方法下时延抖动情况
Fig.5-11 Delay Jitter Using Different Method

从图中可以看出，对于 VL3 和 VL5 而言，RSP 方法有效的降低了时延的抖动范围，提高了链路的稳定性。对于 VL1 而言，由于 RM 和 EDF 方法下的 VL1 具有最高优先级，时延抖动很小，但在实际的网络中使用 RM 或 EDF 调度方法 VL1 会抢占高优先级链路的传输，而 RSP 方法相对于 FIFO 方式降低了 VL1 的时延抖动范围。

由以上结果分析，可以看出 RSP 方法对于保证关键链路 VL4 的优先传输，降低其他链路时延抖动，提高网络稳定性有很好的借鉴意义。

5.4 本章小结

本章完成了在 OPNET 平台上对 AFDX 网络的仿真和数据采集。主要工作包括 AFDX 仿真网络的搭建，仿真参数的设置，以及网络仿真、数据采集和结果分

析。在网络仿真中，模拟单链路工作情况和多链路工作情况，分析了两种情况下不同调度算法对网络性能的影响，论证了 **RSP** 方法对于降低网络时延抖动和保证关键链路优先传输的有效性。

第六章 结论与展望

伴随着航空电子技术的不断发展，航空电子系统和航空数据总线也在不断的更新换代。新一代的航空电子系统正向着集成度高、数据量大、实时性和稳定性要求高的方向发展。在这样的背景下，空客公司在其最新的 A380 飞机上研制了一种基于商用以太网的新型机载航空数据总线：航空电子系统全双工交换式以太网（AFDX）。并作为新一代航空数据总线的标准收录于 ARINC664 part7 中。

在 AFDX 网络中，使用虚拟链路对不同航空电子系统的数据进行逻辑隔离，并通过添加帧间最大间隔（BAG）来限制链路的最大流量。目前，AFDX 网络中使用的是 FIFO 的调度方法进行虚拟链路调度，针对 FIFO 方式下可能存在的时延抖动大和关键链路实时性无法得到保证的问题，我们提出了一种基于重整思想的链路调度方法，并通过 OPNET 平台建模对 RSP 方法进行了验证。

本文深入研究了 AFDX 网络标准，提出了一种新的调度算法 RSP 方法，并通过 OPNET 网络仿真平台对 AFDX 网络进行了建模仿真。文章主要的成果和结论如下：

（1）深入分析了 AFDX 网络标准，对其中虚链路机制，整形机制，调度机制进行了全面的研究，同时总结了衡量 AFDX 网络性能的主要参数：时延和时延抖动，描述了他们形成的原因和对网络的影响。

（2）针对传统调度方式时延抖动大，关键链路实时性无法得到保证的问题，提出了一种新的调度方法 RSP 方法，描述了 RSP 方法执行的流程，并通过对比分析总结了传统调度方式和 RSP 方式的优劣。

（3）在 OPNET 网络仿真平台上建立了 AFDX 网络的模型，针对航空电子系统、端系统和交换机等网络设备建立了对应的模型，每个网络设备都包括节点层模型和进程层模型，对模型中重要的模块和配置参数进行了说明。

（4）建立了一个仿真网络，并对该网络运行了仿真，得到不同调度算法下网络的性能参数。通过网络仿真，可以看出，RSP 方法有效的降低了网络的时延抖动，并能够保证关键数据的优先传输，对于提高网络的实时性和稳定性有重要的意义。

本文对航空电子系统网络标准的研究只是一个开始，目前完成的只是 AFDX 网络在 OPNET 仿真平台上的实现，并未在实际的硬件环境下进行实现。后续的工作可以在如下几个方面开展：

