

时间触发 AFDX 调度设计及实时性分析

焦文喆¹, 翟正军¹, 王国庆^{1,2}

(1. 西北工业大学 计算机学院, 西安 710129; 2. 中国航空无线电电子研究所, 上海 200233)

摘 要: 针对时间触发航空电子全双工交换式以太网 (TTAFDX) 的特点, 设计基于帧长度优先原则的系统调度策略, 解决时间触发虚拟链路的调度问题。搭建典型网络, 采用网络演算方法对 TTAFDX 和使用静态优先级调度策略的 AFDX 进行实时性比较, 结果表明, TTAFDX 中时间触发虚拟链路的延迟主要由固定延迟部分组成, 而速率限制虚拟链路的实时性较 AFDX 中的低优先级虚拟链路也有所提高, 证明 TTAFDX 调度策略在兼容 AFDX 的同时能够改善网络的时间确定性, 适用于具有硬实时传输要求的航空电子系统。

关键词: 分布式综合模块化航空电子; 航空电子全双工交换式以太网; 时间触发; 调度策略; 实时性分析

中文引用格式: 焦文喆, 翟正军, 王国庆. 时间触发 AFDX 调度设计及实时性分析[J]. 计算机工程, 2016, 42(7): 42-48.

英文引用格式: Jiao Wenzhe, Zhai Zhengjun, Wang Guoqing. Scheduling Design and Instantaneity Analysis of Time-triggered AFDX[J]. Computer Engineering, 2016, 42(7): 42-48.

Scheduling Design and Instantaneity Analysis of Time-triggered AFDX

JIAO Wenzhe¹, ZHAI Zhengjun¹, WANG Guoqing^{1,2}

(1. School of Computer Science, Northwestern Polytechnical University, Xi'an 710129, China;

2. China National Aeronautical Radio Electronics Research Institute, Shanghai, 200233, China)

【Abstract】 According to the characteristics of Time-triggered Avionics Full Duplex Switched Ethernet (TTAFDX), this paper proposes a scheduling strategy based on frame-length priority to solve the scheduling problem of Time-triggered Virtual Link (TTVL). It builds a typical network, and compares the instantaneities of TTAFDX and AFDX which uses Static Priority (SP) scheduling strategy. The result shows that, the delay of TTVL consists mainly of fixed delay, and the instantaneity of Rate Constraint Virtual Link (RCVL) is also improved compared with the low-priority VL in AFDX, which demonstrates that the scheduling strategy of TTAFDX can enhance the time-deterministic feature of the network while being compatible with AFDX. It can be implemented in the avionics system with hard real-time transmission requirement.

【Key words】 Distributed Integrated Modular Avionics (DIMA); Avionics Full Duplex Switched Ethernet (AFDX); time-triggered; scheduling strategy; instantaneity analysis

DOI: 10.3969/j.issn.1000-3428.2016.07.007

1 概述

综合模块化航空电子 (Integrated Modular Avionics, IMA) 通过采用开放式体系结构和标准化以及通用化的设计, 提高了系统的兼容性、可移植性、可扩展性和可维护性^[1]。分布式综合模块化航空电子 (Distributed IMA, DIMA) 在继承 IMA 结构的系统综合化、网络一体化、产品商用化、功能软件化等特点的同时^[2], 改变了 IMA 中将所有计算资源集中在一个功能区的做法, 采用联合式结构中将综合化模块电子

(Distributed Modular Electronics, DME) 分布在靠近信号源 (作动器和传感器) 的区域, 且与前端信号源的预处理相结合的方式, 有效地解决了 IMA 结构中系统复杂度、机柜尺寸、布线及散热等问题^[3-4]。DIMA 通过分布式系统架构和高容错的实时通信网络, 将分布于整个飞机范围的多个 DME 相连接, 不仅为各子系统提供了故障隔离, 同时也提高了整个航空电子系统的资源综合与信息共享的程度。

已成功应用于空客 A380 和波音 B787 的航空电子全双工交换式以太网 (Avionics Full Duplex

基金项目: 国家“973”计划基金资助项目 (2014CB744900)。

作者简介: 焦文喆 (1985 -), 男, 博士研究生, 研究方向为航电网络、网络存储安全; 翟正军、王国庆, 教授、博士、博士生导师。

收稿日期: 2015-11-03 修回日期: 2015-12-10 E-mail: wenzhejiao@163.com

Switched Ethernet, AFDX)^[5-6]虽然在 IEEE802.3 基础上引入虚拟链路 (Virtual Link, VL)、静态路由和流量管制等机制,在一定程度上提高了网络数据传输的实时性和可靠性,但当多任务传输时可能产生的竞争链路共享冲突,仍会影响数据传输的实时性和通信系统的时间确定性。具备 100 Mb/s 固定传输速率的 AFDX 允许的抖动最大值为 500 μ s,对于时间关键消息,其传输时延及时延抖动的确定性界限仍不易得到保证^[7-8]。为了提高 DIMA 架构中时间关键消息的时间确定性,并尽可能降低既有设备改造的技术与商业风险,刘成等人将时间触发机制引入 AFDX,提出了时间触发 AFDX (Time-triggered AFDX, TTAFFDX) 体系结构^[9]。

