摘 要

随着我国客运专线和高速铁路的迅猛发展，列车通信网络的自主研发与设计成为近年国内的研究热点。作为列车内部控制指令和信息数据的传输通道，列车通信网络在列车控制系统中起着极其重要的作用，其工作状态直接关系到列车运行的可靠性和安全性。因此，对列车通信网络进行监控可以及时地了解网络的工作状况，发现网络传输中的错误信息，保障列车安全可靠地行驶。

本文以国家科技支撑计划项目—“高速列车（II 型车）牵引传动和列车网络系统—网络系统虚拟仿真”为背景，以项目中所开发的CRH2型列车通信网络仿真平台作为运行环境，设计并实现了一个基于以太网的 CRH2 列车通信网络仿真平台监控系统。该系统可以监控网络仿真平台的报文传输、令牌循环和网络性能，对列车通信网络进行测试，并完成列车内部控制逻辑的运算。本监控系统对了解列车通信网络的工作原理和研究ARCNET协议的模拟传输有一定的参考价值。

本文首先分析了CRH2列车通信网络仿真平台的组成结构以及ARCNET协议的工作原理，根据仿真平台的特点对监控系统进行需求分析和总体设计；然后对系统的各功能模块进行了详细设计，说明其实现流程。针对CRH2列车通信网络仿真平台中可能出现的多令牌问题，在监控系统告警模块的详细设计中，对令牌异常进行了研究，从报文时序、令牌周期和网络吞吐量 3 个参数的变化定位多令牌异常，并采用通知工作站丢弃令牌的方法解决该问题；最后对监控系统进行了测试，测试结果表明本文设计的监控系统可以正确有效地监控CRH2列车通信网络仿真平台的运行状态，及时发现并解除异常。该系统在大吞吐量的网络环境下运行稳定，达到了项目要求。

关键词：CRH2；列车通信网络；网络仿真；网络监控；ARCNET

Abstract

With the rapid developement of passenger dedicated line and high speed railway in our country, the research and design of train communication network has become a research focus in recent years. As a tranmission path of control command and information data, train communication network plays an extremely important role in train control system.Its working performance is directly related to the reliability and security of train running. Therefore, train communication network monitoring can help know the working status of train communication network, find the error information in network transmission and guarantee the safety and reliability of train.

The background of the thesis is national science and technology supporting project — "High speed train (CRH2) traction drive and train network system — Network System Simulation". Running on the CRH2 EMU communication netwrok simulation platform developed in the project, a train communitaion network simulation platform monitoring system based on ethernet is designed and implemented. This monitoring system can monitor packet transmission, token cycle and network performance, test the train communication network, and perform the control logic of train. It can provide reference for the research of train communicaiton network and ARCNET transmission.

In this thesis the structure of CRH2 EMU communication network simulation platform and working principle of ARCNET is analysed. Requirement analysis and general design is proposed according to the characteristics of the simulation platform. After that the detailed design of each module is presented and realization process is introduced. The possibly problem of extra token is studied and solved according to the variation of packet order, cycle period and throughput. When extra token occuring, a notice is sent to the workstation so as to drop the extra token. Finally the monitoring system is tested. The test result shows that it can monitor the running status of network simulation platform correctly and effectively, and find fault and recover it in time. The monitoring system runs stable in high throughput network envirement and is fit to the project standard.

Key words: CRH2; Train Communication Network; Network Simulation; Network Monitoring; ARCNET

第1章 绪论

1.1 课题的背景与意义

随着我国列车运行速度的不断提升，中国铁路进入跨越式发展阶段。列车通信网络作为支持高速列车和动车组高效运行的重要技术，其自主研发、控制逻辑、运行监控和故障诊断等方面的研究已成为国内的研究热点[1]。我国在列车通信网络方向的研究起步较晚，技术相对较落后，目前虽然引进了国外的车载设备产品，但是对方并没有转让核心技术，因此开发具有自主知识产权的列车网络控制系统，缩短与发达国家列车通信技术的差距，对我国高速铁路建设具有重大意义。

在列车中挂载的设备分散在不同编组的机车车辆中，要使分布于列车中各车辆的设备协调工作，就必须借助于一个分布式计算机网络系统，即列车通信网络来实现[2]。列车通信网络针对列车可靠性要求高、实时性强等特点，对车载设备进行集散式监视、控制和管理，即将列车车辆上的可编制设备通过车厢总线和列车总线互连起来，经过总线进行各设备间的数据传输，实现列车运行过程中牵引、制动等控制信息的传递，车辆运行信息的反馈等功能，完成车辆的控制、状态检测和诊断功能，实现列车控制系统的智能化、网络化和信息化[3,4]。理想的列车通信网络应是可靠传输、实时响应并能做出快速故障恢复的，其网络状况直接关系到列车的整体性能和安全性。因此，对列车通信网络的网络性能、运行状态进行监控可以及时的发现和纠正网络传输中的错误信息，保障列车安全可靠地行驶。

本课题来自国家级科技支撑计划项目——“高速列车（II型车）牵引传送和列车网络系统—通信网络虚拟仿真”，该项目通过运用当前先进的软件仿真技术，根据CRH2型车的网络结构模型在以太网环境下进行纯软件仿真，模拟列车网络信息的产生、传输和控制过程，从而全面掌握CRH2型车的网络系统控制逻辑，为我国自主研制高速列车网络控制系统提供网络性能及功能仿真支持。本文所实现的监控系统作为整个仿真平台的一部分，是在该仿真平台的基础上，监控列车网络信息的传输情况和整个平台的运行状态，通过监控数据和故障信息来协助仿真平台的构建及调试。同时，通过协议分析可以直观的展示网络中传输的报文情况，从底层研究网络协议的分层结构，对了解ARCNET协议的模拟传输有一定的参考价值。

1.2 列车通信网络发展现状分析

世界各国铁道机车车辆生产企业在各自发展过程中使用了不同的列车通信网络技

术[5,6]。目前广泛使用的列车通信网络有 TCN 网络（IEC61375）[7-9]、符合 IEEE 标准的列车通信网络(IEEE1473,包括 TCN 网络和 Lonworks 网络)[10]，以及其他工业网络，如应用于 TGV 高速列车 ARGAT 控制系统的 WorldFIP 网络[11]、应用于日本新干线高速列车的 ARCNET（Auxiliary Resource Computer Network）网络等[12,13]，以上各种标准协议各有其特点和优势[14]，在现代铁路运行中发挥重要作用。

本文中所监控的仿真平台采用 ARCNET 协议进行通信，ARCNET 是一种基于令牌传递(Token Passing)协议的现场总线，其最初是美国 Datapoint 公司在 20 世纪 70 年代末作为办公自动化网络发展起来的[15]。1999 年成为美国国家标准 ANSI/ATA-878.1。该系统具有快速性、确定性、可扩展性和支持长距离传输等特点，非常适合过程实时控制，近年来被广泛应用在各种自动化领域，是一种理想的现场总线技术。ARCNET 的可靠、高速及稳定的性能已被许多工业领域应用，成为工业自动化的重要组成部分[16]。日本的高速列车所使用的列车通信网络主要采用 ARCNET 网络，我国 CRH2 型动车组也使用了 ARCNET 网络技术。

目前我国列车使用的列车通信和控制网络并没有完全的自主知识产权技术，国内还没有成熟产品[17,18]，大部分列车通信网络由国外厂家从产品提供到系统集成完全控制，这对系统后期的调试和维护存在相当大的影响。在 CRH2 列车通信网络研究方面，目前国内对列车通信网络的软件仿真方向研究较少，研究范围局限于 ARCNET 网络的硬件设计，以及对 CRH2 列车网络进行基于 OPNET 等仿真工具的仿真研究等方面[19]，这种利用仿真软件对 ARCNET 网络模型的仿真并没有涉及到网络通信系统核心技术，因此需要构建列车通信网络仿真平台来深入研究列车网络的传输特性。

在列车网络监控方面，目前国内对 ARCNET 通信协议和数据传输方面的监控涉及较少，随着工业以太网技术在列车领域的应用，多媒体信息服务如视频监控、VOD 点播等服务将陆续推出，对列车网络的监控将不仅限于设备状态和环路通断，还需要对网络性能方面的监控进行完善。

因此，构建 CRH2 列车通信网络纯软件仿真平台，模拟列车信息的传输与监控，对研究列车网络的数据产生与传输机制具有重要的理论意义与现实意义。

1.3 课题的主要研究内容与论文组织结构

本文从列车通信网络监控系统的功能需求出发，结合 ARCNET 协议的传输原理，采用以太网环境下数据包截获与分析作为技术路线，设计并实现一个 CRH2 型动车通信网络监控系统。该系统以已开发的基于以太网的 CRH2 型车通信网络仿真平台作为运行环境，可以监控仿真平台的报文传输情况和各节点的运行状态，通过监控数据和故障信

息来协助仿真平台的构建及调试。

第一章介绍了本论文课题的来源、背景以及研究意义，以及目前国内外列车通信网络的发展现状，概述本文的研究内容。

第二章首先介绍监控系统的运行环境——列车网络通信仿真平台，分析其整体结构和工作原理，然后根据仿真平台的特点和要求对监控系统作需求分析，按照功能需求进行模块划分，最后对监控系统作总体设计。

第三章是监控系统的详细设计，分别介绍每个模块的设计原理和实现流程。

第四章对监控系统的运行结果进行分析和测试。

最后总结了本文完成的工作，并提出了下一步的研究方向。

第2章 监控系统需求分析与总体设计

2.1 CRH2列车通信网络虚拟仿真平台

2.1.1 CRH2列车通信网络基本结构

CRH2型动车组的列车编组为4M4T（M为动车，T为拖车），由2个动力单元组成。每个动力单元由2个动车和2个拖车(T-M-M-T)组成，其编组如图2-1所示[20]。

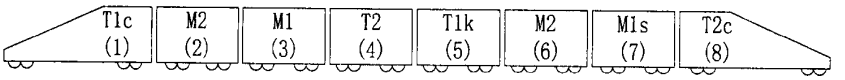

图2-1 CHR2列车编组

CRH2动车组共8节车厢，其中2、3、6、7号车厢为动车，安装有牵引装置，其他车厢为拖车。车厢所挂载的装置分为中央装置和终端装置，通过贯穿列车的光纤双环网络来传送信息。列车的每节车厢均装载有一台信息控制终端装置，车厢内的显示器、读卡器等车载设备通过终端装置实现车载设备信息的控制与传输功能。两端头车(1、8号车)装载有由控制传送部和监视器所构成的车辆信息控制中央装置，拥有管理全列车整体信息和向司控台显示器部发送数据的功能。

CRH2 动车通信网络由列车级网络和车辆级网络组成。列车级网络为连接各车辆的通信网络，以列车运行控制为目的，采用光纤双环网连接各中央装置和终端装置。车辆级网络为单节车辆内部各设备的通信网络，是车辆中的中央装置/终端装置与其挂载的车载设备之间信息交换的通道[21]。

（1）列车总线

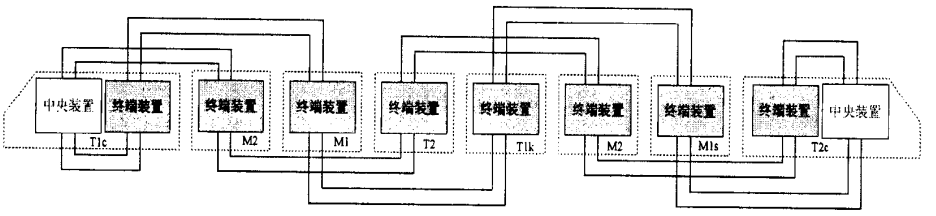

图2-2 列车网络总线结构

列车总线传送线路包括两种类型，即光纤双环网和自我诊断信息传输线(双绞屏蔽线)。光纤双环网以环形拓扑结构连接各车厢的中央设备/终端设备，采用ARCNET作为通信协议，以令牌传递方式进行数据发送，令牌传送周期以10ms为标准，传输速率为2.5Mbit/s。在光纤环网中若向一个方向发送信息后没有检测到应答，可以向另一个方向发送信息进行环绕传送，以此避开故障部位。自我诊断传输线以总线拓扑结构连接全车的中央装置和终端装置，采用HDLC作为通信协议，作为光纤环网传输的冗余备份。

构成列车总线的设备有中央装置、终端装置、显示器、显示控制装置、IC 卡架以及车内信息显示器构成。各装置在列车内的配置情况如下：

表 2-1 信息控制系统设备配置

车辆编号	T1c-1	M2-2	M1-3	T2-4	T1k-5	M2-6	M1s-7	Tc2-8
中央装置	1							1
终端装置	1*[1]	1*[1]	1	1*[1]	1*[1]	1*[1]	1	1*[1]
显示器	2						1	2
显示控制装置	2						1	2
卡架	2							2
车内信息显示器	2	2	2	2	2	2	2	2

*[1]：有模拟输入（AIN）卡

（2）车辆总线

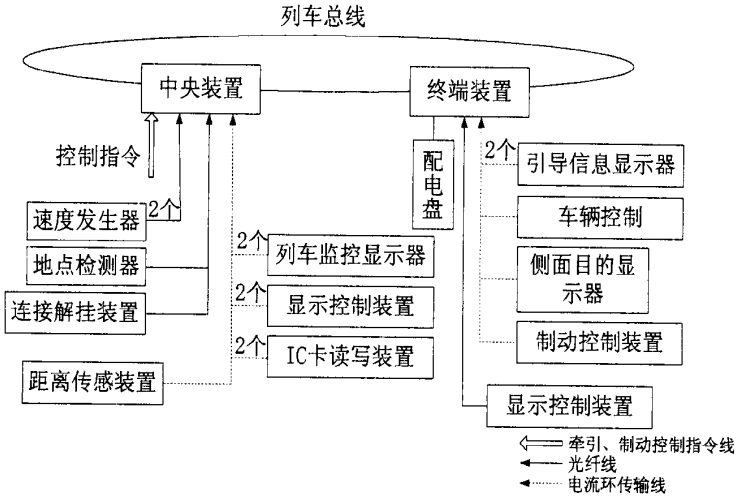

图 2-3 车辆级网络结构

车辆总线指中央装置/终端装置与车辆内设备之间信息交换的通道。中央装置/终端装置与设备之间采用点对点通信方式，牵引变流器、制动控制单元与终端之间采用光纤连接，其他设备与中央装置、终端装置再采用电流环连接，如图 2-3 所示。

车辆内部设备与列车网络节点（中央装置/终端装置）之间的通信协议有多种，包括 20mA 电流环、30mA 电流环以及 HDLC 方式。

连接车厢内部车辆总线的车载设备包括信息显示器、牵引变流器、制动装置等，通过与终端装置的信息交互，实现与车载设备信息的控制与传输功能，另外 7 号车作为列车长车还额外装载有显示控制装置显示器，列车长通过该监控显示器对整个列车的状态进行实时监控以便及时处理列车运行过程中突发的各种故障信息，同时还负责对全车的信息的广播。各车厢的中央/终端装置与各车载设备之间的连接接口遵循光传输、电流环传输、DIDO 等形式构成信息网络节点的联系通道。

（3）工作方式

CRH2 列车启动时，自动为中央装置和终端装置分配对应的逻辑地址，遵循 ARCNET 协议标准在短时间内进行组网，待组网成功，令牌(ITT)从低位地址节点向高位地址节点传递，只有获得令牌的节点才有权发送数据[22]。因此，环网上的所有节点共享总线使用权，使得其网络性能在时间上具有确定性及可预测性[23, 24]。

列车信息控制中央装置与终端装置之间通过由环形网以及备份传输线相连接，具有向左及向右两条传输通道，对控制指令以及其他要求有应答的重要数据信息同时向网络环路的两个方向发送，目的是为了及时避开故障点，保证这些重要数据传输的可靠性，对于监控等信息采用单方向传输，若发信源的光传输节点，没有检测到应答，则向另一方向的传输回路发送信息[25]。当线路发生 2 处以上的故障时，可继续由其他线路进行传送，此外，当环形网络发送故障时可以通过自我诊断传输线传输控制指令，对各设备进行管理控制[26]。

2.1.2 ARCNET 工作原理分析

（1）ARCNET 协议特点

ARCNET 是一种采用了优化的令牌总线协议(IEEE802.4)标准，提供 OSI 参考模型中物理层和数据链路层的服务。其时间上的确定性、组网上的灵活性和传输上的可靠性可以很好的满足列车通信网络对网络性能的要求，目前 CRH2 型动车即采用 ARCNET 传输协议。

ARCNET 采用令牌循环机制，各节点通过令牌来协调网络使用权。节点使用唯一的 MAC 地址标识自己，单个 ARCNET 子网最多可有 255 个节点，ARCNET 支持点对点的定向消息和单点对多点的广播消息。ARCNET 速率为 2.5Mbit/s，使用光纤时的新型 ARCNET plus 速率已增加到 100Mbit/s。

在 ARCNET 网中，由于采用了令牌传递协议，任何节点都不能独占网络，只有在持有令牌后才能成为网络的临时主节点，才能发送一次有限长的消息。一旦消息发送完毕，必须将令牌传递给逻辑环上的下一个节点，收到令牌的节点就成了网络的临时主节点，不断循环最终构成令牌环。ARCNET 使用令牌传递机制来仲裁各网络节点对网络的访问权，不存在竞争，在传递时间上是可预测的（事实上，能够计算出在最坏情况下节点间传递信息所需的时间），即使在网络负载重、流量较大的情况下，也不会造成网络阻塞。

（2）ARCNET 协议帧结构

ARCNET 遵循 IEEE802.4 协议，其帧类型总共有 5 种，分别是：令牌帧(ITT)、缓冲区查询帧(FBE)、应答确认帧(ACK)、应答否认帧(NAK)、数据帧(PAC) [27]。其帧结构定义如下所示：

(a) ITT 帧

ALERT	EOT	DID	DID

(b) FBE 帧

ALERT	ENQ	DID	DID

(c) ACK 帧

ALERT	ACK

(d) NAK 帧

ALERT	NAK

(e) PACK 帧

ALERT	SOH	SID	DID	DID	CP	DATA	CRC	CRC

ARCNET 的 5 种帧结构都包括 ALERT 前导码，前导码由 6 比特间隔的传号（1）组成。传号（1）由正负脉冲组成的双脉冲表示，空号（0）由无脉冲表示。各个帧的说明如下：

（a）令牌帧(ITT)在换网上的各个节点间依次传递，只有获得令牌帧的节点才有权发送数据。EOT 是传输结束控制符（0x04），后跟的两个字节 DID 是目的地址标识符，即后继工作站的逻辑地址。重复使用 DID 是为了增加传输可靠性。

（b）缓冲区查询帧(FBE)用来查询目的节点是否有足够的缓冲区空间来接收源节点将要发送的数据。ENQ 是 ASCII 字符集中的询问字符(0x05)。后面的两个 DID 是所询问缓冲区空间的目的节点逻辑地址。

（c）应答确认帧(ACK)是对源节点发送的缓冲区查询帧(FBE)的确认应答，表示目的节点有足够的缓冲区空间来接收数据。由于发送方式为广播，因此该帧没有 DID 字段，ACK 字段是确认字符(0x06)。

（d）应答否认帧(NAK)是对源节点发送的缓冲区查询帧(FBE)的否认应答，表示目的节点没有足够的缓冲区空间。由于发送方式为广播，因此该帧没有 DID 字段，NAK 字段是否认字符(0x15)。

（e）数据帧(PAC)包含了要传输的数据、错误校验等信息。其中数据信息本身存储在 DATA 字段，数据要发送的目的工作站地址存储在 DID 字段。SOH 是标题开始字符（0x01），SID 是产生和发送数据的源工作站的逻辑地址，DID 为数据要发送至的目的工作站的逻辑地址，CP 为指针字段指示源工作站在存储器中所要传输的数据信息的起始地址，DATA 为可变长度的数据字段，范围为 1 至 508 字节，存储数据信息，CRC 为 2 个字节的校验字段，由源工作站添加，用来保护 DATA 字段。

（3）ARCNET 协议工作原理

ARCNET 网络采用令牌环网的形式实现介质访问控制，逻辑环网中每一个工作站均被指定为唯一的逻辑地址 ID(0-255)，启动网络时，每个节点按照自身 ID 值由小到大

的顺序构成逻辑环路，每个工作站内部均保存并不断跟踪其上一站的逻辑地址以及下一站的逻辑地址，设置方式为各工作站将后继工作站 NID 设为自身 ID 加 1，并按照如下方式设置超时时间（timeout）：

Timeout = 146×（255 - ID）us

逻辑地址最大的工作站最先超时，该站创建令牌帧（ITT）并发送给它的后继节点，如果在 74us 没有接收到 ACK 应答，便认为该 NID 节点不存在，并将 NID 地址加 1，再次向其发送令牌帧。NID 寻址及令牌发送过程按此方式重复，直至收到 ACK 应答，然后被找到的后继工作站按照同样方式寻找自身的 NID 地址。

