

时间触发以太网技术及演示系统介绍

潘伟涛

wtpan@mail.xidian.edu.cn

西安电子科技大学
ISN国家重点实验室网络交换团队

内容提要

- 1 团队介绍
- 2 时间触发以太网关键技术
- 3 演示系统介绍



团队介绍



专注专用定制的网络交换技术20余年

隶属于西安电子科技大学通信工程学院综合业务网理论及关键技术 (ISN) 国家重点实验室。目前由4位教师、4位博士生及50余位硕士生组成。团队负责人是邱智亮教授。



星载交换机路由器



航空航天领域专用总线

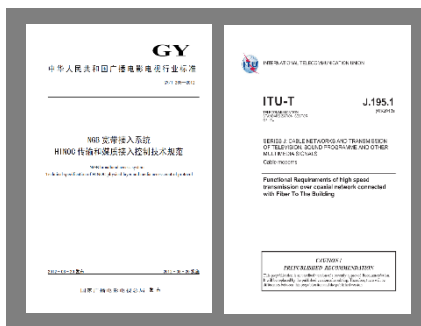
广播电视
频段

0.75GHz

同轴有线接入

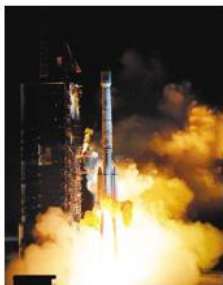
具有从协议标准到FPGA样机和商用芯片的全流程定制能力

秉承“工匠”精神

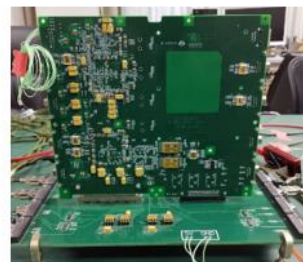


西安电子科技大学
XIDIAN UNIVERSITY

星载交换机路由器



高轨



低轨

星载交换机路由器

- FPGA实现
- 高可靠性代码设计
- 某高轨星载交换机稳定运行数年



航空航天领域专用总线

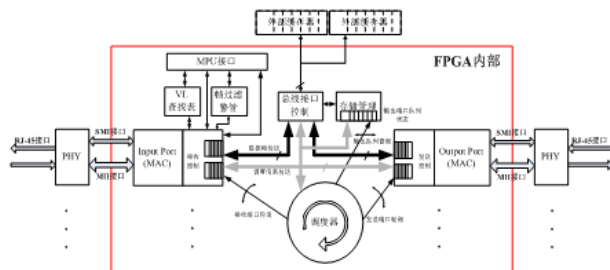
AFDX交换机

2007年与某单位合作开发了AFDX交换机，已被成功推广应用。

2008年承担陕西省13115项目，在AFDX交换机基础上实现RC+BE业务交换。

关键技术积累：

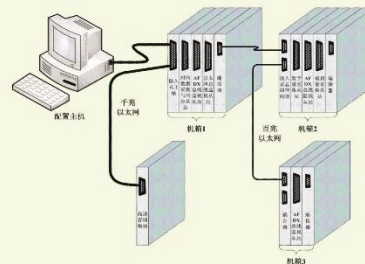
- 片内实现ARINC 664规范要求的过滤、警管及输出流控，即RC业务。
- 存储队列管理、路由查找和分组调度功能。



16端口AFDX交换机

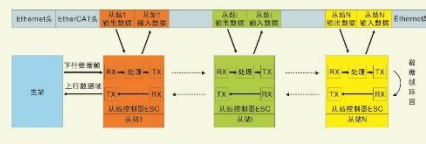


新型网络化机载数据采集系统



主要技术特点：

- 采用EtherCAT实时以太网技术
- 传输速率高（达100Mbps）
- 同步性能好（<1us）
- 支持周期数据、非周期数据采集
- 数据采集周期短（100us）
- 组网灵活



