EPT 怿星科技

# 车载TSN设计与实践

Design and Practice of TSN in Automobile

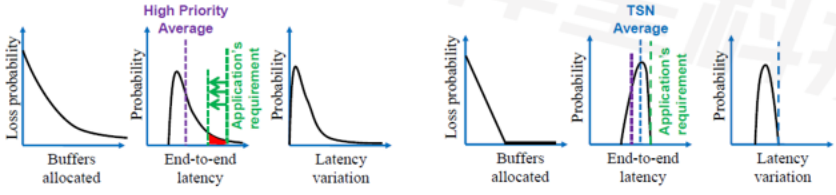
# 01
# TSN概述

## Overview of TSN

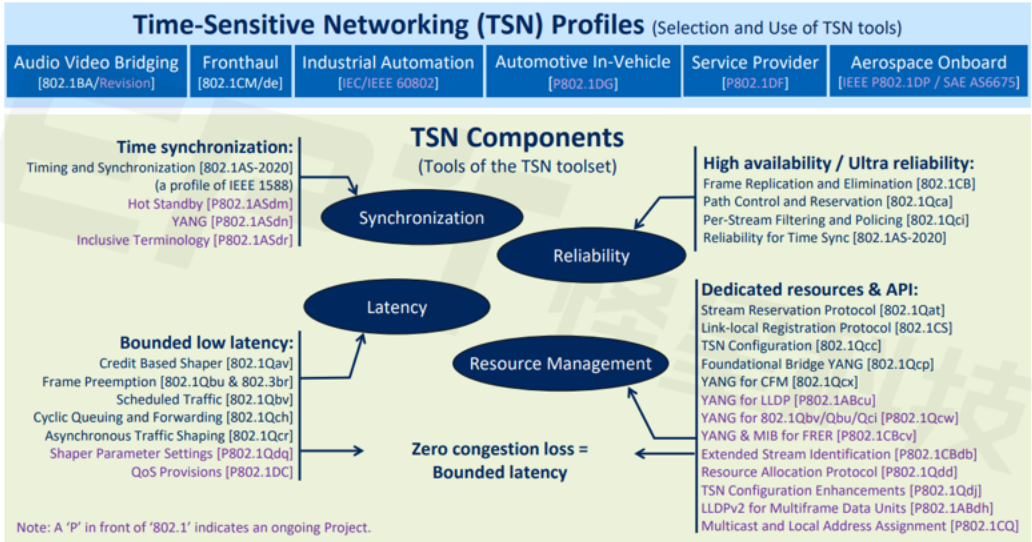
# TSN概述 Overview of TSN

- AVB解决了音视频在以太网中的实时同步传输问题
- AVB的特性逐渐被工业和汽车等行业关注，对其提出更多需求
- 2012年11月，AVB任务组扩展了其工作范围，并正式更名为TSN(Time Sensitive Networking)
- TSN协议是在IEEE 802.1标准框架下，基于特定需求制定的一组“标准”，旨在通过IEEE 802网络提供确定性服务，如有限延迟，低延迟变动和低包丢失率

