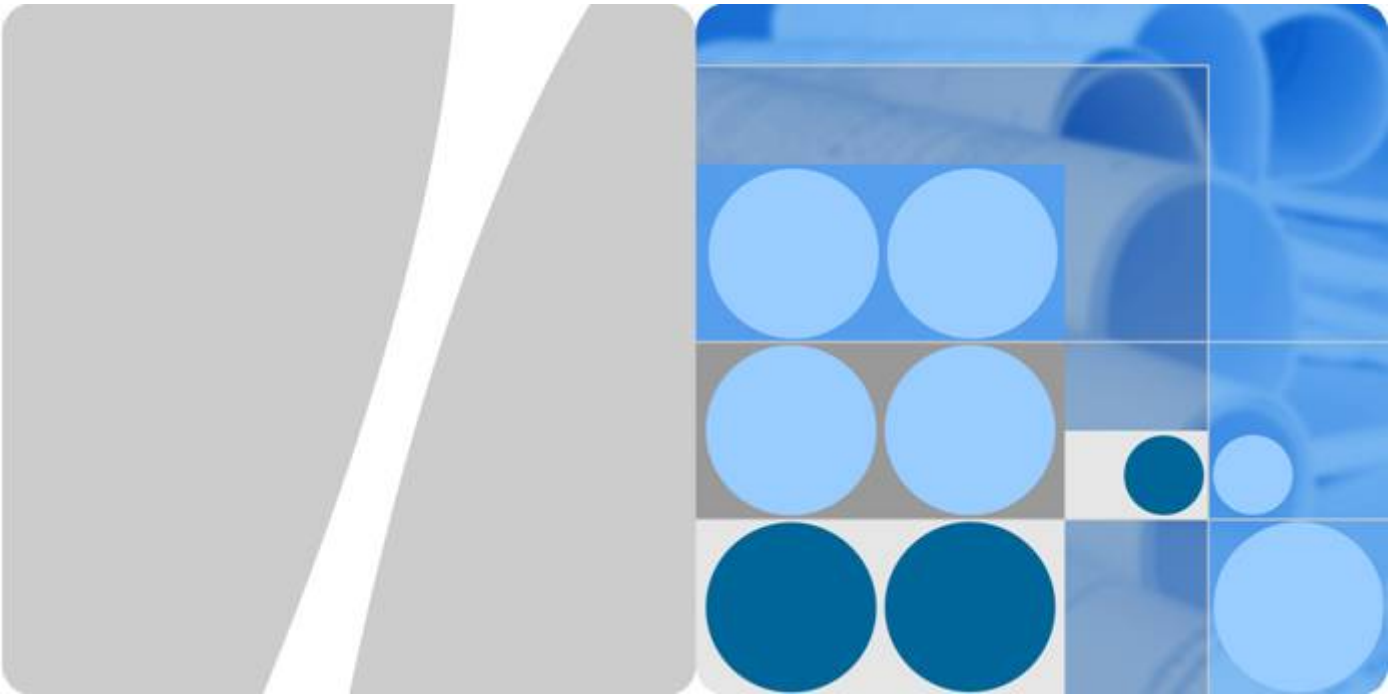


Huawei AR2200-S 系列企业路由器

V200R001C01

特性描述-基础配置

文档版本　01

发布日期　2012-01-06

华为技术有限公司

HUAWEI

商标声明


和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编：518129

网址： http://www.huawei.com

客户服务邮箱： support@huawei.com

客户服务电话： 4008302118

前 言

读者对象

本文档针对 AR2200-S 的基础特性，从简介、原理描述和应用三个方面介绍基础特性以及协议。包括，FTP、TFTP、Telnet、SSH 等。

本文档与其它类型手册相结合，便于读者深入掌握特性的实现原理。

本文档主要适用于以下工程师：

- 网络规划工程师
- 调测工程师
- 数据配置工程师
- 系统维护工程师

符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
危险	以本标志开始的文本表示有高度潜在危险，如果不能避免，会导致人员死亡或严重伤害。
警告	以本标志开始的文本表示有中度或低度潜在危险，如果不能避免，可能导致人员轻微或中等伤害。
注意	以本标志开始的文本表示有潜在风险，如果忽视这些文本，可能导致设备损坏、数据丢失、设备性能降低或不可预知的结果。
窍门	以本标志开始的文本能帮助您解决某个问题或节省您的时间。
说明	以本标志开始的文本是正文的附加信息，是对正文的强调和补充。

命令行格式约定

格式	意义
粗体	命令行关键字（命令中保持不变、必须照输的部分）采用**加粗**字体表示。
斜体	命令行参数（命令中必须由实际值进行替代的部分）采用*斜体*表示。
[ ]	表示用“[ ]”括起来的部分在命令配置时是可选的。
{ x | y | ... }	表示从两个或多个选项中选取一个。
[x | y | ...]	表示从两个或多个选项中选取一个或者不选。
{ x | y | ... }*	表示从两个或多个选项中选取多个，最少选取一个，最多选取所有选项。
[x | y | ...]*	表示从两个或多个选项中选取多个或者不选。
&<1-n>	表示符号&前面的参数可以重复 1 ～ n 次。
#	由“#”开始的行表示为注释行。

修订记录

修改记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

文档版本 01 (2012-01-06)

第一次正式发布。

目 录

1 基础配置

关于本章

1.1 介绍

定义

在配置管理中，终端服务特性提供了设备配置的管理接入接口和交互界面，为用户提供操作场所。

主要包括：

- Console 口登录
- Telnet Server/Client
- SSH 登录，支持 Password、RSA 验证
- 支持定制 User-interface，提供对登录用户多种方式的认证和授权功能

文件传输特性可以提供系统文件、配置文件的传输控制和文件系统的远程简单管理。

主要包括：

- FTP Server/Client
- TFTP Client
- 基于 SSH 协议的文件传输 SFTP Client/Server

本文档将按照协议分类分别介绍各协议特性的原理描述。

主要包括：

- FTP 协议
- TFTP 协议
- Telnet 协议
- SSH 协议
- 用户管理
- 虚拟文件系统
- 夏令时
- 定时重启

目的

终端服务特性提供了设备配置的管理接入接口和交互界面，为用户提供了操作场所。文件传输特性提供了系统文件、配置文件的传输控制和文件系统的远程简单管理功能。

1.2 参考标准和协议

本特性的参考资料清单如下：

文档	描述	备注
RFC775	Directory oriented FTP commands	-
RFC959	File Transfer Protocol	-

