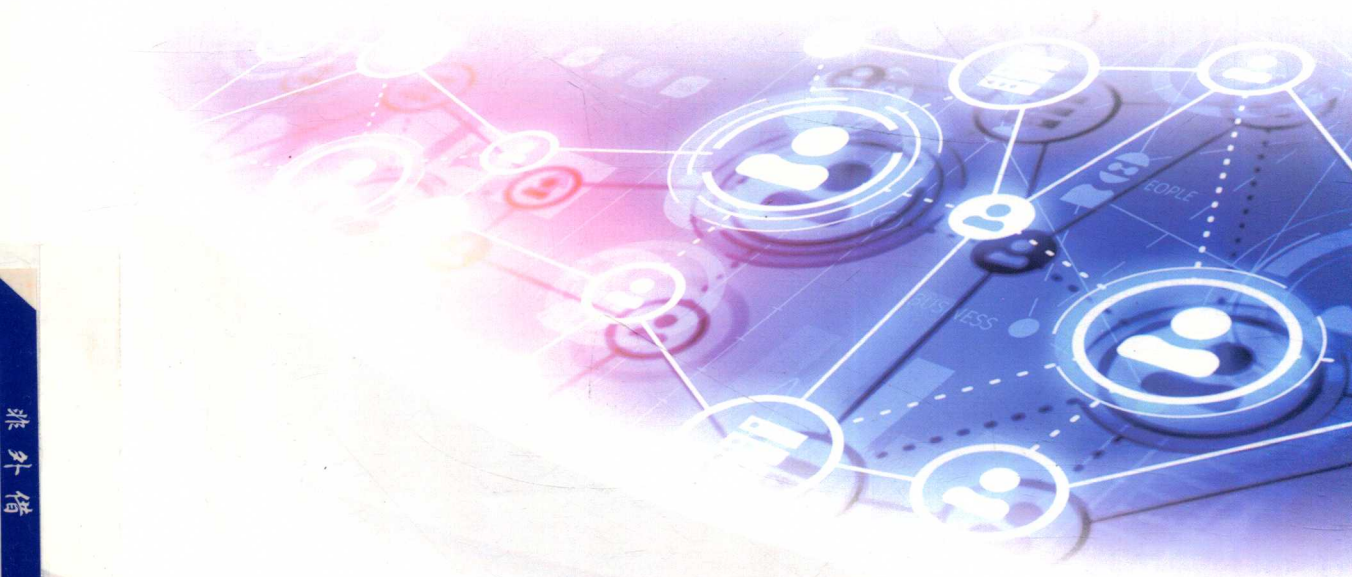
网络技术实验实训系列教材

Experimental Course of Software Defined Networking

软件定义网络实验教程

黄 韬 魏 亮 刘 江 杨 帆◎编著



非外借

中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

网络技术实验实训系列教材

Experimental Course of Software Defined Networking

软件定义网络实验教程

黄 韬 魏 亮 刘 江 杨 帆◎编著

人民邮电出版社
北京

图书在版编目（CIP）数据

软件定义网络实验教程 / 黄韬等编著. -- 北京 : 人民邮电出版社, 2018.9
网络技术实验实训系列教材
ISBN 978-7-115-48665-3

Ⅰ. ①软… Ⅱ. ①黄… Ⅲ. ①计算机网络－实验－高等学校－教材 Ⅳ. ①TP393-33

中国版本图书馆CIP数据核字(2018)第184297号

内 容 提 要

本书是针对 OpenLab 开放网络实验平台中的 SDN 课程实验编写的实验指导书，涵盖了工具使用、协议分析、控制器配置、应用开发和综合项目等多个方面的实验项目，共计 46 个实验，内容由浅入深，文字指导清晰，参考图例丰富。

本书兼顾高校 SDN 课程实验指导与学生自主学习能力提高的需要，配合 SDN 网络理论课程教学使用，促进学生对理论的理解和实践能力的提升，本书对 OpenLab 开放网络实验平台中 SDN 课程实训的教学效果有很大的帮助。