## Traditional VS TSN

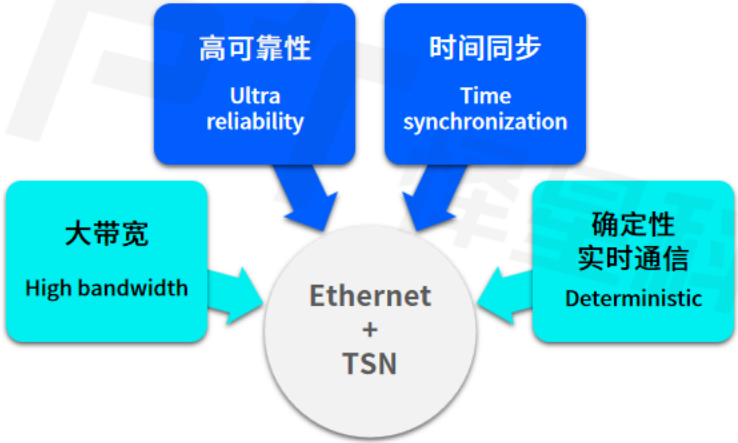



Source：Bounded Low Latency in IEEE 802.1 and IETF DetNEt




Source：https://1.ieee802.org/tsn/

- 智能化、网联化对汽车网络的新需求



- 车载领域TSN协议框架

- 车载领域TSN应用场景


**自动驾驶**

传感器数据融合
V2X
关键控制

**影音娱乐**

360环视
影音同步

**信息安全**

防DoS攻击
流量控制

# 02

# 车载TSN设计

## Design of Automotive TSN

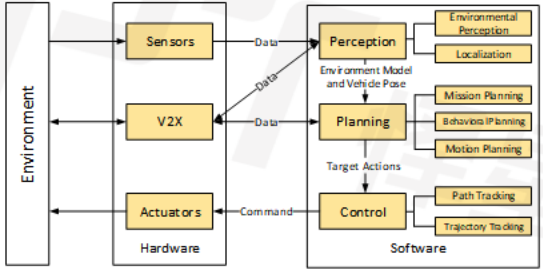
# 车载TSN设计 Design of Automotive TSN



- 车载网络有其独特性，车载TSN的设计也需要因地制宜。
- 以自动驾驶为例，简要介绍TSN设计过程

场景及需求分析 → 协议选取 → 参数设计

- 场景及需求分析
  - 分析数据流特点和需求，自动驾驶有如下交互数据，传感器、V2X、控制数据



场景及需求分析 → 协议选取 → 参数设计

- 场景及需求分析
  - 自动驾驶数据需求

| 传感器类型 | 采集频率 | 数据帧发送周期 | 传输延迟要求 | 丢包容忍性 | 带宽 |
|---|---|---|---|---|---|
| 毫米波雷达 | 10Hz~50Hz | 20ms~100ms | ＜2ms | 不可丢包 | 10~100 kbps |
| 车载摄像头(raw) | 24~60fps | <0.1ms | ＜2ms | 不可丢包 | >500Mbps |
| 激光雷达 | >10kHz | <2ms | ＜2ms | 可丢包 | 3~15Mbps |
| 超声波传感器 | >10Hz | <100ms | ＜10ms | 可丢包 | 5~20kbps |
| GPS/IMU | >10Hz | <100ms | ＜10ms | 可丢包 | 5~20kbps |
| 高精地图 | — | TBD | ＜10ms | 可丢包 | TBD |

| 应用场景 | 通信时延/ms | 数据速率/(Mbit/s) | 通信距离 |
|---|---|---|---|
| 车辆编队 | 10~500 | 0.012~65 | 80~350 |
| 先进驾驶 | 3~100 | 10~53 | 360~700 |
| 传感器扩展 | 3~100 | 10~1000 | 50 |
| 远程驾驶 | 5 | 上行25，下行1 | 无限制 |

| 数据类型 | 发送周期 | 传输延迟要求 | 丢包容忍性 | 数据帧大小限制 |
|---|---|---|---|---|
| Safety-relevant Control | ≤10ms | < 1ms | 不可丢包 | 64~128字节(100Mbps) |
| Safety-irrelevant Control | < 200ms | < 20ms | 可丢包 | 64~256字节(100Mbps) |

- 协议选取
  - 考虑通信数据需求、网络拓扑

**主要整形协议**

| 协议 | 缩写 | 中文名称 | 应用场景 |
|---|---|---|---|
| 802.1 Qav | FQTSS | 队列、转发及流量整形协议 | 保证延迟低于上限值，支持对每个队列整形 |
| 802.1Qcr | ATS | 异步整形协议 | 保证延迟低于上限值，支持对每个数据流整形 |
| 802.1 Qbv | TAS | 时间感知整形协议 | 周期数据流，延迟要求高的应用 |
| 802.1 Qch | CQF | 周期队列及转发协议 | 周期数据流，延迟要求较高的应用 |

- 协议选取
  - 考虑通信数据需求、网络拓扑

| 整形器类型 | 无人驾驶系统数据 |
| --- | --- |
| 802.1Qbv | 关键控制数据 |
| 802.1Qav（Class A）或802.1Qcr | 车载摄像头、毫米波雷达和激光雷达 |
| 802.1Qav（Class B）或802.1Qcr | 超声波传感器、GPS/IMU、高精地图 |
| Best effort (higher Priority) | 超声波传感器、GPS/IMU、高精地图、一般控制数据 |