本文根据 TTAFFDX 的特点,提出基于帧长度优先原则的端系统调度策略,实现降低设计复杂度并提高网络带宽利用率的目的。通过网络演算的方法分别对 TTAFFDX 及 AFDX 网络中各类虚拟链路的时延进行分析和比较,验证 TTAFFDX 网络适用于具有时间确定性要求的数据传输。

2 时间触发虚拟链路端系统调度策略

TTAFFDX 网络按照所支持的流量类型定义了 2 类虚拟链路:时间触发 VL (Time-triggered VL, TTVL) 和速率限制 VL (Rate-constrained VL, RCVL)^[9]。TTVL 用来承载时间关键消息以确保其时间确定性和实时性。TTVL 的发送活动严格按照时刻调度表的规划时刻来触发,在全局时钟精确同步的条件下,可以避免数据在链路上发生共享冲突。RCVL 承载了 AFDX 网络中的 VL,优先级低于 TTVL。TTAFFDX 端系统虚拟链路调度过程如图 1 所示。

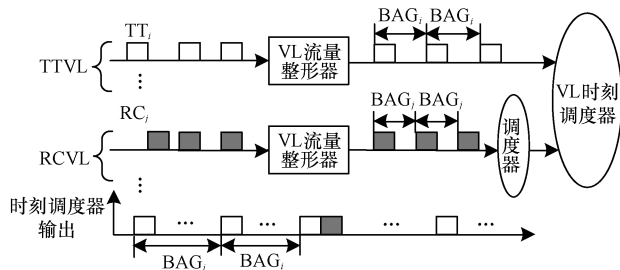


图 1 TTAFFDX 端系统虚拟链路调度过程

TTAFFDX 中时间触发调度本质上是一种基于表的静态调度,但不同与 TTCAN 等总线结构的时间触发调度^[10-12]——它们的各个端系统具有相同的调度表,TTAFFDX 网络中每个 TT 帧的发送端都存有一个 TT 帧发送时刻调度表。

VL 时刻调度表包含一个矩阵周期 (Matrix Cycle, MC),它由若干个基本周期 (Basic Cycle, BC) 组成^[13]。其中,MC 为时间触发虚拟链路数据帧调度

的任务周期;BC 为时间触发虚拟链路数据帧传输的最小周期。对于系统中的 N 个周期消息,取最小周期作为 BC。为了保证每一消息的平均周期,MC 的持续时间必须是所有消息周期的最小公倍数或最小公倍数的整数倍。根据 AFDX 协议中规定的带宽分配间隔 (Bandwidth Allocation Gap, BAG) 取值范围 2^k ms, $k = 0, 1, \dots, 7$,可以设定 VL 时刻调度表的 MC 为所有 BAG 取值的最小公倍数,即 128 ms,BC 为所有 BAG 取值的最大公约数 1 ms。

VL 时刻调度表中的每个 BC 都开始于一个时钟同步数据帧 (SYNC),并且以下一个 SYNC 的开始作为结束。以 BC 为时间单位进行时钟同步可以保障网络中各个节点的时钟完全同步,使得完善的离线设计 VL 时刻调度表能够保证 TT 帧在端系统不会发生争用物理链路的情况。

如图 2 所示,每个 BC 又分为 2 段,前一段专用于发送 TT 流量数据,后一段用于发送 RC 流量数据。在 TT 段中,时间资源又被分为多个时窗 (time window)。由于 TTAFFDX 中 TT 流量的帧长是不固定的,BC 中各个时窗的大小也不确定,但 MC 中同一列的时窗的大小相同,其大小等于在此时窗中传输消息的最大帧长。

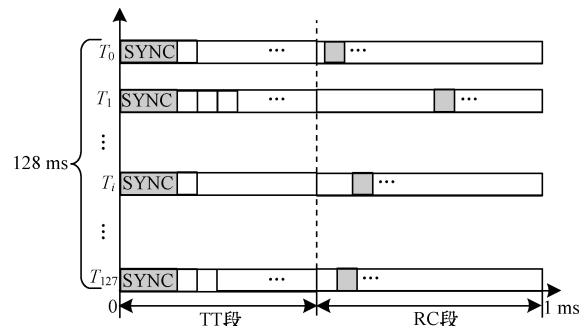


图 2 虚拟链路时刻调度表示意图

2.1 帧长度优先的调度表设计

当同周期 TTVL 的数据帧长存在较大差异时,使用文献[9]中的调度表设计方法会造成时窗的利用率较低、带宽利用不足的问题。帧长度优先调度表设计方法在保证 TT 流量数据无冲突发送的同时,能够有效减少 TT 时窗的时间总和,提高了 TT 时窗的总线利用率,也使得 RC 数据流量有了更多的传输时间,具体设计流程如下:

(1) 按照数据帧长度从大到小的顺序给端系统发送的 N 条 TTVL 排序,若数据帧长度相同,则按照 BAG 从小到大的顺序进一步排序,排序后的 TTVL 对应标记为 VL_k (k 大小为 $1 \sim N$),相应的最大线路帧长记为 L_k ,转到第(2)步。

