

---

# DWDM 网络网管协议的解析与适配

## 摘 要

密集型光波分复用 (DWDM) 能组合一组光波长用一根光纤进行传送。这是一项用来在现有的光纤骨干网上提高带宽的激光技术。随着光网络的发展，各家光网络设备制造商自成一家，严重阻碍了光网络设备的互联互通。特别是在光网络网管协议上产生歧义，导致不同厂商的设备无法通信。因此解析光网络网管协议，在互通多厂商多协议的光网络设备显的尤为重要。

本文较为系统的阐述了解析 DWDM 设备网管协议的思路：首先在网管端下达命令时，同时用 Wireshark 软件抓包，在网管端记录网管协议栈的数据包；然后研制光监控通道（OSC）数据捕获软件，包括提出需求分析、软件设计、软件测试和修改软件，在 OSC 通道上捕获数据，记录全部数据；进一步分析在 OSC 通道上录到的数据，逐步排除同步字节和公务字节，提取包含网管信息的数据通信通路（DCC）字节，并且重组 DCC 字节；最后借助 Wireshark 开源环境，编写插件模拟解析网管协议。在此基础上提出了验证解析出的网管协议是否正确的方法。

当光传输网的容量越来越大，承载的业务越多，互联互通多区域多厂商的光网络设备意义就越大。互联互通的关键是网管的衔接，除了制定统一的管理标准外，研制一种具有自适应功能的密集型光波复用 (DWDM) 设备，互通不同厂商的设备是本文研究的最终目的。

本人的主要工作包括：

- 1、系统学习 DWDM 设备的原理、组成和功能，熟悉常见 DWDM 网管协议栈中的各种协议。设计解析 DWDM 网管协议的思路和方法。
- 2、完成光监控通道（OSC）数据捕获软件的需求分析和设计，进行软件的测试和修改工作，使用研制的软件做实验记录数据，大量分析和处理数据，猜测网管协议。

---

**3、在 Wireshark 开源环境下，编写插件进行二次开发，模拟解析网管协议。**

**关键词：**密集型光波分复用 网管协议 解析 数据通信通路 光监控通道

---

# **ANALYSE AND ADAPTATION OF DENSE WAVE-LENGTH DIVISION MULTIPLEXING NETWORK MANAGEMENT PROTOCOL**

## **ABSTRACT**

Dense wave-length division multiplexing (DWDM) can transmit a set of light waves in one fiber. This is a optical technique used to raise the bandwidth on existing fiber backbone networks. Along with the development of optical networks, each equipments manufacturer becomes isolated, which seriously obstructs mutual communication of optical networks equipments. Especially the ambiguity of optical network management protocol can not cause communication of different manufacturers' equipments. Consequently analyzing optical network management protocol is more important in mutually communication of optical networks equipments of many manufactures and many protocols.

This paper systematically elaborates the thinking of analyzing DWDM network management protocols. At first, record network management protocol stack data by software of Wireshark grasping the packets, when give orders in networks management end. Then research Optical Supervisory Channel (OSC) capturing data software, including requirements analysis, software design and software test. Catch data on OSC and record total data. Further, analyze the data recorded on OSC, gradually remove the synchronous byte and the official bytes, withdraw the Data Communication Channel (DCC) bytes of network management information and reorganize DCC bytes. Finally under Wireshark of open source environment write a plug-in to analyze network management protocol assumedly. Base on above, put forward the

---

method to verify whether the analyzed protocol is correct or not.

Mutual communication of optical networks equipments of many manufacturers in lots of districts is more important, when the capacity of optical networks is bigger and the business loading on it is more. The key of mutual communication is the linking of network management. In addition to draw the uniform management standard, It is this paper's final aim that a kind of DWDM equipment with adaptive function can communicate different manufacturers' equipments.

My main work includes:

1. Systemically learn the principle, component and function of DWDM equipments. Be familiar with various network management protocols. Design the way of analyzing network management protocol.
2. Complete requirements analysis and design of OSC data catching software, test and modify the software. Do experiments to record data using it. Analyze and process a large number of recorded data. Guess the network management protocol daringly.
3. Under Wireshark of open source environment develop again for a plug-in to analyze network management protocol assumedly.

**KEY WORDS:**Dense Wave-length Division Multiplexing Network Management Protocol Analyze Data Communication Channel Optical Supervisory Channel

## 独创性（或创新性）声明

本人声明所呈交的论文是本人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京邮电大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名: 李海伟 日期: 2009.3.19

## 关于论文使用授权的说明

学位论文作者完全了解北京邮电大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属北京邮电大学。学校有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存、汇编学位论文。（保密的学位论文在解密后遵守此规定）

保密论文注释：本学位论文属于保密在\_\_年解密后适用本授权书。非保密论文注释：本学位论文不属于保密范围，适用本授权书。

本人签名: 李海伟 日期: 2009.3.19

导师签名: 徐大伟 日期: 2009.3.19

---

# 第一章 绪论

## 1.1 DWDM 网络的产生背景和原理概述

### DWDM 网络的产生背景

随着话音业务的飞速增长和各种新业务的不断涌现，特别是 IP 技术的日新月异，网络容量必将会受到严重的挑战。传统的传输网络扩容方法采用空分复用（SDM）或时分复用（TDM）两种方式。

#### 空分复用 SDM (Space Division Multiplexer)

空分复用是靠增加光纤数量的方式线性增加传输的容量，传输设备也线性增加。在光缆制造技术已经非常成熟的今天，几十芯的带状光缆已经比较普遍，而且先进的光纤接续技术也使光缆施工变得简单，但光纤数量的增加无疑仍然给施工以及将来线路的维护带来了诸多不便，并且对于已有的光缆线路，如果没有足够的光纤数量，通过重新敷设光缆来扩容，工程费用将会成倍增长。而且，这种方式并没有充分利用光纤的传输带宽，造成光纤带宽资源的浪费。作为通信网络的建设，不可能总是采用敷设新光纤的方式来扩容，事实上，在工程之初也很难预测日益增长的业务需要和规划应该敷设的光纤数。因此，空分复用的扩容方式是十分受限。

#### 时分复用 TDM (Time Division Multiplexer)

时分复用也是一项比较常用的扩容方式，从传统 PDH 的一次群至四次群的复用，到如今 SDH 的 STM-1、STM-4、STM-16 乃至 STM-64 的复用。通过时分复用技术可以成倍地提高光传输信息的容量，极大地降低了每条电路在设备和线路方面投入的成本，并且采用这种复用方式可以很容易在数据流中抽取某些特定的数字信号，尤其适合在需要采取自愈环保护策略的网络中使用。

但时分复用的扩容方式有两个缺陷：第一是影响业务，即在“全盘”升级至更高的速率等级时，网络接口及其设备需要完全更换，所以在升级的过程中，不得不中断正在运行的设备；第二是速率的升级缺乏灵活性，以 SDH 设备为例，当一个线路速率为 155Mbit/s 的系统被要求提供两个 155Mbit/s 的通道时，

---

就只能将系统升级到 622Mbit/s，即使有两个 155Mbit/s 将被闲置，也没有办法。对于更高速率的时分复用设备，目前成本还较高，并且 40Gbit/s 的 TDM 设备已经达到电子器件的速率极限，即使是 10Gbit/s 的速率，在不同类型光纤中的非线性效应也会对传输产生各种限制。

现在，时分复用技术是一种被普遍采用的扩容方式，它可以通过不断地进行系统速率升级实现扩容的目的，但当达到一定的速率等级时，会由于器件和线路等各方面特性的限制而不得不寻找另外的解决办法。不管是采用空分复用还是时分复用的扩容方式，基本的传输网络均采用传统的 PDH 或 SDH 技术，即采用单一波长的光信号传输，这种传输方式是对光纤容量的一种极大浪费，因为光纤的带宽相对于目前我们利用的单波长信道来讲几乎是无限的。我们一方面在为网络的拥挤不堪而忧心忡忡，另一方面却让大量的网络资源白白浪费。

DWDM 技术就是在这样的背景下应运而生的，它不仅大幅度地增加了网络的容量，而且还充分利用了光纤的宽带资源，减少了网络资源的浪费。

### DWDM 网络的原理概述

DWDM 技术是利用单模光纤的带宽以及低损耗的特性，采用多个波长作为载波，允许各载波信道在光纤内同时传输。与通用的单信道系统相比，密集 WDM (DWDM) 不仅极大地提高了网络系统的通信容量，充分利用了光纤的带宽，而且它具有扩容简单和性能可靠等诸多优点，特别是它可以直接接入多种业务更使得它的应用前景十分光明。

在模拟载波通信系统中，为了充分利用电缆的带宽资源，提高系统的传输容量，通常利用频分复用的方法。即在同一根电缆中同时传输若干个信道的信号，接收端根据各载波频率的不同利用带通滤波器滤出每一个信道的信号。同样，在光纤通信系统中也可以采用光的频分复用的方法来提高系统的传输容量。事实上，这样的复用方法在光纤通信系统中是非常有效的。与模拟的载波通信系统中的频分复用不同的是，在光纤通信系统中是用光波作为信号的载波，根据每一个信道光波的频率（或波长）不同将光纤的低损耗窗口划分成若干个信道，从而在一根光纤中实现多路光信号的复用传输。

由于目前一些光器件（如带宽很窄的滤光器、相干光源等）还不很成熟，因此，要实现光信道非常密集的光频分复用（相干光通信技术）是很困难的，但基于目前的器件水平，已可以实现相隔光信道的频分复用。人们通常把光信道间隔较大（甚至在光纤不同窗口上）的复用称为光波分复用（WDM），再把

在同一窗口中信道间隔较小的 DWDM 称为密集波分复用 (DWDM)。随着科技的进步，现代的技术已经能够实现波长间隔为纳米级的复用，甚至可以实现波长间隔为零点几个纳米级的复用，只是在器件的技术要求上更加严格而已，因此把波长间隔较小的 8 个波、16 个波、32 个波乃至更多个波长的复用称为 DWDM。

如图 1-1, 发送端的光发射机发出波长不同而精度和稳定度满足一定要求的光信号，经过光波长复用器复用在一起送入掺铒光纤功率放大器（掺铒光纤放大器主要用来弥补合波器引起的功率损失和提高光信号的发送功率），再将

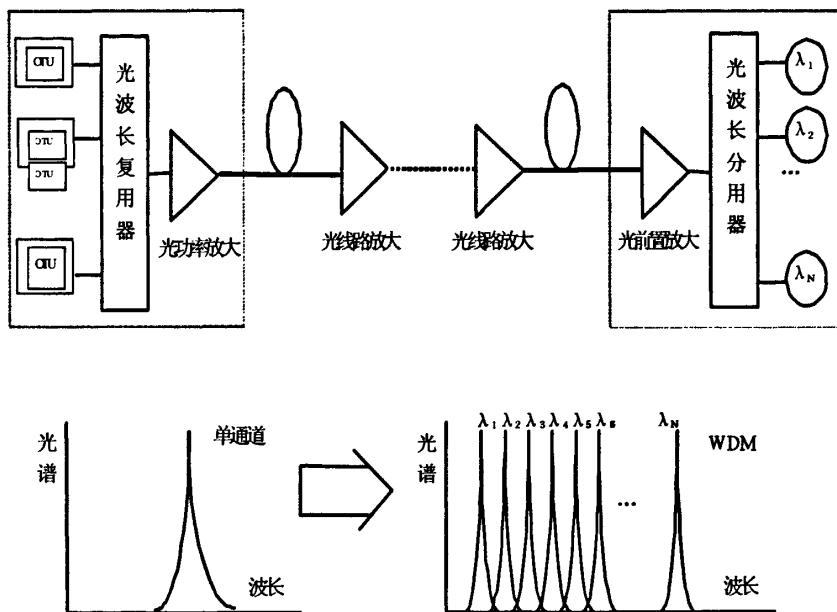


图 1-1

放大后的多路光信号送入光纤传输，中间可以根据情况决定有或没有光线路放大器，到达接收端经光前置放大器（主要用于提高接收灵敏度，以便延长传输距离）放大以后，送入光波长分波器分解出原来的各路光信号[1]。

## 1.2 DWDM 网络的发展

WDM 向更大容量更透明传输的方向发展

传输容量和效率正进一步提高。为了更充分利用光纤低损耗带宽，人们在 WDM 的基础上又提出了密集波分复用技术(DWDM)。DWDM 技术指在当前

---

$1.55\mu m$  波段密集放置更多信道，同时在同一芯光纤中传输。然而，人们现在已不满足这样的密集度，开始想方设法寻找更窄的信道间隔，使传输容量进一步加大，力求更充分地利用光纤的可用带宽。目前，中兴公司研发的大容量、超长距离传输 DWDM 系统容量可升级到  $160\times10Gbit/s$ ，实现超过 5000km 的无电中继传输；烽火通信成功推出  $80\times40GDWDM$  系统，取得了新的突破，并且将该系统成功应用于中国电信  $80\times40Gb/sDWDM$  上海—杭州工程。

另外，WDM 对各种类型的业务透明传输的优势正越来越明显。WDM 技术具有很多优势，在近几年得到快速发展。多波长复用在单模光纤中传输，在大容量长途传输时可大量节约光纤，由于同一光纤中传输的信号波长彼此独立，因而可以传输特性完全不同的信号，完成各种电信业务信号的综合与分离，包括数字信号和模拟信号，以及 PDH 信号和 SDH 信号的综合与分离；波分复用通道对数据透明传输。现在，一个 WDM 系统可以承载多种格式的业务信号，如 ATM、IP 业务等；在网络扩充和发展中，借助 WDM 系统不仅是理想的扩容手段，也是引入宽带新业务，如 CATV 等的有利手段。我们可以利用 WDM 技术实现网络交换和恢复，从而可以实现未来透明的、具有高度生存性的光网络。

### WDM 应用范围将更加广泛

首先，长途骨干传输网对于波长的需求正在不断增加，随着业务的不断增加，对传输距离、波长数量和传输速率的要求不断提高，使得 DWDM 系统在骨干网层面的应用正不断走向深化。目前，常规长距离 WDM 传输系统在我国的干线网络上已经有大量的应用，亚超长距离系统的标准化也已经完成，超长距离的技术和标准都在研究当中，已经有相当多的厂家有了商用化产品。超长距离 WDM 系统可以减少电再生站、光放站的数量，扩大光放站之间的距离，有效降低系统成本，尤其对于我国这样幅员辽阔的国家，WDM 系统的应用将更具有实际意义。因此，在未来，尤其是超长距离 WDM 系统的应用将更加全面。

其次，城域 DWDM 是近年来的新兴应用。随着长途传输骨干网的大规模建设、用户接入及驻地网的宽带化技术的普及，网络的瓶颈逐渐转移到了城域网，原先以承载话音为主的城域传输网络，已无法适应城域数据业务的快速增长，城域网络环境发生了很大变化。由于城域 DWDM 系统具有大容量、光层面保护、支持多业务、后向兼容性等特点，使得城域 DWDM 成为未来城域光网的很重要的发展方向之一。城域 DWDM 网络除了可以提高光纤的利用率之外，另外一个很重要的特点就是，可以提供带保护的波长通道，用于传送数据业务，这比以往通过光纤直连的数据业务具有更好的服务质量保障[2]。

---

总之，可以预见，随着数据、视频等新型业务的迅猛发展，随着 3G 的日益临近，大容量、高速率的 DWDM 系统也一定会迎来更加广阔的发展空间。

### 1.3 DWDM 网管介绍及其发展

网络管理是对网络的性能、品质，进行监测和控制的过程，其目的是最大限度的利用网络资源、提高网络的运行质量和效率，向用户提供良好的通信服务。网络管理涉及到一个网络的服务提供、维护和处理所需要的各种活动。

网络管理的概念来源已久。19世纪末的电话话务员就是电话网的管理员，但是管理的内容和功能都非常有限。从本世纪 50 年代到 70 年代之间，先后出现引起对传统的网络管理方式进行重大变革的三个事件：直接长途拨号 DDD、存储程序控制（SPC）交换机和网络营运系统 OS。网络管理逐渐变成用机器来管理，而管理的内容也越来越多。至于计算机网络的管理工作，则始于 1969 年 ARPANET 诞生之日。由于计算机和通信技术的飞速发展才刺激和促进了网络管理技术的发展。现在一个有效和实用的网络一刻也离不开网络管理。网络管理技术已经成为重要的前沿技术。

在网络中，对各种资源都要进行统一编址。通信的双方应建立一条最经济的通路。当有新结点入网后撤消某个结点时，应及时通知有关结点。网络的使用一般是有偿的，要将资源的使用情况记录下来以便向用户收费。网络还应设有检测系统、差错和故障的报告、故障定位和恢复系统。有时还要求网络设有统计性能分析的功能。下面给出一个网络管理的较公认的定义[3]。

**定义：**对网络的性能、品质，进行监测和控制的过程。

**目标：**最大限度的利用网络资源、提高网络的运行质量和效率，向用户提供良好的通信服务。

**内容：**对组成网络的各个单元设备（硬件和软件）的性能进行监测，对网络中流通的业务进行监测，在发现设备故障时进行处理，包括启用备用设备和把业务转移到其他路由等，在发现网络过载（拥塞）时进行调度处理：包括路由调度和业务调度等。

网络管理包括以下五大功能：

**故障管理（Fault Management）：**对网络的运行情况异常和设备安装环境异常进行监测、隔离和修复的一组功能，包括告警监测、故障定位、故障修复及测试等。

---

**配置管理 (Config Management):** 包括提供状态、控制及安装功能：对网络单元的配置、业务投入、开始或者停止业务等进行管理，对网络的状态进行管理等。

**安全管理 (Security Management):** 控制进网和保护网络及网管系统，防止有意和无意的滥用、未经许可的接入和通讯的丢失等：包括接入和用户权限的管理、安全审查及安全告警等。

**性能管理 (Performance Management):** 保证最有效地利用网络资源和网络资源的能力，对网络设备的性能和网络单元的有效性进行评价，并提出评价报告的一组功能。

**计费管理 (Accounting Management):** 处理和计算业务和资源使用情况的记录，并产生客户使用所有业务的计帐报告[4]。

随着我国电信市场的逐步开放，电信传输网开始进入一个崭新的发展阶段，从原先电话交换网的配属地位将逐渐成为可以直面市场的电信网络资源。

传输资源经营观念的转变对传输网络的管理和资源管理都提出了一个新的要求，经过几年的实践和摸索，为适应传输网面向市场的转变，我们提出了安全、可靠、可控、可管的传输网络维护思想。

近几年来我国的电信传输基础网络建设取得了长足的发展，一个以光传输为核心，微波、卫星传输为辅的立体传输网已基本形成。传输网作为话音、数据、IP、多媒体等电信业务网的承载网络，正在朝着大容量、高带宽的方向发展。密集波分复用技术 (DWDM) 的出现和商用，满足了传输网络继续发展的急切要求，自从 98 年开始我国开始引进 DWDM 设备，并在各省长途干线上做应用实验并逐步推广。到 2008 年底，广东省内传输网上，建成的 DWDM 系统已达到十五个，组网方式有环形、链形等，而容量则有八波、十六波、三十二波等型号不等，共使用了包括朗讯、西门子、大唐等六个国内外厂商的波分设备。

我们将光传输网分为三个层面，即光缆线路层（光纤）、全光网络层（光路）和 SDH 链路层（电路），但是在资源管理和业务调度上我们对这三个层面则采取集中管理，因为传输网络运行质量和效率，对其他业务网的服务质量有着极其重要的作用。当光传输网的容量越来越大，承载的业务越来越多，光传输网络的安全性对整个企业的市场、业务和形象的影响也越大。我们认为建设具有集中维护管理功能的网络管理系统是控制和保证传输网络运行质量的主