（1）AFDX 网络标准的硬件实现，在硬件环境下搭建 AFDX 网络环境，模

拟实际飞机的数据传输环境。

(2) 进一步改善 AFDX 网络的流量整形功能, 提高网络对突发数据的适应能力, 提高网络的稳定性。

(3) 拓展 AFDX 网络的 OPNET 模型, 完成 AFDX 网络应用层的 OPNET 模型, 形成完整的航空电子系统网络结构。

参考文献

- [1] ARINC specification 429, Aeronautical Radio Inc, 2001
- [2] C. E. Incorporated. MIL-STD-1553 Designer guide. 1982.
- [3] Aircraft Data Network Part 7:Deterministic Networks, ARINC664 Std, 2003
- [4] ARINC. Arinc project paper 664: Aircraft data network,part 7-avionics full duplex switched ethernet(afdx) network,2005.
- [5] J. G. Leishman, Network Calculus,Cambridge University Press, 2000
- [6] 熊华钢,李峭,黄永葵.航空电子全双工交换式以太网标准研究 [J].航空标准化与质量 2, no. 1 (2008): 25-28.
- [7] 周强,熊华钢.民机航电 AFDX 互连可靠性建模研究.遥测遥控 4 (2008): 009.
- [8] 杨万春,喻松,张杰,顾晚仪.数据通信网中的自相似模型.现代有线传输 3 (2004): 017.
- [9] 杨绪强,支高飞.新一代航空总线协议 AFDX.科技传播 17 (2010).
- [10] 杨孝平,尹春华.复杂网络可靠性评价指标.北京信息科技大学学报: 自然科学版 25, no. 003 (2010): 92-96.
- [11] 章筠.计算机网络可靠性分析与设计.PhD diss.,浙江大学,2013.
- [12] 朱玉建.工业网络实时性模型的分析.莱钢科技 3 (2008): 76-78.
- [13] 苟建兵.工业网络中以太网可靠性探讨.自动化博览 6 (1999): 26-31.
- [14] 石改辉,张原.下一代航空数据网络体系结构研究.电子工程 4 (2006): 29-33.
- [15] 刘晓娥,唐涛,万丽军,黄樟灿.基于链路可靠性的网络拓扑结构设计.武汉理工大学学报 (信息与管理工程版)3 (2002): 004.
- [16] 杨云, 熊华刚.计算 AFDX 延迟的网络演算方法, 电光与控制, 2008
- [17] 李硕, 王学望, 康锐.面向完整性要求的航空电子全双工交换式以太网可靠性评价参数研究.西安交通大学学报.47.3 (2013): 126-131.
- [18] 任向隆,马捷中.航空电子全双工交换式以太网终端系统研究[J].计算机测量与控制.2009(05)
- [19] 杜宏伟,马捷中.航空电子全双工交换式以太网及其关键技术研究[J].测控技术.2008(12)
- [20] 陈瑶,李峭, 赵长啸,熊颖.基于 OBDD 的航空电子网络可靠性分析.系统工程与电子技术,35(1),230-236.
- [21] 罗杰,霍曼.AFDX 通信链路技术及其在航空电子系统的应用 [C].全国第十届信号与信息处理, 第四届 DSP 应用技术联合学术会议论文集.2006.
- [22] 陈昕,周拥军,万剑雄.AFDX 端系统关键技术的研究与实现.计算机工程 35,no.5 (2009).
- [23] Bauer, Henri, J-L. Scharbarg, and Christian Fraboul. "Applying and optimizing trajectory approach for performance evaluation of AFDX avionics network." In Emerging Technologies & Factory Automation, 2009. ETFA 2009. IEEE

- Conference on, pp. 1-8. IEEE, 2009.
- [24]Hua, Yu, and Xue Liu. "Scheduling design and analysis for end-to-end heterogeneous flows in an avionics network." In INFOCOM, 2011 Proceedings IEEE, pp. 2417-2425. IEEE, 2011.
- [25]Boyer, Marc, and Christian Fraboul. "Tightening end to end delay upper bound for AFDX network calculus with rate latency FIFO servers using network calculus." Factory Communication Systems, 2008. WFCS 2008. IEEE International Workshop on. IEEE, 2008.
- [26]Dong, Song, et al. "The design and implementation of the AFDX network simulation system." Multimedia Technology (ICMT), 2010 International Conference on. IEEE, 2010.
- [27]Ridouard, Frédéric, J-L. Scharbarg, and Christian Fraboul. "Probabilistic upper bounds for heterogeneous flows using a static priority queueing on an AFDX network." Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on. IEEE, 2008.
- [28]Charara, Hussein, et al. "Methods for bounding end-to-end delays on an AFDX network." Real-Time Systems, 2006. 18th Euromicro Conference on. IEEE, 2006.
- [29]Bauer, Henri, J. Scharbarg, and Christian Fraboul. "Improving the worst-case delay analysis of an AFDX network using an optimized trajectory approach." Industrial Informatics, IEEE Transactions on 6.4 (2010): 521-533.
- [30]Li, Xiaoting, J. Scharbarg, and Christian Fraboul. "Improving end-to-end delay upper bounds on an AFDX network by integrating offsets in worst-case analysis." Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on. IEEE, 2010.
- [31]Anand, Madhukar, et al. "Formal modeling and analysis of the AFDX frame management design." Object and Component-Oriented Real-Time Distributed Computing, 2006. ISORC 2006. Ninth IEEE International Symposium on. IEEE, 2006.
- [32]Scharbarg, J., Frédéric Ridouard, and Christian Fraboul. "A probabilistic analysis of end-to-end delays on an AFDX avionic network." Industrial Informatics, IEEE Transactions on 5.1 (2009): 38-49.
- [33]J. Jasperneite, P. Neumann, M. Theis, and K. Watson. Deterministic Real-Time Communication with Switched Ethernet. In Proceedings of the 4th IEEE International Workshop on Factory Communication Systems, pages 11–18, Västerås, Sweden, August 2002. IEEE Press.
- [34]J.-Y. Le Boudec and P. Thiran, Network Calculus: A Theory of Deterministic