| 协议 | 缩写 | 中文名称 | 作用 | 备注 |
| --- | --- | --- | --- | --- |
| 802.1AS | gPTP | 广义精确时钟同步协议 | 在网络内实现各节点的时钟同步 | N/A |
| 1722 | AVTP | 音视频传输协议 | 实现音视频和传统总线数据封装、传输 | N/A |
| 802.1Qat | SRP | 流预留协议 | 在数据流传输路径上预留带宽 | 采用静态配置，不使用动态预留 |
| 802.1Qav | FQTSS | 队列、转发及流量整形协议 | 实现毫秒级传输延迟 | N/A |
| 802.1Qbv | TAS | 时间感知整形协议 | 实现亚毫秒级传输延迟 | N/A |
| 802.1 CB | FRER | 帧复制与消除协议 | 实现关键数据的无缝冗余 | 高功能安全数据 |

场景及需求分析 → 协议选取 → 参数设计

- 参数设计
  - 依据标准、需求规范、场景分析的成果，结合网络拓扑，设计各TSN协议配置参数

| No. | Name | Data type | Default value | 备注 |
|---|---|---|---|---|
| 1 | StreamID | | | |
| 2 | MAA | | | |
| 3 | DestM | | | |
| 4 | EndS | | | |
| 5 | VlanI | | | |
| 6 | VlanI | | | |
| 7 | avtpS | | | |
| 8 | avtpS | | | |

| 名称 | 说明 | 备注 |
|---|---|---|
| queueMaxS | 应用相同封装格式的 Sequence encode decode 功能的流识别 ID 列表 | |
| GateEnable | 此 Sequence encode/decode 功能所在的端口 | True：out-facing<br>False：in-facing |
| AdminGateS | 此 Sequence encode/decode 功能放置在端口的具体位置 | |
| AdminCont | 指定此条目是被动（False），因此仅用于解码通过协议栈的输入数据包；还是主动（True），因此既用于识别输入数据包，也用于对正在通过的输出数据包进行编码 在协议栈中。 | Boolean |
| AdminCycle | 枚举值，指示用于此 Sequence encode/decode 功能实例的封装类型 | R-TAG、HSR sequence tag、PRP sequence trailer |
| AdminCycle | 要添加到输出数据包中的 HSR 序列标签的 PathId 字段或 PRP 序列尾标的 LanId 字段中的 4 位整数值。 | |
| AdminBase | | |

| 名称 | 说明 | 备注 |
|---|---|---|
| queueMaxS | | |
| GateEnable | 此 Sequence encode/decode 功能所在的端口 | |
| AdminGateS | 此 Sequence encode/decode 功能放置在端口的具体位置 | |
| AdminCont | 指定此条目是被动（False），因此仅用于解码通过协议栈的输入数据包；还是主动（True），因此既用于识别输入数据包，也用于对正在通过的输出数据包进行编码 在协议栈中。 | |
| AdminCycle | 枚举值，指示用于此 Sequence encode/decode 功能实例的封装类型 | |
| AdminCycle | 要添加到输出数据包中的 HSR 序列标签的 PathId 字段或 PRP 序列尾标的 LanId 字段中的 4 位整数值。 | |
| AdminBase | | |

| 名称 | 说明 | 备注 |
|---|---|---|
| frerSeqRcvyS | | |
| frerSeqRcvyPo | | |
| frerSeqRcvyDe | | |
| frerSeqRcvyAl | | |
| frerSeqRcvyRe | | |
| frerSeqRcvyRe | | |
| frerSeqRcvyTa | | |
| frerSeqRcvyEn | | |
| frerSeqRcvyLa | | |
| Latent error det | | |

| 名称 | 说明 | 备注 |
|---|---|---|
| frerSeqEncStreamList | | |
| frerSeqEncPort | | |
| frerSeqEncDirection | | |
| frerSeqEncActive | | |
| frerSeqEncEncapsType | | |
| frerSeqEncPathIdLanId | | |

场景及需求分析 → 协议选取 → 参数设计

- 参数设计
  - 整形参数计算
    - Qav、Qcr参数计算参考P802.1Qdq中burst流计算方法

$$\text{MaxFrameSize} = \min\left(\text{floor}\left(\frac{\text{dataSize}}{\text{targetLatency}} \times \text{classMeasurementInterval}\right), \text{Maximum SDU Size}\right)$$

$$\text{MaxIntervalFrames} = \text{ceil}\left(\frac{1}{\text{MaxFrameSize}} \times \frac{\text{dataSize}}{\text{targetLatency}} \times \text{classMeasurementInterval}\right)$$

$$\text{CommittedInformationRate} = \frac{\text{dataSize}}{\text{targetLatency}}$$