(2) 按照标记顺序对 TTVL 进行发送时刻规划,初始状态令 $k = 1, L_{sum} = L_{SYNC}, S_k = 0$,其中, L_{sum} 表示

基本周期已配置的帧长和; L_{SYNC} 为 28 Byte; S_k 表示第 k 条 TTVL 数据流首帧发送所在的 BC 周期; C 为网络带宽, 转到第(3)步。

(3) 若 $k > N$, 跳至第(4)步; 否则配置 VL_k 的发送时刻, 虚拟链路 VL_k 的 BAG 值为 G_k , 则在一个 MC 内 VL_k 被循环调度的次数为 $128/G_k$; 从调度表 0 ms 开始, 在 G_k 时间范围中, 依然按照“左紧凑”原则查找可分配的发送时窗。在查找合适时窗的过程中, 当找到一个位于第 a 个 BC 中的空闲时窗时, 若此空闲时窗所处的时窗列为空时, 即 $a = 0$ 时, 则更新 BC 已配置帧长和 $L_{\text{sum}} = L_{\text{sum}} + L_k$, 若 L_{sum} 大于 BC 上最多可以配置的帧长和, 则跳到第(5)步; 否则, 将 VL_k 的 $128/G_k$ 个调度时刻配置为 $t_{k,m} = (m - 1) \times G_k + 8 \times (L_{\text{sum}} - L_k) \times 1000 / (C \times 10^6)$, $m \in \{1, 2, \dots, 128/G_k\}$, 其中 $t_{k,m}$ 表示 VL_k 在 MC 中的第 m 次调度时刻, 赋值当前所在时窗列长度 $L_{\text{TW}} = L_k$, 并将 a 赋值给 S_k ; 若此空闲时窗所处的时窗列不为空, 则需要依次与此时窗列中已规划好的 TTVL 通过 $(a - S_i) \% (\min(G_k, G_i))$, $i < k$ 是否等于 0 来判断是否产生冲突, 其中 S_i, G_i 分别表示所比较的各 TTVL 的首帧发送 BC 周期及其 BAG。若 $(a - S_i) \% (\min(G_k, G_i)) = 0$, 则产生冲突, 重新开始第(3)步; 若 $(a - S_i) \% (\min(G_k, G_i)) \neq 0$, 则 VL_k 的 $128/G_k$ 个调度时刻配置为 $t_{k,m} = (a - 1) + (m - 1) \times G_k + 8 \times (L_{\text{sum}} - L_k) \times 1000 / (C \times 10^6)$, $m \in \{1, 2, \dots, 128/G_k\}$; 最后将 a 赋值给 S_k 。

(4) 在该端系统中, 所有 TTVL 的发送时刻都已规划完成。

(5) 带宽有限, 无法配置该端系统中所有 TTVL 的发送时刻。

2.2 时间触发虚拟链路交换机调度算法设计

对于到达的 TTVL, 交换机首先核对 TTVL 的到达时刻是否在规划的接收时窗内, 将时窗之外的 TTVL 丢弃, 然后对到达的 TTVL 按照漏桶算法 (Leaky Bucket) 进行流量管制, 最后将完成管制的数据帧送到输出端口。对于到达的 RCVL, 交换机首先进行流量管制, 然后完成管制的数据帧将通过调度器按照指定的调度算法进行处理, 最后送到输出端口。在交换机的输出端口中, VL 时刻调度器按照转发时刻调度表的规划来转发 TTVL, 并在未被安排转发 TTVL 的空闲时间段发送 RCVL。

交换机的每一个端口都维护一张 TTVL 的转发时刻调度表。交换机是 TTVL 的中间节点, TTVL 帧从不同的端系统汇聚到交换机的转发端口, 帧的到达时刻不是由交换机端口决定, 所以, 转发表的规划将不采用基本周期组成矩阵周期的方式, 而是直接以时间跨度为 128 ms 的周期进行转发时刻表的

规划。在满足时钟同步方案所需要发送资源开销的基础上, 当所有 VL 的静态路由和所有 TTVL 的发送时刻配置完成后, 交换机通过如下步骤实现所有交换机内部的所有端口的转发时刻调度表配置。

(1) 根据所采用的不同端系统调度表设计方法, 对 TTVL 的预处理过程有所不同。当端系统采用基于周期优先的调度表设计时, 按照 BAG 从小到大的顺序给端系统发送的 N 条 TTVL 排序, 若 BAG 相同则按照数据帧长度由从大到小的顺序排序, 排序后的 TTVL 对应标记为 VL_k (k 大小为 $1 \sim N$), 相应的最大线路帧长记为 L_k , 转到第(2)步。

(2) 按照标记顺序对 TTVL 在交换机中的转发时刻进行规划, 初始状态令 $k = 1$, 转到第(3)步。

(3) 若 $k > N$, 则跳至第(4)步; 否则按照 VL_k 所经过交换机转发端口的顺序规划其在这些端口的转发时刻。TT 虚拟链路 VL_k 的 BAG 值为 G_k , 则在一个调度周期中都将转发 $128/G_k$ 帧该 TT 数据, 以 $V_{k,i}$ ($1 \leq i \leq 128/G_k$) 依次标识各个帧。规划 VL_k 在其转发路径中的任一端口的转发时刻时, 按照帧 $V_{k,1}$ 到帧 $V_{k,w}$ ($w = 128/G_k$) 的顺序规划各帧数据的转发时刻。根据帧 $V_{k,i}$ ($1 \leq i \leq 128/G_k$) 在上一节点的发送时刻 t_1 通过式(1)能够计算出其到达此交换机转发端口的可能最晚时刻, 即接收窗口的关闭时刻 t_2 :

$$t_2 = t_1 + \frac{8 \times L_k \times 1000}{C \times 10^6} + D_{\text{switch}} + 2 \times \Delta T \quad (1)$$