在所有节点都找到自己的后继工作站后，正常的令牌循环过程开始并可以传输数据。每个工作站拥有平等的优先权，一旦工作站收到 ITT 后，该工作站即可拥有了发送数据的权力，此刻如果该工作站没有数据需要发送，则将 ITT 向下一站传递，若有数据发送，则首先需要询问目的地址是否有足够的缓冲区空间来接受数据帧（PAC），执行这种查询功能的数据帧是 FBE 帧，如果目的节点有足够的空间，则返回 ACK 帧，否则返回 NAK 帧。源节点根据收到的返回是 ACK 还是 NAK 决定是否发送 PAC 帧。

如果 ARCNET 网络传输过程中有新加入工作站或者换网上有工作站退网，整个环网必须重构。如果一个节点在 840ms 内没有接收到 ITT 帧，便向环网上发送由 8 个传号间隔组成的 RECON 图样（后跟一个空号便发送 765 次），RECON 图样持续 2754us，以确保破坏网络上传输的令牌帧并使其丢失。在 ARCNET 环网上 74ms 内任何工作站没有活动后，所有节点均得知环网重构正在发生，于是每个节点将 NID 设置为自身 ID 加 1，并设置超时时间，按照正常启动时的机制进行重构。

节点退网时的情况相对简单，不需要调用重构机制。当某节点退网时，其前驱工作站向其发送的 ITT 帧将不会接收到响应。在 74us 之后，前驱站点便认为其已经从环网上撤离，于是前驱工作站重新设置自身的 NID 加 1，并发送令牌，直至找到可用的后继节点。

2.1.3 CRH2 列车通信网络仿真平台总体结构

根据上述 CRH2 型列车通信网络的总体结构和 ARCNET 协议的传输原理，仿真平台的总体结构设计如下图所示。其中本文所设计并实现的监控系统属于仿真平台的一个软件包，运行于图中的“网络监控站点”，完成对仿真平台运行状态的监控。

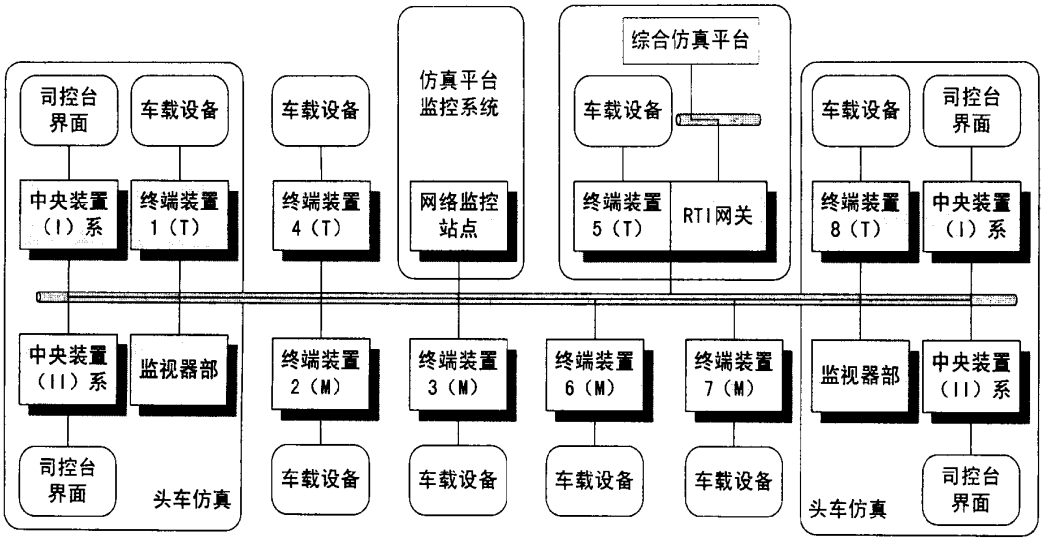

图 2-4 CRH2 列车虚拟仿真平台总体结构

如图 2-4，对于列车级网络的仿真，用 PC 模拟中央装置、终端装置等，各装置间的通信采用 ARCNET 协议；对于车辆级网络的仿真，将终端装置挂载车载设备，双方通过串口连接，车载设备模拟产生列车运行时的状态信息后传输至终端装置。仿真平台的结构说明如下：

（1）头车仿真采用 4 台 PC 模拟，分别模拟中央装置（Ⅰ）系、中央装置（Ⅱ）系、终端装置和监视器部。

（2）5 号车厢仿真采用 2 台 PC 模拟，其中一台模拟终端装置，另一台作为 RTI 网关与“综合仿真平台”进行通信。“综合仿真平台”不同于本文中的“列车虚拟仿真平台”，其功能是模拟产生列车运行时的牵引指令，通过 RTI 网关与以太网连接，将牵引指令封装成 ARCNET 协议的数据帧（PAC）后发送给目的节点。

（3）其余每个车厢的终端装置均用 1 台 PC 模拟，车载设备产生的数据到达终端装置后，封装成 ARCNET 协议的数据帧（PAC）后发送给目的节点。

（4）此外，用 1 台 PC 安装本论文的监控系统，对整个平台的运行状态进行监控。

根据仿真平台的总体设计，共用 15 台 PC 对列车运行进行仿真，1 台 PC 进行监控。仿真平台采用如下方式构建：

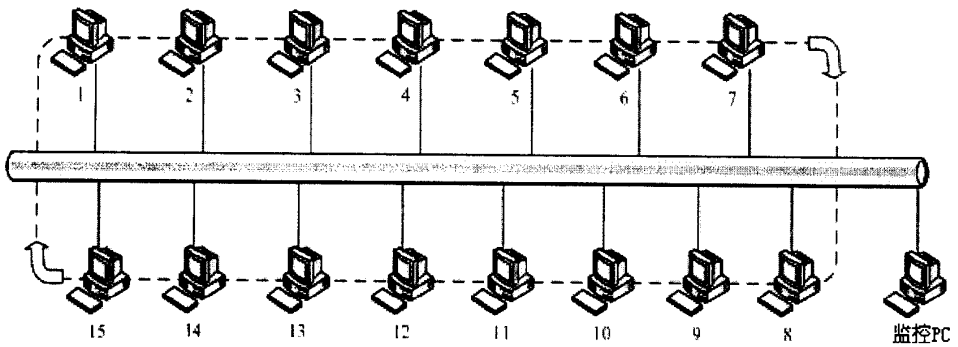

图2-5 II型车通信网络纯软件仿真平台网络拓扑

如图2-5所示，CRH2列车通信网络仿真平台基于以太网实现，在总线结构的以太网上模拟仿真CRH2列车通信网络系统的工作过程(ARCNET的工作过程)，即物理上为总线结构，逻辑上为环形结构。

CRH2列车通信网络纯软件仿真平台的软件层次结构为：通信网络仿真软件作为底层通信基础，其他软件为上层应用。

2.1.4 CRH2列车通信网络仿真平台工作原理

（1）报文定义

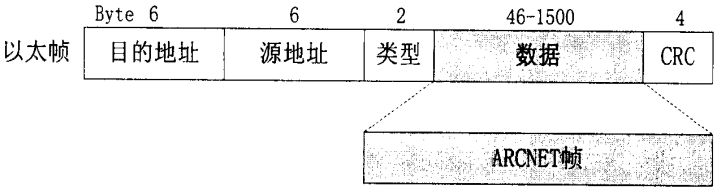

图2-6 仿真平台报文定义

仿真平台的报文构造如图2-6所示，由于在需要在以太网环境下进行通信，因此将ARCNET帧封装到以太帧的数据字段中进行传输，对于ARCNET的不同帧类型，分别定义不同的以太帧类型字段值，接收端通过解析以太帧类型字段值来识别数据的内容。

以太网帧由以下字段组成：目的地址（DA）、源地址（SA）、类型字段（DT）、数据字段（DATA）、循环冗余校验（CRC）字段。目的地址字段由6个字节组成，装载数据包要发送的目的节点的MAC地址；源地址字段也是6个字节的字段，装有发送以太数据包的节点的MAC地址；类型字段由2个字节组成，该字段定义了后续数据的类型；数据字段的长度可以在0—1500字节之间变化，超过1500字节长度的数据包是不符合Ethernet标准的，将会被丢弃，如果数据包字段长度小于46个字节，将会对数据包自动添加0以达到46个字节的有效长度；CRC字段由4个字节组成，用来检测数据传输过程中是否有数据发现错误，如果有错误则丢弃该帧，CRC校验由以太网卡自动进行处理[28]。

（2）传输机制

仿真平台节点间的数据传输机制参照ARCNET协议，当仿真平台开始工作时，按照ARCNET的组网方式进行初始化，初始化完毕后进入令牌循环过程[29]。各个节点的数据信息产生后按照图2-6中的报文定义方式进行封装，当有令牌到达后，经过查询目的地址缓冲区，计算机将封装好的ARCNET数据帧发送出去。数据发送完毕之后，将令牌发送到下一个工作站并重复循环。

2.2 监控系统需求分析

2.2.1 功能需求分析

本网络监控系统运行于CRH2型动车通信网络仿真平台之上，面向的用户群体主要为网络监控人员，另一方面仿真平台的开发人员也可以通过监控系统对仿真平台进行修改和调试。由于仿真平台的数据传输没有可视化界面，对ARCNET传输机制的仿真效果如何、各节点是否已成功入网、报文传输所用的时间是多少等问题用户均无法得知，因此需要根据监控系统来对仿真平台进行评测。Wireshark、Sniffer等抓包软件虽然可以捕获报文并进行分析，但是由于仿真平台传输的报文类型均为自定义的，因此其显示效果并不利于用户观测。此外，还需要在监控软件上开发一些特有的功能来满足仿真平台的运行，该功能无法在抓包软件上实现。

总体来说，抓包软件只能监测、不能控制，并且其监测的参数有限，远达不到设计要求，因此需要开发独立的、针对的、多功能的监控系统来监控仿真平台。以下对仿真平台的工作特点进行分析，将监控系统的功能需求归纳为报文监控、信息过滤、令牌监控、性能监控、节点监控、网络测试、控制逻辑运算、异常告警8个方面。

（1）报文监控

仿真平台传输的报文分为两类：

第一类为ARCNET协议的仿真报文，实现的功能是仿真令牌循环，实现数据传输。该类报文是完成仿真网络通信的关键，包括5种类型：令牌报文（ITT）、应答确认报文（ACK）、应答否认报文（NAK）、缓冲区查询报文（FBE）、数据报文（PAC）。监控系统需要对ARCNET仿真报文的传输进行监控，检验其实际工作情况是否符合标准。

第二类为功能性报文，即为了实现如网络测试功能、控制逻辑运算等其他功能而定义的报文。需要对此类报文进行监控以检验该功能完成的正确性。

（2）令牌监控

令牌循环机制是仿真平台工作的核心，令牌循环周期是衡量仿真平台对ARCNET协议的仿真效果的重要标准，也是评价网络传输能力的指标之一[30]。因此监控系统不仅要能够对令牌循环路径进行跟踪，而且需要精确统计和计算令牌循环的周期。

（3）性能监控

仿真平台的数据传输是否稳定可靠、时延大小等问题决定数据报文的传输效率，进而关系到仿真平台是否达到项目的设计要求，因此监控系统需要对传输结果进行统计，分析当前网络的传输性能，例如时延、传输成功率、差错率等。

（4）信息过滤

在实际运行过程中，由于令牌循环周期很短、吞吐量较大，随着加入节点的增多，

负载的会更加增大，不仅迫使监控软件占用更多的系统资源，也不利于用户在进行协议分析的时候定位报文，因此需要提供数据包过滤功能，将用户感兴趣的数据类型筛选出来，减少CPU负载，提高执行效率。

（5）节点监控

仿真平台中最多有15个节点进行工作，监控人员需要得知当前加入环网的节点数量和其工作状态，因此需要对各节点进行监控，向用户实时展示仿真平台上各节点的工作情况，显示节点加入环网时和退网时其工作状态的变更。

（6）网络测试

CRH2型列车通信网络具有环路测试功能，通过测试来判断节点与链路是否正常。监控系统中的网络测试不仅是为了了解各节点的连接状态，也是对CRH2型列车通信网络仿真的一部分。

（7）控制逻辑运算

控制逻辑运算是列车控制系统的重要组成部分，由于仿真平台以纯软件方式实现，因此对列车控制逻辑的仿真也需要以软件方式来进行，将中央装置的输入指令在监控系统内部进行判断，并将结果发送到各车厢。

（8）异常告警

在仿真平台运行时，监控系统得到的监测参数很多，需要对各类参数逐一检查来判断平台的当前运行状态，不仅不便于用户使用，而且对监控人员的专业要求有一定的限制。因此监控系统需要对监控结果进行综合判断并自动对异常信息发出告警，以便及时纠正错误和排除故障。

2.2.2 性能需求分析

根据仿真平台的运行特点和相关设计要求，监控系统在性能方面应达到以下要求：

（1）准确性

准确性为监控数据与真实结果的符合程度，不仅各类报文数据与到达时间应计算准确、没有误差，而且性能参数和令牌循环周期的计算方式应考虑到在各种异常状态下也能反映其实际结果。

此外，由于仿真平台在局域网环境下运行，传输信道较为理想，因此报文到达时间、时延统计、令牌周期统计、网络时延测试的时间结果数据均应精确到微妙级。

（2）实时性

当事件发生时，监控系统应能够及时响应，具体来说即应及在1秒内响应当前节点状态变更，对于DI报文的接收、处理和DO报文的发送应即时完成，性能参数的更新频率应维持在每秒一次。

（3）高效性

为了达到ARCNET协议标准的令牌循环周期（10ms），仿真平台的令牌循环很快，相应产生的吞吐量很高，为每秒产生3000到5000个数据包。因此要求监控系统能够

快速、高效的处理报文的接收、分析和显示，降低系统资源占用。

根据上述8个主要监控方向和性能需求，采用模块化的开发方法，通过对模块划分保证监控系统各部分间的独立性，降低系统的整体复杂度，提高开发效率。

2.3 系统总体设计

2.3.1 硬件架构设计

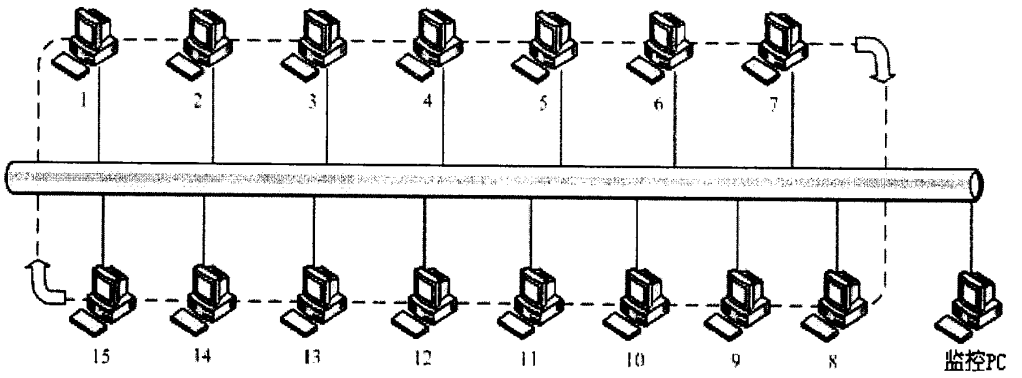

图2-8 监控系统运行环境

如图2-8所示，安装监控软件的计算机以及仿真平台的15台计算机分别与24口交换机连接，监控PC虽然接入仿真平台，但是并不参与令牌循环。要实现监控PC对仿真平台的数据采集，需要设置网卡的混杂模式及端口镜像。

（1）网卡混杂模式设置

在以太网环境中，当网络适配器接收到来自网络的数据包后，首先判断该报文的目的MAC地址是否与本机网络适配器的MAC地址相符，如果目的地址是本站则接收该报文，并对报文中的数据内容进行CRC校验，然后将报文由MAC子层向上提交与LLC（Logical Link Control逻辑链路控制）子层进行处理，如果目的地址不是本站则丢弃该报文。由于运行网络监控系统的主机并不参与ARCNET仿真网络上的令牌循环与数据传输，仿真平台的数据包在到达监控系统所在网卡接口后均将被丢弃，因此为了实现网络监控，首先需要改变网络适配器对数据包的接收模式，使其接受所有来自网络的数据包。

要改变网络适配器的工作模式并接受所有数据，必须过绕过操作系统原有的工作机制，直接访问网络底层，将网卡工作模式设置为混杂（promiscuous）模式。在该模式下，网卡对接收到的所有数据包都产生硬件中断，并通知操作系统对到达本机的数据包进行处理，操作系统收到提醒后直接访问数据链路层，截获数据信息，并由应用层的网络监控程序对数据进行处理，完成监听功能。由于Windows操作系统的高度封装性，程序开发人员很难直接对底层进行编程，需要借助Winpcap开发包提供的底层访

问函数对网卡进行设置，改变其工作模式。

（2）端口镜像

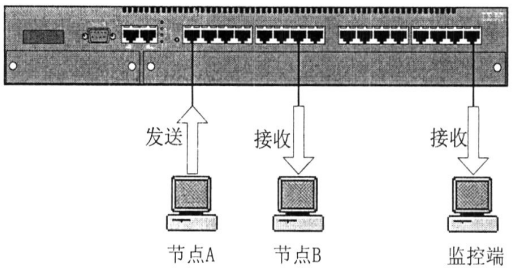

图 2-9 交换机端口镜像

虽然设置了网卡的工作模式，使其可以接受所有到达的数据包，但是由于交换机的转发过滤机制，因此仿真平台上的节点间进行数据通信时，交换机通过 MAC 地址判断后，数据包只会转发给连接目的节点的端口，而连接网络监控系统主机的端口将不会接收到报文，监控系统将无法接收数据。为了监视仿真网络上的所有数据，需要启动交换机的端口镜像功能，把交换机所有端口转发的报文全部复制到连接网络监控系统主机的端口。

设置端口镜像的具体操作为，将交换机的 console 端口与 PC 连接，进入地址为 192.168.0.254 的 web 配置页面后，将交换机的 24 号端口设为镜像端口并连接监控系统 PC，以接收 1-23 端口的数据。

2.3.2 软件结构设计

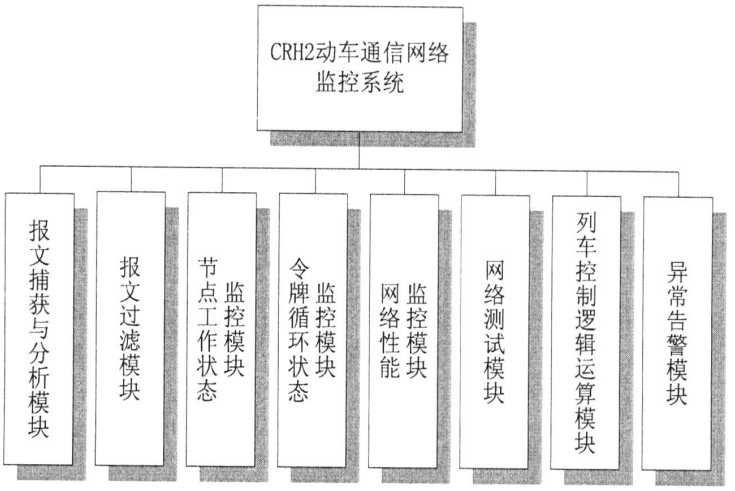

图 2-7 监控系统模块图

结合 CRH2 型动车通信网络监控系统的功能需求，将监控系统划分为 8 个模块，分别是报文捕获与分析模块、报文过滤模块、节点工作状态监控模块、令牌循环状态监控模块、网络性能监控模块、网络测试模块、列车控制逻辑运算模块、异常告警模块。

（1） 报文捕获与分析模块

报文捕获与分析模块是监控系统的数据来源和报文分析的基础[31-34]。该模块的功能为捕获环网上传输的数据包，显示其类型、长度、源地址、目的地址和到达目的缓冲区时间，以16进制打印报文信息并解析内容。

通过报文监控可以对仿真系统作更深入的分析，验证仿真正确性、协助故障排查。按照仿真平台的构建原则，数据包的到达时序应严格遵守ARCNET协议规范，如果出现报文时序错乱则通信机制可能出现错误。打印报文数据可以直观的向监控用户显示报文内部数据的封装情况，通过查看报文实际数据与该报文的帧结构定义是否相符可以检查报文数据的正确性。报文内容解析将报文的数据还原并显示，用户通过内容解析界面可以直观得到报文数据的含义。

（2） 报文过滤模块

本模块的主要功能是对网络到达的数据包进行过滤处理，报文过滤提供的过滤选项包括：节点、方向和类型，在功能上可对仿真平台上15个节点所收、发的主要类型的报文进行过滤并显示。提供的报文类型选项包括令牌报文、性能参数报文、ARCNET数据报文、网络测试报文、DI/DO报文等。