    - Qbv参数主要涉及CycleTime及GCL参数，将其转化为NP问题，利用SMT求解器求解

# 03

# TSN仿真分析

Simulation analysis of TSN

# TSN仿真分析 Simulation analysis of TSN

- TSN设计参数繁多，车内网络行为受到拓扑、数据流特性和流量调度等影响，需要借助仿真工具，对网络设计进行定量评估。
- TSN仿真平台协议支持情况

| 协议 | OMNeT++ 模型 | | | | 其他模型 | | |
|---|---|---|---|---|---|---|---|
| | INET | NeSTiNg | CoRE4INET | TSimNet | OPNET Model | TCN | RTaW-Pegase |
| Credit-Based Shaper (Qav) | √ | √ | √ | | | √ | √ |
| Scheduled Traffic (Qbv) | √ | √ | √ | | √ | √ | √ |
| Asynchronous Traffic Shaping (Qcr) | √ | | | | | | √ |
| Clock Sync. (gPTP) | √ | | | | √ | √ | √ |
| Frame Preemption(Qbu) | √ | √ | | √ | | | √ |
| Per-stream Filtering and Policing(Qci) | √ | | √ | √ | √ | | √ |
| Frame Replication and Elimination(CB) | √ | | | √ | √ | √ | √ |
| Configuration Protocols(Qcc) | √ | | √ | | √ | | |

- 开源软件平台（Omnet++&INET框架&TSN模块）

```
network ZonalArchitecture extends TsnNetworkBase
{
    parameters:
        @display("bgb=1773.408,962.752");
    submodules:
        ZonalECU1: <default("TsnSwitch")> like IEthernetNetworkNode {
            @display("p=774.592,254.016");
            @display("i=device/device");
        }
        ZonalECU2: <default("TsnSwitch")> like IEthernetNetworkNode {
            @display("p=774.592,453.152");
            @display("i=device/device");
        }
        switch1: <default("TsnSwitch")> like IEthernetNetworkNode {
            @display("p=1077.2161,351.232");
        }
        ZonalECU3: <default("TsnSwitch")> like IEthernetNetworkNode {
            @display("p=1375.136,254.016");
            @display("i=device/device");
        }
        ZonalECU4: <default("TsnSwitch")> like IEthernetNetworkNode {
            @display("p=1375.136,453.152");
            @display("i=device/device");
        }
        CentralComputer: <default("TsnClock")> like IEthernetNetworkNode {
            @display("p=1077.2161,453.152");
            @display("i=device/cpu");
        }
        RadarECU: <default("TsnDevice")> like IEthernetNetworkNode {
            @display("p=774.592,128.576");
        }
        RogueECU: <default("TsnDevice")> like IEthernetNetworkNode {
            @display("p=774.592,685.68");
        }
        SpeedECU: <default("TsnDevice")> like IEthernetNetworkNode {
            @display("p=1375.136,134.848");
        }
        FuelControlECU: <default("TsnDevice")> like IEthernetNetworkNode {
            @display("p=1375.136,685.68");
        }
        CameraECU: <default("TsnDevice")> like IEthernetNetworkNode {
            @display("p=1238.72,134.848");
            @display("i=device/camera");
        }
        scenarioManager: ScenarioManager {
            @display("p=100,800;is=s");
        }
    connections:
        RadarECU.ethg++ <--> Eth100M <--> ZonalECU1.ethg++;
        ZonalECU1.ethg++ <--> Eth1G <--> switch1.ethg++;
        ZonalECU1.ethg++ <--> Eth1G <--> ZonalECU2.ethg++;
```

