

中图分类号: TP302.1
学科分类号: 081200

论文编号: 1028716 15-S025

硕士学位论文

基于网络演算的 AFDX 网络确定性 评估的研究与实现

| | |
|---------|-----------|
| 研究生姓名 | 孙 权 |
| 学科、专业 | 计算机科学与技术 |
| 研究 方 向 | 航空电子系统安全性 |
| 指 导 教 师 | 王立松 副教授 |

南京航空航天大学

研究生院 计算机科学与技术学院

二〇一五年一月

I

Nanjing University of Aeronautics and Astronautics
The Graduate School
College of Computer Science and Technology

Research and Implementation of Deterministic Assessment in AFDX Network Based on Network Calculus

A Thesis in

Computer Science and Technology Engineering

by

Sun Quan

Advised by

Associate Prof. Wang Lisong

Submitted in Partial Fulfillment

of the Requirements

for the Degree of

Master of Engineering

January, 2015

摘 要

AFDX 网络作为现代航空电子系统网络的重要体系结构, AFDX 网络确定性评估的准确性是保障整个航空电子系统安全性的关键, 发达国家使用 AFDX 的机型都必须采用数学方法来证明网络的确定性性能需求是否得到了保证。目前主要的确定性性能分析方法有轨迹法、模型检测和网络演算, 相对于前两种方法网络演算能够利用简单的证明提供确切的边界, 这对于航空电子系统的适航认证尤其重要, 但是由于航电网络的流量特性存在诸多随机影响因素, 因此现有的网络演算得出时延抖动的上界普遍偏大, 造成航电网络确定性需求虽然得到满足, 但是资源配置和分配不合理, 浪费很大。为了获得更精确的边界值, 需要对传统的网络演算算法进行改进, 以提高资源配置的合理性, 从而提高飞机系统整体性能。

针对上述问题, 本文完成了以下主要工作:

首先对 AFDX 进行了分析和研究, 根据 AFDX 网络传输过程和特征, 设计了网络时延模型。

其次, 针对航空电子系统对网络实时性、确定性的需求, 设计了基于多链路分组技术的网络演算算法, 该算法评估获得更为紧致的上界, 可以为适航认证提供证据, 同时获得不同信息流端到端延时的分布, 为改善 AFDX 网络系统性能提供了更准确的依据。

最后, 基于 EMF 设计并实现了一个航电系统网络确定性分析插件, 并集成到了 IMA 资源配置工具 IMACT 中, 使得 IMACT 具有了对 AFDX 网络系统进行确定性分析的能力。

关键词: AFDX, 网络演算, 端到端延迟, 确定性网络传输, 上界

ABSTRACT

AFDX network is important network architecture of modern avionics system and the accuracy of deterministic assessment in AFDX network is the key to guarantee the safety of avionics system. Developed countries which using AFDX models must be adopting mathematical method to prove that network deterministic performance requirement is satisfied. At present, the main deterministic performance analysis methods have trajectory, model checking and network calculus. Network calculus can provide proofs of the exact boundary using simple method compare with the former two methods, this is especially important for the airworthiness certification of aviation electronic system. Due to the flow characteristics of avionics network has many random factors, the upper bound of delay and jitter with network calculus existing are generally large which resulting in although avionics network deterministic demand is satisfied, the allocation and configuration of resources are unreasonable, large waste. In order to obtain more accurate boundary value, we need to improve the traditional algorithm of network calculus, so as to improve the rational allocation of resources and improve the overall performance of aircraft system.

This article finished the following works:

Firstly, we analyzed and studied AFDX protocol and designed the network delay model according to transmission process and characteristics in AFDX network.

Secondly, according to the requirement of deterministic and real-time in avionics system, we design a new network calculus algorithm based on multi-link grouping technology which can get tighter upper bound. At the same time we can obtain different end to end delay distribution with different information flow which provides more accurate basis for improving the performance of AFDX network system.

Finally, we design and implement deterministic analysis plug-in in avionics system network based on EMF, and integrated in the IMA resource configuration tool IMACT which makes IMACT have ability to analysis deterministic of AFDX network system.

Keywords: Avionics Full Duplex Switched Ethernet (AFDX), network calculus, end to end delay, deterministic transmission, upper bound

目 录

| | |
|----------------------------|----|
| 第一章 绪论..... | 1 |
| 1.1 课题研究背景..... | 1 |
| 1.2 国内外研究现状..... | 3 |
| 1.3 研究目的和意义..... | 5 |
| 1.4 主要研究内容..... | 6 |
| 1.5 论文组织结构..... | 6 |
| 第二章 航空电子机载网络研究..... | 8 |
| 2.1 AFDX 网络技术研究..... | 8 |
| 2.1.1 终端与航电子系统..... | 11 |
| 2.1.2 虚拟链路..... | 13 |
| 2.1.3 虚拟链路调度与抖动..... | 14 |
| 2.1.4 冗余管理..... | 16 |
| 2.1.5 传输协议栈..... | 17 |
| 2.2 民用航空网络性能要求与建模..... | 19 |
| 2.2.1 AFDX 网络性能参数..... | 19 |
| 2.2.2 AFDX 网络时延模型..... | 19 |
| 2.3 常用 AFDX 网络确定性分析方法..... | 23 |
| 2.3.1 模型检测..... | 23 |
| 2.3.2 轨迹法..... | 23 |
| 2.4 本章小结..... | 25 |
| 第三章 基于多链路分组技术的网络演算..... | 26 |
| 3.1 网络演算..... | 26 |
| 3.1.1 基本概念..... | 26 |
| 3.1.2 统计网络演算..... | 29 |
| 3.2 网络计算方式..... | 30 |
| 3.2.1 VL 简单流模型..... | 30 |
| 3.2.2 只为突发数据付费一次..... | 30 |
| 3.2.3 组与组的包络..... | 32 |
| 3.3 多链路分组技术的分析..... | 33 |

| | |
|---------------------------------------|----|
| 3.3.1 精确流分析..... | 33 |
| 3.3.2 多链路分组..... | 37 |
| 3.4 算法设计实现..... | 40 |
| 3.4.1 AFDX 的网络建模..... | 40 |
| 3.4.2 设计与实现..... | 41 |
| 3.4.3 实验结果分析..... | 45 |
| 3.5 本章总结 | 47 |
| 第四章 多链路分组网络演算的 AFDX 网络确定性分析功能实现 | 48 |
| 4.1 系统概述 | 48 |
| 4.2 开发环境与关键技术..... | 51 |
| 4.2.1 系统架构介绍..... | 51 |
| 4.2.2 关键技术..... | 52 |
| 4.3 AFDX 建模..... | 53 |
| 4.4 实例分析 | 59 |
| 4.5 本章总结 | 63 |
| 第五章 总结与展望 | 64 |
| 5.1 工作总结 | 64 |
| 5.2 研究展望 | 64 |
| 参考文献 | 66 |
| 致 谢 | 70 |
| 在学期间的科研成果及发表的学术论文..... | 71 |
| 附录 | 72 |

图表清单

| | |
|-------------------------------|----|
| 图 1.1 航电体系结构发展..... | 2 |
| 图 1.2 边界值与实际值..... | 4 |
| 图 2.1 AFDX 网络..... | 9 |
| 图 2.2 AFDX 网络帧格式..... | 9 |
| 图 2.3 AFDX 协议层次..... | 10 |
| 图 2.4 AFDX 与 ARINC429 架构..... | 11 |
| 图 2.5 终端模型..... | 11 |
| 图 2.6 取样端口..... | 12 |
| 图 2.7 队列端口..... | 13 |
| 图 2.8 以太网帧的目的地址格式..... | 13 |
| 图 2.9 帧路由例子..... | 14 |
| 图 2.10 虚链路调度..... | 15 |
| 图 2.11 虚链路整流器..... | 15 |
| 图 2.12 BAG 整流..... | 16 |
| 图 2.13 A 与 B 网络..... | 16 |
| 图 2.14 以太网帧接收过程..... | 16 |
| 图 2.15 冗余帧到达时序图..... | 17 |
| 图 2.16 AFDX 发送协议栈..... | 18 |
| 图 2.17 AFDX 接收协议栈..... | 18 |
| 图 2.18 AFDX 拓扑..... | 19 |
| 图 2.19 交换机中帧处理过程..... | 20 |
| 图 2.20 交换机模型..... | 20 |
| 图 2.21 AFDX 配置..... | 23 |
| 图 2.22 VL1 调度..... | 24 |
| 图 3.1 水平偏差与垂直偏差..... | 28 |
| 图 3.2 到达曲线位移..... | 30 |
| 图 3.3 组示例..... | 31 |
| 图 3.4 累积服务曲线..... | 31 |
| 图 3.5 串联服务曲线..... | 31 |

| | |
|--------------------------------|----|
| 图 3.6 聚合流示例..... | 32 |
| 图 3.7 数据流组的到达曲线..... | 33 |
| 图 3.8 精确到达曲线..... | 34 |
| 图 3.9 受流量管制的到达曲线..... | 34 |
| 图 3.10 离散数据流..... | 35 |
| 图 3.11 利用各精确流的比较..... | 35 |
| 图 3.12 不同服务曲线下积压..... | 36 |
| 图 3.13 100Mb/s 下积压 | 36 |
| 图 3.14 500it 突发下积压 | 37 |
| 图 3.15 单链路分组的到达曲线..... | 38 |
| 图 3.16 单链路分组叠加..... | 38 |
| 图 3.17 多链路分组..... | 38 |
| 图 3.18 多链路叠加示意..... | 39 |
| 图 3.19 网络拓扑..... | 40 |
| 图 3.20 需求架构..... | 41 |
| 图 3.21 环路检测..... | 42 |
| 图 3.22 终端可达..... | 43 |
| 图 3.23 VL 最差情况下延迟 | 43 |
| 图 3.24 最差情况下的 VL 抖动 | 43 |
| 图 3.25 VL 分配的带宽 | 44 |
| 图 3.26 交换机输出端口的最在帧延迟..... | 44 |
| 图 3.27 终端输出端口最大积压..... | 44 |
| 图 3.28 交换机输出端口最大积压..... | 45 |
| 图 3.29 交换机端口延时..... | 46 |
| 图 3.30 Smax 对于 VL1 的延迟影响 | 46 |
| 图 3.31 BAG 对于 VL1 的延时影响..... | 47 |
| 图 4.1 系统框架图..... | 48 |
| 图 4.2 系统建模流程设计..... | 49 |
| 图 4.3 系统顶层部分模型..... | 49 |
| 图 4.4 管理员用例图..... | 50 |
| 图 4.5 系统设计员用例图..... | 51 |
| 图 4.6 系统架构 | 51 |

| | |
|----------------------------|----|
| 图 4.7 BIRT 使用流程..... | 53 |
| 图 4.8 AFDX 结构建模..... | 54 |
| 图 4.9 AFDX 构建的 IMA 系统..... | 55 |
| 图 4.10 网络分析活动..... | 56 |
| 图 4.11 网络确定性分析流程..... | 56 |
| 图 4.12 虚链路分析无效示例..... | 58 |
| 图 4.13 水处理系统..... | 59 |
| 图 4.14 系统虚链路..... | 60 |
| 图 4.15 水处理系统(IMA)..... | 61 |
| 图 4.16 全局参数..... | 61 |
| 图 4.17 IMA 分析过程..... | 62 |
| 图 4.18 报告列表..... | 62 |
| 图 4.19 分析报告..... | 63 |
| 表 1.1 常用机载数据总线对比..... | 1 |
| 表 2.1 工业配置下端到端延迟的总结..... | 24 |
| 表 2.2 工业配置下端到端延迟的比较..... | 24 |
| 表 3.1 网络时延结果对比..... | 45 |
| 表 4.1 AFDX 网络模型中各类的说明..... | 54 |
| 表 4.2 确定性分析-拓扑分析..... | 58 |
| 表 4.3 确定性分析-环路分析..... | 58 |
| 表 4.4 链路配置报表..... | 60 |
| 表 4.5 参数列表..... | 61 |

缩略词

| 缩略词 | 英文全称 | 中文全称 |
|---------|---|------------------|
| AFDX | Avionics Full Duplex Switched Ethernet | 航空电子全双工交换式以太网 |
| AFDX ES | Avionics Full Duplex Switched Ethernet End System | 航空电子全双工交换式以太网端系统 |
| BAG | Bandwidth Allocation Gap | 带宽分配间隔 |
| COTS | Commercial-off-the-shelf | 商用现成品或技术 |
| EMF | Eclipse Modeling Framework | 基于 Eclipse 的模型框架 |
| FAA | Federal Aviation Administration | 美国联邦航空管理局 |
| GEF | Graphical Editor Framework | 图形化编辑框架 |
| HFRQ | Hosted Function Requirements | 驻留功能需求 |
| HSDB | High Speed Data Bus | 高速数据总线 |
| IMA | Integrated Modular Avionics | 综合模块化航空电子 |
| IPM | Integrated Processing Module | 综合处理模块 |
| LRM | Line Replaceable Modules | 外场更换模块 |
| LRU | Line Replaceable Units | 外场更换单元 |
| MTU | Maximum Transmission Unit | 最大传输单元 |
| RDC | Remote Data Collector | 远程数据采集器 |
| SOA | Service-Oriented Architecture | 面向服务的体系结构 |
| SP | Static priority | 静态优先级 |
| SNMP | Simple Network Management Protocol | 简单网络管理协议 |
| SWT | Standard Widget Toolkit | 窗口小部件工具箱 |
| UD | Usage Domain | 功能域 |
| VL | virtual Link | 虚链路 |

第一章 绪论

1.1 课题研究背景

当前航空器机载设备数量日益增加及其功能日益复杂意味着交换数据量将会显著上升^[1]，并且传递的数据不再是单纯的控制命令和状态信息（如早期航空电子系统），总线中不但传递着大量的实时控制消息，而且要求总线能够传递文件等冗长信息，在处理器到处理器、处理器到I/O等不同模块（子系统）之间，数据互连的带宽需求相差很大，这也是为什么在以“宝石柱”计划为代表的美军第三代航空电子系统中引入了7种总线和网络技术的原因。通过对7种机载总线的综合研究，从数据传输速率、单消息字节数和最大节点数等方面进行对比，结果如表1.1所示。

表 1.1 常用机载数据总线对比

| | | | | | | | |
|-------|-----------|-----------------------|---------------------------|------|-------------|------------------------|-----------|
| 总线技术 | ARINC 629 | MIL-STD-1553B | MIL-STD-1773 | HBDB | LTPB | FDDI | AFDX |
| 传输速率 | 2 | 1 | 1 | 20 | 20 | 100 | 100 |
| 通信方式 | 双向 | 双向 | 单向 | 双向 | | | 双向 |
| 消息字节数 | 256 | 32 | 32 | 4096 | 4096 | 2250 | 1471 |
| 终端数量 | | 32 | 32 | | 128 | 500 | 可级联扩展 |
| 拓扑结构 | 总线型 | 总线型 | 总线型 | | 星型、总线型 | 环型 | 星型 |
| 总线长度 | 300 英寸 | 190m | | | 1km | 1000km | >100m |
| 应用飞机 | B777 | F-15、F-16、B-52、KC-135 | SAMPEX 探测卫星 F-18、F-117 实验 | | F-22、RAH-66 | 美海军 GNCR、B777、“自由”号空间站 | A380、B787 |

ARINC-629 总线^[2]相对于 20 世纪 70 年代投入研发使用的 ARINC-429 机载数据总线，结构更加合理、速率更高。ARINC-629 数据总线是无主控机多发送器的广播式数据总线，由波音飞机公司提出、研制，并应用到 B777 商用飞机上。使用 ARINC-629 数据总线的显著特点是可在 LRU 之间使用较少的导线以较高的速率传递较多的数据，具有更高的工作可靠性，基本满足传统航空电子系统数据的传输要求，但是 ARINC-629 总线需要开发大量的专用组件，增加了成本，

多种功能也不能共享同一个模块。军用机载数据总线 MIL-STD-1553B 以及在此基础上发展的 MIL-STD-1773 总线由于传输速率低、终端数目少等原因正逐渐被光纤通信(FC)、1394 串行总线以及以太网等高速总线技术取代。同时随着计算机技术、通信技术、微电子技术和网络技术的发展,航空电子系统从综合化向着以信息交换为中心、高度综合化的深层发展,因此而产生的大量复杂的数据也急需超高速数据连接总线进行传输。另外,随着航空电子系统的复杂性的增加,为了确保各种复杂飞行条件下的安全飞行和乘客的需求,对于安全和关键的系统提供高速、可靠的实时通信也是必要的,且是基本的要求,而这也是未来机载数据总线的发展趋势。

面对航空电子系统对机载数据总线提出的时间确定性传输、可靠性高和质量轻的要求,空中客车公司与波音公司在 20 世纪就已经将重点转移到开发商用以太网技术来构建下一代的机载数据总线,这项研究促进了航空电子全双工交换式以太网 AFDX 的发展。航空全双工交换式以太网(AFDX)主要是应用在航电系统中传输飞机确定性数据的一种总线系统^[3-4]。空客 A380 则应用了 AFDX 网络,其传输速率是 ARINC-429 的几千倍。另一方面,采用大量 COTS 产品诸如电缆、测试设备等其他以太网的硬件设备,而这些设备经过市场的检验,被市场检验应用和验证,可以节省在飞机数据总线网络中的投入。此外,以太网有着长时间的行业运营经验,使我们对硬件的可靠性和易维护性更有信心,这也是 AFDX 基于以太网的主要原因。而数据总线作为航空电子系统的“神经”,直接与航空电子体系结构的发展同步^[3]。因此了解航空电子体系结构的发展过程,有助于理解机载数据总线的特点。如图 1.1 描述了航空电子体系结构从 40 年代到如今的发展过程。

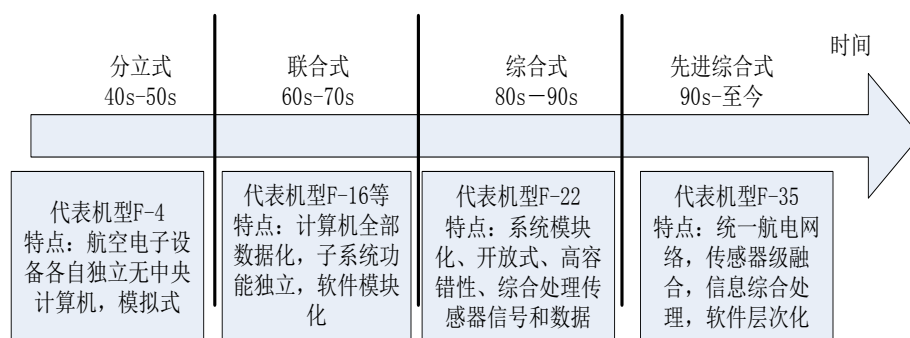


图 1.1 航电体系结构发展

从图 1.1 和表 1.1 看出,民用机载数据总线发展过程中,最初 ARINC-429 传输速率只有 100kb/s,而发展到 ARINC-629,其传输速率达到 2Mb/s,目前 ARINC-664 传输速率实现 100Mb/s。而且随着以太网技术的发展,其传输速率仍然会进一步提高。在总线传输速率不断提高的同时,总线结构从 ARINC-429 的网状形式,发展到 ARINC-629 的总线型,再到 ARINC-664 的星型结构,结构更趋合理,可靠性进一步提高,设备互连也更加方便。但航空电子系统通信过程中的可靠性、数据吞吐率和实时性、安全性始终在网络技术的发展变革中起着主导作用,其重要性

也越来越大。作为基于以太网的 AFDX 解决方案,它利用了交换式网络优点,如取代了共享介质的总线或传统的分立式电缆连接,克服了后者如布线复杂,维护成本高,改型代价大等缺点,同时必须克服以太网本身的硬件处理的不确定性的缺点,保障了网络的确定性,使得其能够符合航电网络中的相关要求并应用在航电领域。因此,AFDX 在以太网基础上增加了确定性定时机制和可靠的信息传输机制来适应航电应用,确定性定时则主要是通过增加规定帧间传输间隔和采用虚拟链路技术加以完成,并采用对帧管理机制从而达到信息的可靠传输与通信,例如信息的传输采用了双冗余机制以及在终端对信息进行完整性检查。然而由于适航当局对 AFDX 网络的性能有着严格要求,所以发达国家使用 AFDX 的机型都必须采用数学方法来证明其网络性能的确切性^[5-8]。而 AFDX 是基于商用以太网的星型分组交换结构,由于交换节点必然存在的拥塞情况,导致数据传输的端到端时延存在一定抖动,同时数据存在由于拥塞而被丢弃的可能。如何在 AFDX 数据传输中保证端到端时延上界并且为缓冲区分配合适的缓存大小成为了 AFDX 网络满足航电系统确定性要求和通过适航认证所重点关注的问题。

1.2 国内外研究现状

目前使用 AFDX 的机型包括了空客 A380,A400M 以及 A340-500,-600,A318,A319,A320,A321 的座舱娱乐网络,波音 B777 AIMS 改进以及 B747,B767,B777 IFE 项目,苏霍伊超级喷气机 Sukhoi Superjet100,直升机 AgustaWestland AW101, AgustaWestland AW149, Irkut MS-21,庞巴迪 Bombardier CSeries,中国商飞 Comac ARJ21, C919。对于 AFDX 网络进行基础研究的机构包括:法国的 Toulouse(图卢兹)大学和法国 INRIA(法国国立计算机及自动化研究院),而欧洲对 AFDX 的研究一直在持续。国内对于 AFDX 分析较多的则是北航熊华钢教授的团队,包含了 AFDX 标准研究以及相应的端系统调度算法和交换机的调度算法研究。周强等人分析了 A380 的 AFDX 互联模型,对任务进行分级,分析任务可靠性并提出了可靠性模型,并实际应用在 A380 中的互联 AFDX 网络,进行可靠性分析。

目前被广泛研究的可以进行确定性分析方法有随机排队论,模型检测,轨迹法和网络演算。最为成熟也被用于实践的是网络演算,其中空客的 A380 就采用了网络演算方法进行适航认证。当采用随机排队论等方法来研究进行网络分析,没有考虑与实际网络的拟合程度,特别是随着网络应用的发展,不能提供能够保证网络端到端性能的确保服务。此外由于存在“只为突发数据付费一次现象”,数据通过多个节点时的最坏时延要比各个节点的最坏时延的简单累加要小,所以该方法不再适合,需要研究更精确的端到端性能的计算方法,不采用模型检测和轨迹法主要是由于模型检测在针对于航空电子复杂网络时,会导致状态爆炸式增长,从而使分析的难度加大,甚至无法进行分析,而轨迹法在多数情况下,虽然能够获得紧致的上界,但轨迹法不同于整体理论的方法,它是在假设最悲观条件下进行信息的传输,通过信息传递的每个访问节点,

而不考虑整个网络的节点对其的影响，构建数据包在每个节点中的时序，因此主观性强，缺乏数学可证明的能力，不适合适航认证高要求的需求，同时直观上的最差情况无法证明是性能上界的情况。网络演算理论是由 R.L.Cruz 等人首次提出，并由 C.S.Chang 和 J.Y.Le Boudec 等人发展起来，提出基于到达曲线和服务曲线的网络分析方法和网络环境中的重要定理和结论，在这基础之上 Boudec 深入研究并总结关于网络演算的结论和定义，将之发展完善并系统化为网络演算理论体系，即实际应用于网络环境性能分析的系统理论。而网络演算另一分支统计网络演算正在缓慢发展，由于性能确定边界往往是最坏情况下的解，而网络系统处于这种情况的几率是相当低的，并在一些保证服务网络中使用确定解可能导致网络资源的过于优化。因此，与确定网络演算对立的统计网络演算理论得到了相应的发展。统计网络演算是用来定量求解网络性能的统计边界。利用统计网络演算能够从统计复用独立通信流中获得更大的增益，并有效地提高资源利用率。虽然它能克服确定网络演算的缺点，但在实践使用中比较复杂。而网络演算真正的目的是提供性能上界，即在最坏情况下也绝不超出的值。在实践中，预计会出现图 1.2 所示的情况：

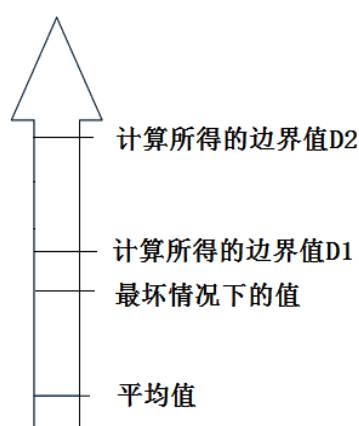


图 1.2 边界值与实际值

上图 1.2 是在一个给定的交换机端口，某个队列尺寸的情况。在设计阶段，该队列的大小很可能比最坏情况下小得多，因为最坏的情况只有在非常特殊的条件下才可能发生，如多个帧数据帧同时到达。因为在证明计算过程中总是保守地采用了大量近似值，如在流量建模、网络建模和各种计算方法，可以认为计算所得的边界绝对比实际值大。所以这些保守的假设解释了为什么我们的分析方法实际上并不能用于研究网络运行，而只是给出某些值的边界。但作为边界值 D1 的指导意义要远远大于边界值 D2，利用边界值 D1 分配缓冲区资源要明显优于边界值 D2，特别是在航空电子网络领域中，避免了资源的浪费。同时边界值 D1 为适航认证提供了更精确的计算结果。根据上述分析，网络分析的目的使计算所得的值更加接近于最坏情况下实际测量的值，要做到这一点，就应该减少网络分析各个阶段所做的加法。第一是改进网络模型，开发与硬件制造商选用的元件更接近的模型，虽然可以更精确代表实际的应用情况，但当更换

交换机时，模型也就失效了。同时由于飞机制造商各个部门组织问题相关的行业特色，这种方法从技术角度来说难以执行（难以评估当前的或短期的实际需要）。最后，潜在效益最大的改善即是改变计算方法与流量建模，也是本文基于的网络演算的方法。

国内的赵永库等人在分析了 AFDX 网络^[4]中信息传输的延迟特点以及对应的分布，设计了最坏情况下的 AFDX 网络延迟计算方法。杨云等人则更进一步，采用了网络演算理念计算 AFDX 网络延迟，并指出此结果可作为适航依据。国外 Henri Bauer, Jean-Luc Scharbarg, Christian Fraboul 集合 NC 理论中分组技术，采用轨迹法对 FIFO 调度策略的 AFDX 网络最差情况下的延迟进行优化，同时将 Trajectory 与 Network Calculus 方法的结果在 BAG 与 S_{\max} 两个维度分别进行了分析比较，并提出了下一步的工作，即集合 Trajectory 与 Network Calculus 方法的优点，进行网络的分析。而北航刘成^[5]等则面向工程化应用更进一步，设计了 AFDX 网络虚拟链路的路由配置算法，对航空电子网络配置提供了研究方向。

1.3 研究目的和意义

根据 FAA 针对民用航空器适航认证的要求，民用航电系统网络总线必须能提供数据传输“确定性”的能力^[6]，满足航空电子网络性能相应的约束。鉴于航空电子系统必须通过测试认证，这点尤其重要。在认证的过程中，整机制造商必须向认证机构—欧洲联合航空局提供证据，证明飞机的各种性能都符合要求，作为关键性的航空电子网络，必须按照最严格的标准进行认证，然而综合化航空电子系统的复杂性增加了认证难度。而常用的方法过于直观或获得的网络性能上界过大，无实际意义，所以无法用于适航的认证^[7]。

为了满足适航要求，保障航空电子网络性能需求，采用基于最小加代数的一系列结论网络定量分析理论网络演算，对网络进行分析并获得定量公式，并对 AFDX 网络中性能指标如网络时延、时延抖动以及队列缓冲区大小等的作定量的研究，从而确定网络性能优劣。但是 Cruz 的基本网络演算理论，未考虑到航电网络的诸多流量特性，采用最一般的方法会导致实际网络的性能无法得到精确的描述，会造成无法达到认证的需求和按一般方法分析的性能结果进行航电系统的配置造成资源的浪费和配置复杂性的增加。而性能分析的结果对整个航空电子系统配置优化重构都有着重要的指导意义。航空电子网络性能优劣总是跟安全性分析相关的，同时也是安全性的重要方面。为了飞机的安全，航空电子网络的性能需求满足飞机各个系统级别的需求，如数据通讯系统中时延的确定性保证。