（3） 节点工作状态监控模块

该模块的功能是向用户实时展示仿真平台上各节点的工作状态，显示节点加入环网时和退网时其工作状态的变更。通过对各节点发送的数据进行分析，可以得到该节点的工作情况，本模块将节点的工作状态分为3种：如果节点参与令牌循环并可以正常收发数据，则状态为“正常”；如果节点已经运行了仿真程序但是令牌循环机制出现错误，则状态为“异常”；如果节点没有加入环网，则状态为“离线”。

通过对节点工作状态的监控可以及时发现异常节点，结合对该节点收发的报文进行分析，可以定位异常原因。

（4） 令牌循环状态监控

监控令牌在各个节点间的传递情况，监视并维护令牌循环体系，当某时刻节点收到令牌时，在其上方显示令牌标志。同时，监控环网令牌循环周期，计算其随着负载节点的数量变化而出现的变动。

令牌循环周期是衡量仿真平台对ARCNET协议的仿真效果的重要标准，也是评价网络性能的指标之一。此外，该模块可及时发现令牌传输异常，在正常情况下仿真平台上应该只有一个令牌参与循环，如果出现多余的令牌监控系统应能够及时发现并发出告警，如果在全网内出现令牌丢失应及时产生新令牌来引导并恢复令牌循环。

（5） 网络性能监控模块

网络性能监控对各节点的性能进行统计计算[35]，所监控的参数包括：平均时延；最大时延；最小时延；传输成功率；丢包率；差错率；发包数；收包数；发送字节数；接收字节数；吞吐量。对节点性能参数作累加和平均计算可以得到整个网络的整体性

能参数。各参数含义如下：

表 2-2 网络性能参数含义

参数	含义
时延	源节点从发送完毕 ARCNET 数据报文（PAC）开始，到接收到目的节点返回的 ACK 响应报文的时间
成功率	成功发送 ARCNET 数据报文的概率
差错率	ARCNET 数据报文的误码校验正常，并被目的节点正常接收的概率
丢包率	由于目的节点的缓冲区不足，源节点未能发送 ARCNET 数据报文并将其丢弃的概率
收发包数、字节数	均以 ARCNET 数据为统计对象
吞吐量	仿真平台每秒产生的数据包数量（packets/second）和比特数(Mbits/second)。

需要说明的是，“吞吐量”将仿真平台上的所有类型的数据包都纳入统计范围，反应的是整个网络的流量情况；而“发包数”、“收包数”、“发送字节数”、“接收字节数”均只统计 ARCNET 数据报文（PAC），反应的是有效数据的传输情况。

（6） 网络测试模块

监控系统的网络测试以 CRH2 列车通信网络的环路测试为参考，测试的主要目的是测试监控端与各节点的时延大小。如果目的节点可达并且时延符合标准，那么该节点在物理上是无故障的。

CRH2 列车通信网络中环路测试是双向的，通过接收各节点的应答可以定位故障源来自节点还是链路，由于本项目中仿真平台的拓扑结构在物理上为总线型，与 CRH2 列车网络不同，因此不具有测试链路故障的能力，以节点时延测试为主。

（7）列车控制逻辑运算模块

控制逻辑即列车控制网络的工作逻辑，列车运行时在中央装置输入操作指令（操作指令如打开/关闭车厢广播及室内照明、上升/下降受电弓、电源切换/复位等），通过列车内部的工作逻辑输出结果并下发到相应车厢，实现列车控制的遥控操作[36, 37]。

本模块以软件实现列车控制系统的工作逻辑，对包含有控制指令的 DI 报文进行捕获及运算，将计算结果封装成 DO 报文并发送。

（8）异常告警模块

异常告警模块结合报文监控、令牌监控及网络性能监控的监控结果，根据异常特征进行综合判断，并作出相应告警。告警信息分为 3 类：令牌丢失、令牌过多、报文时

序错乱。平台开发人员可以根据异常信息作出相应调整，对于令牌告警，监控系统可以进行通过控制令牌生成与销毁来解除异常。

此外，该模块也是整个平台仿真效果的评价标准，如果仿真平台可以较长时间无故障运行，则仿真平台的实现是比较成功的。

本系统的主要功能是从仿真平台上捕获数据报文，对报文进行过滤及保存后将数据传递给报文分析部分，通过对报文进行分析可以得到监控系统所需的数据，最后将监控数据传入UI界面显示。软件总体结构分为3大部分：用户界面、报文收发、数据分析。其中用户界面不仅包括对监控系统进行的基本配置，如选择网卡、设置过滤规则等，还包括监控数据的显示；报文分析包括对报文基本信息的分析、令牌帧的分析、性能参数帧的分析等，是监控系统的核心部分；报文捕获则是对仿真平台的数据包进行捕获及保存，为数据分析提供基础。监控系统软件总体设计如图2-10所示：

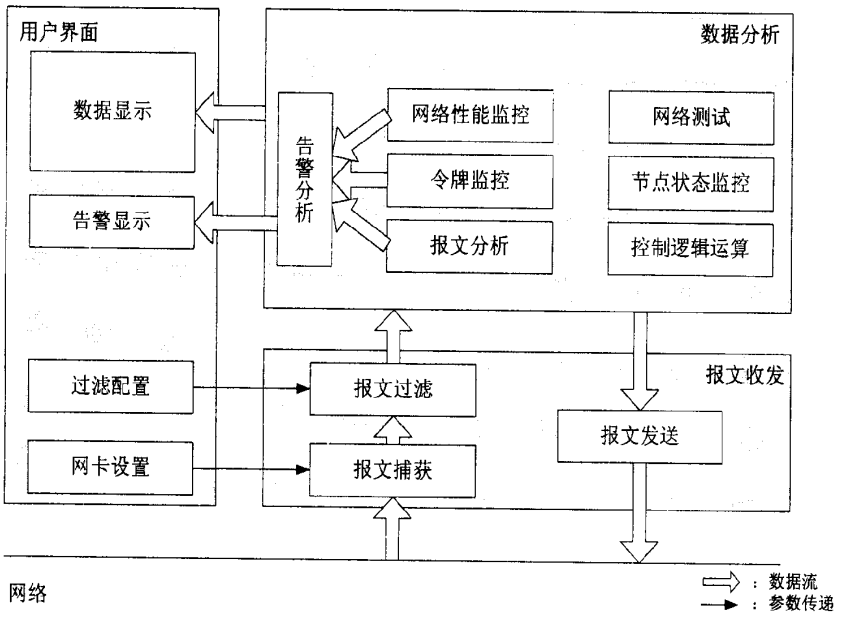

图2-10 监控系统总体结构

（1）系统运行时，用户首先通过界面进行网卡设置和过滤配置，网卡设置为报文捕获提供捕获句柄，过滤配置为报文过滤模块提供过滤表达式，如果对过滤配置不进行操作将采用默认值。在初始化成功后，报文捕获模块将捕获从网络上到达的数据包，将其保存并提交给数据分析部分。

（2）数据分析对捕获模块提交上来的数据包进行分析，该部分包括7个模块。其中网络测试、控制逻辑运算和网络性能监控这3个模块需要通过向网络发送报文并对接收的返回信息进行分析，异常告警模块对网络性能监控模块、报文分析模块、令牌监控模块的监控结果进行综合分析。下表详细说明了数据分析部分的7个模块所接收和发送的报文类型：

表 2-3 各模块收发报文类型

模块	封装并发送	接收并分析
节点状态监控	N/A	所有报文
报文基本信息分析	N/A	所有报文
网络性能监控	性能参数请求报文	性能参数应答报文
令牌监控	N/A	令牌报文
控制逻辑运算	DO 报文	DI 报文
网络测试	测试请求报文	测试应答报文
异常告警	令牌报文/令牌销毁报文	N/A

（3）在数据分析完成之后显示监控结果。各模块分别将数据分析结果通过 MFC 控件在相应的界面中显示。

2.4 本章小结

本章首先介绍列车网络通信仿真平台，分析其整体结构和工作原理，明确了监控系统在列车网络通信仿真平台中的位置，然后根据仿真平台的特点和要求对监控系统作需求分析，按照功能需求进行模块划分，并对监控系统作总体设计。

第 3 章 监控系统详细设计

3.1 报文捕获与分析模块

3.1.1 报文捕获原理

报文捕获与分析模块使用 Winpcap 技术来实现。Winpcap 是一个应用于 Win32 平台的，用以捕获网络数据包并进行分析的开源库，为 win32 应用程序提供了访问网络底层的能力[38-40]，其捕获原理如图 3-1 所示。

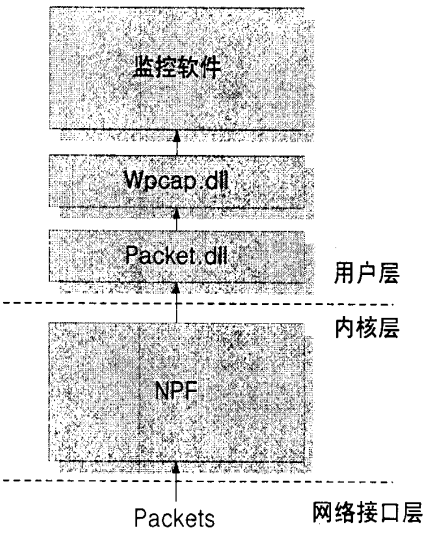

图 3-1 监控系统报文捕获原理

Winpcap 体系结构包扩 3 个工作层次：

（1）网络接口层。

即仿真平台硬件设备工作的层次，网络数据包通过硬件设备接口进入系统内核层。通过设置交换机的端口镜像功能，可以将仿真平台上所有节点收发的数据包都转发到监控计算机上，仿真平台上负载越多需要监控的数据量也越大。

（2）系统内核层。

系统内核层的核心是包过滤驱动程序 NPF(Netgroup Packet Filter)，其主要功能是高效的捕获和过滤数据包，在数据包上附加时间戳、包长度等信息，并将这些数据包传递给上层应用模块。低级动态链接库 packet.dll 在 win32 平台上给开发者提供了与 NPF 的一个通用公共接口。

用户级的 wpcap.dll 是独立于操作系统的高级系统链接库。它也工作在用户层，也给开发者提供了一个通用接口，但是相对于 packet.dll，它提供了更加高层、功能更

加强大、抽象的函数调用。它通过调用 packet.dll 提供的函数生成，包括过滤器生成等一系列可以被用户级调用的高级函数，并且还有诸如数据包统计及发送等功能。

（3）用户层

用户层包含数据包监控模块、节点工作状态监控模块、网络性能监控模块等。通过对底层捕获的数据包进行类型解析后提交到不同的处理模块。

Winpcap 通过以上 3 个层次逐级向上提交报文数据。同时为了方便对仿真平台的历史数据进行分析，需要提供在捕获的同时对数据包进行保存的功能。Winpcap 的数据包保存函数提供了直接在内核模式下保存报文到硬盘上的功能

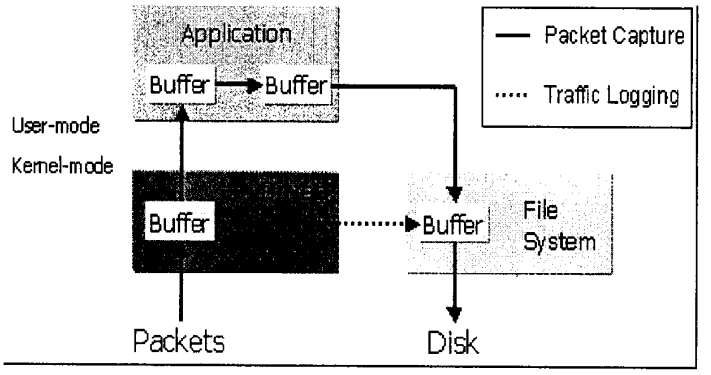

图 3-2 Winpcap 报文保存方式

传统方式中保存数据包的路径如图 3-2 的黑色箭头，每个数据包都被复制了几次，通常分配了四个缓冲区：一个处于捕获驱动里，一个在捕获数据的应用程序里，一个在应用程序用来进行写文件的标准输入输出函数里，最后一个在文件系统里。当处于内核级的 NPF 流量记录功能被启用时，捕获驱动直接同文件系统进行通信，因此数据包保存到磁盘里的路径是红色箭头所指向的：仅需要两个缓冲区，并且只需复制一次，系统调用的次数锐减，因此性能更好。当前转储到磁盘的实现是广泛使用 libpcap 格式。它可以在转储前过滤数据包，以便只选择那些需要保存到磁盘的数据包。

3.1.2 Winpcap 安装与配置

使用 Winpcap 编程的准备工作如下：

（1）下载 Winpcap 驱动程序和相应版本的 WpdPack 包(Developer’s Pack)。WpdPack 包中含有 Winpcap 编程所需的头文件和 lib 库文件，以及一些示例程序和帮助文档。下载地址 htti://www.winpcap.org/install/default.htm。当前最近版本为 Winpcap4.1.2。

（2）根据提示安装 Winpcap 驱动程序。

（3）将 WpdPack 包解压到某个自定义的目录下，在该目录中会看到 dos、Include、lib、Examples 等文件夹。

（4）在VC中设定Include目录以及Library目录。具体做法是：打开VC后，点击“Tools->Option->Directories”，在include files中添加……\wpdpack\Include目录(“……”为步骤3中自定义的目录)，在Library files中添加……\wpdpack\Lib目录。

（5）点击“Project->settings->Link”，在Object/library modules中添加wpcap.lib。

此外，通过以上配置之后，在编写报文捕获程序时需要在程序中加入包含头文件语句#include “pcap.h”来调用Winpcap函数。

3.1.3 报文分析原理

在执行报文捕获的pcap_next_ex()函数后，系统读取下一个可用报文并将获得的报文信息保存在header指针和数据指针pkt_data中[41]。其中header是pcap_pkthdr结构体指针，其数据结构定义如下：

```
struct pcap_pkthdr {
    struct timeval ts;      /* time stamp */
    bpf_u_int32 caplen;     /* length of portion present */
    bpf_u_int32 len;        /* length this packet (off wire) */
};
```

ts是捕获时间戳，表示报文到达缓冲区的时间，时间戳由秒数tv_sec和微秒数tv_usec组成，timeval的数据结构定义如下：

```
struct timeval {
        long    tv_sec;         /* seconds */
        long    tv_usec;        /* and microseconds */
};
```

caplen表示数据包在捕获时的长度，len表示数据包在发送端发出时的长度。在这里我们使用len变量，因为它更真实的描述了数据包的实际长度。pkt_data是const u_char类型指针，指向捕获包的数据部分首地址，根据以太网的帧结构，从pkt_data指向的首位地址作不同的偏移可以得到相应位置的数据。因此，在作报文分析时，可以解析出该报文的源地址、目的地址和协议类型。

从缓冲区读取报文成功之后调用GetSourNumber()和GetDestNumber()函数来将报文中的源MAC地址和目的MAC地址解析为逻辑地址，同时由于pcap_pkthdr结构体指针中存放了时间戳、长度等信息，因此可以直接解析并显示。在显示数据时，由于数据量过大，报文条目很多，因此需要不断的对显示结果进行刷新，而且为了方便用户对报文内容的查看，需要保存报文的内容指针，这样在用户点击某行报文时可以即

刻显示。在本软件中设置每显示 10 万条报文信息后进行刷新并释放内存。

仿真平台的报文类型共 18 种，表 3-1 列出了帧类型和其帧结构中类型字段的对应关系：

表 3-1 报文类型与以太帧类型字段对应关系

报文类型	以太帧类型字段	报文类型	以太帧类型字段
令牌(ITT)	0x0100	RECON 重构帧	0x6100
应答确认(ACK)	0x0300	性能参数请求帧	0x0a00
应答否认(NAK)	0x0400	性能参数应答帧	0x0b00
缓冲区查询(FBE)	0x0200	测试请求帧	0x1600
数据帧(PAC)	0x5y00	测试应答帧	0x1700
销毁令牌报文	0x2300	令牌重构帧(consITT)	0x0e00
MAC 地址请求帧	0x0900	令牌重构应答帧 (consITTACK)	0x0f00
MAC 地址应答帧	0x1000	DI	0x1c00
对时帧	0x1400	DO	0x1d00

对数据包的以太帧头部类型字段按照上表进行比较，可以得到报文类型。下列出了几种主要报文结构定义：

（1）令牌报文

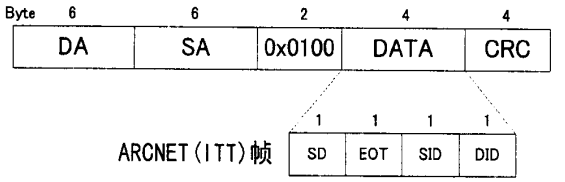

图 3-3 仿真平台令牌(ITT)结构定义

图 3-3 为仿真平台使用的令牌报文，其以太帧类型字段 Type = 0x0100，该报文主要是仿真 ARCNET 标准令牌报文的功能。为了在实际仿真过程中明确区分发送令牌的工作站和令牌传递的下一工作站，修改了 ARCNET 标准令牌帧的第一个 DID 字段为 SID 字段存储发送令牌工作站的逻辑地址。

（2）FBE 报文

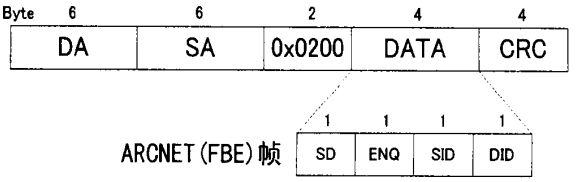

图 3-4 仿真平台缓冲区查询报文(FBE)结构定义

图 3-4 为以太网环境下的缓冲查询帧(FBE)的帧格式，其中以太帧类型 Type = 0x0200，该报文主要是仿真了 ARCNET 标准报文的缓冲查询帧的功能，向目的工作站

查询是否有足够的内存空间。为了在实际仿真过程中明确发送 FBE 包问的来源，修改了 ARCNET 标准 FBE 帧的第一个 DID 字段为 SID 字段以存储发送 FBE 帧工作站的逻辑地址。

（3）ACK 报文

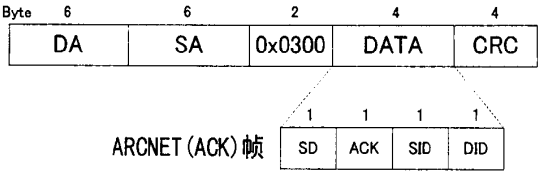

图 3-5 仿真平台应答确认报文(ACK)结构定义

图 3-5 为以太网环境下的 ACK 确认报文帧的帧格式，以太帧类型字段 Type = 0x0300，该报文仿真了 ARCNET 标准报文的确认帧的功能，在 ARCNET 标准的 ACK 报文帧结构的基础上添加了源工作站的地址字段 SID 以及目的工作站的地址字段（DID）。

（4）NAK 报文

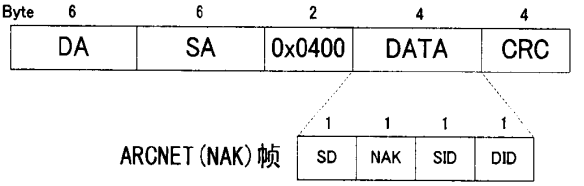

图 3-6 仿真平台应答否认报文(NAK)结构定义

图 3-6 为以太网环境下的 NAK 否定报文帧的帧格式，以太帧类型字段为 Type=0x0400，该报文仿真了标准 ARCNET 否定帧的功能。在 ARCNET 标准的 NAK 报文的帧结构的基础上添加了源工作站的站地址字段 SID 以及目的工作站的站地址字段（DID）。

（5）ARCNET 数据报文（PAC）

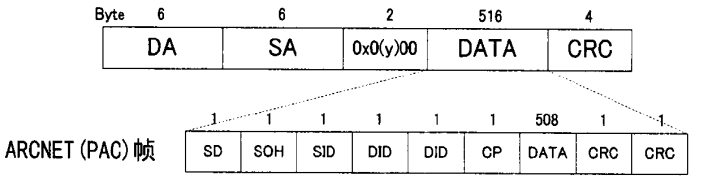

图 3-7 仿真平台数据报文(PAC)结构定义

图 3-7 为仿真环境下的数据帧的定义，类型为 Type= 0x5y00，根据 y 的值标识网络传送的各种数据报文。报文仿真的标准的 ARCNET 数据帧格式，各类型的报文的定义遵循标准的车载设备报文规范。