本书适合有一定网络基础知识的读者以及相关领域从业人员阅读，也适合本科、高职、中职等院校相关专业的师生作为教学参考书。

◆ 编　著　黄　韬　魏　亮　刘　江　杨　帆
责任编辑　邢建春
责任印制　彭志环

◆ 人民邮电出版社出版发行　北京市丰台区成寿寺路 11 号
邮编　100164　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京市艺辉印刷有限公司印刷

◆ 开本：787×1092　1/16
印张：23.5　2018 年 9 月第 1 版
字数：572 千字　2018 年 9 月北京第 1 次印刷

定价：128.00 元

读者服务热线：(010)81055488　印装质量热线：(010)81055316
反盗版热线：(010)81055315

前 言

随着我国现代信息技术的蓬勃发展，信息化建设模式发生根本性转变，一场以云计算、大数据、移动应用、人工智能等技术为核心的“新 IT”浪潮风起云涌，信息化应用进入“新常态”。然而，随着网络复杂度的提升，以及“互联网+”、三网融合等新兴产业对互联网在可扩展性、安全性、可控可管等方面需求的激增，现有网络构架已不能更好地支撑未来网络的发展。

SDN 是一种新型的网络创新架构与技术，其倡导的控制与转发分离，网络功能虚拟化使网络更开放、可编程、能灵活支撑上层业务，得到了业界广泛的认同。未来在 5G、物联网时代，SDN/NFV 必将是主流技术，促进 ICT 融合与发展。目前，越来越多的高校紧跟技术发展方向，开设了 SDN/NFV 的相关课程。然而，传统的网络实验室由于硬件设备功能固化，面对新兴技术的变革，高校需要投入大量的人力、物力重新建设实验室，以满足新课程、新技术的实验实训需求。因此，开放、可扩展的虚拟仿真实验实训平台得到了越来越多高校的青睐，并得到了广泛地运用。

OpenLab 开放网络实验平台是江苏省未来网络创新研究院自主研发的一款创新实验教育平台。该平台采用 SDN 与虚拟化等技术解决了传统实验室在时间、空间、实验内容等方面的限制，可以快速构建复杂度高、隔离性强的各种实验环境。通过快速构建相互隔离的实验网络，能够让学生快速掌握计算机网络、SDN/NFV 等相关原理，提高实践能力，培养信息网络领域创新型人才。目前，OpenLab 开放网络实验平台已经在国内多个高校进行了部署，并取得了良好的教学效果。

本教材针对 OpenLab 开放网络实验平台中的 SDN 课程实验进行编写，总结了在 SDN 实践实训中常用的相关实验内容与前沿技术，以真实项目案例进行参考，涵盖工具使用、协议分析、控制器配置、应用开发、综合项目等多个方面的实验内容，共计 46 个。本教材以图文并茂的形式对实验进行步骤分析与效果演示，既能满足教师对于实验课程的教学与实训需求，又能促进学生对于 SDN 课程相关实验的学习与理解。

本教材兼顾高校 SDN 课程实验指导与学生自主学习能力提高的需要，由浅入深地设计了实验课程指导，内容文字清晰，参考图例丰富。配合高校的 SDN 理论课程进行教学，能够良好地融入理论教学的各个实践环节。高校教师在教学过程中，可以根据教学需求，选择本教材中的相关实验进行针对性的教学和扩展，以获得更好的教学效果。

本书参与撰写和审校的人员包括：北京邮电大学的谢人超、张娇，江苏省未来网络创新研究院的崔丽娴、付志鹏、黄娟、李金葵、李蓉、梁田、周洪利、方辉、陈昌盛等。在此对大家表示衷心感谢。

最后，感谢人民邮电出版社的大力支持和高效工作，使本书能尽早与读者见面。由于编者水平有限，编写时间紧迫，书中难免会有不足与疏漏，恳请专家与广大读者不吝指正。

OpenLab 开放网络实验平台体验地址是 https://www.sdnlab.com/experimental-platform/，相关实验 QQ 交流群是 424219702，本书联系邮箱地址是 service@sdnlab.com。