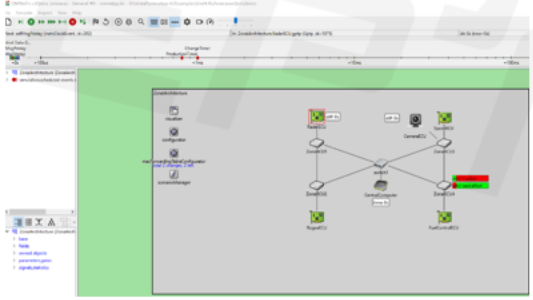
网络建模

```
[General]
network =ZonalArchitecture
**.displayGateSchedules = true
**.gateFilter = "**.eth[2].**"
**.gateScheduleVisualizer[0].height = 16
**.gateScheduleVisualizer[0].placementHint = "right"

*.visualizer.typename = "IntegratedMultiCanvasVisualizer"
*.visualizer.infoVisualizer.displayInfos = true

# enable egress traffic shaping
*.ZonalECU4.hasEgressTrafficShaping = true
*.switch1.hasEgressTrafficShaping = true

# enable time synchronization in all network nodes
*.CentralComputer.hasTimeSynchronization = true
*.RadarECU.hasTimeSynchronization = true
*.CameraECU.hasTimeSynchronization = true

# CentralComputer  applications
*.CentralComputer.numApps = 4
*.CentralComputer.app[0..2].typename = "UdpSinkApp"
*.CentralComputer.app[0].io.localPort = 1000
*.CentralComputer.app[1].io.localPort = 1001
*.CentralComputer.app[2].io.localPort = 1002

*.CentralComputer.app[3].typename = "UdpSourceApp"
*.CentralComputer.app[3].display-name = "Control Data"
*.CentralComputer.app[3].io.destAddress = "FuelControlECU"
*.CentralComputer.app[3].io.destPort = 1003
*.CentralComputer.app[3].source.packetLength = 128B - 54B # 428 = 88 (UDP) + 200 (
*.CentralComputer.app[3].source.productionInterval = exponential(1ms) # ~1Mbps

# time-aware traffic shaping
*.ZonalECU4.eth[*].macLayer.queue.numTrafficClasses = 2
*.ZonalECU4.eth[*].macLayer.queue.*[0].display-name = "best effort"
*.ZonalECU4.eth[*].macLayer.queue.*[1].display-name = "Control"
*.ZonalECU4.eth[*].macLayer.queue.transmissionGate[0].offset = 1ms
*.ZonalECU4.eth[*].macLayer.queue.transmissionGate[0].durations = [200us, 1800us]#
*.ZonalECU4.eth[*].macLayer.queue.transmissionGate[0].initiallyOpen =false
*.ZonalECU4.eth[*].macLayer.queue.transmissionGate[1].offset = 1ms
*.ZonalECU4.eth[*].macLayer.queue.transmissionGate[1].durations = [200us, 1800us]


# link breaks between switches
*.scenarioManager.script = xml("<scenario> \
                                <at t='0.1'> \
                                  <disconnect src-module='s1' dest-module='s2a'/
                                </at> \
                                <at t='0.2'> \
                                  <disconnect src-module='s2b' dest-module='s3b'
                                </at> \
```