3.1.4 实现流程

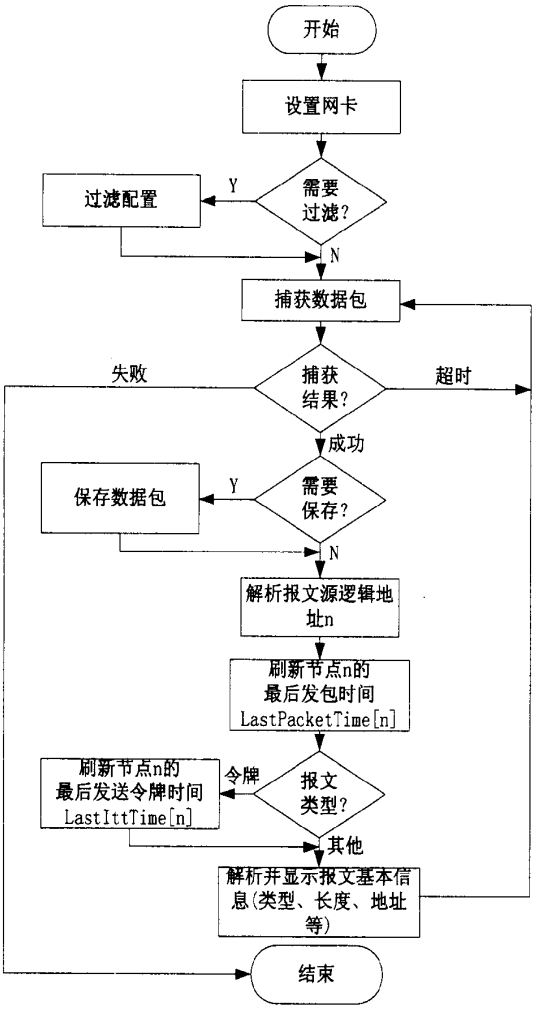

图 3-8 数据包捕获及分析流程

图 3-8 为数据包捕获及分析线程流程图。首先由用户选择捕获所用的网卡，并在过滤器配置界面中选择所需要捕获的报文类型，如果过滤器不进行设置将使用默认值并捕获所有报文。在设置完毕可用网卡和过滤规则后，调用 Winpcap 函数 pcap_next_ex()来捕获数据包，如果捕获成功程序将执行 3 个主要的操作：

（1） 保存数据包。该操作是为了方便以后对历史记录进行分析；

（2） 记录每个节点的最后发送数据包的时间 LastPacketTime 和最后发送令牌报文的时间 LastIttTime。这两个数组是捕获模块与节点状态监控模块之间的接口，节点状态的评估以其作为参考；

（3） 解析报文基本信息。包括报文类型、源地址、目的地址、报文长度、时间戳等；

报文解析

编号	类型	源地址	目的地址	时间(S)	长度(字节)	以太帧内容			
22	令牌(ITT)	7	8	0.005126	60	0023ae9df58b	001372813c19	0100	fc04070800 00
23	应答确认(ACK)	8	7	0.005484	60	001372813c19	0023ae9df58b	0300	fc06080700 00
24	令牌(ITT)	8	1	0.005815	60	0023ae9e00d0	0023ae9df58b	0100	fc04080100 00
25	应答确认(ACK)	1	8	0.006180	60	0023ae9df58b	0023ae9e00d0	0300	fc06010802 00
26	令牌(ITT)	1	2	0.006515	60	0023ae98cadf	0023ae9e00d0	0100	fc04010202 00
27	应答确认(ACK)	2	1	0.006869	60	0023ae9e00d0	0023ae98cadf	0300	fc06020102 00
28	令牌(ITT)	2	3	0.007205	60	00188b1225e3	0023ae98cadf	0100	fc04020302 00
29	应答确认(ACK)	3	2	0.007287	60	0023ae98cadf	00188b1225e3	0300	fc06030202 00
30	令牌(ITT)	3	4	0.007362	60	0023ae9decf7	00188b1225e3	0100	fc04030402 00
31	应答确认(ACK)	4	3	0.007716	60	00188b1225e3	0023ae9decf7	0300	fc06040305 00
32	令牌(ITT)	4	7	0.008060	60	001372813c19	0023ae9decf7	0100	fc04040705 00
33	应答确认(ACK)	7	4	0.008133	60	0023ae9decf7	001372813c19	0300	fc06070400 00
34	令牌(ITT)	7	8	0.008207	60	0023ae9df58b	001372813c19	0100	fc04070800 00
35	应答确认(ACK)	8	7	0.008562	60	001372813c19	0023ae9df58b	0300	fc06080700 00
36	令牌(ITT)	8	1	0.008907	60	0023ae9e00d0	0023ae9df58b	0100	fc04080100 00
37	应答确认(ACK)	1	8	0.009261	60	0023ae9df58b	0023ae9e00d0	0300	fc06010802 00
38	令牌(ITT)	1	2	0.009595	60	0023ae98cadf	0023ae9e00d0	0100	fc04010202 00
39	应答确认(ACK)	2	1	0.009950	60	0023ae9e00d0	0023ae98cadf	0300	fc06020102 00
40	令牌(ITT)	2	3	0.010283	60	00188b1225e3	0023ae98cadf	0100	fc04020302 00
41	应答确认(ACK)	3	2	0.010367	60	0023ae98cadf	00188b1225e3	0300	fc06030202 00
42	令牌(ITT)	3	4	0.010452	60	0023ae9decf7	00188b1225e3	0100	fc04030402 00
43	应答确认(ACK)	4	3	0.010806	60	00188b1225e3	0023ae9decf7	0300	fc06040305 00

图 3-9 报文基本信息数据

3.2 报文过滤模块

3.2.1 过滤表达式生成

过滤模块的实现方式为采用 MFC 编程创建过滤配置界面，过滤配置界面提供节点、传输方向和类型的选项，用户进行选择后系统产生相应的过滤表达式，并将过滤表达式应用的捕获线程中，实现过滤功能。

数据包过滤器决定一个进来的数据包是否要被接受和拷贝给监控程序，由于仿真平台的吞吐量很大，因此一个多功能和高效率的数据包过滤器起着关键性的作用。数据包过滤器是一个应用在数据包上，最终返回布尔值的函数。如果函数返回 true 的话，捕获驱动会把数据包拷贝给应用程序，否则就直接丢弃数据包。应用程序需要一个用户定义的过滤并使用 wpcap.dll 将它们编译成一个 BPF 程序[42]。然后，应用程序在内核里注入过滤器，从这点上看，每进入一个数据包，程序就执行一次，且只有符合条件的数据包被接收。

在 tcpdump 中，过滤表达式作为筛选条件对接收的报文进行存储。如果没有设置表达式，就存储网络上的全部报文，否则只存储过滤表达式的值为 true 的报文。过滤表达式由一个或多个原语(primitive)组成。原语通常由标识(id，名称或数字)，和标识前的一个或多个限定语(qualifier)组成[43]。限定语有三种类型。

(1) Type:类型限定语，指出标识名称或标识数字代表的类型，可用的类型有 host、net 和 port。

(2) Dir：方向限定语，指出相对于标识的传输方向(数据是传入还是传出标识)。可用的方向有 src、dst、src or dst 和 src and dst。

(3) proto[expr:size]：服务限定语。Proto 是 ether、fddi、ip、arp、rarp、tcp、udp 或者 icmp 之一，同时也指出了下标操作的协议层。expr 给出字节单位的偏

移量，该偏移量相对于指定的协议层。

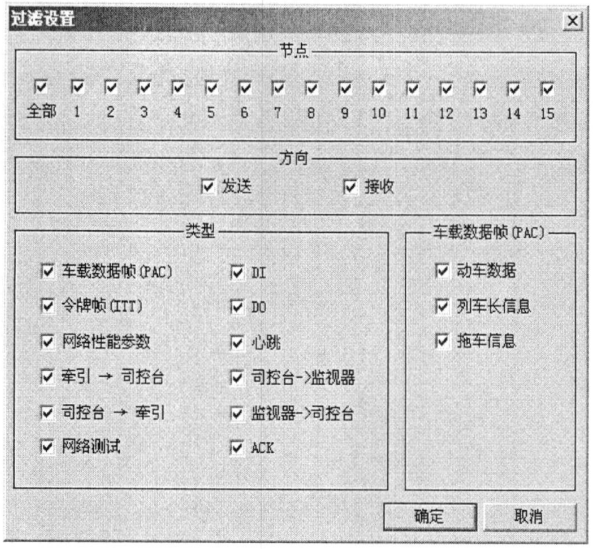

图 3-10 过滤配置界面

在本系统中，过滤配置包括地址、方向和类型 3 个选项。过滤表达式即由这 3 个选项的选择结果来生成。

（1）地址过滤

监控系统后台维护每个节点的逻辑地址与其 MAC 地址的对应关系，用户在选择地址之后，会按照地址映射关系获得该节点的 MAC 地址。

（2）方向过滤

tcpdump 中以 src、dst、host 分别用来表示发出的、接收的、以及收发的含义。结合地址选择，节点和方向的过滤表达式以“ether src 00:0a:e4:33:c5:48”格式产生，表示从 MAC 地址为 00:0a:e4:33:c5:48 节点发出的以太帧。针对用户在界面上的不用选择，用相应的过滤原语进行替换。

（4） 协议过滤

由于以太帧类型字段相对于其首部地址的偏移量为 12，参考定义的仿真平台帧格式，类型过滤表达式以“ether[12]==0a”格式产生，表示第 13 个字节为 0a 的以太帧。

两个地址过滤表达式和协议过滤表达式以“and”连接得到最终表达式。在多选情况下，更复杂的过滤条件可以通过“and”、”or”或“not”来组建。

3.2.2 实现流程

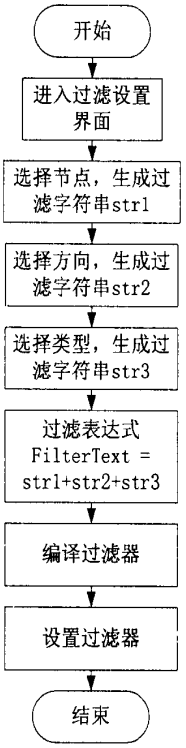

图 3-11 报文过滤模块实现流程

Winpcap 向开发者提供了数据包过滤引擎，使用 pcap_compile() 和 pcap_setfilter()函数来实现。

过滤过程采用 winpcap 提供的内核级过滤机制 NPF (Netgroup Packet Filter)。首先由用户通过界面选择特定的过滤条件，如某节点发送或接受的某种报文，系统根据用户的选择生成相应的过滤表达式(expression)，过滤表达式由 pcap_compile()函数编译成内核级的字节码，该字节码可被过滤引擎所解释。然后 pcap_setfilter()将过滤器与捕获会话相关联，并应用到已经打开并在使用的网卡上。过滤时，NPF 会检查从网络到达的所有数据包的相应字段的值，把符合过滤条件的数据包转存到缓冲区。

3.3 网络性能监控模块

3.3.1 网络性能监控实现原理

监控节点的性能表现。参数为：平均时延；最大时延；最小时延；传输成功率；丢包率；差错率；发包数；收包数；发送字节数；接收字节数。对节点性能参数作累加和平均计算可以得到全网的性能参数。

对网络性能的统计可以采用以下两种方式：

（1）监控端通过对各节点所收发的数据进行统计分析，进而获得各节点的性能参数。以传输时延的统计为例，当监控端接收到节点A发送给节点B的数据帧（PAC）时记录系统时间t1，当收到节点B返回的ACK帧时记录系统时间t2，t2与t1的时间差值即为当前数据传输的往返时延。此方式可以监测到各节点的收、发数据包数与字节数，以及其时延大小，所有参数的统计可以由监控系统独立完成，缺点无法监测到各节点的丢包率、差错率和传输成功率。

（2）各节点分别统计自己的性能参数，将性能数据按照以太帧格式封装后发送给监控系统，监控系统直接解析性能数据。该方式虽然需要各节点协同监控端来完成，但是可以分散统计和计算数据的压力，降低了代码的复杂度，对性能参数的监测也更加全面。

因此，网络性能监控模块的实现方法为，当监控系统需要某个节点的性能数据时，向目的节点发送性能参数请求报文，节点收到该报文后应立刻返回性能参数应答报文，监控系统通过解析报文内容获取数据。

各节点的主要性能参数统计按照如下方式：

（1）丢包率统计计算：丢包率用来衡量仿真系统在传输应用数据报文的过程中所丢失的数据包数，设置长整型全局变量发送包数 PacketSentNum 和丢失的包数 PacketDropNum，源工作站发送端每发送一个数据报文 PacketSentNum 自动加1，接收端接收响应 FBE 报文的 NAK 报文时，认为目的工作站没有足够的内存空间接收存储当前数据包，此时源工作站做丢包处理，PacketDropNum 自动加 1，丢包率即为 PacketDropNum/PacketSentNum。

（2）误码率统计计算：误码率通过统计错误发送的应用数据报文来反应当前网络环境的干扰，设置长整型全局变量 PacketErrNumb 表示网络传输过程中的错误传输的数据报文数，当源工作站接收到目的工作站针对数据报文响应来的 NAK 报文时，认为数据传输错误，PacketErrNum 自动加 1， 误码率即为 PacketErrNum/PacketSentNum。

（3）时延的统计计算：此处统计的时延为端到端时延，即从发从完毕 PAC 数据报文开始，到接收到目的节点返回的 ACK 应答为止的时间。

对节点性能参数进行累加和平均可以获得全网的综合性能参数，其计算方法如下：

表 3-2 全网综合性能参数计算方法

综合性能参数	计算方法
全网发送的总包数	Σ站点的发送的包数
全网发送的总字节数	Σ站点的发送的字节数
全网接收的总包数	Σ站点的接收的包数
全网接收的总字节数	Σ站点的接收的字节数
全网平均包时延	Σ站点包的平均时延 / 站数

最大时延	max (站点包最大时延)
最小时延	min (站点包最小时延)
全网成功发送概率	Σ（站点的发送包数×站点成功发送的概率）/ 站点数
全网丢包概率	Σ（站点的发送包数×站点的包丢失概率）/ 站点数
全网差错率	Σ（站点的发送包数×站点的差错率）/ 站点数
吞吐量	全网每秒产生的包数（pps）及比特数(Mbps)

3.3.2 性能监控报文定义

（1）性能参数请求报文

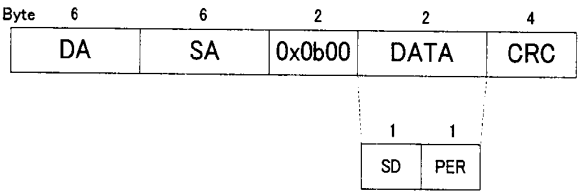

图 3-12 性能参数请求报文结构定义

图 3-12 为性能参数请求报文，以太帧类型字段为 0x0b00，性能参数请求报文的发送频率为每秒发送一次，在 MFC 定时器 ontimer()中执行发送函数，根据逻辑地址与 MAC 地址映射关系获得目的节点的 MAC 地址，性能参数请求帧向环网上的所有节点发送。

（2）性能参数应答报文

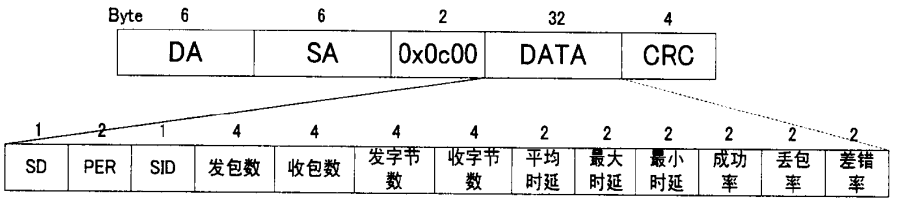

图 3-13 性能参数应答报文结构定义

图 3-13 为性能参数应答报文的帧格式，以太网类型 Type=0x0C00，报文用来响应性能参数请求报文向发送请求报文的监视部报告本站的各种性能参数数据。报文的定义遵循 ARCNET 标准报文格式。

3.3.3 实现流程

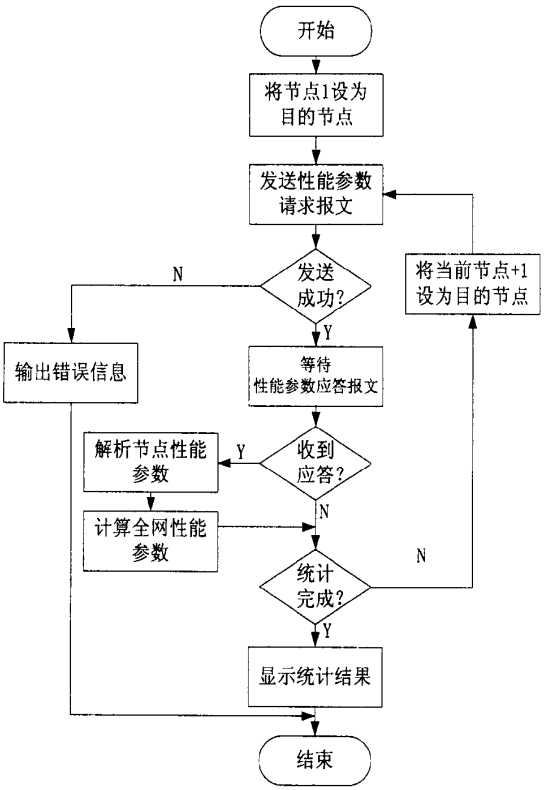

图 3-14 网络性能监控模块实现流程

图 3-14 为网络性能监控模块的实现流程，其中在性能参数数据的封装和提取时，需要按照字节序进行网络字节序与主机字节序的转换。

网络字节顺序是 TCP/IP 中规定好的一种数据表示格式，它与具体的 CPU 类型、操作系统等无关，从而可以保证数据在不同主机之间传输时能够被正确解释。网络字节顺序采用 big endian 排序方式。即大端字节序系统。主机字节序是数据在内存中保存的顺序，根据 CPU 的不同主机字节序分为两种存储方式，一种是 Little endian，即将低序字节存储在起始地址；另一种是 Big endian，即将高序字节存储在起始地址。

如果网络上全部是相同字节序的计算机那么不会出现任何问题，但由于实际可能会使用到不同字节序的计算机，所以如果不对数据进行转换，就会出现错误。因此为了避免网络传输中字节的顺序结构与 PC 内存中存储的顺序结构颠倒，在从报文中提取性能数据时需要运行 nthos()函数，将数据格式由网络字节序转换为主机字节序，相应的解析 32 位数据时需要运行 nthol()函数进行字节序转换。

3.4 节点状态监控模块

3.4.1 节点状态监控实现原理

ARCNET 的工作机制使得在物理上为总线拓扑结构的网络可以在逻辑上生成环网，如图 3-15 所示：

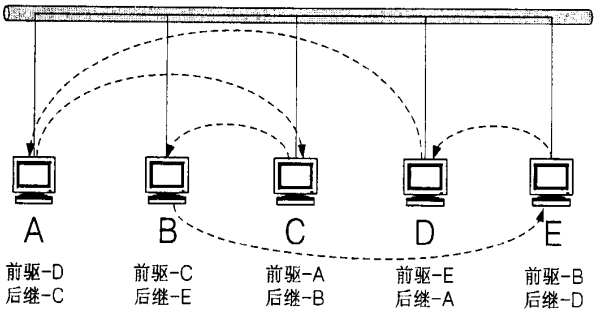

图 3-15 ARCNET 方式工作的总线型网络

环网上每一个节点都动态的维护着自己的“前驱节点”和“后继节点”，令牌将由前驱节点传递到本工作站，然后由本工作站发送给自己的后继结点。因此上图中令牌传递的实际顺序为 A-C-B-E-D-A，按照 ARCNET 的传输标准，令牌的传递按照节点逻辑地址由小到大的顺序依次传递，因此 A-C-B-E-D 也即逻辑地址大小的正向排序，图 3-16 按照这种顺序说明了环网的逻辑结构。

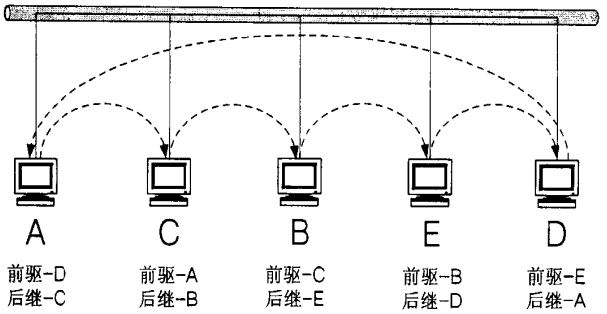

图 3-16 重新排序 ARCNET 网络节点

图 3-16 不仅重新排列了节点地址顺序，在设计节点监控和令牌监控模块时，参考上图中的逻辑结构，固定环网上节点的逻辑地址顺序，而不关心各 PC 与总线的物理结构，编写程序判断节点是否已经接入环网、令牌的传递状态是否正常。

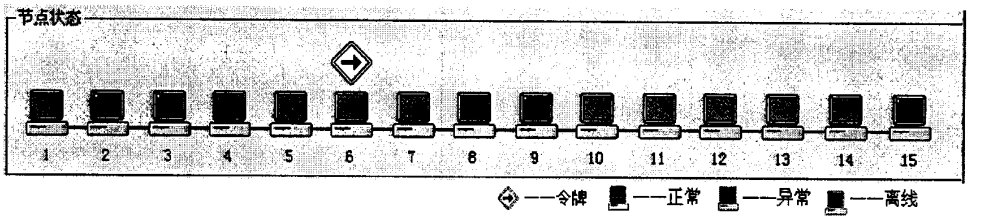

图 3-17 系统节点状态界面

如图 3-17 所示，固定网络上的节点地址为 1-15，分别对应与交换机连接的 15 台 PC。需要说明的是，按照 ARCNET 协议标准，环网上节点的逻辑地址设定范围为 1-255，因此理论上仿真平台上的节点应支持最大地址为 255 的设置，但在开发人员协同完成项目过程中为了结构清晰和说明方便，将逻辑地址的设置范围限定到 1-15，用来对应仿真平台的 15 台 PC，并且每一个逻辑地址都与特定的仿真模块绑定，其对应关系已在第 2 章中的表 2-1 中说明。