---

要手段[5]。

#### 1.4 论文内容及结构

本文阐述了 DWDM 网管协议的解析和适配过程，为研制自适应 DWDM 设备奠定了基础，内容概要如下：

第一章，介绍 DWDM 网络和网管的发展，提出了建立具有集中维护管理功能的网络管理系统的迫切需要；

第二章，提出互联互通光网络存在的问题和需要解决的关键技术；

第三章，介绍自适应 DWDM 网络组成结构及其优点；

第四章，设计光监控通道（OSC）数据捕获软件，对该软件进行全面测试；

第五章，应用该软件进行实验捕获并处理数据，提取包含网管信息的数据，编写插件模拟解析高层网管协议；

最后总结全文，提出了当前解析中存在的问题和实际验证解析是否成功的方法。

---

## 第二章 互通 DWDM 设备的问题、背景和关键技术

### 引言

随着光网络的发展，各家光网络设备制造商自成一家，底层网管往往采用自己私有的协议。因此在地震、泥石流等特殊情况下，光网络发生毁坏时，采用其他厂商的设备无法恢复通信。互通多厂商多协议的光网络设备显的难题迫切需要解决。

### 2.1 实验环境简介

DWDM 实验环境为我提供了一个很好的实验平台，解析 DWDM 系统网管协议就在该实验环境下进行。该系统在有三个网络节点，在 DWDM 层是链状，在 SDH 上是环形，可以使用其中任意两个节点之间的通信进行 OSC 网管协议栈的解析。

### 2.2 问题与背景

当今复杂自然环境背景条件下，用于支撑陆地光通信网络的 DWDM、SDH 传输系统将是通信中易损坏和破坏的重点。光网络中设备复杂、供货厂家较多，如何最大限度地恢复受损或者被摧毁的 DWDM、SDH 节点，替换不同厂家的设备，这个问题已经摆到面前。研制互联互通光通信设备，为我国光网络的发展提供技术支撑，已经刻不容缓。

图 2-1 是正常情况下，某厂家 DWDM 设备的连接情况，假定图中整个链路采用的是一家厂商(厂家 A)的设备，端局(甲、丙两地)厂家通过标准的接口将网管和上级的“大网管”互连，实现统一网管的目的。但是底层网元的配置信息、管理信息是任何厂家都是不愿公开的。

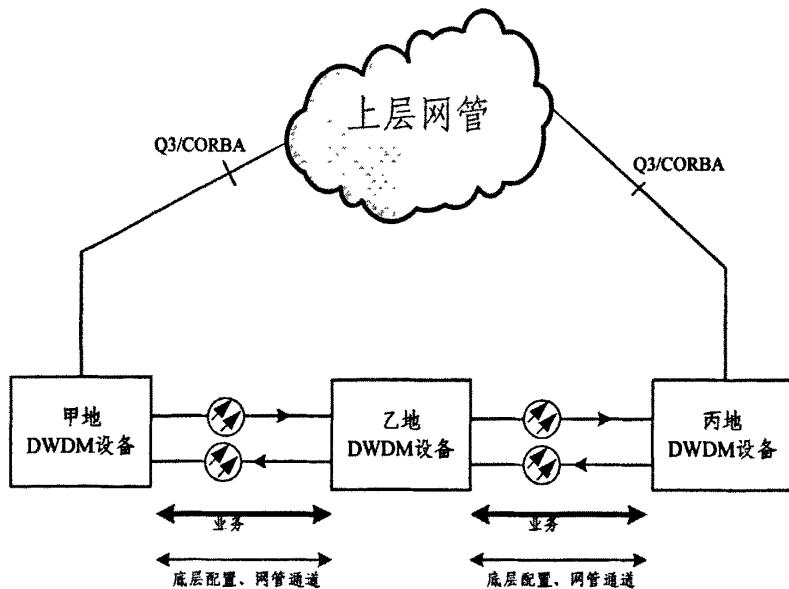
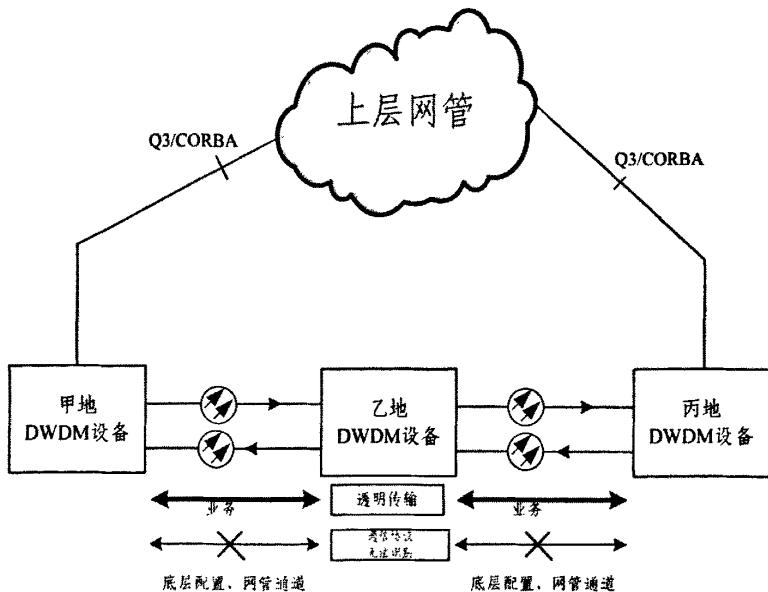


图 2-1 DWDM 互联互通设备的连接关系

从图 2-1 中可以看到, 甲、乙之间或者乙、丙之间的通道包括业务通道(粗线所示)和网管信息通道(细线所示)。网管信息通道中传输的是各个厂家内部自定的协议, 甲地的网管可以通过这个通道实现对乙地设备的管理和配置。这也就是利用常说的光监控通道(OSC)。目前来说, 如果 OSC 通道出现故障, 乙地网管通信通道中断、乙地的设备也无法得到正常的配置信息、网管信息也无法传送, 乙地将成为一个“孤岛”; 同时甲地、丙地之间的网管信息也无法互通。



---

## 图 2-2 DWDM 互联互通设备连接

如图 2-2 所示，若乙地设备遭毁坏，用另外一个厂家(假如是 B 厂家)的设备来代替，不难发现，由于 A、B 厂家间 OSC 通道存在不能互连互通、字节歧义等原因，甲地、丙地均无法对其进行管理和配置，甚至乙地连基本的信息也无法传送。要想网络连通，目前惟一的方法就是仍旧使用原厂家的设备来替换。

研发一种替换设备，最大程度地恢复网管通道，适应尽可能多厂家的 DWDM 设备，已经成为光通信的一个迫切需求。只需要准备一种类型的 DWDM 设备，替代不同厂家的 DWDM 设备，以不变应万变，这也降低了操作的复杂程度。

## 2.3 关键技术

基于现在研究的难题，我提出一下三个关键点，其中 OSC 通道协议栈分析最为重要。本文以后章节将详细讨论 OSC 协议栈的解析。

### 2.3.1 OSC 通道协议分析

OSC 信息的智能识别、提取、缓存、适配与恢复；多厂商、多协议下 DWDM 传送网的网管互连。其中，“OSC 信道”解析是技术关键点。

OSC 信道协议栈的解析入手点是二层协议的解析，其中对以太网有常见三种封装协议。简要介绍三种协议如下：

PPP ( Point-to-Point Protocol ) 点到点协议；LAPS ( Link Access Procedure--SDH) 协议；GFP ( Generic Framing Procedure ) 通用成帧规程协议。这三种协议均为面向无连接的数据链路层，并有相应的国际标准。

#### 1、PPP 协议

PPP 是一种提供点到点链路上传输、封装网络层数据包的数据链路层协议，它处于 TCP/IP 协议栈的第二层，可在支持全双工同异步链路上进行点到点的数据传输。

该协议的特点是：PPP 是数据通信中的一种常用的重要协议，像许多协议一样，PPP 也是一个协议族，主要由三类协议组成：链路控制协议族 (LCP)：用于建立、拆除和监控 PPP 数据链路；网络层控制协议族 (NCP)：用于协商在数据链路上传输的数据包的格式和类型；PPP 扩展协议族：主要用于提供对

---

PPP 功能的进一步支持。此外，PPP 还提供用于网络安全方面的验证协议族（PAP、CHAP），丰富了协议族组群，使得 PPP 协议功能更加完善。

PPP 协议是一套最早的封装协议，技术成熟，已经获得了普遍应用。该协议主要应用于点到点链路上直接相连的两个设备间的通信。例如在拨号上网时，计算机与接入服务器之间采用的是 PPP 连接，DDN 路由器之间的连接通常也采用 PPP 协议。

## 2、LAPS 协议

LAPS 是 HDLC 的一个子集，它包括数据链路服务和协议规范，主要用于 IP over SDH 网络、Ethernet over SDH，通过 LAPS 建立面向字节同步的点对点链路。

该协议的特点是：LAPS 协议主要针对的是大颗粒业务的映射，用于提高封装效率，尤其适用于 GE over SDH 的封装。针对 IP over SDH、Ethernet over SDH 的特点对 PPP-HDLC 进行了简化，封装效率得到了很大提高。

## 3、GFP 协议

GFP 是一种通用映射技术，它可将变长或定长的数据分组，进行统一的适配处理，实现数据业务在多种高速物理传输通道中的传输。

一般来说，一个 GFP 帧包含 GFP (GFPCore Header) 和 GFP 净负荷 (GFP Payload) 两部分。GFP 帧信息头共 4 个字节，其中前 16 bit 为 PLI 域，用于指示净负荷长度，后 16 bit 为 cHEC 域，用于进行帧头校验；GFP 净负荷包含三部分，净负荷头为净负荷的前 4-64 字节，其中包括 16 bit 净负荷类型域、8 bit 净负荷头校验域和 0-60 字节的净负荷扩展头；净负荷信息区包含净负荷帧校验域和经过扰码后的净负荷；GFP 的空闲帧没有 GFP 信息头。

GFP 采用了类似于 ATM 中基于差错控制的帧定界方式，这与利用 HDLC 成帧方式的适配方法所采用的帧定界方式不同。基于 HDLC 的适配技术（如 PPP）需要依赖于某些特殊字符进行帧定界和提供控制信息，对于净负荷内出现的保留字符需要插入转义字符进行区分，从而造成了非确定性的带宽增加。GFP 采用一种基于 HEC 检错的自定界技术来实现协议数据单元的定界。为了能够处理不同长度的协议数据单元，GFP 在帧头开销中提供了一个净负荷长度指示单元，可在数据流中方便地提取出封装好的协议数据单元。这种显示帧长度指示的方式可减少边界搜索处理时间，这对于有较高同步需求的数据链路来说相当重要。由于 GFP 针对各种长度（包括变长）的用户协议数据单元，并对其进行完整的封装，不需要进行协议数据单元的分段和重组，从而大大地简

---

化了链路层的映射/解映射的逻辑关系。

GFP 是一种通用的适配机制，采用先进的数据信号适配和映射技术，将基于 PDU(Protocol dataunit)的客户信号映射到 SDH/OTN 帧结构之中。

与其它的封装技术相比(如 PPP/LAPS)，GFP 由于帧头长度固定，因此具有更高的封装效率；GFP 采用 HEC 校验，因此具有更高的可靠性；GFP 采用多物理端口复用到同一通道，因此减少了对带宽的需求，GFP 支持点对点和环网结构，GFP 是一种通用的适配机制，因此其是实现各厂商的 MSTP 设备线路互通的基本要求。

三种封装协议，各有特点，各自适用于不同的网络或业务需求。从目前的应用情况来看，采用不同封装协议的设备，往往也是造成网络互通问题的主要原因。在实际系统中，大多数厂商会选用一个或两个选项，有的厂家甚至还会采用私有协议。即使这些设备采用了相同的封装协议，仍会有一些选项差异，如 PPP CRC 编码格式，常常导致实际系统无法互通。

以太网封装格式的互通性十分重要，如果不同厂商的封装格式能够互通，则意味着 GE 以太网业务不仅可以 VC 级联跨越不同厂商的 SDH 网络，而且可以不再要求两端的 SDH 设备采用同一厂家设备。由不同厂商设备组成的 SDH 网络对于以太网业务变成透明通道、为更大范围地组织二层网络都提供了基础。

为了解决不能完全互通的问题，主要采取以下两种方式：第一种是设备厂家统一封装协议。普遍看好最新的 GFP 协议，它代表着未来封装协议的发展方向。从三种数据封装映射方式来看，相对于 PPP、LAPS，GFP 协议的标准化程度更高，适用程度更广，是数据业务封装映射到 SDH/OTN 的标准方式之一，具有良好的市场前景。第二种是处理板采用多种封装协议的可选方式。处理板内置多种封装协议，通过软件转换来适应网络的封装协议。

第二种方式适应于在近期实现不同厂商设备的互联问题。但是作为一项长期的策略，则必须通过统一的 GFP 标准进行全网互通，从而实现以太网端到端业务的灵活调度和穿通。

### 2.3.2 自适应光网络技术

自适应光网络使用先进的光域监测技术，实现复杂环境下的 DWDM 光谱智能识别和分析。自适应技术包括：宽范围(160 波)波长稳定调谐技术、速率

---

调谐、抗路由变化、温漂、抖动的线路码型、自适应偏振模色散补偿、放大器均衡、自适应色度色散补偿等。在破解了“OSC信道”后，实现网管协议的适配需要突破的关键点：解决多波长适配问题，研究自适应偏振模色散补偿、放大器均衡、自适应色度色散补偿等技术。

### 2.3.3 网络资源管理

着重研究“自适应光网络”的网管技术、路由和波长分配算法，互联互通设备的智能管理。网管协议解析出来后，进行智能网管的设计，包括管理站和代理端的设计和实现。

---

### 第三章 自适应 DWDM 的组成结构和优点

#### 引言

自适应 DWDM 设备能自动代替不同厂家的设备，适应尽可能多厂家的 DWDM 设备的互联互通。只需要准备一种类型的 DWDM 设备，替代不同厂家的 DWDM 设备，以不变应万变。解析出的协议在该设备上进行适配，实现不同厂家 DWDM 设备的互通[6]。

#### 3.1 自适应 DWDM 组成结构和功能

本系统由网管计算机和 DWDM 终端设备构成，网管计算机即安装了网管软件的普通 PC 机；DWDM 终端设备采用多机箱分立式结构，每个模块装在一个标准的 19 吋机箱中，所有机箱装在 19 吋机架上。该系统是一套 DWDM 的终端设备，由若干个板卡及板载软件和一台运行网管软件管理端的 PC 组成，可实现波长的动态上下功能，并能接入不同厂商的 WDM 子网，被该子网的网管软件识别，和该子网协同工作。其来路光信号为若干波符合  $32*2.5G$  标准的光信号和一波符合  $160*10Gbps$  标准光信号的合波信号。去路光信号的配置和来路光信号的配置完全相同。下路到本地的光波长需要变成标准的 SDH 波长[7]。自适应 DWDM 系统结构框图如图 3—1。

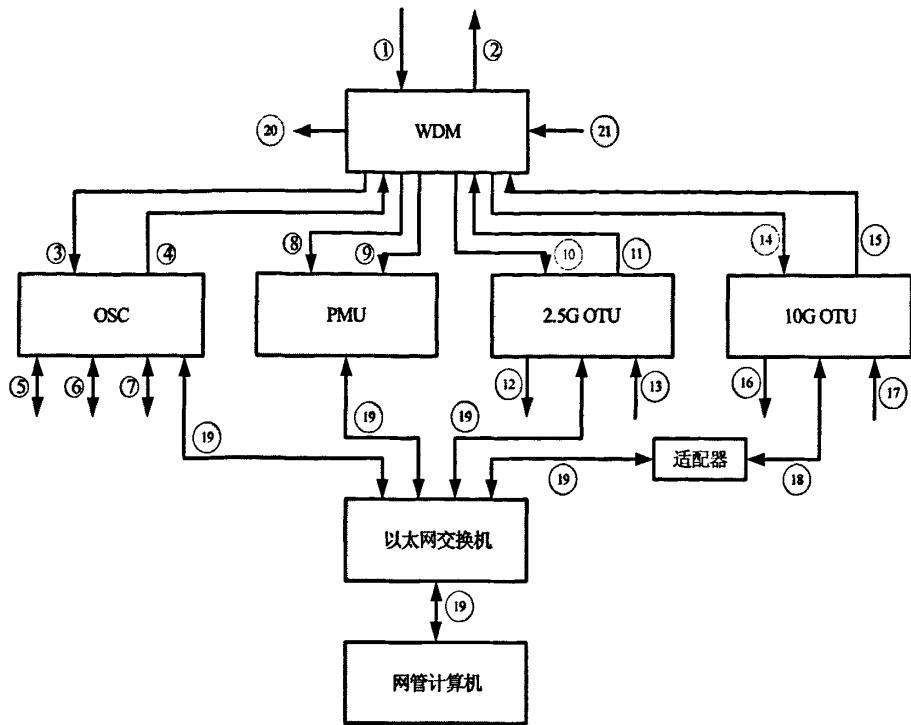


图 3-1 自适应 DWDM 系统结构框图

自适应 DWDM 系统中 DWDM 模块的功能是将输入 DWDM 光信号接入，首先将此光信号按照功率分成两份，一份为 5% 的光功率。这一份光功率输入到 PMU 板，在那里进行光波长的扫描和监控。另一份光由一个滤波器进行 OSC 通道和主光通道的分离。OSC 光信号送到 OSC 处理板，主光通道信号继续在 WDM 板进行处理。主光通道光信号接下来进行 C/L 波段分离，L 波段光信号输出到 L 波段输出接口。C 波段光信号继续在 WDM 板处理，按照功率分成 3 等份，送到 3 个 C 波段输出接口。由这 3 个接口输出的光信号中，一个送给 2.5G 可调 OTU 板处理。另一个送给 10G 可调 OTU 板处理。另外一个悬空。输出方向进行完全对等的反向操作。

OUT 模块的功能是：下路方向将网管指定的某一路下路 DWDM 波长转换成 SDH 波长，上路方向将 SDH 波长转换成相应的 DWDM 波长。

PMU 模块的功能是：周期性和指令性扫描来路和去路光信号中的波长序列，记录每个波长的波长值、功率值、OSNR 等参数。和网管软件进行通信，执行网管指令，上报相关参数。

网管系统模块由管理站和代理端构成。管理站是运行在 PC 机上的一套软

---

件系统。代理端有多个，其形式和作用是相似的，其中的任意一个为运行在电路板上的嵌入式软件。管理站和代理端协同工作，完成以下功能：1) 从来路 OSC 通道及本电路板提取信息，并提交给管理站；2) 管理站接收网管操作员的指令，并下发给代理端，由代理端软硬件协同工作，以实现指令在物理上的执行。

网管适配器模块由适配器板卡和板载软件构成。管理站和代理端协同工作，完成以下功能：1) 从来路 OSC 通道及本电路板提取信息，并提交给管理站；2) 管理站接收网管操作员的指令，并下发给代理端，由代理端软硬件协同工作，以实现指令在物理上的执行。

OSC 模块是用来解析来路 OSC 信息，适配写入去路 OSC 信息；和网管计算机通信；提供公务联络、网络使用者通信、网管信息传递等功能。解析网管协议栈后，最终需要在 OSC 模块上进行适配，对公务字节和 DCC 字节进行处理，然后送给网管计算机进行操作。OSC 通路监控波长是 1510nm，速率是 2Mbit/s，物理接口符合 G703 要求[8]，帧结构和速率符合 G704 规定[9]。