文档	描述	备注
RFC1635	How to Use Anonymous FTP	-
RFC1350	The TFTP Protocol (Revision 2)	-
RFC698	Telnet Extended ASCII Option	-
RFC775	Directory oriented FTP commands	-
RFC854	Telnet Protocol Specification	-
RFC855	Telnet Option Specification	-
RFC930	Telnet Terminal Type Option	-
RFC1091	Telnet Terminal-Type Option	-
RFC2119	Key words for use in RFCs to Indicate Requirement Levels	-
RFC4250	The Secure Shell (SSH) Protocol Assigned Numbers	-
RFC4251	The Secure Shell (SSH) Protocol Architecture	-
RFC4252	The Secure Shell (SSH) Authentication Protocol	-
RFC4253	The Secure Shell (SSH) Transport Layer Protocol	不支持压缩、不支持 ssh-dss 公钥格式
RFC4254	The Secure Shell (SSH) Connection Protocol	以下报文或功能不支持：X11 转发功能 Env 通道请求报文、xon-xoff 通道请求报文、Signal 通道请求报文、exit-status 通道请求报文、exit-signal 通道请求报文、端口转发功能
RFC4344	The Secure Shell (SSH) Transport Layer Encryption Modes	-
RFC4345	Improved Arcfour Modes for the Secure Shell (SSH) Transport Layer	-
draft-ietf-secsh-publickey-subsystem-01	Authentication Mechanism that Is Based on Public Keys	-
RFC4741	NETCONF Configuration Protocol	-
RFC4743	Using NETCONF over the Simple Object Access Protocol(SOAP)	-
RFC2616	Hypertext Transfer Protocol HTTP/1.1	-

1.3 可获得性

涉及网元

无需其他网元的配合。

License 支持

无需获得 License 许可，即可获得该特性的服务。

版本支持

表 1-1 基础特性的版本支持

产品	支持版本
AR2200-S	V200R001C01

1.4 原理描述

1.4.1 FTP 协议

FTP（File Transfer Protocol）在 TCP/IP 协议族中属于应用层协议，是文件传输的 Internet 标准。主要功能是向用户提供本地和远程主机之间的文件传输，尤其在进行版本升级、日志下载和配置保存等业务操作时，广泛地使用 FTP 功能。FTP 协议基于相应的文件系统实现。

FTP 采用 C/S（Client/Server）结构，如图 1-1 所示。

图 1-1 FTP 采用的 Client/Server 结构

在 AR2200-S 系统中提供的 FTP 功能包括：

- FTP Server：运行于路由器上的 FTP 服务。提供远程客户端访问和操作的功能，用户可以通过 FTP 客户端程序登录到路由器上，访问路由器上的文件。
- FTP Client：FTP 的客户端。提供本地路由器对远程服务器的文件进行操作的命令。用户在 PC 上通过终端程序或 Telnet 程序与路由器建立连接后，可以输入 FTP

命令建立与远程 FTP Server 的连接并访问远程主机上的文件，对远程主机上的文件进行操作。

FTP 除了完成文件传输基本功能外，同时还提供交互式的访问，允许用户指定存储数据的类型和格式，并允许文件具有存取权限（如访问文件的用户必须经过授权）。

FTP 实现主机间文件的传输，并提供常用的文件操作命令，供用户进行文件系统的简单管理。客户可以利用路由器外部的 FTP 客户端程序与路由器进行文件的上传、下载和目录访问等操作；还可以使用路由器内部的 FTP 客户端程序与其他路由器或其他设备的 FTP 服务器端的程序进行文件传输。

FTP 协议的基本概念

在介绍 FTP 协议之前，先介绍文件传输中的几个基本概念：

- 文件类型
  - ASCII 码文件类型

    即文本文件以 ASCII 码格式在数据连接中传输。发送方将本地文本文件转换成 ASCII 码格式，接收方收到该文件后再将 ASCII 码格式还原成本地文本文件格式。
  - EBCDIC 文件类型

    要求文件传输的两端都是 EBCDIC 系统。
  - 二进制文件类型

    即传输的是一个连续的比特流。通常用于传输图像文件和程序文件。
  - 本地文件类型

    用于在具有不同文件系统的主机之间传输二进制文件。每一字节的比特流由发送方规定。

  说明

  FTP 支持 ASCII 码、二进制文件类型。二者的区别是:

  - ASCII 传输使用 ASCII 字符，把回车键和换行符分开
  - 二进制不用转换格式就可传字符

  FTP 传输模式由客户端进行选择，系统默认 ASCII 方式。客户端可使用模式切换命令进行切换（ASCII 和 Binary）。
- 文件结构
  - 字节流

    也称为文件结构。文件被认为是一个连续的字节流。
  - 记录结构

    该结构只用于文本文件（ASCII 或 EBCDIC）。
  - 页结构

    发送单位为页，每页发送时都带有页号，以便接收方可以随机存储页。

  说明

  FTP 支持字节流和记录结构。
- 传输方式
  - 流方式

    文件以字节流的形式传输。对于文件结构，发送方在文件结尾提示关闭数据连接；对于记录结构，有专用的两字节序列码标识记录结束和文件结束。

- 块方式

  文件以一系列块来传输，每块前面都带有一个或多个首部字节。

- 压缩方式

  即压缩连续出现的相同字节。

说明

AR2200-S 仅支持流方式。

- PORT 命令

  用于打开端口的命令。格式为 PORT *a,b,c,d,e,f*。其中 *a,b,c,d* 为点分十进制形式的 IP 地址；*e,f* 表示端口号，为 2 个 10 进制数字，用以标识 *e*×256 ＋ *f* 运算出来的端口号。例如：

  ftp> debug

  Debugging On .

  ftp> ls

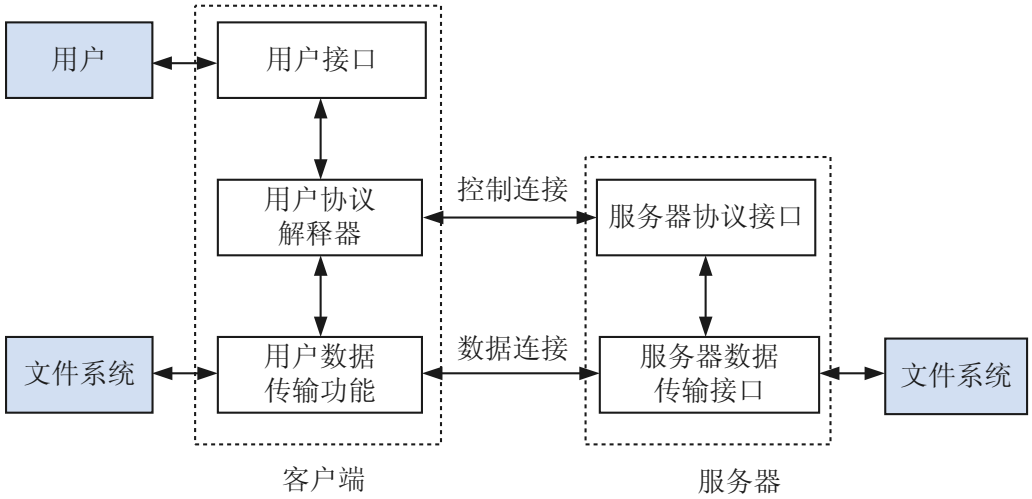
  ---> PORT 10,164,9,96,5,28

  这里 10.164.9.96 就是 IP 地址，5,28 通过公式计算：5*256+28=1308 就是选定的端口号。

FTP 中的连接

如图 1-2 显示了 FTP 文件传输的处理过程。

图 1-2 FTP 中的文件传输



FTP 采用 2 个 TCP 连接来传输文件。

- 控制连接

  以客户端/服务器方式建立。服务器以被动方式打开用于 FTP 的公共端口 21，等待客户端来连接；客户端则以主动方式打开公共端口 21，发起连接的建立请求。

  控制连接始终等待客户端和服务器之间的通信，并且将相关命令从客户端传送给服务器，同时将服务器的应答传送给客户端。

- 数据连接

服务器的数据连接端使用端口 20。服务器执行主动打开数据连接，通常也执行主动关闭数据连接，但是，当客户端向服务器发送流形式的文件时，则需要客户端关闭数据连接。

FTP 中传输方式是流方式，并且文件结尾以关闭数据连接为标志，所以对每一个文件传输或目录列表来说，都要建立一个全新的数据连接。因此，当一个文件在客户端与服务器之间传输时，一个数据连接就建立起来了。