编　者

2018 年 7 月

试读结束，需要全本PDF请购买 www.ertongbook.com

目 录

项目 1　学习使用开源工具

任务一　抓包分析工具 Wireshark 使用介绍

任务目的

1．熟悉并掌握 Wireshark 的基本使用方法，利用 Wireshark 抓取网络数据包。

2．掌握 Wireshark 过滤器的使用方法，从而提高分析数据包的效率。

任务环境

抓包分析工具 Wireshark 使用介绍实验的拓扑如图 1-1 所示。

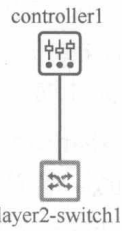


图 1-1　实验拓扑

实验环境镜像配置说明如表 1-1 所示。

表 1-1　实验环境镜像配置说明

设备名称	软件环境	硬件环境
控制器	Ubuntu 14.04 桌面版 Floodlight 1.0	CPU：1 核 内存：2 GB 磁盘：20 GB
交换机	Ubuntu 14.04 命令行版 Open vSwitch 2.3.1	CPU：1 核 内存：2 GB 磁盘：20 GB

注：系统默认的账户为 root/root@openlab、openlab/user@openlab。

任务内容

1．从 Wireshark 的界面入手，深入了解 Wireshark 的功能。

2．使用 Wireshark 捕获数据包，利用捕获过滤器和显示过滤器快速筛选出 OpenFlow 数据包，并学会简单分析。

实验原理

Ethereal 是一个免费开源的网络数据包分析软件，用于捕获网络数据包并尽可能显示出最为详细的网络数据包数据。1997 年年底，Gerald Combs 需要一个能够追踪网络流量的工具软件作为其工作上的辅助，于是他开始撰写 Ethereal 软件。随后越来越多的人参与 Ethereal 的开发工作，2006 年 6 月，Ethereal 更名为 Wireshark。Wireshark 的特点如下。

1．支持数百种协议。

2．支持实时捕捉数据包和进行离线分析。

3．支持标准三窗格分组显示。

4．支持 Windows、Linux、macOS、Solaris、FreeBSD 和 NetBSD 等多平台。

5．支持捕获的网络数据通过 GUI 或 TTY 模式的 TShark 实用程序浏览。

6．业界最强大的显示过滤工具。

7．支持丰富的 VoIP 分析。

8．支持许多不同的捕获包格式。

9．用 gzip 压缩方式对实时捕获的数据进行压缩和加密。

10．支持对多种协议的解密，包括 IPsec、ISAKMP、Kerberos、SNMPv3、SSL/TLS、WEP 和 WPA/WPA2。

11．着色规则可以应用于数据包列表，以进行快速、直观的分析。

12．输出可以导出为 XML、PostScript、CSV 或纯文本格式。

Wireshark 的实现基于分解器（Dissector），网络上每一层的协议都有对应的分解器，分解器的作用是把每一层的信息分解，显示出首部字段，把有效载荷字段传递给上一层的分解器，以达到逐层分解的目的。使用 Wireshark 抓取网络数据包的流程如下。

1．确定在哪个目录下启动 Wireshark。若目录选择错误，将无法抓取到预期的数据包。

2．选择捕获接口。一般选择连接到 Internet 网络的接口，才能捕获到与网络相关的数据。

3．使用捕获过滤器。通过设置捕获过滤器，可以避免产生过大的捕获文件。用户在分析数据时，不会受其他数据干扰，还可以为用户节约大量的时间。

4．使用显示过滤器。为了使过滤的数据包更细致，可以使用显示过滤器进行过滤。

5．使用着色规则。通常使用显示过滤器过滤后的数据，都是有用的数据包。如果想更加突出地显示某个会话，可以使用着色规则高亮显示。

6．构建图表。如果用户想要更明显地看出一个网络中数据的变化情况，使用图表的形式可以很方便地展现数据分布情况。

7．重组数据。由于传输的文件较大，信息分布在多个数据包中。为了能够查看到整个图片或文件，需要使用重组数据的方法来实现。

实验步骤

一　实验环境检查

登录交换机，执行 **ovs-vsctl show** 命令检查网络连通性，当控制器连接状态显示为“true”时，表示交换机连接上控制器，如下所示。

```
root@openlab:~# ovs-vsctl show
3f2d706b-8776-440d-b4e3-da0ef103d120
    Bridge br-sw
        Controller "tcp:20.0.1.3:6633"
            is_connected: true
        fail_mode: secure
        Port "eth7"
            Interface "eth7"
        Port "eth4"
            Interface "eth4"
        Port "eth2"
            Interface "eth2"
        Port br-sw
            Interface br-sw
                type: internal
        Port "eth8"
            Interface "eth8"
        Port "eth5"
            Interface "eth5"
        Port "eth1"
            Interface "eth1"
        Port "eth3"
            Interface "eth3"
        Port "eth6"
            Interface "eth6"
```