其中, D_{switch} 为交换机时延; ΔT 为系统所允许的最大时钟漂移。

在不改变已经规划好的 TTVL 帧的原则下, 按照 $0 \leq \Delta t \leq 128$ 从小到大的方向, 在一个调度周期中寻找最近的能够完成转发任务的空闲时间段, 若能找到, 则将该空闲时间段的时刻起点 $t_3 = (t_2 + \Delta t) \% 128$ 作为转发时刻, 更新交换机端口转发时刻表; 若找不到, 则跳至第 5 步。若 $V_{k,i}$ ($1 \leq i \leq 128/G_k$) 在所经过的交换机转发端口都能完成转发时刻的规划, 则 $k++$, 循环第(3)步, 否则跳至第(5)步。

(4) 所有 TTVL 在交换机端口中的转发时刻都规划完成。

(5) 网络带宽有限, 无法配置所有 TTVL 在 TTAFFDX 网络中的转发时刻。

3 TTAFFDX 网络实时性分析

端到端时延是 TTAFFDX 网络性能的一个重要评价指标, 主要是指数据从网络中的源端节点传输到目的端节点所花费的时间, 包括链路传输时延 $D_{\text{propagation}}$ 、源端系统时延 D_{tx} 、交换机时延 D_{switch} 和目的端系统时延 D_{rx} , 则 TTAFFDX 数据帧的网络时延

定义可表示如下:

$$D = D_{\text{propagation}} + D_{\text{tx},k} + \sum_{i=0}^j D_{\text{switch},k} + D_{\text{rx},k} \quad (2)$$

其中, $D_{\text{propagation}}$ 是数据帧在物理链路上传输所引起的时延, 由于物理介质上传播速度是固定的, 因此这一时延具有确定值; 发送端的时延 $D_{\text{tx},k}$ 可用式 (3) 定义:

$$D_{\text{tx},k} = G_k + \tau_{\text{tech}} + \tau_{\text{frame},k} + EJ_{\text{max},k} \quad (3)$$

其中, G_k 为虚拟链路 VL_k 的带宽分配间隔; τ_{tech} 为源端系统的技术时延, ARINC 664 协议中规定端系统的技术延时不超过 $150 \mu\text{s}$; $\tau_{\text{frame},k}$ 是虚拟链路 VL_k 将数据帧发送至物理链路层所造成的延迟; $EJ_{\text{max},k}$ 是由复用调度策略导致的数据队列等待时延; $D_{\text{rx},k}$ 主要为技术延时。

TTAFDX 数据帧在交换机中将经过数据帧过滤、数据帧转发和数据帧发送 3 个步骤, 其时延计算公式如下:

$$D_{\text{switch},i} = \tau_{\text{filter}} + \tau_{\text{fw}} + \tau_{\text{rv}} + SJ_{\text{max},i} \quad (4)$$

其中, τ_{filter} 为数据帧过滤时延, AFDX 协议规定每个交换机端口必须能及时过滤以线速到达的任意大小的有效数据帧; τ_{fw} 为数据帧转发时延, 通常设为 $8 \mu\text{s}$; τ_{rv} 是交换机接收端口在接收数据帧时产生的时延, 由各虚拟链路所传输的最大帧长度决定; $SJ_{\text{max},i}$ 为交换机系统多路复用时延。

3.1 TTAFDX 网络模型

TTAFDX 采用与传统 AFDX 相同的星型拓扑网络结构, 具有转发速度快、时延小、实时性好、能够降低网络负荷等特点。本文采用的网络模型如图 3 所示, 模型中包括 3 个交换机 (SW_1, SW_2, SW_3)、8 个端系统 (ES_1, ES_2, \dots, ES_8) 和 12 条虚拟链路 ($VL_1, VL_2, \dots, VL_{12}$), 各个虚拟链路均为单播虚拟链路。且相邻节点间的距离为 100 m , 所使用物理链路的传输速率为 $2 \times 10^8 \text{ m/s}$ 。