图 3-1 中各个接口描述如下表。

表 3-1 接口描述表

序号	方向	接口描述
1	单向	主光通路输入接口，为若干波符合 32*2.5G 标准的光信号和一波符合 160*10Gbps 标准光信号的合波信号。
2	单向	主光通路输出接口，为若干波符合 32*2.5G 标准的光信号和一波符合 160*10Gbps 标准光信号的合波信号。
3	单向	来路 OSC 信号光接口。
4	单向	去路 OSC 信号光接口。
5	双向	OSC 光中继段公务联络接口，该接口为双向接口，分来路和去路。
6	双向	OSC 光复用段公务联络接口，该接口为双向接口，分来路和去路。
7	双向	OSC 使用者通路接口，该接口为双向接口，分来路和去路。
8	单向	信号 1 除去 OSC 波长后的信号。
9	单向	信号 2 除去 OSC 波长后的信号。

10	单向	C 波段输出接口，为信号 1 中 C 波段所有波长功率三等分后的信号。
11	单向	单波长符合 32*2.5G DWDM 标准的 2.5Gbps 光信号。
12	单向	符合 SDH 标准的 2.5Gbps 下路光信号。
13	单向	符合 SDH 标准的 2.5Gbps 上路光信号。
14	单向	和信号 10 完全相同。
15	单向	单波长符合 160*10G DWDM 标准的 10Gbps 光信号。
16	单向	符合 SDH 标准的 10Gbps 下路光信号。
17	单向	符合 SDH 标准的 10Gbps 上路光信号。
18	双向	基于 232 的双向网管接口。
19	双向	基于以太网的双向网管接口。物理上为以太网接口，协议栈为 Ethernet+UDP+SNMP。
20	单向	和信号 10 完全相同。
21	单向	C 波段光信号输入接口，可以是合波信号，也可以是单波信号。

### 3.2 自适应 DWDM 组成结构优点

在只有 OSC 通道连通而主光通道未曾连通的情况下，公务通道、使用者通道和 DCC 通道可用；冷启动后自动被所在 DWDM 子网的网管系统识别，接收并执行该子网网管系统发出的指令，在该子网中良好地协同工作；在本系统自带的网管系统中显示所在子网的信息，如子网名称、子网设备生产厂家等；显示来路和去路主光通道波长信息、显示各板卡的基本信息等；可通过网管动态指定需要上下路的波长；主光通路中任一 2.5G 波长的动态上下路和波长转换；主光通路中任一 10G 波长的动态上下路和波长转换。

实现解析网管协议，要求在 OSC 通路上捕获数据，记录网管信息。因此研究符合 OSC 接口要求的数据捕获软件是我所做的下一步工作。

---

## 第四章 OSC 信道数据捕获软件

### 引言

解析网管协议需要在 OSC 通路上捕获数据，OSC 上传送着包含网管信息的数据。捕获到的包含网管信息的数据是我们要解析的数据。本节阐述 OSC 信道数据捕获软件的实现过程：借助数字中继板卡，编写驱动程序，开发应用软件，实现数据的捕获。对软件反复测试，直到满足软件需要[10]。

### 4.1 OSC 信道数据捕获软件的需求分析

同时捕获两路 E1 信号，分别存储在硬盘中的两个数据文件中；如有错误发生，则生成相应的告警文件；实现一路 E1 中给定时段的语音重放。物理接口：E1 链路物理接口符合 ITU-T G.703 要求[8]；帧结构符合 ITU-T G.704[9]。

0	1	2	3	…	…	16	17	…	…	29	30	31
---	---	---	---	---	---	----	----	---	---	----	----	----

图 4—1 OSC 帧结构图

在 E1 帧中两路勤务电话占两个时隙，但不能预知具体占用哪两个时隙。其余时隙传输数据信号，但具体内容不可预知。本方案目的为了记录 0 到 31 时隙的所有数据，为后续分析做准备。

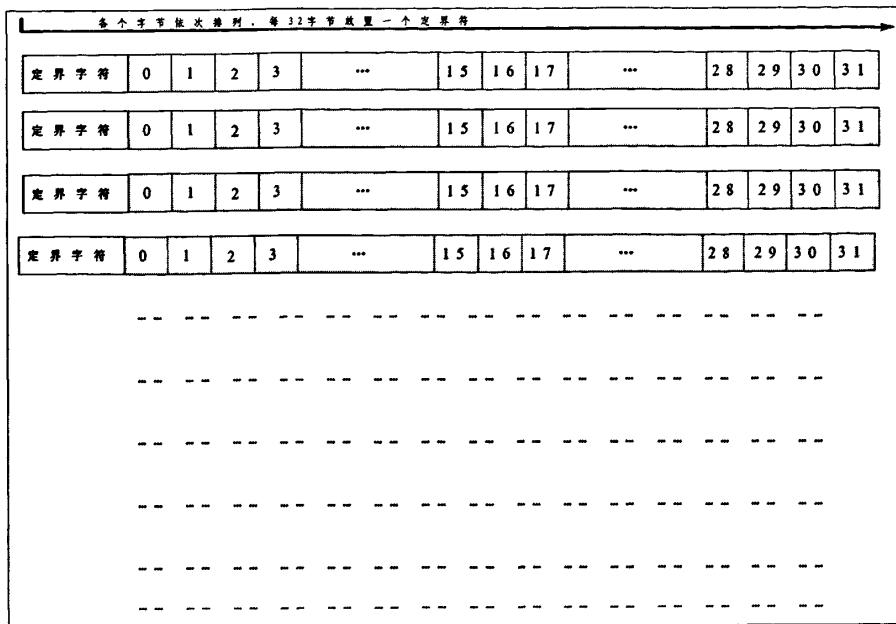


图 4-2 存储格式

在记录数据的过程中生成两个文件，一个数据文件，另外一个是告警文件。本节只说数据文件的格式，告警文件的格式在第七节告警功能中说明。数据文件格式：将 E1 链路上的所有帧准确无误的按照顺序存储于 PC 机的硬盘上，生成数据文件，并且在该数据文件开头加入记录开始和记录结束的时间标签，时间精确到微秒级。E1 数据的存储格式如上图所示。定界字符必须是能够识别的特殊字符，建议以 0x5555 5557 开头作为定界字符，定界字符是宏定义，做到可以随时更改，以达到区别帧内数据，完成查找功能。存储时间应该能达到连续 2 个小时左右，所存文件应该大约是 2GB，并且存储完毕后以开始、结束记录的时间作为文件名，例如：20070910171825-20070910192026.dat(表示开始时刻为 2007 年 9 月 10 号 17 点 18 分 25 秒，结束时刻为 2007 年 9 月 10 号 19 点 20 分 26 秒)。

**显示功能需求：**由用户界面输入开始时间和帧的数目后，将存储的文件打开并且以二进制或十六进制数据显示在屏幕上。例如，20070910171825、800 (表示从 2007 年 9 月 10 号 17 点 18 分 25 秒时刻开始，显示 800 帧)。  
**语音重放功能需求：**实现 0 到 31 时隙的任一指定时隙的语音重放，所指定的时隙序号由用户界面输入。例如、输入时隙序号 0，则对时隙 0 进行语音播放。  
**告警功能需求：**在记录并存储 E1 链路数据时，若发生帧失步/复帧失步/滑码告警等

---

意外情况应有报警功能，并且存储在告警文件中。告警文件名与数据文件同名，但为文本文件。告警文件中所记录的告警要求记录告警序号、告警时刻和告警错误类型。

## 4.2 OSC 信道数据捕获软件的设计

VC 编程可以实现许多功能，诸如播放音频/视频文件、会话通讯等等，但是在存储 E1 数据时，由于数据流过大，内存不够占用，需要特殊的方法实现。本节设计的软件通过 VC 编程调用 WINDOWS API 函数实现对 E1 数据的存储，并且对存储下来的数据进行分析[11]。

### 4.2.1 DN600 硬件平台简介

DN600 是插在计算机 PCI 插槽的一块板卡。该卡上有 8 个 DSP 处理芯片，采用 DSP 架构，处理能力强大，减轻了主机 CPU 的负担。一端是 E1 接口，数据流从此接口输入；另一端是 PCI 插槽，经过 PCI 总线将数据传入内存，在应用程序的控制下实现数据的存储。

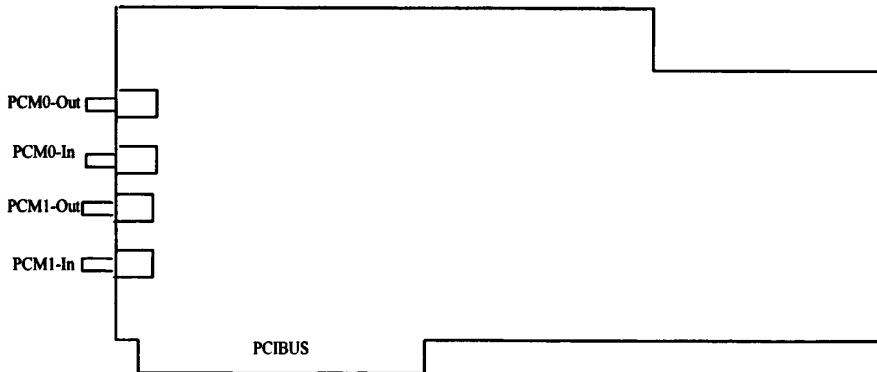


图 4-3 DN600 板卡示意图

### 4.2.2 VC 程序设计

VC 程序是一种基于对话框的编程。程序分为三大模块：存储模块、数据分析模块和显示模块[12]。

#### 1. 存储模块

在硬件板卡启动后，选择 E1 号（有 0、1 和所有 E1 三种选择）、定界字符（定界字符为 16 进制，比如 0x12345678，在界面上只需填入”12345678”，不需

---

换算成 10 进制)、保存的文件夹路径, 点击”捕获”按钮, 开始捕获, 同时显示部分会显示对应通道在“捕获数据中”。

## 2. 数据分析模块

该模块实现对数据的提取、修改、查找和语音重放功能。

选择源文件名、开始帧号(多个开始帧号; 隔开)、要提取的帧数(不同时; 隔开, 注意起始帧号间隔)、时隙号(多个时隙中间用分号; 隔开), 如果需要也可对提取出的文件命名。点击”提取”按钮, 即可实现对指定数据的提取功能。

选择文件名、开始帧号、修改的帧数、时隙号(多个时隙中间用分号; 隔开)、要修改的数据(数据为 16 进制, 比如 0x12345678, 在界面上只需填入”12345678”, 不需换算成 10 进制, 并且时隙号的数量要和要修改的数据对应, 比如时隙号为: 1, 数据则为 xx; 时隙为: 1;2, 数据则为 xxxy; 时隙号为: 1;2;3, 数据则为 xxyyzz; 时隙号为: 1;2;3;4, 数据则为 aabbccdd, 可以多个时隙号修改的数据相同, 比如, 时隙号为 1;2;3;4, 数据为 xx) 点击”修改”, 即可实现对指定数据的修改功能。

输入文件名、开始帧号、显示帧数、要查的数据, 点击查找后, 显示查找到的信息。点击属性后, 显示文件中该数据出现的次数。

选择通道号或提取的文件、文件名、开始帧号、播放帧数, 点击”重放”按钮, 从声卡播放相应的音频数据。

## 3. 显示模块

选择文件名、开始帧号、显示的帧数, 点击”显示”按钮后, 分别以 16 进制和 2 进制在数据显示部分显示, 由于 2 进制的数据比较长, 所以分成两行显示, 每行 16 个字节。点击属性, 显示所显示的数据信息。输入要跳到的帧数, 点击跳到。跳到制定帧的位置, 并且高亮显示该帧。

### 4.2.3 编程实现原理

在 DJDBDK3.4.0 基础上安装 DJDBDK\_SPBY 后, DSP 驱动程序为数据捕获软件提供了 tce1\_32.dll、tce1\_32.h 和 tce1\_32.lib 三个文件, 构成了数据捕获的动态链接库。OSC 数据捕获软件调用 system32 下的动态库 tce1\_32.dll, DJDBDK\Inc 下的 tce1\_32.h 和 DJDBDK\Lib 下的 tce1\_32.lib 三个文件, 实现每个 E1 捕获 31 个时隙的数据。

程序使用轮询捕获数据，每隔 200ms 以上轮询一次。

对 E1 数据的存储可以有很多种实现方法，例如将 E1 转换为以太网，然后导入计算机内存，实现存储。本文独辟蹊径，采用 PCI 总线，通过硬件平台 DN600 借助 VC 编程实现该功能。攻克了内存不足，大文件读写的难题。

### 4.3 OSC 信道数据捕获软件的测试

OSC 物理接口符合 G.703 要求，所以对该软件的进行全面测试，包括不成帧、成帧和成复帧数据的测试。在测试的过程中又提出了添加了发送数据功能，为以后的解析提供了验证方法。

#### 4.3.1 捕获不成帧数据

如图 4-4 连接，误码仪发送 2Mbit/s 的数据流

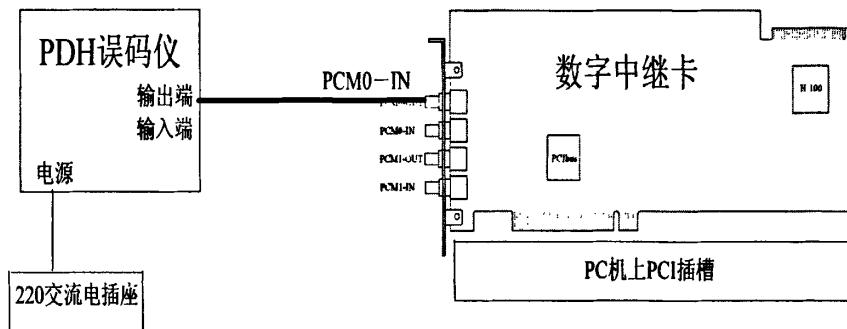


图 4-4

测试记录数据：

表 4-1

发送的数据	接收到的数据	结果
0000 0000	0000 0000	前移了 6 位
0000 0001	0100 0000	前移了 6 位
0000 0010	1000 0000	前移了 6 位
0000 0011	1100 0000	前移了 6 位

0000 0100	0000 0001	前移了 6 位
0000 0101	0100 0001	前移了 6 位
0000 0110	1000 0001	前移了 6 位
0000 0111	1100 0001	前移了 6 位
0000 1000	0000 0010	前移了 6 位
0000 1001	0100 0010	前移了 6 位
0000 1010	1000 0010	前移了 6 位
0000 1011	1100 0010	前移了 6 位
0000 1100	0000 0011	前移了 6 位
10001000(88)	01000100(44)	前移了 3bit
10011001(99)	11001100(CC)	前移了 3bit
11011101(DD)	11101110(EE)	前移了 3bit
11001100(CC)	01100110(66)	前移了 3bit
10001001(89)	01001100(4C)	前移了 3bit
11011100(DC)	11100110(E6)	前移了 3bit

结果：比特移位，生成告警文件，告警类型=1(帧失步)。

分析：DN600 数字中继板卡，对不成帧的数据捕获，有移位现象发生，主要是因为不能识别同步时隙，收发端不能同步造成的。

#### 4.3.2 捕获成帧数据

1. 设置误码仪发送 0XF0AA，所发数据只占用时隙 31；

然后启动 OSC 信道数据捕获软件捕获数据，并打开数据文件如图 4-5 所示：

帧号	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0001	1B	FF	AA																													
0002	DF	FF	FO																													
0003	1B	FF	AA																													
0004	DF	FF	FO																													
0005	1B	FF	AA																													
0006	DF	FF	FO																													
0007	1B	FF	AA																													
0008	DF	FF	FO																													
0009	1B	FF	AA																													
0010	DF	FF	FO																													
0011	1B	FF	AA																													
0012	DF	FF	FO																													

图 4-5

2. 启动示波器，观察波形，显示误码仪所发送的数据，即观察示波器显示

A A D F 和 F 0 1 B

1010 1010 1101 1111 1111 0000 0001 1011 的 HDB3 码型。

结果：示波器显示误码仪发送的数据与捕获到的一致，验证了 DN600 板卡如实捕获了误码仪正在发送的成帧数据。

重复步骤 1、2，所发数据分别占用时隙 30 31，时隙 29 30 31，时隙 28 29 30 31。

重复步骤 1、2，所发数据分别占用时隙 01 02，时隙 01 02 02，时隙 01 02 03 04 和时隙 01。

所得结果与上面一致：验证了 DN600 板卡在 OSC 数据捕获软件的控制下如实捕获了误码仪正在发送的成帧数据。

#### 4.3.3 捕获成复帧数据

1. 设置误码仪发送 0XF0AA，所发数据只占用时隙 01；

然后启动板卡捕获数据，并打开数据文件如图 4-6 显示；

2. 启动示波器，观察波形，显示误码仪所发送的数据，即观察示波器显示

1 B A A 和 D F F 0

0001 1011 1010 1010 1101 1111 1111 0000 的 HDB3 码型。

帧号	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0001	DF	F0	FF	55	FF																											
0002	1B	AA	FF	55	FF																											
0003	DF	F0	FF	55	FF																											
0004	1B	AA	FF	55	FF																											
0005	DF	F0	FF	55	FF																											
0006	1B	AA	FF	55	FF																											
0007	DF	F0	FF	55	FF																											
0008	1B	AA	FF	55	FF																											
0009	DF	F0	FF	55	FF																											
0010	1B	AA	FF	55	FF																											
0011	DF	F0	FF	55	FF																											
0012	1B	AA	FF	08	FF																											

图 4-6

结果：误码仪端 LOS（信号丢失）、AIS(告警指示信号)、MAIS(复帧告警指示信号)、CASL(复帧失步)、MRAI(复帧随路信令远端告警指示)、LOF(丢帧)、RAI(远端告警)、P-AIS(静荷告警指示信号)、LSS(序列同步丢失)都为 0 秒，表示正常。数据显示和示波器的 HDB3 码型一致，并且误码仪端没有告警。

重复步骤 1、2，所发数据分别占用时隙 30 31，时隙 29 30 31，时隙 28 29 30 31。

重复步骤 1、2，所发数据分别占用时隙 01 02，时隙 01 02 02，时隙 01 02 03 04 和时隙 01。

所得结果与上面一致：验证了 DN600 板卡在 OSC 数据捕获软件的控制下如实捕获了误码仪正在发送的成复帧数据。

通过以上全面测试证明了 DN600 板卡在 OSC 数据捕获软件的控制下如实捕获到误码仪正在发送的数据，满足了见需求分析中提出的存储功能。提取、查找和重放数据功能进行大量反复测试，也满足了当初提出的要求。