二　Wireshark 抓包设置

步骤 1　选择控制器，打开终端。执行 **sudo wireshark** 命令，启动 Wireshark，如图 1-2 所示。

Wireshark 界面主要包括主工具栏、过滤工具栏和状态栏。

（1）主工具栏中左起前 5 个按钮与抓包操作有关，其余按钮分别与文件操作、数据包选择操作、配色及自动滚屏设置等有关。

（2）过滤工具栏用于过滤数据包，在输入框中手动输入过滤表达式或单击“Expression”按钮根据预定义的模板构造过滤表达式。

（3）状态栏主要显示抓包文件名及其所在目录。抓包文件中包含数据包数量、Wireshark 实际显示出的数据包数量以及人为打上标记的数据包数量。

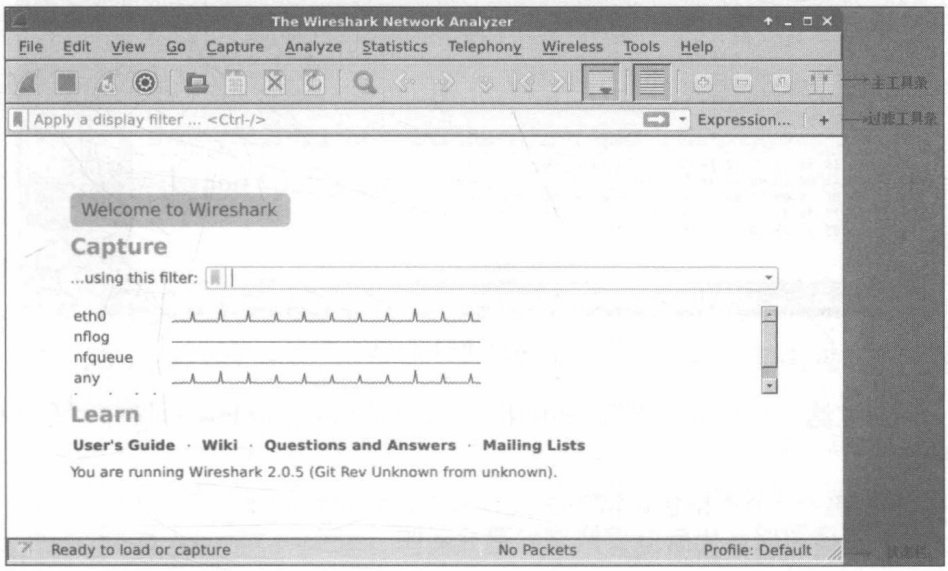


图 1-2　Wireshark 开始界面

试读结束，需要全本PDF请购买 www.ertongbook.com

步骤 2 选择抓包网卡，单击“Capture> Options”，选择网卡 eth0，界面如图 1-3 所示。

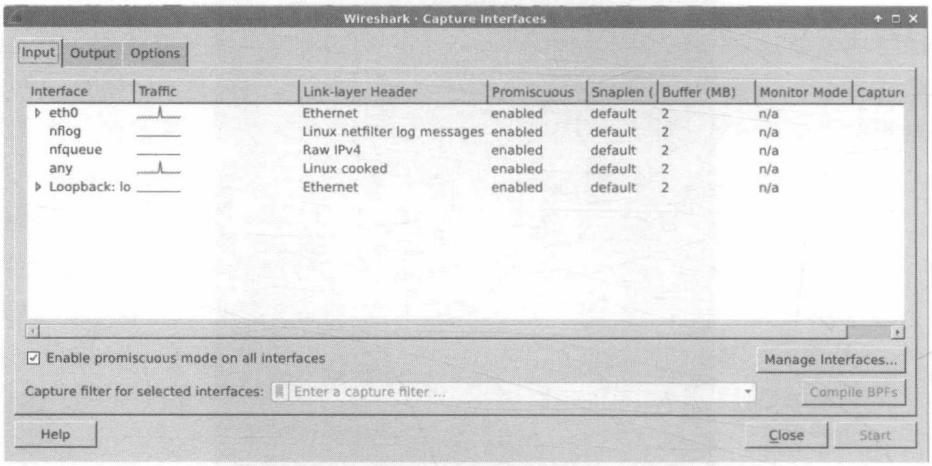


图 1-3 Wireshark 抓包页面

三 捕获 OpenFlow 数据包

步骤 1 单击“开始”按钮开始抓包。

数据包列表的面板中会滚动显示所抓取的数据包，包括数据包的编号、时间戳、源地址、目标地址、协议、长度以及相关信息，如图 1-4 所示。

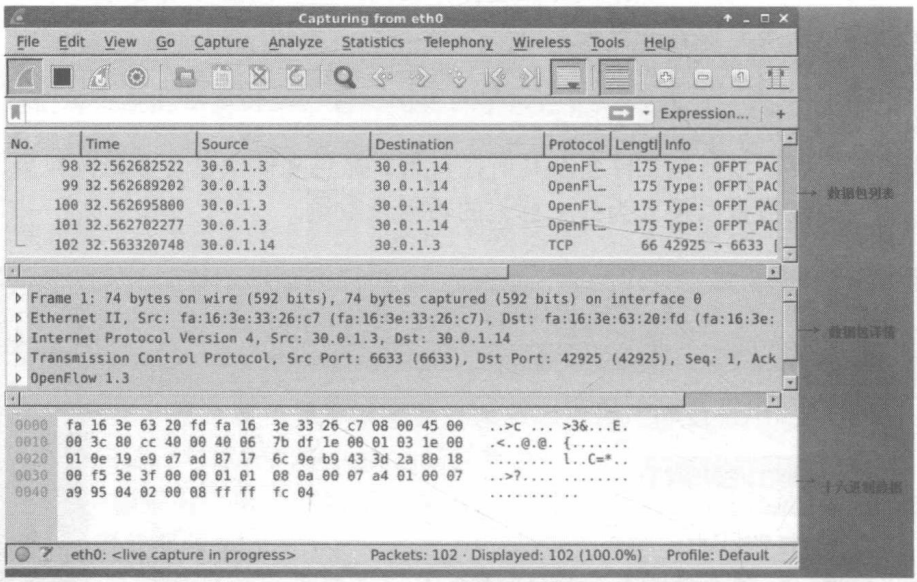


图 1-4 Wireshark 抓包信息

步骤 2 在过滤工具栏中输入“openflow_v4”，过滤出 OpenFlow 协议并查看数据包详情，如图 1-5 所示。