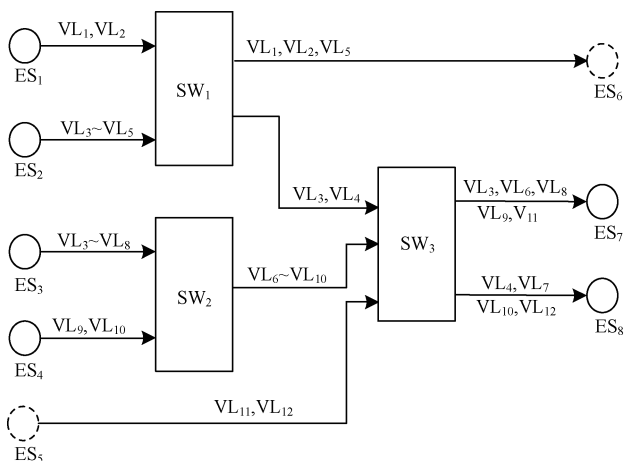


图 3 TTAFDX 网络模型

表 1 列出了图 3 中 12 条虚拟链路的相关参数, 包括带宽分配间隔 (BAG)、最大帧长 L_{max} 以及虚拟链路类型。

表 1 时间触发 AFDX 网络模型虚拟链路配置

虚拟链路 ID	L_{max} /Byte	BAG/ms	类型
VL ₁	512	16	时间触发
VL ₂	256	8	速率限制
VL ₃	128	32	时间触发
VL ₄	256	64	时间触发
VL ₅	1 024	32	速率限制
VL ₆	512	32	时间触发
VL ₇	256	32	时间触发
VL ₈	512	64	时间触发
VL ₉	128	128	速率限制
VL ₁₀	128	4	速率限制
VL ₁₁	1 024	16	时间触发
VL ₁₂	64	64	速率限制

3.2 TTAFDX 时间触发虚拟链路实时性分析

以时间触发虚拟链路 VL_3 为例进行分析, VL_3 和 VL_4 由端系统 ES_2 发送至交换机 SW_1 , 所以, 在端系统 ES_2 需要对 VL_3 和 VL_4 设计时刻调度表, 按照周期优先调度表设计方法可得 ES_2 的时刻调度表, 如图 4 所示。

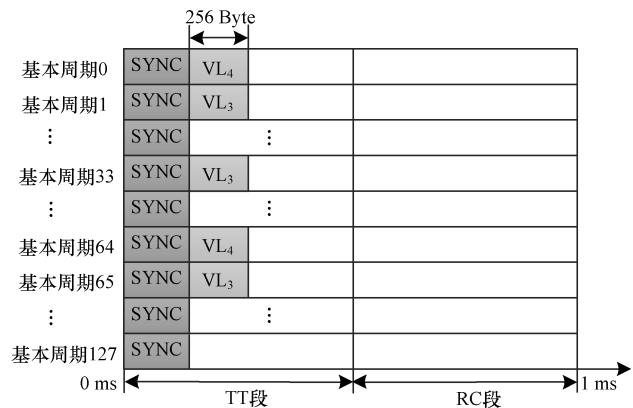


图 4 端系统 ES_2 调度时刻表

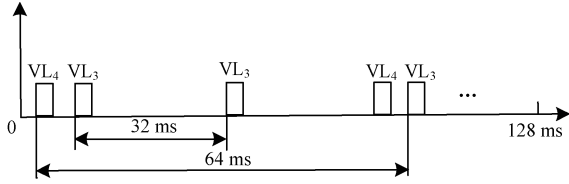
在端系统 ES_3, VL_6 首次被调度的时刻 $V_{4,1}, t_{4,1}$

$$= \frac{L_{\text{SYNC}} \times 8 \times 1\,000}{100 \times 10^6} = \frac{28 \times 8 \times 1\,000}{100 \times 10^6} = 0.002\,24 \text{ ms},$$

其他数据帧被调度的时刻根据设计方法第 (3) 步中所述的不同情况选用相应的计算公式, 所得统计结果如表 2 所示。 VL_3 和 VL_4 两条 TTVL 到达交换机 SW_1 之后将转发至交换机 SW_3 , 由交换机调度表设计算法可得 $SW_1 \sim SW_3$ 的时刻调度表, 如图 5 所示。

表2 TTVL₄在ES₂中的调度时刻 ms

TTVL 帧	调度时刻
V _{4,1}	0.002 24
V _{4,2}	64.002 24

图5 交换机SW₁对虚拟链路VL₃和VL₄的时刻调度表

依照式(4)可得VL₄在交换机SW₁中第一次被调度的时刻为:

$$t_{4,1} = \frac{L_{\text{SYNC}} \times 8 \times 1\,000}{100 \times 10^6} + 2 \times \frac{L_{\text{VL}_4} \times 8 \times 1\,000}{100 \times 10^6} + \tau_{\text{filter}} + \tau_{\text{fw}} + D_{\text{propagation}} = 0.059\,7\text{ ms}$$

依次对VL₄一个矩阵周期中的2个数据帧进行类似计算,所得统计结果如表3所示。

表3 TTVL₄在SW₁中的调度时刻 ms

TTVL 帧	调度时刻
V _{4,1}	0.059 7
V _{4,2}	64.059 7

同理可得在交换机SW₃中VL₄第一次被调度的转发时刻为:

$$t_{4,1} = \frac{L_{\text{SYNC}} \times 8 \times 1\,000}{100 \times 10^6} + 4 \times \frac{L_{\text{VL}_4} \times 8 \times 1\,000}{100 \times 10^6} + 2 \times (\tau_{\text{filter}} + \tau_{\text{fw}} + D_{\text{propagation}}) = 0.117\,16\text{ ms}$$