FTP 协议

当前系统 FTP 通过用户协议解析（User-PI：User Protocol Interpretation）和服务器协议解析（Server-PI：Server Protocol Interpretation）配合进行控制连接的管理，通过用户数据传输处理（User-DTP：User Data Transport Process）和服务器数据传输处理（Server-DTP：Server Data Transport Process）完成数据连接的文件传输。

- 客户端

  FTP 的用户接口（UI）给用户提供了一个交互式的命令行操作界面，接收并解释用户输入的命令行，给出帮助信息等。在接收到用户接口发出的命令后，触发用户端协议解析（User-PI）将命令转化成 FTP 标准的命令格式，然后对 FTP 用户端进行控制连接管理。

  - 用户输入的是登录命令，用户端协议解析（User-PI）负责创建与 FTP Server 的控制连接；
  - 用户输入的是目录操作命令，用户端协议解析（User-PI）完成与 FTP Server 之间控制数据的传送与接收；
  - 用户输入文件传输命令，用户端协议解析（User-PI）启动 User-DTP 以完成与 FTP-Server 之间文件数据的传输。用户数据传输处理（User-DTP）负责与 FTP Server 建立起数据连接，向 FTP Server 发送或接收数据，完成数据的传输。数据连接是临时建立的。它在 FTP Server 与 FTP Client 之间需要进行文件或目录列表等数据传输时建立，传输完成后或接收到中断请求时关闭。

- 服务器端

  服务器协议解析（Server-PI）监听 FTP 标准端口（21）等待客户的连接，收到登录连接则触发处理客户端发来的控制命令，并返回应答。

  - 如果收到的是登录命令，则触发启动登录认证过程，认证通过后创建与 FTP Client 的控制连接。
  - 如果收到包含文件数据的传送，则触发启动并控制服务器数据传输处理（Server-DTP）和用户数据传输处理（User-DTP）创建数据连接来传输文件。

  服务器数据传输处理（Server-DTP）支持主动和被动两种数据连接请求。缺省时服务器数据传输处理（Server- DTP）处于主动状态。

  服务器数据传输处理（Server-DTP）在传输数据时，支持用户的强行中断请求，停止数据传输并关闭数据连接。正常状态时，文件传输结束自动关闭数据连接。

FTP 连接建立过程

通过主动模式建立 FTP 数据连接的过程如下：

1. 服务器以被动方式打开端口 21 启动监听，等待连接。
2. 客户端主动发起控制连接的建立请求，建立连接。
3. 客户端控制连接的临时端口与服务器 21 号端口之间的控制连接建立完毕。
4. 客户端发起建立数据连接的命令。

5. 客户端为该数据连接选择一个临时端口号，并且使用 PORT 命令通过控制连接把端口号发送给服务器。
6. 服务器通过控制连接的接收端口号，向客户端发布一个数据连接的打开。
7. 客户端数据连接的临时端口与服务器的 20 号端口之间的数据连接建立完毕。

通过被动模式建立 FTP 数据连接的过程如下：

1. 服务器以被动方式打开端口 21 启动监听，等待连接。
2. 客户端主动发起控制连接的建立请求，建立连接。
3. 客户端控制连接的临时端口与服务器 21 号端口之间的控制连接建立完毕。
4. 客户端发起建立数据连接的命令。
5. 客户端发送给服务器一个命令字 PASV，向服务器请求端口号。
6. 服务器为该数据连接选择一个临时端口号，并且通过控制连接把端口号发送给客户端。
7. 客户端通过控制连接向服务器发布一个数据连接的打开。
8. 客户端数据连接的临时端口与服务器的用户数据连接的临时端口之间的数据连接建立完毕。

1.4.2 TFTP 协议

TFTP（Trivial File Transfer Protocol）即简单文件传送协议。

TFTP 客户端模块提供了使用 TFTP 协议进行文件上传和下载功能，为了保持实现上的简单特性，TFTP 协议使用 UDP 协议进行文件的传输。

与 FTP 相比，TFTP 不具有复杂的交互存取接口和认证控制，适用于客户机和服务器之间不需要复杂交互的环境。例如，系统启动时使用 TFTP 获取系统内存映像。为了保持报文长度短小，TFTP 在 UDP（User datagram Protocol）基础上实现。

当前 AR2200-S 实现了 TFTP 客户端功能，没有提供 TFTP 的服务器端功能。使用 TFTP 协议的客户端功能可以进行文件上传和下载。

TFTP 协议的基本概念

- 操作码

  TFTP 报文的头 2 个字节表示操作码，其取值和含义如下：

  - 1：Read request (RRQ)，表示读请求
  - 2：Write request (WRQ)，表示写请求
  - 3：Data (DATA)，表示数据
  - 4：Acknowledgment (ACK)，表示肯定应答
  - 5：Error (ERROR)，表示出错

- 文件模式

  TFTP 传输文件有两种模式：

  - 二进制模式：用于传输程序文件。
  - ASCII 码模式：用于传输文本文件。

目前，AR2200-S 实现情况是只能作为 TFTP 客户端，且只能使用二进制模式传输文件。

TFTP 的基本原理

- TFTP 不提供用户名和口令

  因为 TFTP 初始设计是用于系统引导进程的，所以它不提供用户名和口令。

- TFTP 协议传输

  TFTP 协议传输由客户端发起

  - 当需要下载文件时，由客户端向 TFTP 服务器发送读请求包，然后从服务器接收数据包，并向服务器发送确认。
  - 当需要上传文件时，由客户端向 TFTP 服务器发送写请求包，然后向服务器发送数据包，并接收服务器的确认。

1.4.3 Telnet 协议

Telnet（Telecommunication Network Protocol）起源于 ARPANET，是一种最早的 Internet 应用，Telnet 提供了一种通过终端远程登录到服务器的方式，呈现一个交互式操作界面。用户可以先登录到一台主机，然后再通过 Telnet 的方式远程登录到网络上的其他主机上去，而不需要为每一台主机都连接一个硬件终端，然后对设备进行配置和管理。