由图 1-5 可知，一个数据包详情中主要包含 5 层，依次如下。

（1）Frame：物理层数据概况。

（2）Ethernet II：数据链路层以太网帧头部信息。

（3）Internet Protocol Version 4：互联网 IP 包头信息。

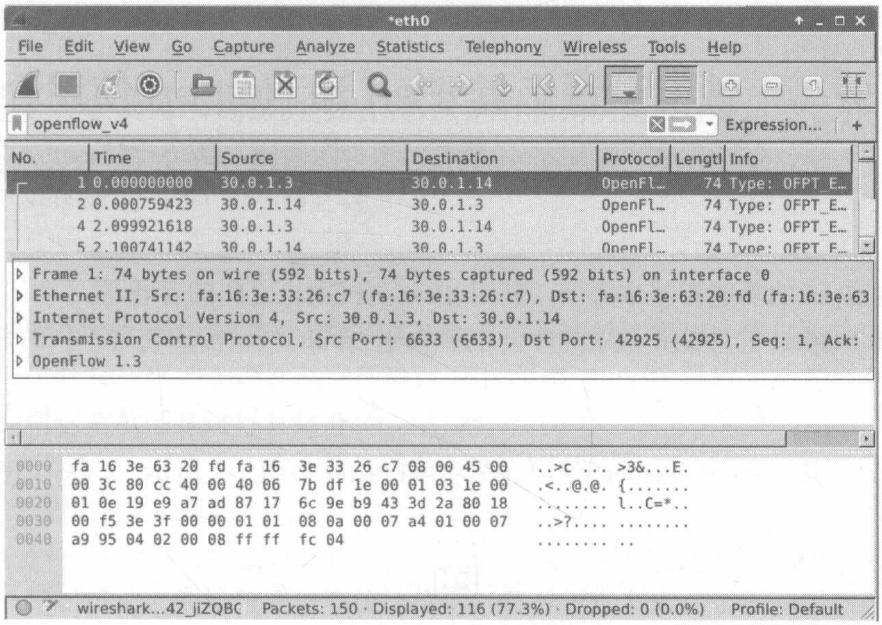


图 1-5　过滤 OpenFlow 协议数据包

（4）Transmission Control Protocol：传输层数据段头部信息，此处是 TCP。

（5）OpenFlow：位于 TCP 之上，负责解析应用层。

步骤 3　在数据包详情中查看 OpenFlow 包的具体内容。OpenFlow 包中主要字段包括协议版本、消息类型、消息长度、事件 ID 等，如图 1-6 所示。

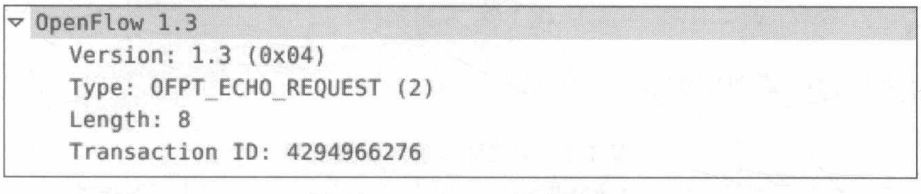


图 1-6　OpenFlow 消息字段

步骤 4　保存数据。

（1）选择“File > Save”或“File > Save As”将所有数据包保存到一个文件中。

（2）选择“File > Export Specified Packets”，保存已抓取数据包中一部分。导出数据的具体方法如下。

- 选择 All packets，则保存所有已抓取的数据包。
- 选择 Selected packets only，则保存当前鼠标选中的数据包。
- 选择 Marked packets only，则保存当前标记的数据包。标记数据包的方法是在数据包列表中选中目标数据包，然后右击选择“mark packet toggle”。