单一节点的工作状态大致分为以下三种情况：

（1） 如果监控系统周期性的收到该节点发出的令牌报文，说明节点已经接入环网、并参与到环网上的令牌循环，将该节点工作状态设为“正常”，用蓝色的 PC 图标表示；

（2） 如果监控系统能够接收到该节点发出的非令牌报文，例如可以周期性的接收到性能参数应答报文，但是没有接收到令牌报文，说明节点虽然已经接入环网，但是令牌传输出现错误（可能的错误为前驱节点的后继并没有设为本节点，而是设为本节点的后继节点，导致令牌传递不经过本站等），因此将该节点的工作状态设为“异常”，用红色的 PC 图标表示；

（3） 如果监控系统接收不到该节点的任何报文，说明该节点已经退网，因此将其工作状态设为“离线”，用黑色的 PC 图标表示。在一般情况下，“离线”不代表该节点出现故障，而是表示该工作站没有运行 ARCNET 仿真软件，或是开发人员基于调试的需要而主动关闭该工作站。

3.4.2 实现流程

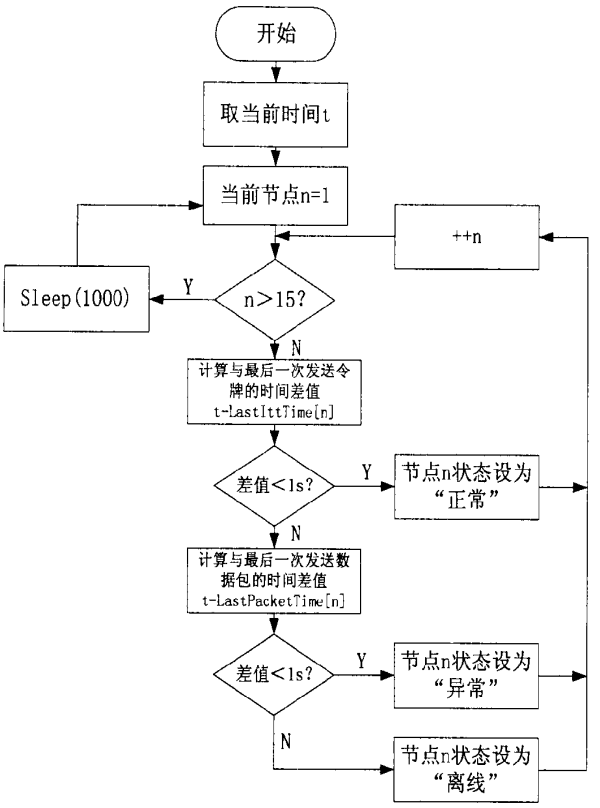

图 3-18 节点状态监控实现流程

节点监控在 MFC 定时器线程中实现，该定时器的执行频率为 1 秒，即每过 1 秒对所有节点的状态进行循环判断并刷新状态。其中 LastIttTime 为各节点最后一次发送到令牌报文的时间数组，LastPacketTime 为各节点最后一次发送数据报文的时间，LastIttTime 和 LastPacketTime 在数据包捕获线程中被记录并刷新。

3.5 令牌循环状态监控模块

令牌监控以节点监控为基础，完成的功能为监视令牌在环网上的循环。实现方式是跟踪令牌的传递路径，当节点在某时刻接收到令牌报文后，在其上方显示令牌标志，将令牌传输的状态实时的反馈给用户。 由于在仿真网络的实际运行时令牌的循环周期很短，均在 ms 级，在如此短的周期内对于令牌的每次传递都在界面做出响应，不仅会增加 CPU 的负担，而且用户无法分辨。因此需要令牌在节点上方停留 50 个周期，使用对焦点进行循环判断的方式实现，其流程图如图 3-19 所示：

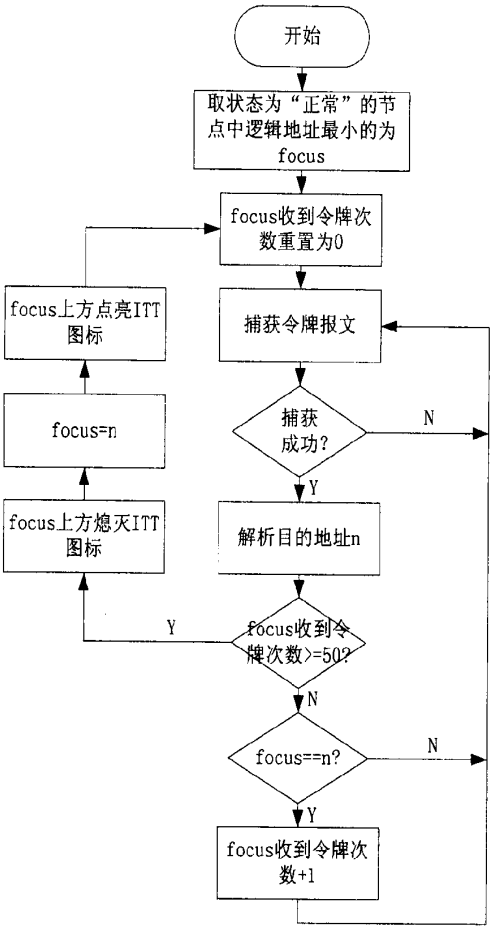

图 3-19 令牌循环状态监控实现流程

其中对当前焦点（focus）收到令牌次数的判断是实现的核心步骤：

（1） 如果当前焦点收到令牌次数≥50，则将捕获到的令牌报文的目的节点 N 作为新焦点，并显示令牌；

（2） 如果当前焦点收到的令牌次数＜50，则继续捕获循环直到其收到的令牌次数达到 50 为止；

因此每次令牌的传递都表示环网上的令牌已经循环了 50 个周期。此外，令牌监控模块还包括对令牌循环周期的计算，令牌循环周期的计算方式有两种：

（1） 节点每两次收到令牌报文的时间差：

周期 ＝t2-t1

（2） 统计每秒内各在线节点所接收的令牌报文数量 n_1，n_2，n_3...n_{15}，取平均值 n，循环周期为：

周期 ＝$1\times10^6/n$

这两种计算方式在令牌循环机制正常时均能真实的反应令牌周期，但是仿真平台实际运行中出现报文时序错乱异常时，第一种方式的统计误差很大，而第二种方式在各种异常下都可以正确的监测周期值，因此采用后者。

3.6 网络测试模块

3.6.1 网络测试模块实现原理

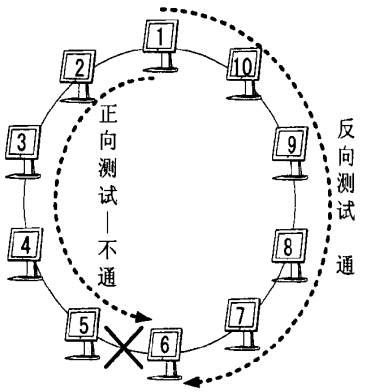

图 3-20 节点 6 上行链路故障

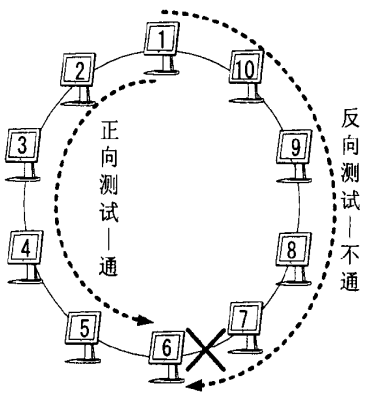

图 3-21 节点 6 下行链路故障

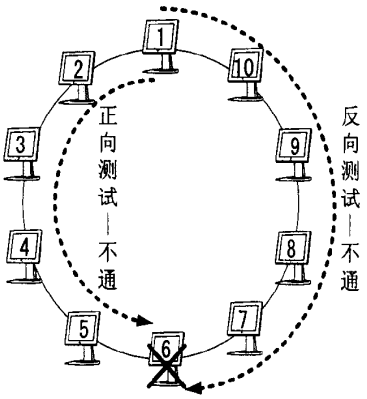

图 3-22 节点 6 故障

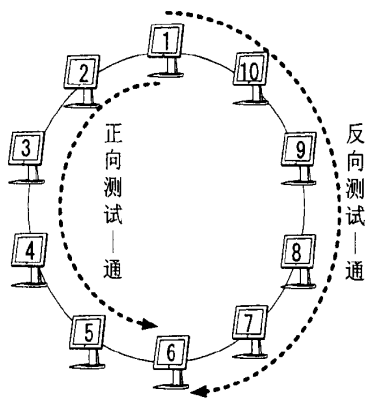

图 3-23 节点 6 正常

CRH2 列车通信网络具有环路测试功能，测试的主要目的是检测网络故障、定位故障位置。在进行测试时，监控端向被测节点发送测试信号，该信号沿环网上的两个方向分别传输，监控端通过收集两条传输路径上的反馈来判断故障原因，故障原因主要分为 3 种：连接该节点的上行链路故障、连接该节点的下行链路故障、该节点自身故障。

图 3-20、图 3-21、图 3-22 分别对上述 3 种故障原因进行说明，为了说明方便图中只画出了两条光纤环路中的主传输线。根据图中示例，1 号节点向 6 号节点发送测试信号，该测试信号通过两个方向分别传输。考虑到 ARCNET 协议中的令牌传输顺序，这里将与令牌传递顺序相同的测试路径称为“正向”，反之称为“反向”。对于不同的网络故障，1 号节点将收到不同的反馈结果。图 3-20 中的故障为 5 号-6 号节点链路故障，因此正向测试将无法收到 6 号节点的反馈，测试结果为“不通”；在反向测试的路径上

系统运行正常，因此结果为“通”。1 号节点综合两个方向的测试结果，判断故障原因为“连接 6 号节点的上行链路故障”。图 3-21、图 3-22 分别说明了 6 号-7 号节点链路故障、6 号节点自身故障的情况。监控端对故障的判定条件如表 3-3 所示：

表 3-3 故障判定条件

测试结果		故障原因
正向	反向	
通	通	无故障
通	不通	连接被测节点的下行链路故障
不通	通	连接被测节点的上行链路故障
不通	不通	被测节点自身故障

监控系统中测试模块的实现思路是，参考实际 2 型车通信网络的测试模式，完成适用于仿真平台的测试功能。虽然仿真平台模拟了 ARCNET 协议的传输，但实际 ARCNET 网络的光纤双环网与仿真平台的以太网总线型网络在结构上的不同为模拟 2 型车网络的测试功能增加了障碍：

（1）2 型车的网络测试有“正向”和“反向”两条路径，而仿真平台只有一条路径，即测试报文只能通过交换机转发后到达被测节点。

（2）2 型车通信网络的“链路故障”和“节点故障”是可以区分的，而仿真平台的“链路故障”和“节点故障”是绑定在一起的。由于仿真平台各 PC 节点通过双绞屏蔽线与交换机连接，因此链路路故障（包括双绞线与节点的连接、交换机端口与内部交换阵列的连接）与 PC 故障是可以看作相互触发、并且在监控端是无法区分的。

针对以上两点不同，本测试模块的实现以测试仿真平台的节点故障为主，如果节点的测试结果正常，那么相应的链路也应是无故障的。测试模块的详细实现参考 2 型车通信网络的测试模式，安装监控系统的 PC 向仿真平台上的各节点发送测试报文，如果某节点是正常工作的则应立即返回测试应答，监控端以是否收到该节点的应答报文、经过多久收到应答报文为参考基准对该节点作出评价。

3.3.2 网络测试报文定义

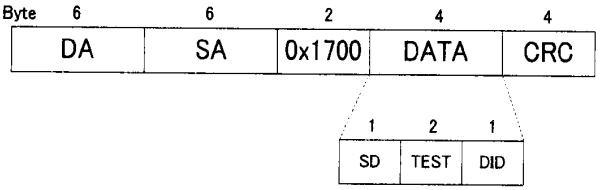

图 3-24 测试报文结构定义

图 3-24 为测试报文结构，DA 字段为待测节点的 MAC 地址，由于监控系统后台始终

维护各节点逻辑地址与其 MAC 地址对应关系的数组，因此可以直接获取。SA 为监控系统网卡的 MAC 地址，该地址在用户选择网卡时已经自动获取并保存。测试报文类型字段值设为 0x1700，DID 为待测试节点的逻辑地址。

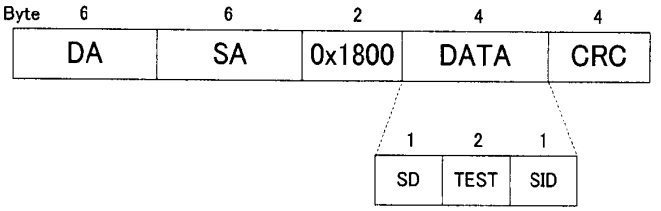

图 3-25 测试应答报文结构定义

图 3-25 为测试应答报文结构，DA 为监控系统网卡的 MAC 地址，该值从节点收到的网络测试帧的 SA 字段获取。SA 为该节点的 MAC 地址。测试应答报文类型字段值设为 Type=0x1800。

3.4.2 实现流程

网络测试的实现流程如图 3-26 所示：

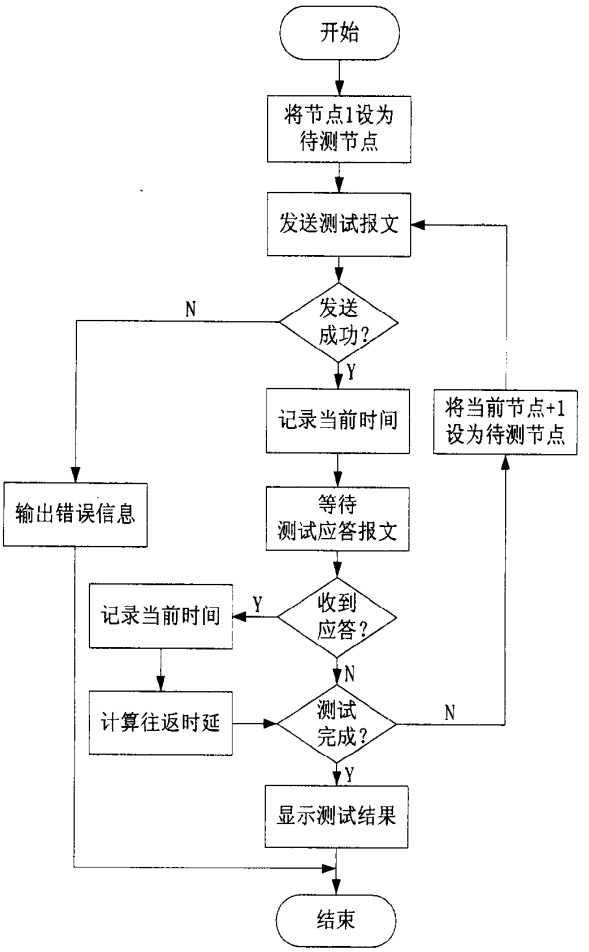

图 3-26 网络测试流程

测试流程按照节点逻辑地址为 1-15 的顺序依次进行测试。首先按照已定义的测试报文格式进行封装，然后调用 Winpcap 函数 pcap_sendpacket()发送测试报文，发送函数的执行成功与否和待测节点的状态无关，即使目的节点不存在测试报文也应能够成功发送。因此如果函数执行错误应立即停止测试循环并输出错误信息，可能的错误原因之一是所选的网卡不可用，例如选择了安装的虚拟机网卡等。

此外，监控系统对于测试应答报文的接收等待是有时间限制的，此处设置为 10ms 内没有接收到待测节点返回的应答则判为超时，否则计算时延。在向所有节点都进行网络测试之后，测试结束并显示测试结果。

3.7 列车控制逻辑模块

3.7.1 列车控制逻辑模块实现原理

本模块以软件程序完成列车控制系统的工作逻辑，实现仿真平台运行时通过中央装置对全车进行遥控操作。列车控制逻辑包括对全车的照明、广播、指示灯的开/关控制，以及处理电源切换、受电弓升降、VCB 断合等指令[46-48]，仿真平台中定义了 20 种操作类型，以下分别说明其中的空档开关和恒速等的控制逻辑：

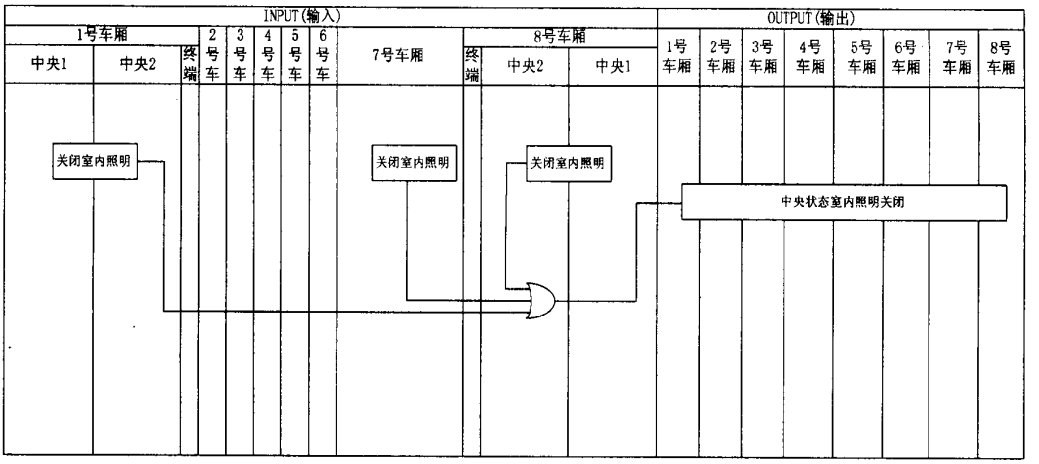

图 3-27 列车车室内照明开/关控制逻辑

图 3-27 为全车室内照明的开/关的控制逻辑，能够发出控制信息的节点有 5 个，分别是两端的中央装置和 7 号车厢，各节点指令通过或门判断后输出结果，因此任何一个节点的操作都能改变全车的照明状态。

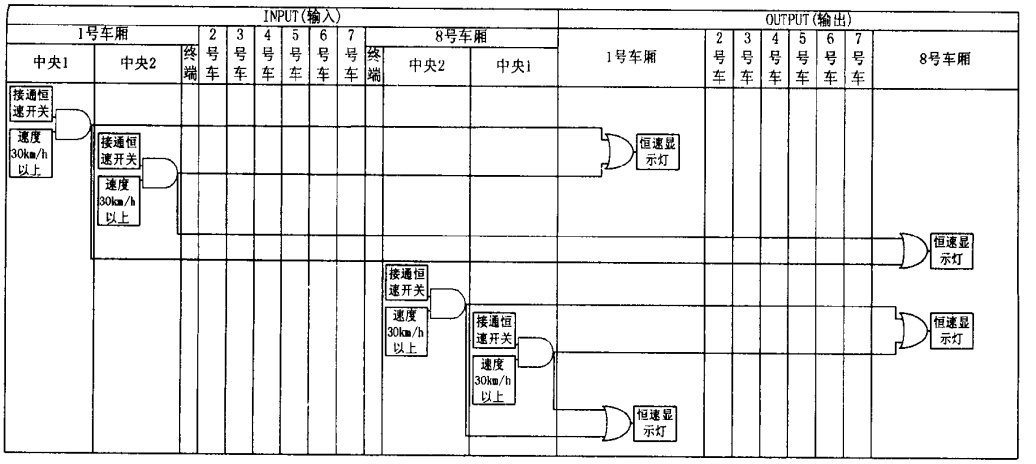

图 3-28 列车恒速灯开/关控制逻辑

图 3-28 为恒速灯的开/关控制逻辑，列车的中央装置接通恒速开关并且列车当前运行速度为 30km/h 以上时，1 号车厢和 8 号车厢的恒速显示灯亮。

控制逻辑在仿真平台上的实现方式为，首先定义 DI 报文，各节点将相应的指令操作按照 DI 报文的格式保存至相应字段中，并以广播方式发送至网络，监控系统识别并接收该报文后对指令判断并输出结果，结果按照 DO 报文的格式进行封装并发送。

对于 DI、DO 报文的各字段统一采用 0x01 为打开，0x00 为关闭，监控系统以条件判断语句来模拟数字电路中的与门、非门、或门，当捕获 DI 报文后依次取各字段的值，将相关的字段值例如“恒速”和“速度>30km/h”进行与或判断并输出结果，结果作为数据内容封装到 DO 报文中的相应字段，在对 DI 报文的所有字段都进行判断之后，将 DO 报文发送到相应节点。