#### 4.4 OSC 信道数据捕获软件的使用

程序运行后界面如图 4-7：

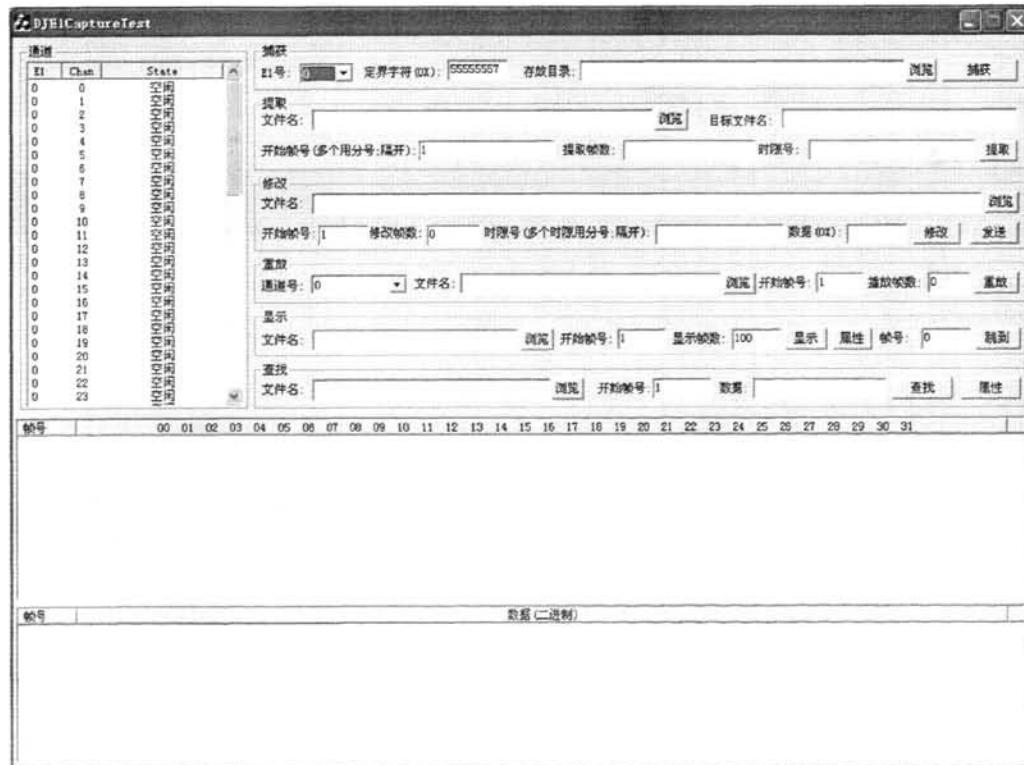


图 4-7

界面大致可以分成三部分，

## 1.显示 E1 通道部分

这部分主要是显示系统有多少个 E1，及每个 E1 的状态(空闲，还是在捕获数据)：

## 2.操作部分

这部分是界面的主要部分，包括：

**捕获：**选择 E1 号(可以选择所有 E1 同时捕获)、定界字符(定界字符为 16 进制，比如 0x12345678，在界面上只需填入“12345678”，不需换算成 10 进制)、保存的文件夹，点击“捕获”按钮，开始捕获，同时 E1 显示部分会显示“捕获数据中”。如图 4-8：

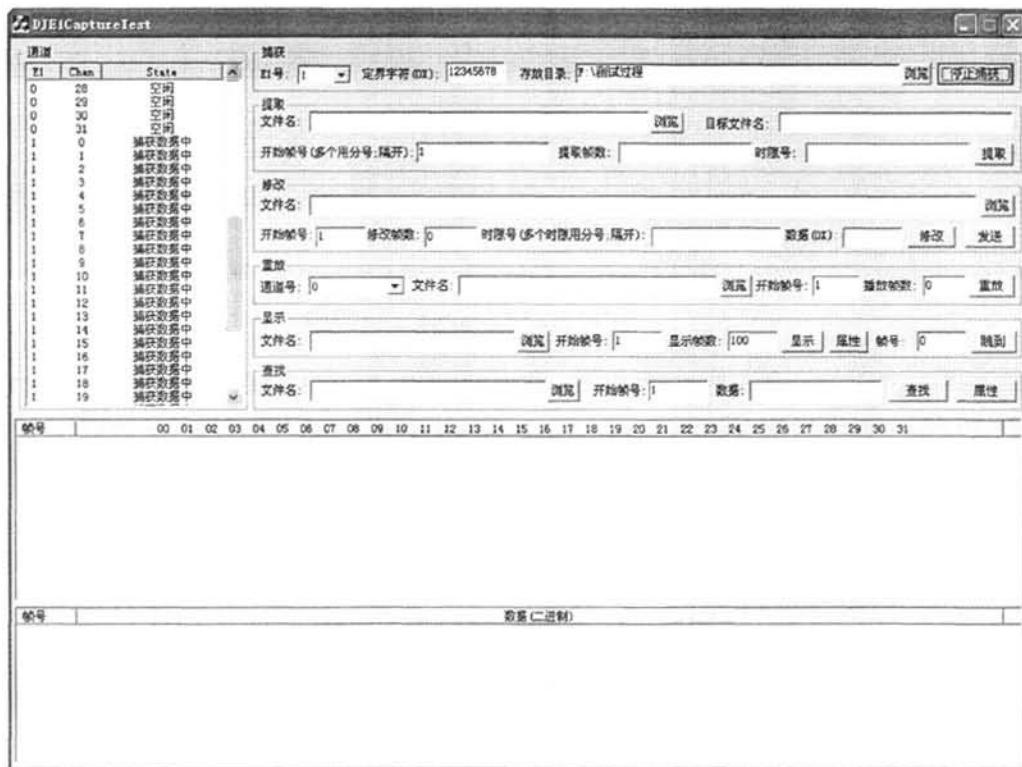


图 4-8

**提取：**选择源文件名、开始帧号、要提取的帧数、时隙号(多个时隙中间用分号;隔开)，点击“提取”按钮，(如果跳帧提取，请保证开始帧号和提出帧数一致，即：有几个开始帧号，就要有几个提取帧数，比如开始帧号为 1;10;30，修改帧数为 2;5;8) 如图 4-9：

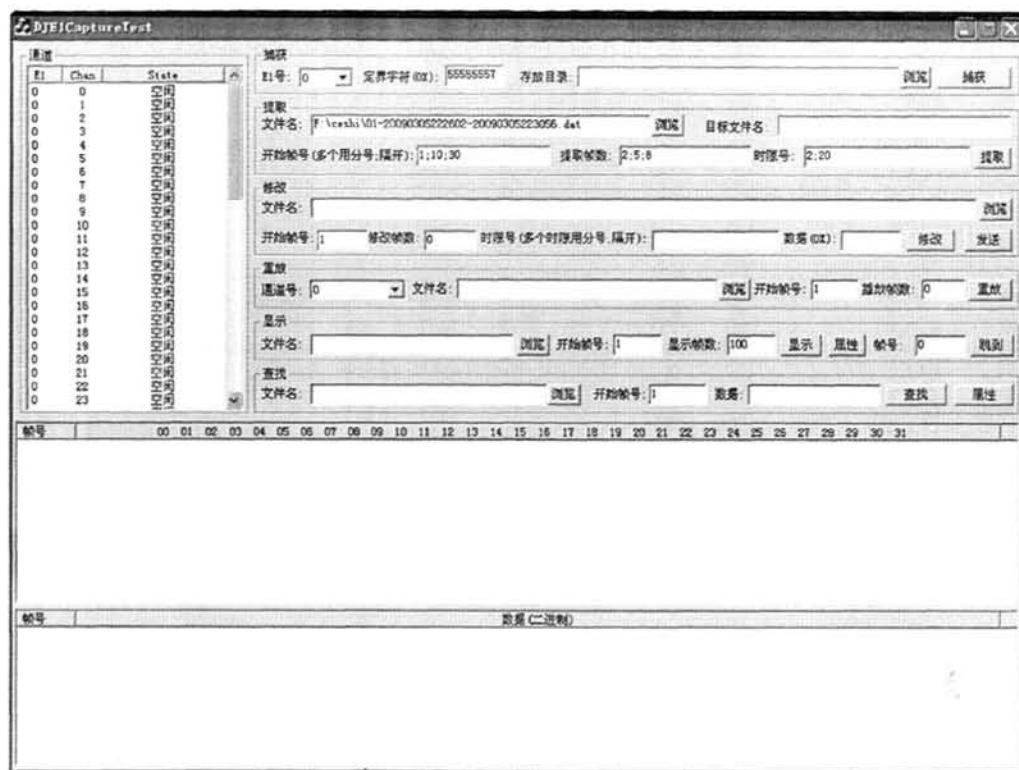


图 4-9

提取数据成功后弹出对话框,如图 4-10, 新文件存放在源文件相同目录下。



图 4-10

**修改:** 选择文件名、开始帧号、修改的帧数、时隙号(多个时隙中间用分号;隔开)、要修改的数据(数据为 16 进制, 比如 0x12345678, 在界面上只需填入"12345678", 不需换算成 10 进制, 一次不能超过 4 个时隙, 并且时隙号的数量要和要修改的数据对应, 比如时隙号为: 1, 数据则为 xx; 时隙为: 1;2, 数据则为 xxxy; 时隙号为: 1;2;3, 数据则为 xxxyz; 时隙号为: 1;2;3;4, 数据则为 aabbccdd, 可以多个时隙号修改的数据相同, 比如, 时隙号为 1;2;3;4, 数据为 xx) 点击“修改”, 如图 4-11:

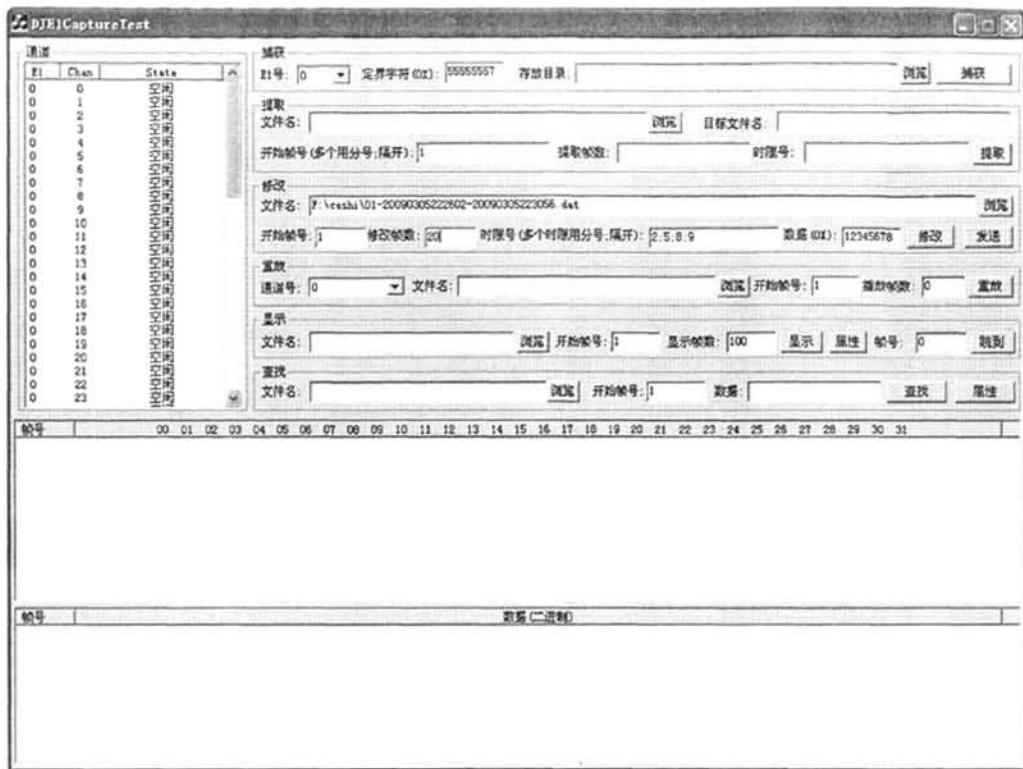


图 4-11

修改完成后，弹出对话框：



图 4-12

如果修改完成后要发送，点击发送，则发送整个数据文件。

重放：选择通道号(也可以选择重放以前提取的文件)、文件名，开始帧号和播放帧数点击“重放”按钮，如图 4-13：

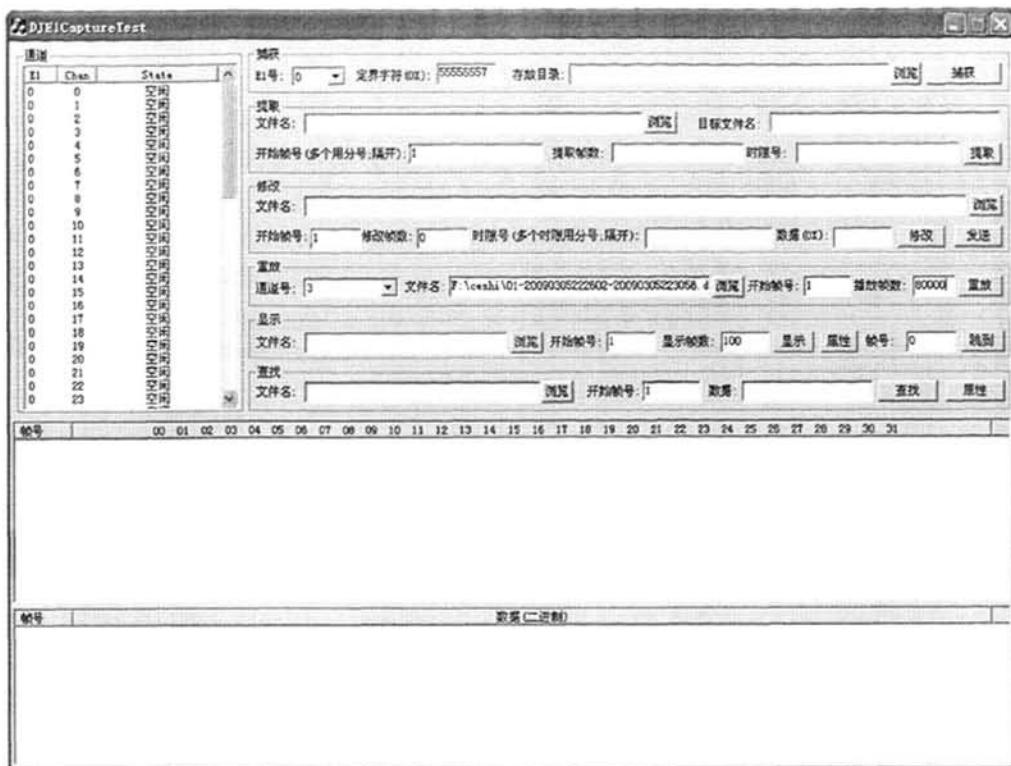


图 4-13

从声卡播放相应的数据。播放结束后，会弹出对话框：



图 4-14

显示：选择文件名、开始帧号和显示的帧数，点击“显示”按钮，如图 4-15：

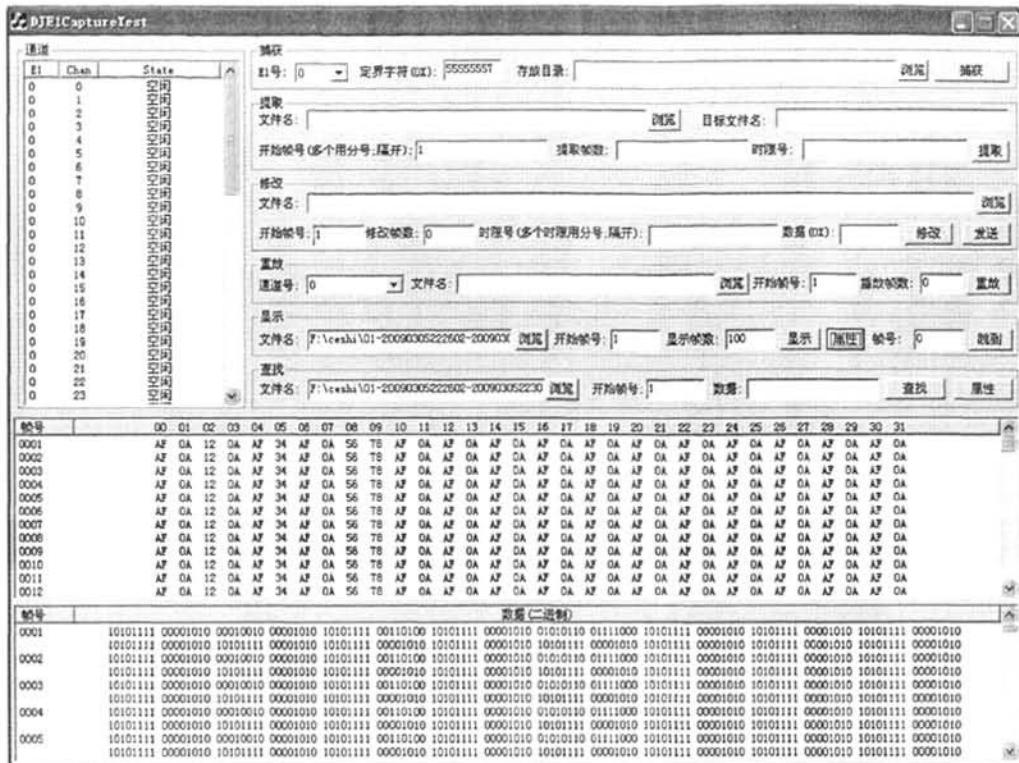


图 4-15

分别以 16 进制和 2 进制在数据显示部分显示，由于 2 进制的数据比较长，所以分成两行显示，每行显示 16 个字节。

点击属性，弹出对话框，显示当前显示文件和帧数。

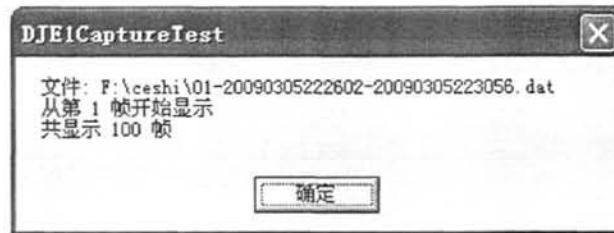


图 4-16

填写帧号，点击“跳到”，则跳到想要查看的数据帧，如图 4-17：

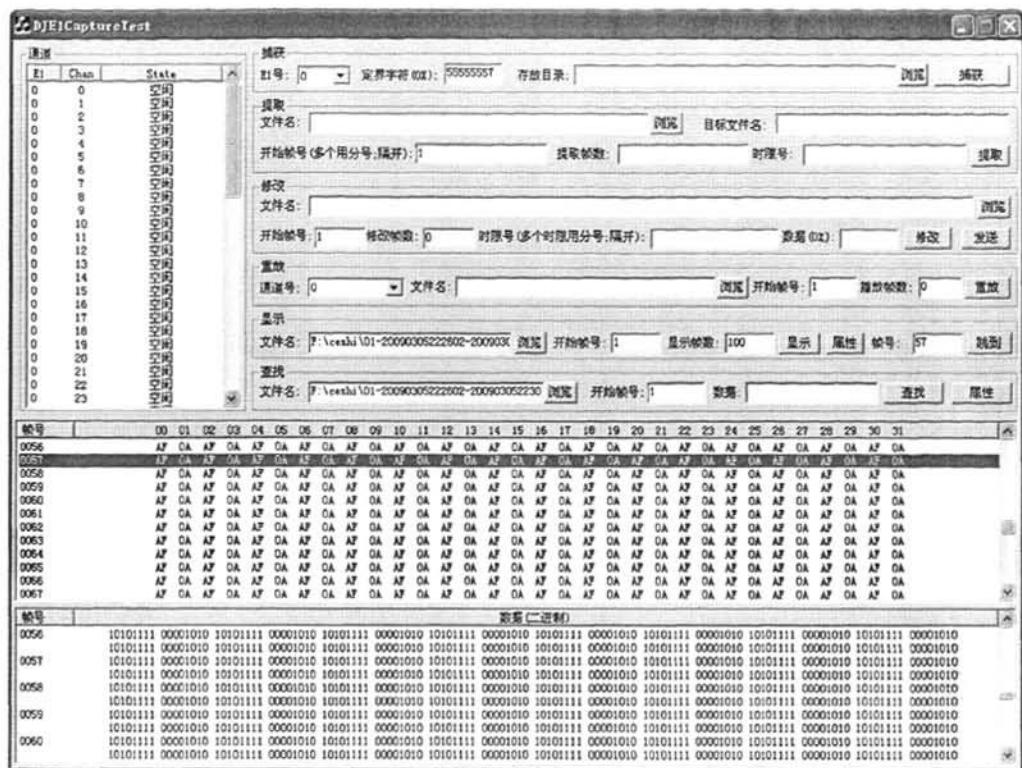


图 4-17

查找：选择文件名、开始帧号和要查找的数据，点击“查找”按钮，如图

4-18：

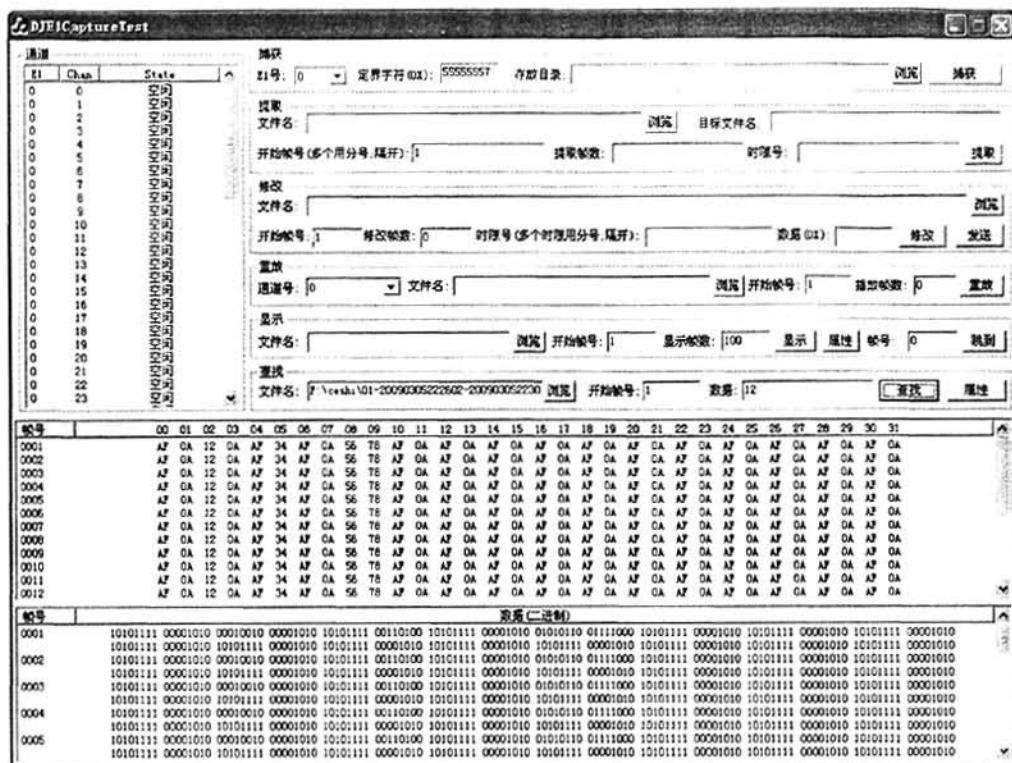


图 4-18

查找完毕，弹出对话框，表示在该文件中查找的数据出现的次数。



图 4-19

点击“属性”，弹出对话框，可以重新查看查找的结果，如图 4-20。



图 4-20

---

### 3.数据显示部分

这部分是调用显示功能时，用十六进制和二进制显示指定的数据

---

## 第五章 DWDM 网管协议解析实现过程

### 引言

使用 OSC 数据捕获软件，在 OSC 线路上记录数据；同时用 Wireshark 软件在网管端抓包，记录网管协议栈。对在 OSC 上记录的数据用软件进行分析，逐步排除同步字节和公务字节，提取包含网管信息的数据通信通路（DCC）字节，并且重组 DCC 字节。对于重组的 DCC 字节后的数据，我们在 Wireshark 开源环境下编写插件，模拟解析网管协议[13]。

### 5.1 OSC 线路上捕获数据和网管端抓包

OADM 系统中，我们在 OSC 上捕获的数据从 TS18 到 TS29 都是 0x7E，由于 DCC 字节总共有 12 个，所以猜测是 DCC 字节，用软件对其提取。通过抓包发现网管协议栈是基于以太网的，包括 Ethernet-IP-UDP-SNMP，由于 Wireshark 软件中包含常见 SNMP 协议的插件，无需我们进一步解析。DWDM 系统中，由于 OSC 线路上可能经过扰码，捕获到的数据有点乱，遇到了解扰的难题，无法识别同步字节，提取包含网管信息的 DCC 字节，但是基于我们对常见的网管协议，我们猜测网管协议名为“xy”的协议，在 Wireshark 开源环境下编写插件，通过 Sniffer 发送数据，成功模拟解析包含网管信息的数据[14]。

#### 5.1.1 OADM 设备上捕获

##### 5.1.1.1 第一次捕获 OSC 数据

测试目的：

1. 搭建三种实验环境，验证 OSC 数据捕获软件能够在 OSC 信道上准确捕获数据，用该软件进行数据分析，提取网管信息。
2. 验证光电转换模块正确完成光电转换功能。
3. 打公务电话，记录拨号音。

测试一：直通测试

1.如图连接测试链路：

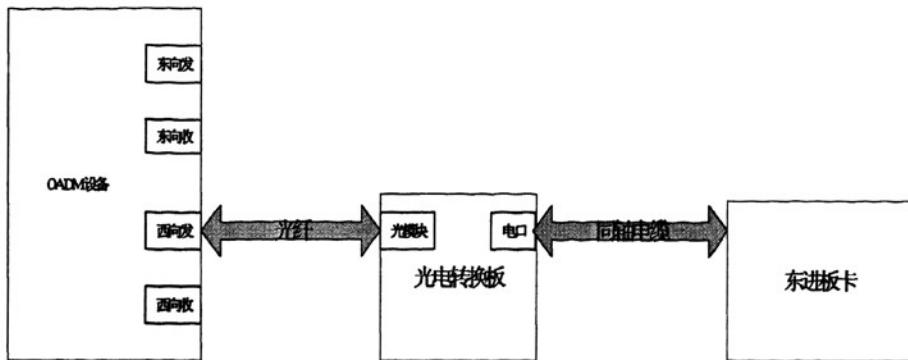


图 5-1

2.结果：捕获数据如下：

帧号	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0001	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0002	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0003	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0004	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0005	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0006	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0007	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0008	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0009	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0011	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
0012	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	

图 5-2

3.分析：东西向接口没有自环，OSC 通道中没有传输数据。

## 测试二：自环东西向测试

1. 如图连接测试链路：

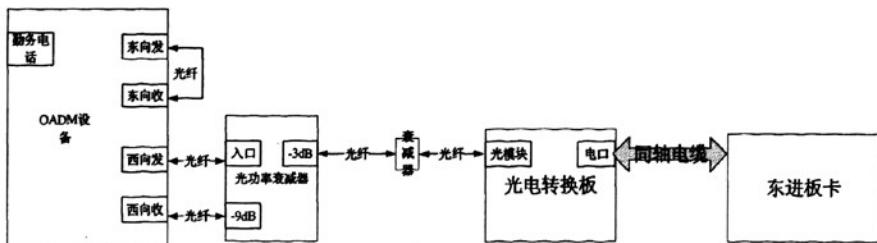


图 5-3

2. 测试过程：

1) 光电转换板使用好的光模块，测得五组数据文件；

2) 光电转换板使用另外一块较差的光模块, 测得七组数据文件;

### 3. 结果如图:

图 5-4

图 5-5

#### 4. 分析：

- 1) 帧同步 0 时隙具有一定规律性，是 0x6C 和 0x00 交替出现，还有其他组合，例如 0x80 和 0x36、0xD8 和 0x00、0x1B 和 0x40；
  - 2) 时隙 1 和时隙 2 总是为 0x00,偶然回出现一个字节的 0x01 或 0x04；
  - 3) 时隙 18 到时隙 28 总是相同的数据，是 0xCF 或 0xE7 等[15]；

### 测试三：公务电话测试

如图连接测试链路：

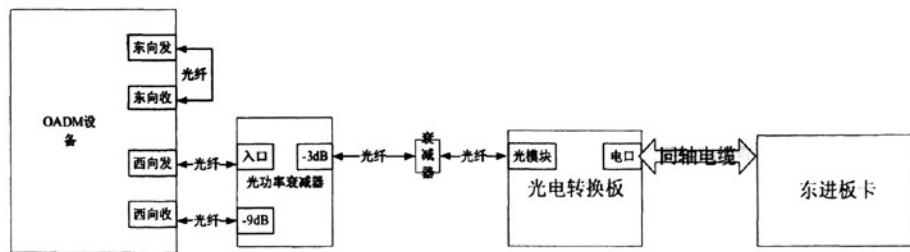


图 5-6

## 2. 测试过程:

光电转换板使用好的光模块，测得三组数据文件：

### 3. 结果:

帧号	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0620	40	00	00	00	EA	ZT	BF	3C	20	00	00	F3	OC	03																		
0621	1B	00	00	00	EA	01	50	00	02	3F	3F	3F	3F	3F	3F	81	01	00	F3	7E	78											
0622	40	00	00	00	2A	01	57	3F	3C	28	00	60	F3	7E	78																	
0623	1B	00	00	00	2A	ZT	BF	3C	20	00	00	F3	OC	03																		
0624	40	00	00	00	2A	ZT	BT	3C	20	00	00	F3	OC	03																		
0625	1B	00	00	00	EA	01	50	00	02	3F	3F	3F	3F	3F	3F	81	01	00	F3	7E	78											
0626	40	00	00	00	6A	01	57	3F	3C	28	00	60	F3	7E	71																	
0627	01	70	00	00	6A	Z4	78	AF	ZT	ZT	ZT	ZT	Z3	54	78	40	00	01	ZT	7E	18	06										
0628	80	00	00	00	51	CF	TE	78	40	00	01	ZT	DE	18	06																	
0629	36	00	00	01	56	02	A0	00	04	TE	72	TE	72	TE	72	02	02	01	ZT	DE	PC	70										
0630	80	00	00	01	54	02	A2	AE	72	TE	72	TE	72	TE	72	50	00	39	ZT	DE	70											
0631	36	00	00	00	05	CF	TE	78	40	00	01	ZT	DE	18	06																	

图 5-7

#### 4. 分析:

- 1) 帧同步 0 时隙具有一定规律性，是 0x1B 和 0x40 交替出现，还有其他组合，例如 0x80 和 0x36、0xD8 和 0x00、0x6C 和 0x00；
  - 2) 时隙 1 和时隙 2 总是为 0x00，偶然回出现一个字节的 0xF0 或 0x04；
  - 3) 时隙 10 和时隙 12 总是相同的数数据，是 0x70 或 0x71 等。

### 5. 问题及原因

数据跟测试三中数据比较，没能发现勒冬时隙

可能是测试二中勤务电话没打通，所以两次测试得到的数据规律一样[16]

### 5.1.1.3 第二次捕获 OSC 数据

## 1 测试目的.

- 1) 找出好的光电转换模块;
  - 2) 使用软件的语言重放功能, 查找公务电话拨号音和其他信令信息。

2.自环东向测试，如下图：

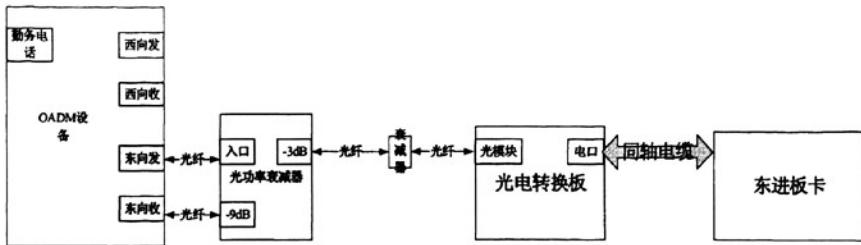


图 5-8

### 3. 测试过程:

- 1) 光电转换板使用编号为 28 光接收模块, 电话挂机状态, 记录 5 组数据;
- 2) 摘起勤务电话, 聆听拨号音, 再记录 5 组数据;
- 3) 打开文件 00-20080708103543-20080708103611.dat, 分别重放 32 个时隙, 和拨号音对比。
- 4) 光电转换板使用编号为 29 光接收模块, 电话挂机状态, 记录 5 组数据;
- 5) 摘起勤务电话, 再记录 5 组数据;
- 6) 打开文件 00-20080708105056-20080708105403.dat, 分别重放 32 个时隙, 和拨号音对比。
- 7) 光电转换板使用编号为 73 光接收模块, 电话挂机状态, 记录 5 组数据;
- 8) 摘起勤务电话, 再记录 5 组数据;
- 9) 打开文件 00-20080708111911-20080708112146.dat, 分别重放 32 个时隙, 和拨号音对比。

### 4. 结果:

- 1) 查看记录的数据文件, 使用 28 和 29 光模块记录的数据, 跟上次录的数据一样, 如下图:

帧号	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0001	80	00	00	00	2F	FF	07	FF	FF	FF	FD	EE	FF	E9	45	D7	E7	E0	0F	F4												
0002	36	00	00	00	B5	FF	FF	FF	FF	CD	2B	FF	FF	FF	FF	30	FC	FD	FF	E4												
0003	80	00	00	00	CA	07	04	00	00	6E	FF	FF	FF	FF	1A	7F	FF	F2	FC	FD	FF	F2										
0004	36	00	00	01	82	05	7F	FF	FC	FD	8F	FF																				
0005	80	00	00	01	A5	FF	CO	5F	FF	FC	38	FF																				
0006	36	00	00	01	99	7F	FF	FD	FF	9A																						
0007	80	00	00	01	3D	7B	FF	ED	1F	FF	FC	FD	FF	F2																		
0008	36	00	00	00	06	15	FF	3F	FF	84	15	5E	FF	FF	FF	ED	FC	C3	F3	F7	FF	02										
0009	00	00	00	00	04	1F	0D	FF	CD	3F	FB	F3	F7	FF	F8																	
0010	18	00	00	00	0F	FF	FF	25	FF	FF	70	16	FF	FF	FF	FF	FF	3E	FB	F3	F0	0F	F4									
0011	00	00	00	00	06	95	F2	C2	05	BF	FF	F8																				
0012	18	00	00	01	17	FF	FF	04	EB	FF	FF	FD	FF	FF	FF	FF	FF	13	F3	F7	FF	02										

图 5-9

2) 使用 73 光模块记录的数据, 如图:

帧号	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0001	40	00	00	00	00	33	7E	FF	FF	38	19	8F	FF	FF	FF	12	36	FF	12	72	72	72	72	72	72	72	72	72	72	72	72	
0002	1B	00	00	00	00	19	2B	FF	FF	FF	FF	77	C7	FF	76	00	4F	FF	FF	72	72	72	72	72	72	72	72	72	72	72	72	
0003	40	00	00	00	00	3C	7F	FB	A5	FF	72	80	7F	38	FF	FF	80	5F	FF	72	72	72	72	72	72	72	72	72	72	72	5E	FF
0004	1B	00	00	00	00	1F	7B	20	A8	FF	FF	FF	AD	C8	FF	FF	FF	11	72	72	72	72	72	72	72	72	72	72	72	72	72	
0005	40	00	00	00	00	46	FF	FF	FB	01	72	FF	FF	FF	12	32	FF	10	07	72	72	72	72	72	72	72	72	72	72	23	FF	
0006	1B	00	00	00	00	86	FF	FF	FF	77	45	FF	38	09	FF	FF	FF	F3	72	72	72	72	72	72	72	72	72	72	72	72	72	
0007	40	00	00	00	00	42	FF	FF	ED	03	FF	FF	FF	72	4A	FF	FF	72	72	72	72	72	72	72	72	72	72	72	72	72	72	
0008	1B	00	00	00	00	1F	FF	FF	FF	20	FF	12	72	72	72	72	72	72	72	72	72	72	3D	74								
0009	40	00	00	00	00	A8	FF	DC	3F	FF	FF	FF	FF	FF	41	FF	FF	72	72	72	72	72	72	72	72	72	72	72	72	72	49	FF
0010	1B	00	00	00	00	71	FF	FF	FF	77	72	BD	5F	FF	FF	FC	01	72	72	72	72	72	72	72	72	72	72	72	72	7A	07	
0011	40	00	00	00	00	84	FF	FF	FF	FF	72	FF	FC	12	FF	FF	FF	72	72	72	72	72	72	72	72	72	72	72	80	4F	FF	
0012	1B	00	00	00	00	63	05	3F	77	00	39	FF	FF	10	17	14	0D	77	72	72	72	72	72	72	72	72	72	72	35	FF		

图 5-10

3) 摘起电话拨号音是每个 4、5 秒两声“滴—滴—”音，重放 32 个时隙听不到拨号音。

### 5.分析：

- 1). TS0 是 0x1B、40 交替的出现，但是 TS16 没有明显的复帧标记迹象，应该是成帧模式，TS01、02、03 都为全零，TS18 到 TS29 为 0x7E,猜测是 DCC 字节；
- 2). 猜测出 DCC 字节，可以用软件的提取功能进行数据提取；
- 3). 好的光模块标号为 73，其中标号为 28 和 29 的有误码；
- 4). 拨号音是交换机传给主叫方的，在我们的设备中拨号音不是以语音编码方式在时隙中传的，所以听不到拨号音[17]；公务电话信令信息应该跟话音在同一时隙上传送，但很难找到具体的数据信息。

#### 5.1.1.3 网管端抓包

通过装有 Wireshark 软件的 PC 机在网管端抓包，当网管下达命令调整 OTU 板卡温度为 50 时，抓到的数据如图 5-11[18]:

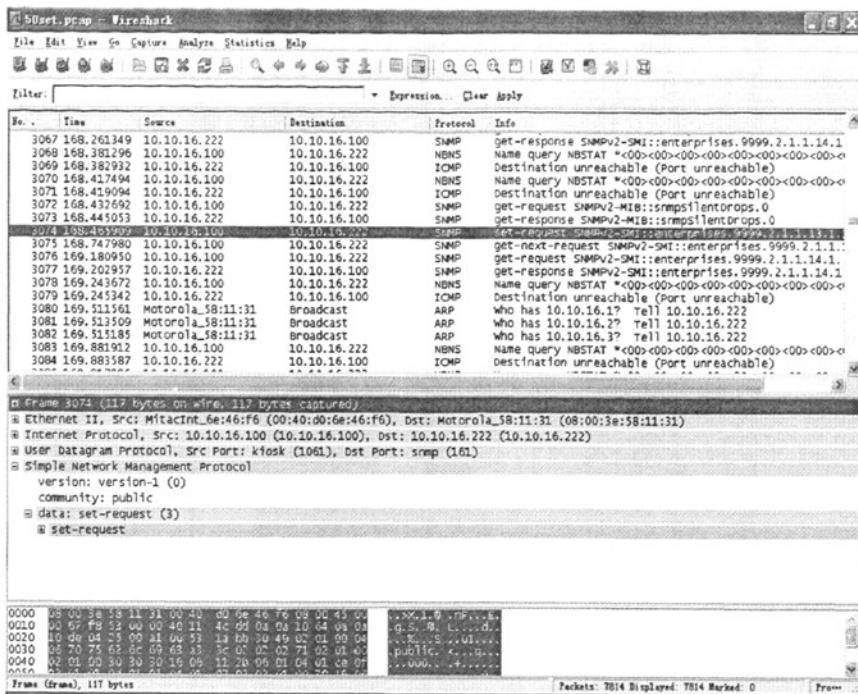


图 5-11

OADM 设备的网管协议栈是基于以太网的，包括 Ethernet-IP-UDP-SNMP。

### 5.1.2 DWDM 设备上捕获

实验目的：

1. 查找同步字节，实现同步记录；
2. 记录公务电话，寻找公务字节；
3. 记录网管信息，查找 DCC 字节，提取要解析的网管数据。

所需器件[19]：

装载语音板卡的台式机、显示器、鼠标、集线器	各一个
安装 Wireshark 软件的计算机	一台
光电转换模块(带光收发模块)	4 路
光分路器（分光比 1/1）	4 个
SC/PC 接头跳线（方形）	2 个
SC/PC 转 FC/PC 接头（方形转圆形）	4 个