- 选择 First to last marked，则将两个标记数据包之间所有的数据包导出。
- 选择 Range，并输入数据包的编号范围，则将编号连续的数据包导出。
- 选择 Remove ignored packets，则放弃部分数据包并导出其他所有数据包。放弃数据包的方法是，在数据包列表中选中数据包，右击选择“ignore packet toggle”。

任务二　性能测试工具 iPerf 和 Netperf 使用介绍

任务目的

1．掌握 iPerf 的功能和基本命令，利用 iPerf 测试主机之间的吞吐量。

2．掌握 Netperf 的功能和基本命令，能够根据不同的测试需求使用不同的命令参数。

任务环境

性能测试工具 iPerf 和 Netperf 使用介绍实验的拓扑如图 1-7 所示。

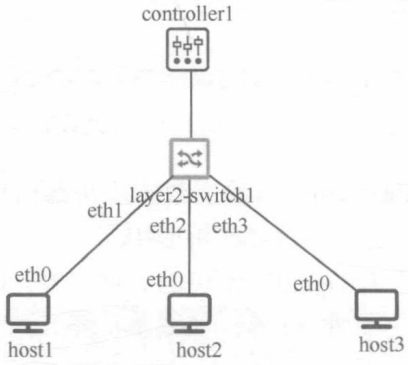


图 1-7　实验拓扑

实验环境镜像配置说明如表 1-2 所示。

表 1-2　实验环境镜像配置说明

设备名称	软件环境	硬件环境
控制器	Ubuntu 14.04 桌面版 Floodlight 1.0	CPU：1 核 内存：2 GB 磁盘：20 GB
交换机	Ubuntu 14.04 命令行版 Open vSwitch 2.3.1	CPU：1 核 内存：2 GB 磁盘：20 GB
主机 1	Ubuntu 14.04 命令行版	CPU：1 核 内存：2 GB 磁盘：20 GB
主机 2	Ubuntu 14.04 命令行版	CPU：1 核 内存：2 GB 磁盘：20 GB
主机 3	Ubuntu 14.04 命令行版	CPU：1 核 内存：2 GB 磁盘：20 GB