3.7.2 列车控制逻辑模块报文定义

（1）DI 报文

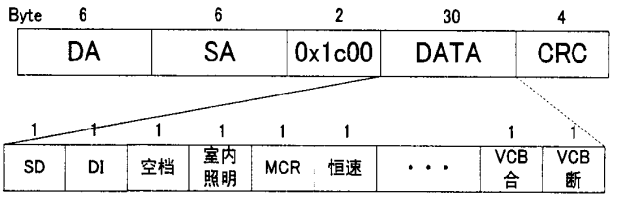

图 3-29 DI 报文结构定义

图 3-29 为 DI 报文结构，以太帧类型字段设为 0x1c00，DATA 字段包含了 28 种指令类型，包括空档、室内照明、MCR、恒速等指令信息，各字段以 0x01 表示打开操作，0x00 为关闭操作。

（2）DO 报文

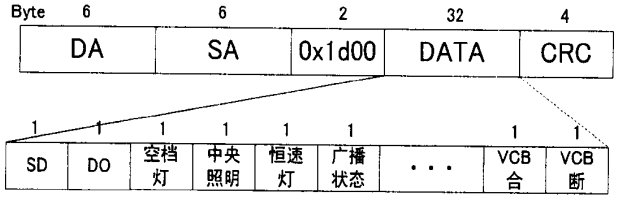

图 3-30 DO 报文结构定义

图 3-30 为 DO 报文结构，以太帧类型字段设为 0x1d00，DATA 字段包含了 30 种操作结果，包括空档灯、室内照明、MCR、恒速灯等。各字段 0x01 表示打开，0x00 为关闭，其余值将不进行操作。

3.8 异常告警模块

3.8.1 多令牌异常研究

仿真平台在实际运行中有时会出现报文失序的现象，从某一时刻开始报文的到达顺序不符合 ARCNET 协议规范，甚至出现多个令牌同时到达同一节点的情况。如图 3-31 所示：

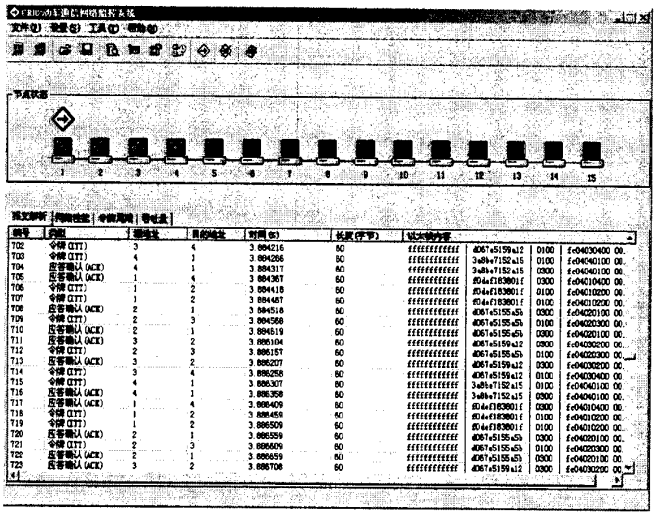

图 3-31 报文时序错乱

（1）异常现象及原因推测

在报文时序和到达时间（微秒级）上，由于监控系统与 Wireshark 的监控结果完全一致，因此排除监控系统本身错误。报文失序情况一般出现在在报文传输路径较为复杂的网络环境中，后来发送的报文由于路径和时延的不同比先前发送的报文更早到达目的地址。但是在本仿真平台的实验环境较为简单，传输信道较为理想，因此由于网络原因而造成失序的可能性很小，在局域网中验证方式为将 10 万条报文依次由小到大编号，将编号封装到报文内部并发送，可以发现目的节点是按照发送顺序接收的，并

未出现时序错乱。因此，本文将异常原因判断为令牌过多，从而导致环网上同时存在多重令牌循环。

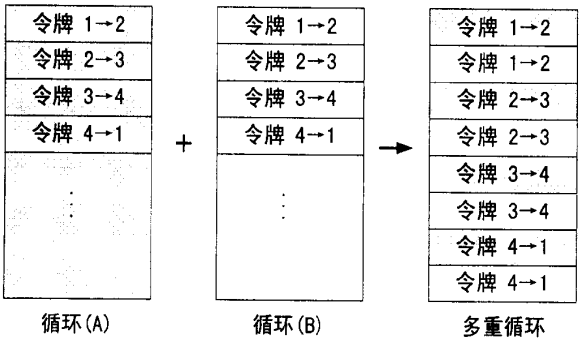

图 3-32 多重令牌循环

图 3-32 为报文时序错乱的异常原因解析，即当前环网上同时存在多重令牌循环。在正常循环的基础上，当加入多余的令牌时，会引起仿真平台建立一套新的令牌循环并与原有循环重叠，并导致报文失序的现象。

（2）理论分析

如果报文失序不是由多余的令牌引起，那么按照监控系统的令牌周期计算方式，所检测的周期应该为正常值；而如果令牌过多为异常原因，单位时间内各节点收到的令牌次数会显著增加，那么循环周期也会明显减小。因此，可以统计在正常情况下环网的令牌周期，在此基础上监控系统向网络中发送一条令牌报文，生成额外的循环以模拟异常发生，分别记录不同节点数量和不同令牌数量情况下，仿真平台的令牌循环周期，统计值将作为参考标准，仿真平台出现报文乱序时通过与相应统计值进行比较即可以确定是否由多余令牌引起。

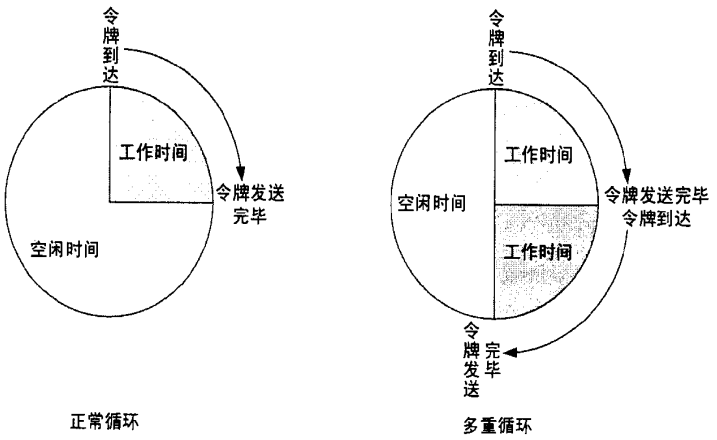

图 3-33 节点工作状态分布

此外，网络吞吐量也可以作为判断令牌过多异常的依据。当环网上令牌过多时，网络吞吐量也会相应提升。这里将节点在一个令牌循环周期内的时间分为“工作时间”和“空闲时间”，工作时间表示从令牌到达该节点开始，到该节点将令牌向下一站（NID）发送完毕为止的时间。当令牌在其他节点间传递时，该节点进入空闲状态。工作时间

的大小基本为固定值，主要与令牌报文和 ACK 报文的处理时间有关，而空闲时间的大小随着当前环网上的节点数量而变动，当前节点越多，每个节点的空闲时间越长。

在实际运行中，节点在空闲时间内仍有可能发送并接收一些报文，例如发送性能参数应答报文，或者接收数据（PAC）报文，但是由于该类报文的收发间隔一般大于 1 秒，平均至一个令牌周期后其所占时间的比重很小，因此忽略不计。对令牌报文和 ACK 报文的处理占用了节点的大部分工作时间。

如果仿真平台上只存在一个令牌循环，那么节点将令牌传递出去后将进入空闲状态，但是如果平台上有多个令牌参与循环，那么当多余的令牌到达时，会迫使节点增加工作时间来对其进行处理，处理过程中产生的大量 ACK 报文和 ITT 报文会带来网络吞吐量的显著提升。因此，可以对正常循环与多重循环情况下仿真平台的吞吐量进行比较来判断令牌是否过多。从理论上来说，如果节点的工作时间没有达到极限，那么吞吐量会随着令牌个数的增加而线性增长，因此利用吞吐量来判断令牌个数在节点空闲时间有剩余的情况下是可行的。

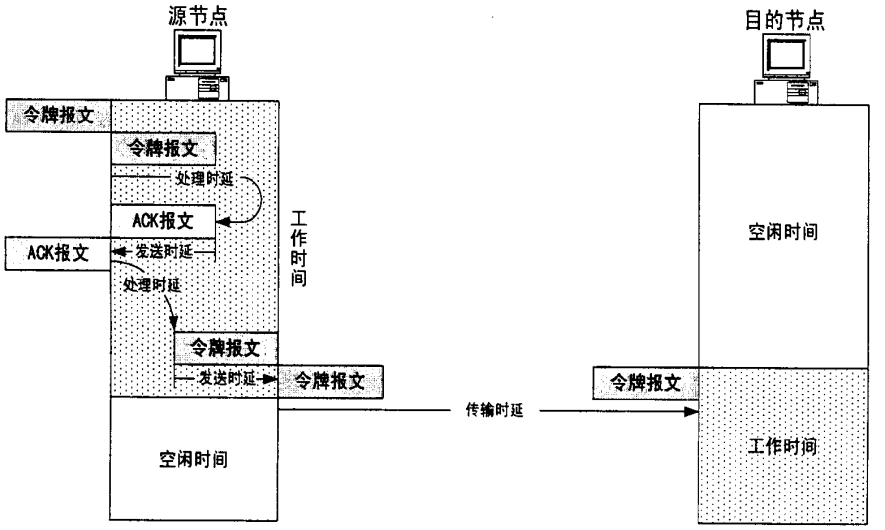

图 3-34 节点工作时间构成

图 3-34 说明了节点工作时间的构成，即在无数据报文（PAC）发送时，从令牌报文到达某节点开始，到令牌到达其下一节点（NID）为止。更准确的说，该时间为：

$$\tau_{\text{节点工作时间}} = \tau_{\text{令牌接收处理时延}} + \tau_{\text{ACK 发送时延}} + \tau_{\text{令牌发送处理时延}} + \tau_{\text{令牌发送时延}}$$

在理想情况可以只计算发送时延，由于令牌报文和 ACK 报文均为 60 字节，在网络带宽为 10Mb/s 时，节点工作时间的最小值为（60+60）/5×100000 = 0.012 ms，实际运行时统计的最小值为 0.07ms。该值表示节点处理效率达到极限时的工作时间，随着令牌个数的增多，节点的工作时间将逐渐逼近该值。

如果令牌达到节点所能应付的极限，吞吐量将会达到峰值，此时令牌再多也不会产生更多的吞吐量，而节点的缓冲区将会积压大量报文，当堆积的数据量超过 Winpcap

设置的缓冲区大小（8Mb）时，会造成严重丢包[49,50]，即使监控程序发送了销毁令牌的信息也未必能够到达节点。

（3）实际验证

启动 4 个节点的仿真程序构成环网，此时令牌循环正常，令牌循环周期均值为1.584ms，吞吐量均值为 4.058kpacket/s。此时点击“生成令牌”来模拟多令牌异常，点击一次之后当前环网上有 2 个令牌参与循环，可以监测到令牌周期下降为 0.803ms，吞吐量上升为 7.948 kpacket/s。当再次点击“生成令牌”之后环网上有 3 个令牌参与循环，此时周期降为 0.563 ms，吞吐量上升为 11.836 kpacket/s。如图 3-35 所示：

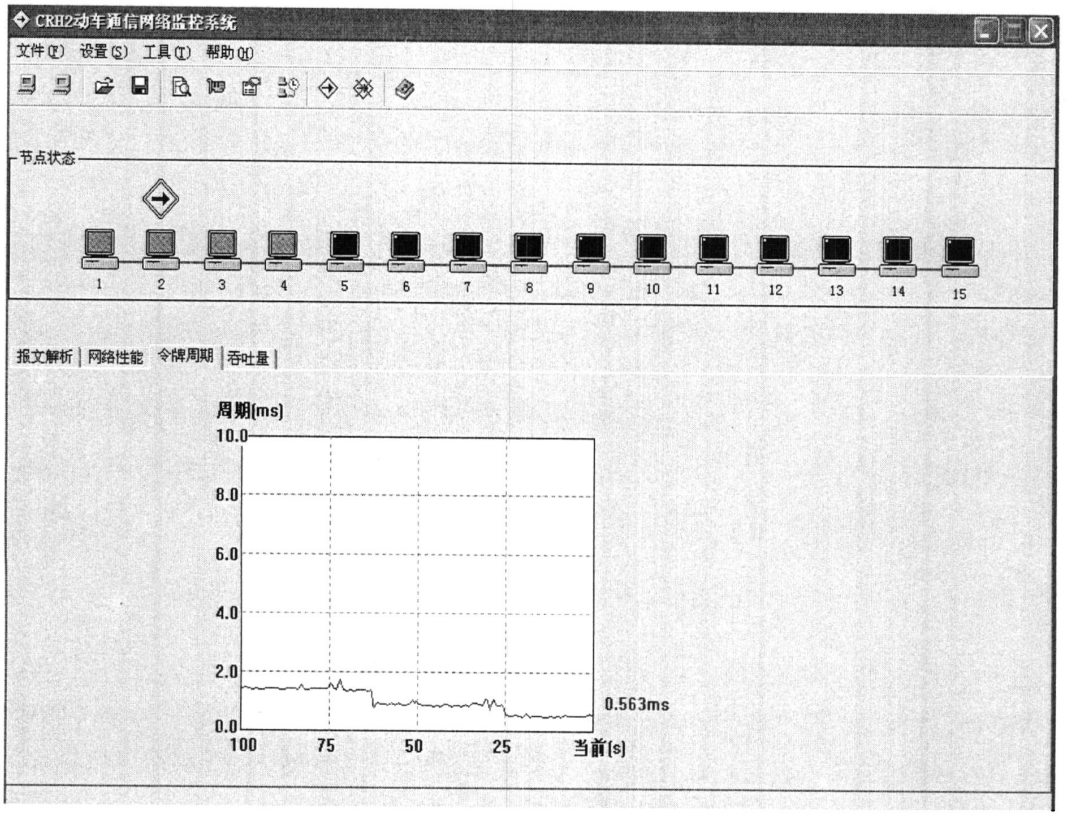

图 3-35 令牌由 1 增加到 3 时令牌周期的变化

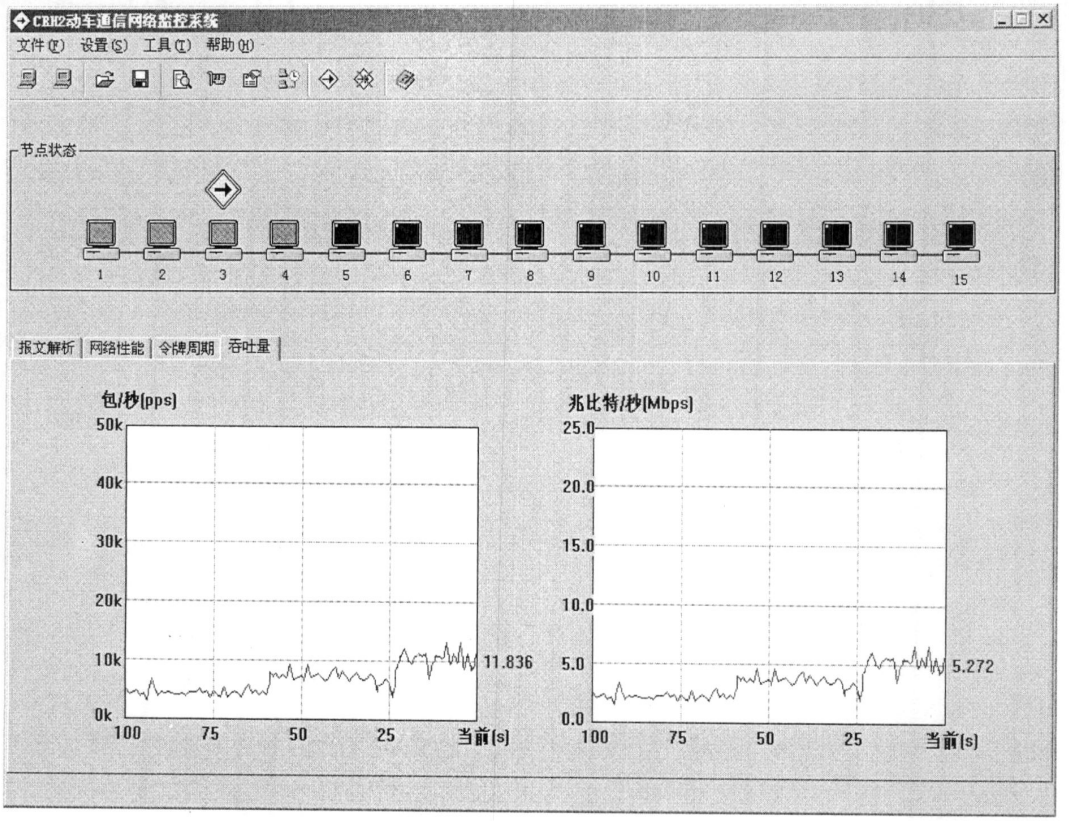

图 3-36 令牌由 1 增加到 3 时吞吐量的变化

因此可以验证上述分析，即当节点的空闲时间有剩余时，多余的令牌会使令牌周期明显降低，吞吐量随令牌个数的增加线性升高。

因此在平台运行时可以根据该值对当前令牌个数进行判断并告警，需要说明的是，令牌周期和吞吐量的监测数值与 PC 的硬件配置、交换机性能和 PAC 数据报文的发送频率相关，PC 的配置越高，对各类报文的处理时延越小，相应的令牌周期越短。

3.8.2 报文定义

（1） 令牌报文

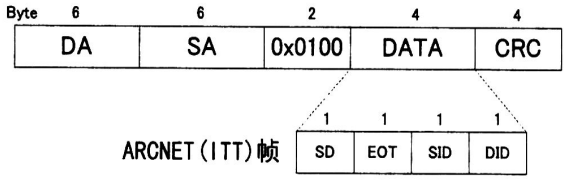

图 3-37 仿真平台令牌(ITT)结构定义

在重新生成令牌报文时，按照仿真平台的令牌报文格式进行封装，如图 3-37 所示。DA 为目的 MAC 地址，如果工作状态为“异常”的节点数≥2，那么在所有“异常”节点中选择逻辑地址最小的作为目的地址。SA 为源 MAC 地址，如果工作状态为“异常”的节点数≥2，那么在所有“异常”节点中选择逻辑地址最大的作为目的地址。其它字段按照相应标准依次赋值。

按照上述方式封装令牌报文可以使令牌到达环网中第一个可用的节点，令牌依次向下传递并恢复循环。

（2）销毁令牌报文

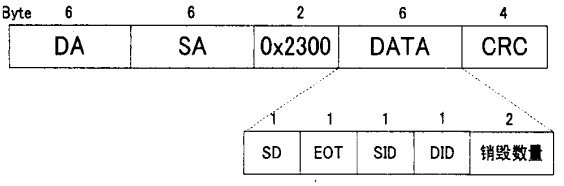

图 3-38 销毁令牌报文结构定义

销毁令牌报文的定义方式如图 3-38 所示，以太帧类型字段设为 0x2300，DATA 字段中包含销毁数量，该值默认值为 1。接收端接收到销毁报文后，将根据销毁数量 n 对后续传递而来的 n 个令牌报文不作处理。

3.8.3 实现流程

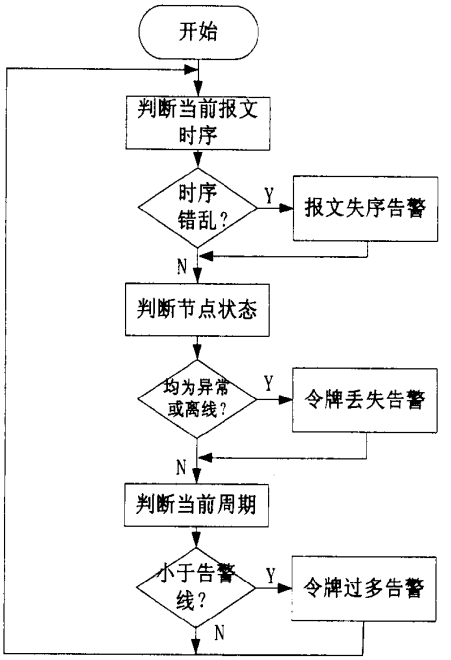

图 3-39 异常告警实现流程

图 3-39 为异常告警模块实现流程。对于报文失序的判断，监控系统记录当前到达的报文类型，并与上一个报文类型进行比较，如果两者都是 ARCNET 报文但是不符合工作顺序则进行告警。