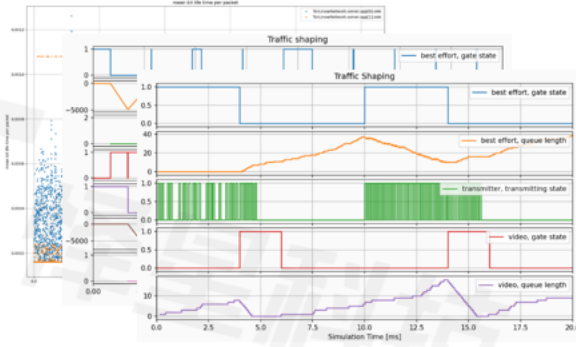
参数配置及数据流建模

# TSN仿真分析 Simulation analysis of TSN

- 开源软件平台（Omnet++&INET框架&TSN模块）

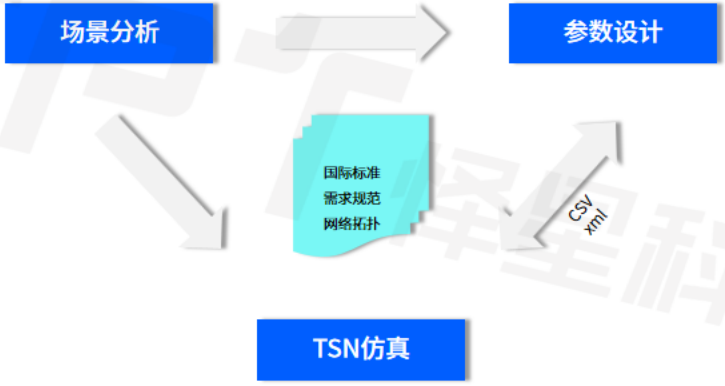


仿真




结果分析（延迟、流量、缓存…）

- TSN设计流程



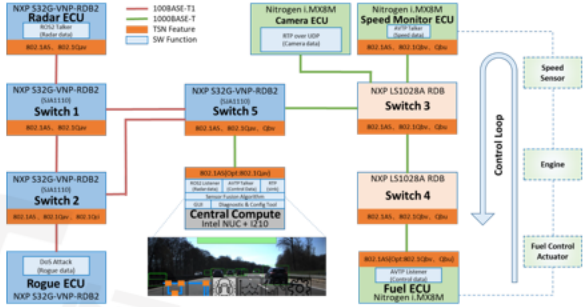
# 04

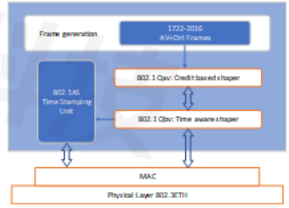
# Demo开发及演示

Development and demonstration

# Demo开发及演示 Development and demonstration

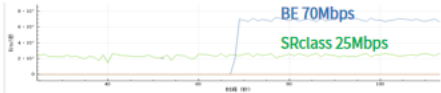
- Demo硬件组成
  - 中央计算单元：
    - 计算模块：intel NUC PC + I210
    - 交换机5：SJA1110
  - Zonal ECU 1&2：
    - S32G-VNP-RDB（S32G+SJA1110）
  - Radar ECU、Rogue ECU
    - S32G-VNP-RDB（S32G+SJA1110）
  - Zonal ECU 3&4：
    - LS1028ARDB（处理器：LS1028A）
  - Speed ECU、Fuel Control ECU、Camera ECU
    - Nitrogen i.MX 8M（CPU：i.MX 8MQuad）

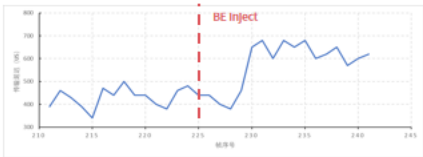


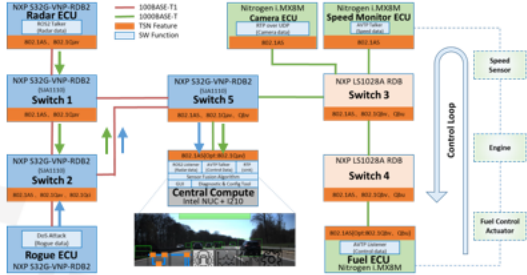



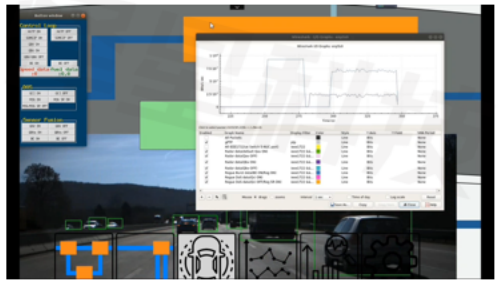
# Demo开发及演示 Development and demonstration

- 传感器融合场景
  - 中央计算单元融合摄像头和雷达ECU数据
  - 802.1Qav协议对雷达数据进行整形
  - Rogue ECU发送干扰流
  - Switch2支持802.1Qci
  - 雷达数据通过802.1CB协议冗余传输(TBD)
  - 网络通过802.1AS协议实现时钟同步冗余(TBD)