由此可以看到，对包括抖动、延迟等性能参数的上界范围的获得，是网络确定性性能的重要指标。在证明网络性能界限的基础上，分析网络性能方法优劣则影响着认证过程，越紧致的结果越是能精确的描述网络性能，对资源的分配有着重要的指导意义，对整个航空电子系统性能的提升起着至关重要的作用，进而关系到整个航空电子网络的安全性，可靠性。本论文的目的

的就是针对网络确定性性能问题，分析网络性能的关键因素和 AFDX 网络的特点，并改善基于一般网络演算的分析方法，精确评估网络确定性性能的分析结果，获得更具有指导意义的性能参数上界值，为航电配置和适航认证提供依据，设计针对基于网络演算理论的网络性能评估分析的一般算法。

1.4 主要研究内容

本论文在综合化航空电子的背景下，分析了航空电子全双工交换式以太网的网络协议标准和比较分析了常用网络分析方法诸如模型检测(Module Check)、轨迹法(Trajectory)，对比以上方法的在描述航空电子评估网络性能上优劣，通过改进传统网络演算(Network Calculus)并指出其在 AFDX 网络设计中的应用，最后基于一个典型的综合化航空电子系统-机载水处理系统进行其 AFDX 网络的分析，以评估网络性能是否满足需求。

论文的主要工作有：

1、研究 ARINC664 协议中的第七部分^[8]，即 AFDX 协议部分，并根据 AFDX 网络传输过程和特征，设计了网络一般时延模型。

2、研究 AFDX 网络性能分析方法，对比分析常用网络分析的方法，设计基于多链路分组技术的网络演算算法，同时对输出流曲线做更精确化调整，从而改进传统的网络演算分析方法，从而获得更紧致的上限，更精确的描述航空电子网络性能，优化航空电子网络性能结果。利用分析的结果，使分析结果能更好的指导 IMA 设计。

3、将本文提出的多链路分组算法应用于工程实践中，在 IMA 配置工具 IMACT 中，基于插件开发应用本方法，使得 IMACT 具有了对 AFDX 网络系统进行性能分析的能力，可以进行 IMA 系统的评估验证分析。

1.5 论文组织结构

本论文一共分为五章，内容组织安排如下：

第一章 绪论，介绍了本课题的相关背景、国内外当前研究现状、研究目的和意义以及论文的主要工作以及组织结构安排。

第二章 介绍了 AFDX 协议的相关定义、协议规则以及关键技术，为后面 AFDX 性能的分析做好准备。设计了 AFDX 网络时延模型，研究常用的网络分析技术，并进行比较分析。

第三章 介绍说明了网络演算在网络性能分析中的相关理论，精确分析流量特性，提出基于多链路分组的技术，使其应用在传统网络演算过程中，对网络演算的评估结果进行优化，以获得更紧致的上限，同时设计了网络性能评估的分析算法。

第四章 基于综合化航空电子系统设计工具 IMACT，设计开发基于多链路分组技术的网络性能分析插件，使 IMACT 能够对综合化航空电子系统设计进行网络性能分析，一方面为适航认证提供证据，另一方面评估 IMA 系统配置的优劣。

第五章 总结与展望，总结了论文的研究成果，并指明后续待改进地方以及未来重点研究方向。

第二章 航空电子机载网络研究

机载航空电子系统是飞机上安装的全部装置、信息系统或电子设备的组合，是一个复杂的系统，包括计算机及其运行的软件、传感器、处理器以及各单元之间的通信工具。在当今时代，这种航空电子系统具有许多功能，例如，飞行控制、自动驾驶、导航、信息显示或相应的所谓“舒适性”功能，如控制指令的反馈功能。关于这些系统功能的深入研究可参见著作[9]。航空电子系统正日益复杂：从 1983 年(A310)到 1993(A340)，机载设备的数量增加了 50%(由 77 到 115)，而到后来的 A380，机载设备达到了 22 吨，一般情况下，人们认为航空电子系统的复杂度每 5 年增加一倍^[10]，如今航空电子系统的成本比重正在逐年增加，约占民机总成本的 40%以上^[11]，其重要性日益显著。尽管航空电子系统近似于分布式计算机系统，但因此它所受到的约束会更大，如由于成本的原因需要对体积和重量进行限制，并保证机载系统在恶劣的条件下（高温、振动、电磁环境等）的正常运行。航空电子系统与传统计算机系统的另一个区别是运行可靠性方面的限制。事实上，在飞行任务中，极小的故障都可能危及机乘人员的生命安全。如近来马航 MH370(B777)和亚航 QZ8501(A320)飞机失事，造成了不可估量的经济损失。因此 DO-178B^[12]和 ARP 4754^[13] 标准将系统或设备分为三类：关键系统、基本以及非基本的系统，分别进行安全级别的定义，而承载数据传输的航空电子网络也获得了发展。

2.1 AFDX 网络技术研究

民用飞机在商用以太网基础上进行了改进和扩充，提出了航空电子全双工交换以太网技术，同时制定了 ARINC664 标准，ARINC664 协议目前已经有中文的翻译版本颁布，包括 2007 年一飞院自己翻译的中文版本以及 2008 年 301 所联合北航翻译了一个的中文版本。ARINC664 规范一共包含了八个部分，其中第六部分(Part6)保留，其中的 Part7 即 AFDX 相关规范，描述了如何将商用以太网各层协议映射到机载网络应用中，以及如何将商用以太网改造为适用于航电系统的一种实时以太网。目前该标准已广泛应用到新一代民用飞机（如 A380，波音 787）和大型运输机(A400M)的航空电子主干网络上，国内也开展了对该网络的研制，并应用于大型军用运输机、大型客机等适用机型。

虽然 ARINC664 Part7 阐述了有关终端和交换机设计实现的框架性需求，但是并未告知我们为什么要实现这些需求，以及哪些需求是和确定性有关，进而哪些是可选的，哪些是必须的，哪些是可以采用其他实现方法的以及哪些部分可以改进与自定义的。本节介绍了 AFDX 网络的特点与相关规范，AFDX 网络系统有几个基本要素：航电子系统、AFDX 网络终端系统、AFDX 互联结构^[14]，如图 2.1 所示。

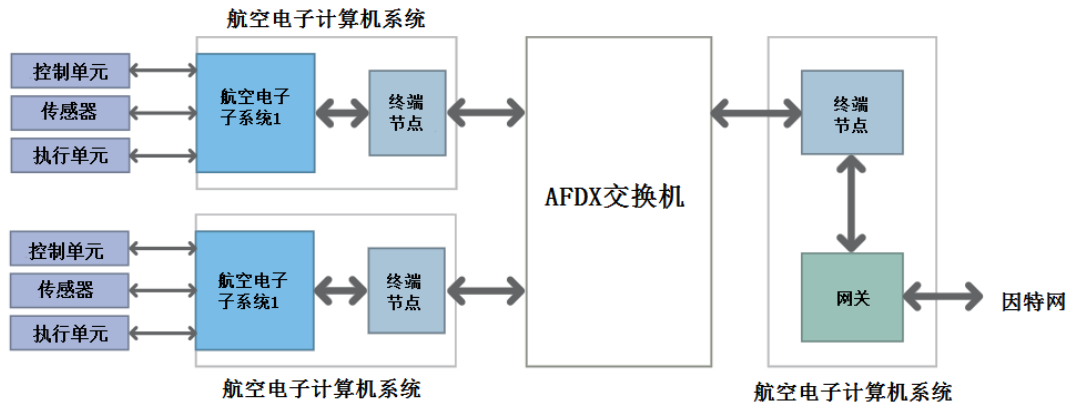


图 2.1 AFDX 网络

航空电子子系统：它与航空电子设备子系统并无区别，如飞行控制计算机，GPS 系统，胎压监测系统等一系列子系统。航空电子计算机系统为航空电子系统提供了一个计算的环境，航空电子计算机系统包含的嵌入式终端节点系统用于实现航空电子子系统与 AFDX 的连接沟通。

AFDX 终端节点系统：在航电子系统与 AFDX 网络互联结构之间提供一个接口，提供高效可靠以及安全的信息传输服务。借助终端所提供的端口，实现各航空电子子系统之间的信息交互。

AFDX 互联结构：一个全双工以太网互联体系，通常包括一组网络交换机，主要功能是转发以太网帧到目的节点，实现终端之间的信息正确路由。

采用 AFDX 实现网络的互联，主要技术特点如下：

- 基于普通以太网：全双工，物理连接介质为双绞线，发送和接收独立，帧格式如图 2.2 所示；

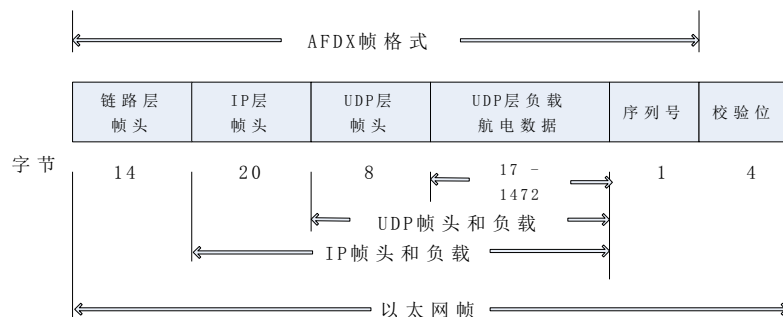


图 2.2 AFDX 网络帧格式

- 全交换式网络：交换机提供无碰撞数据交换，保证点到点传输带宽；
- 确定性的传输：网络通过建立虚链路和带宽分配保证点到点的数据通信延时可确定性；
- 高可靠性：采用双交换机提供可靠的数据链路；
- 传输速率：相比传统网络提供更高的传输速率，可提供 10Mbps 和 100Mbps；

- 通信协议：支持 ICMP，IP，UDP，SNMP，TFTP；
- 上层接口：支持取样端口、队列端口以及服务存取端口；

AFDX 网络物理层符合 IEEE802.3 定义的 10Mb/s/100Mb/s 全双工以太网。链路层增加了虚链路和余度技术，通过虚链路实现网络带宽的控制，通过双冗余技术提高网络可靠性。协议栈主要完成数据收发工作，负责将主机发送的数据进行 UDP/IP 协议处理，通过硬件发送；将硬件接收的数据进行协议解析后，传递给主机。主机应用软件包括数据加卸载软件的网络管理协议。数据加卸载软件采用 ARINC615A 协议，通过 AFDX 通信端口实现数据的加载和卸载操作，采用 AFDX 网络进行传输。将 AFDX 网络协议与开放系统互联的 7 层模型对照，AFDX 网络的协议如图 2.3 所示。

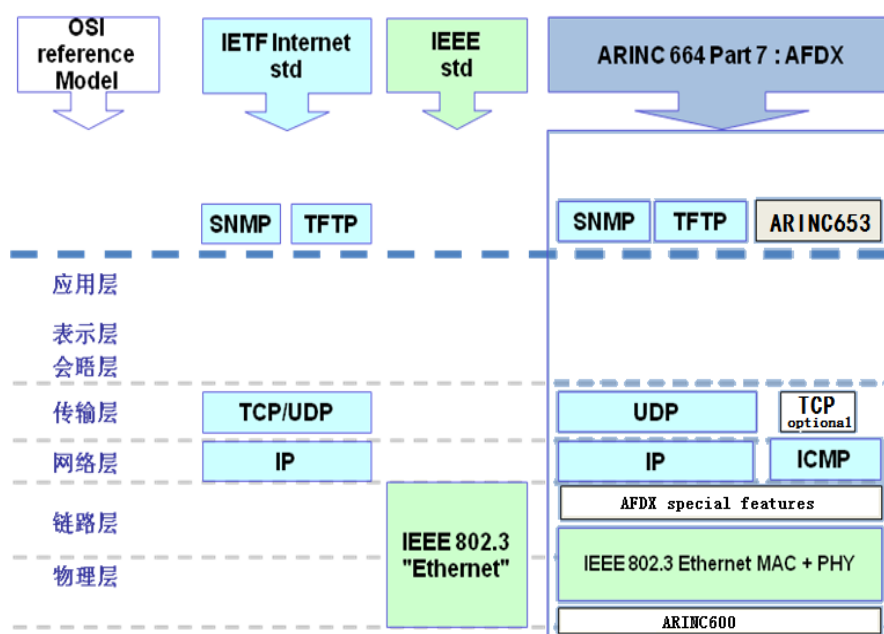


图 2.3 AFDX 协议层次

ARINC664 Part7 中与确定性和可靠性相关的需求主要都集中在对商用以太网的链路层进行改造。网络层以上的协议与商用以太网类似，是作为商用以太网的一个子集设计，AFDX 交换机是二层交换机，实现了物理层和链路层。终端实现则覆盖了从物理层协议到应用层协议。如图 2.4 所示 ARINC429 与 AFDX 的区别，每一个航电子系统，如自动驾驶，惯性导航，显控系统，都是通过两组双绞线对互联，一组负责传输信息，一组负责接收信息，同时还有其它诸多益处^[2]，避免了信息的碰撞。

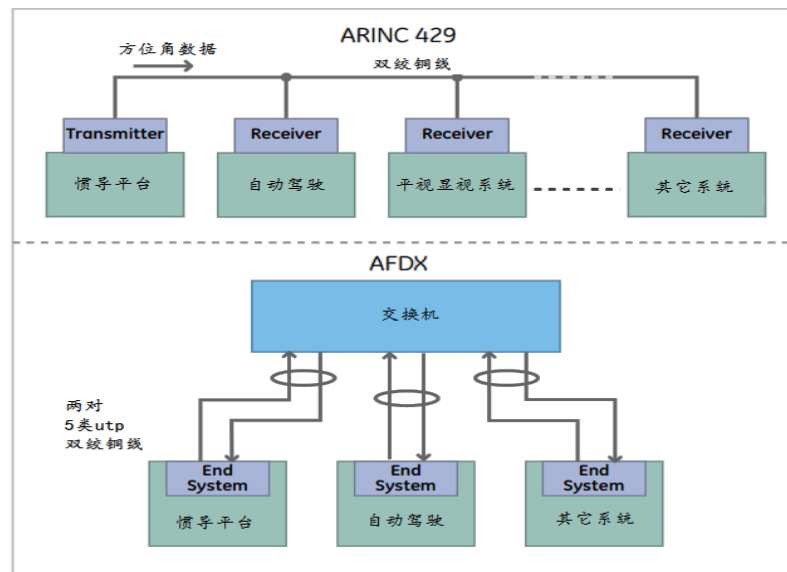


图 2.4 AFDX 与 ARINC429 架构

在 ARINC429 总线中，一根双绞线必须连接每个设备，从而接收由惯导系统所发出的方位角信号。ARINC429 这种单向的点到点，点到多点的属性，意味着航电系统必须为每条信息传送路径布置 ARINC429 总线。因此在一个有着较多终端的系统中，点到点的布线所带来的开销十分巨大，因为在飞机上每增加一公斤的设备，其起飞重量会增加大约 10 公斤。但在 AFDX 中，如图 2.4 所示，每个子系统或者设备只连接一个交换机，通过交换机进行终端的互联，不再需要单独的与惯导系统相联，直观上的一个好处即是极大的减少布线所带来的重量开销。

2.1.1 终端与航电子系统

一般情况下，一个航电计算机可以支持多个航电子系统，使用分区技术实现了计算资源的有效利用，这种分区技术包括两种，一种是时间分区，一种是空间分区。时间分区是配置 CPU 为每个分区所分配的时间间隔。空间分区则是限制每个分区所处的地址空间。其主要目的是使航电子系统中一个分区中出现的错误，不会影响到子系统中其它的分区^[2]，实现分区间的隔离。

AFDX 终端嵌入在每个航空电子系统上，并提供给航空电子系统相互间的安全、高效和可靠的数据传输。终端模型如图 2.5 所示。

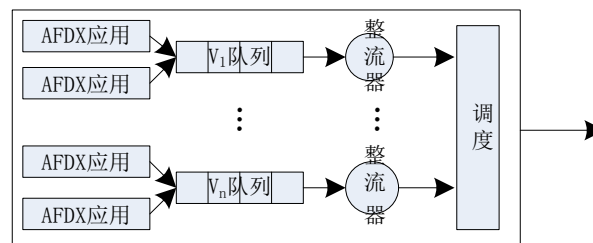


图 2.5 终端模型

对于 AFDX 终端的标识采用最大 16 位的字段来识别, 该字段分为域 ID(Domain ID)、端 ID(Side ID) 和位置 ID(Location ID), 它们又通常组合成网络 ID(Network ID)和设备 ID(Equipment ID)两个部分(16bits 的标识字段是 ARINC664 规定的, 但不是必须的, 重点是要有一个全局唯一的标识字段)。利用 AFDX 终端^[15]主要可以提供的功能有: 系统管理、信息封装、虚链路调节、多路调度、冗余管理和完整性检查。其中系统管理是指对链路的参数进行配置如 BAG 和 L_{\max} , 信息封装是指根据协议层次进行包的封装^[16], 虚链路调度和调度是确定性研究的关注点, 冗余管理是采用双冗余网络保障可靠性, 完整性检查也是采用序列号的方法保障帧的完整性。而面对航空电子网络性能的要求, 终端提出了虚链路的概念, 控制虚链路的发送速率, 使得使用数据计算时延成为可能, 并通过虚链路进行信息的传输, 而航空电子子系统使用通信端口实现它们之间的通信, 通信端口是操作系统应用程序接口(API)的组成部分, 为发送和接收消息提供了编程机制, 在航空电子子系统起着重要作用的有三种端口: 取样(Sampling)端口、队列(Queueing)端口, 以及服务存取(SAP)端口, AFDX 的取样端口与队列端口与 ARINC653 中定义的端口功能一致, 而 AFDX 引进的第三种端口 SAP, 主要是用于非同 AFDX 系统之间的信息交互^[17]。SAP 端口实现了与现有的应用层协议如 TCP/IP 兼容, 借助 SAP 端口, 很多网络中的一般应用, 如文件传输的协议 TFTP 和网络管理的协议 SNMP 都能够接入 AFDX 网络, 发挥作用了。SAP 端口特点:

- SAP 端口可用来访问与 AFDX 网络相兼容的网络;
- SAP 端口最大允许处理 8k 字节长度的数据(其中网络层数据可以进行分片)

其中取样端口则必须提供是否新到帧的标识, 如果没有这个标识, 航电子系统则不可能判断是否停止信息传递或是否需要重复传递。如图 2.6 所示, 对于取样端口特点:

- 与 SAP 端口不同的是取样端口中传输的数据是无法进行分片, 那么由于 IP 报文的最大长度限制, 数据包长度则无法超过 IP 报文的最大负载长度;
- 每个取样端口对应唯一的缓冲区, 且是有固定的长度的, 新进入端口缓存数据对已经在缓冲区数据进行覆盖, 由于取样端口可以保存数据, 所以可以进行多次读取, 即多个应用程序可连续访问;
- 由于在取样队列中读取的信息不会被出队, 则需要标识位进行新旧数据的标识, 并控制它的更新, 取样端口对从它读取数据的应用程序更新相应的标识;

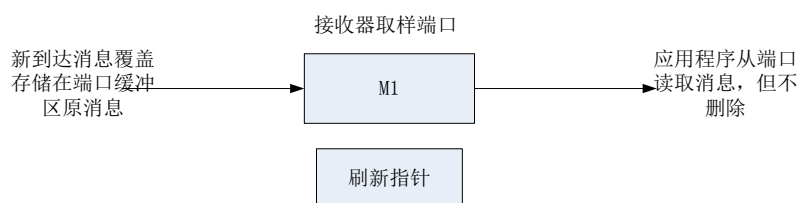


图 2.6 取样端口

队列端口则有足够的存储空间(通过配置参数),新消息可以进入队列,读取队列中信息时,同时会移除该消息。其中,队列端口的特点:

- 通过队列端口传输的数据包的长度可以各不相同;
- 每一个队列都维护一个 FIFO(First In First Out)缓冲区,用于发送或者接收过程中的数据缓存;
- 如果此时队列已满,这时新到的数据则无法入队,采取直接丢弃的做法;
- 队列端口最大允许处理 8k 字节长度的数据(网络层可进行数据的分片);
- 如果对数据进行了分片,则需要在提交上层应用时进行重组,所以当所有的分片都接收到后,队列才会重组并上传给上层应用;

典型的接发送信息的程序接口函数为:

Send_Msg(port_ID,message)

Recv_Msg(port_ID,message)

port_ID 表示端口号, message 指示一个缓冲区是否包含发送的信息或可否接收信息,如图 2.7 所示。

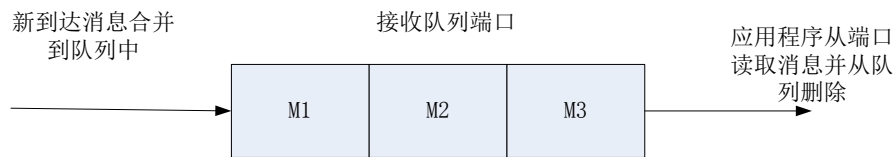


图 2.7 队列端口

2.1.2 虚拟链路

虚拟链路能够通过它进行以太网帧的传输,一条 VL 可以对应多个目的 ES,但源 ES 仅有一个,同时一个 ES 不能既是发送方亦是接收方,从而造成 AFDX 网络形成环路。在 AFDX 网络中,用 16 比特的 VL ID 来标识每一个 VL,通过 VL_ID 来路由 AFDX 帧^{[2][16]},图 2.8 给出了在 AFDX 网络中以太网目标地址的格式。

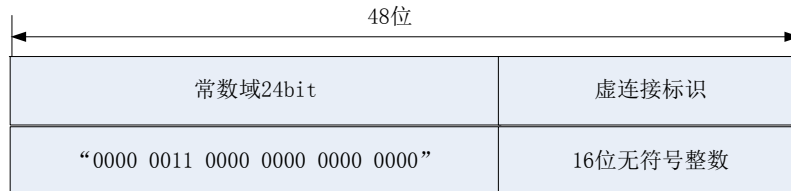


图 2.8 以太网帧的目的地址格式

根据 AFDX 网络中交换机配置,以太网帧最终可以被分配到若干个输出。AFDX 网络中一个重要的特性:路径是静态的,也就是说对应于特定链路标识相关的以太网帧起初就已经配置

好，并且一一对应于各自的终端节点系统。对于预先所设置好的使用同一虚连接标识所传递信息帧，需要对 AFDX 交换机进行配置，图 2.9 显示帧的路由过程。

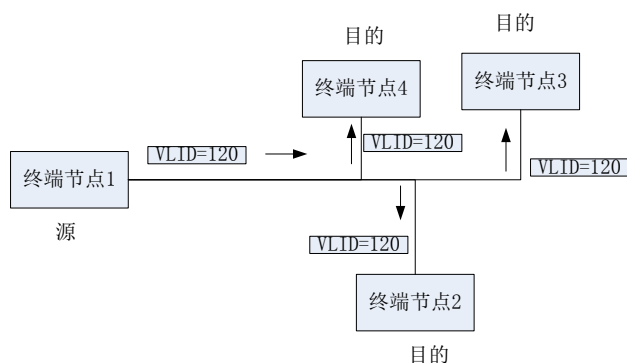


图 2.9 帧路由例子

ES 可以连接多条 VL，所以需要在可用带宽基础上提供链路逻辑上的隔离，系统与 VL 的可用带宽之间没有必然关系，即一个分系统的使用不会影响其它的 VL 带宽。规范指出，一条虚拟链路，至少有管理四条子虚拟链路的能力，但由于在 AFDX 网络中，受到带宽分配间 1ms-128ms 的限制，终端节点可处理虚连接的总数不超过 128 个，所以规范规定一个虚连接可以创建的子连接数是 4 个。四条子虚拟链路的目的终端是相同的，因此这四条子虚拟链路通过一条大的虚拟链路便可以发送至 AFDX 网络。终端需要保证 VL 的带宽，同时还需要保证数据的发送的顺序完整性，一条虚链路不能被多个源分系统共同使用，从而可以在分系统之间提供网络层的隔离。流量控制机制是虚链路也可以提供网络层的隔离，即控制终端不同源 VL 的数据流量的大小。虚链路包含两个重要的参数：

- 带宽分配间隔(BAG)，取值为 2 的倍数，取值范围 1 至 128ms（ARINC664 规范定义的取值范围）
- 最大帧长(L_{\max})，表示虚链路上传输的最长帧

BAG 的意义是指在同一条虚链路发送的两帧之间的最小时间间隔，例如，一条虚拟链路的 BAG 为 4ms，那么在虚拟链路上，数据包的间隔发送时为 4ms，可以超过 4ms 发送，但是绝不能低于 4ms。 L_{\max} 的单位是字节(byte)，最小帧长为 64 字节，最大不超过 1538 字节。同样是当 BAG 设置为 4ms 的虚拟链路，当最大帧长 L_{\max} 为 200 字节时，那么这条虚拟链路的最大带宽就是 $200 \times 8 \times 1000 / 4 = 400,000\text{bps}$ 。

2.1.3 虚拟链路调度与抖动

在端系统存在虚链路调度器，从不同端口发送来的虚链路通过此调度器传输到 AFDX 交换机上。此调度器可以选择不同的调度方式，例如轮循调度(Round-Robin)，最早截止日期优先(EDF)^[20]，固定优先级(Fix Priority)，等等。图 2.10 表示链路调度过程。

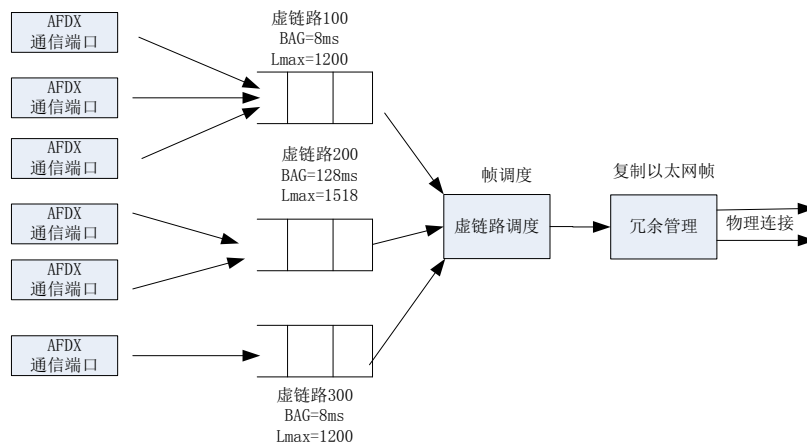


图 2.10 虚链路调度

上图的虚链路调度，作用是对终端系统的数据发送进行虚拟链路的配置，这样可以将终端的虚拟链路确定化，特别在相关通信时间的确定化。VL 调度要保证满足每条 VL 的 BAG 和 L_{\max} 的限制，并且对虚链路切换进行管理控制，从而保证抖动可控以及在确定性要求范围之内。如传感器读数的传输速率为 10Hz，在消息进入虚链路队列时则会引入抖动，类似的，冗余管理中、虚链路切换、物理连接的传输次序都是抖动产生的因素。ARINC-664 规范中规定，在 ES 中输出的 VL 的传输抖动大小要在公式 2.1 约束的范围之内：

$$\begin{cases} \max_{j \in \{VLs\}} jitter \leq 40 \mu s + \frac{\sum (20 + L_{\max j}) * 8}{Nb w} \\ \max_{j \in \{VLs\}} jitter \leq 500 \mu s \end{cases} \quad (2.1)$$

公式 2.1 中的 Nb w 代表是网络带宽，表示以太网帧的抖动大小与链路帧长的关系，500us 为抖动的硬性要求，对链路属性无关。它对所有的 AFDX 网络都适用。

虚链路调度共由两部分组成，一个是数据帧流的调整，另一个是多路切换以及转发。当终端需要转发多 VL 的信息时，调度器采用多路复用技术对经过调整器后的信息流进行处理转发。如图 2.11 和图 2.12 所示。