异常告警模块结合报文监控、令牌监控及网络性能监控的监控结果，根据异常特征进行综合判断，并作出相应告警。告警分为 3 类：令牌丢失、令牌过多、报文时序错乱。

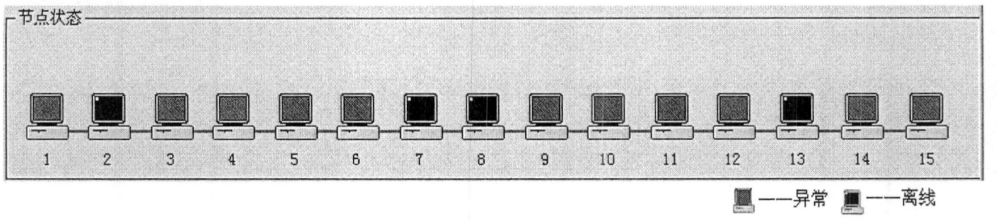

图 3-40 令牌丢失时节点状态表现

如果环网上所有节点的工作状态均为“异常”或者“离线”，并且所有节点在一段时间内均未出现令牌图标，可以判断为令牌传输丢失，需要监控系统重新生成并发送令牌报文，尝试引导令牌循环回归正常。

3.9 本章小结

本章对各个模块进行了详细设计，在每个模块的实现中，分别介绍了实现原理、报文定义和实现流程。其中，在异常告警模块中对多令牌异常和随之引起的报文乱序现象进行了理论分析，从报文时序、令牌周期和吞吐量来定位异常，并通过发送令牌销毁报文来解除故障。

第 4 章 系统的实现与测试

4.1 系统实现环境

硬件环境：

(1) Cisco SL224 10/100M 交换机；

(2) 计算机 CPU P4 2.0GHz，内存 2G 及以上，标准以太网卡；

软件开发环境：

(1) 编译器要求：Visual C++ 6.0;

(2) 操作系统： Windows XP Professional SP3

(3) 通信中间件: Winpcap-4-0-1.exe, wpdpack.zip;

(4) 通用运行库：<ANSI stdio, stdlib, winpcap>

4.2 系统部署

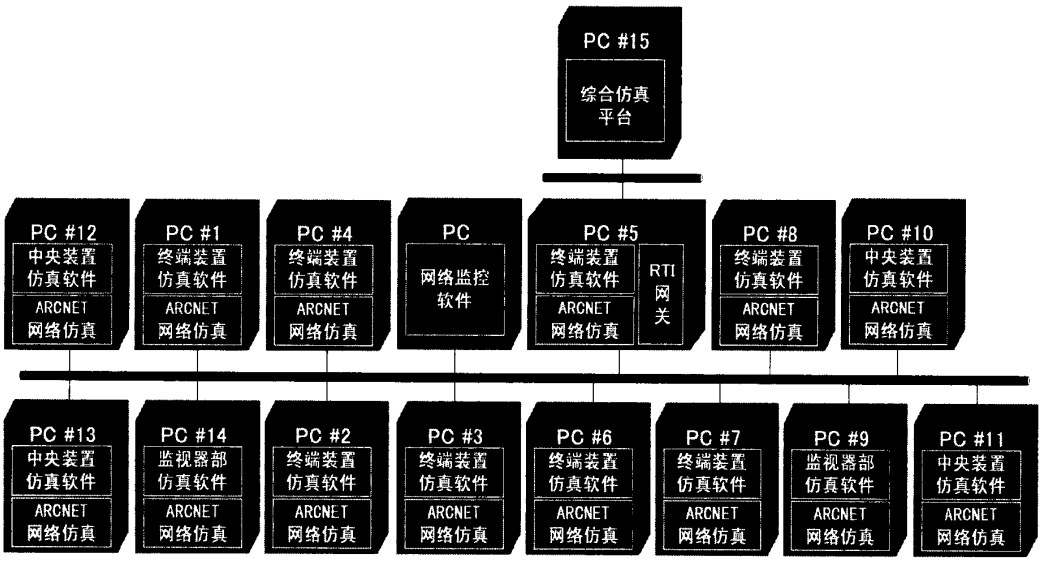

图 4-1 仿真平台部署图

CRH2 列车通信网络系统仿真平台的部署结构如图 4-1 所示，仿真平台共需要 16 台计算机，其中 PC#1 至 PC#14 节点均安装 ARCNET 网络传输仿真软件，作为中央装置站点仿真软件、监视器部站点仿真软件以及终端装置站点仿真软件的后台程序，PC#15 安装综合仿真平台，通过以太网提供的接口与 PC#5 节点上的 RTI 网关软件通信。在此基础上，安装有监控系统的 PC 连接到网络进行监控测试。

4.3 系统测试

4.3.1 测试方案设计

表4-1 测试方案

测试对象	测试目的	测试步骤	期望结果和判定依据
报文捕获与分析模块	测试能否正常捕获数据包，报文的分析显示是否正确	运行仿真平台，各节点构建环网，用监控系统和Wireshark捕获数据	监控系统捕获的数据包与Wireshark在报文内容、长度、时间上完全一致，没有漏包现象
报文过滤模块	测试是否可以从捕获数据中过滤出用户指定类型的报文	过滤界面进行用户配置并应用	数据包显示界面只显示过滤后的报文
节点状态监控模块	测试是否正常显示各节点工作状态	依次将各节点加入环网，检查对应节点的状态显示是否正确	各节点的状态显示与其实际情况相符
令牌循环监控模块	测试令牌是否正常显示，令牌周期的监控结果是否正确	节点加入环网过程中，检查令牌显示及周期变化	随着环网节点的增多，令牌周期相应增大
网络性能监控模块	测试是否正确从性能参数报文中提取数据并显示	根据在线节点数量，检查是否动态更新各节点性能参数	显示结果与实际情况相符
网络测试模块	测试能否正常反应各节点的通断状态和时延	运行网络测试	未连接网络的PC显示超时，连接到交换机的PC正常显示时延
异常告警模块	测试是否对报文错乱、令牌丢失、令牌过多进行告警	监控系统生成令牌，模拟产生令牌过多异常并检查告警反应	异常发生时，告警及时显示

4.3.2 测试步骤及结果分析

（1）比较监控系统的报文时序、捕获时间、报文长度以及报文内容，均与 Wireshark 一致，在捕获期间内监控系统所捕获的数据包数量也与 Wireshark 相同，没有漏抓的现象，因此报文捕获与解析模块可以正常工作。

报文解析 | 网络性能 | 令牌周期 | 吞吐量

编号	类型	源地址	目的地址	时间(S)	长度(字节)	以太帧内容			
107	应答确认(ACK)	10	9	3.881187	60	d067e513db db	5404a691f347	0300	fc060a0900 00..
108	令牌(ITT)	10	5	3.881228	60	d067e5139b12	5404a691f347	0100	fc040a0500 00..
109	应答确认(ACK)	5	10	3.881308	60	5404a691f347	d067e5139b12	0300	fc06050a00 00..
110	令牌(ITT)	5	9	3.881392	60	d067e513db db	d067e5139b12	0100	fc04050900 00..
111	应答确认(ACK)	9	5	3.881470	60	d067e5139b12	d067e513db db	0300	fc06090500 00..
112	令牌(ITT)	9	10	3.881542	60	5404a691f347	d067e513db db	0100	fc04090a00 00..
113	应答确认(ACK)	10	9	3.881602	60	d067e513db db	5404a691f347	0300	fc060a0900 00..
114	令牌(ITT)	10	5	3.881633	60	d067e5139b12	5404a691f347	0100	fc040a0500 00..
115	应答确认(ACK)	5	10	3.881724	60	5404a691f347	d067e5139b12	0300	fc06050a00 00..
116	令牌(ITT)	5	9	3.881795	60	d067e513db db	d067e5139b12	0100	fc04050900 00..
117	应答确认(ACK)	9	5	3.881896	60	d067e5139b12	d067e513db db	0300	fc06090500 00..
118	令牌(ITT)	9	10	3.881978	60	5404a691f347	d067e513db db	0100	fc04090a00 00..
119	应答确认(ACK)	10	9	3.882018	60	d067e513db db	5404a691f347	0300	fc060a0900 00..
120	令牌(ITT)	10	5	3.882059	60	d067e5139b12	5404a691f347	0100	fc040a0500 00..
121	应答确认(ACK)	5	10	3.882150	60	5404a691f347	d067e5139b12	0300	fc06050a00 00..
122	令牌(ITT)	5	9	3.882226	60	d067e513db db	d067e5139b12	0100	fc04050900 00..
123	应答确认(ACK)	9	5	3.882313	60	d067e5139b12	d067e513db db	0300	fc06090500 00..
124	令牌(ITT)	9	10	3.882383	60	5404a691f347	d067e513db db	0100	fc04090a00 00..
125	应答确认(ACK)	10	9	3.882435	60	d067e513db db	5404a691f347	0300	fc060a0900 00..
126	令牌(ITT)	10	5	3.882476	60	d067e5139b12	5404a691f347	0100	fc040a0500 00..
127	应答确认(ACK)	5	10	3.882567	60	5404a691f347	d067e5139b12	0300	fc06050a00 00..
128	令牌(ITT)	5	9	3.882643	60	d067e513db db	d067e5139b12	0100	fc04050900 00..

图 4-2 监控系统报文数据显示

CapData.CAP [Wireshark 1.6.8 (SVN Rev 42761 from /trunk-1.6)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Length	Info
107	3.881187	AsustekC_91:f3:47	Dell_13:db:db	RPL	60	Unknown Type[Malformed Pa
108	3.881228	AsustekC_91:f3:47	Dell_13:9b:12	RPL	60	Unknown Type[Malformed Pa
109	3.881308	Dell_13:9b:12	AsustekC_91:f3:47	LLC	60	S, func=RNR, N(R)=5; DSAP
110	3.881392	Dell_13:9b:12	Dell_13:db:db	LLC	60	S P, func=RNR, N(R)=4; DS
111	3.881470	Dell_13:db:db	Dell_13:9b:12	LLC	60	S P, func=REJ, N(R)=2; DS
112	3.881542	Dell_13:db:db	AsustekC_91:f3:47	LLC	60	S, func=REJ, N(R)=5; DSAP
113	3.881602	AsustekC_91:f3:47	Dell_13:db:db	RPL	60	Unknown Type[Malformed Pa
114	3.881633	AsustekC_91:f3:47	Dell_13:9b:12	RPL	60	Unknown Type[Malformed Pa
115	3.881724	Dell_13:9b:12	AsustekC_91:f3:47	LLC	60	S, func=RNR, N(R)=5; DSAP
116	3.881795	Dell_13:9b:12	Dell_13:db:db	LLC	60	S P, func=RNR, N(R)=4; DS
117	3.881896	Dell_13:db:db	Dell_13:9b:12	LLC	60	S P, func=REJ, N(R)=2; DS
118	3.881978	Dell_13:db:db	AsustekC_91:f3:47	LLC	60	S, func=REJ, N(R)=5; DSAP
119	3.882018	AsustekC_91:f3:47	Dell_13:db:db	RPL	60	Unknown Type[Malformed Pa
120	3.882059	AsustekC_91:f3:47	Dell_13:9b:12	RPL	60	Unknown Type[Malformed Pa
121	3.882150	Dell_13:9b:12	AsustekC_91:f3:47	LLC	60	S, func=RNR, N(R)=5; DSAP
122	3.882226	Dell_13:9b:12	Dell_13:db:db	LLC	60	S P, func=RNR, N(R)=4; DS
123	3.882313	Dell_13:db:db	Dell_13:9b:12	LLC	60	S P, func=REJ, N(R)=2; DS
124	3.882383	Dell_13:db:db	AsustekC_91:f3:47	LLC	60	S, func=REJ, N(R)=5; DSAP
125	3.882435	AsustekC_91:f3:47	Dell_13:db:db	RPL	60	Unknown Type[Malformed Pa
126	3.882476	AsustekC_91:f3:47	Dell_13:9b:12	RPL	60	Unknown Type[Malformed Pa
127	3.882567	Dell_13:9b:12	AsustekC_91:f3:47	LLC	60	S, func=RNR, N(R)=5; DSAP

```
0000  d0 67 e5 13 db db 54 04  a6 91 f3 47 03 00 fc 06   .g....T. ...G....
0010  0a 09 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 00 00 00  00 00 00 00              ........ ....
```

File: "C:\Documents and Settings\Adminis... Packets: 156891 Displayed: 156891 Mark... Profile: Default

图 4-3 WireShark 报文数据显示

（2）依次将 1、2、3、4、7、8、9 号节点加入环网，并发送 PAC 数据报文到司空台。

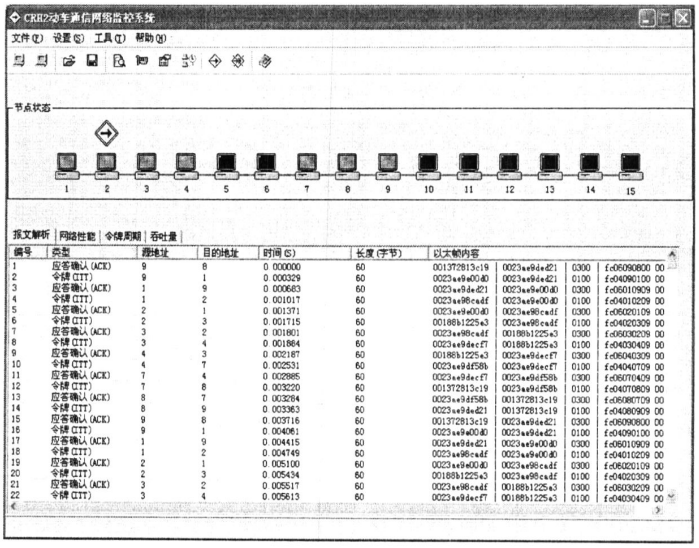

图 4-4 当前网络报文显示

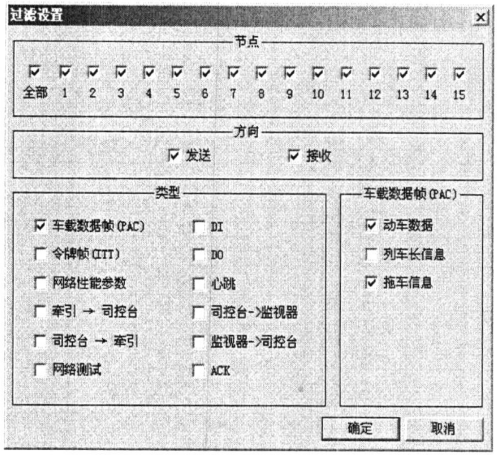

图 4-5 设置过滤规则

如图 4-5 所示，打开过滤配置界面，选择的选项为“所有节点”“收发的”“动车数据和拖车信息”，并点击确定。

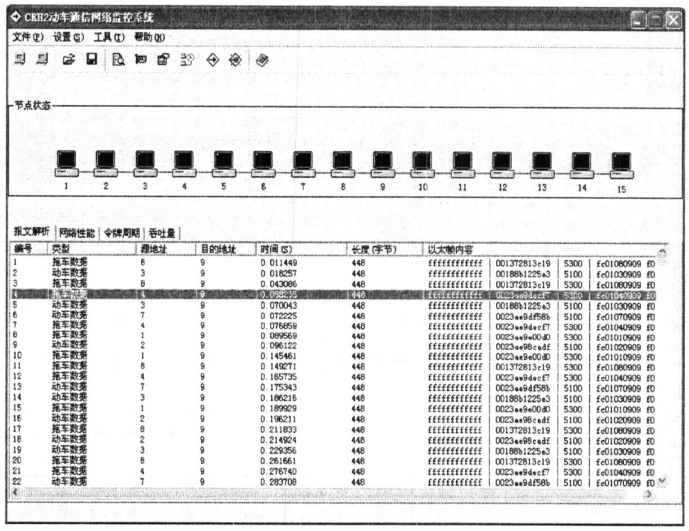

图 4-6 过滤后的报文显示

图 4-6 为过滤后的报文信息，从报文列表中可以看到，监控系统已将所需的报文成功过滤出来。此外，经过了其他选项的过滤选项测试，此处不一一列举，表明报文过滤模块能够正常工作。

(3)在 1、2、3、4 号节点加入环网的过程中，对令牌循环周期进行监测。

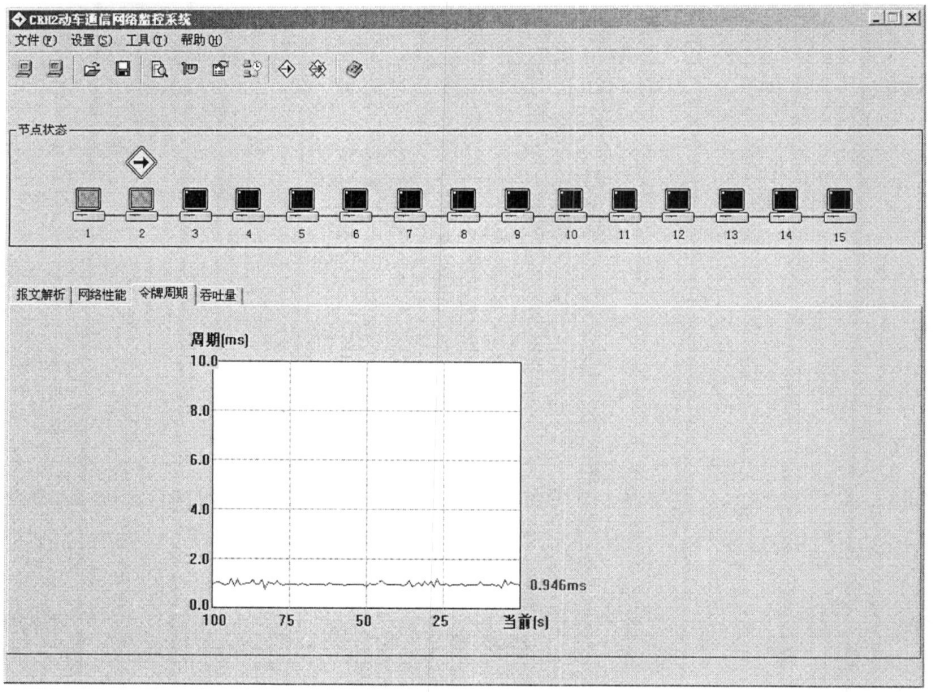

图 4-7 2 个节点时令牌循环周期

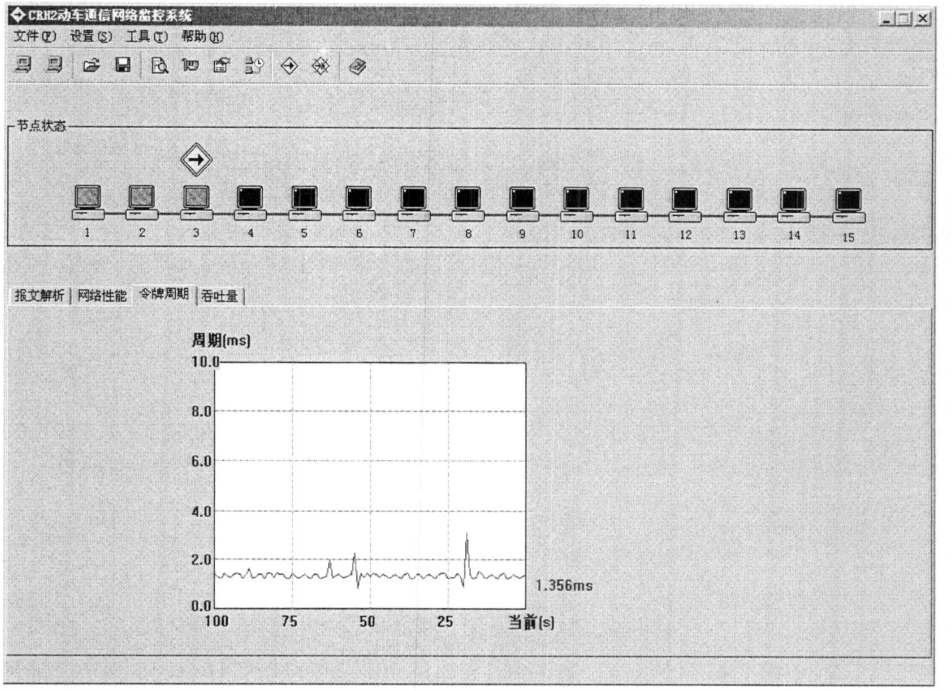

图 4-8 3 个节点时的令牌循环周期

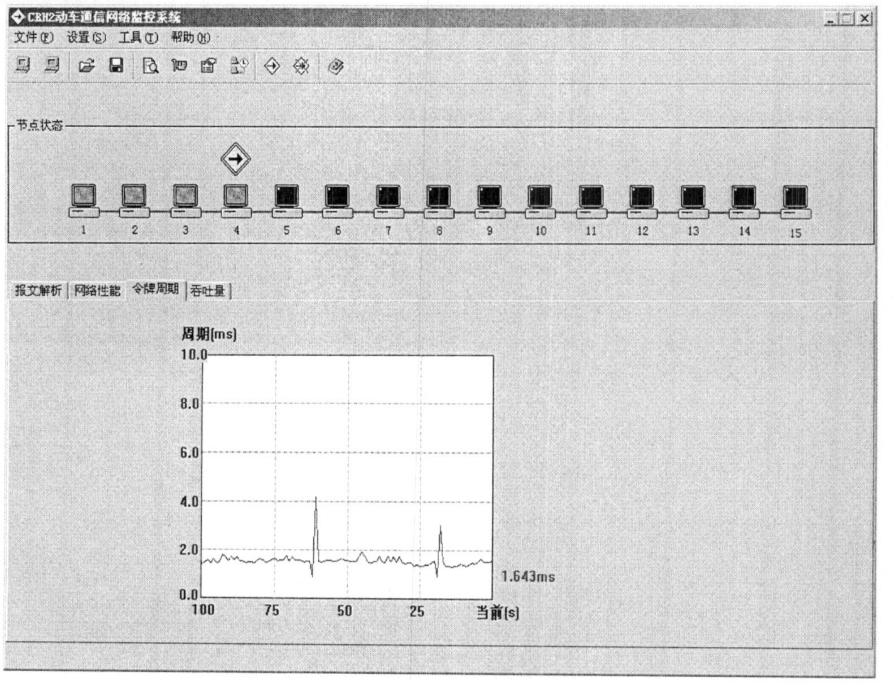

图 4-9 4 个节点时的令牌循环周期

由以上统计可以发现，随着加入环网节点的增多，令牌循环周期不断增大，在环网上的节点为 2、3、4 时，令牌周期均值分别约为 0.946ms，1.356ms，1.643ms，并且各节点工作状态的显示也与入网节点的实际情况相符，因此节点状态监控和令牌循环监控可以正常工作。