所需人员:

人员 a

人员 b

人员 c

已地配合人员 d

以下是实验总体框图和实际光路连接示意图:

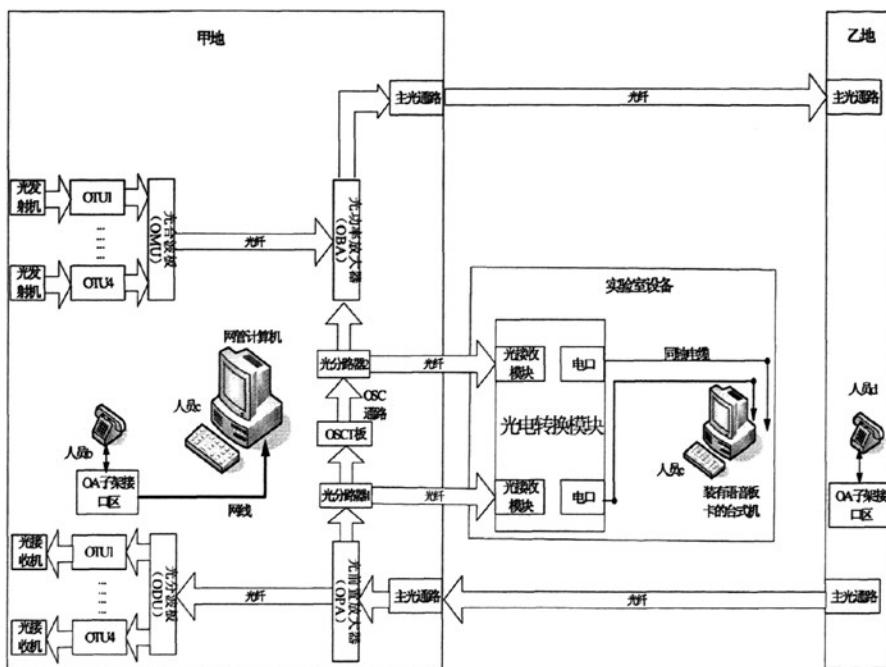


图 5-12 实验环境总体框图

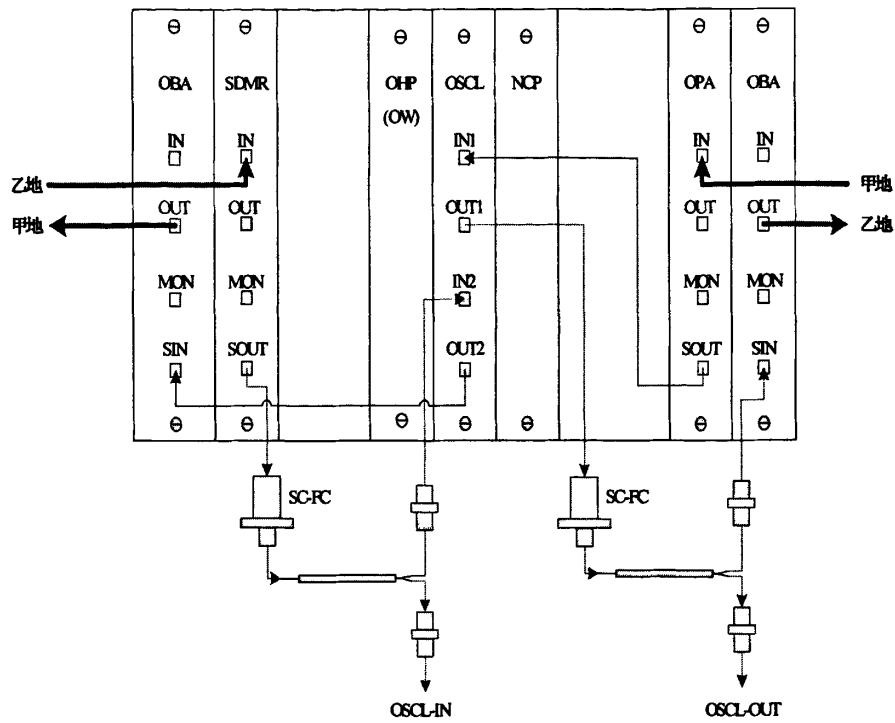


图 5-13 甲地 DWDM 设备光路连接图

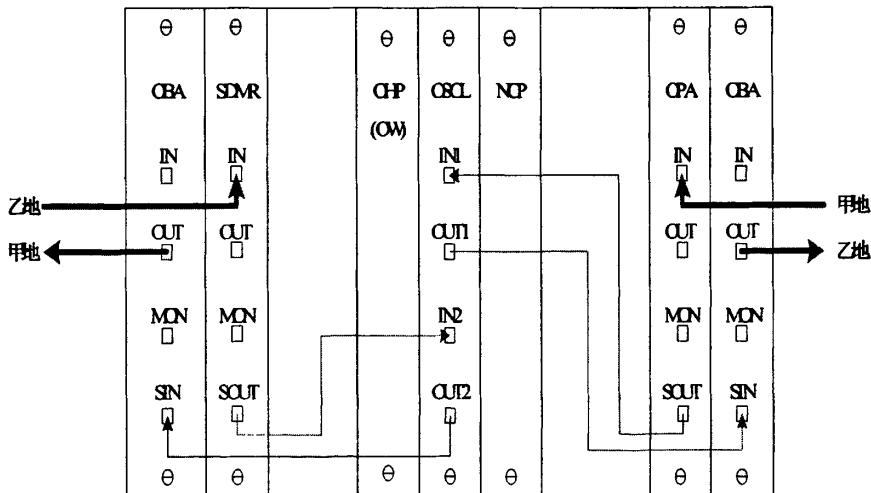


图 5-14 增加光分路器光路连接图

### 5.1.2.1 记录同步

对 DWDM 设备不做任何操作，记录数据。方便查找同步字节，为以后准

确记录数据做好准备。

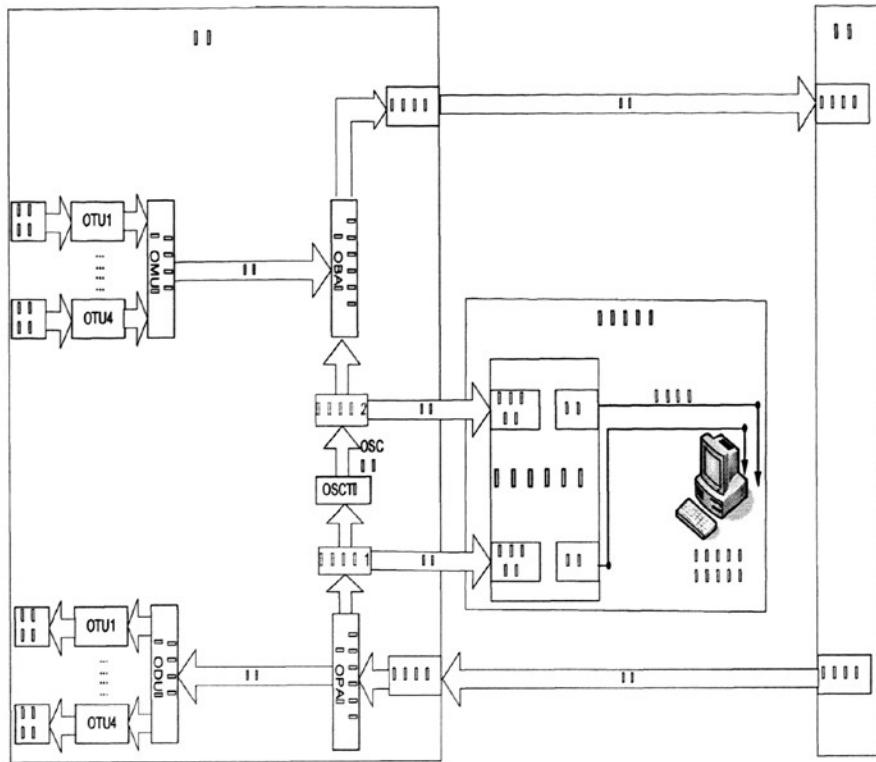


图 5-15

如图 5-15 连接设备，在甲地断开 OSC 通路，在光前置放大器（OPA）和 OSCT 板之间接上光分路器 1，在 OSCT 板和光功率放大器（OBA）连接光分路器 2，开启光电转换模块和装载语音板卡的台式机；

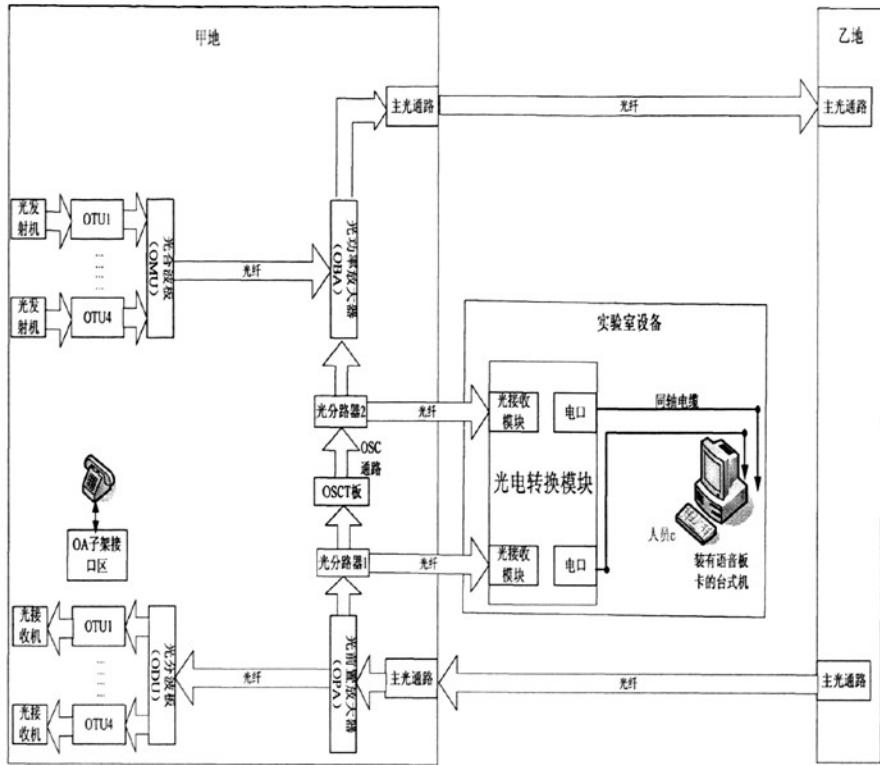


图 5-16

如图 5-16 所示，安排人员 c 捕获 5 到 8 分钟，所录文件放在指定的文件夹下；人员 c 再重复捕获两次，每次捕获 5 到 8 分钟，所录文件分别保存在指定的文件夹下。重复记录样本有利于数据分析。

### 5.1.2.2 记录公务

为了找到包含网管信息的 DCC 字节，我采用排除法，首先找到公务字节将其排除。查找公务字节，两人通话时记录数据，对记录的所有时隙进语音重放，确定公务字节。为了便于分析，摘机、摘机拨号不通话和摘机通话三种情况下分别多次记录。

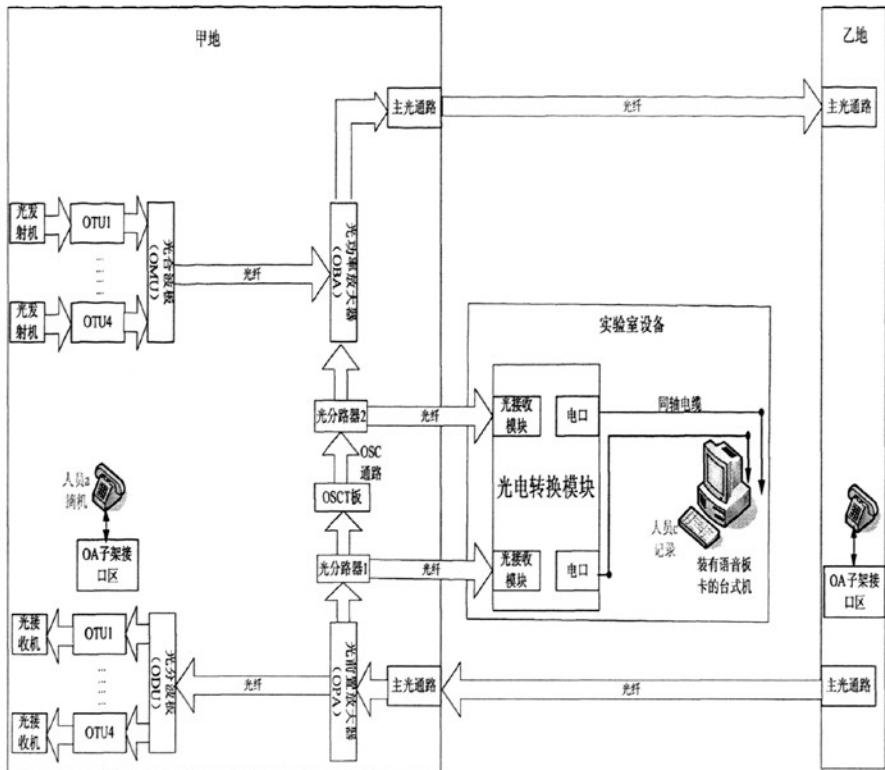


图 5-17

如图 5-17 所示，人员 a 摘起勤务电话，不进行拨号，一分钟后挂机，然后重复该过程两次；人员 c 捕获三次，每次捕获 5 到 8 分钟，所录文件分别放在指定的文件夹下。

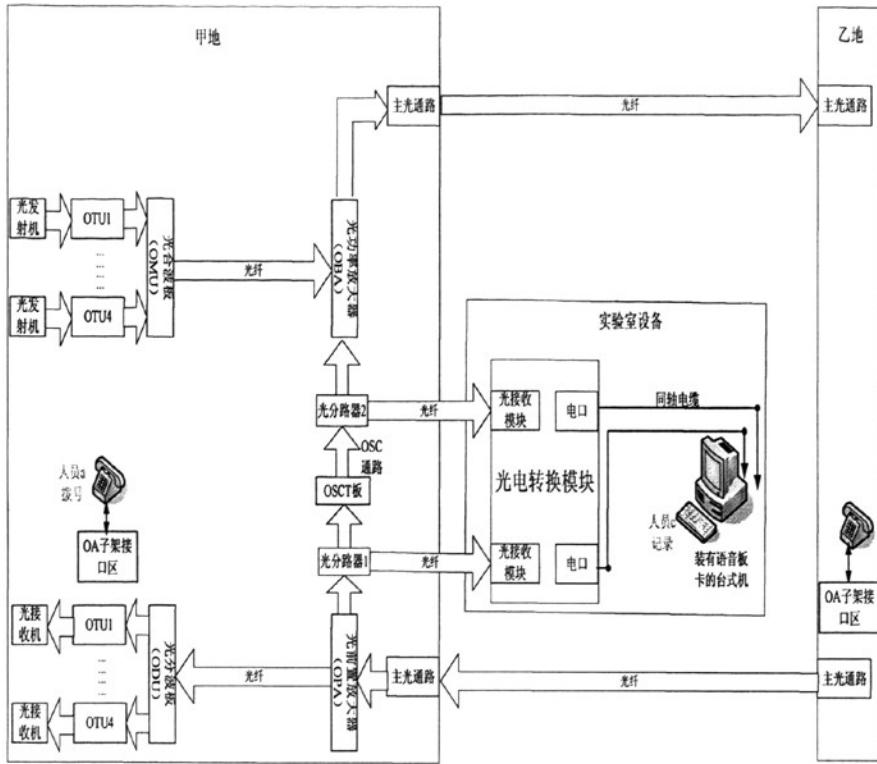


图 5-18

如图 5-18 所示，人员 a 摘机，拨乙地的号码，不进行通话，挂机，然后重复该过程两次；人员 c 捕获三次，每次捕获 5 到 8 分钟，所录文件分别放在指定的文件夹下。

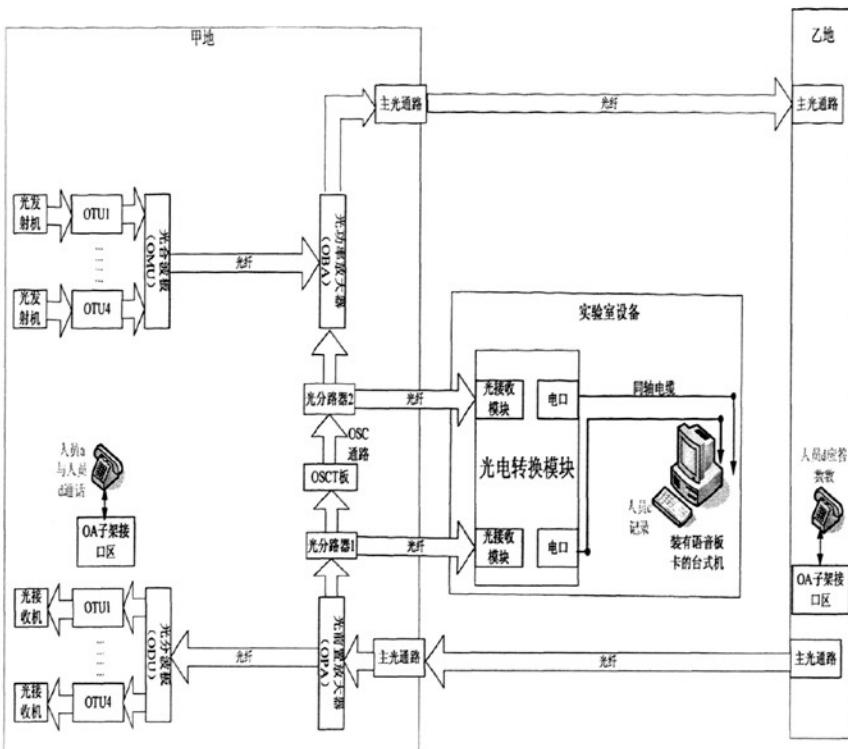


图 5-19

如图 5-19 所示，人员 a 摘起勤务电话向乙地的人员 d 拨号，拨通后呼叫人员 d 姓名，人员 d 听到自己的姓名后循环数数字 0、1，然后重复该过程两次；人员 c 捕获三次，每次捕获 5 到 8 分钟，所录文件分别放在指定的文件夹下。

### 5.1.2.3 记录网管

记录网管信息，分析找到 DCC 字节，为以后解析网管协议提供有用数据。为全面记录网管的各种数据，我分了三种情况进行记录：网管不进行任何操作；网管对本地（甲地）设备进行操作，包括查询和调整命令；网管对远端设备（乙地）设备进行操作。