航空航天领域专用总线

- FPGA实现
- 高完整性代码设计
- 部分产品已被合作单位流片并型号应用



同轴接入技术

2005年 广科院、北大与西电构思HINOC



2010年至2012年 多款1.0芯片研发成功



2012年 HINOC2.0工作组成立



2014年底 HINOC2.0 HN1000系列芯片研发成功



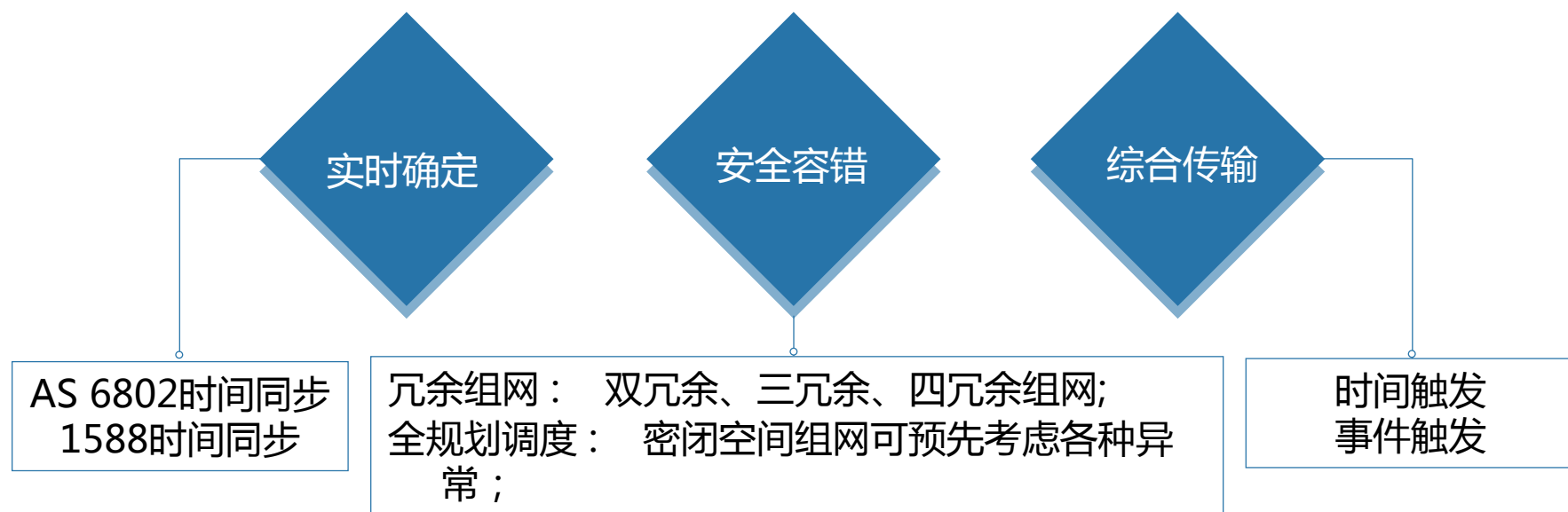
HINOC，1.0到2.0到3.0的发展之路

- 10年的前沿技术积累和关键技术攻关、30余项专利、4款试验芯片、2代技术演进，HINOC技术逐步成熟并走向产业化
- 西电团队负责MAC层协议标准及芯片研发，参与并主导了HINOC的发展过程
- 2019年，国家重点研发计划HINOC3.0项目获批，接口速率10Gbps，将部署100万户用户



西安电子科技大学
XIDIAN UNIVERSITY

时间触发以太网关键技术

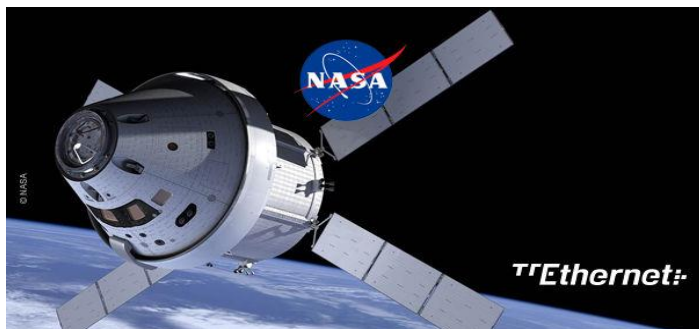


TTE技术优势

将时间触发协议（TTP）的实时性、确定性、容错性与传统以太网的普及性、成熟性、组网灵活性相结合，能在同一网络中高可靠、高性能地支持各种不同应用属性业务的综合传输



时间触发以太网关键技术



航天领域飞行器：

- 高可靠（拜占庭容错）
- 抗辐照（单粒子翻转等）
- 动态可重构



航空领域飞行器（飞控系统）：

- 强实时（极低延迟，零抖动）
- 高完整性设计（EDE、自说自话、伪装等）
- 高带宽



轨道交通等民用领域：

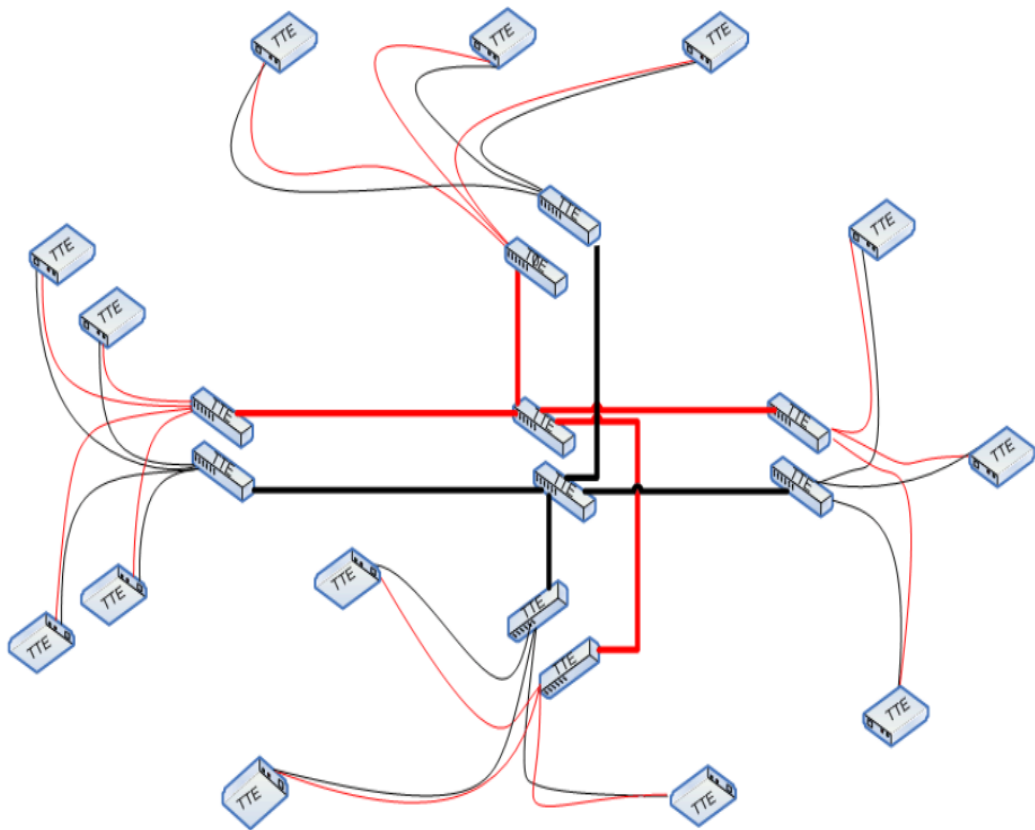
- 强实时（低延迟抖动）
- 高可靠（TSN）
- 低成本、低功耗
- 复杂度、灵活性高