Telnet 协议的基本概念

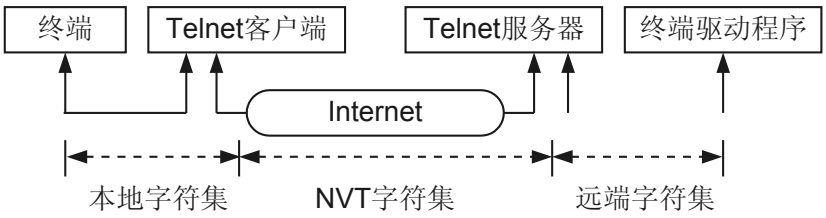
- NVT

  网络虚拟终端（Network Virtual Terminal），是一种双向的虚拟设备，连接的双方都必须把它们各自的物理终端同 NVT 之间进行转换。Telnet 协议可以工作在任何主机（例如，任何操作系统）或任何终端之间正是由于使用了统一的 NVT。

  NVT 是虚拟设备，对于连接的双方，即客户机和服务器，都必须把它们的物理终端和 NVT 进行相互转换。也就是说，不管客户进程终端是什么类型，操作系统必须把它转换为 NVT 格式。同时，不管服务器进程的终端是什么类型，操作系统必须能够把 NVT 格式转换为终端所能够支持的格式。

  物理终端与 NVT 的转换模型如图 1-3 所示。

  图 1-3 物理终端与 NVT 的转换模型

- NVT ASCII

  NVT ASCII 是一个 7 比特的 ASCII 字符集。发送时，每个 7 比特的字符的最高比特位加 0 后，以 8 比特的格式发送。网间网协议族使用的字符集是 NVT ASCII，例如 FTP、SMTP 等。

- IAC

  Telnet 通信的两个方向都是采用带内信令的方式，字节 0xFF 叫做 IAC（Interpret As Command），意思是作为命令来解释。该字节后面的一个字节代表命令。

  以下列出设备用到的命令及其含义：

- SE：子选项结束
- SB：子选项开始
- WILL：选项协商
- WONT：选项协商
- DO：选项协商
- DONT：选项协商
- IAC：其后面的一个字节作为命令来解释

表 1-2 RFC 规定的 Telnet 命令集

名称	代码（十进制）	描述
EOF	236	文件结束符
SUSP	237	挂起当前进程（作业控制）
ABORT	238	异常中止进程
EOR	239	记录结束符
SE	240	子选项结束
NOP	241	无操作
DM	242	数据标记
BRK	243	中断
IP	244	中断进程
AO	245	异常中止输出
AYT	246	对方是否还在运行？
EC	247	转义字符
EL	248	删除行
GA	249	继续进行
SB	250	子选项开始
WILL	251	选项协商
WONT	252	选项协商
DO	253	选项协商
DONT	254	选项协商
IAC	255	数据字节 255

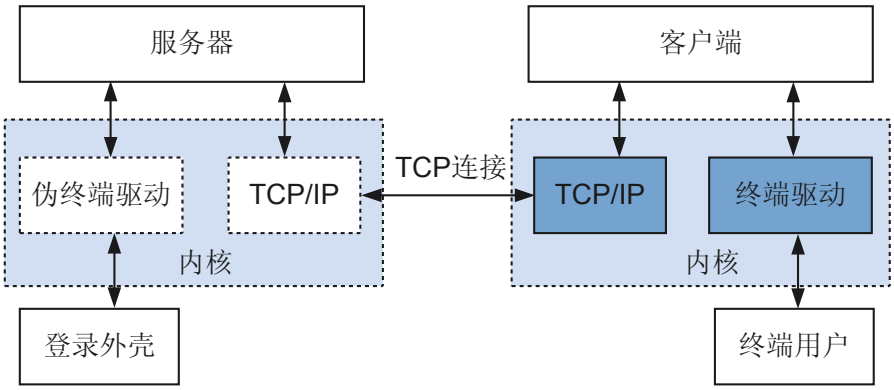
- Telnet 连接

  一个 Telnet 连接就是一个用来传输带有 Telnet 控制信息数据的 TCP 连接。

- Telnet 的 C/S 模式

  Telnet 采用客户端/服务器模式。如图 1-4 显示了 Telnet 客户端和服务器连接的示意图。

图 1-4 Telnet 连接的示意图



通过该示意图，可知：

- Telnet 使用的传输协议为 TCP。
- Telnet 连接的任何回显信息，最终都会输出到终端上。
- 服务器进程直接与“伪终端设备”交互。
- 客户端和服务器通过一条 TCP 连接来传输命令和数据。
- 客户端登录到服务器上。

Telnet 工作原理

Telnet 协议可以工作在任何主机或者任何终端之间。因为，无论客户终端是什么类型，操作系统都会把它转换成 NVT 格式；同时，操作系统也会将 NVT 格式转换成服务器终端所支持的类型。那么，可以屏蔽具体的客户端和终端类型，简单地认为 Telnet 双方都连接在 NVT 上。

说明

Telnet 采用对称性模型，因此理论上一个 Telnet 连接的每一端都必须有一个 NVT。

Telnet 连接的两端，通过“WILL、WONT、DO、DONT”请求来进行选项协商，从而确定 Telnet 服务的具体内容。这些选项包括回显、改变命令字符集、行方式等。

本节从以下几个方面介绍 Telnet 的工作原理：

- Telnet 中的请求

  Telnet 连接的任何一方都可以主动发起请求。请求的含义和用法如表 1-3 所示。

表 1-3 Telnet 中的请求的使用

发送方发出请求	含义	接受方应答			
		WILL	WONT	DO	DONT

发送方发出请求	含义	接受方应答			
WILL	发送方想激活选项	—	—	接收方同意	接收方不同意
WONT	发送方想禁止选项	—	—	—	接收方必须同意(1)
DO	发送方想让接收方激活选项	接收方同意	接收方不同意	—	—
DONT	发送方想让接收方禁止选项	—	接收方必须同意(1)	—	—

说明

发起方发送“选项失效”请求（WONT 和 DONT）时，接收方必须同意；

发起方发送一些“选项有效”的请求，接收方可以接受或者拒绝这些请求：

- 如果接受请求，则选项立即生效；
- 如果拒绝请求，则选项不生效，而发送方仍然能保留 NVT 的特性。

- 选项协商

  选项协商由 3 个字节组成，如下：

  < IAC, WILL/DO/WONT/DONT 之一,选项代码 >

  下面举例说明 Telnet 中如何进行选项协商。

  例如：服务器想跟客户端请求激活“远程流量控制”（选项标识是 33），客户端表示同意激活该选项。两者交互的命令如下：

  - 服务器：< IAC,WILL,33 >
  - 客户端：< IAC,DO,33 >

- 子选项协商

  在主机之间传递选项时，除了一个选项代码外，可能还需要其他更多的信息。例如，要求对方指定终端类型，则对方必须回应一个用 ASCII 字符串表示的终端类型。

  子选项协商的格式如下：

  < IAC,SB,选项代码,子选项内容信息,IAC,SE>

  一次完整的子选项协商过程如下：

  - 发送方发送一个带有选项代码的 DO/WILL 命令来请求激活该选项。
  - 接收方发送一个带有选项代码的 WILL/DO 命令来同意激活该选项。

    至此，双方都同意使用该选项。

    其中一方通过 SB 命令后跟子选项代码，并且以 SE 结尾来开始子选项协商。
  - 对方同样以 SB 命令后跟子选项代码和相关协商信息，并且以 SE 结尾来回应子选项协商。
  - 另一方回应 DO/WILL 命令来同意该子选项。