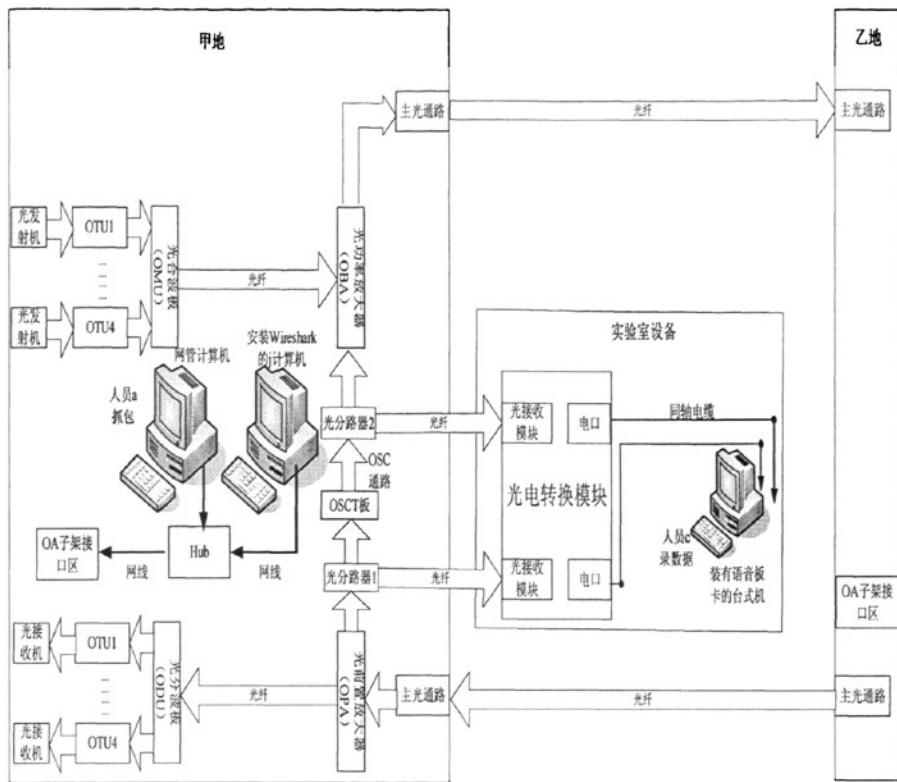


图 5-20

如图 5-20 所示，人员 a 抓包 5 到 8 分钟，重复三次，所录文件分别放在指定的文件夹下。如图 5-20 所示，人员 c 用 OSC 信道数据捕获软件记录 5 到 8 分钟，重复三次，所录文件放在指定的文件夹下；

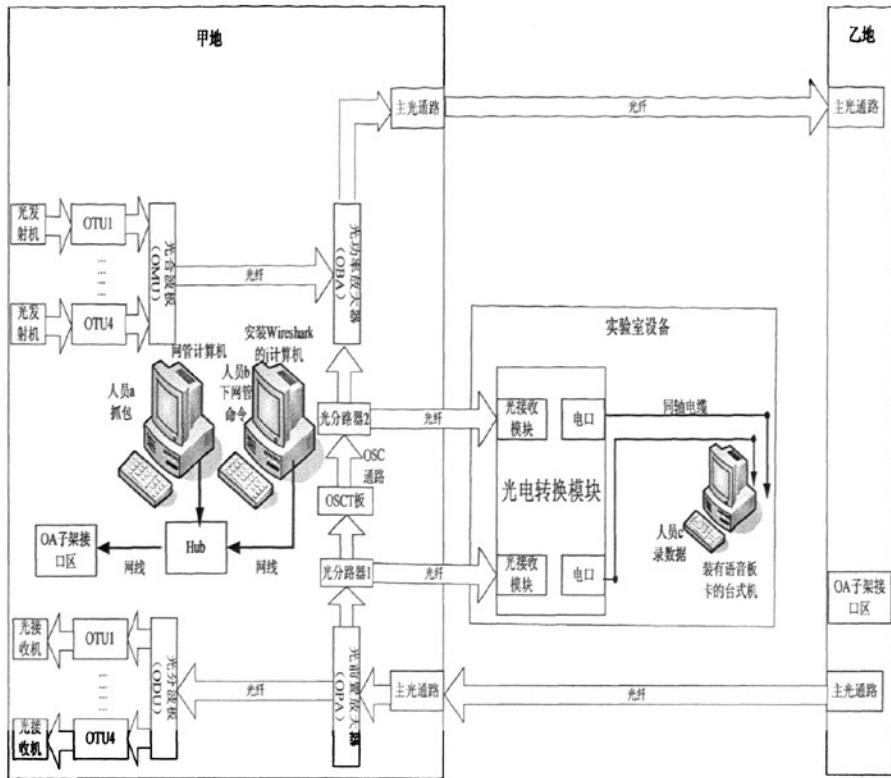


图 5-21

如图 5-21 所示，人员 b 网管端下达命令：查询甲地 OTU 子架 1 槽位 1 的 OTU1 板卡的中心波长，然后重复该过程两次。人员 a 与人员 b 配合，用 Wireshark 软件抓包三次，每次预先开启抓包程序，持续 5 到 8 分钟，所录文件分别放在指定的文件夹下。人员 c 与人员 b 配合，用 OSC 信道数据捕获软件捕获三次，每次预先开启该软件，捕获 5 到 8 分钟，所录文件分别放在指定的文件夹下。

人员 b 网管端下达命令：调整甲地 OTU 子架 1 槽位 1 的 OTU1 板卡的中心波长，然后重复该过程五次。人员 a 与人员 b 配合，用 Wireshark 软件抓包 5 到 8 分钟，所录文件放在指定的文件夹下。人员 c 与人员 b 配合，用 OSC 信道数据捕获软件捕获 5 到 8 分钟，所录文件放在指定的文件夹下。

人员 b 网管端下达命令：查询乙地的 OTU 子架 1 槽位 1 的 OTU1 板卡的中心波长，然后不断重复该过程。人员 a 与人员 b 配合，用 Wireshark 软件抓包三次，每次预先开启抓包程序，持续 5 到 8 分钟，所录文件分别放在指定的文件夹下。人员 c 与人员 b 配合，用 OSC 信道数据捕获软件捕获三次，每次预先开启该软件，捕获 5 到 8 分钟，所录文件分别放在文件夹指定的文件夹下。

人员 b 网管端下达命令：调整乙地的 MCP 板卡备注为 01234567，然后不断重复该过程。人员 a 与人员 b 配合，用 Wireshark 软件抓包 5 到 8 分钟，所录文件放在指定的文件夹下。人员 c 与人员 b 配合，用 OSC 信道数据捕获软件捕获 5 到 8 分钟，所录文件放在指定的文件夹下[20]。

实验结果：如下图

图 5-22

数据没有明显规律，可能是由于 OSC 信道经过扰码，无法识别同步字节，实现准确记录，所以不能提取网管信息，这就为下一步的解析带来困难。

在网管端抓到的数据包见图 5-23

设置网卡名由01234567.pcap - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.1.1.16	192.1.1.1	TCP	RST > diff-port [PSH, ACK] Seq=1 Ack=1 Win=0 Len=0
2	0.138629	192.1.1.1	192.1.3.18	TCP	diff-port > 8050 [ACK] Seq=1 Ack=8 Win=16483 Len=0
3	0.159553	192.1.1.3	192.1.1.1	TCP	8050 > diff-port [PSH, ACK] Seq=1 Ack=1 Win=4096 Len=0
4	0.160282	192.1.1.1	192.1.1.3	TCP	diff-port > 8050 [PSH, ACK] Seq=1 Ack=22 Win=16489 Len=0
5	0.170159	192.1.1.3	192.1.1.1	TCP	8050 > diff-port [ACK] Seq=22 Ack=8 Win=16489 Len=0
6	0.370681	192.1.1.1	192.1.3.18	TCP	diff-port > 8050 [PSH, ACK] Seq=8 Ack=22 Win=16489 Len=0
7	0.571631	192.1.3.18	192.1.1.2	TCP	8050 > diff-port [ACK] Seq=22 Ack=23 Win=4096 Len=0
8	0.706531	192.1.2.18	OSPF-ALL-MCAST.NET	OSPF	Hello Packet
9	0.744791	192.192.1.1	192.192.1.18	TCP	ums > distinct32 [PSH, ACK] Seq=1 Ack=1 Win=17520 Len=0
10	0.872096	192.192.1.18	192.192.1.1	TCP	distinct32 > ums [ACK] Seq=1 Ack=1025 Win=4096 Len=0
11	1.326172	192.192.1.18	192.192.1.1	TCP	distinct32 > ums [PSH, ACK] Seq=1 Ack=1025 Win=4096 Len=0
12	1.451180	192.192.1.1	192.192.1.18	TCP	ums > distinct32 [ACK] Seq=1025 Ack=1025 Win=16496 Len=0
13	1.498500	192.1.1.1	192.1.2.18	TCP	ams > 8050 [SYN] Seq=0 win=16384 Len=0 MSS=1460
14	1.500279	192.192.1.1	192.192.1.18	TCP	ums > distinct32 [PSH, ACK] Seq=1025 Ack=1025 Win=16496 Len=0
15	1.500284	192.1.1.18	192.1.1.1	ICMP	destination unreachable (Network unreachable)
16	1.503720	192.192.1.18	192.192.1.1	TCP	distinct32 > ums [PSH, ACK] Seq=1025 Ack=2049 Win=4096 Len=0
17	1.561270	192.1.1.1	192.1.3.18	TCP	mfserver > distinct32 [ACK] Seq=1 Ack=1 Win=16496 Len=0
18	1.561289	192.1.2.18	192.1.2.3	TCP	mfserver > distinct32 [ACK] Seq=1 Ack=1 Win=16496 Len=0
19	1.561646	192.1.2.3	192.1.3.18	TCP	distinct32 > mfserver [ACK] Seq=1 Ack=1025 Win=4096 Len=0
20	1.591744	192.1.3.18	192.1.2.1	TCP	distinct32 > mfserver [ACK] Seq=1 Ack=1025 Win=4096 Len=0
21	1.597899	192.1.2.3	192.1.2.1	TCP	distinct32 > mfserver [PSH, ACK] Seq=513 Ack=1025 Win=4096 Len=0
22	1.598316	192.192.1.1	192.1.2.3.18	TCP	mfserver > distinct32 [ACK] Seq=1025 Ack=1025 Win=16496 Len=0
23	1.669882	192.192.1.1	192.1.2.1.18	TCP	ums > distinct32 [ACK] Seq=1025 Ack=1025 Win=17520 Len=0
24	2.618984	192.1.3.18	192.1.1.1	TCP	8050 > diff-port [PSH, ACK] Seq=223 Win=4096 Len=0
25	2.763841	192.1.1.1	192.1.3.18	TCP	diff-port > 8050 [ACK] Seq=23 Ack=29 Win=16462 Len=0
26	2.781950	192.1.3.18	192.1.1.1	TCP	8050 > diff-port [PSH, ACK] Seq=29 Ack=23 Win=4096 Len=0

# Frame 1 (61 bytes on wire, 61 bytes captured)  
# Ethernet II, Src: dellComp\_f0:d9:76 (00:08:74:f0:d9:76), Dst: dellComp\_f0:d9:76 (00:08:74:f0:d9:76)  
# Internet Protocol, Src: 192.1.1.18 (192.1.1.18), Dst: 192.1.1.1 (192.1.1.1)  
# Transmission Control Protocol, Src Port: 8050 (8050), Dst Port: diff-port (2251), Seq: 1, Ack: 1, Len: 7  
# Data (7 bytes)

0000 00 08 74 f0 d9 76 00 00 00 33 44 01 08 00 45 00 ...T.V... 3D...E.  
0010 00 2f d1 40 00 03 ff 24 72 c0 01 03 12 c0 01 .../.A./?...T.....  
0020 01 01 1f 72 08 cb 38 2f dc 52 56 3b 48 b9 50 18 ...F.B./R/H.P.  
0030 00 10 15 fe 00 fd fd 0e 00 00 00 00 .....

File "F:\第三阶段所测试第3台的网卡名由01234567.pcap" has been opened by this 01234567.pcap... Packets: 379 Displayed: 379 Marked: 0

图 5-23

---

可见 DWDM 设备的网管协议栈是基于 TCP 的，包括 Ethernet-IP-TCP-?，网管协议是未知的，需要我们解析。

## 5.2 理论分析预测

在 OSC 线路上捕获的数据经过解析整个协议栈后，与网管端抓到的数据包，是一一对应吻合的。这一结论我最终希望看到的结果，也是我解析协议的归宿。如果实现这一结果，最终在 OSC 模块适配后，自适应 DWDM 设备可以被所在的 DWDM 子网的网管系统识别并协同工作，就能够证明我们解析出来的网管协议的却是正确无误的。

但是由于 DWDM 实验环境下，OSC 通道经过扰码，经过软件处理后无法恢复有规律的数据，进而提取 DCC 字节，得到准确的网管信息数据。所以我模拟实际的实验环境进行解析。

## 5.3 模拟解析网管协议

本节阐述模拟解析 OSC 协议栈中的高层网管协议。包括搭建模拟实验环境；猜测整个 OSC 信道协议栈，特别是对高层网管协议的内容进行详细推测；在 Wireshark 开源环境下编写解析高层网管协议的插件；最后模拟解析验证插件是否正确。

### 5.3.1 搭建模拟环境

如图 5-24，A、B 计算机通过集线器连在一起。用装有 Sniffer 的软件从 A 计算机上发出数据，模拟 OSC 通路传过出来的信息；装有解析高层网管协议插件的 Wireshark 在 B 计算机上捕获数据，来验证插件是否正确解析高层网管协议。

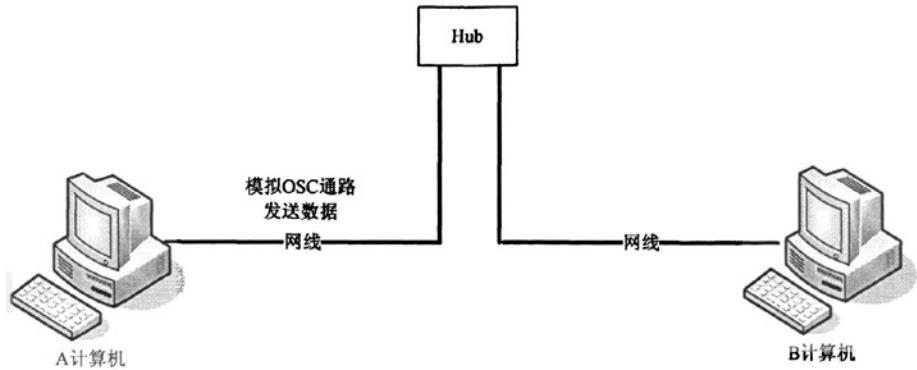


图 5-24

### 5.3.2 猜测高层网管协议

基于我们对网管协议的了解和熟悉，我们可以大胆猜测 OSC 协议栈为 HDLC/Ethernet-IP-UDP-xy，见图 5-25：

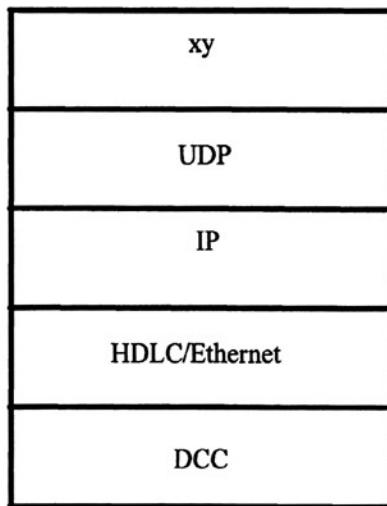


图 5-25

假定高层网管协议名为“xy”，猜测该协议由以下内容组成：

数据包类型占 8bit： 1 表示开始、2 表示结束、3 表示数据；

标志位占 8bit： 0x01 代表开始包，0x02 代表结束包，0x04 代表优先包；

序列号占 16bit；

---

IP 地址占 32bit。

基于对 OADM 设备捕获到的数据中从 TS18 到 TS29 均为 0x7E，我猜测二层协议很可能是 HDLC 封装的 Ethernet 数据包。

### 5.3.3 编写解析高层网管协议的插件

Wireshark 解码器有两种：外置解码器是以插件方式提供的解码函数，不会被编译到 libethereal.dll 中；内置解码器和 Wireshark 软件整体编译，嵌入到该软件当中。内置和外置解码器之间没有本质区别，即使以插件方式编写 packet-xxx.c，也可以在编译时指定 ENABLE\_STATIC 宏，使得 packet-xxx.c 被内置。内置和外置解码器对同一端口解析时，外解码器优先。为了方便开发测试，我先写了插件，即外置解码器，当测试无误时，再将其内置。

第一步，创建 Wireshark 编译环境

1. 安装 Microsoft C 编译器和 Platform SDK
  - a. C compiler: 安装 "Visual C++ 2005 Express Edition"
  - b. Platform SDK : 安装 Platform SDK Server 2003 R2

按照正常流程安装 MSVC，需要注意的是一定要安装 vcvars32.bat 或者在编译 Wireshark 之前手动的调用该文件来完成必须的环境设置（如路径等）。需要注意的是，也可以使用其他版本的 Microsoft C 编译器。但是需要对 Wireshark 开发文件夹下的 config.nmake 文件进行适当设置。具体的细节可参见开发手册。另外 cygwin 的 gcc 编译器是不能通过编译的，请勿使用。

#### 2. 安装 Cygwin

下载并安装 Cygwin，在"Select Packages" 页面时，需要注意在默认安装基础上选择一些附加包，以下就是需要选择的附加包：

- Archive/unzip
- Devel/bison
- Devel/flex
- Interpreters/perl
- Utils/patch
- Web/wget

---

由于没有可以直接使用的 Win32 版本的 bash 可用，在安装时还需要加上 bash。同样，你也可以在这一步时选择将 perl 安装上，这也将在以后的编程会更加便利。

### 3. 安装 Python

下载并安装 python，需要注意最好将 python 安装于默认的目录下，否则需在编译前修改 wireshark 目录下的 config.nmake 文件，调整 python 的路径，才能够顺利通过编译。之所以需要安装 python，是因为 Cygwin 内含的 python 包由于兼容性问题，可能无法在任意环境下顺利运行，因此，安装原版就显得非常必要。安装 Subversion Client，如果只是要编译 Wireshark，那该软件并非必要的，但它可以在管理源码时提供很大的帮助，能够非常方便的对源码进行更新和修改，并时刻保持修改的记录。

### 4. 安装并准备源码

在开始 wireshark 编程之前，一定要保证源码能够顺利的编译运行，至少要保证源码通过编译并运行成功。编译之前需要对源码安装路径下的 config.nmake 做适当的编辑修改：

- a. VERSION\_EXTRA：赋予 wireshark 私人版本号，以区别与官方的版本。
- b. PROGRAM\_FILES：程序的存放路径，一般默认为 C:\Program Files
- c. MSVC VARIANT：找到 #MSVC VARIANT=MSVC2005EE 这一行，并将“#”号去除，指明使用的 C 编译器版本。

如果使用其他的 MS C 编译器，则 c 这一步需采取相应的修改，具体可参加 config.nmake 文件。

准备 command.com

打开 dos 命令行 command.com 并设置路径：

- a. 调用 "C:\Program Files\Microsoft Visual Studio 8\VC\bin\vcvars32.bat" 来设置 MS VC 的编译路径。
- b. 调用 "C:\Program Files\Microsoft Platform SDK for Windows Server 2003 R2\SetEnv.Cmd" 设置 Platform SDK 的环境变量。
- c. 输入 cd C:\wireshark 跳转至 Wireshark 源码的存储路径。

### 5. 检查以上安装的成果。

---

完成 Wireshark 的源码安装后，你可以通过对源码中 Makefile.nmake 使用 verify\_tools 指令来检验上述安装是否成功。需注意的是，要想成功完成这一步，要求已经具备 wireshark 源码以及 bash 和 nmake。

进入 dos 命令行，并输入如下指令：

```
nmake -f Makefile.nmake verify_tools
```

如果一切顺利，将会出现如下的输出信息：

This will check for the various tools needed to build Wireshark:

Checking for required applications:

cl: /cygdrive/c/Program Files/Microsoft Visual Studio/VC98/bin/cl

link: /cygdrive/c/Program Files/Microsoft Visual Studio/VC98/bin/link

nmake: /cygdrive/c/Program Files/Microsoft Visual Studio/VC98/bin/nmake

bash: /usr/bin/bash

bison: /usr/bin/bison

flex: /usr/bin/flex

env: /usr/bin/env

grep: /usr/bin/grep

/usr/bin/find: /usr/bin/find

perl: /usr/bin/perl

env: /usr/bin/env

C:/python24/python.exe: /cygdrive/c/python24/python.exe

sed: /usr/bin/sed

unzip: /usr/bin/unzip

wget: /usr/bin/wget

这里需要尤其注意的是，cygwin 和 MSVC 都具备 link 命令，而我们需要的是 MSVC 中的 link 命令。所以，如果你的 link 命令出现了如/usr/bin/link 的结果，说明链接到的是 cygwin 中的 link，要避免这种情况，只需改变 PATH 路径的环境变量设置，或者直接将 cygwin 中的 link.exe 重命名即可。

---

## 6. 安装库文件 (Libraries)

- 1) 打开 dos 命令行;
- 2) 执行 nmake -f Makefile.nmake verify\_tools verify 指令检查安装及路径是否正确;
- 3) 执行 nmake -f Makefile.nmake setup 安装下载的库文件。

清理源码，在第一次编译 wireshark 之前一定要对源码进行清理：

- a. 打来 dos 命令行
- b. 执行 nmake -f Makefile.nmake distclean 指令完成源码清理
- c. 编译 Wireshark

确定以上准备工作都完成后就可以开始编译 wireshark 源码了。

第二步，编写高层网管协议插件

添加如下代码，编写一个名为“xy”的协议插件。

解析思路是：数据 → epan\_dissect\_run() → dissect\_packet(最高的tvbuff) → dissect\_frame(到达时间、帧序号、优先度) → dissector\_try\_port → dissector\_eth\_common() → dissector\_try\_port() → .....