TTE应用场景

不同场景对TTE的要求不尽相同，必须结合实际场景去设计相应的TTE架构



时间触发以太网关键技术



时钟同步技术 (6802或1588协议)



调度表生成算法及生成软件



网络管理及文件分发协议实现



TTE交换机及端节点设计

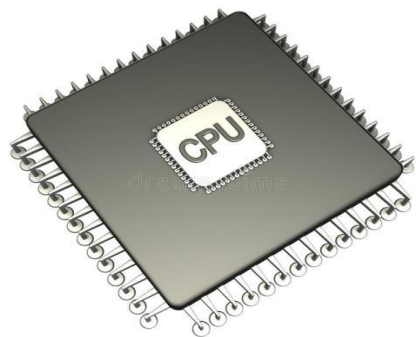


测试及健康状态监测

TTE网络五大关键技术



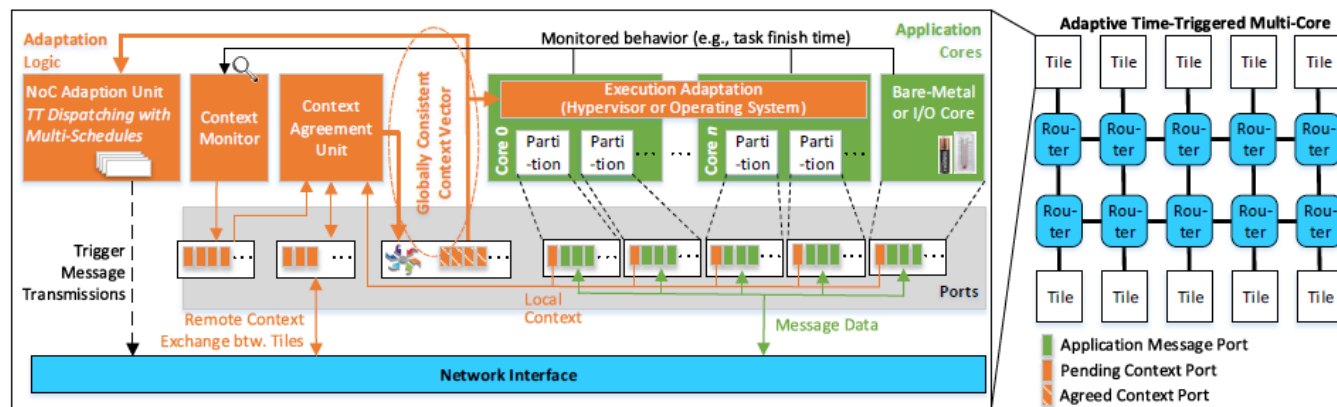
时间触发以太网关键技术



RTOS

确定性嵌入式CPU及OS设计

- Cache机制带来不确定性
- MMU，虚拟地址向物理地址转换时可能页缺失
- 内部各种总线冲突带来的不确定性
- ARM-R系列、RISC-V也在开发相应的实时CPU
- 时间触发操作系统（653），时间触发任务调度，分时分区
- TTOS，鸿蒙等



Predictable Processors for Mixed-Criticality Systems and Precision-Timed I/O

Michael Zimmer

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2015-181
<http://www.eecs.berkeley.edu/Pubs/TechRpts/2015/EECS-2015-181.html>

August 10, 2015

Article

Adaptive Time-Triggered Multi-Core Architecture

Roman Obermaier*, Hamidreza Ahmadian, Adele Maleki, Yosab Bebawy, Alina Lenz and Babak Sorkhpour

Department of Electrical Engineering and Computer Science, University of Siegen, 57068 Siegen, Germany; hamidreza.ahmadian@uni-siegen.de (H.A.); Adele.Maleki@uni-siegen.de (A.M.); Yosab.Bebawy@uni-siegen.de (Y.B.); alina.lenz@uni-siegen.de (A.L.); Babak.Sorkhpour@uni-siegen.de (B.S.)