注：系统默认的账户为 root/root@openlab、openlab/user@openlab。

任务内容

1．使用 iPerf 测试 SDN 网络的性能，熟悉 iPerf 常用的测试命令。
2．使用 Netperf 测试 SDN 网络的性能，熟悉 Netperf 常用的测试命令。
3．总结 Netperf 与 iPerf 的不同之处。

实验原理

在构建或管理一个网络系统时，除了关心网络的可用性，也要注意其整体的性能，这时就要用合适的工具对网络的各项性能进行测试。iPerf 和 Netperf 是用于衡量网络性能的测试工具，本实验对这两款工具进行具体介绍。

一　iPerf 工具

iPerf 是一种网络性能测试工具，可以运行于 Linux、BSD、UNIX 及 Windows 等操作系统。iPerf 具有多种参数和特性，支持协议、定时、缓冲区等参数的配置调整，能够测试 TCP/UDP 的最大带宽、延迟抖动、数据包丢失等统计信息，可以根据需求采用不同的参数从而达到不同的测试目的。iPerf 基于 Server/Client 的工作模式，客户端向服务端发送一定数量的数据，服务端统计并计算带宽、延时抖动等信息。客户端将数据全部发送后，服务端会回复一个数据包给客户端，将测试数据反馈给客户端。如果网络较为拥塞或误码率较高，客户端无法收到服务端回复的数据包，则只能显示本地记录的部分测试结果，所以服务端和客户端的测试结果可能有所不同。iPerf 工作模式如图 1-8 所示。

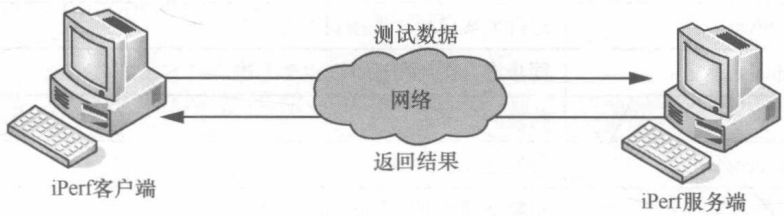


图 1-8　iPerf 工作模式

在命令提示符中输入 iperf 命令即可运行 iPerf。iPerf 命令语法格式和参数解释如下。

1．命令格式。

iperf [-s|-c host] [options]

2．参数配置如表 1-3~表 1-5 所示。

表 1-3　客户端/服务器端通用参数

参数	作用
-f/--format	[kmKM]分别表示以 kbit、Mbit、kB 和 MB 显示报告，默认以 Mbit 为单位。
-i/--interval	以秒为单位显示报告间隔
-l/--len	读写的缓冲区大小，默认 8 kB
-m/--print_mss	打印最大的 TCP 数据段大小（MTU - TCP/IP header）
-o/--output	将报告和错误信息输出到指定文件
-p/--port	指定服务器端使用的端口或客户端所连接的端口

（续表）

参数	作用
-u/--udp	使用 UDP
-w/--window	指定 TCP 窗口（socket 缓冲区）大小，默认是 8 kB
-B/--bind	绑定一个主机地址或接口
-C/--compatibility	兼容旧版本（当 Server 端和 Client 端版本不一样时使用）
-M/--mss	设置 TCP 最大数据段大小（MTU - 40 byte）
-N/--nodelay	设置无延迟 TCP，禁用 Nagle's Algorithm
-V/--IPv6Version	设置传输 IPv6 数据包

表 1-4　服务器端参数

参数	作用
-s/--server	服务器模式下运行
-U/-- single_udp	单线程 UDP 模式下运行
-D/-- daemon	以守护进程模式运行服务器