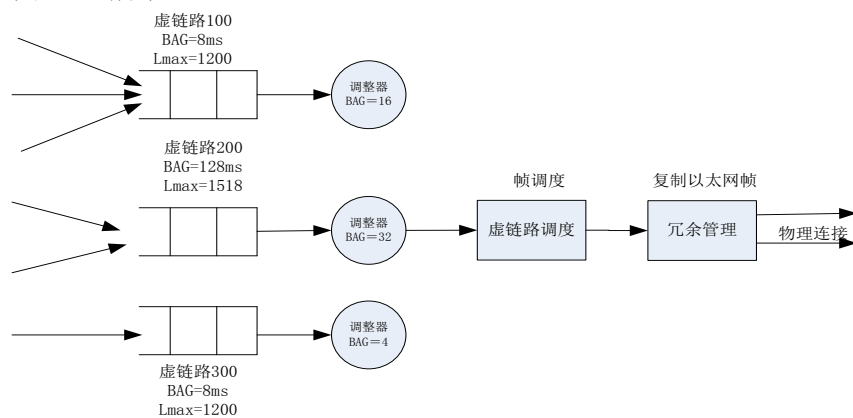


图 2.11 虚链路整流器

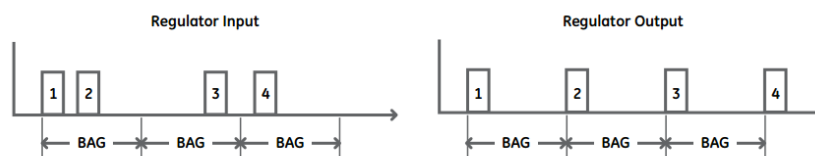


图 2.12 BAG 整流

对于每个虚链路，在调度器的输出端，帧允许在一定的抖动时间界限内出现，当然这个抖动的产生是由于调度器自身的原因而与消息流本身没有关系。作为航电网络确定性分析关键此处即是消息流的整流处理。如图 2.12 所示在每条 VL 上，每个 BAG 时隙要求进行传送的消息流必须不能多于一个帧。能通过每个 VL 的 BAG 和 L_{\max} 可以很容易计算出最大可用带宽为 L_{\max} / BAG 。对于每个 VL，终端中存在终端配置表，表里配置了所有关于此终端链路的 BAG 值。在进行收发的过程当中，终端可以对虚链路的帧长进行调整，为充分利用带宽，可增加到 1518 个字节。

2.1.4 冗余管理

在 AFDX 系统中实际存在两个网络，从终端发出的所有数据包经过冗余管理后都会在 AB 网络中传送，双冗余的设计也是为了安全性^[17]，确保信息的可达。因此在目的终端上会接收到两个内容一样的数据包，如图 2.13 所示。

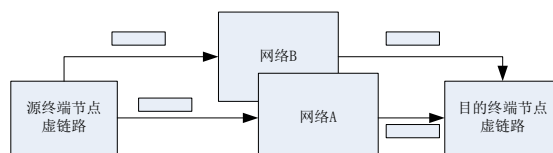


图 2.13 A 与 B 网络

因此端系统需要一种策略来整合从 AB 网络接收到的数据帧，这种策略即称之为冗余管理。在 AFDX 中，所有传输的包都会留有一个 1 字节的序号，这个序号的位置在 FCS 之前。序号的取值范围从 1 到 255，当增加到 255 之后则变为 1，而序号 0 则是代表了端系统的重启。序号的存在使到达目的端系统的同一条虚拟链路内数据包的先后顺序区别出来^[18]，目的端系统的冗余管理系统根据同一虚拟链路上序号来判断是否存在丢包，错误包出现的状况。以太网帧接收过程，如图 2.14 所示。

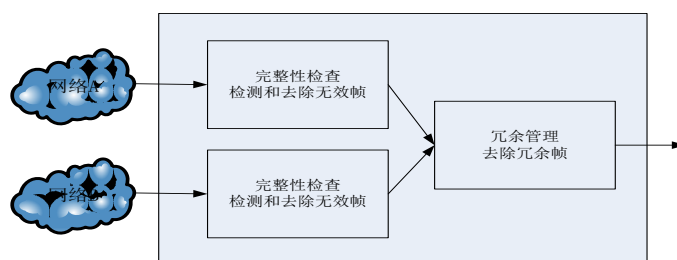


图 2.14 以太网帧接收过程

但是如果若 B 网络上的冗余帧先于 A 网络上的后续帧即达, 则 B 网络上的冗余帧则不会发挥作用^[19]。如图 2.15 所示。SkewMax 参数表示 AFDX 网络中终端系统接收一帧的两个复本之间的最大时间间隔。

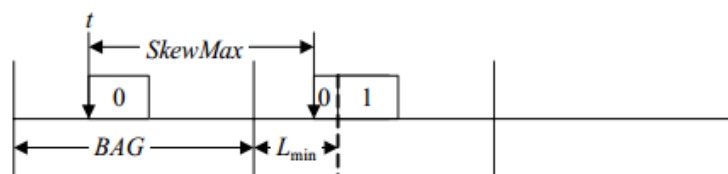


图 2.15 冗余帧到达时序图

如图 2.15 所示, t 时刻作为链路上 0 号数据帧的到达时刻, 紧接着的 1 号帧的到达时刻则为 $BAG + L_{\min}$, 而在另一条链路中, 冗余的 0 号数据帧的到达时刻为 $t + SkewMax$, 这样为使冗余管理合理有效, 帧按顺序到达, 则须保证: $t + SkewMax < BAG + L_{\min}$ 即 $SkewMax < BAG + L_{\min} - t$ 。在这种情况下, 冗余管理只须一个缓冲区即可。由于实际中 $SkewMax$ 难以控制, 应用中, 可以适当放宽要求, 适当增加缓冲区。

2.1.5 传输协议栈

以太网帧封装了 IP 报文, IP 报文则封装了 UDP 报文, UDP 报文包含了要传送给航电子系统的数据。当 UDP 层负载的数据小于 18byte 时, 协议会自动补全至 18byte, 当数据长度大于或等于 18byte 时, 则不需要添加任何字节。而 IP 层报文则可以利用 IP 报头进行分片操作, 将超长的 UDP 信息分割传输。IP 报头中设计了可以识别分片归属的字段, 使得分片可以在终端重组。IP 目的地址中有 VL 的目的地址以及标识 ID 号。UDP 报头中包含消息源以及目的地的 UDP 端口号, 借助以上的地址信息, 使用 AFDX 网络能够准确的将信息从源传送至目的地。端系统的对于帧的收发, 即也可以看作是数据帧的层层封装以及解封装^{[2][21]}。如图 2.16 所示。

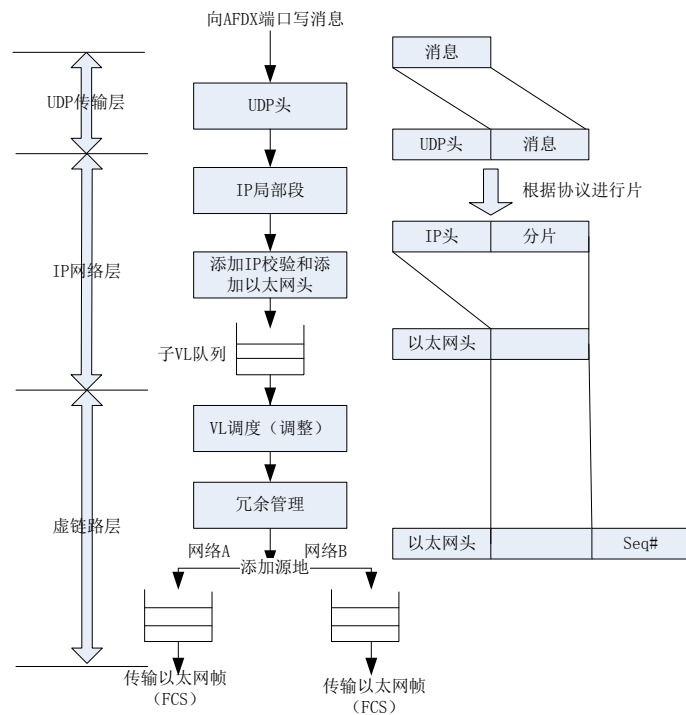


图 2.16 AFDX 发送协议栈

接收过程帧的封装是相对于发送过程的逆过程，如图 2.17 所示：

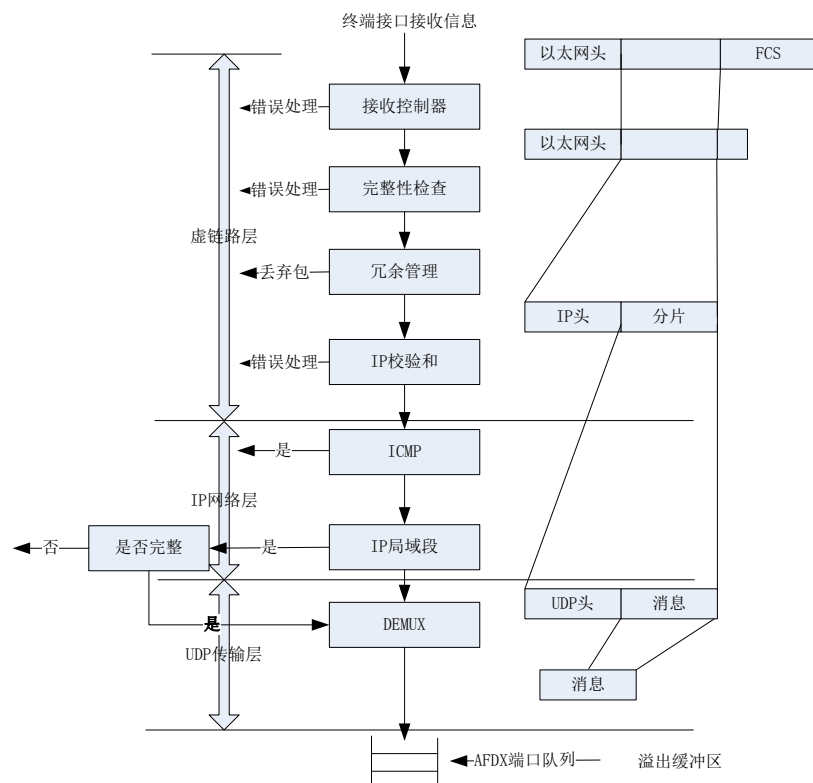


图 2.17 AFDX 接收协议栈

2.2 民用航空网络性能要求与建模

2.2.1 AFDX 网络性能参数

在常见的网络系统中，关于网络性能的参数，主要有：时延，业务流经过系统所经历的最大可能时间间隔；数据积压，指系统在某时刻等待处理服务的数据比特数；有效带宽，是指在确定给定到达曲线以及时延值的情况下所需的最小线路速率；等效容量，与有效带宽类似，但考虑了缓冲区的大小；时延抖动，是指业务流中任意两个数据分组所经历的时延的最大差值，即数据分组所经历的最大时延和最小时延之间的差值。

在飞机网络中，没有必要区分几种分类或流量种类之间的不同，尽管必须保证数据类型和最大传输延迟，但是每一个网络传输请求必须被服务。因此，飞机网络中的服务类型是有保证的服务。需要满足确保性能服务，则需要提供端到端的性能保证，同时能够提供一个确定的、数学可证明的、有上界的端到端帧的传输延迟等网络性能参数，对于确定性的服务中性能参数，最重要的就是：

- 端到端有界延迟保证。保证数据在规定的时延界限内必须传递至终端的评价指标。
- 端到端的时延抖动保证。同一个数据流中任何两个帧之间的端到端传输延迟变化是不固定的，因为它取决于给定时间内整个网络流量。然而，延迟抖动的范围是可计算的。
- 缓冲区大小必须能够保证数据包不会由于阻塞而丢弃。需要分析网络的流量特性，特别是网络性能的积压参数，对缓冲区进行合理分配。

2.2.2 AFDX 网络时延模型

为了分析在 AFDX 网络中，数据传输的过程和确保服务的关键性能参数：时延，对 AFDX 中网络的时延建立模型。AFDX 的多端点系统中，端到端链路开销会很大，以太网可以为确定性网络提供一种有效的方法来解决这个问题。当然仅通过硬件延迟计算是不能确定通过某个节点的延迟时间的，还应包括一个变量：抖动，它是由于网络上数据传输存在竞争而产生的。根据累积延迟和虚拟链路的传输带宽来对网络进行分析是通用的解决方案^{[2][8]}。以太网时间分配可以参见图 2.18 所示。

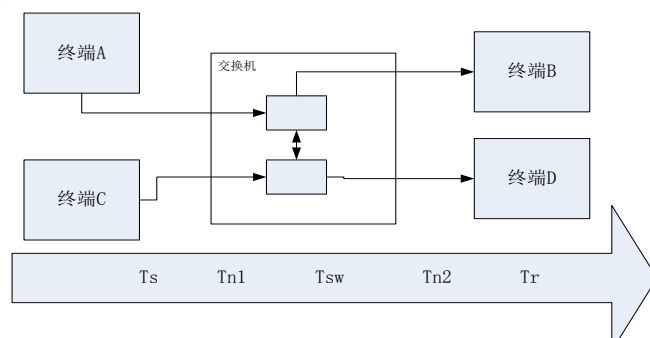


图 2.18 AFDX 拓扑

其中 T_s 表示在源端系统中的延迟, 可通过公式 2.2 计算:

$$T_s = \tau_{tech} + J_{max} + \tau_{frame} \quad (2.2)$$

τ_{tech} 表示发送端技术延时, 是在缓冲存储器为空且没有冲突和 IP 分片的前提下计算的, 表现在硬件上是有界延迟, 可以取值为 150us, J_{max} 表示发送端多路复用延时, 直接与多路利用策略相关, τ_{frame} 表示帧的延时, 取最大值即 1538B/(100Mb/s)=123.01us, 取 124us。Tn1, Tn2 是在链路中的传输延时, 对延迟结果影响很小, 完全可以忽略。Tr 为主要是指在目的端数据帧处理技术延时, 可取 150us。

而 T_{sw} 为在交换机的延时, 是需要重点分析的部分, 利用有界延时建模: 正如 ARINC664 标准所要求的, 滤波运行和帧中继不能被交换机的其它功能所干扰, 因此设计的交换机模型中, 只考虑与帧处理有关的处理过程。更具体地说, 即完成了帧的整体接收, 帧的完整性验证, 帧的有效性验证, 虚链路流量滤波, 切换台所有权, 重定向正确的输出端口, 最早的帧传输。设计的交换机模型要反映这一运行序列, 因此, 其总体结构相对简单, 根据空客的标准来定义, 设计相应的交换机模型, 考虑到交换机缓冲区的延时取决于它接收的最大帧, 因此应该根据其间的流量建立模型, 所有接收缓冲区接收到的最大的以太网帧, 即 1538 字节。在技术条件的一项要求规定交换机的每个端口都有在 1ms 内过滤 125 帧。因此可以使用有延时边界的元件来模拟这些过程, 延时值设置为 8us。如图 2.19 为帧的处理过程。



图 2.19 交换机中帧处理过程

在输入端中, 存储转发延迟 τ_{sf} 为 1538B/(100Mb/s)=123.01us, 取 124us。数据过滤时, 本模型采用的规范相比 A380 中交换机规范更为严格, 它要求 1s 内可以过滤 148800 帧, 也就是说数据过滤的延迟 τ_{fl} , 可以通过 $1/148800 \times 10^6 = 6.72us$, 当然仍然取 8us。接下来, 转换到一个或多个正确的输出端口采用相同的逻辑, 速度可以根据所考虑的交换机或快或慢, 由航空公司规定, 当然即使当所有端口的端到端都接收到最小的帧, 交换机也应该能够在 $N \times 125ms$ 内切换, 其中 N 是端口的数量。所以 τ_{fw} 同样取 8ms。而在多路转换器中的转换延时, 则是与调度算法相关, 是需要研究的可变延时 J_{max} 。如图 2.20 所示为交换机的一般模型。

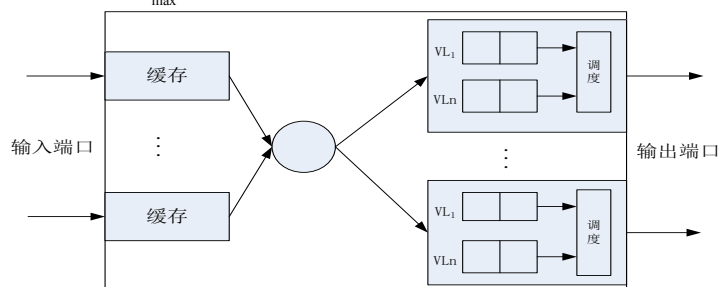


图 2.20 交换机模型

T_{sw} 可以通过公式 2.3 来计算:

$$T_{sw} = \tau_{sf} + \tau_{fl} + \tau_{fw} + J_{\max} \quad (2.3)$$

最后, 正确输出端口上帧的传输可以通过 FIFO 多路转换器元件建模, 其中输出能力设计为物理链路的输出能力, 即 100Mb/s。

多路转换器的功能是将所有输入的数据流传输到唯一的输出端。Cruz 假设一个多路转换器可以在“直通转发”模式下运行, D 表示多路转换器内持续的特定延时比特值, 用 $Q(t)$ 表示在时间 t 出现的比特量。Cruz 针对三类多路转换器, 分别是先进先出、按流量先进先出或通用条款。第一类中, 数据包按照先到先发的原则。接收协议是一旦多路转换器收到第一个比特值, 则表示收到数据包了。只要还有工作要完成, 多路转换器就绝不会处于非工作状态, 即如果 $Q(t) > 0$, 则瞬时流量输出等于输出链路的最大容量。

除通用多路转换器外, 其它两类策略可能会有休止期, 如可以对轮询机制或其它可在一定时间内防止发送的因素建模。根据定义, 只有当 $Q(t)=0$ 和输出链路的瞬时速率为零时, 休止期才开始。在休止期间, 多路转换器不传输任何数据包, 而且到达的所有数据都被置于队列中。这些休止期是有界的, 我们用 V 表示其边界。通用多路转换器可以随机转发接收到的数据包, 对非休止期的时间损失没有进一步的约束。按流量先进先出多路转换器会确保按照到达顺序转发属于同一流量的数据包, 但对流量之间的数据不做任何担保。实践中, 如果休止期的最大持续时间为零时, 则即是 FIFO (先进先出) 多路转换器。

以下是对多路转换器性能研究的描述。用不同的公式表述了只有两个数据流的情况, 可以同时扩展到更复杂的 n 个数据流的情况。两个输入数据流量包络分别用 R_1 和 R_2 表示, 数据包的大小是确定的, 并标注为 L 。输入链路速率用 C_1 和 C_2 表示, 输出链路速率用 C_{out} 表示, 对于非保守的多路转换器, 休止期的最大值用 V 表示。最后我们采取以下稳定性假设:

$$\lim_{t \rightarrow \infty} (R_1(t) + R_2(t) - C_{out} \cdot t) = -\infty \quad (2.4)$$

上面的充分条件说明在经过较长的一段时间后, 多路转换器最终总是会清除所有积压, 因此系统不会变得不稳定。在通用多路转换器中, 任何流量每比特的延时边界为:

$$\overline{D} = \sup \{ \alpha : \alpha \geq 0, R_1(t) + R_2(t) + C_{out} \cdot V \geq C_{out} \cdot \alpha \} \quad (2.5)$$

对于通用多路转换器, 流量 1 输出端具有以下包络:

$$R_{1,j+1}(t) = \min \{ C_{out} \cdot t, \max_{\Delta \geq 0} [R_1(t + \Delta + V) + \min \{ R_2(\Delta + V), C_{out} \cdot \Delta \} - C_{out} \cdot \Delta] \} \quad (2.6)$$

对于通用多路转换器, 即按流量先进先出多路转换器, 与上述原因相同, 数据流 1 的队列积压边界为:

$$\overline{Q}_1 = \max_{\Delta \geq 0} [R_1(\Delta + V) + \min \{ R_2(\Delta + V), C_{out} \cdot \Delta \} - C_{out} \cdot \Delta] \quad (2.7)$$

而 AFDX 多路复用延时,提供了两种在实际中应用广泛的策略,Cruz 已经证明了以下结果,第一种是先进行出复用排队延时,数据流 1 的延时边界为:

$$D_{FIFO,1} = \frac{1}{C_{out}} \max_{u \geq 0} [R_1(u) + R_2(u + (L/C_2)) - C_{out} \cdot u] \quad (2.8)$$

而同时针对数据流 1 的输出端的流量包络,可以为:

$$R_{1,j+1}(t) = \min \left\{ C_{out} \cdot t, \max_{\substack{D \geq 0 \\ \Delta \geq 0}} \left[\min \left\{ R_1(t+D), R_1(t+D+\Delta) + R_2\left(\Delta + \frac{L}{C_2}\right) - C_{out} \cdot (D+\Delta) \right\} \right] \right\} \quad (2.9)$$

在推广至 n 个流时,进一步的,可以得到延时边界的计算公式:

$$D_{FIFO,i} = \frac{1}{C} \max_{u \geq 0} \left[\sum_{i=1}^n \left(\sigma_i(u) + \frac{S_{max,i}}{C} \right) - C \cdot u \right] \quad (2.10)$$

其中 C 表示物理带宽, S_{max} 为线路最大数据帧长度; $\sigma(t)$ 是指 AFDX 数据流符合的流量特性,符合漏桶模型(Leaky Bucket)。

第二种是指静态(SP)优先级的复用排队延时分析:

$$D_{i_max, SP(i)} = T_{SP(i)} + \frac{\sum_{j \in SP(i)} \sigma_j}{R_{SP(i)}}, \quad (2.11)$$

$$R_{SP(i)} = C - \sum_{j \in SP_i} \rho_j, T_{SP(i)} = \frac{\sum_{j \in SP_i} \sigma_j + \max_{j \in [SP_{i+1}, SP_n]} \{l_j^{max}\}}{R_{SP(i)}}$$

式中, $\sum_{j \in SP_i} \sigma_j$ 是所有比当前虚链路优先级高的虚链路突发度之和, $\sum_{j \in SP_i} \rho_j$ 是指所有比当前虚链路优先级高的虚链路速度之和, $\max_{j \in [SP_{i+1}, SP_n]} \{l_j^{max}\}$ 是指所有比当前虚链路优先级低的虚链路最大帧的数据长度。

在 AFDX 网络中,发送终端系统假设是不同步的,但是帧传输遵从有限到达分配。有限到达意味着精确有限带宽通信量控制。因此,通过网络的每个连接的延迟分配遵从有限分配原则。所以,网络中的最大延迟可以通过计算所得,而不是估计的。通过使用精确有限带宽通信量控制机制完全有可能限制带宽使用,当然在极限的时间间隙中。因此,限制端系统的时延是可能的。虚链路可以通过增加帧间隔或减小帧长度来使用较少的带宽。然而,通过限制每一个虚拟链的特性,最终我们可以分析整个网络真实性能。

AFDX 网络数据传输的可靠性与网络的实际设计有关,AFDX 网络数据传输的可靠性指标是根据相关的安全性分析得来的,没有一成不变的数值。如民用飞机系统设计和安全性分析的两个工业标准(其余 DO 标准和 FAA 的相关标准类似),包括:ARP 4754A 与 ARP 4761,其中 ARP4761 中定义的安全性分析过程包括 FHA(Functional Hazard Assessment),PSSA(Preliminary System Safety Assessment)和 SSA(System Safety Assessment)。ARP4761 中所描述的安全性分析方法有包括了故障树分析(FTA),关联图法(DD),马尔可夫分析法(MA)^[22]等相关的方法。

2.3 常用 AFDX 网络确定性分析方法

目前被广泛研究的可以网络确定性性能分析方法有三种, 包括模型检测(Module Check), 轨迹法(Trajectory)和网络演算(Network Calculus)^[23-25]。本节分析了前两种方法在航空电子网络中应用的不足并指出网络仿真以及确定性性能分析算法在航空电子网络设计中的作用。

2.3.1 模型检测

Module Check 是一种基于时间自动机的方法, 这种方法的本质是遍历系统的所有可能状态, 并以此来计算最坏的系统时延。时间自动机^[26]最早提出是为了描述系统随时间的变化。它是在有限状态机中加入了时钟的概念, 系统内的各个正的实变量随着时间的递增而递增, 用 Module Check 方法来解决 AFDX 时延的计算问题^[27]。

Module check 基于对系统状态的遍历, 它可以准确地计算出最坏时延和最坏时延对应的场景, 但是它的缺点在于对于复杂的系统, 系统内各个组件的状态规模庞大, 遍历所有的组合是不现实的, 所以模型检测针对实际应用的系统并不适用。

2.3.2 轨迹法

轨迹(Trajectory)法是分析一条数据流路径最坏场景下的相关性能参数的方法, 它要遍历分析数据流路径上所有经过的节点。它适用于各种网络的时延分析, 也可以用来分析 AFDX 网络^[28]。下面给出了使用轨迹法对 AFDX 网络时延进行求解的一个例子^{[28][29]}, 但没有在航电中大规模使用, 是因为它不考虑整个网络的节点对数据流的影响, 只构建数据包在每个节点中的时序, 主观性强, 且获得的结果不可证明性, 无法作为适航认证的依据, 当然在某些应用场景下, 轨迹法的计算结果会优于网络演算的计算结果, 所以在非要求认证的网络中如何整合这两个方法以获得更优的结果, 也是研究的一个重点。

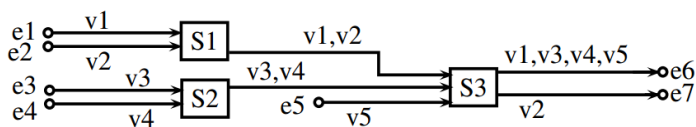


图 2.21 AFDX 配置

所有的 VL 上最大帧长为 400bytes, BAG 都取等于 4ms。网络带宽 100Mb/s。

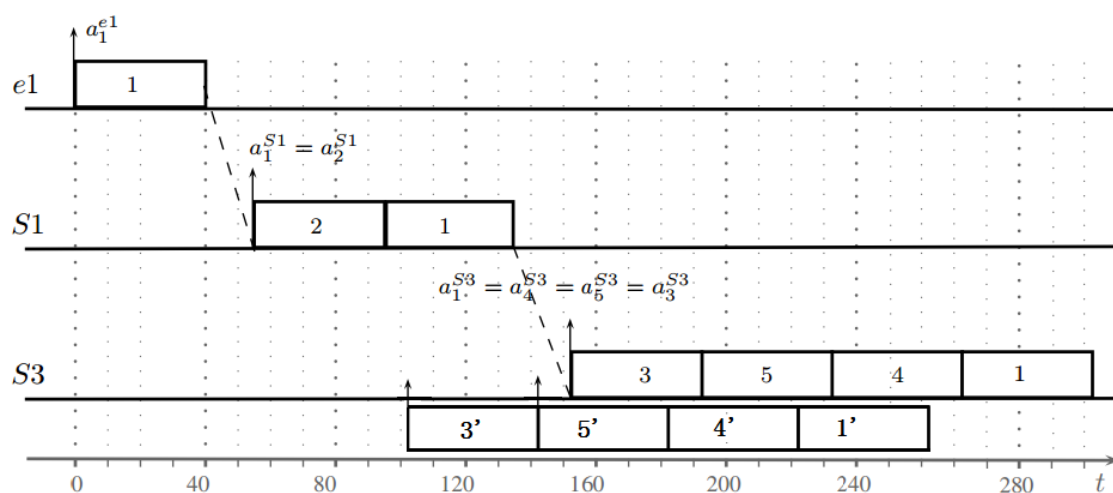


图 2.22 VL1 调度

上图通过轨迹法计算了 VL1 在最差情况下的延迟，第一种计算中，由于 VL3 与 VL4 到达 S3 是同一条输出链路，即 VL3 与 VL4 上的帧不可能同时到达，所以这种场景不可能出现。并且这两帧之间至少相差 40us，即转发时间，第二种计算中，考虑了 40us 的转发延迟。

轨迹法和网络演算的比较，其中 NC_Ser 和 Traj_Ser 指的是分别利用了组技术，从而考虑了序列化后的方法。

表 2.1 工业配置下端到端延迟的总结

| | Average gain | Best gain |
|--------------------|--------------|-----------|
| NC_Ser VS NC | 24.21% | 51.07% |
| Traj VS NC | 16.69% | 37.83% |
| Traj VS NC_Ser | -10.45% | 10.37% |
| Traj_Ser VS Traj | 16.60% | 63.38% |
| Traj_Ser VS NC | 30.89% | 64.89% |
| Traj_Ser VS NC_Ser | 9.01% | 39.83% |