  如果没有其他子选项需要协商，则本次协商结束。

说明

上述过程假设接收方同意发送方的请求。实际应用中，接收方根据需要，在任何时候都可以拒绝发送方的请求。

下面举一个终端类型协商的示例。

例如：客户端要激活本地的“终端类型”（选项标识是 24）选项；服务器同意；服务器会发出询问客户端的终端类型的请求；客户端于是向服务器发送其终端类型为“DELL PC 机”。两者交互的命令如下：

- 客户端：< IAC, WILL, 24 >
- 服务器：< IAC, DO, 24 >
- 服务器：< IAC, SB, 24, 1, IAC, SE >
- 客户端：< IAC, SB, 24, 0，‘D’，‘E’，‘L’，‘L’，‘P’，‘C’, IAC, SE >

说明

- 只有 DO 类型的发送端可以发送请求。
- 只有 WILL 类型的发送端可以发送实际的类型信息。

终端类型信息不能以自动方式发送，而只能是以<请求－响应>的方式。

终端类型信息是 NVT ASCII String 字符串类型。该类型编码不区分大小写的差别。

- 操作方式

  Telnet 协议规定有 4 种操作方式。

  - 半双工
  - 一次一个字符
  - 一次一行
  - 行方式

设备中提供的 Telnet 服务

设备提供的 Telnet 服务包括：

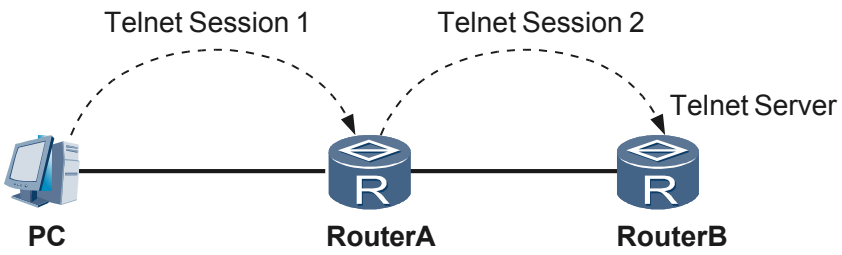
- Telnet Server

  用户在 PC 上运行 Telnet 客户端程序登录到设备，对设备进行配置管理。

- Telnet Client

  用户在 PC 上通过终端仿真程序或 Telnet 客户端程序建立与设备的连接后，再执行 **telnet** 命令登录到其它设备，对其进行配置管理。如图 1-5 所示，RouterA 此时既作为 Telnet Server，也同时提供 Telnet Client 服务。

图 1-5 提供 Telnet Client 服务



1.4.4 SSH 协议

SSH 是 Secure Shell（安全外壳）的简称，标准协议端口号 22。

Telnet 缺少安全的认证方式，而且传输过程采用 TCP 进行明文传输，存在很大的安全隐患。单纯提供 Telnet 服务容易招致 DoS（Deny of Service）、主机 IP 地址欺骗、路由欺骗等恶意攻击。

随着人们对网络安全的重视，传统的 Telnet 和 FTP 通过明文传送密码和数据的方式，已经慢慢不被人接受。SSH（Secure Shell）是一个网络安全协议，通过对网络数据的加密，解决了这个问题。它在一个不安全的网络环境中，提供了安全的远程登录和其他安全网络服务。

SSH 通过 TCP 进行数据交互，它在 TCP 之上构建了一个安全的通道。另外 SSH 服务除了支持标准端口 22 外，还支持其他服务端口，以防止受到非法攻击。

SSH 支持 Password 认证和 RSA 认证，对数据进行 DES、3DES、AES 等加密，有效防止了对密码的窃听，保护了数据的完整性和可靠性，保证了数据的安全传输。特别是对于 RSA 认证的支持，对称加密和非对称加密的混合应用，密钥的安全交换，最终实现了安全的会话过程。

由于 SSH 数据加密传输，认证机制更加安全，SSH 已经越来越被广泛使用，成为了当前最重要的网络协议之一。

SSH 协议有两个版本：SSH1（SSH1.5）协议和 SSH2（SSH 2.0）协议，两者是不同的协议，不兼容。SSH2.0 在安全、功能和性能上均比 SSH 1.5 有优势。

设备支持 STelnet 的客户端和服务器端，以及 SFTP 的客户端和服务器端，均支持 SSH1（SSH1.5）协议和 SSH2（SSH 2.0）协议。

STelnet 是 Secure Telnet 的简称，使得用户可以从远端安全登录到设备，提供交互式配置界面，所有交互数据均经过加密，实现安全的会话。

SFTP 是 Secure FTP 的简称，使得用户可以从远端安全地登录到设备上进行文件管理，这样使远程系统升级等需要文件传送的地方，增加了数据传输的安全性。同时，由于提供了客户端功能，可以在本设备上安全 FTP 到远程设备，进行文件的安全传输。

SSH 的基本概念

- SFTP

  SFTP（SSH File Transfer Protocol）。在一个传统不安全的网络环境中，服务器通过对客户端的认证及双向的数据加密，为网络文件传输提供了安全的服务。

- STelnet

  一种安全的 Telnet 服务。在一个传统不安全的网络环境中，服务器通过对客户端的认证及双向的数据加密，为网络终端访问提供了安全的服务。

- RSA 认证

  RSA（Revest-Shamir-Adleman Algorithm）身份认证是基于客户端私钥的一种认证方式。它是一种公开密钥加密体系，是一种非对称加密算法，其原理是基于大整数因子分解这一著名的数学难题，主要用来传递对称加密算法所使用的密钥，通过这种方法可以有效地提高加密的效率并能简化对密钥的管理。

  服务器必须检查用户是否是合法的 SSH 用户，检查公钥对于该用户是否合法，用户数字签名是否合法。若三者同时满足，则身份认证通过；若其中任何一项不能验证通过均告验证失败，拒绝该用户的登录请求。

- DSA 认证

  DSA（Digital Signature Algorithm，数字签名算法）是由一种用于用户认证的非对称加密算法，由公钥和私钥两部分组成。

  和 RSA 认证相同，服务器必须检查用户是否是合法的 SSH 用户，检查公钥对于该用户是否合法，用户数字签名是否合法。若三者同时满足，则身份认证通过；若其中任何一项不能验证通过均告验证失败，拒绝该用户的登录请求。