表 1-5　客户端参数

参数	作用
-b/-- bandwidth	对于 UDP，使用 bit/s（默认 1 Mbit/s，即-u）传送带宽
-c/-- client	运行为客户端，连接到“主机”
-d/-- dualtest	同步进行双向测试
-n/-- num	传输的字节量
-r/-- tradeoff	分别进行双向测试
-t/-- time	传输持续时间（默认 10 s）
-F/-- fileinput	从文件中读取要传输的数据
-I/-- stdin	从标准输入（stdin）中读取要传输的数据
-L/-- listenport	双向测试接收端口
-P/-- parallel	并行客户线程数量
-T/-- ttl	多点传送的生存时间（默认 1）

二　Netperf 工具

Netperf 是一种网络性能测量工具，主要用于测试 TCP 或 UDP 和 Berkeley 套接字接口的批量数据传输（Bulk Data Transfer）和请求/应答（Request/Reponse）性能。Netperf 工具以 Client/Server 方式工作，服务端是 netServer，用来侦听来自客户端的连接，客户端是 Netperf，用来向服务发起网络测试。在客户端与服务端之间，首先建立一个控制连接，传递有关测试配置的信息，以及测试的结果。在控制连接建立并传递了测试配置信息以后，客户端与服务端之间再建立一个测试连接，用于来回传递特殊的流量，以测试网络的性能。Netperf 工作模式如图 1-9 所示。

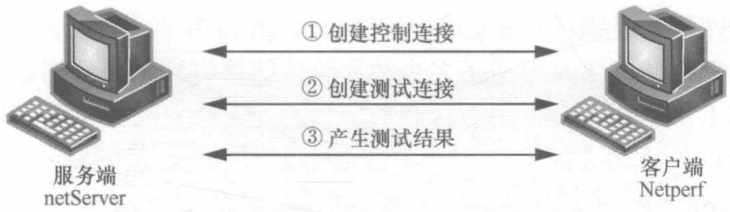


图 1-9　Netperf 工作模式

Netperf 通过命令行参数来控制测试的类型和具体的测试选项，根据作用范围的不同，nNetperf 的命令行参数可以分为全局命令行参数和测试相关的局部参数两大类，两者之间使用--分隔。Netperf 命令语法格式和参数解释如下。

1．命令格式。

netperf [global options] --[test-specific options]

2．[global options]可选参数，可选的参数如表 1-6 所示。

表 1-6　局命令行参数

参数	作用
-H host	指定远端运行 netServer 的 Server IP 地址
-l testlen	指定测试的时间长度（s）
-t testname	指定进行的测试类型（TCP_STREAM，UDP_STREAM，TCP_RR，TCP_CRR，UDP_RR）

3．[test-specific options]可选参数，可选的参数如表 1-7 所示。

表 1-7　测试相关的局部参数

参数	作用
-s size	设置本地系统的 socket 发送与接收缓冲大小
-S size	设置远端系统的 socket 发送与接收缓冲大小
-m size	设置本地系统发送测试分组的大小
-M size	设置远端系统接收测试分组的大小
-D	对本地与远端系统的 socket 设置 TCP_NODELAY 选项
-r req,resp	设置 request 和 reponse 分组的大小

三　网络性能指标

学习网络性能测试工具不仅要掌握基本使用方法，还要学会分析测试结果，常见的网络性能参数如下。

（1）网络吞吐量：单位时间内通过某个网络（信道或接口）的数据量，吞吐量受网络的带宽或者网络的额定速率限制，单位通常表示为 bit/s。

（2）网络延时：一个数据包从用户的计算机发送到网站服务器，然后立即从网站服务器返回用户计算机的来回时间。影响网络延时的主要因素是路由的跳数和网络的流量。交换机延时（Latency）是指从交换机接收到数据包到开始向目的端口复制数据包之间的时间间隔。有许多因素会影响交换机延时大小，如转发技术等。

（3）抖动：用于描述包在网络中传输延时的变化，抖动越小，说明网络质量越稳定、越

试读结束，需要全本PDF请购买 www.ertongbook.com