如 NC_Ser 与 NC 相比，则考虑串行化效应的网络演算与基本网络演算，上界平均有 24.21% 增益（上限相比减小了 24.21%）。而针对某些链路，则最大的增益有 51.07%。

对于上述使用的轨迹法，在使用了序列化技术之后，对于考虑串行化效应的轨迹方法，我们无法证明这是最坏情况下的一种场景，同时考虑串行化效应的轨迹方法，仍然是一个悬而未决的问题，其并不能证明是上界，旨在确定这些不利的情况下，假设是准确的最坏情况。当然在某些实时性要求不是很强的应用下，可以使用。或者使用轨迹法与 NC 理论集合。表 2.2 是网络演算与考虑串行化效应的轨迹方法在平均，最大，最小三个维度的比较。

表 2.2 工业配置下端到端延迟的比较

| Benefit | Trajectory/WCNC | Best/WCNC |
|---------|-----------------|-----------|
| Mean | 10.46% | 10.71% |
| Maximum | 34.02% | 34.02% |
| Minimum | -10.39% | 0% |

2.4 本章小结

本章主要介绍了主要介绍了 AFDX 相关的基础知识，包括端系统、虚拟链路、调度以及冗余管理和航空网络的性能以及针对 AFDX 网络设计的时延模型。并针对三种交换机策略作了简单介绍，对时延、积压、输出流包络的 Cruz 证明结果进行说明。最后对网络确定性性能参数时延的分析方法：模型检测和轨迹法。分别指出两种方法在航空电子网络领域的不足，即模型检测状态空间的爆炸性增长和轨迹法不可证明。另外对轨迹法与网络演算结果进行了对比，分析不同链路时延差距，为以后改善链路配置指方向。

第三章 基于多链路分组技术的网络演算

3.1 网络演算

Cruz 提出的网络演算^[23]网络分析方法的主要特点是能够提供确定的结果^{[24][25]}。输入网络的数据流可以通过包络的概念进行建模, 由该方法得出的边界采用的是绝对值, 且对流的包络考虑不是特定的, 是给定幅值内的所有时间间隔, 不同的边界是通过考虑所有不同幅值的极值得出的, 这样就避免了研究最坏的情况, 因为最坏情况可以用确定性边界的研究方法得出。事实上, 最坏情况的研究主要是针对特定时间间隔内的事件。可以通过赋予给定事件某些特性来将所考虑的时间间隔特定化, 例如当某个元件数据流第一个数据包到达时, 这种方式必然引起一个问题, 就是我们是否真的考虑了最坏情况发生的时间间隔, 最坏情况的发生总是分为两个阶段: 首先是从直觉上考虑最坏情况的场景, 然后证明所考虑的情况就是最坏的情况。然而对最坏情况的证明经常非常困难, 经验表明, 对最坏情况的直觉很少是有道理的, 尤其是在 AFDX 网络复杂的情况下。Cruz 方法考虑了所有的可能时间间隔, 能够利用简单的证明提供确切的边界, 这点对于航空电子系统的认证尤其重要, 并可以证明计算得出的所有边界。

我们在第二章对网络时延进行了建模并分析, 分析的总体性能将取决于建模方法的有效性和质量。对于符合 ARINC 664 标准的网络, 标准必须对网络的静态特性进行了定义, 且在航空器执行任务的所有阶段, 包括地面和飞行期间, 单一模型都是有效的, 因此有可能用单一模型来分析网络。而在互联网中, 很难应用网络演算的原因无法建立网络的整体模型, 主要由于网络中设备具有很大的异质性造成对网络特性的不确定。然而航空电子网络的情况则不同, 实际中, 对每个不同的系统都进行明确的定义, 每个设备供应商都对其硬件负责, 飞机制造商对航空器所有设备的集成负责。因此, 每个设备供应商都可以提供一个硬件模型, 而因为制造商对硬件有充分的了解, 所以提供的模型一定是非常精确的。正如 2.2.2 节所介绍的, 同样地, 飞机制造商能够提供不同设备之间互连的精确模型, 因为设备间的互连是由飞机制造商定义并实现的。由此可见, 根据 ARINC 664 标准对航空电子网络建模不仅是可能的, 而且模型的精确度会很高。

3.1.1 基本概念

网络演算理论是基于最小加代数(Min-Plus Algebra)和最大加代数(Max-plus Algebra)^[30]的网络性能分析工具。在该代数中用最小值替换加法, 而用加法替换乘法, 网络演算主要包括了最小加卷积(Min-plus Convolution)与最小加反卷积(Min-plus De-convolution), 以及到达曲线与服务曲线。其中到达曲线是指对流的进入过程的限制, 另一个服务曲线是对网络节点的服务能力进

行描述，从而对流的输入、输出行为进行限制和描述。为了使得网络演算可以很好地分析评估数据流在网络中传输的性能，到达曲线与服务曲线则必须具备一定的网络性能分析理念的基本特性^[31-33]。

网络演算主要是使用广义递增函数以及相关的操作和相关概念。

定义 3.1 广义递增函数。对于 $\forall s, t \in (R \cup \{+\infty\})$ ，且 $s \leq t$ ，若有 $f(s) \leq f(t)$ 成立，则称 f 为广义递增函数。

定义 3.2 广义递增函数集合。若：

$$F = \{f(t) \mid f(t) = 0, \forall t < 0; f(0) \geq 0, \forall u \leq t, f(u) \leq f(t), t \in [0, +\infty)\}$$

则称 F 为广义递增函数集合。

定义 3.3

$$[\bullet]^+ : [x]^+ = \max(x, 0)$$

$$[\bullet] : \forall x \in R, [x] \in N, [x] - 1 < x \leq [x]$$

定义 3.4 最小运算

$$(f \oplus g)(t) = \min[f(t), g(t)], f, g \in F. \quad (3.1)$$

定义 3.5 次可加函数

$$\forall s, t \geq 0, f(t+s) \leq f(t) + f(s), f \in F$$

由于网络演算是针对 DEDS 理论中计数器进行研究，同时次可加性作为计数器的一个基本性质，重要性非常在，次可加函数同时具有另外两个性质：两个次可加函数相加仍然是次可加函数；次可加函数的极小卷积仍然是次可加函数。

定义 3.6 极小卷积

$$(f \otimes g)(t) = \inf_{0 \leq u \leq t} [f(u) + g(t-u)], f, g \in F \quad (3.2)$$

定义 3.7 反卷积运算

$$(f \oslash g)(t) = \sup_{u \geq 0} [f(t+u) - g(u)], f, g \in F \quad (3.3)$$

值得一提的是定义 7 反卷积不是卷积的逆运算，通常 $f \otimes g \oslash g$ 不一定等于 f 。在 $t=0$ 时

$(f \oslash g)(0) = \sup_{t \geq 0} \{f(t) - g(t)\}$ ，极小反卷积可以被看成是对双子 $\{\varphi, \oplus, \otimes\}$ 中函数之间差异程序的一种度量。

定义 3.8 水平偏差 $h(\bullet)$ 和垂直偏差 $v(\bullet)$

$$h(f, f', t) = \inf\{\tau \geq 0 \mid f'(t+\tau) \geq f(t)\}, t \geq 0$$

$$h(f, f') = \sup_{t \geq 0} h(f, f', t) \quad (3.4)$$

$$v(f, f') = \sup_{t \geq 0} \{f(t) - f'(t)\} \quad (3.5)$$

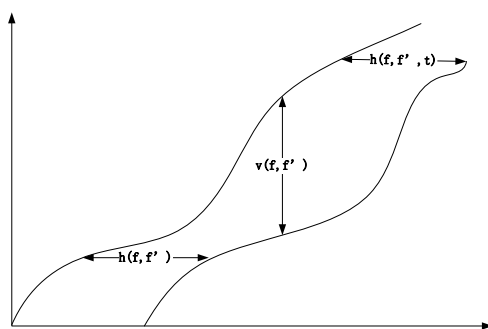


图 3.1 水平偏差与垂直偏差

定义 3.9 积压与延时

在一个系统中，设 R 为其输入， R' 为输出，则在时间 t 时的积压是：

$$b(t) = R(t) - R'(t)$$

显然， $b(t) \leq v(R, R')$ 。所以对于流 R 经过系统 S ，积压：

$$b(R, S) = \max_{R \xrightarrow{S} R'} v(R, R')$$

虚延时 $d(t)$ 表示当一个比特进入系统直到出系统的等待时间，定义为

$d(t) = h(R, R', t)$ ，显然 $d(t) \leq h(R, R')$ 。所以对于流 R 进入系统 S ，延时为：

$$d(R, S) = \max_{R \xrightarrow{S} R'} h(R, R')$$

定义 3.10 积压时刻

服务器 S ， R 为输入， R' 为输出。时间间隔 I 为产生积压时间区间，可知

$\forall x \in I : R(x) > R'(x)$ 。设 t 为积压过程的某一时刻，则产生积压的开始时刻定义为：

$$StBl(R, R', t) = \sup\{u \leq t \mid R'(u) = R(u)\}$$

定义 3.11 到达曲线

对于 $R \in F$ ， $\alpha \in H$ ， H 表示定义在 R_+ 至 R 上的函数，如果对于所有的 $s \leq t$ ，有：

$$R(t) - R(s) \leq \alpha(t - s)$$

则称流 R 有到达曲线 α ，这条件等同 $R \leq R \otimes \alpha$ ，特别地，当 $R(t) = \sigma + \rho t$ 时，称 R 受限 (σ, ρ) 。

定义 3.12 服务曲线

服务曲线是约束一个系统输入与输出关系的工具。设 R 为系统输入流量累积函数， R' 为系统输出流量累积函数。则我们有定义^[34]：

简单服务曲线 $\beta \in H_0$ ， $H_0 \in H$ 且 $f(0) = 0$ ：

$$R' \geq R \otimes \beta$$

$$S_{simple}(\beta) = \{(A, B) \in F \times F \mid A \geq B \geq A \otimes \beta\}$$

弱严格服务曲线 $\beta \in H_0$ ：

$$R'(t) - R(StBl(t)) \geq \beta(t - StBl(t))$$

严格服务曲线 $\beta \in H_0$:

$$R'(t) - R'(s) \geq \beta(t - s)$$

定理 3.1 输出流到达曲线

假设到达曲线为 α 限制的流, 该数据流通过系统 S , 并为其提供服务曲线为 β , S 输出流到达曲线是由 $\alpha^* = \alpha \otimes \beta$ 限制。

定理 3.2 对经过串接网络节点(2 个)的业务流, 定义网络节点的服务曲线为 S_k 其中 $k=1,2$, 则其复合服务曲线可以通过计算 $S_1 \otimes S_2$ 并作为该业务的最大服务曲线。

定理 3.3 端到端的服务曲线 $S_{e2e} = S_1 \otimes S_2 \otimes S_3 \otimes \dots \otimes S_h$, 其中第 $h(h=1,2,\dots)$ 个网络节点的服务曲线为 S_k 。

定理 3.4 当受限于函数 α 的相关业务流通过网络中的节点时, 其中该节点提供服务曲线 β , 其虚延迟的上界为 $d(t) \leq h(\alpha, \beta)$, 其中 $h(\alpha, \beta) = \sup_{u \geq 0} [\inf\{T : T \geq 0 \text{ 且 } \alpha(u) \leq \beta(u + T)\}]$ 也被称作曲线与曲线之间的水平方向上的偏差。

定理 3.5 受限于 α 的业务流经服务曲线为 β 的网络节点时, 可计算积压上界 $R(t) - R^*(t)$ 满足为 $R(t) - R^*(t) \leq \sup_{u \geq 0} [\alpha(u) - \beta(u)]$ 。

3.1.2 统计网络演算

下面介绍的是统计网络演算技术相关基础知识^[35-36]。

定义 3.13 (统计流量包络): 使用累积函数 $R(t)$ 定义通信流的流量特性, 若在对应的时间 $[t, t + \tau]$ 上的累积到的流量 $R(t, t + \tau)$ 适用于以下关系:

$$Pr\{R(t, t + \tau) \leq \mathcal{G}(\tau)\} \geq 1 - \varepsilon$$

则 $\mathcal{G}(\tau)$ 表示此流量过程的统计型流量包络, ε 表示最大违背概率。

定义 3.14 (有效服务曲线): 使用累积函数 $R(t)$ 定义通信流的流量特性, 那么当通信流的输出函数 $D(t)$ 满足以下关系:

$$Pr\{D(t) \geq R \otimes \beta(t)\} \geq 1 - \varepsilon \quad (3.6)$$

则称该 $\beta(t)$ 为通信流 $R(t)$ 提供的有效服务曲线。

定义 3.15 (有效服务曲线): 如果通信流依次通过有效服务曲线 $\xi_1(t)$, $\xi_2(t)$, \dots , $\xi_N(t)$ 的网络节点 1, 节点 2, 节点 3, \dots , 节点 N 时, 则这些若干个节点串联后能够提供的端到端有效服务曲线为:

$$\xi = \xi_1 \otimes \xi_2 \otimes \dots \otimes \xi_N$$

满足:

$$Pr\{D(t) \geq \inf_{s \in [0, N(T + \xi)]} \{R(t - s) + \xi(s)\}\} \geq 1 - \varepsilon \quad (3.7)$$

式中：N 为网络节点个数；R(t)为通信流的输入函数；D(t)为通信流的输出函数。

定义 3.16（统计时延上界）：假设一个通信流过一个网络系统，它的到达曲线 $\alpha(t)$ ，并且系统为该通信流提供的有效服务曲线为 $\xi(t)$ ，对 $t > 0$ 的任何时间，则经过此系统的通信流延迟低于 $d^{net} = \inf\{d \geq 0 : \alpha(t-d) \leq \beta(t), \forall t \geq 0\}$ 的概率为 $1 - \varepsilon$ ， $P[W(t) \leq d^{net}] \geq 1 - \varepsilon$

3.2 网络计算方式

3.2.1 VL 简单流模型

网络演算考虑的是数据流，即一系列数据包的集合，大小可能是可变的。利用增函数给每个数据流定义特性，增函数可以用来表示在一定的时间段积累的网络流量。通过水平与垂直两个方向的计算，可获得时延，积压上界。确切地说，我们用 R 表示累积流量函数，那么 $R(t_1, t_2)$ 表示 t_1 和 t_2 时间段内该点数据流传输的信息总量，同时标注为 $R(t) = R(0, t)$ 。即函数 R(t) 是正值并且是增长的，突发流量的特点是在低幅值的时间间隔内 R 值急剧增大。由到达曲线的定义可知，到达曲线是流量的一个包络。

可以看出某些网络元件输入和输出包络之间的关系。一个有限时限为 D 的元件意味着输入的所有信息停留的时间不超过 D， R_i 是该类元件中数据流的输入包络。Cruz 证明函数^[24] R_{i+1} ：

$$R_{i+1}(t) = R_i(t + D) \quad (3.8)$$

R_{i+1} 是输出流的包络。这一结果很重要，因为它与元件中延时边界的确定方法一起构成了整个网络分析方法的基础。如图 3.2 所示。

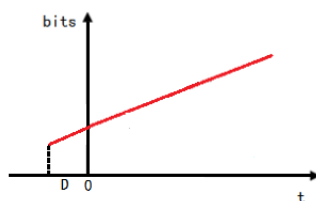


图 3.2 到达曲线位移

3.2.2 只为突发数据付费一次演绎

通过定理 3.2 与定理 3.3 可知，可以方便的定义数据流路径上两个连续元件的累积服务曲线。如图 3.3 示例。其中数据流 1 满足漏桶模型 (σ, ρ) ，服务节点 1, 2 提供服务曲线 C_1, C_2 。

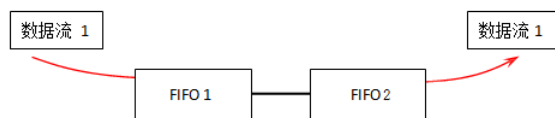


图 3.3 组示例

则 $R(t) = \sigma + \rho t$ ，则在 FIFO1 中的延迟通过定义 3.9，可得 $D_1 = \frac{\sigma}{C_1}$ ，因为数据流 1 经过

FIFO1 后的输出曲线通过定义， $R_1(t) = \rho(t + D_1) + \sigma$ ，同理经过 FIFO2 后，相应的延迟为

$D_2 = \frac{\sigma + \rho D_1}{C_2}$ ，且经过 FIFO2 后的输出曲线为 $R_2(t) = \rho(t + D_1 + D_2) + \sigma$ 。因此，最终数

据的总延迟为 $D = D_1 + D_2$

针对数据流 1，根据定理 3.2，定理 3.3 可以计算 FIFO1，FIFO2 的为数据流 1 提供的累积服务曲线，这里给出计算过程：

$$\beta = \beta_1 \otimes \beta_2 = \inf_{0 \leq u \leq t} [C_1 u + C_2(t - u)] = \inf_{0 \leq u \leq t} [(C_1 - C_2)u + C_2 t]$$

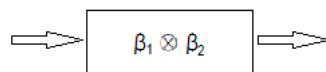


图 3.4 累积服务曲线

所以 $\beta = \min(C_1, C_2)t$ ，通过计算可知聚合流经过累积服务后的延迟为：

$$D_{12} = \frac{\sigma_1}{\min(C_1, C_2)} \quad (3.9)$$

可以通过计算注意到该系统采用串联服务曲线计算得出的延时边界值优于前两者之和。这可以解释了“只为突发数据付费一次(pay bursts only once)”的现象。

如图 3.5 所示，可以通过图形化显示相应的结果。

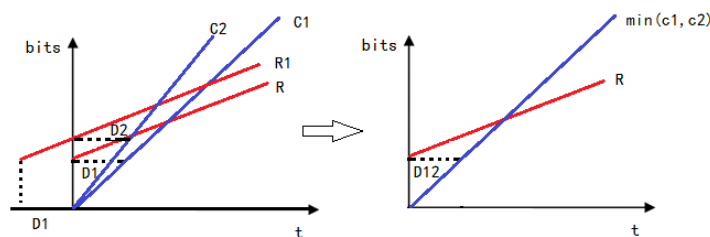


图 3.5 串联服务曲线

3.2.3 组与组的包络

常见的对网络演算的批评即其计算结果的悲观性，该方法计算出延迟结果是延迟的上界，并不是准确的最差情况下的延迟。然而事实上，网络演算结果所计算出的结果并不是那么悲观，在某些场合完全可以指导应用实践，大约有 16.5%^[37]。

在基本的网络演算过程中，已经在前一个节点汇聚的流在发送端口还是作为单个的流量计算，进行到达曲线的累加。到达曲线的值可以表示单个数据流可以瞬间发送的最大比特量，则所有数据流的总和即累加每个数据流量值的总和来确定串联数据流可以发送的最大比特值边界。这种计算规则是基于一个不太准确的假想，即所有数据流都可以“同时”发送一个帧。但在实际应用中，某些流是无法同时传输，尤其是那些一起通过同一物理链路的数据流：对这些流而言，帧是必须被序列化，帧的到达不可能是同时的。

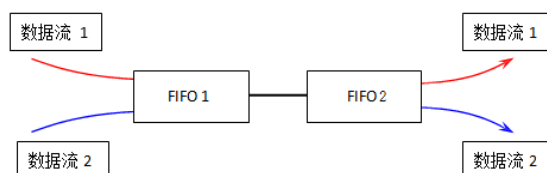


图 3.6 聚合流示例

如图 3.6 所示，假设有两个先进先出队列，由容量为 C 的单一物理链路连接。我们在首先放置两个数据流，标记为 (σ_i, ρ_i) 。其中 $\sigma_1 = \sigma_2 = b, \rho_1 = r1, \rho_2 = r2$ 。

通过网络演算我们得出两个队列，D1 和 D2 的延时：

$$D_1 = \frac{\sigma_1 + \sigma_2}{C} ; \quad D_2 = \frac{\sigma_1 + \sigma_2 + D_1(\rho_1 + \rho_2)}{C}。$$

在目前的情况下，一方面由于两个元件的输出速率是 C ，比特位不应在第二队列延迟，另一方面比特位已经在交换机 1 排序，所以数据流 2 不会给数据流 1 造成阻塞延迟， $D_2=0$ ，即数据流 1 与数据流 2 组成了数据流组。考虑序列化的效果可使我们得出更小的边界值，因为需要准确地描述由一个组产生的流量，同时定义它的到达曲线。

首先，用 $R_g(t)$ 表示数据流组产生的流量包络， $R_i(t)$ 表示数据流 i 产生的数据流量，我们可以直接得出： $R_g(t) \leq \sum_i R_i(t)$ 。显然在一段时间 t 内，数据流组发送的流量不能多于各个数据流所发送的流量总和。另一方面，任何单独的数据流的传播速度都没有多路转换器（用 C 表示）的输出速率快，因为后者代表的是与多路转换器连接的物理链路的最大速率。数据流 1 与数据流 2 在第一个交换机输出端口已经汇聚成一条流量，有了集合到达曲线，即在数据流共享物理链路，相应地数据流组也不能超过这个限值，所以同时在包络方面我们得出： $R_g(t) \leq Ct$ 。组的包络满足这两个条件，因此在任意点都应符合这两个条件的最低要求，其可被重写为：

$$\forall t, R_g(t) \leq \min\left\{\sum_i R_i(t), Ct\right\}$$

图 3.7 说明了数据流 1, 2 的合成过程:

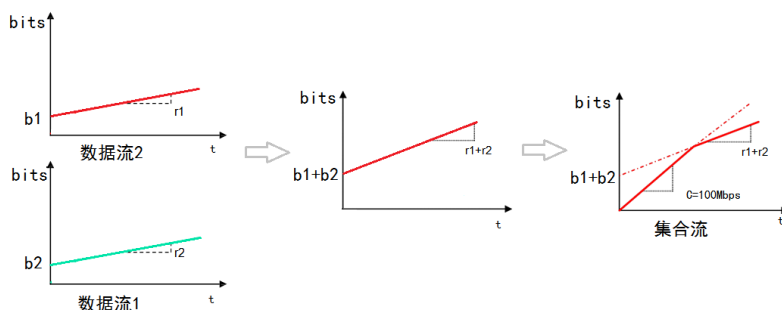


图 3.7 数据流组的到达曲线

通过使用上文定义的组的到达曲线, 发现汇聚后的集合到达曲线的包络要更好, 应用网络演算获得的延时上界更紧致。且带来的另一个好处, 是针对 FIFO 可立即确认得出先进先出队列 2 零延迟的结果。即考虑并保持了数据流 1 和数据流 2 在交换机 1 中序列化的因素。

对于 FIFO 提供 C 的速率, 可认为服务节点最大的服务能力, 令 $\lambda(t) = Ct$, 则相应的整合数据流的输出包络为:

$$R^*(t) = (\lambda(t) \otimes \sum_i R_i(t)) \odot \lambda(t) \quad (3.10)$$

3.3 多链路分组技术的分析

3.3.1 精确流分析

通过上面的分析可知, 通过组的概念可以得到更小的包络, 所以计算的结果会更加精确, 组的策略保持的越久, 则对于我们计算带来的好处则就越大。但是, 当最终构成组的数据流相互分开, 那么组的合并对于各分支流带的特性还没有理论分析的结果, 因此本文中, 是同时是使用组和单个数据流进行分析, 在组分离时找到每个流的包络。对于多链路分组技术, 是针对于进入交换机同一个输出缓冲区的流可能是来自不同的物理链路^[38-40]。可以将这些链路分为两类, 一类是序列化的链路, 一类是非序列化的链路, 序列化中链路的聚合流计算满足公式 3.11, 非序列化的链路满足原始的漏桶模型。注意此处针对某个流的输出曲线, 同时也是下一个节点的输入曲线, 集合公式 3.8 的证明结果, 输出流曲线可以通过公式 3.11 计算 (如图 3.2 所示):

$$a^*(t) = \alpha(t + h(\alpha, \beta)) \quad (3.11)$$

即偏移后的链路突发量变大了, 考虑在 AFDX 网络中, 对于虚链路有 BAG 的约束, 所以可以进一步精确化到达曲线。通过研究流的到达规则, 考虑下面帧到达场景:

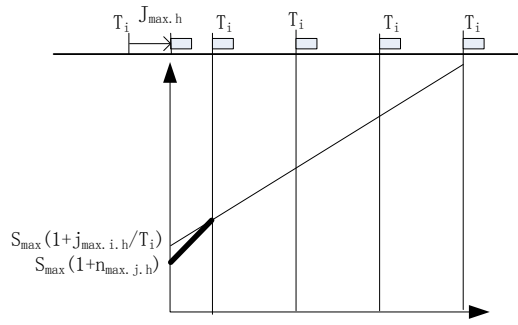


图 3.8 精确到达曲线

如图 3.2 所示，其反映的物理意义是只要有 VL 被阻塞，相应的 VL 的突发度都会增大，而如图 3.8 所示，其物理意义是，由于数据包即使在被阻塞后也是离散到达的，突发度不会随被阻塞的时间连续地增大。其主要是因为交换机存在流量管制，流量并不会无限制增加。流量管制的策略。定义了交换机 h 针对虚链路 i 的流量帐户 $AC_{i,h}$ ，在初始帐户设置为 $AC_{\max,i,h}$ ，且定义如公式 3.12 所示：

$$AC_{\max,i,h} = S_{\max,i} (1 + J_{\max,i,h} / BAG_i) \quad (3.12)$$

且流量帐户是随着时间的推进，按照 $S_{\max,i} / BAG_i$ 的比例增长，但不能超过初始的帐户值。交换的流量定制，可以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，表示如果流量帐户值大于到达的帧长 S 时，则接收此帧，并更新流量帐户为 $AC_{\max,i,h} - S$ 。而基于帧的管理，当如果流量帐户值大于到达的帧长 S 时，则接收此帧，同时更新流量帐户为 $AC_{\max,i,h} - S_{\max,i}$ ， $S_{\max,i}$ 表示该虚链路中最大的帧长。将交换机中的最大抖动描述为：

$$J_{\max,i,h} = n_{\max,j,h} T_i + \Delta_{J_{\max,i,h}} \quad (3.13)$$

所以当 $D_{\max,i,h} > J_{\max,i,h}$ 时，表明该流量的抖动将优先会受到交换机流量管制的限制，则可以利用上面的精确流模型和其定义的到达曲线计算相应的结果^[38]。在如图 3.9 所示，考虑流量管制后的到达曲线计算包络更小，因此计算的延时上界 $h(\alpha', \beta)$ 更小。

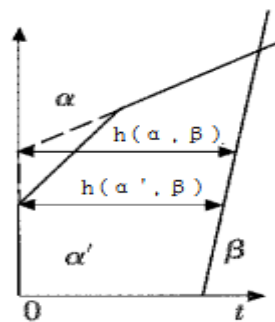


图 3.9 受流量管制的到达曲线

再从虚拟链路的属性帧间带宽间隔考虑，图 3.10 则更精确分析了帧的到达过程。

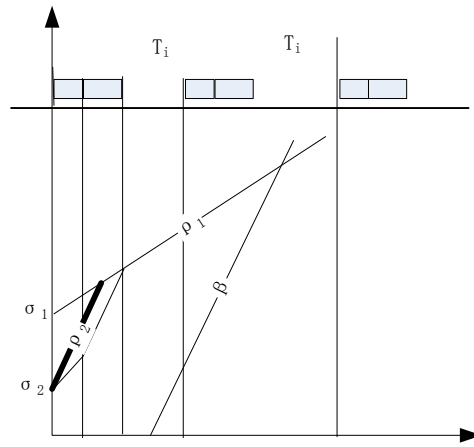


图 3.10 离散数据流

如图 3.10 所示，不管到达曲线的突发量是多少，在链路上进行传输时，都要服从 AFDX 的 BAG 规则，考虑过输出流的到达曲线，进行链路分配时的不确定性，所以规定其初始突发流量为链路中最大帧长 S_{\max} ，以速率 $\rho = S_{\max} / \min(BAG)$ 进行流量的增加，直到满足 Cruz 证明的输出包络。所以设该服务曲线为 $\beta_{R,T}^{FIFO}(t) = R[t - T]^+$ ，通过计算延迟可得：

$$D = \min \left\{ \frac{\sigma_2}{\beta} + T, \frac{\sigma_1}{\beta} + T, \frac{(\sigma_1 - \sigma_2)\rho_1 + \sigma_1}{\rho_2 - \rho_1} + T \right\} \quad (3.14)$$