每个解码器解析对应数据后，创建一个新的 tvbuff 送至 dissector\_try\_port，由 dissector\_try\_port 查找是否有解码器可以解析剩余的数据，直到解析完成。

```
static const value_string packettypenames[] = {  
    { 1, "Initialise" },  
    { 2, "Terminate" },  
    { 3, "Data" },  
    { 0, NULL }  
};  
  
{ &hf_xy_pdu_type,  
{ "XY PDU Type", "xy.type",  
FT_UINT8, BASE_DEC,
```

---

```
VALS(packettypenames), 0x0,
NULL, HFILL }

}

#define XY_START_FLAG 0x01
#define XY_END_FLAG      0x02
#define XY_PRIORITY_FLAG 0x04

static int hf_xy_startflag = -1;
static int hf_xy_endflag = -1;
static int hf_xy_priorityflag = -1;

...
{ &hf_xy_startflag,
{ "XY PDU Start Flags", "xy.flags.start",
FT_BOOLEAN, 8,
NULL, XY_START_FLAG,
NULL, HFILL }
},
{ &hf_xy_endflag,
{ "XY PDU End Flags", "xy.flags.end",
FT_BOOLEAN, 8,
NULL, XY_END_FLAG,
NULL, HFILL }
},
{ &hf_xy_priorityflag,
{ "XY PDU Priority Flags", "xy.flags.priority",
FT_BOOLEAN, 8,
```

---

```
    NULL, XY_PRIORITY_FLAG,
    NULL, HFILL }

},
...
static void
dissect_xy(tvbuff_t *tvb, packet_info *pinfo, proto_tree *tree)
{
    guint8 packet_type = tvb_get_guint8(tvb, 0);

    if (check_col(pinfo->cinfo, COL_PROTOCOL)) {
        col_set_str(pinfo->cinfo, COL_PROTOCOL, "XY");
    }

    /* Clear out stuff in the info column */
    if (check_col(pinfo->cinfo, COL_INFO)) {
        col_clear(pinfo->cinfo, COL_INFO);
    }

    if (check_col(pinfo->cinfo, COL_INFO)) {
        col_add_fstr(pinfo->cinfo, COL_INFO, "Type %s",
                     val_to_str(packet_type, packettypenames, "Unknown (0x%02x)"));
    }

    if (tree) { /* we are being asked for details */
        proto_item *ti = NULL;
        proto_tree *xy_tree = NULL;
        gint offset = 0;

        ti = proto_tree_add_item(tree, proto_xy, tvb, 0, -1, FALSE);
        proto_item_append_text(ti, ", Type %s",
                               val_to_str(packet_type, packettypenames, "Unknown (0x%02x)"));
    }
}
```

---

```
xy_tree = proto_item_add_subtree(ti, ett_xy);
proto_tree_add_item(xy_tree, hf_xy_pdu_type, tvb, offset, 1, FALSE);
offset += 1;
proto_tree_add_item(xy_tree, hf_xy_startflag, tvb, offset, 1, FALSE);
proto_tree_add_item(xy_tree, hf_xy_endflag, tvb, offset, 1, FALSE);
proto_tree_add_item(xy_tree, hf_xy_priorityflag, tvb, offset, 1, FALSE); offset
+= 1;
}}
```

### 第三步，编译插件

1. 打开 dos 命令行，进入插件文件所在目录；
2. 执行 nmake -f Makefile.nmake all 指令进行编译；
3. 等待 Wireshark 编译完成；
4. 将生成的 xy.dll 文件放到指定的

C:\wireshark\wireshark-gtk2\plugins\0.99.7huxiaozi 路径下；

5. 运行 C:\wireshark\wireshark-gtk2\wireshark.exe 来查看能否正常运行。

#### 5.3.4 模拟解析高层网管协议

在模拟环境中，在 A 计算机上运行 Sniffer 软件，在其界面下点 Tools 菜单下的 Pakets Generator，如图 5—26：

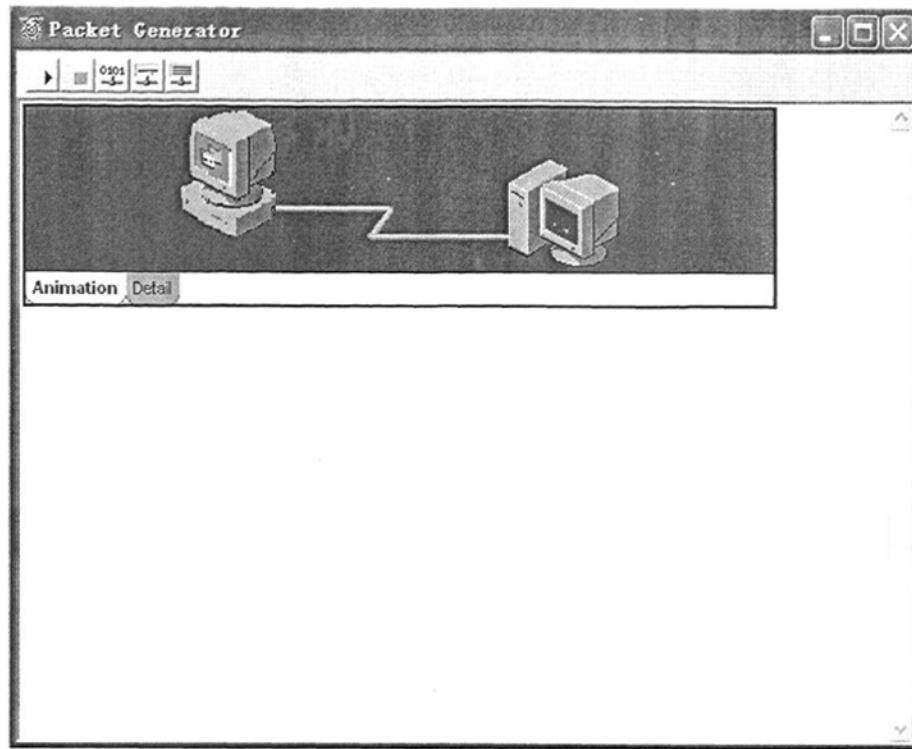


图 5-26

再点击图 5-26 所示标有 0101 的按钮，编辑数据，模拟 OSC 信道上发送的数据，所发数据如图 5-27：

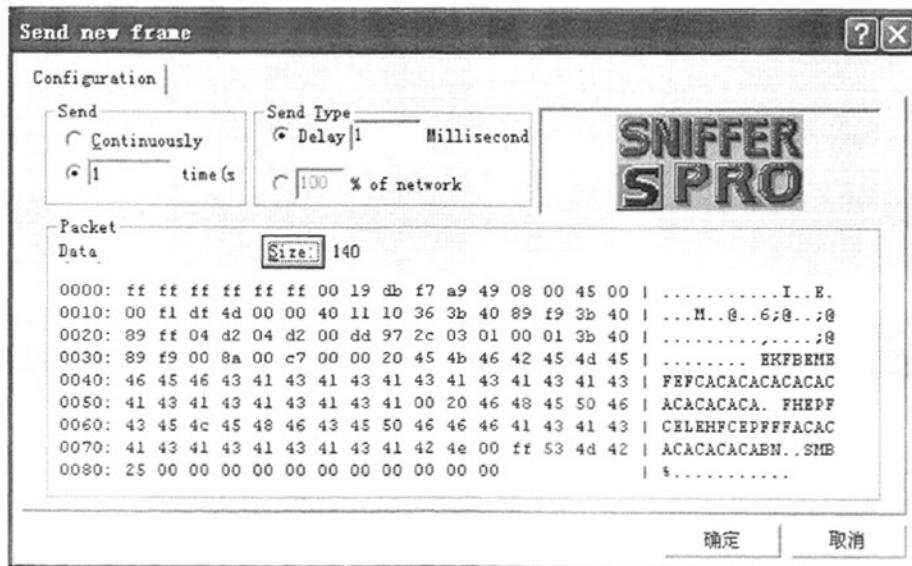


图 5-27

在 B 计算机上运行带有解析高层网管协议“xy”插件的 Wireshark，捕获的数据如图 5-28：

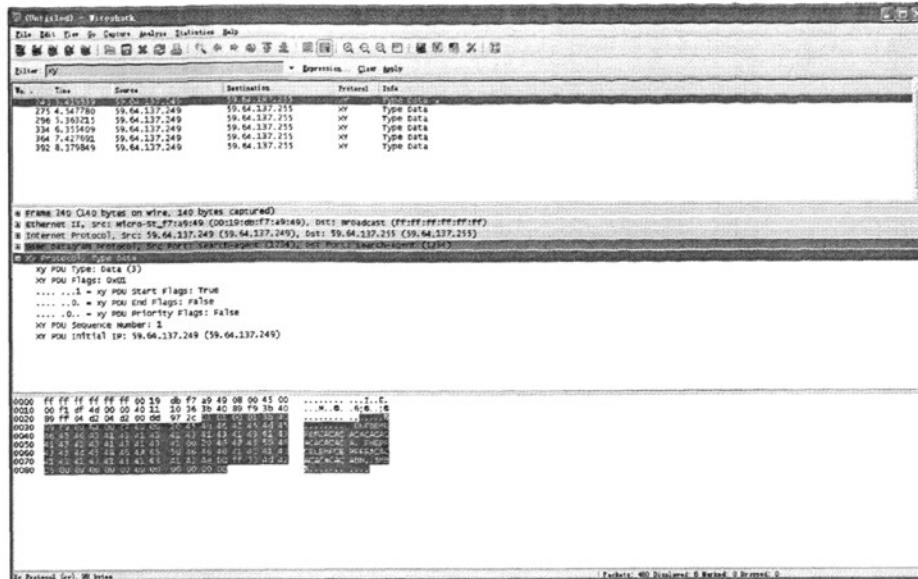


图 5-28

上图 Wireshark 捕获的数据中

数据包类型：0x03 表示数据；

标志位：0x01 表示开始包；

序列号：0x00 01 代表该数据序列为 1；

IP 地址：初始化 IP 地址是 59.64.137.249

这与我们猜测的高层网管协议吻合，验证了我编写的插件可以正确解析名为“xy”的高层网管协议。

### 5.3.5 进一步研究的问题

对底层的协议解析可以仿照这一思路进行解析，如果 Sniffer 发送的数据跟解扰后提取的 DCC 字节内容一样的话，就可以证明我们的理论分析预测是正确的。

实现这一理论预测的研究关键处还有俩点：1 从 DCC 字节内容中提取 Ethernet 数据包；2 进行文件格式的转换。

---

.dat 文件格式：OSC 数据捕获软件存下来的文件格式，文件头 28 字节，记录时间标签（年、月、日、时、分、秒、毫秒，各占 2 个字节），每帧帧界定字符 4 字节，可以随便指定。

.cap 文件格式：ethereal/wireshark 保存捕获文件的默认格式，其文件头是固定的 24 字节，然后每个包前是 16 个字节，16 字节的后 8 个字节分别用 4 字节重复表示了后续的包的长度，最后一个包完了以后直接结束文件，没有文件尾。

猜测并验证第二层封装以太网的网管协议，完成两种文件格式的转换是我们需要就一步研究的问题。

---

## 总 结

自适应 DWDM 是一种替换设备，最大程度地恢复网管通道，适应尽可能多厂家的 DWDM 设备。只需要准备一种类型的 DWDM 设备，替代不同厂家的 DWDM 设备，以不变应万变，是我们最终要实现的目标。能够实现真正意义上的互联互通设备是个有前瞻性、光通信迫切需求，且有一定难度的课题。这也就是该论文研究的重要意义。

本文阐述了解析网管协议的思路：通过在网管端抓包和在 OSC 信道上捕获数据，对比处理数据，模拟解析网管协议。本文提出了 OSC 数据捕获软件的需求分析、编写应用软件程序，进行反复测试和修改；设计实验测试方案，捕获并处理数据。在 OADM 系统中，处理后得到要解析的数据，协议栈是基于以太网的，包括 Ethernet-IP-UDP-SNMP；在 DWDM 系统中，由于 OSC 线路上可能经过扰码，不能识别同步字节，无法得到我们要解析的真正数据。但是基于常见的网管协议，猜测网管协议名为“xy”的协议，在 Wireshark 开源环境下开发插件，模拟解析高层网管协议。

离实际的成功解析还有一定距离，但是基于已有工作的进展，研究出了验证解析是否成功的方法：通过 OSC 信道数据捕获软件的发送功能，我们按照解析出来的网管协议在 OSC 信道上发送数据，如果自适应 DWDM 设备可以被所在的 DWDM 子网的网管系统识别，就能够证明我们解析出来的网管协议是正确的；如果自适应 DWDM 设备不能被所在的 DWDM 子网的网管系统识别，可以重新猜测，修改插件反复解析，直到成功解析出整个 OSC 协议栈。该方法也是实现自适应 DWDM 系统网管协议跟其他厂商的 DWDM 系统适配，实现互联互通的最好途径。

---

## 参考文献

- [1].顾畹仪，李国瑞编著.光纤通信系统.北京邮电大学出版社 1999.11
- [2].Rajiv Ramaswami, Kumar N.Sivarajan 编著 乐孜纯译.光网络（第2版）.机械工业出版社 2004.9
- [3].信息产业部电信管理局主编.电信设备进网管理.人民邮电出版社 2003
- [4].Total SNMP, Sean Harnedy 编著 胡谷雨等译.简单网络管理协议教程（第2版）.电子工业出版社 2000.8
- [5].韦乐平编著.光同步数字传送网.人民邮电出版社.通信工程丛书 1998.12
- [6].乐孜纯，张明，全必胜编著.光网络实用组网技术.西安电子科技大学出版社 2008
- [7].张海懿，张成良，张柏成起草.YD/T 1060—2000 光波分复用系统(WDM) 技术要求—— $32 \times 2.5\text{Gbitps}$  部分 2000.10
- [8].Physical/electrical Characteristics of Hierachical Digital Interfaces.IITU-T Recommendation G.703 November 2001
- [9].Synchronous Frame Structures Used at 1554, 6312, 2048, 8448 and 44736 kbit/s Hierarchical Levels.IITU-T Recommendation G.704 October 1998
- [10].黄照祥，雷蕾译.光网络-第三代传送系统.机械工业出版社 2003.4
- [11].王海龙，董智勇编著.Visual C++ 6.0 设计师之路.电子工业出版社 2005
- [12].任哲编著.MFC Windows 应用程序设计.清华大学出版社 2004
- [13].郭军编著.网络管理.北京邮电大学出版社 2001.9
- [14].李明江编著.SNMP 简单网络管理协议.电子工业出版社 2007
- [15].余建国，何建易，河清编著.OXC 及 OADM 网络节点的研究.光通信研究 1999 年第 1 期
- [16].韩群，迟荣华，吕可诚编著.OADM 及其应用的最新研究进展.激光与光电子学进展 2002 年 第 5 期
- [17].何建明编著.OXC/OADM 关键技术及其发展.中兴通信技术 2002 年第 4 期
- [18].王郁编著.波分复用 DWDM 网元管理系统的技术要求.电信网技术 2003 年 第 12 期
- [19].孙学军，张述军编著.DWDM 传输系统原理与测试.人民邮电出版社 2000
- [20].金明晔，张智江，陆斌编著.DWDM 技术原理与应用.电子工业出版社 2004。

---

## 致 谢

在校两年多的学习和科研工作不仅使我在专业知识上大有收获，同时也在考虑问题，分析问题、解决问题的思维和方式方法上受益匪浅。我的成长是与众多关心我、爱护我和帮助我的老师、兄长和同学们分不开的。

首先，我要在此感谢我的导师徐大雄院士，他广博的学识，严谨的治学态度，实事求是的科研精神，都给我留下了深刻的印象。在他的言传身教之下，我不仅学会了如何去做研究工作，更学会了怎样去做一个有价值的人。徐老师对我的教诲使我终生受益。谢谢您，徐老师！

然后，我要衷心地感谢李老师、张阳安和张明伦老师在学习、生活和工作中给予的热心帮助与指导！感谢张老师对我工作学习的关心、教导和督促！

感谢师兄张锦南博士和袁学光博士在科研项目中给予的耐心指导和无私的帮助，他们认真严谨的治学态度让我获益匪浅。感谢浮光宾、高翠娟等师兄、师姐几年来对我的关心和帮助，在他们的帮助下我有了更多的实践机会也学到了很多东西。

感谢耿良、庞琳娜、周黎明以及和我同宿舍的朋友们，作为同一届的同学，大家互相关心，互相帮助，让我度过了一段难忘的研究生生活。

我还要特别感谢我的家人，正是他们的支持和鼓励，我才能够全身心地专注于我的研究工作，在人生道路上不断前进！

最后感谢诸位评审老师，谢谢您们于百忙之中审阅我的论文。

---

## 攻读学位期间发表的学术论文目录

- [1].李海伟, 张明伦.基于硬件平台 DN600 对 E1 数据的存储与分析.中国科技论文在线 2008.7
- [2].李海伟, 徐大雄, 李玲, 张阳安, 张明伦.基于 DN600 对 WDM 网络中 OSC 信道的存储与分析.光通信技术 已收录待刊载。