* Correspondence: roman.obermaier@uni-siegen.de; Tel.: +49-271-740-3332

Received: 27 September 2018; Accepted: 18 January 2019; Published: 22 January 2019



Abstract: The static resource allocation in time-triggered systems offers significant benefits for the safety arguments of dependable systems. However, adaptation is a key factor for energy efficiency and fault recovery in Cyber-Physical System (CPS). This paper introduces the Adaptive Time-Triggered Multi-Core Architecture (ATMA), which supports adaptation using multi-schedule graphs while preserving the key properties of time-triggered systems including implicit synchronization, temporal predictability and avoidance of resource conflicts. ATMA is an overall architecture for safety-critical CPS based on a network-on-a-chip with building blocks for context agreement and adaptation. Context information is established in a globally consistent manner, providing the foundation for the temporally aligned switching of schedules in the network interfaces. A meta-scheduling algorithm computes schedule graphs and avoids state explosion with convergence horizons for events. For each tile, the relevant part of the schedule graph is efficiently stored using difference encodings and interpreted by the adaptation logic. The architecture was evaluated using an FPGA-based implementation and example scenarios employing adaptation for improved energy efficiency. The evaluation demonstrated the benefits of adaptation while showing the overhead and the trade-off between the degree of adaptation and the memory consumption for multi-schedule graphs.

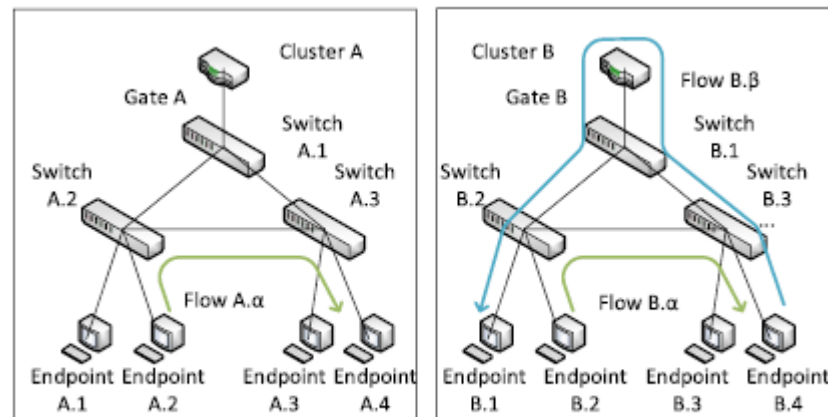
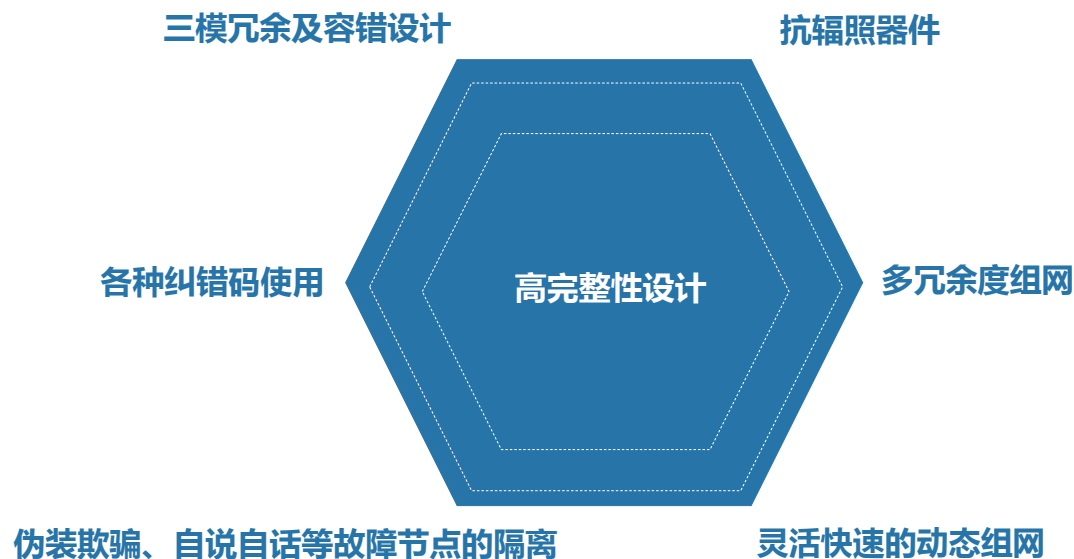
挑战1 端系统设计

- 种类繁多，带CPU、不带CPU或者需要硬件UDP/IP解析
- CPU存在不确定性
- 操作系统存在不确定性

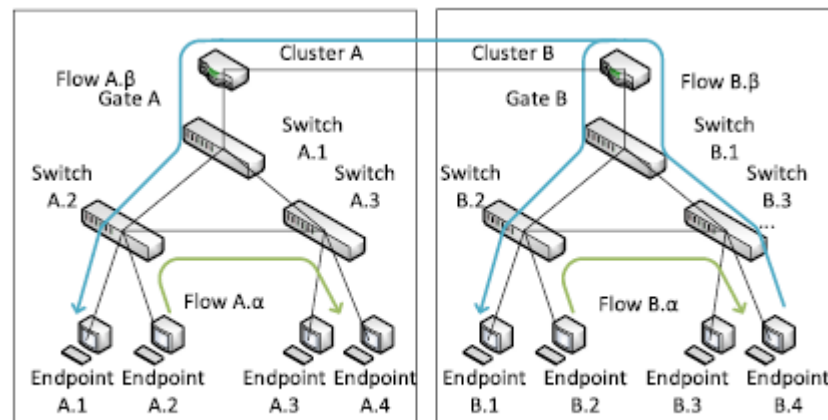


西安电子科技大学
XIDIAN UNIVERSITY

时间触发以太网关键技术



(a)



(b)

Ningchen W, Qinghan Y, Hai W, et al. Adaptive Scheduling for Multi-cluster Time-Triggered Train Communication Networks[J]. IEEE Transactions on Industrial Informatics, 2018:1-1.