依次对VL₄一个矩阵周期中的2个数据帧进行类似计算,所得统计结果如表4所示。

表4 TTVL₄在SW₃中的调度时刻 ms

TTVL 帧	调度时刻
V _{4,1}	0.117 16
V _{4,2}	64.117 16

由以上所得到的网络中各个节点的调度表能够计算出各个TTVL的端到端延迟,如表5所示。

表5 TTAFFDX中时间触发虚拟链路端到端时延

TTVL 帧	时延上界/ μs	TTVL 帧	时延上界/ μs
VL ₁	139.88	VL ₇	135.90
VL ₃	84.70	VL ₈	238.30
VL ₄	135.90	VL ₁₁	262.76
VL ₆	238.30		

表5中各时间触发虚拟链路端到端时延仅包含固定时延部分,时延抖动均为0ms,能够满足具有硬实时要求的数据传输。

3.3 TTAFFDX速率限制虚拟链路实时性分析

当混合数据流到达交换机的某个输出端口时,交换机按照配置时间对TT数据流进行转发,TT数据流在配置时域内独占整个带宽,不受RC数据流的影响。当TT数据流完成传输后,调度器会对所在端口的RC数据流执行发送。当有多个RC数据流等待发送时,调度器按照FIFO顺序,依次对RC数据流进行操作。对于共享TTAFFDX交换机同一输出端口的多个VL,定义 $G^{\text{TT}} = \cup \text{TTVL}_i$ 和 $G^{\text{RC}} = \cup \text{RCVL}_i$ 分别代表TTVL和RCVL的聚合流。

由于TT数据流不会受到RC数据流的阻塞,因此输出端口提供给TTVL的服务曲线即交换机输出端口能够提供的总服务曲线,当其物理带宽为C时,则其为: $\beta_{G^{\text{TT}}}(t) = \beta(t) = C[t - 0]^+ [14]$ 。

而对于RCVL聚合流 G^{RC} 的服务曲线为: $\beta_{G^{\text{RC}}}(t) = [\beta(t) - A^{\text{TT}}(t)]^+$,假设VL的到达曲线为 $\alpha(t) = rt + b$,则 G^{TT} 的到达曲线 $A^{\text{TT}}(t)$ 为: $A^{\text{TT}}(t) = \sum_{i \in G^{\text{TT}}} \alpha_i(t) = \sum_{i \in G^{\text{TT}}} r_i \times t + \sum_{i \in G^{\text{TT}}} b_i$ 。

综上所述,可推导出式(5):

$$\beta_{G^{\text{RC}}}(t) = [Ct - \sum_{i \in G^{\text{TT}}} r_i \times t - \sum_{i \in G^{\text{TT}}} b_i]^+ = [C - \sum_{i \in G^{\text{TT}}} r_i] \left[t - \frac{\sum_{i \in G^{\text{TT}}} b_i}{C - \sum_{i \in G^{\text{TT}}} r_i} \right]^+ \quad (5)$$

记RCVL聚合流 G^{RC} 的服务速率和服务时延参数分别为 $R_{G^{\text{RC}}} = C - \sum_{i \in G^{\text{TT}}} r_i$ 和 $T_{G^{\text{RC}}} = \sum_{i \in G^{\text{TT}}} b_i / R_{G^{\text{RC}}}$ 。

对于RCVL聚合流 G^{RC} 的组成虚拟链路VL_i,其在 G^{RC} 中仍以FIFO调度策略接受交换机的服务,按照FIFO调度策略^[15]的相关推导,可得:

$$\beta_{\text{VL}_i}(t) = \left[R_{G^{\text{RC}}}(t - T_{G^{\text{RC}}}) - \left[\sum_{j \in G^{\text{RC}}_i} r_j - r_i \right] \times \left[t - T_{G^{\text{RC}}} - \frac{\sum_{j \in G^{\text{RC}}_i} b_j - b_i}{R_{G^{\text{RC}}}} \right] - \left[\sum_{j \in G^{\text{RC}}_i} b_j - b_i \right] \right]^+ = \left[R_{G^{\text{RC}}} - \sum_{j \in G^{\text{RC}}_i} r_j + r_i \right] \times \left[t - \frac{R_{G^{\text{RC}}} \times T_{G^{\text{RC}}} + \sum_{j \in G^{\text{RC}}_i} b_j - b_i}{R_{G^{\text{RC}}}} \right]^+ \quad (6)$$

根据表2中所设定的各虚拟链路参数,经过计算,得到5条速率限制虚拟链路的延迟上界,如表6所示。

表 6 TTAFDX 中速率限制虚拟链路端到端时延上界

RCVL 帧	时延上界/ μs	RCVL 帧	时延上界/ μs
VL ₂	201.74	VL ₁₀	243.80
VL ₅	324.78	VL ₁₂	119.48
VL ₉	373.30		

4 基于非抢占 SP 调度的 AFDX 实时性分析

非抢占静态优先级 (Static Priority, SP) 调度策略下先将不同类型的数据按照紧急程度划分不同的优先级, 高优先级的数据优先被调度, 高优先级队列为空时查询下一优先级队列进行服务。根据普通 AFDX 系统能够支持两级优先级的特点, 将图 3 网络拓扑结构中的 TTVL 设为高优先级, 其余 VL 为低优先级。