（4）点击“生成令牌”来模拟多令牌异常，此时报文时序已经出现错误，如下所示：

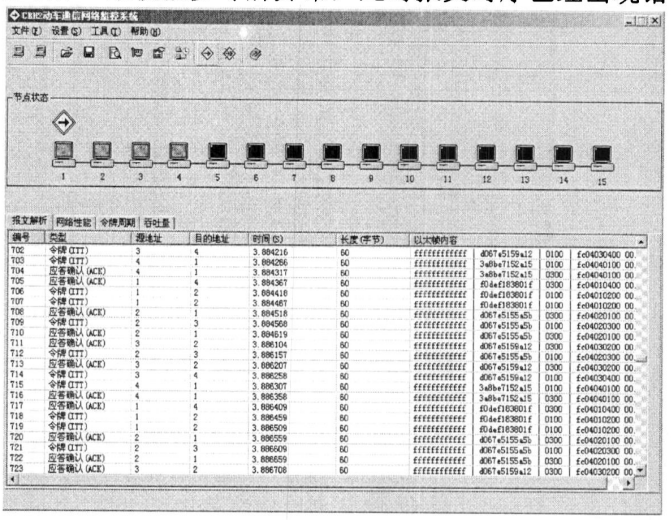

图 4-10 多令牌异常模拟

系统自动发出报文乱序告警和令牌过多告警。

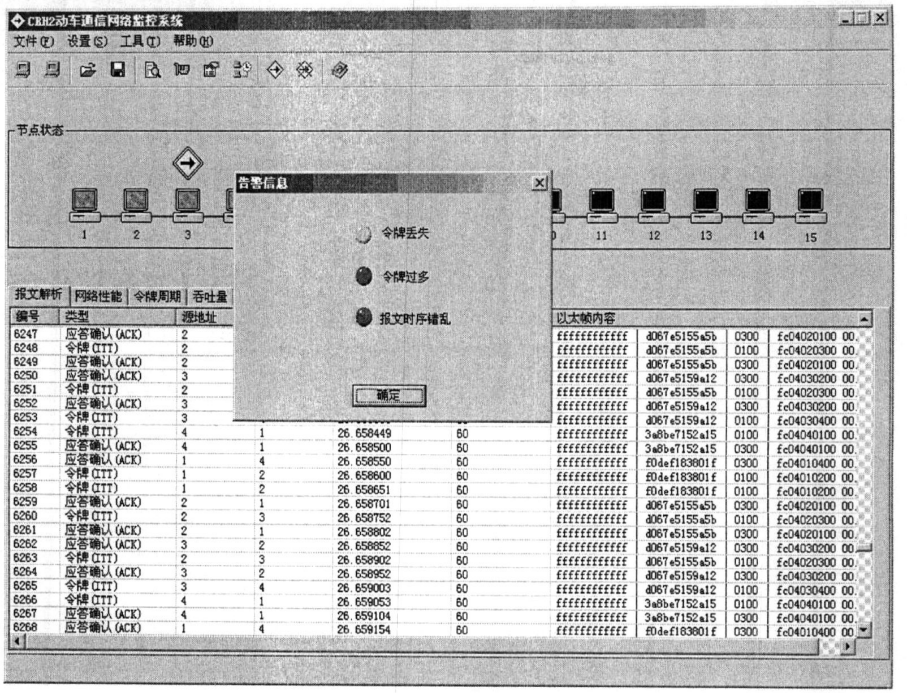

图 4-11 异常告警

（5）在环网上节点增加到 7 时，监测网络性能参数的变化，如下所示：

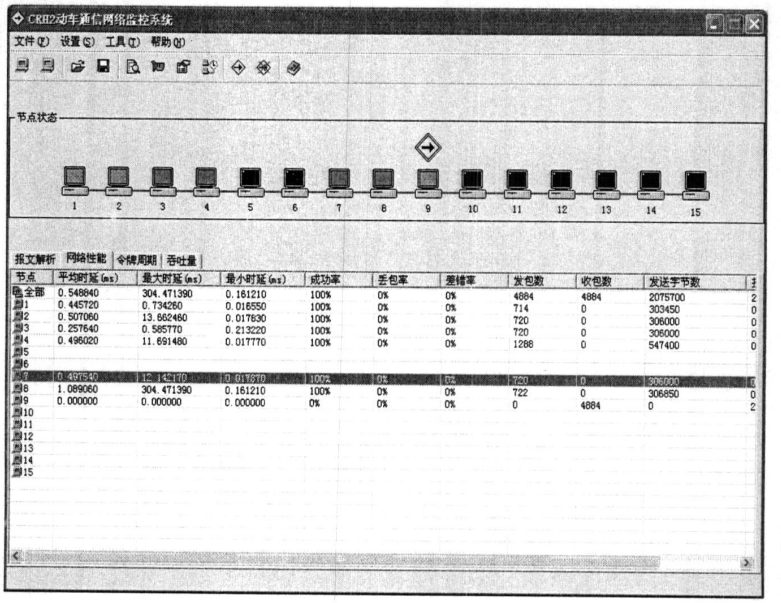

图 4-12 网络性能监控

图 4-12 为网络性能监控的运行状态，在仿真平台运行过程中，PAC 数据报文由 1、2、3、4、7、8 号节点发往 9 号节点，从图中可以看到在线节点的网络性能参数动态的变化，同时全网的网络性能参数对各节点的累加/平均计算结果也显示正确。

(6)打开网络测试界面，如图 4-13 所示：

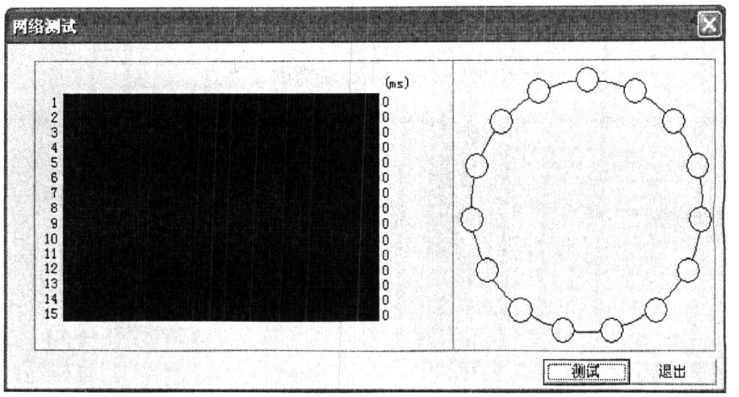

图 4-13 网络测试开始

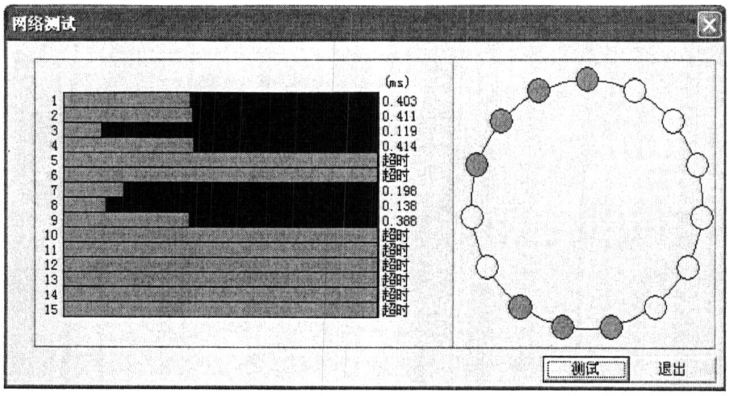

图 4-14 网络测试结束

点击测试按钮后，系统将 1-15 各节点的时延大小按照进度条的方式显示出来，可以看到 1、2、3、4、7、8、9 号节点的时延显示正常，其余节点均为超时。该结果与实际情况相符。同时网络测试界面右侧将当前可达节点以蓝色圆点显示，不可达节点以空心圆点显示。

4.4 本章小结

本章对监控系统的功能进行了测试。按照仿真平台的系统部署搭建测试环境，使用 7 台 PC 运行仿真程序并构成环网，在此基础上对监控系统的功能按模块进行了功能测试，测试结果表明该系统能够正确、有效的监控仿真平台的运行，达到了设计要求。

总结与展望

1. 总结

本文在CRH2型列车通信网络仿真平台的基础上，研究并实现了一个基于以太网仿真的列车通信网络监控系统，该系统可以监控仿真平台的报文传输、令牌循环和网络性能，并完成列车内部控制逻辑的运算。经过测试分析，本文开发的监控系统达到了课题要求，对了解列车通信网络的工作原理和研究ARCNET协议的模拟传输有一定的参考价值。作者在设计完成过程中主要做了以下几个方面的工作：

（1）分析了仿真平台的组成结构以及ARCNET协议的工作原理，根据仿真平台以令牌传递为工作机制的特点对监控系统进行需求分析和总体设计，将监控系统划分为报文监控、报文过滤、节点监控、令牌监控、网络性能监控、控制逻辑运算、网络测试、异常告警8个模块。

（2）对各模块进行了详细设计，说明其实现原理、报文定义和实现流程。

（3）研究了tcpdump过滤原语的生成方式，以逻辑地址转换MAC地址来进行地址判断，以首部地址字节偏移量判断来代替协议判断，实现了仿真平台的数据过滤方式。

（4）分析了CRH2列车通信网络的光纤环路测试方式，根据仿真平台的以太网总线结构以时延测试来对节点连接状态进行判断，实现测试功能的模拟。

（5）对仿真平台运行过程中多令牌异常进行了研究，从报文时序、令牌周期和网络吞吐量3个参数的变化定位异常，并采用通知工作站丢弃令牌的方法解决该问题。

本文设计的监控系统在仿真平台的大吞吐量环境下运行稳定，没有漏包现象发生，在数据统计方法上经过数次改进和优化，监控数据在正确性和精确性均已达到项目要求。

2. 展望

本文作者参与的项目下一步将由纯软件仿真到半实物仿真过渡，本文所实现的监控系统为半实物仿真提供一种监控模型，明确列车通信网络监控所需的监控参数和实现效果。在下一步工作中，监控系统的运行环境将由以太网变为半实物ARCNET网卡组成的光纤环网，对ARCNET网卡的数据捕获方法和工作特性的研究将是下一步工作的重点。

致谢

随着论文的最后完成，我的研究生生活即将结束，感谢求学生涯中关心、支持、帮助过我的老师、同学和朋友们。

由衷的感谢我的导师谭献海老师。谭老师在我三年的读研期间给予我的指导和帮助，令我在学业上获得了很大的进步。作者参与项目中的每一个细节问题都留下了与谭老师深刻探讨的回忆，谭老师对学生平易近人、对工作认真负责的态度将使我终生受益。

感谢我的家人，你们的默默支持是我前进的动力，感谢你们在我成长过程中所付出的一切，在我求学的道路上你们提供给我物质和精神上的无私支持，是你们的鼓励和帮助让我顺利完成学业。

感谢舍友侯世良、皮亮、钱勇对我的帮助、鼓励和照顾，感谢三年同窗的点点滴滴和你们的一路陪伴。

感谢0510实验室、2型车项目组的各位同学，论文的顺利完成离不开你们的帮助，与你们在技术上和学术上的讨论让我受益良多，感谢你们带来愉快的气氛和欢乐的笑声。

最后，特别感谢百忙之中抽出时间参加论文评阅和答辩的各位老师。

在读期间的科研工作及论文发表情况

科研工作：

[1] 2011.3-2012.4 国家科技支撑计划项目“高速列车（II 型车）牵引传动和列车网络系统-网络系统虚拟仿真”，项目组成员，主要负责网络监控软件开发与测试工作.

学术论文：

[1] 孙小盛, 谭献海, 侯世良. 基于以太网仿真的 CRH2 型动车组通信网络监控系统[J]. 铁路计算机应用. 已录用(将在 2013 年第 22 卷第 2 期上发表).

参考文献

[1] 刘铭. 列车通信网络系统形式化建模与验证方法研究[D]. 哈尔滨工程大学, 2011.

[2] 曾嵘，杨卫峰，刘军. 列车分布式网络通信与控制系统[J]. 机车电传动, 2009, (03) : 56-59.

[3] ShangGuan Wei, Cai Baigen, Gou Chenxi. Research on key techniques of high-speed train control system simulation & testing[C]. 2010 International Conference on Mechatronics and Automation(IMCA), Beijing, 2010: 1695-1700.

[4] 常振臣，牛得田，王立德，等. 列车通信网络研究现状及展望[J].电力机车与城轨车辆, 2005, 28(3):5-7.

[5] 张元林. 列车控制网络技术现状与发展趋势[J]. 电力机车与城轨车辆,2006,29(4):1-4.

[6] Hairong Dong, Bin Ning.Automatic train control system development and simulation for high-speed railways[J].IEEE Circuits and Systems Magazine,2010,11(6):10-14.

[7] 刘群欣. TCN 列车网络管理及监视配置软件的研究与实现[J]. 机车电传动, 2010,12(3):71-74.

[8] Yang Fengping, Zhu Qixin. Research of subway's train control system based on TCN[J].Computer and Computing Technologies in Agriculture,2011(4):279-285.

[9] Zeng Yunbing, Teng Yun. A TCN Organization and Command Simulation Training System Based on Network[J].Proceedings of the 2009 International Conference on Signal Processing Systems, 2009(5):121-124.

[10] Yu Zujun, Shi Hongmei, PAi Li.LonWorks-based intelligent train's fire alarming control network[J].Proceedings of the Autonomous Decentralized System,2002,2(11)33-36.

[11] Geng Liang, Guotian Yang. A Kind of Communication Simulation System for WorldFIP Field Intelligent Control Network[C]. International Asia Conference on informatics in Control Automation and Robotics, Bangkok, 2009:386-395.

[12] 魏俊超. ARCNET 列车数据网络的研究[D].西南交通大学, 2011.

[13] 左峰，王立德，聂晓波，等. 基于 ARCNET 的轻轨列车通信网络[J]. 电力机车与城轨车辆,2009,32(6):27-29.

[14] 李国平. 列车通信网络 WTB/MVB 与 LonWorks 的技术比较与应用[J]. 铁道车辆,2004,(1):22-25.

[15] 王利峰，何鸿云，王玉松，等. 基于 ARCNET 的高速列车分级控制系统[J].工业控制计算机, 2007,20(10):11-12.

[16] Zude Zhou, Bing Tang, Cheng Xu. Design of Distributed Industrial Monitoring System Based on Virtual Token Ring[C]. 2nd IEEE Conference on Industrial Electronics and Applications,

Harbin, 2007:598-603.

[17] 张曙光. CRH2 型动车组[M].北京: 中国铁道出版社, 2008.

[18] Zhao Yin, Yue Li, Weixi Xing. Performance Models of E-Business Traffic on the High Speed Train[C]. 2009 International Conference on E-Business and Information System Security, Beijing, 2009:233-237.

[19] 彭权威. 基于 OPNET 的列车通信网络仿真研究[D]. 成都:西南交通大学,2010.

[20] 倪文波，王雪梅. 高速列车网络与控制技术[M]. 成都:西南交通大学出版社,2011.

[21] 刘先恺. CRH2 型 200km/h 动车组列车网络控制系统[J]. 机车电传动, 2008(4): 121-124.

[22] 朱琴跃. 列车通信网络实时性理论与方法研究[D]. 同济大学, 2008.

[23] 聂晓波，王立德. ARCNET 网络系统实时性能分析与研究[J].铁道学报,2011,33(1):58-62.

[24] 王利锋，何鸿云，王玉松，等. 基于 ARCnet 列车网络控制系统的安全性和可靠性分析[J]. 机车电传动, 2007,(06):91-93.

[25] 中国南车株洲电力机车研究所. 列车网络系统随车技师教材[M]. 2008.

[26] Zhang Na, Li, Yanping. Petri net based on behavior of token and its applications[J]. Communications in Computer and Information Science, 2011(227):656-663.

[27] 环形 ARCNET 网络系统的设计与实现[D]. 成都: 西南交通大学, 2008.

[28] W. Richard Stevens. TCP/IP Illustrated Volume1:The Protocols[M].北京:机械工业出版社,2000,4.

[29] 巴全龙，苟先太，姚凤阳. 逻辑双环型 ARCNET 列车通信网络建模与仿真[J].2011,(2): 15-21.

[30] 田大庆，殷国富，陈珂，等. 基于令牌的 ARCNET 故障侦听与解码原理及实现[J]. 2005(9)89-93.

[31] 郭静，杜劲松，高宏亮. 基于 PCI 总线的 ARCNET 数据采集系统设计[J]. 计算机工程与设计,2010(14):136-139.

[32] 管天，卢泽新，白建军. 基于半实物网络仿真的包截获关键技术研究[J].2006(12):99-102.

[33] 郭静，郭涛，杜劲松，等. 基于以太网技术的 ARCNET 数据采集系统设计[J].2009(10):254-157.

[34] 彭国平，杜亚江. 以太网技术在列车通信网络中的应用探讨[J]. 铁道车辆, 2008(12):25-27.

[35] Bandula W. Abeysundara, Ahmed E. Kamal. High-speed local area networks and their performance: a survey. ACM Computing Surveys (CSUR), 2001(6):138-143.

[36] B. Ning. Advanced Train Control Systems[M]. WIT Press, 2010.

[37] 吴汶麒. 基于通信的列车控制系统 IEEE 标准简介[J]. 城市轨道交通研究, 2004,(06):78-81.

[38] Lu Xiaofan, Sun Weijia, Li Huiping. Design and research based on WinPcap network protocol analysis system[C]. 2010 International Conference on Computer, Mechatronics, Control and

Electronic Engineering, Beijing, 2010:486-488.

[39] Xie Kun, Zhang DaFang, Wen JiGang. A real-time network monitor system based on WinPcap[J]. Hunan Daxue Xuebao/Journal of Hunan University Natural Sciences, 2006(33):118-121.

[40] 刘忠文. 基于 Winpcap 的网络信息监听系统研究与实现[D]. 华中科技大学, 2007.

[41] 谭献海等. 网络编程技术及应用[M].北京:清华大学出版社,2006.

[42] Feng Li, Nana, Yu. Design and implementation of TCP/IP protocol learning tool. 2010 Digital Techniques and Systems - 5th International Conference on E-learning and Games, Zhenjiang, 2010:46-52.

[43] 谭思亮.监听与隐藏——网络侦听揭密与数据保护技术[M].北京:人民邮电出版社,2002.

[44] 侯俊杰. 深入浅出 MFC[M]. 华中科技大学出版社, 2001.

[45] 张宇，刘建，李莉. 基于令牌环网络拓扑结构的地铁电动客车网络监控系统[J]. 电力机车与城轨车辆, 2006,(04):31-33.

[46] Yu Gang, Xu, Zhong Wei. Safety requirements model-based safety test automation of train control system of high speed railway in china[J]. Applied Mechanics and Mechanical Engineering, 2010(29-32):2768-2774.

[47] 杨晓娟，贾利民. 列车控制系统架构与技术现状及发展方向[J]. 铁路计算机应用, 2012(03):34-37.

[48] Matsumoto Masayuki, Hosokawa Akiyoshi, Kitamura Satoru. Development of the autonomous decentralized train control system[J]. Autonomous Decentralized Systems and Systems Assurance, 2001(10):133-136.

[49] WinPcap 中文技术文档, http://www.ferrisxu.com/WinPcap/html/main.html.

[50] Hu WenJing, Li Ming; Qiu RunHe. Real-time capturing and measurement of traffic flow based on WinPcap. Journal of Donghua University (English Edition), 2006(23):103-106.