挑战2 高完整性设计及完美调度算法

- 航空航天领域高安全性的要求
- 单位时间内出现灾难性故障的几率小于 10^{-9}
- 如何验证



演示系统介绍

自主实现了TTE系统的六项核心技术

- 网络同步技术采用6802或1588时钟同步协议；
- 调度表生成算法及生成软件；
- 615A文件分发协议；
- 千兆TTE交换机设计；
- TTE端系统设计—主机式、嵌入式及瘦终端；
- TTE网络管理技术—配置、读取、监控网络设备的工作状态；

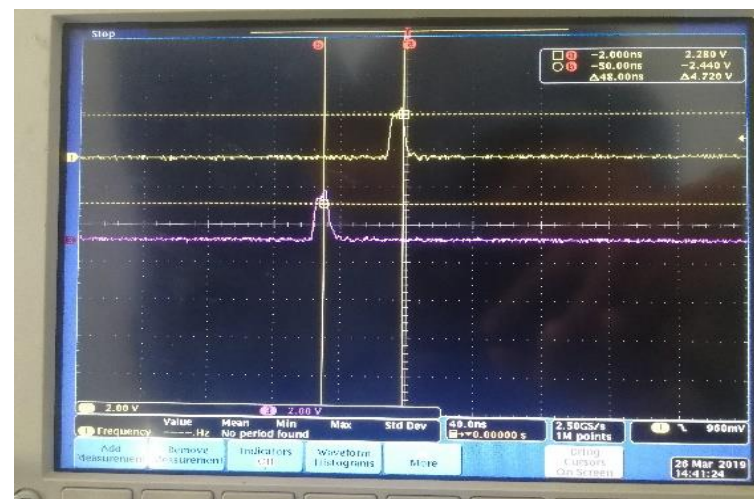
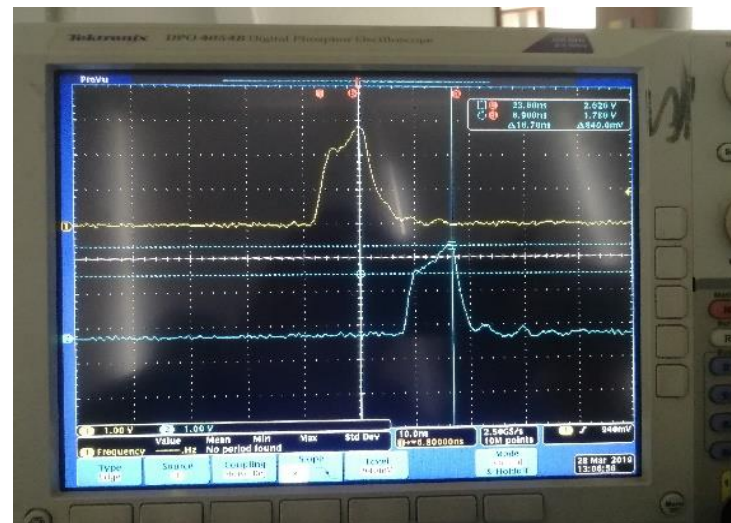


演示系统视频链接: <https://v.qq.com/x/page/x08980oral7.html?sf=qq>



演示系统介绍

团队自主实现了6802或1588两种同步方式，端到端同步精度均小于100ns。以6802同步为例说明，TTE交换机作为压缩主机CM，TTE端系统作为同步主机SM，主要实现固化功能、压缩功能以及结团检测功能，同时定义了冷启动服务、重启动服务以及时间同步服务。



6802和1588时钟同步技术



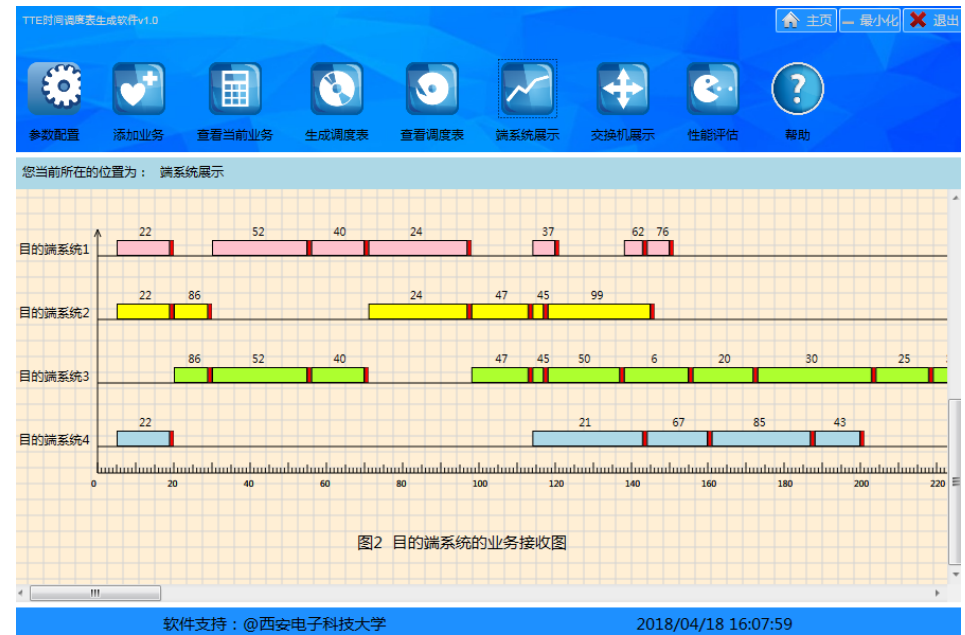
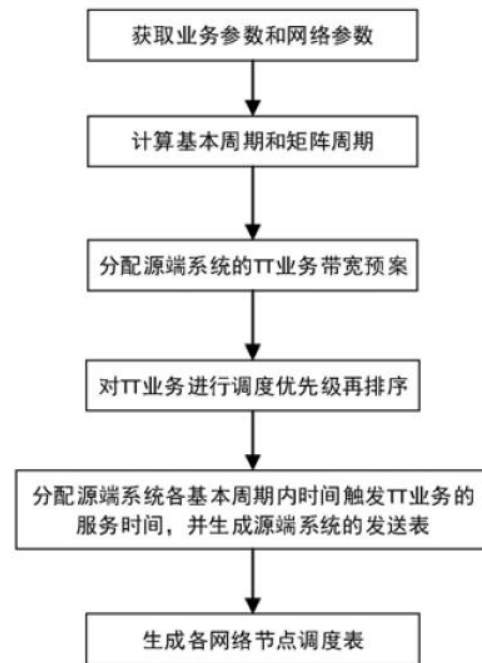
演示系统介绍