通过实验表明，特别在针对优先级调度时且对虚链路带宽较低的情况下，上面的紧致方法的增益为 0.6%-2%。如图 3.11 所示。

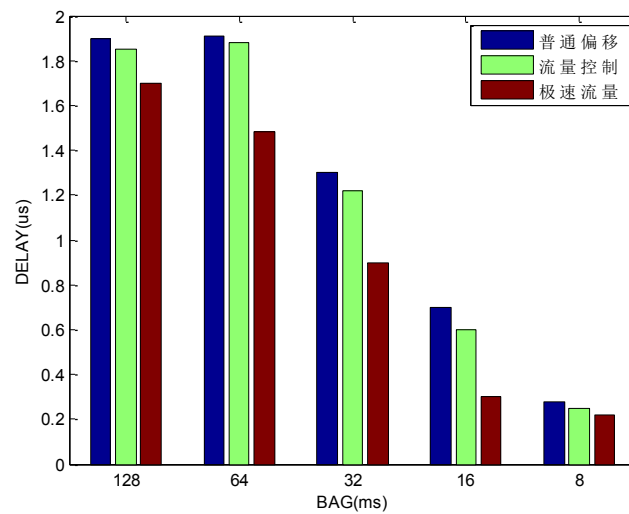


图 3.11 利用各精确流的比较

在以上的流模型下，除了有得到更紧致延时上界。在如图 3.12 的情况下，在极端情况下，对于积压理论上也有增益。

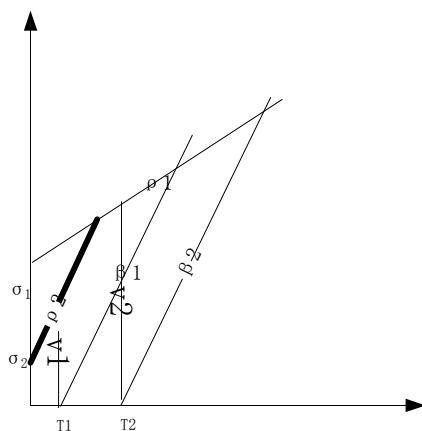


图 3.12 不同服务曲线下积压

如图 3.12 所示，第一条服务曲线 $\beta_1(t) = R[t - T_1]^+$ ，第二条服务曲线 $\beta_2(t) = R[t - T_2]^+$ ，拐点满足

$$T_1 < \frac{(\sigma_1 - \sigma_2)}{\rho_2 - \rho_1} \text{ 时，此时积压 V1 上界为 } \sigma_2 + \rho_2 T_1$$

$$T_2 > \frac{(\sigma_1 - \sigma_2)}{\rho_2 - \rho_1} \text{ 时，此时积压 V2 上界为 } \sigma_1 + \rho_1 T_2$$

而 $V_2 - V_1 > 0$ ，即表明此集合流在第一条服务曲线上在积压计算上带来了增益，为 $\sigma_1 + \rho_1 T_2 - \sigma_2 - \rho_2 T_1$ 。较基本网络演算的结果上界，降低了

$$1 - \frac{\sigma_2 + \rho_2 T_1}{\sigma_1 + \rho_1 T_2}$$

在考虑第一种服务曲线时，通过 MATLAB 设计得出突发量和速率对积压的影响及增益，如图 3.13 所示，表示为在 ρ_2 为 100Mb/s 下，不同突发度下，积压的变化。

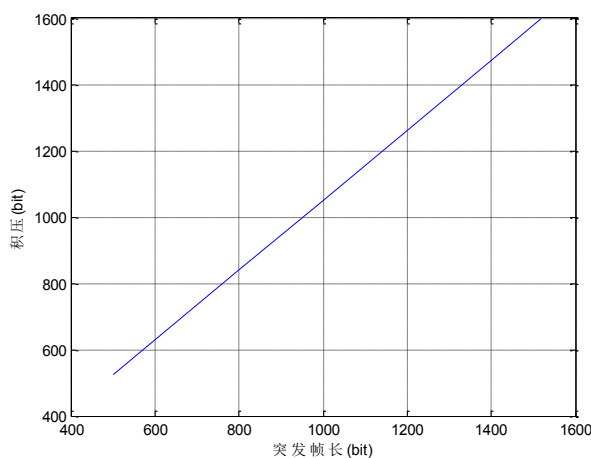


图 3.13 100Mb/s 下积压

如图 3.14 所示, 在突发度为 500bit 下, 积压随着速率的变化结果。

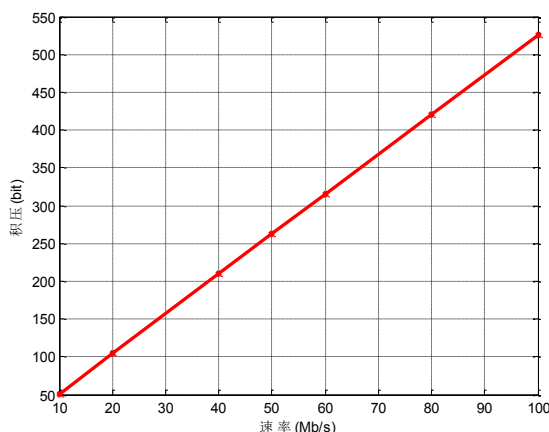


图 3.14 500it 突发下积压

3.3.2 多链路分组

多链路分组技术, 则是将链路分为两种状态, 序列化与非序列化。设 L_s 为序列化链路集合, L_{ns} 为非序列化链路集合。对于一条物理链路上虚链路分组的延迟上界计算, 假设节点 k 上输出缓冲区 P 上存在 n 个流, 每个流都适用于漏桶模型 (σ, ρ) , 并且对应的到达曲线为 $a_i^{k(p)} = \sigma_i^{k(p)} + \rho_i^{k(p)} \cdot t$, 这 n 个流可以聚合成一个流, 其聚合到达曲线为:

$$a^{*k(p)}(t) = \sum_{\tau_i \in (I)} a_i^{k(p)}(t) = \sum_{\tau_i \in (I)} \sigma_i^{k(p)} + \sum_{\tau_i \in (I)} \rho_i^{k(p)} \cdot t = \sigma^{*k(p)} + \rho^{*k(p)} \cdot t \quad (3.15)$$

其中 $\sigma^{*k(p)}$ 最大突发流量, $\rho^{*k(p)}$ 是聚合流的平均最大速率。此为各流曲线的简单数据叠加。

然而, 共享同一条物理链路上的不同流是无法同一时间到达缓冲区, 这些聚合流还要受链路物理带宽的限制, 此即流的序列化过程, 此处进行序列化。所以共享链路 l 上针对节点 k 中的缓冲区 p 的分组聚合流不仅仅简单流的叠加。而是

$$a_{(l)}^{k(p)}(t) = \min \left\{ \sum_{\tau_i \in (I)} \sigma_i^{k(p)} + \sum_{\tau_i \in (I)} \rho_i^{k(p)} \cdot t, \max_{\tau_i \in (I)} \left\{ \sigma_i^{k(p)} \right\} + C \cdot t \right\} \quad (3.16)$$

其中 $\max_{\tau_i \in (I)} \{\sigma_i^{k(p)}\}$ 表示共享链路 l 上的各虚链路中突发流量最大值, 且拐点的计算表达式为:

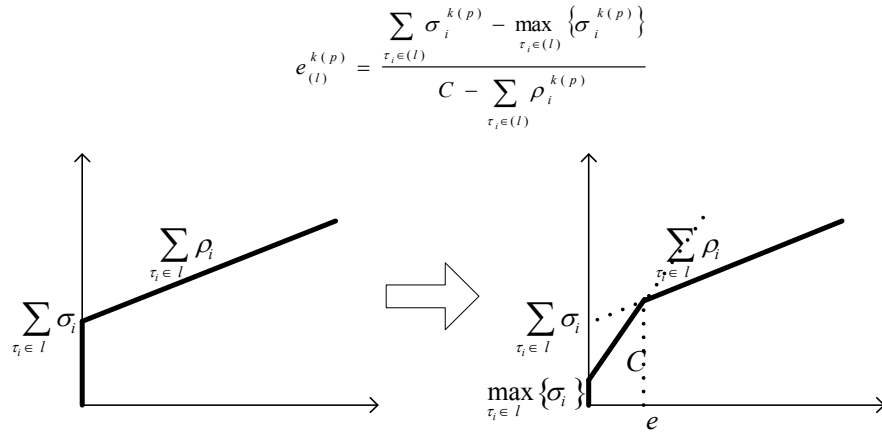


图 3.15 单链路分组的到达曲线

而对于多个分组且聚合进入相同的输入缓冲区，其中 $a_{SL(m)}^{k(p)}$ 即为单链路分组下的到达曲线，对每个单链路分组到达曲线，按拐点升序排序，如图 3.16 所示。

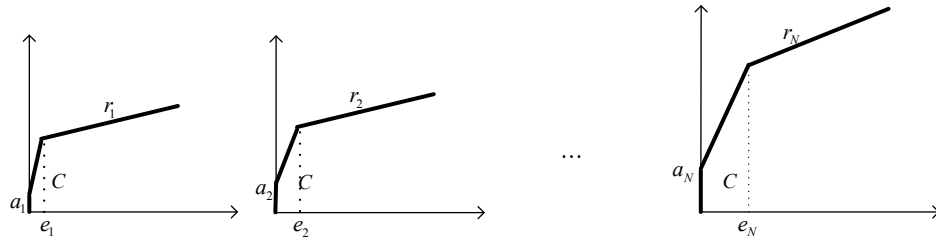


图 3.16 单链路分组叠加

经过聚合后的到达曲线如图 3.16 所示，可以计算相应的延迟。

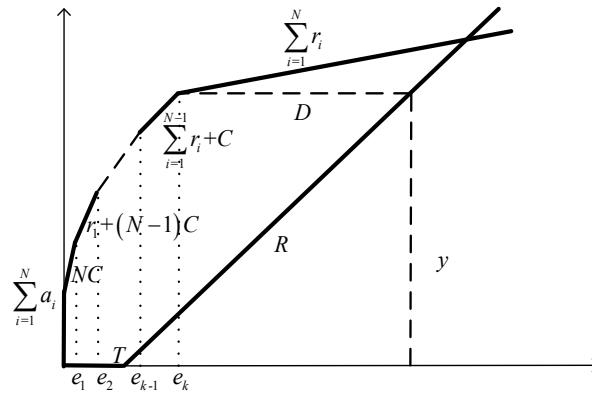


图 3.17 多链路分组

针对静态优先级的服务曲线 $\beta_{R,T}(t)$ ，其通过公式 3.17 计算得：

$$\beta^{SP_i}(t) = \left[Ct - \sum_{\tau_j \in SP_i} \alpha_j(t) - \max_{\tau_j \in [SP_{i+1}, SP_n]} \{l_j^{\max}\} \right]^+ \quad (3.17)$$

其中, SP_i 代表优先级 i , l_j^{\max} 代表流 τ_j 上的最大帐长, 每个满足漏桶模型流的延迟, 计算服务曲线:

$$\beta^{SP_i}(t) = R[t - T]^+, \quad R = C - \sum_{\tau_j \in SP_1} \rho_j, \quad T = \frac{\sum_{\tau_j \in SP_1} \sigma_j + \max_{\tau_j \in [SP_{i+1}, SP_n]} \{l_j^{\max}\}}{R} \quad (3.18)$$

网络演算可以计算对应延迟, 其在节点 k 的聚合流 i 经过服务曲线 β 最大延迟:

$$D_{i_max, BUF(i)}^{(k)} = T^{(k)} + \frac{y}{R^{(k)}} - e_N \quad (3.19)$$

, 其中

$$y = \sum_{j=1}^N a_j + \sum_{s=1}^N \left\{ \left[\sum_{i=0}^{s-1} r_i + (N - s + 1) \cdot C \right] \cdot (e_s - e_{s-1}) \right\}$$

$$r_0 = e_0 = 0$$

而非序列化 $a_{NSL(m)}^{k(p)}$ 到达曲线聚合则为简单叠加。再与上文序列化流的到达曲线叠加^[39-44], 即可得到多链路分组下的流的聚合曲线。叠加过程如图 3.18 所示。

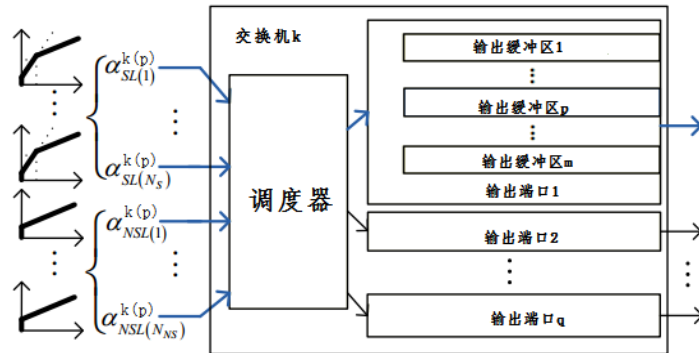


图 3.18 多链路叠加示意

最终序列化流与非序列化流叠加结果如公式 3.20 所示:

$$a^{k(p)}(t) = \sum_{m=1}^{L_s} a_{SL(m)}^{k(p)}(t) + \sum_{m=1}^{L_{ns}} a_{NSL(m)}^{k(p)}(t) \quad (3.20)$$

L_s 表示序列化链路, L_{ns} 表示非序列化链路, $a_{SL(m)}$ 表示序列化流到达曲线, $a_{NSL(m)}$ 表示非序列化流的到达曲线。计算出组的延迟后, 利用 3.3.1 节精确流分析, 可求出各虚链路过此交换机后的输出曲线, 同时也是下一个节点 (交换机) 的到达曲线。

3.4 算法设计实现

3.4.1 AFDX 的网络建模

对于交换式以太网 AFDX，网络的输入和输出称为端系统（ES）。所有终端系统之间通过交换机。所有终端系统是 FIFO，交换机支持静态优先级（SP）排队调度策略。网络中的所有链接都是全双工，因此没有碰撞的发生。

图 3.19 显示一个假定的 AFDX 网络拓扑结构，其中包含 13 个 ES, 6 个交换机和 14 的 VL，其中 V5 是多播链路。

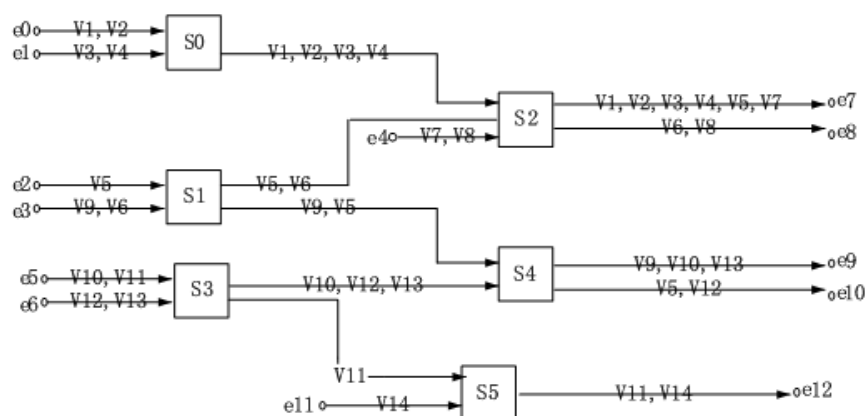


图 3.19 网络拓扑

对 AFDX 网络中虚链路上流的延迟进行抽象分成以下几个部分

- 1) 最大的固定延迟的部分我们称之为技术延迟 L_{tech_max} 。这包括交换机中检查、过滤以及转换的时延，这部分可以作为固定的时延；
- 2) 最大等待时间 D_{i_wait} ，由于多个流共享同一个输出缓冲区，从而造成的等待时间；
- 3) 最大传输时间， $D_{i_trans} = S_{i_max} / C$ ， C 为链路带宽

在 AFDX 网络中，每个 VL 在端系统都符合漏桶模型。满足

$$\sigma_i = \sigma_i^{(0)} = S_{i_max}, \quad \rho_i = \rho_i^{(0)} = \frac{S_{i_max}}{BAG_i}$$

最坏情况下的延迟且端口采用 FIFO 的输出，则

$$\sigma_i^{(k)} = \sigma_i^{(k-1)} + \rho_i^{(k-1)} \cdot D_{i_wait}^{(k-1)}, \quad \rho_i^{(k)} = \rho_i^{(k-1)}$$

最坏情况下的延迟且端口采用 SP 的输出，则可由公式 4，公式 5 计算相关参数。

端到端延迟最坏情况下的延迟可以通过简单地累加每个经过的终端和交换机的最大延迟来获得，以及相应的技术延迟，计算公式如 3.21 所示：

$$D_{i_EtoEWorst} = \sum_{(k)=first_i}^{last_i-1} D_{i_max}^{(k)} + (|P_i|-1) \cdot L_{tech_max} \quad (3.21)$$

$|P_i|$ 为流 τ_i 访问的节点的数目。

3.4.2 设计与实现

在这一节，设计一组伪代码表示的功能，以此来满足项目的每一个需要。图 3.20 表示设计的需求架构。

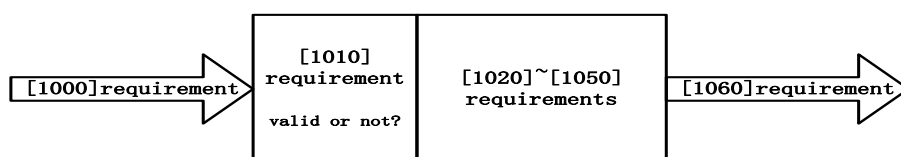


图 3.20 需求架构

由上图所示，定义需求共有 7 个部分，分别是 Requirements1000 至 Requirements1060。Requirements1000 表示作为分析算法，需要对相应输入信息进行获取，Requirements1010 表示算法需要验证每个虚链路在分析前是否有效，判断虚链路有效即是否存在环路以及终端是否可达。Requirements1020 则表示每个链路的最大延迟可与配置值进行比较。Requirements1030 表示每条链路和带宽应该得到保证。Requirements1040 表示每个交换机输出端口可计算并可与定义的 Ddef_maxport 比较。Requirements1050 表示每个终端和交换机的输出缓冲区的积压可计算。Requirements1060 表示网络分析报告必须包括以下几个输出信息：

1. 环路检测与终端可达验证结果；
2. 每条虚链路最差情况下的延迟以及与配置值的校验；
3. 每条虚链路最差情况下的抖动以及与配置值的校验；
4. 每条虚链路的分配带宽；
5. 交换机输出端口的最大延迟以及与配置值的校验；
6. 源终端与交换机输出缓冲区的最大积压；

整个拓扑通过矩阵保存，定义：

```
VLPath[totalVLNum][ maxMultiNumPerVL][ maxNodeNumPerVLMulti]
{
  {{ {ES0,Port0},{Sw0,Port0},{Sw2,Port1}}, //multi0
    {aNULL,aNULL},{aNULL,aNULL},{aNULL,aNULL}}, //multi1 //VL1
  {{ {ES0,Port0},{Sw0,Port0},{Sw2,Port1}}, //multi0
    {aNULL,aNULL},{aNULL,aNULL},{aNULL,aNULL}}, //multi1 //VL2
```

```

{{{ES1,Port0},{Sw0,Port0},{Sw2,Port1}},           //multi0
{{{aNULL,aNULL},{aNULL,aNULL},{aNULL,aNULL}}},    //multi1  //VL3
{{{ES1,Port0},{Sw0,Port0},{Sw2,Port1}},           //multi0
{{{aNULL,aNULL},{aNULL,aNULL},{aNULL,aNULL}}},    //multi1  //VL4
{{{ES2,Port0},{Sw1,Port0},{Sw2,Port0}},           //multi0
{{{ES2,Port0},{Sw1,Port1},{Sw4,Port1}}},          //multi1  //VL5
.....
}

```

3.4.2.1 环路检测与终端可达

网络拓扑中的 VL 链路不允许形成环路，即一个端无法接收自己发送的信息，终端可达是指终端节点一定有路径可能到达，下面显示了设计环路检测的伪代码。返回值为 **true** 表示链路不存在环路，**false** 相反。

```

bool CheckVLRing(int vl,int multi)
    bool flag=true;           //valid, without a ring
    for(int i=0;VLPath[vl][multi][i].NodeID!=aNULL && i< maxNodeNumPerVLMulti;i++)
        for(int j=i+1;VLPath[vl][multi][j].NodeID!=aNULL && j< maxNodeNumPerVLMulti;j++)
            if(VLPath[vl][multi][i].NodeID==VLPath[vl][multi][j].NodeID)
                flag=false;    //invalid, with a ring
                break;
    if(flag==false)
        break;
    return(flag);

```

图 3. 21 环路检测

每条虚链路必须配置正确的目的终端，所有的端系统都可以通过 ES ID 索引。如图表示终端可达的伪代码。返回值 **true** 表示终端可达，**false** 则相反。

```

bool CheckVLdesES(int vl,int multi)
    int desES;
    int l=0,h=0,m;
    for(int i=0;VLPath[vl][multi][i+1].NodeID!=aNULL && i+1< maxNodeNumPerVLMulti;i++);
    desES=VLPath[vl][multi][i].NodeID;           //find the destination ES of the VL
    while()           //dichotomy search
        if(l>h)
            return(false);
        m=(l+h)/2;
        if(ES[m]==desES)

```

```

    return(true);
else if(ES[m]>desES)
    h=m-1;
else
    l=m+1;

```

图 3.22 终端可达

3.4.2.2 最差情况下的 VL 延迟

下面是最差情况下的 VL 延迟伪代码，指端到端信息传输过程中由于调度、硬件等原因引起的最大延迟时间，方法返回计算的延迟值如果满足 Ddef_max 约束，否则返回 false。

```

float WorstDelayEtoE(int vl,int multi)
int node=0;
float delayEtoE=0;
delayEtoE= WorstDelayES(VLPath[vl][multi][0].NodeID, vl);    //worst-case delay in SES
for(node=1;VLPath[vl][multi][node+1].NodeID!=aNULL && (node+1)<
maxNodeNumPerVLMulti;node++)    //worst-case delay in SES and Sw
    delayEtoE=delayEtoE+ WorstDelaySw(VLPath[vl][multi][node].NodeID,VLPath[vl][multi]
[node].PortID, vl)+Lmax;
delayEtoE=delayEtoE+Lmax;    //worst-case delay in SES, Sw and DES
if(delayEtoE<= Ddef_max)
    return(delayEtoE);
else
    return(false);

```

图 3.23 VL 最差情况下延迟

3.4.2.3 最差情况下的 VL 抖动

下面是最差情况下的 VL 抖动伪代码，方法返回计算的抖动值如果满足 Jdef_max 约束，否则返回 false。

```

float WorstJitterEtoE(int vl,int multi)
float jitterEtoE=0;
jitterEtoE=WorstDelayEtoE(vl,multi)-BestDelayEtoE(vl,multi);
if(jitterEtoE<=Jdef_max)
    return(jitterEtoE);
else
    return(false);

```

图 3.24 最差情况下的 VL 抖动

3.4.2.4 VL 带宽

VL 分配的带宽是由网络配置保证，并且可以通过最大帐长与 BAG 进行计算，并必须留有足够空间以应对 VL 的抖动影响。所以相应的带宽计算代码如下：

```
float Bandwidth(int vl,int multi)
    float bandwidth;
    bandwidth=VL[vl].Smax/(VL[vl].BAG-WorstJitterEtoE(vl,multi));
    return(bandwidth);
```

图 3. 25 VL 分配的带宽

3.4.2.5 交换机输出端口的最大帧延迟

下面是交换机转出端口最大帧延迟伪代码，方法返回计算的延迟值 maxd 如果满足 Ddef_maxport 约束，否则返回 false。

```
float MaxFrameDelaySwPort(int SwID,int port)
    float d=0;
    float maxd=0;
    for(int sp=SP0;sp<=SP3;sp++)
        for(int i=0; i<maxVLNumPerSwPortSP && SwPortVL[SwID][port][sp][i].vl!=aNULL;i++)
            d= WorstDelaySw(SwID,port, SwPortVL[SwID][port][sp][i].vl);
            if(maxd<d)
                maxd=d;
    if(maxd<Ddef_maxport)
        return(maxd);
    else
        return(false);
```

图 3. 26 交换机输出端口的最在帧延迟

3.4.2.6 终端或交换机输出缓冲区最大积压

如下所示为终端输出端口的最大积压的伪代码：

```
float BacklogESPort(int ESID)
    float backlog=0;
    for(int i=0; i<maxVLNumPerSESPort && SESPortVL[ESID][i]!=aNULL;i++)
        backlog=backlog+CalcuBurstSES(SESPortVL[ESID][i],ESID);
    return(backlog);
```

图 3. 27 终端输出端口最大积压

如下所示交换机输出端口的最大积压伪代码：

```
float MaxFrameDelaySwPort(int SwID,int port)
    float d=0;
    float maxd=0;
```

```

for(int sp=SP0;sp<=SP3;sp++)
    for(int i=0; i<maxVLNumPerSwPortSP && SwPortVL[SwID][port][sp][i].vl!=aNULL;i++)
        d= WorstDelaySw(SwID,port, SwPortVL[SwID][port][sp][i].vl);
        if(maxd<d)
            maxd=d;
if(maxd<Ddef_maxport)
    return(maxd);
else
    return(false);

```

图 3.28 交换机输出端口最大积压

3.4.3 实验结果分析

利用多链路分组技术计算流的端到端的延迟，但因为缺乏可以与我们的研究成果对比的客观数据。所以有必要对实例中的网络进行仿真，以期获得可以比较的数据，同时仿真的目的也是评估本文基于网络演算方法的真实有效性。本次仿真的结果是关注每个帧经过网络总的延时，暂且不考虑在航电子系统中消息的时间。文献[45][46][47]详细描述在 OPNET 工具中，使用仿真的步骤流程。为了计算简单，假设交换机的端口速率都为 100Mb/s。在设置最大虚链路帧长为 4000b，BAG 为 8ms，在基于静态优先级排队策略时，为方便计算，定义优先级为二级，其中 v1,v2,v5,v7 为高优先级，v3,v4,v6,v8 为低优先级。通过 2.2.2 节的时延模型，可计算的实验结果如表 3.1 所示。

表 3.1 网络时延结果对比

| | FIFO | | | | SP | | | |
|-----|-------|-----------|------------|-------|-------|-----------|------------|-------|
| | NC | E2E_delay | Simulation | MGNC | NC | E2E_delay | Simulation | MGNC |
| VL1 | 480 | 9118 | 6445 | 436.8 | 400 | 9038 | 5743 | 360 |
| VL2 | 480 | 9118 | 7452 | 436.8 | 400 | 9038 | 6463 | 360 |
| VL3 | 480 | 9118 | 7823 | 436.8 | 486 | 9124 | 7935 | 388 |
| VL4 | 480 | 9118 | 6902 | 436.8 | 486 | 9124 | 7213 | 447 |
| VL5 | 360.4 | 8998 | 7442 | 288 | 240.4 | 8878 | 7443 | 216 |
| VL6 | 201.4 | 8838 | 5884 | 181 | 236 | 8874 | 6194 | 174.6 |
| VL7 | 321 | 8818 | 6133 | 295 | 360 | 8778 | 6022 | 316 |
| VL8 | 161 | 8685 | 6798 | 145 | 161 | 8658 | 7290 | 144 |

事实中，工业中的网络拓扑结构是对称的，其间传输的流也是对称的。如图 4.14 所示的机载水处理系统，但为了进行分析，利用图 4.16 所示的网络拓扑。同时决定计算交换机端口的极限值。事实上，其它元件的值也可以计算出来。在没有端到端的延时的结果时，或网络复杂，

仿真较为困难，即缺少可以进行对比的客观数据。但是，国内某研究所为我们提供了交换机的最大允许通过时间为 1ms，构成可参考 2.2.2 节。图 3.29 显示了交换机的每个输出端口最大延时的边界值。