- Queuing Systems for the Internet. Berlin, Germany: Springer-Verlag, 2001, vol. 2050, Lecture Notes in Computer Science, ISBN 3-540-42184-X.
- [35] S. Martin and P. Minet, "Schedulability analysis of flows scheduled with FIFO: Application to the expedited forwarding class," in Proc. 20th Int. Parallel and Distrib. Process. Symp., Rhodes Island, Greece, Apr. 2006.
- [36] C. CHARARAH, FRABOUL C, Modeling and simulation of an avionics full duplex Switched Ethernet, Advanced Industrial Conference on Telecommunications/Service Assurance with Partial and Intermittent Resources Conference, 2005.
- [37] S. Martin and P. Minet, "Worst case end-to-end response times of flows scheduled with FP/FIFO," in Proc. 5th IEEE Int. Conf. Networking, Mauritius, Apr. 2006.
- [38] 陈文刚, 卢选民, 单长, 王平. 基于 AFDX 自适应优先调度算法的实时性分析. 测控技术 30, no. 10 (2011): 73-76.
- [39] 范红艳. 基于 OPNET 的路由协议性能分析. Master's thesis, 北京邮电大学, 2008.
- [40] 李俊鹏, 王勇, 白焱, 刘安. AFDX 虚连接自适应调度策略设计与实现. 计算机测量与控制 20, no. 007 (2012): 1986-1988.
- [41] 李哲, 田泽, 张荣华. AFDX 网络中 SkewMax 的研究. 计算机技术与发展 (2010).
- [42] 王威振. AFDX 网络中关键任务端到端延时优化. Master's thesis, 上海交通大学, 2013.
- [43] 宋东, 曾星星, 丁丽娜, 胡琼. AFDX 网络系统建模与仿真实现. 测控技术 31, no. 2 (2012): 76-80.
- [44] 李晓波. 基于组合调度的 AFDX 航空网络总线性能分析. Master's thesis, 上海交通大学, 2013.
- [45] 姜丽云. AFDX 网络关键技术研究. MS thesis. 西安电子科技大学, 2013.
- [46] <http://www.opnet.com.tw/>

致 谢

本文是在付宇卓老师的悉心指导和严格要求下完成的。从课题选题、项目开展到编码实现、论文写作，都凝聚了付老师的心血和汗水。在我的论文写作期间，付老师为我提供了专业耐心的指导和建议，并在论文初稿完成之后，仔细审阅了我的文章，提出了宝贵的修改意见。付老师渊博的知识、开阔的视野、严谨的治学态度和精益求精的研究态度都给我留下了深刻的印象；同时，付老师严于律己、宽厚待人的品格也是我今后做人和做事的榜样。没有付老师的无私帮助和细心指导，我的毕业论文不会完成的这么顺利，在此向付老师表达崇高的敬意和深深的感谢。

本文还要感谢实验室刘婷老师在项目开展和论文写作期间帮助我解决了遇到的各种问题。刘老师思路清晰，逻辑缜密，在项目遇到困难或论文遇到难题时，能够很快指出问题所在并给予精确完整的解决方案，同时，刘老师也能够细心审阅我的毕业论文和学术论文，指出了很多问题。在此对刘老师的关心和指导表示深深的感谢。

本文还要感谢整个 ICAT 实验室的师兄师姐和师弟师妹们。在进行毕业设计的这段时间里，焦佳佳博士对文章的内容提出了很多意见和建议，韩兴博士在设计实验方案和实际编码中给予了很多宝贵的指导意见，楚朋志博士和童随兵博士在项目的开展和论文的写作中帮助我避免了很多弯路。在此，感谢这些师兄师姐们给我的很多启发和指导意见。同时，还要感谢李维嘉、李涵和赵东升同学，我们经常一起讨论项目的进展，交流项目中遇到的问题，彼此给出可行的建议，同时也包括生活中遇到的一些问题，感谢你们给与我的无私的帮助。最后，也希望实验室的师弟刘芳铭、孟帅帅、廖若晨、孙晓奇、余先波以及师妹张金玲、黄燕清能够在科研的道路上继续努力，为实验室添光添彩。

在临近毕业之际，感谢上海交通大学在两年半的研究生生活中给我提供的优质的资源和广阔的平台。对所有给予我帮助和指导的老师和同学表示由衷的谢意，感谢你们的辛勤栽培和无私关怀。

最后，再次感谢，所有关心和帮助过我的老师和同学们。

攻读硕士学位期间已发表或录用的论文

[1] 第一作者 .A new re-shaping method for reducing delay jitter on AFDX network, ICITEC 2014.(已录用)