  相比 RSA 认证，DSA 认证采用数字签名算法进行加密，具有更广泛的应用性。

  - 在 SSH 中，很多工具仅支持使用 DSA 进行服务器和客户端认证。
  - 按照最新的 SSH RFC 定义，DSA 认证比 RSA 更优先被选择使用。

- Password 认证

  Password 身份认证是基于“用户名＋口令”的一种认证方式。

  在服务器端由 AAA 为每一个合法用户分配一个用于登录时进行身份验证的口令，即在服务器端存在“用户名＋口令”的一一对应的关系。当某个用户请求登录时，服务器需要对用户名以及其口令分别进行鉴别，其中任何一项不能验证通过均告验证失败，拒绝该用户的登录请求。

- RSA-Password 认证

  SSH 服务器可以要求客户端进行身份认证的过程中同时进行 Publickey 身份认证和 Password 身份认证，只有当两者同时满足的情况下，才认为客户端身份认证通过。

- ALL 认证

  服务器可以要求客户端进行身份认证的过程中进行公钥认证或密码认证，只要满足其中一个认证，就认为客户端身份认证通过。

设备中支持的 SSH 特性

- 支持基本的 SSH 协议
  - 支持进入、外出使用不同加密算法。
  - 支持进入、外出使用不同 MAC 算法。
  - 支持 3DES-cbc、DES、AES128（Advanced Encryption Standard）加密算法。
  - 支持 HMAC-sha1 验证算法。
  - 支持 diffie-hellman-group1-sha1 密钥交换算法。
  - 支持 SSH-RSA 公钥格式。
  - 支持密钥重交换 Key Re-Exchange：重新进行密钥协商，协商的内容包括：使用的算法、算法使用的密钥。
  - 支持 Public key、Password 认证方式。
- SSH 支持的客户端功能

  SSH 客户端功能允许用户与支持 SSH Server 的设备、UNIX 主机等建立 SSH 连接。如**图 1-6**、**图 1-7** 所示，可以建立 SSH 通道进行本地连接或广域网连接。

图 1-6 在局域网内建立 SSH 通道

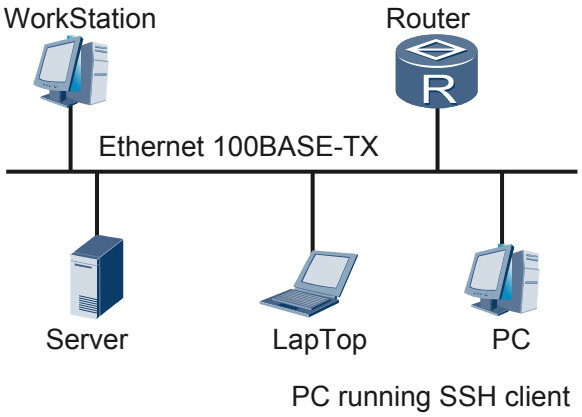
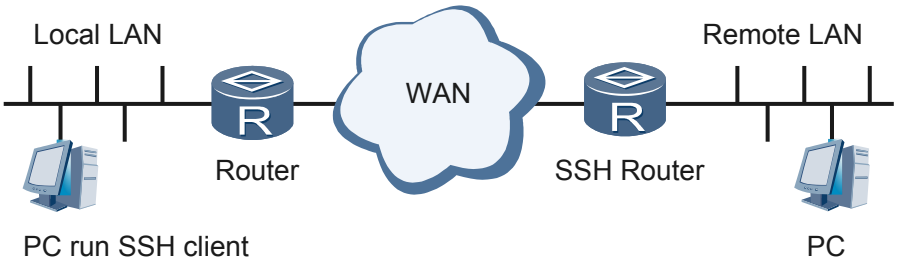


图 1-7 通过广域网建立 SSH 通道



- 支持 SFTP 协议

  SFTP 协议基于 SSH2。在一个传统不安全的网络环境中，服务器通过对客户端的认证及双向的数据加密，为网络文件传输提供了安全的服务。

  设备支持的 SFTP 协议有如下功能：

  - 支持 SFTP 客户端、服务器功能。
  - 支持使能/去使能 SFTP 服务器功能（默认关闭）。
  - 支持对应用户的 SFTP 访问默认目录设定。

- 支持 STelnet 协议

  设备支持的 STelnet 协议有如下功能：

  - 支持 STelnet 客户端、STelnet 服务器功能。
  - 支持使能/去使能 SSH Telnet 服务器功能（默认关闭）。

- SSH 服务支持其他端口

  SSH 协议的标准监听端口号为 22，攻击者不断访问标准端口，导致带宽和服务器性能的下降，其他正常用户无法访问，这是一种 DoS（拒绝服务）攻击。

  设定 SSH 服务端的监听端口号为其他端口，攻击者不知道 SSH 监听端口号的更改，有效防止攻击者对 SSH 服务标准端口访问消耗带宽和系统资源。正常用户通过对非标准端口的 SSH 服务的访问，降低 DoS（拒绝服务）攻击可能性。

  SSH 服务支持其他端口，有如下应用：

- STelnet/SFTP 客户端支持指定其他端口号访问。
- SSH 服务器支持监听端口号的设定。

SSH 协议的基本原理

SSH 采用了传统 Client/Server 应用模型，其安全特性通过以下方式保证。

数据加密：通过 Client/Server 协商交换生成的 Encryption Key，实现对数据报文的对称加密，确保数据在传输过程中的机密性。

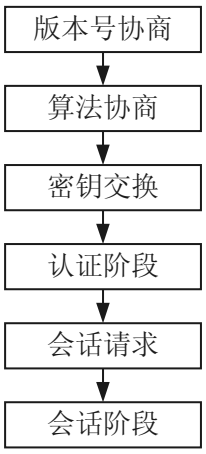
数据完整性：通过 Client/Server 协商交换生成的 Integrity Key 唯一标识一条会话链路，所有会话交互报文被 Integrity Key 标识。一旦数据被第三方修改，接收方就能够检查出来，并丢弃报文，确保数据在传输过程中的完整性。

权限认证：通过提供多种认证方式，确保惟有认证通过的合法用户才能和 Server 进行会话，提高系统安全，同时保障合法用户的权益。

SSH 处理流程

SSH 连接在整个通讯过程中，经过如图 1-8 所示六个阶段，协商实现 SSH 认证安全连接，下面介绍一下 SSH 协商阶段的整个过程。

图 1-8 SSH 连接协商过程



1. 版本协商阶段

   SSH Client 向 Server 发起 TCP 连接请求。TCP 连接建立后，SSH Server 和 Client 之间协商版本号。协商出匹配的版本协议，对不同版本协议走不同的状态机流程。如果版本匹配，进入密钥协商阶段；否则 SSH Server 断开 TCP 连接。

2. 算法协商

   当进入算法协商过程后，发送方向接收方发送算法协商消息包，发送方同时将随机 Cookie、密钥交换的算法、主机密钥算法、支持的 MAC（Message Authentication Code）认证方法以及支持的语言等作为消息的参数发送给接受方。

   接收方等待接收到发送方的算法协商消息后，将发送方的算法列表集合与本地算法列表集合相比较。若密钥交换算法、公钥加密算法和 MAC 算法当中有一项没有找到，则断开与发送方的连接，算法协商失败。