本文使用 $p(i)$ 代表各个 VL 的优先级, 其中高优先级时 $p(i) = H$, 而低优先级时 $p(i) = L$ 。对于共享交换机同一输出端口的多个 VL, 定义 $G^H = \bigcup_{\{p(i)=H\}} VL_i$ 和 $G^L = \bigcup_{\{p(i)=L\}} VL_i$ 分别代表高优先级和低优先级虚拟链路的聚合流, $l_{\max}^L = \max \{L_{\max}^i : p(i) = L\}$ 为低优先级 VL 的最长数据帧长度。假设 VL 的到达曲线为 $\alpha(t) = rt + b$, 则 $A^L(t) = \sum_{\{p(i)=L\}} \alpha_i(t)$ 和 $A^H(t) = \sum_{\{p(i)=H\}} \alpha_i(t)$ 分别为 G^L 和 G^H 的到达曲线。

当输出端口物理带宽为 C , VL 聚合流 G^H 的服务曲线为: $\beta_{G^H}(t) = [\beta(t) - l_{\max}^L]^+ = C[t - l_{\max}^L/C]^+$, 记 VL 聚合流 G^L 的服务速率和服务时延参数分别为: $R_{G^L} = C, T_{G^L} = l_{\max}^L/C$ 。而对于 VL 聚合流 G^L 的服务曲线为: $\beta_{G^L}(t) = [\beta(t) - A^H(t)]^+$, 其中, $A^H(t) = \sum_{\{p(i)=H\}} \alpha_i(t) = \sum_{\{p(i)=H\}} r_i \times t + \sum_{\{p(i)=H\}} b_i$ 。综上推导, 可得:

$$\begin{aligned} \beta_{G^L}(t) &= \left[Ct - \sum_{\{p(i)=H\}} r_i \times t - \sum_{\{p(i)=H\}} b_i \right]^+ \\ &= \left[C - \sum_{\{p(i)=H\}} r_i \right] \left[t - \frac{\sum_{\{p(i)=H\}} b_i}{C - \sum_{\{p(i)=H\}} r_i} \right]^+ \end{aligned} \quad (7)$$

记 VL 聚合流 G^L 的服务速率和服务时延参数分别为 $R_{G^L} = C - \sum_{\{p(i)=H\}} r_i, T_{G^L} = \frac{\sum_{\{p(i)=H\}} b_i}{R_{G^L}}$ 。

对于聚合流 G^L 的组成虚拟链路 VL_i , 其在 G^L 中仍以先入先出 (First In, First Out, FIFO) 调度策略接受交换机的服务, 可得:

$$\begin{aligned} \beta_{VL_i}(t) &= \left[R_{G^L}(t - T_{G^L}) - \left[\sum_{\{p(j)=L\}} r_j - r_i \right] \right]^+ \\ &\quad \times \left[t - T_{G^L} - \frac{\sum_{\{p(j)=L\}} b_j - b_i}{R_{G^L}} \right] - \left[\sum_{\{p(j)=L\}} b_j - b_i \right]^+ \\ &= \left[R_{G^L} - \sum_{\{p(j)=L\}} r_j + r_i \right] \times \left[t - \frac{R_{G^L} \times T_{G^L} + \sum_{\{p(j)=L\}} b_j - b_i}{R_{G^L}} \right]^+ \end{aligned} \quad (8)$$

VL_i 的服务速率和服务时延参数分别为:

$$\begin{aligned} R_{VL_i} &= R_{G^L} - \sum_{\{p(j)=L\}} r_j + r_i \\ T_{VL_i} &= T_{G^L} + \left(\sum_{\{p(j)=L\}} b_j - b_i \right) / R_{G^L} \end{aligned}$$

类似地, 可得聚合流 G^H 中的 VL_i 服务速率和服务时延参数分别为: $R_{VL_i} = R_{G^H} - \sum_{\{p(j)=H\}} r_j + r_i = C - \sum_{\{p(j)=H\}} r_j + r_i$ 和 $T_{VL_i} = T_{G^H} + \left(\sum_{\{p(j)=H\}} b_j - b_i \right) / R_{G^H}$ 。

根据上文所设定的各虚拟链路参数, 经过计算, 得到 12 条虚链路的延迟上界, 如表 7 所示。

表 7 采用 SP 调度策略的 AFDX 虚拟链路中端到端时延

TTVL 帧	时延上界/ μs	TTVL 帧	时延上界/ μs
VL ₁	221.80	VL ₇	258.86
VL ₂	201.74	VL ₈	453.99
VL ₃	246.04	VL ₉	373.30
VL ₄	176.94	VL ₁₀	243.87
VL ₅	324.78	VL ₁₁	348.49
VL ₆	453.89	VL ₁₂	119.88

对比表 5 ~ 表 7 中各虚拟链路的端到端时延可以看出, 时间触发链路能够有效降低时延上界, 消除时延抖动, 而速率限制虚拟链路较采用静态优先级调度策略中低优先级虚拟链路的时延也有一定程度的降低, 适用于非硬实时要求的数据传输。