(54)发明名称

最小时延最大匹配的时间触发业务调度表生成方法

(57)摘要

本发明公开了一种最小时延最大匹配的时间触发业务调度表生成方法。主要解决现有技术生成调度表时TT业务时延大,网络链路利用率低的问题。其包括:获取业务参数和网络参数,并计算业务调度的矩阵周期和基本周期;将TT业务的传输时间点放入矩阵周期的一个或多个基本周期内,生成源端系统的TT业务带宽预案;采用最大匹配算法,对TT业务进行调度优先级排序,分配源端系统各基本周期中TT业务的发送时间,依次生成源端系统的发送表、交换机的接收表和发送表以及目的端系统的接收表。本发明生成的调度表能保证各网络节点有序、无冲突地传输TT业务,提高了链路利用率,且TT业务总时延最小,可用于时间触发以太网中TT业务的调度。

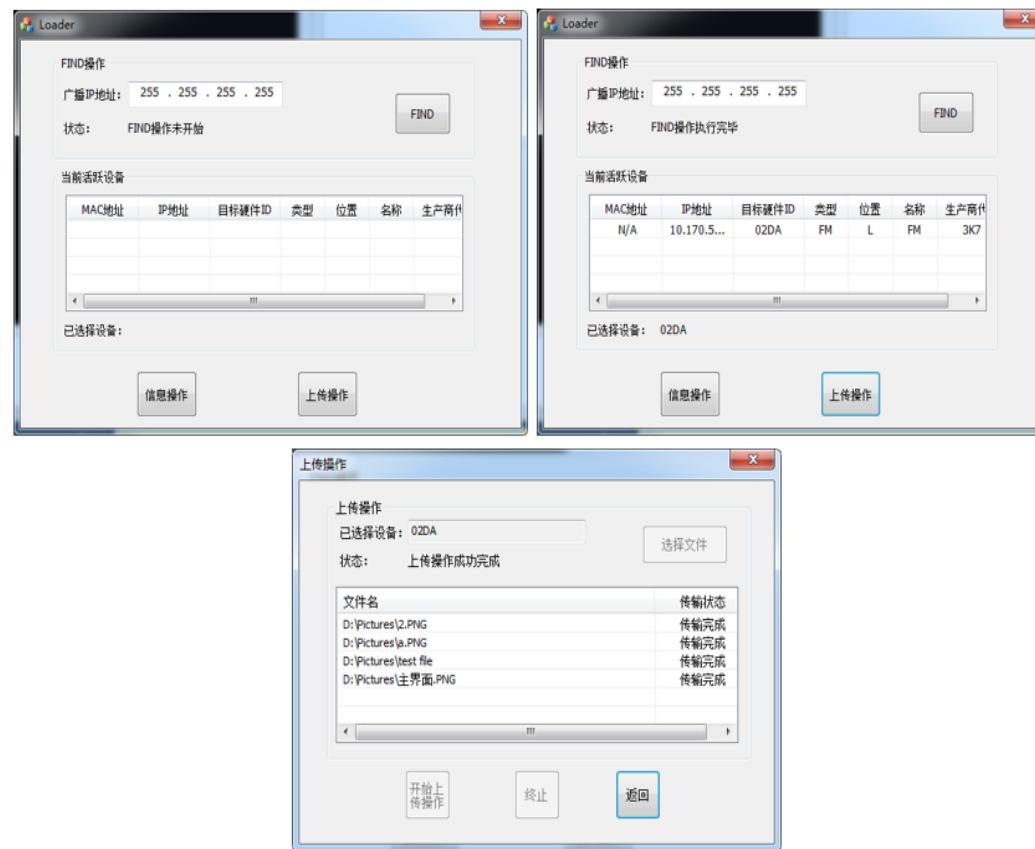


时间调度表生成算法及软件



演示系统介绍

时间调度表文件生成后，需要使用**615A**远程文件传输协议将调度表文件分发到各网络节点。团队自主实现了基于**615A**协议的调度表分发软件，**615A**协议分为加载端和目标端两部分。加载端相当于管理端，主要进行操作命令的输入。目标端负责对加载端的请求进行相应的消息回复。