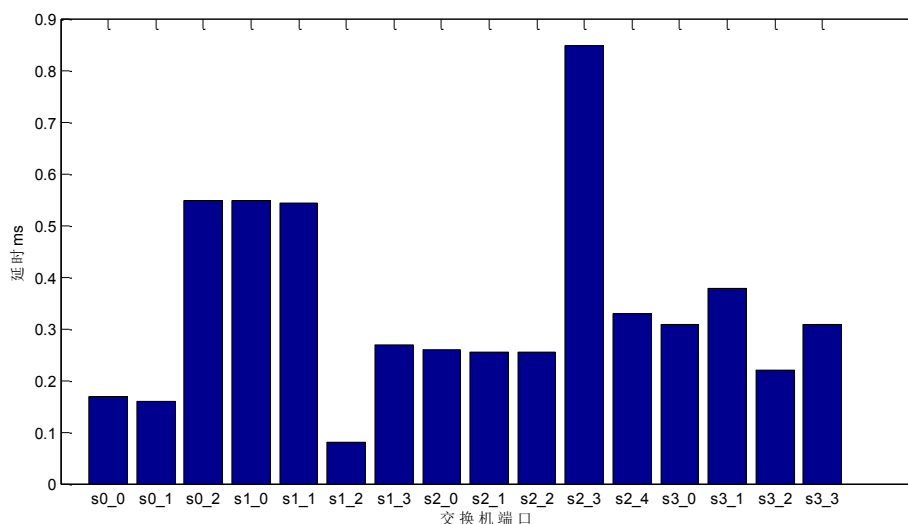


图 3.29 交换机端口延时

由此，可以认为我们的结果是正确的，同时我们的网络演算得出的结果是令人满意的，对于确定网络优劣是有效的。同时基于分析的一般方法，即改变某一个量的同时保证其它量不变，由于对于 AFDX 网络，虚链路上重要的两个参数即为 Smax 和 BAG，所以研究这两个量对于网络确定性的影响。以下的结果由于信息在硬件的的固定时延是一定的，对研究的结果不会产生影响，所以不使用 2.2.2 节中的模型进行研究，只分析可变的时延。首先，计算 Smax 针对我们方法的影响结果如图 3.30 所示。

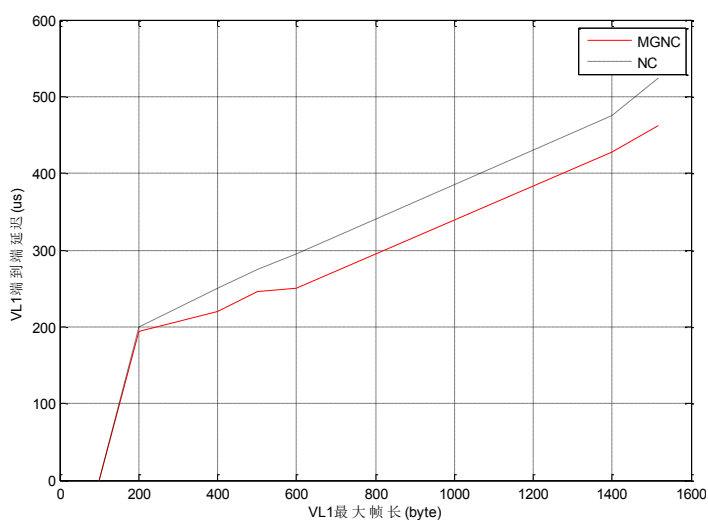


图 3.30 Smax 对于 VL1 的延迟影响

如图 3.30 所示描述了虚链路 V1 在最大帧长(200-1500B)的端到端延迟。在较低帧长时,本文的方法效果不明显,但随着帧长度逐渐增大,本文的方法的优势明显。计算结果明显要优于普通网络演算的结果,上限的精度(更紧了)9%-23%。如果与 2.3.2 节的轨迹法相比,依旧是存在优势的。

同理可测得另一个参数 BAG 对于延迟的影响结果,针对传统 NC 与 MGNC 作比较,结果如图 3.31 所示。

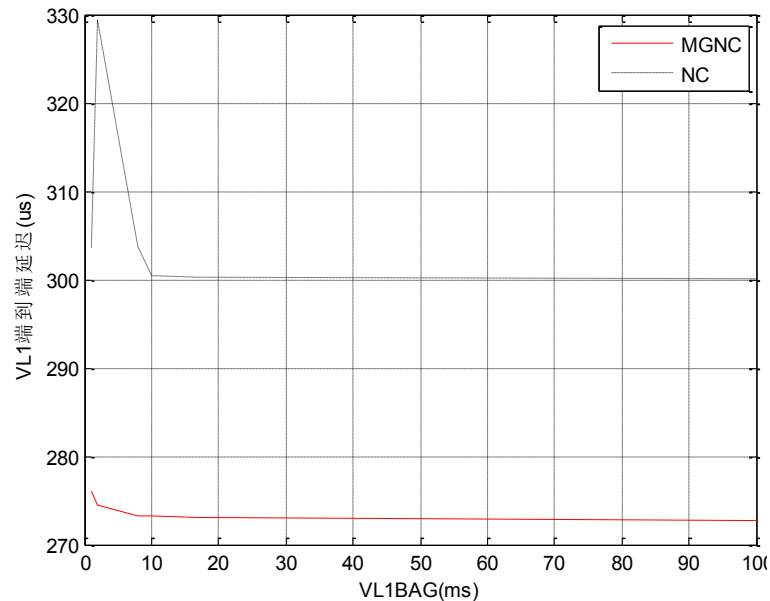


图 3.31 BAG 对于 VL1 的延时影响

如图 3.31 所示,同理控制 BAG(1ms-128ms)的大小,对比 NC 与 MGNC 的效果, MGNC 的效果依旧好过 NC 的结果,且在较长的区间内都保持这种好的状态。

3.5 本章总结

本章利用网络演算相关理论,分析 AFDX 网络中采用的数据流包络应用,应用网络演算理论成果,并根据实际网络流的特性,精确流的到达曲线,设计了基于多链路分组的技术,计算获得更紧致的技术参数上界,最后设计了相关的网络确定性性能评估分析算法,对网络分析的相关确定性参数作了算法上的实现。

第四章 多链路分组网络演算的 AFDX 网络确定性分析功能实现

4.1 系统概述

本系统是基于航空电子领域所开发的，不同于传统的网络仿真软件，其所有的建模对象都是领域对象，集成航空电子领域特有的属性。IMA 系统是由一组硬件组成，并在这基础上提供了相应的计算、通信和相应的接口功能，如飞行控制、自动驾驶、制动、燃料管理等。这些硬件包括了 RDCs、控制器、AFDX 端系统、IPM 和 AFDX 交换机等^[48]。

而本文中涉及的利用基本多链路分组技术，来对 AFDX 网络进行时延、抖动等相关确定性参数的验证。作为网络确定性分析的一个重要组成部分嵌入在本系统中，作为构建 IMA 整个系统的关键的一个部分。作为 IMA 系统，其配置文件在整个设计开发过程中至关重要，对于 AFDX 网络的拓扑，相关参数如 BAG, L_{\max} 以及虚链路的分配都是通过配置文件来实现的。从图 4.1 中可以看出本功能实现的整个框架，以及功能实现的逻辑。

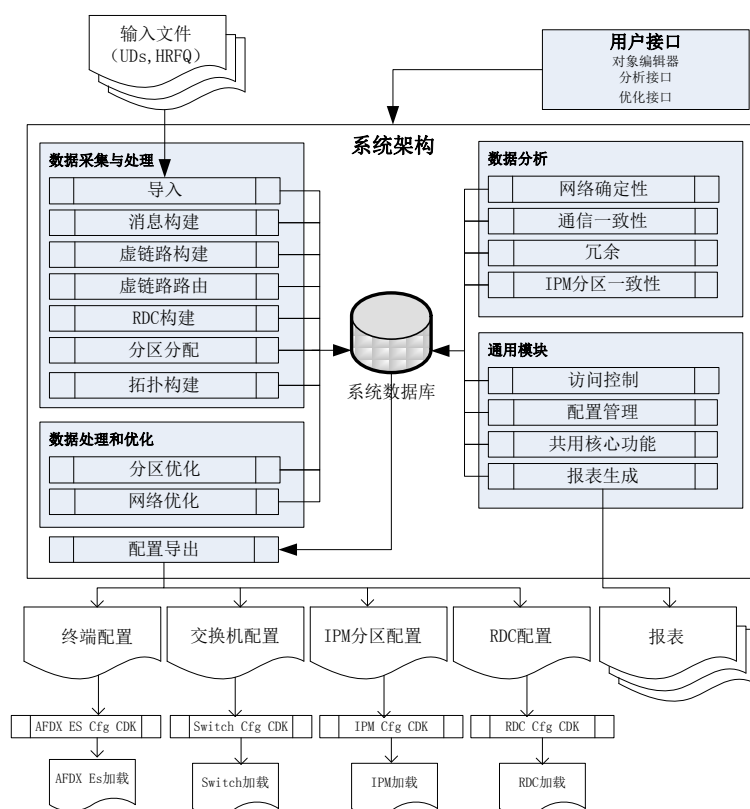


图 4.1 系统框架图

系统使用过程分成三个部分，第一为设计，构建航电子系统，包括了消息设计、虚链路设计、RDC 设计、分区设计、拓扑设计，第二为分析验证，是对设计结果的分析，直接反应设计

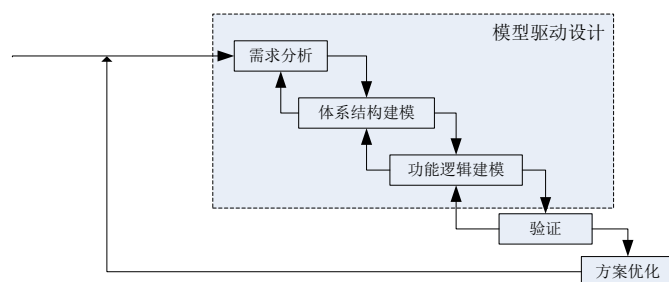


图 4.2 系统建模流程设计

本分析插件主要在利用在 IMACT 中,使得 IMACT 的工具不仅有设计综合化航空电子系统的能力,还有分析所设计系统确定性的能力。如所示为系统顶层部分模型。

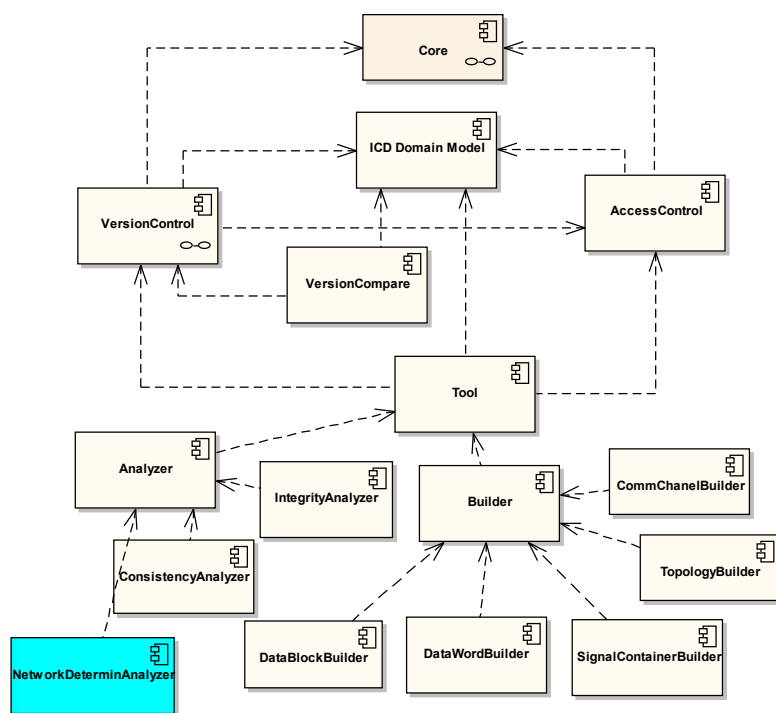


图 4.3 系统顶层部分模型

作为 IMACT, 用户共分为 SystemDesigner, SystemDeveloper, SystemTester, VersionManager, Administrator。各用户有各自的工作职责。如图 4.4 所示是对关键的用户进行用例分析。

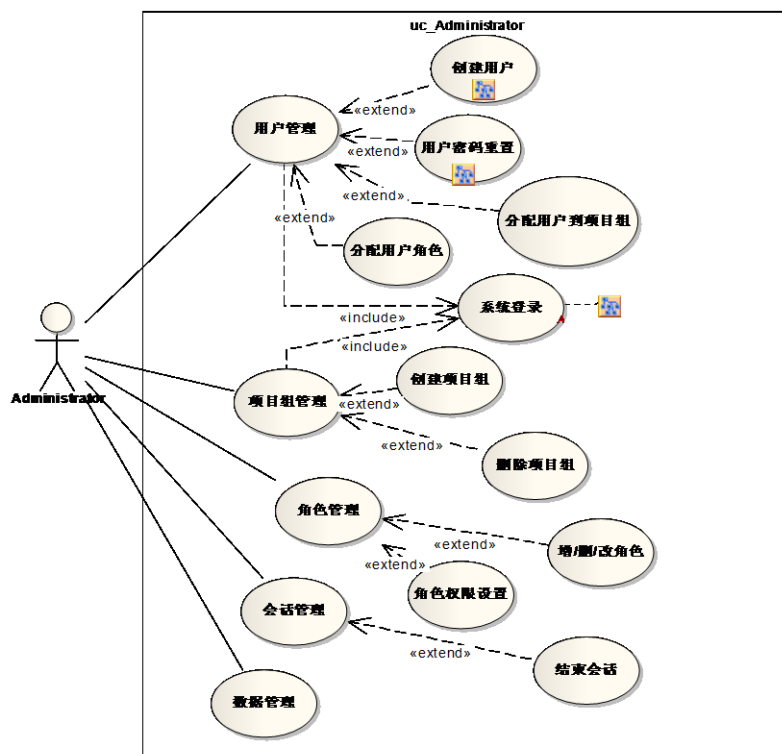


图 4.4 管理员用例图

而作为系统设计人员，承担着对综合化航空电子系统的设计功能，同时相应的分析验证功能也是由该角色实现完成。所以如图 4.5 为系统设计员用例图。

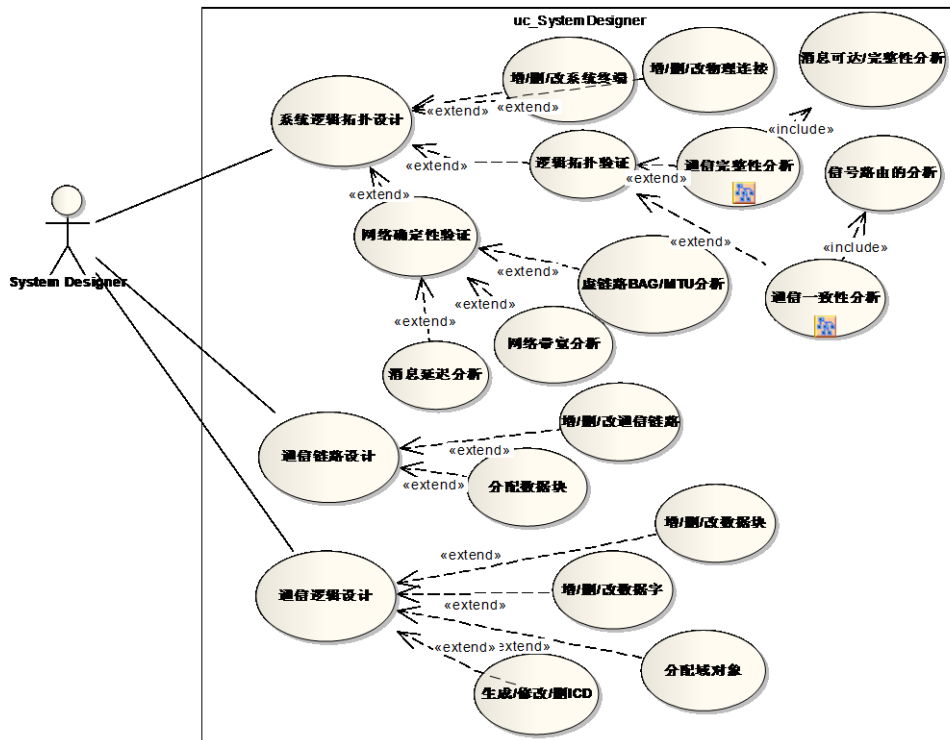


图 4.5 系统设计员用例图

4.2 开发环境与关键技术

4.2.1 系统架构介绍

本系统是采用 Client/Server(客户机/服务器)结构, 基于 Eclipse 平台进行开发, 利用 Eclipse 的插件机制, 将系统的各模块开发成 Eclipse 的插件^[48]。具体架构如图 4.6 所示。

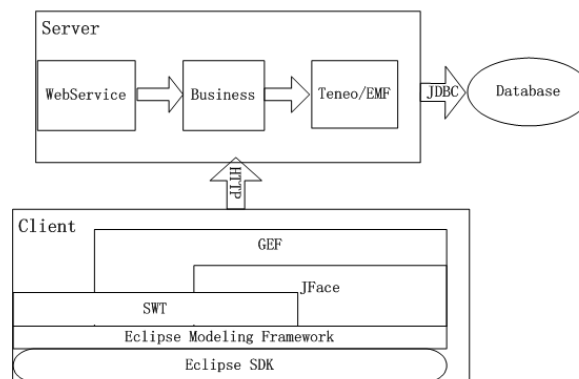


图 4.6 系统架构

由于采用的是 C/S 架构, 所以该系统服务器向客户端提供各种 Web 服务接口, 处理客户端的各种请求。由于客户端可以运行在不同平台上, 都可以调用服务器端的 Web 服务, 则多个平台的客户端则可以不用专用或通用的其它软件或硬件, 即可以实现相互之间的无障碍交互。完成整个系统功能^[48]。

4.2.2 关键技术

Eclipse 是一个开源的可扩展开发平台，主流的 JAVA 语言开发 IDE。当然通过提供插件的方式，它也可以进行 C、C++等语言的开发。

EMF(Eclipse Modeling Framework)是 Eclipse 的采用的一套模型框架，同时作为 Eclipse 模型运动架构中重要部分，在许多 Eclipse 的项目都有应用，诸如 GEF，使用 EMF 技术可以通过建模，并使自动生成准确而可利用定制的 Java 代码^[49]。在本系统的实现过程，底层的数据模型都是通过 EMF 技术实现的，最直接一般的做法，是通过类 UML 建模的来构建数据模型，EMF 采用的是一个简化了的 UML，文献[49]详细介绍了 EMF 相关技术的细节。

SWT(Standard Widget Toolkit)，即窗口小部件工具箱^[50]，它使 JAVA 语言在图形用户界面领域的有了长足的发展。JFace 是 SWT 的扩展，其主要目的是为了开发 Eclipse IDE 环境，相对于 SWT，它的使用更加简捷高效，在本系统的设计开发过程中，优先选用 JFace 的相关控件。

GEF(Graphical Editor Framework)是具有标准 MVC 结构的图形编辑框架^[49]，其中 Model 由我们自己根据业务来设计，进行相应的建模，并赋予模型改变通知的机制（如监听者模式），将 Model 的所作的变化通知 Control 层，系统中是通过前文的 EMF 来定义，自动生成的代码完全符合相关要求；Control 层主要是 EditPart，同时也是 GEF 的核心部件，主要负责前台界面和后台数据的交互沟通；View 层使用 Draw2D 技术，是 Model 的图形化解决方案。

Teneo 是 Eclipse EMF 作为实现关系模型映射和运行时数据持久化的技术，主要是集成了 Hibernate 与 EMF，使用模型的自动配置^[51-52]，完成 ORMapping，使得开发人员从烦琐的配置过程中解脱出来，更多的关注模型本身。从而极大的提高了开发的效率。

Apache CXF 是面向 SOA 服务并提供了一套基础框架，创建适合 SOA 环境的各种各样的 WEB 服务，其中有 SOAP/HTTP 服务及 REST/HTTP 服务。Apache CXF 支持的协议非常广泛，本系统整合 Spring，通过注解与配置来提供 Web 服务供客户端使用，处理用户请求，方便用户使用，节省开发时间^[53]。

百灵报表(BIRT)作为实用的开源报表工具，不仅可以结合 JAVA 语言进行开发，实现丰富的定制报表样式 BIRT 是一个 Eclipse-based 开放源代码的报表系统也能很好的满足商业智能领域中报表设计的相关需求，在客户端程序和网页程序中都有应用。本系统利用 BIRT 技术对系统最后导出的配置文件，分析报告作自定义的设计，通过连接 IMACT 的数据库，并通过系统提供的 IMACT 查询语言(IQL)，进行报表设计。具体流程如图 4.7 所示。

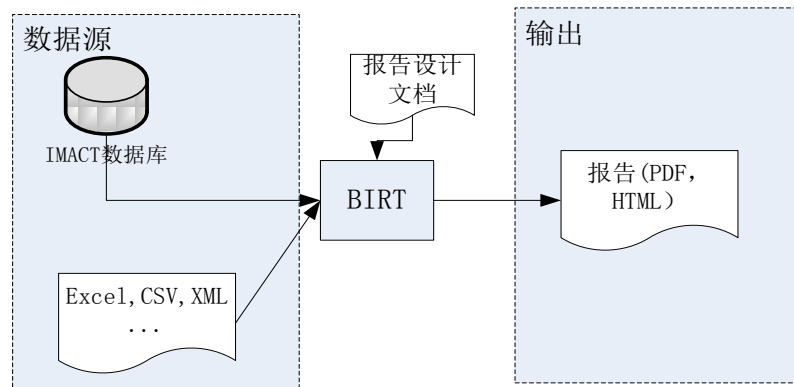


图 4.7 BIRT 使用流程

4.3 AFDX 建模

通信链路是数据块运行的通道，通信链路设计就是需要研发的系统支持多种链路类型，根据不同的协议，连接航空电子各个部件和分子系统，使之可以接入访问。航空电子系统中，数据链路有：A429、1553B、AFDX 虚链路等。

AFDX 是 IMA 中一种总线标准，是数据通信标准。本系统采用 EMF 技术对 AFDX 结构建模，包括虚链路、终端、端口以及之间关系等，如图 4.8 所示。

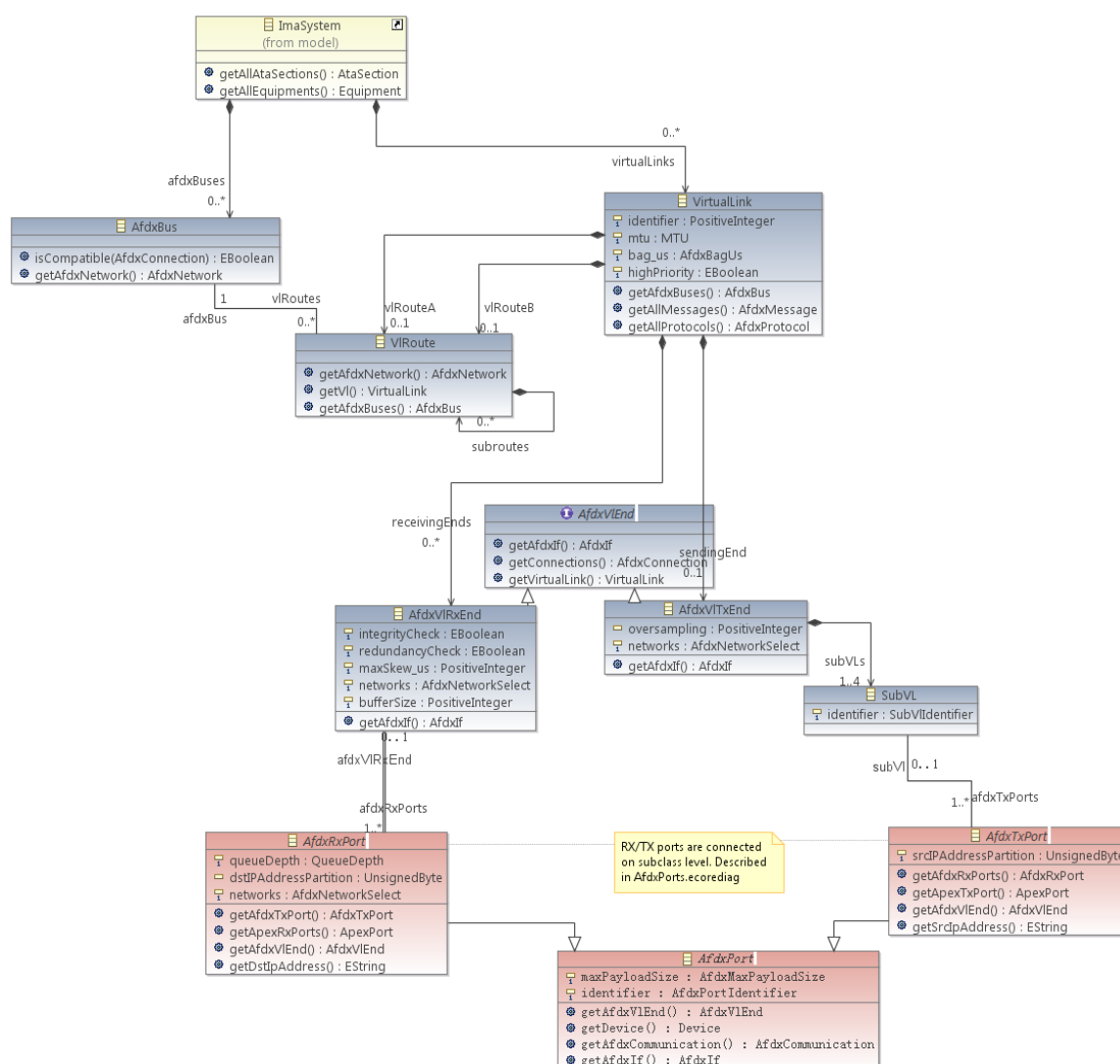


图 4.8 AFDX 结构建模

利用 EMF 对 AFDX 建模后，可以基于模型驱动，自动生成模型的代码，应用于图形化航空电子系统网络设计，该模型中，各个类的详细介绍如表 4.1 所示。

表 4.1 AFDX 网络模型中各类的说明

| 类名 | 说明 |
|--------------|---------------------------------------|
| ImaSystem | 指一个 IMA 的子系统，配置以完成航空电子中的相应功能 |
| AfdxBus | 指 AFDX 总线，包含了 AFDX 协议中的一些约束和参数 |
| Virtual Link | 是指 IMA 系统中，AFDX 网络中配置的虚拟链路，每条链路都有自己属性 |
| VRoute | 针对冗余管理和可靠性，所要设计的 AB 网络路由 |
| AfdxVIRxEnd | AFDX 网络虚拟链路的目的端 |
| AfdxVITxEnd | AFDX 网络虚拟链路的源端 |

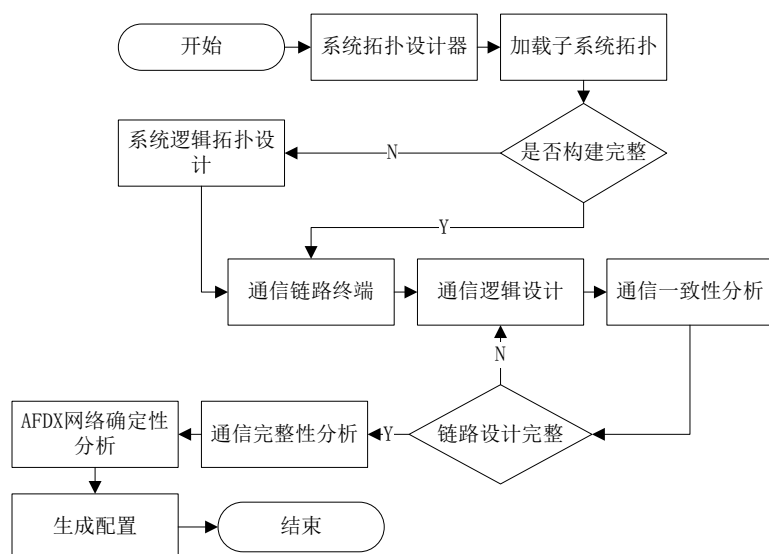


图 4.10 网络分析活动

如图 4.10 所示，是进行网络分析的活动图，对于网络确定性的分析主要包括了信号路由的分析、虚链路 BAG/MTU 分析，网络带宽分析、消息延迟分析、消息可达/完整性分析。从分析的一个主方法，如图 4.11 可以直观的说明确定性分析的流程。