3. 密钥交换阶段

   当算法协商通过后，进入密钥交换阶段。双方开始计算会话 ID。客户端随机生成一个 32 字节的会话密钥（session key），用该密钥的前 16 字节异或（XOR）会话 ID 的 16 字节，后 16 字节不变，所得结果按 MSB 排列成一个 MP 型整数。用主机公钥和服务器公钥中模数较小的公钥进行加密，把结果按 MSB first 顺序排列成一个 MP 型整数，用模数较大的公钥进行加密，并把所得结果和客户端选择的加密算法、服务器传来的 8 字节 Cookie、自身的协议标志一起发给服务器端。

   会话过程中，大数据量传输必须使用处理速度较快的对称密钥算法。对称加解密需要共享密钥，密钥交换流程目的是在不安全的通道中安全传送会话密钥。

   服务器处于等待状态，当收到客户端发送的密钥生成消息包后，服务器回送一个密钥生成消息给客户端，表示密钥交换完成，采用新的密钥进行通讯。如果失败，则返回密钥交换失败消息，并断开连接。

4. 认证阶段

   在计算出会话密钥后，SSH Server 对 Client 进行用户身份验证。SSH Client 向 Server 发送用户身份信息。如果 SSH Server 上配置该用户无需验证，则直接进入请求会话阶段；如果在 SSH Server 上配置对该用户进行验证，Client 将采用配置的验证方法向 Server 提出验证请求，直到验证通过或连接超时断开。

   SSH Server 提供四种验证方法：RSA、Password、RSA-Password 和 ALL。

   - 在 RSA 认证方式下，用户客户端程序生成一个 RSA 密钥对，并将公钥部分发送给服务器端。用户发起认证请求时，客户端程序随机生成一段有私钥加密的密文并发送给服务器端，服务器端利用公钥对其进行解密，解密成功就认为用户是可信的，随机对用户授予相应得访问权限。否则，中断连接。
   - Password 认证依靠 AAA 实现，与 Telent 和 FTP 类似，支持本地数据库和远程 RADIUS 服务器验证，SSH Server 对来自 Client 的用户名/口令和预先配置的用户名/口令进行比较，如果完全匹配则验证通过。

5. 会话请求阶段

   认证完成后，客户端向服务器提交会话请求。会话请求包含 shell 和命令。服务器则进行等待，处理客户端的请求。在这个阶段，无论什么请求只要成功处理了，服务器都向客户端回应认证成功消息；否则回应认证失败消息，这表示认证失败。

   认证失败的原因如下：

   - 服务器处理请求失败
   - 服务器不能识别请求

6. 交互会话阶段

   会话申请成功后，连接进入交互会话模式。在这个模式下，数据在两个方向上双向传送。

   a. 客户端将要执行的命令加密后打包传给服务器。

   b. 服务器接收到报文，解密后执行该命令，并将执行的结果加密、打包、发还给客户端。

   c. 客户端将接收到的结果解密后显示到终端上。

1.4.5 用户管理

在配置了用户界面、用户管理和终端服务后，用户才能登录到设备，对本地或远端的网络设备进行配置、监控和维护。用户界面提供登录场所，用户管理确保登录安全，终端服务则提供登录协议支持。

目前设备支持的登录方式包括：

- 通过 Console 口登录。
- 通过 Telnet 或 SSH 进行本地或远程登录。

用户管理完成用户界面、用户视图配置和终端服务等工作，实现提供用户安全登录并进行安全操作，达到统一管理各种用户界面的目的。

用户界面

用户界面提供登录场所，呈现形式是用户界面（User-interface）视图。用户界面视图是系统提供的一种命令行视图，用来配置和管理所有工作在异步交互方式下的物理接口和逻辑接口，从而达到统一管理各种用户界面的目的。

- 目前系统支持的用户界面
  - Console（CON）口：控制口（Console Port）是一种线设备端口，由设备的主控板提供。

    用户终端的串行端口可以与设备 Console 口直接连接，实现对设备的本地配置。
  - 虚拟线路（VTY）

    虚拟终端连接（Virtual Terminal）属于逻辑终端线设备。

    用户通过终端与设备建立 Telnet / SSH 连接后，即建立了一条 VTY。用户可通过与 VTY 建立的虚拟连接实现对设备进行本地或远程访问。
  - 异步工作方式串口（TTY）

    TTY（True Type Terminal）是指通过工作在异步方式下的串口（可以是异步串口，也可以是工作在异步方式下的同步串口）。
- 用户界面的编号

  用户界面的编号有两种方式：相对编号方式和绝对编号方式。
  - 相对编号方式

    相对编号方式的形式是：用户界面类型＋编号。

    此编号是每种相同类型的用户界面的编号。此种编号方式只能唯一指定某种类型的用户界面中的一个或一组，而不能跨类型操作。相对编号方式遵守的规则如下：

    控制台（CON）的编号：CON 0。

    异步工作方式串口（TTY）的编号：第一个为 TTY 0，第二个为 TTY 1，依此类推。

    虚拟线路（VTY）的编号：第一个为 VTY 0，第二个为 VTY 1，依此类推。
  - 绝对编号方式

    绝对编号可以唯一的指定一个或一组用户界面。

    绝对编号的起始编号是 0，并按照 CON、TTY、VTY 的顺序依次分配。

    每个主控板上 CON 口只有一个，但 VTY 类型的用户界面可以达到 0 ～ 20 个（其中 0 ～ 14 用户提供给普通 Telnet /SSH 用户的用户接口，16 ～ 20 是预留给网管用户的接口）。在系统视图下可以设置最大用户界面个数。

    缺省情况下，CON、TTY、VTY 用户接口，在系统中的绝对编号如**表 1-4** 所示。

表 1-4 用户界面的绝对编号示例

绝对编号	用户界面
0	CON0
1	TTY0 第一个 TTY 类型用户界面
2	TTY1 第二个 TTY 类型用户界面
3	TTY2 第三个 TTY 类型用户界面
4	TTY3 第四个 TTY 类型用户界面
5	TTY4 第五个 TTY 类型用户界面
129	VTY0 第一个 VTY 类型用户界面
130	VTY1 第二个 VTY 类型用户界面
131	VTY2 第三个 VTY 类型用户界面
132	VTY3 第四个 VTY 类型用户界面
133	VTY4 第五个 VTY 类型用户界面

用户登录

第一次启动设备时，设备没有设置用户名和口令。

在不对登录用户进行验证的情况下，只要将计算机终端通过 Console 口与设备连接，任何用户可以对设备进行配置。

如果配置了主控板或接口板的 IP 地址，任何远端用户可能使用 Telnet/SSH 登录到设备。

这对设备和网络是不安全的。为此需要为设备创建用户，并为用户设置口令，对用户进行管理。

用户的优先级

在 AR2200-S 上命令划分为 3 个级别（0 ～ 2 级）：

- 0 级的命令，网络诊断工具命令（ping、tracert）、从本设备出发访问外部设备的命令（Telnet 客户端）等。该级别命令不允许进行配置文件保存的操作。
- 1 级的命令，用于系统维护，包括 display 等命令。
- 2 级的命令，业务配置命令，包括路由、各个网络层次的命令，向用户提供直接网络服务。
- 3 级的命令，用于系统基本运行的命令，对业务提供支撑作用，包括文件系统、FTP、TFTP 下载、配置文件切换命令、备板控制命令、用户管理命令、命令级别设置命令、系统内部参数设置命令；用于业务故障诊断的 debugging 命令等。