615A文件分发协议

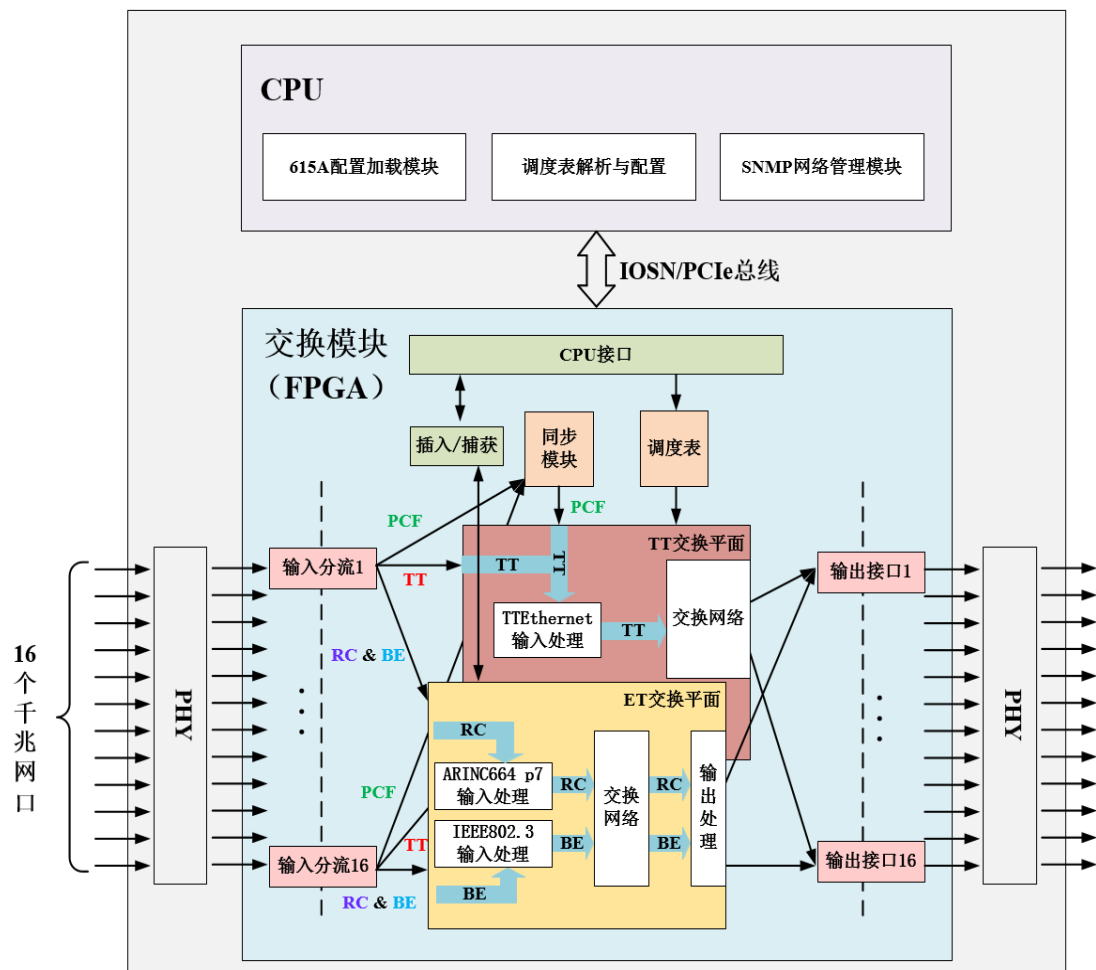


演示系统介绍

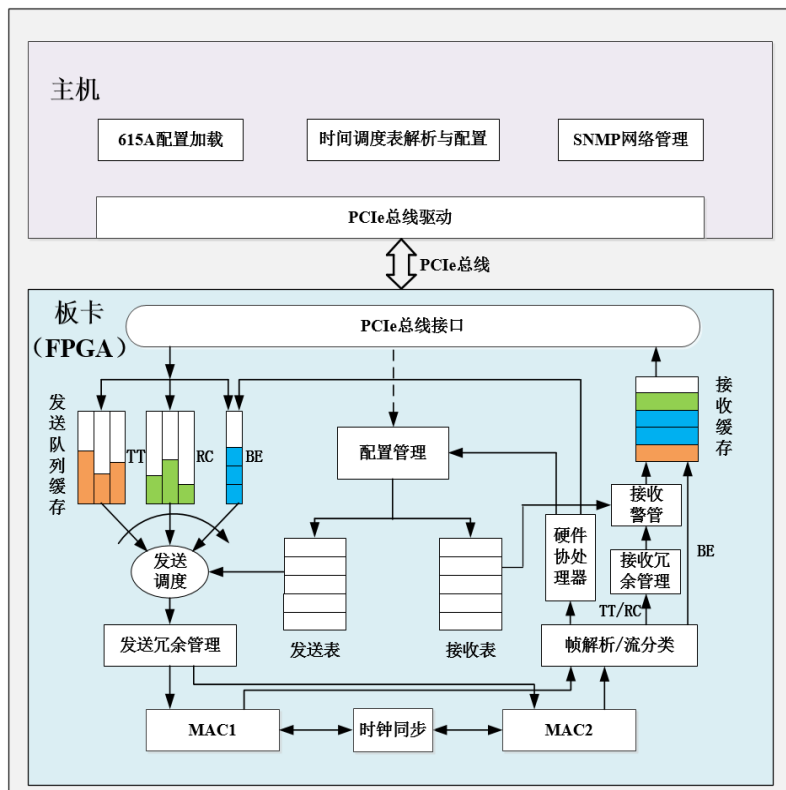
CPU主要完成：SNMP网络管理、615A配置加载、调度表解析与配置

交换模块主要完成：

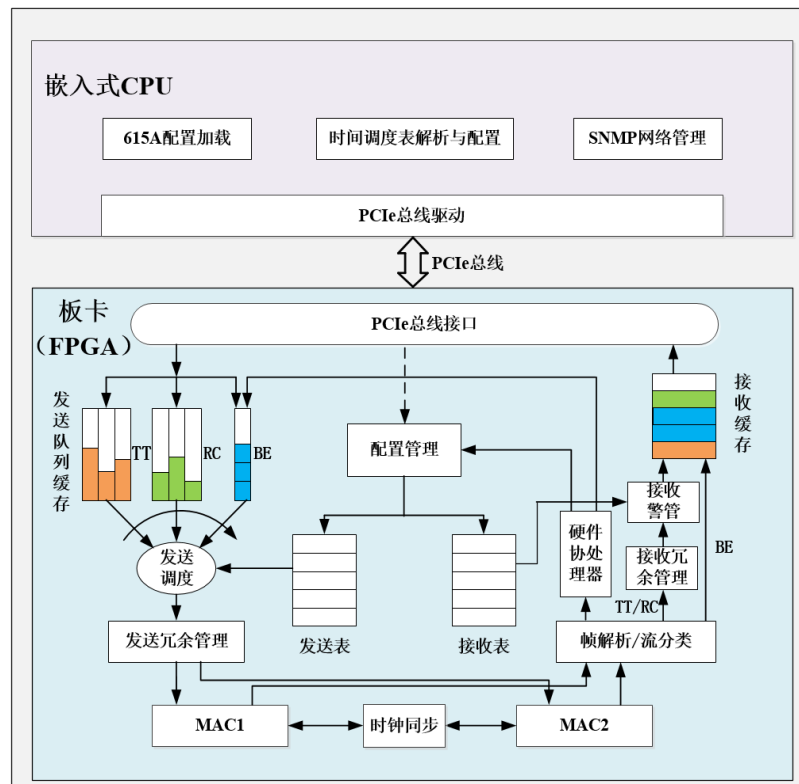
- TT、RC、BE混合业务双平面交换
- TT平面采用Crossbar全互联交换网络
 - 处理时延小于1us
 - 转发时延小于12.5us
- ET交换平面采用交叉点带缓存的Crossbar交换结构
 - 处理时延小于5us
 - 转发时延0.1~1ms