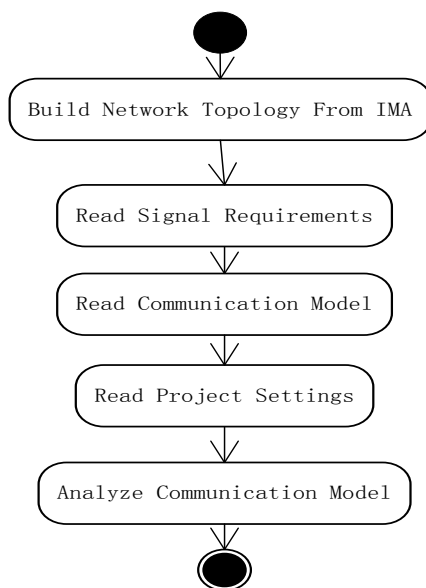


图 4.11 网络确定性分析流程

相应的代码参考如下所示。

```

public void analyzeDataflowAllocation() {

    logTestOnly("----- Build Network Topology From IMA -----");
    network = new Network();
    network.buildTopologyFromIma(imaProject);
    logTestOnly("----- Read Signal Requirements -----");
    RequirementsModelImporter importer = new RequirementsModelImporter(imaProject, network);
    requirementsModel = importer.importRequirementsModel();

    logTestOnly("----- Read Communication Model -----");
    commImporter.setNetwork(network);
    commImporter.setRequirementsModel(requirementsModel);
    commModel = commImporter.importCommunicationModel(imaProject);
    logTestOnly("----- Read Project Settings -----");
    ProjectSettings projectSettings = DomainModelImportSettings.importSettings(imaProject);

    logTestOnly("----- Analyze Communication Model -----");
    NetworkDeterministicAnalyzer analyzer = new NetworkDeterministicAnalyzer(commModel,
requirementsModel,
network, projectSettings, getAnalyzerReporter());

    if (commModel.getAllVIs().size() == 0) {
        getAnalyzerReporter().reportWarn("Virtual Link not found in IMA, nothing to do.", new
NodeId(imaProject.getImaSystem()));
        return;
    }

    analyzer.analyzeCommunicationModel();
}

```

图 4.12 确定性分析框架代码

系统拓扑的分析,分析系统构建的 AFDX 网络拓扑,插件能够自动分析数据库以验证 AFDX 网络拓扑的正确性。主要是确保所有的 IMA 模块配置在指定位置;所有模块中,AFDX 交换机与 AFDX 终端通过 AFDX 总线连接;冗余网络 AB 不存在连接;所有设备都接入电源,满足其需求。以上的分析可能会报告错误且只有通过修改 AFDX 拓扑网络更正,如表 4.2 所示为拓扑分析接口中会报告的信息,了解这些信息,可以帮助设计人员更好的理解该插件的功能。

表 4.2 确定性分析-拓扑分析

| 分析接口 | 分析结果 | 分析报告 |
|-------------------------|---|---|
| verifyIPMAFDXConnection | IPM AFDX 网络连接验证跳过: 无 IPM | IPM A: AFDX 接口无连接 |
| verifyRDCAFDXConnection | RDC AFDX 网络连接验证跳过: 无 RDC | RDC A: 无 AFDX 连接 |
| verifyAFDXNetworkSW | AFDX 网络连接验证跳过: 无 Switch | AFDX 网络验证: 子网溢出; 所有交换机被接入一个网络; 存在交换机不需要 AB 网络; 冗余网络 AB 大小不一样 |
| verifyIPMAAllocation | IPM 机架分配验证跳过: 无 IPM | IPMA: 未分配进机架 |
| verifyPowerBusMatching | 机架电源验证跳过: 无机架 | RDC 电源验证跳过: 无 RDC Switch 电源验证跳过: 无 Switch Controls 电源验证跳过: 无 Switch |
| verifySpareResource | 机架空闲槽验证跳过: 无机架 Switch 空间接口验证跳过: 无 Switch | 机架 A: 使用百分之 X 插槽, 允许百分之 Y 插槽 交换机 A: 使用百分之 X 接口, 允许百分之 Y 接口 |

或者如对虚链路的 BAG 和 MTU 等指标的分析, 即 verifyVls 接口验证, 此接口包含了 verifyVIPaths, verifyVlBag, verifyVlBandwidth, verifyVlMtu。对于 verifyVIPaths 接口说明如表 4.3 所示:

表 4.3 确定性分析-环路分析

| 接口 | 分析结果 | 分析报告 |
|---------------|---------------------------------|--|
| isTreeValid | 检查路径是有效的即没有交错的分支(手动编辑可能会产生这种情况) | VL X 中的节点 Y 有 Z 个父节点, 意味着虚链路的路径树无效 |
| arePathsValid | 验证路径的源和目的节点 | 1. 网络 A 中 VL X 的源节点 Y, 但其它链路 Z 存在接口连接 Y 2. 网络 A 中 VL X 的目的节点 Y, 但 Y 未在链路目的节点列表中 |

isTreeValid 具体的情况有两种, 如图 4.13 所示:

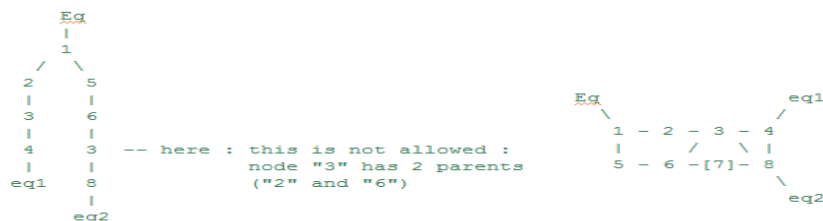


图 4.13 虚链路分析无效示例

verifyVIBag 可以计算各链路的 BAG 大小, 并与需求比较, 提供相应的报告信息, 如 VL X 的 BAG 为 Y, 但此信号承载的 VL 定义的 BAG 为 Z。同理可得 verifyVIBandwidth, verifyVIMtu。

4.4 实例分析

通过 Eclipse 的插件开发, 设计开发了一个插件, 装配进 IMA 系统中, 实现了网络确定性的分析, 利用 IMA 系统中的设计作为输入, 进行网络确定性分析, 如图 4.14 所示设计的机载水处理系统。

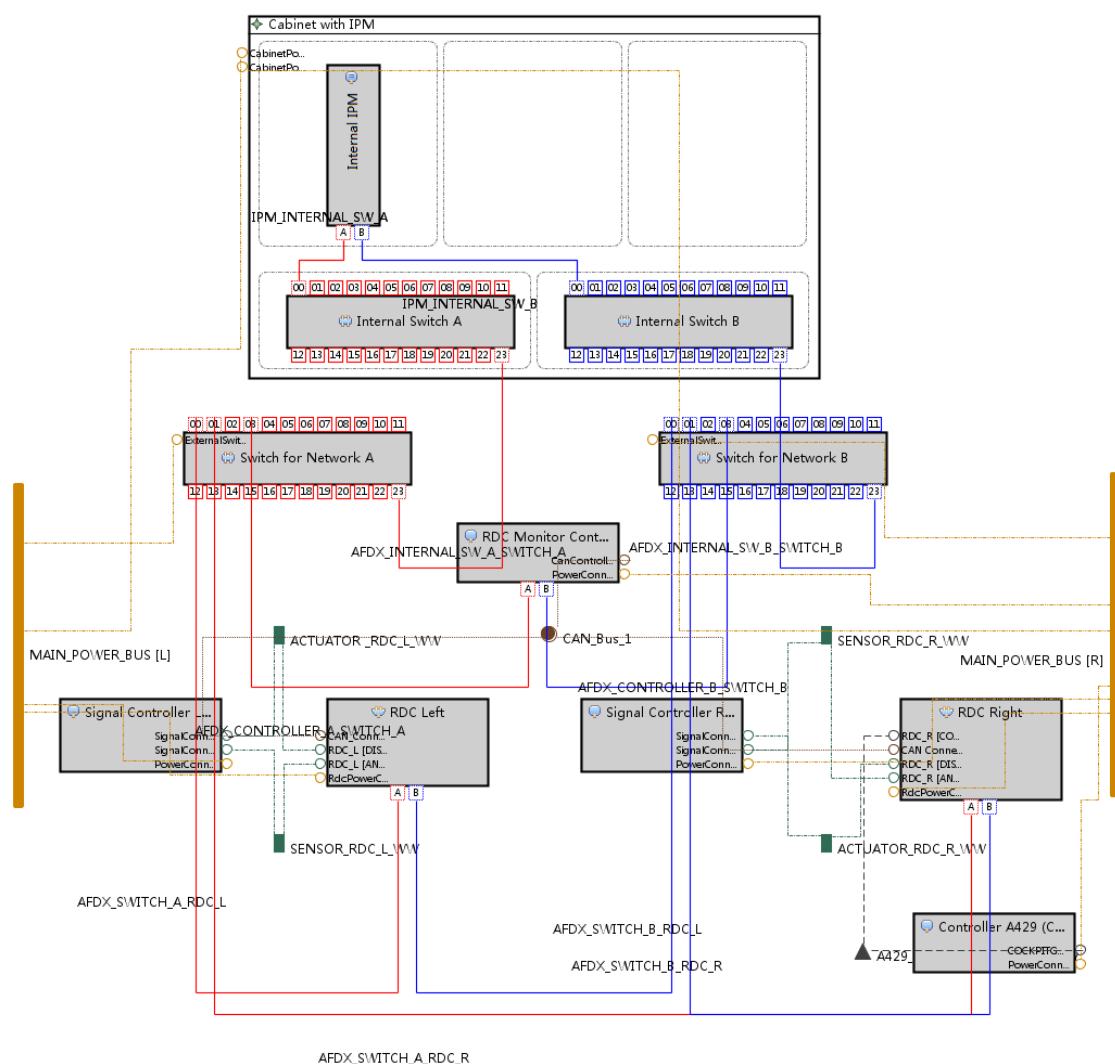


图 4.14 水处理系统

同时针对 AFDX 的分析, 关注的一个重点是虚链路配置, 所以在系统中提供了对于 VL 的显示, 帮助设计人员进行更直观的分析以及配置, 水处理系统共设计了 5 条虚链路。如图 4.15 所示为 VL₁。

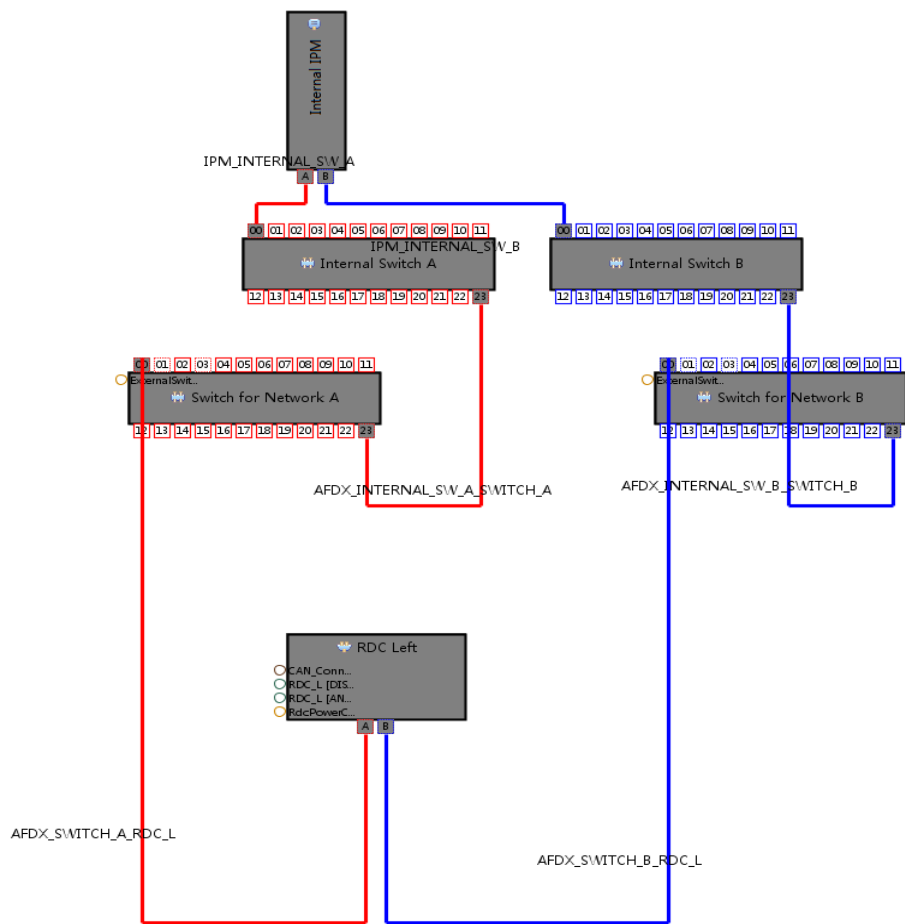


图 4.15 系统虚链路

作为虚链路配置的结果，提供了导出报表的功能，以指导实施人员进行现场部署。如表 4.4 所示是对虚链路配置拟提供的一种报表输出。

表 4.4 链路配置报表

| Virtual Link | Identifier | MTU | Bag (us) | Sending End | Receiving End | Bus |
|--------------|------------|-----|----------|--------------|---------------|-----------------------------|
| VL_1 | 1 | 640 | BAG_1000 | AfdxVITxEnd2 | AfdxVIRxEnd0 | AFDX_INTERNAL_SW_A_SWITCH_A |
| | | | | | | AFDX_INTERNAL_SW_B_SWITCH_B |
| | | | | | | AFDX_SWITCH_A_RDC_L |
| | | | | | | AFDX_SWITCH_A_RDC_R |
| | | | | | | AFDX_SWITCH_B_RDC_L |
| | | | | | | AFDX_SWITCH_B_RDC_R |
| | | | | | | IPM_INTERNAL_SW_A |
| | | | | | | IPM_INTERNAL_SW_B |

对于网络的分析，设计在 Verify 功能菜单中，对每个项目可以进行当前已经分析插件提供的分析功能，在分析框架中，加入本文设计的网络确定性分析。进行 AFDX 网络的分析，并最

终以报表的形式提供给用户下载, 使用和查看。在分析结果满足系统设计的要求(HRFQ)时, 可进行下一步工作。否则需要进一步的迭代, 优化设计。在水处理系统中, 根据需求定义了如表 4.5 所示的参数。

表 4.5 参数列表

| 名称 | 含义 | 类型 |
|------------------------------|------------|--------------------------|
| AC_FILL_RESERVOIR_WW_LSTATUS | 左蓄水池状态 | Global Float Parameter |
| AC_FILL_RESERVOIR_WW_RSTATUS | 右蓄水池状态 | Global Float Parameter |
| AC_VALVE_WW_LSTATUS | 左阀门指令 | Global Boolean Parameter |
| AC_VALVE_WW_RSTATUS | 右阀门指令 | Global Boolean Parameter |
| RDC_LEFT_STATUS | 左 RDC 状态 | Global Boolean Parameter |
| RDC_RIGHT_STATUS | 右 RDC 状态 | Global Boolean Parameter |
| RDC_LEFT_ERR_INFO | 左 RDC 错误信息 | Global Opaque Parameter |
| RDC_RIGHT_ERR_INFO | 右 RDC 错误信息 | Global Opaque Parameter |

对于机载水处理系统需要完成的任务包括, 确定容器是否已满、计算可用的空间、计算飞机平衡、控制阀门、向面板报告。如图 4.16 所示表示 IMA 系统一水处理系统。

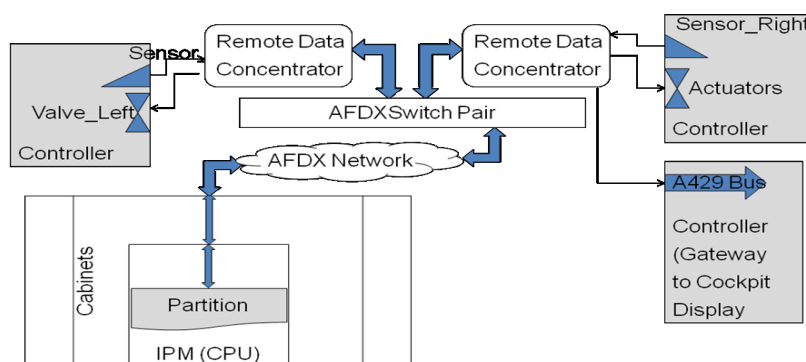


图 4.16 水处理系统(IMA)

如图 4.17 所示表示对于全局参数如表 4.5 所示的传递, 沟通过程。

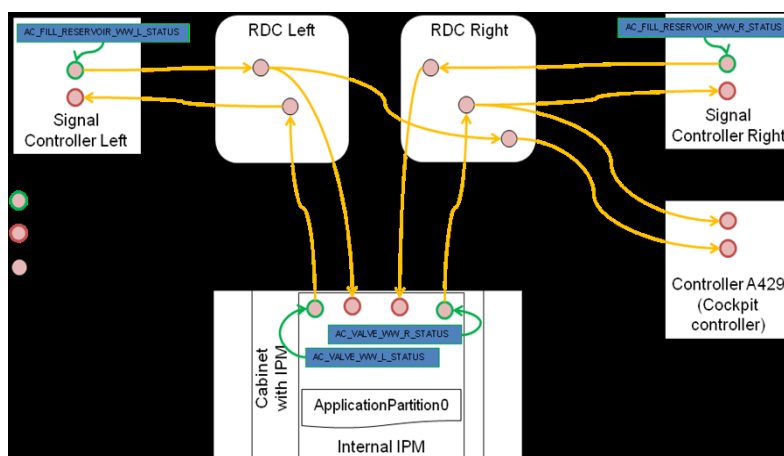


图 4.17 全局参数

在水处理系统设计完成之后，就是对设计分析验证过程，这里使用第四章中设计的算法，启动分析网络确定性分析过程。

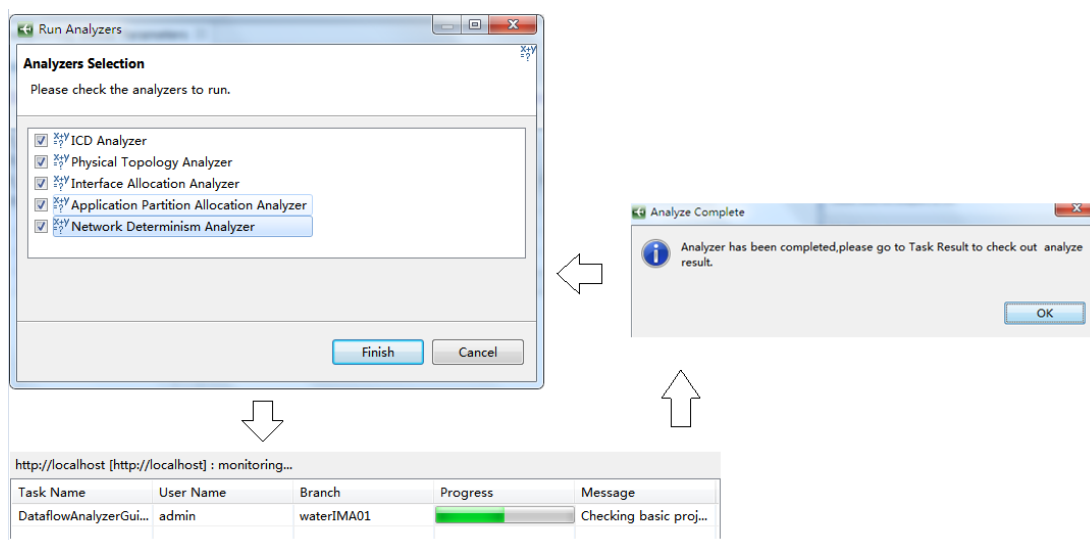


图 4.18 IMA 分析过程

分析结果以报表的形式，如图 4.19 所示。提供以设计人员下载，查看，此处的报表信息以及格式都可以自定义的定制，信息通过自定义的语言 IQL 进行收集处理。

waterIMA01: Reports [Read-only; unlocked]

Branches: waterIMA01 / Archive: waterIMA01 / Reports

| Name | Date | Content | Size | Mime Type | Type Name | Author | Group | Report Configu... |
|----------------|------------------|--------------------|--------|-----------|-----------|--------|-------|--------------------|
| Access | 14-12-11 下午1:... | File [13.3 KB] | 13661 | html | Report | admin | NONE | Access Control |
| A429 Bus | 14-12-19 下午3:... | File [16.0 KB] | 16368 | html | Report | admin | NONE | A429 Bus Hier... |
| A429 Bus | 14-12-19 下午3:... | File [12.1 KB] | 12359 | html | Report | admin | NONE | A429 Bus Table |
| AFDX Bus | 14-12-19 下午3:... | File [114.4 KB] | 117128 | html | Report | admin | NONE | AFDX Bus Hier... |
| AFDX Bus | 14-12-19 下午3:... | File [38.8 KB] | 39760 | html | Report | admin | NONE | AFDX Bus Table |
| AFDX Buses ATA | 14-12-19 下午3:... | File [14.4 KB] | 14706 | html | Report | admin | NONE | AFDX Buses A... |
| AFDX Message | 14-12-19 下午3:... | File [29.6 KB] | 30335 | html | Report | admin | NONE | AFDX Messag... |
| AFDX Message | 14-12-19 下午3:... | File [95.9 KB] | 98201 | html | Report | admin | NONE | AFDX Messag... |
| AFDX Ports | 14-12-19 下午3:... | File [19.6 KB] | 20073 | html | Report | admin | NONE | AFDX Ports Ta... |
| AFDX Switches | 14-12-19 下午3:... | File [45.0 KB] | 46048 | html | Report | admin | NONE | AFDX Switches... |
| AFDX VL | 14-12-19 下午3:... | Download [30.0 KB] | 30682 | html | Report | admin | NONE | AFDX VL Routi... |
| Available | 14-12-19 下午3:... | File [6.8 KB] | 6996 | html | Report | admin | NONE | Available Inter... |
| RX AFDX VL per | 14-12-19 下午3:... | File [5.7 KB] | 5787 | html | Report | admin | NONE | RX AFDX VL p... |
| TX AFDX VL per | 14-12-19 下午3:... | File [5.1 KB] | 5202 | html | Report | admin | NONE | TX AFDX VL p... |
| User Manual | 14-12-19 下午3:... | File [52.4 KB] | 53612 | html | Report | admin | NONE | User Manual ... |
| Analyzer | 14-12-19 下午4:... | File [6.7 KB] | 6849 | pdf | Report | admin | NONE | Analyzer Results |

图 4.19 报告列表

或者通过本系统提供的分析结果查看界面对分析结果直接进行查看，如图 4.20 所示。

| Severity | Message | Object Id |
|----------|---|-----------|
| OK | Signal requirements and realization mapping is OK. | WaterAnc |
| OK | Source and targets of signal requirements are fulfilled. | WaterAnc |
| WARNING | VL VL_51 for network NETWORK_A has (among others) the target nodeRDC Left but its path does not lead to this node | VL_51 |
| WARNING | VL VL_52 for network NETWORK_A has (among others) the target nodeRDC Right but its path does not lead to this node | VL_52 |
| ERROR | VL VL_1 for network NETWORK_A has 1 paths which have as last node Internal Switch A but this node is not defined in the list of target nodes of the VL | VL_1 |
| WARNING | VL VL_1 for network NETWORK_A has (among others) the target nodeInternal IPM but its path does not lead to this node | VL_1 |
| ERROR | VL VL_25 for network NETWORK_A has 1 paths which have as last node Internal Switch A but this node is not defined in the list of target nodes of the VL | VL_25 |
| WARNING | VL VL_25 for network NETWORK_A has (among others) the target nodeInternal IPM but its path does not lead to this node | VL_25 |
| ERROR | VL VL_42 for network NETWORK_A has 1 paths which have as last node Internal Switch A but this node is not defined in the list of target nodes of the VL | VL_42 |
| WARNING | VL VL_42 for network NETWORK_A has (among others) the target nodeInternal IPM but its path does not lead to this node | VL_42 |
| OK | Network bandwidth load of each AFDX bus is OK. | WaterAnc |
| OK | Requirement for Maximum Delay of each AFDX Message is fulfilled in worst case. | WaterAnc |
| OK | Aggregation / Separation Requirements are OK. | WaterAnc |
| OK | Availability / Integrity Requirements of signals are OK. | WaterAnc |

图 4.20 分析报告

如图 4.20 的分析报告指出在信号路由存在问题，根据报告提示，可以发现虚链路 VL_1 定义了路径并没有设置在路由中。所以可以对链路 VL_1 终端目的端进行检查，从而准确的定位错误，同时对于虚链路 BAG/MTU 分析，网络带宽分析、消息延迟分析、消息可达/完整性分析都显示 OK，即满足宿主功能需求。在第二期的开发计划中，拟采用 JFreeChart 等图表插件技术，将相应的技术指标以图表，直观表述出来。供设计人员分析使用。

4.5 本章总结

本章首先对系统进行了介绍概述，分析了系统的架构介绍以及使用的主要关键技术，进而强调了本文所涉及的模块的分析设计，同时基于模型驱动对 AFDX 网络进行 EMF 建模，构建网络确定性性能评估插件，最后对插件的使用流程作了介绍，说明本文方法在工程中的应用

第五章 总结与展望

5.1 工作总结

根据 FAA 针对民用航空器适航认证的要求,民用航电系统机载网络总线必须能提供数据传输的“确定性”能力。而 AFDX 是基于商用以太网的星型分组交换结构,由于交换节点必然存在的拥塞情况,导致数据传输的端到端时延存在一定抖动,同时数据存在由于拥塞而被丢弃的可能。如何在 AFDX 数据传输中保证端到端时延上界并且为缓冲区分配合适的缓存大小(积压上界)成为了 AFDX 网络满足航电系统确定性要求和通过适航认证所重点关注的问题。自从 1991 年 R. Cruz 第一次提出网络演算以来,网络演算迅速成为一种网络定量分析工具,并得到了长足的发展和研究。网络演算是基于最小加代数的一系列结论,通过它对网络进行分析可获得定量公式,通过水平与垂直两个方向的计算,可获得时延,积压上界,有利于对一些分组交换网络的基本属性包括网络时延、时延抖动以及队列缓冲区大小等的定量研究。空客 A380 的 AFDX 网络适航认证即采用了网络演算方法。

本文的主要工作如下:

1. 对 AFDX 协议的各个方面进行研究,分析 AFDX 网络的特点,特别是其中与网络确定性相关的方面。研究分析 AFDX 网络确定性方法,对常用的几种进行分析比较。
2. 研究网络演算的整个理论,并在 AFDX 网络中的应用进行研究,并进行 AFDX 网络时延模型的分析,并应用于第四章的算法当中。提出多链路分组技术,同时对输出流曲线做更精确化调整,从而改进传统的网络演算分析方法,从而获得更紧致的上限。优化分析的结果,使分析结果能更好的指导 IMA 设计。
3. 将本文提出的多链路分组应用于工程实践中,在 IMA 配置工具中,基于插件开发应用本方法,进行 IMA 系统的评估验证分析。

5.2 研究展望

事实上,我们的分析工具提供的是边界,即在最坏的情况下达到的最大值(延迟或队列长度)。但根据直觉和模拟结果,我们认为发生这些最坏情况的概率是非常低的。我们可以考虑开发一种更随机的方法作为该分析方法的补充,如统计网络演算方法,是否能作为我们确定性分析工具的补充。

本文中研究中仍然存在一些可以改进的方面和可研究的方向,主要有:

1. AFDX 针对相关参数的上界中，更精确分析数据流的特征，利用精确数据流^[54-56]，优化到达曲线或服务曲线，同时对相关参数的上界进行更合理的压缩，从而进一步提高相应数据分析的结果。
2. AFDX 网络中存在上千条虚拟链路，每条链路又有不同的参数（最小最大帧长，发送周期等），不同的链路会有不同的端到端时延要求，端到端的路径选择也可以有很多条不同路径。系统工程师完全依靠手工配置这些虚拟链路的参数工作量大，而且不能保证全局最优，文献[5]提供了一种配置方法，但跟国外受到工程检验的相关技术相比，还远远不够。网络优化工具可以依据一定的设计约束，自动进行网络参数分配，路径选择等工作，网络优化工具可以大大减轻航电系统工程师的工作难度和强度，同时也可以提高网络资源的使用率，使网络性能达到一定程度的最优，所以在应用领域设计相应的优化工具，则势在必行，而目前国内相关研究较少^[55]。

参考文献

- [1] Jerome Griebu. Analyze and Evaluation for Techniques for communication Ethernet with interconnection of avionics systems [D]. France: Institut national de polytechnique - Toulouse. 2004.
- [2] GE Fanuc Embedded Systems. AFDX/ARINC 664 Protocol Tutorial [M]. Systerra computer GmbH, 2011.
- [3] 金德琨,敬忠良,王国庆,吴光辉. 民用飞机航空电子系统[M].上海:上海交通大学出版社,2011.
- [4] ARINC. Arinc project paper 664: Aircraft Data Network, Part 7, Avionics Full Duplex Switched Ethernet (AFDX) Network[S]. 2009.
- [5] 刘成,何锋,王彤,周立. 一种 AFDX 网络虚拟链路的路由配置算法[J].电光与控制.2010,17(12):20-23+52.
- [6] Federal Aviation Administration. Data Network Evaluation Handbook[R]. 2009.
- [7] Federal Aviation Administration. Data Network Evaluation Criteria Report[R]. 2009.
- [8] Aeronautical Radio Inc., ARINC specification 664P7, Aircraft Data Network, Part 7, Avionics Full Duplex Switched Ethernet (AFDX) Network[S]. 2009.
- [9] 蒲小勃. 现代航空电子系统与综合[M]. 北京:航空工业出版社.2013.
- [10] Watkins C B, Walter R. Transitioning from federated avionics architectures to Integrated Modular Avionics[C]//Digital Avionics Systems Conference. IEEE, 2007: 2. A. 1-1-2. A. 1-10.
- [11] Schenkelberg R H. Low cost integrated modular avionics (IMA)[C]//Aerospace and Electronics Conference. Proceedings of the IEEE 1996 National. IEEE, 1996, 1: 48-55.
- [12] Requirements and Technical Concepts for Aviation. DO-178B: Software Considerations in Airborne Systems and Equipment Certification[S]. 1992.
- [13] SAE ARP4754. Certification Considerations for Highly-Integrated or Complex Aircraft Systems[S]. 1996.
- [14] 施太平,娄莉,田泽. AFDX 协议及关键技术的实现[J].测控技术,2012,10:81-84.
- [15] 陈昕,周拥军,万剑雄.AFDX 端系统关键技术的研究与实现[J].计算机工程.2009, 35(5):1-3.
- [16] 许燕婷.AFDX 端系统协议栈虚拟链路层分析及仿真研究[D].上海:上海交通大学,2011.
- [17] 尚伟林. AFDX Gateway 软件的设计与实现[D].北京:北京邮电大学,2010.
- [18] 李大鹏,赵明莉,田园.AFDX 端系统中冗余管理的研究和实现[J].电子技术.2012,

12:10-12+16.