5 结束语

针对时间触发 AFDX 网络中时间触发虚拟链路的特性, 本文提出了基于帧长度优先原则的端系统调度时刻设计算法以及相应的交换机调度算法。基于帧长度优先原则的调度时刻设计算法能够提高矩阵周期中时间窗的利用率, 从而有效地使用网络带宽。通过采用网络演算方法对 AFDX 及 TTAFDX 中各类型虚拟链路的时延进行计算和比较, 表明时间触发虚拟链路的时延抖动为微秒级常量, 满足航电系统中具有硬实时要求的数据传输。

对 TTAFDX 网络的研究目前处于理论分析阶段,其调度时刻设计算法仍需进一步研究。调度表设计复杂度和带宽利用率的平衡问题、动态调度表的设计以及对网络演算方法的优化将是下一步的研究方向。

参 考 文 献

- [1] 郑朝辉,陈新中,张晓先.综合模块化航空电子系统的可靠性设计[J].计算机工程,2009,35(23):272-273,277.
- [2] Hull A. Rearchitecting Legacy LRUs to Support Distributed IMA a Strategy to Integrate HMD Systems into Legacy Aircraft [C] // Proceedings of the 30th Digital Avionics Systems Conference. Washington D. C., USA: IEEE Press, 2011.
- [3] Wolfig R, Jakovljevic M. Distributed IMA and DO-297: Architectural, Communication and Certification Attributes [C] // Proceedings of the 27th Digital Avionics Systems Conference. Washington D. C., USA: IEEE Press, 2008.
- [4] 屠晓杰,何 锋,熊华钢,等.分布式 IMA 的网络分区方法及其实时性能分析[J].航空学报,2013,34(1):112-120.
- [5] ARINC. ARINC 664-2005 Aircraft Data Network, Part 7: Avionics Full Duplex Switched Ethernet (AFDX) Networks [S]. 2005.
- [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(续)[J].航空标准化与质量,2008(2):20-23.
- [7] 朱闻渊,尹家伟,蒋祺明.新型航空电子系统总线互连技术发展综述[J].计算机工程,2011,37(S1):398-402.
- [8] Bauer H, Scharbarg J, Fraboul C. Improving the Worst-case Delay Analysis of an AFDX Network Using an Optimized Trajectory Approach [J]. IEEE Transactions on Industrial Informatics, 2010, 6(4):521-533.
- [9] 刘 成,王 彤,李 铮,等.时间触发 AFDX 网络的设计和实时性分析[J].北京航空航天大学学报,2013,39(6):728-733.
- [10] Li Xie. Research on the Real-time Property of TTCAN Protocol in Vehicle Communication Network [J]. Applied Mechanics and Materials, 2014, 487:674-677.
- [11] 陈克伟,宋小庆,李振伟,等.车电系统 TTCAN 网络调度设计与分析[J].装甲兵工程学院学报,2013,27(3):63-65.
- [12] Steinbach T, Korf F, Schmidt T C. Real-time Ethernet for Automotive Applications: A Solution for Future In-car Networks [C] // Proceedings of IEEE International Conference on Consumer Electronics-Berlin. Washington D. C., USA: IEEE Press, 2011:216-220.
- [13] 刘晚春,李 峭,何 锋,等.时间触发以太网同步及调度机制的研究[J].航空计算技术,2011,41(4):122-127.
- [14] Boudec J Y L, Thiran P. Network Calculus: A Theory of Deterministic Queuing Systems for the Internet [M]. Berlin, Germany: Springer-Verlag, 2001.
- [15] 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.
- [5] Kim S K, Carrington D. A Formalism to Describe Design Patterns Based on Role Concepts [J]. Formal Aspects of Computing, 2009, 21(5):397-420.
- [6] 钟金琴,辜丽川,张友华,等.设计模式的一种形式化描述方法[J].微电子学与计算机,2009,26(6):150-153.
- [7] 曾红卫,腾中梅.用 Z 形式化描述的软件设计模式[J].计算机工程,2006,32(13):69-70.
- [8] 王 绘.基于面向对象着色 Petri 网构建设计模式[D].济南:山东大学,2010.
- [9] 袁崇义. Petri 网原理 [M]. 北京:科学出版社,2013.
- [10] Girault C, Valk R. Petri Nets for Systems Engineering: A Guide to Modeling, Verification, and Applications [M]. Berlin, Germany: Springer-Verlag, 2003.
- [11] 卫建国,王建林,庄立伟.气象软件设计模式的研究与实现[J].计算机工程,2010,36(9):59-61,64.
- [12] 梁盟磊,王小平,薛小平,等.基于 TLA 的 UML 模型形式化验证[J].计算机工程,2011,37(2):72-74.
- [13] Gamma E, Helm R, Johnson R, et al. Design Pattern: Elements of Reusable Object-oriented Software [M]. Boston, USA: Addison-Wesley, 1995.
- [14] Dong Jing, Sun Yongtao, Zhao Yajing. Design Pattern Detection by Template Matching [C] // Proceedings of 2008 ACM Symposium on Applied Computing. New York, USA: ACM Press, 2008:765-769.
- [15] 赵俊峰,周建涛,邢冠男. UML 活动图到 Petri 网的转换方法及实现研究 [J]. 计算机科学, 2014, 41(7):143-147.

编辑 金胡考

编辑 金胡考

(上接第 36 页)