- 6802 / 1588时钟同步：同步精度小于100ns
- 支持4096条虚拟链路（VL）
- QoS功能：
 - 支持TT窗口保护
 - 支持RC过滤、警管、优先级功能
 - 支持8个BE优先级
- 接口速率：10/100/1000M



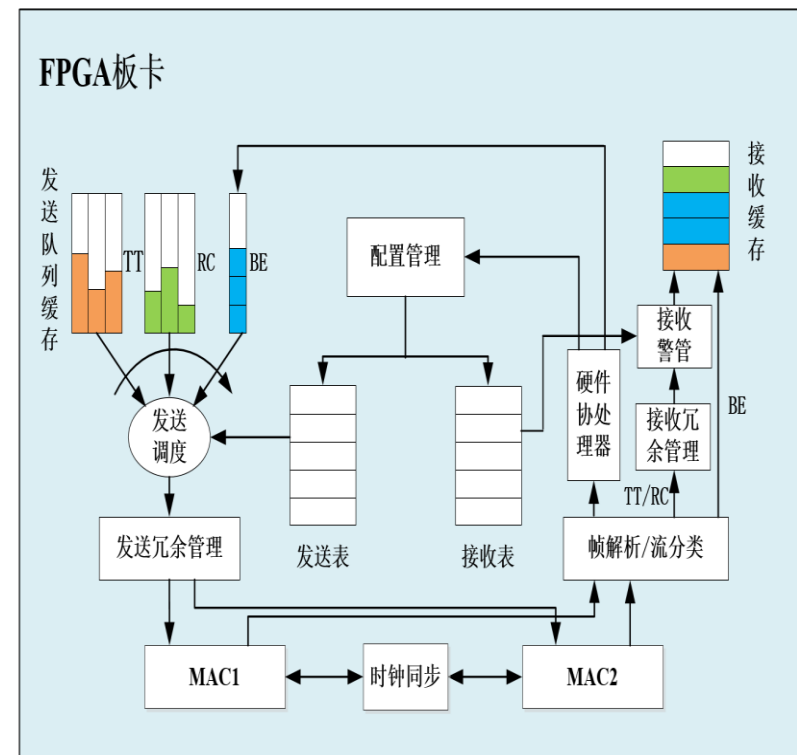
演示系统介绍



主机式端系统



嵌入式端系统



不带CPU的瘦终端



演示系统介绍

监控软件分为三部分：

- 1) Agent，运行在被监控的端系统上，负责采集数据并上报给web服务器；
- 2) Web服务器，运行在Linux系统上，负责数据格式转换并转发给前端页面；
- 3) 前端页面，在浏览器中打开，负责数据可视化等操作。



谢谢！