- [19] 陈昕,路娟,周拥军.AFDX 冗余管理机制的仿真[J].计算机工程.2008, 34(23):92-94.
- [20] 萧伟,冯治宝,应启夏.改进型 EDF 调度算法的研究与实现[J].计算机工程.2009, 35(18):231-233.
- [21] 赵永库,李贞,唐来胜.AFDX 网络协议研究.计算机测量与控制[J].2012, 20(1):8-10+30.
- [22] 罗云林,张巨联,杨建忠.基于马尔可夫方法的飞控系统安全性评估[J].中国民航大学学报.2012,29(4):17-19.
- [23] Jean-Yves, Le Boudec. Network calculus: a theory of deterministic queuing systems for the Internet [M]. Germany: Springer-Verlag Berlin Heidelberg,2001
- [24] R. Cruz. A calculus for network delay, Part I: Network element in isolation [J]. Information Theory, IEEE Transactions on.1991, 37(1):114-131.
- [25] R. Cruz. A calculus for network delay, Part II: Network analysis [J]. Information Theory, IEEE Transactions on.1991, 37(1):132-141.
- [26] R. Alur and D. L. Dill. A theory of timed automata[J]. Theoretical Computer Science. 1994, 126(2):183-235.
- [27] Husein Charara, Jean-Luc. Methods for bounding end-to-end delays on an AFDX network[C]. IEEE Proceeding of the 18th Euromicro Conference on Real-Timi System, 2006.
- [28] Henri BAUER. Jean-Luc, Applying Trajectory approach to AFDX avionics network[C]. IEEE Conference on Emerging Technologies & Factory Automation, 2009.
- [29] Henri BAUER, Jean-Luc. Worst-case end-to-end delay analysis of an avionics AFDX network[C]. Design, Automation & Test in Europe Conference & Exhibition (DATE), 2010.
- [30] 李明辉.基于网络演算的网络建模方法研究[D].成都: 西南交通大学,2007.
- [31] 陈昕,向旭东,张磊,徐彤.网络演算理论及其在分组交换网中的应用[J].北京信息科技大学学报.2011, 26(1):11-16.
- [32] 胡飞.基于 AFDX 航空网络的端到端延时分析[D].上海:上海交通大学,2012.
- [33] 李焕忠.基于随机网络演算的性能分析技术研究[D].湖南:国防科技大学,2011.
- [34] Marc Boyer, Christian Fraboul. Tightening end to end delay upper bound for AFDX network calculus with rate latency FIFO servers using network calculus [C]. Factory Communication Systems, 2008. WFCS 2008.
- [35] 赵一 A,张中荃.基于统计网络演算的融合网络网关节点时延性能研究[J].微型机与应用.2011, 30(19):47-50+53.
- [36] 漆华妹,陈志刚. 基于统计网络演算的无线 mesh 网络流量模型[J].通信学报.2009, 30(7):1-6.

- [37] M. Boyer, N. Navet, and M. Fumey. Experimental assessment of timing verification techniques for AFDX[C]. In Pro. of the 6th Int. Congress on Embedded Real Time Software and Systems, Toulouse, France, February 2012.
- [38] Cheng Liu, Tong Wang, Changxiao Zhao, Huagang Xiong. Worst-case flow model of VL for worst-case delay analysis of AFDX [J]. ELECTRONICS LETTERS. 2012, 48(6): 327-328.
- [39] 王威振. AFDX 网络中关键任务端到端延时优化[D]. 上海:上海交通大学, 2013.
- [40] 李庆华, 陈志刚, 赵明. 基于网络演算的无线自组网端到端延迟上界研究[J]. 系统仿真学报, 2009, 21(8): 22-48.
- [41] 钟杰, 何民, 王怀胜, 郑力. AFDX 构架及协议分析[J]. 电讯技术, 2010, 50(1): 65-67.
- [42] Steven Martin, Pascale Minet. Schedulability analysis of flows scheduled with FIFO: Application to the Expedited Forwarding class [J]. 2006.
- [43] 胡光宇. 基于 AFDX 航空网络的端到端延时分析[D]. 上海:上海交通大学, 2012.
- [44] 赵永库, 王红春, 唐来胜. AFDX 网络端到端时延分析方法[J]. 电光与控制, 2013, 20(4): 81-83.
- [45] Xia Jiqiang, Yuan Weimin. Study on Real-time Performance of AFDX Using OPNET[C]. Control, Automation and Systems Engineering (CASE), 2011 International Conference on.
- [46] Ding Lina, Song Dong, Zeng Xingxing and Hu Qiong, "The research of AFDX system simulation model," Multimedia Technology(ICMT), 2010 International Conference on, Oct. 2010, pp. 1, doi:10.1109/ICMULT.2010.5632133.
- [47] H. Charara, J.-L. Scharbarg, J. Ermont, and C. Fraboul, "Methods for bounding end-to-end delays on an AFDX network," In Proceedings of the 18th ECRTS, Dresde, Germany, July 2006.
- [48] 杨洋. 综合模块化航空电子系统资源配置工具的研究与实现[D]. 南京:南京航空航天大学, 2012.
- [49] 冯煜. 基于 GEF 的状态图和活动图建模工具的设计与实现[D]. 西安:西安电子科技大学, 2011.
- [50] 强锋科技, 那静. Eclipse SWT/JFace 核心应用[M]. 北京:清华大学出版社, 2007: 06-10.
- [51] Hendy Irawan, Martin taal. Teneo/Hibernate[EB/OL]. (2012-11-02)[2014-12-19]. <http://wiki.eclipse.org/Teneo/Hibernate>.
- [52] Martin taal. Teneo/Hibernate/Configuration Options [EB/OL]. (2013-11-17)[2014-12-19]. http://wiki.eclipse.org/Teneo/Hibernate/Configuration_Options.
- [53] Apache. Apache CXF: An Open-Source Services Framework [EB/OL]. <http://cxf.apache.org>.
- [54] 刘成, 周立, 屠晓杰, 王彤. 虚拟链路精确流在 AFDX 网络演算中的应用[J]. 电光与控制, 2013, 20(10): 07-10.

- [55] F.Frances, C. Fraboul, J. Grieu. Using Network Calculus to optimize the AFDX network [J].ERTS, 2006:25-27.
- [56] Actel Corporation. Developing AFDX Solutions [J/OL]. (2005)[2014-10-15].
<http://search.microsemi.com/Reports/AdvancedWebSearch.aspx>.

致 谢

经过不懈的努力，终于顺利完成了毕业论文，论文结束之际，也意味着自己即将结束 3 年的研究生学习，开始步入社会，走向工作岗位。在撰写毕业论文这段充实而又紧张过程中自己也收获了很多。

首先要感谢国家的培育，母校的教育。正因为有了国家，我们才能安然学习，进步。也正因为有了母校和母校的提供的条件，才使我不断成长。

其次要感谢我的导师王立松、谷青范老师，论文能够顺利完成，与他们的辛勤指导是密不可分的，从我一进入实验室，王老师不仅在生活中给予了我很多帮助而且在学习、科研中更是不断教诲我，我会永远感念于心。在我读研期间，王老师提供很多机会指导我参与了多个项目，教会我前沿的计算机技术，最重要的是教会我许多思考问题的方法。同时特别钦佩王老师在日常工作中负责专注的态度和渊博的知识储备。在我 3 年的研究生生涯里，科研研究中他循循善诱，项目实践，他耐心教导，让我迅速成长。而实验室的另一位谷老师一直是我学习的榜样，同样要感谢他在论文中的帮助。在此学生由衷地向你们表示感谢。

同时，还要感谢张育平老师。在张老师的理论课和工程实践课中都尝到了许多宝贵知识。他对知识的孜孜不倦，工作中的热情执着以及教学中的耐心细致，让我钦佩，受益良多，感谢张老师的帮助。

其次，感谢 510 与 520 的师兄师姐们！我忘不了一起相处的快乐时光，一起科研的激情岁月，忘不了师兄师姐们在生活中给予我的帮助和鼓励，忘不了他们在学习和项目上给予我的指导。虽然我们都各自去了不同城市工作，但友谊却是地久天长的！杨洋、付佩、许文平师兄还有章倩师姐。

然后，感谢父母对我的养育之恩。一直以来，父母都一直默默地支持我。在我面对困难挫折时，他们会给予我鼓励，给我加油打气。还有我一位挚友孔凡月，她时时刻刻给予我的关心和爱护都让我备感温馨的同时也敦促我继续努力，谢谢你。

最后，由衷地感谢各位专家、老师在百忙之中能够抽出时间对我的论文进行评审并提出宝贵意见。谢谢！

在学期间的科研成果及发表的学术论文

攻读硕士学位期间发表(录用)论文情况

1. 孙权,王立松,邢逆舟.基于多链路分组技术的 AFDX 网络延迟研究. 计算机与现代化. 2015 第一期.
2. Lisong Wang,Quan Sun,Lei Zhang,Wansheng Miao. A Improved Schnyder Routing Algorithm for Node Failure in Wireless Sensor Networks. Journal of Information & Computational Science.2015.

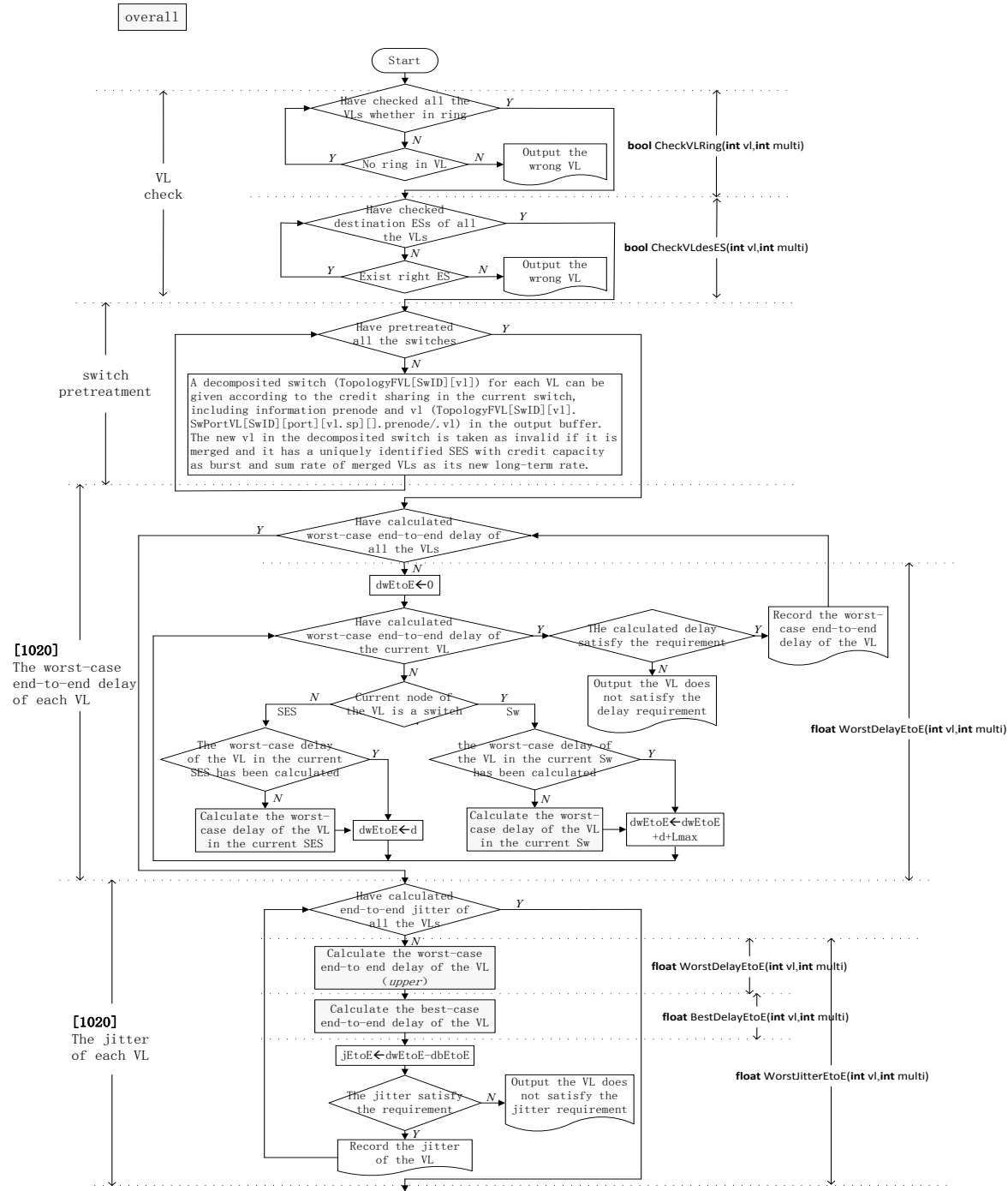
其它成果

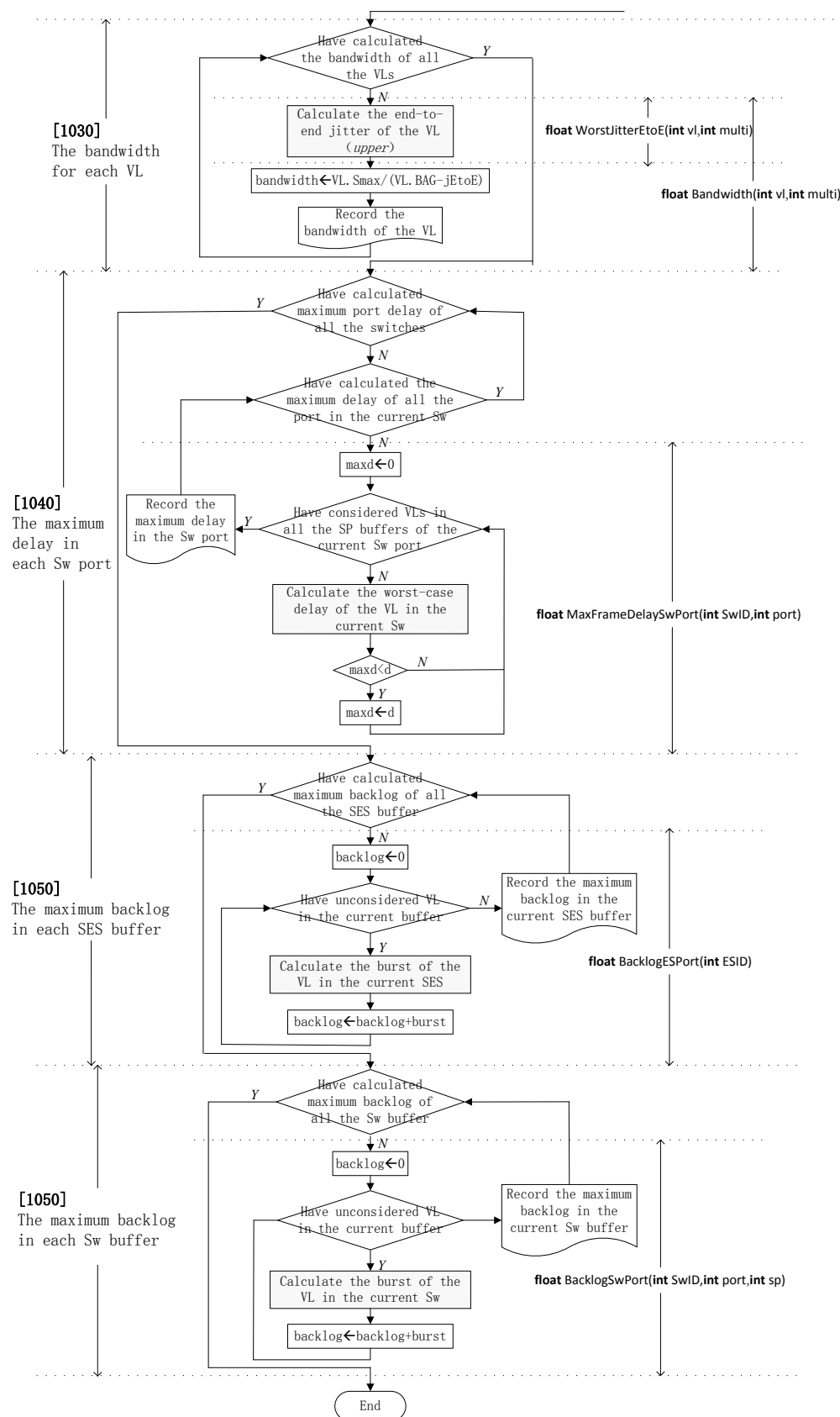
1. 软件著作权: 中小型企业工资管理系统 V1.0

攻读硕士学位期间参加科研项目情况

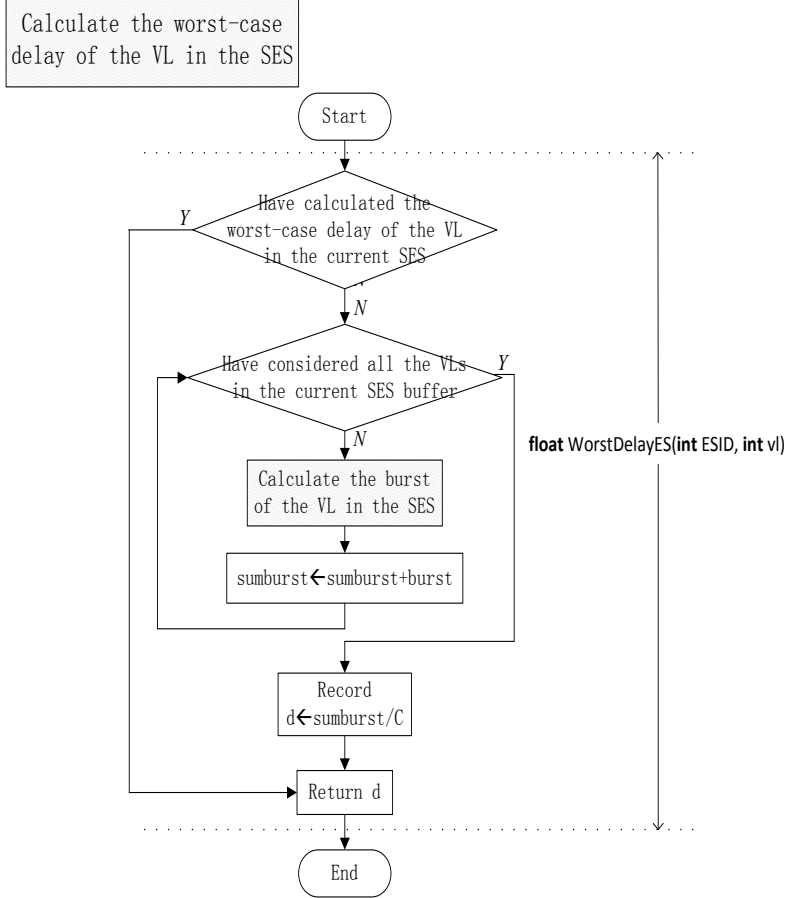
1. 2012.09 — 2013.02 支持 ARINC661 的通用网络协议监视与分析工具
2. 2013.04 — 2014.08 综合化航空电子接口控制文档配置工具
3. 国家重点基础研究发展计划(973 计划),项目编号: 2014CB744900,项目名称: 面向大型飞机综合化航空电子系统安全性基础问题研究

附录

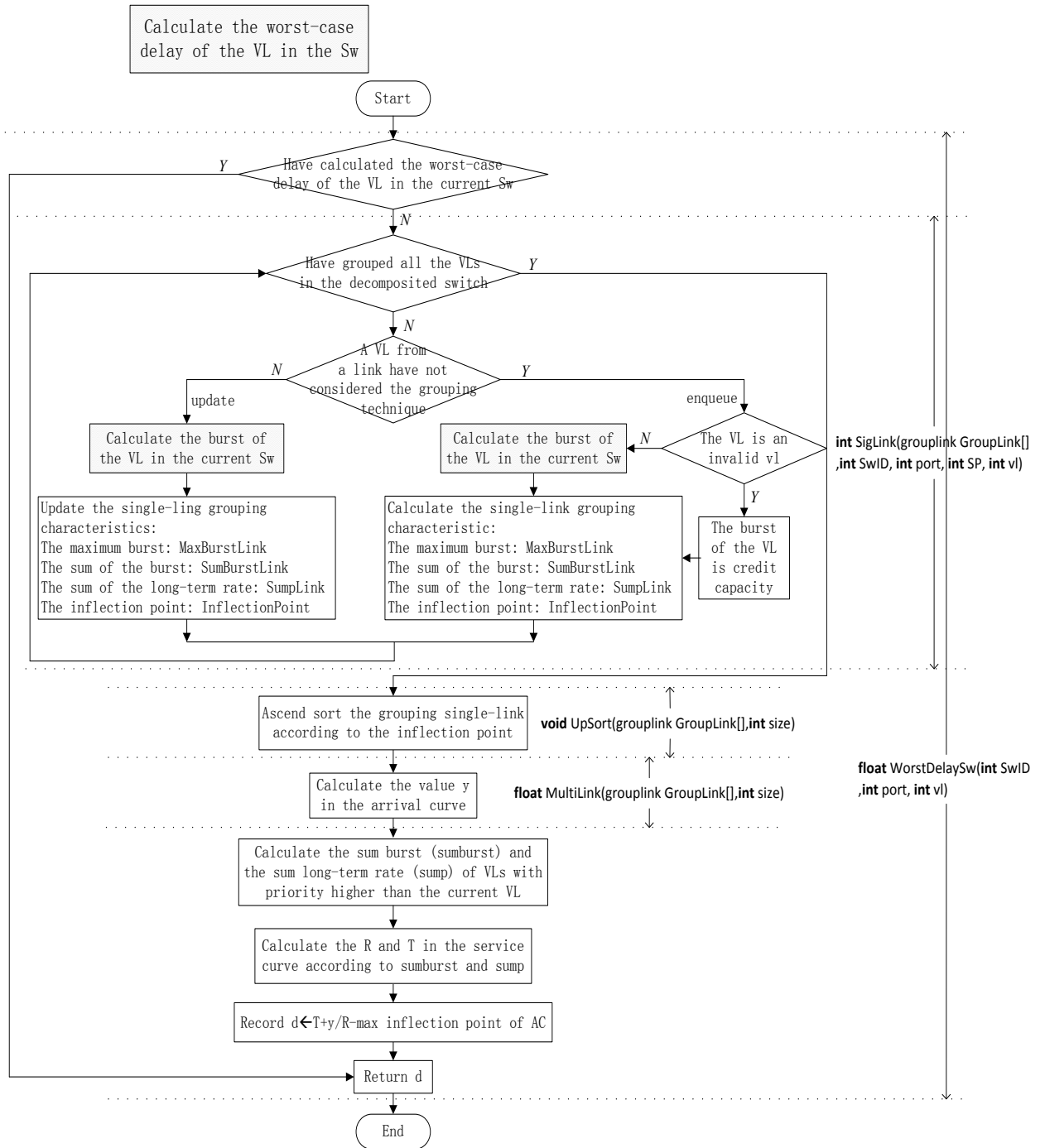




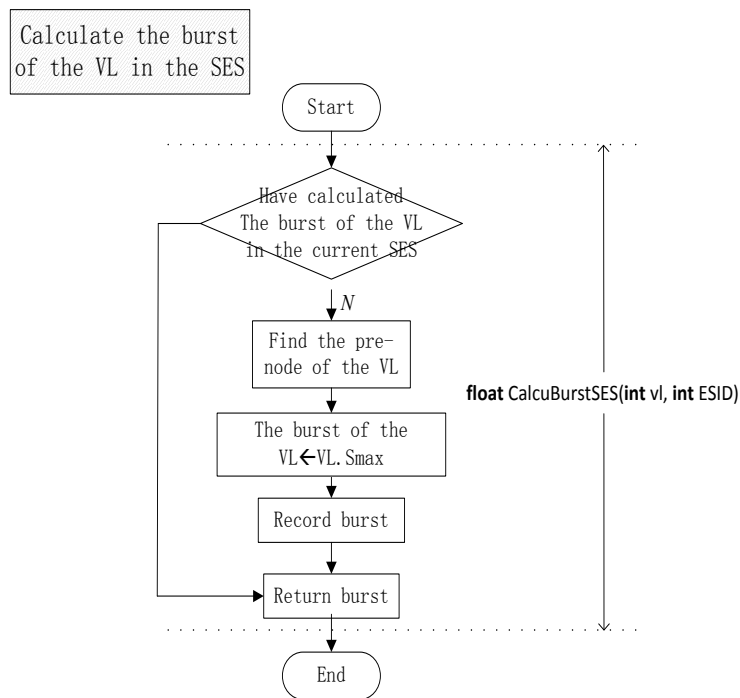
B. 源端延迟



C. 交换机中延迟



D. 源端虚拟链路突发流量



E. 交换机中虚拟链路突发流量