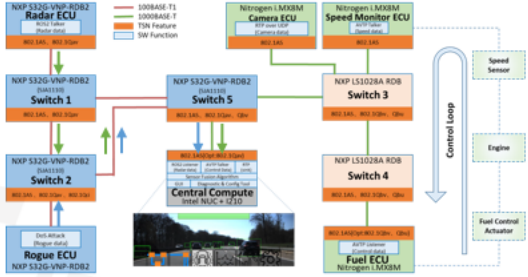



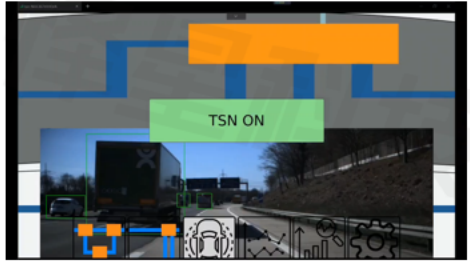



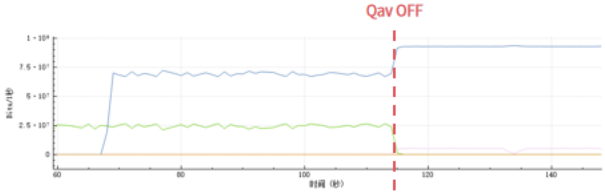



# Demo开发及演示 Development and demonstration

- 传感器融合场景
  - 中央计算单元融合摄像头和雷达ECU数据
  - 802.1Qav协议对雷达数据进行整形
  - Rogue ECU发送干扰流
  - Switch2支持802.1Qci
  - 雷达数据通过802.1CB协议冗余传输(TBD)
  - 网络通过802.1AS协议实现时钟同步冗余(TBD)

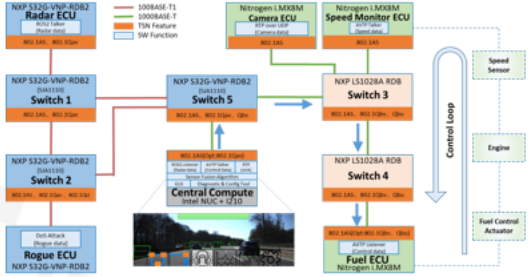


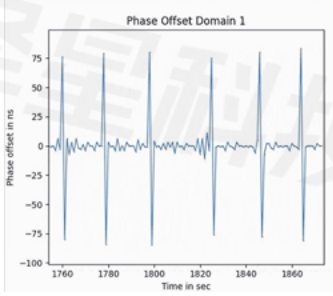



- 关键控制场景
  - 中央计算单元发送控制数据至Fuel ECU
  - 802.1Qbv协议对控制数据整形
  - 网络通过802.1AS协议实现时钟同步
  - Camera ECU发送干扰流


传输延迟




同步精度

# 05
# 总结

Summary

# 总结 Summary

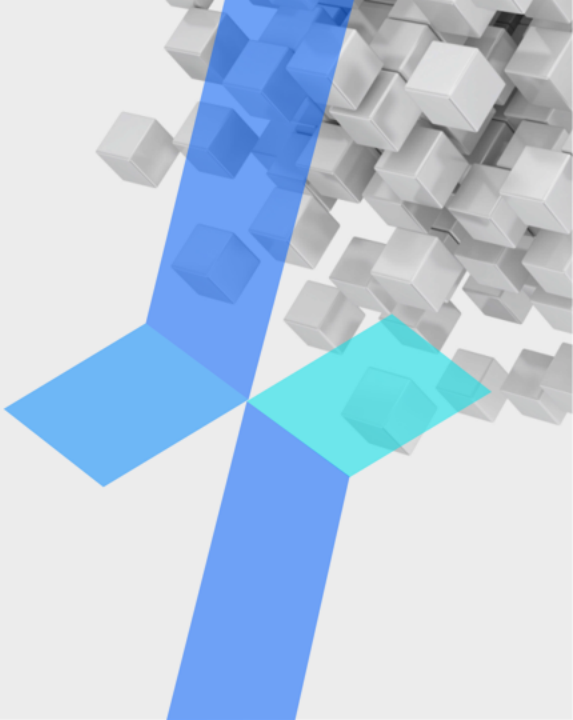
- TSN系列标准庞大但灵活，由设计者根据自己的需要，结合各协议特点，自由组合
- 汽车领域将逐步应用TSN技术，综合考虑成本、性能和易用性
- TSN配置参数多、网络设计复杂，有必要对车载网络行为进行仿真分析

# 怿星科技提供全套车载TSN解决方案 EPT's TSN Solution

| TSN设计 | TSN仿真分析 | TSN开发 | TSN测试 |
| --- | --- | --- | --- |
| TSN协议规范 | 基于Omnet++的GUI建模插件 | Demo开发 | 测试设备 |
| TSN配置规范 | 基于Python的定制化分析插件 | TSN协议栈 | 测试规范 |
| TSN应用场景分析 | | TSN配置工具 | 协议测试 |
| 整形参数计算脚本 | | | 系统测试 |

EPT 恒星科技

# 让每一台智能汽车都有我们的贡献



网站：www.e-planet.cn
电话：+86 21-53393860
邮箱：biz@e-planet.cn
技术咨询：support@e-planet.cn
总部地址：上海市徐汇区田林路487号宝石园20号楼25层

资料下载