AR2200-S 对用户进行分级管理，将用户划分为 16 个级别（0 ～ 15 级）。

用户的级别决定了用户能够执行的命令：

- 用户级别为 0 级的用户，只能执行 0 级的命令。

- 用户级别为 1 级的用户，可以执行 0 级和 1 级的命令。
- 用户级别为 2 级的用户，可以执行 0 ～ 2 级的命令。
- 用户级别为 3 ～ 15 级的用户，可以执行 0 ～ 3 级的命令。

对用户的验证

配置用户后，用户登录设备时，系统对用户的身份进行验证。

对用户的验证有 3 种方式：不验证、password 验证和 AAA 验证。

- 不验证：不需要用户名和口令，可直接登录设备。为安全起见，不推荐采用这种方式；
- Password 验证：这种认证只需要口令，不需要用户名；而是基于终端线的配置，可对一个或者一组终端线配置一个口令；
- AAA 验证：AAA 验证包括 AAA 本地验证和 AAA 远程验证。本地验证需要在本地设备上配置用户名和口令。如果有必要，还需要配置相关的用户属性，比如用户权限，用户 FTP 路径等。远程验证中需要 AAA 服务器的配合，在服务器上配置用户的相关信息。一般来说，对于 VTY 用户，多采用服务器认证；对于 Console 用户，采用本地认证或者不认证。详细可参见 AAA 特性描述。

规划设备的用户

网络管理员可以根据需要规划设备的用户。

- 通常，设备至少需要创建一个超级终端用户。
- 如果需要从远端使用 Telnet/SSH 登录到设备，则需配置一个 Telnet/SSH 用户。
- 为了让远端用户向设备加载或下载文件，可以配置 FTP/SFTP 用户。

1.4.6 虚拟文件系统

文件系统实现两类功能：管理存储设备和管理保存在存储设备中的文件。文件系统是指对存储设备中的文件、目录的管理，包括创建文件系统，创建、删除、修改、更名文件和目录，以及显示文件的内容。为了更有效地管理大容量存储设备，同时屏蔽底层存储设备不同，设备提供了方便用户操作并且可裁减的虚拟文件系统。

基本概念

- 存储设备：存储信息的硬件设备。

  设备目前支持的存储设备为。
- 文件：系统存储信息，并对信息进行管理的一种机制。
- 目录：一种将整个文件集合进行组织的机制，目录是文件的逻辑上的容器。

管理存储设备

- 修复文件系统异常的存储设备

  当某存储设备上的文件系统出现异常时，设备的终端会给出提示信息，建议修复。
- 格式化存储设备

  当文件系统的异常无法恢复或者确认不再需要存储设备上的所有数据时，可格式化存储设备。

如果执行格式化的命令后，存储设备仍然不可用，则可能是物理原因导致的存储设备不可用。

管理目录

当需要在客户端与服务器端进行文件传输时，需要使用文件系统对目录进行配置。具体包括：

- 查看当前的工作目录
- 改变当前目录
- 显示目录或文件信息
- 创建目录
- 删除目录

说明

所有路径都支持绝对路径地址或相对于当前工作路径的相对路径。

管理文件

用户可以针对文件进行如下操作：

- 显示文件内容
- 压缩和解压缩文件
- 拷贝文件：拷贝的文件必须大于 0 字节，否则不能执行。
- 移动文件：用户将文件存放的位置进行更改。
- 重新命名文件：对已经存在的文件更改文件名。
- 删除文件：删除已经存在的文件，该操作实际上是将文件放到回收站中，具有可逆性。删除文件可以使用通配符“*”，一次删除多个文件。
- 彻底删除回收站中的文件：将回收站中的文件彻底删除，不可恢复。
- 恢复删除文件：将回收站中的文件恢复，是“删除文件”的逆向操作。

其他

- 运行批处理文件

  如果已经建立好批处理文件，那么可以执行该文件，以实现执行固定任务的自动化。

  此操作需要在客户端编辑批处理文件，并且上传到设备。
- 配置文件系统提示方式

  在操作文件时，如可能导致数据丢失或破坏，则需配置系统给出提示信息。

注意

如果将文件操作的提醒方式设置为 quiet，则对由于用户误操作（比如删除文件操作）而导致数据丢失的情况不作提示，请慎用。

1.4.7 夏令时

夏令时（Daylight Saving Time：DST），又称“日光节约时制”或“夏时制”，是一种为节约能源而人为规定地方时间的制度，在这一制度实行期间所采用的统一时间称为“夏令时间”。

高纬度地区由于夏季太阳升起时间明显比冬季早，人为将时间提前一小时，减少照明量，以充分利用光照资源，从而实现节约能源。目前全世界有近 110 个国家每年要实行夏令时。

用户可以根据本国或本地区的规定，自由地定制实行夏令时的区间。除此之外，用户还可以设定夏令时向前调整的时间长短，通常是一个小时。使能夏令时功能之后，当进入夏令时实施区间的一刻，系统时间会根据用户的设定进行夏令时时间的调整。当到达夏令时实施区间的结束时刻，系统时间会自动向前调整与设置偏移等同的时间长度，恢复正常的时间设置。

1.4.8 定时重启

企业的设备升级有明确的时间要求，如果设置了定时重启功能，维护人员无需等待，只需要把待升级版本的配置、系统映象文件准备妥当，并设置好设备重启时间和重启文件。待重启时间到达时，系统将自动执行重启，更新系统文件。

1.5 应用

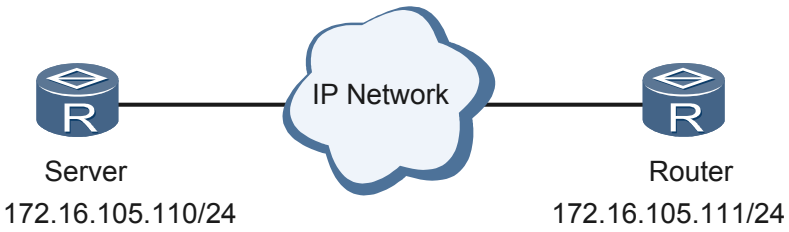
1.5.1 FTP

- 设备作为 FTP 客户端

  从充当 FTP 客户端的设备上登录到 FTP 服务器，从服务器端下载系统文件到客户端的存储设备中。

  如图 1-9 中，IP 地址为 172.16.105.111 的设备作为 FTP 客户端。则可以在作为 FTP 客户端的设备上以 FTP 的方式登录到 FTP 服务器上去。

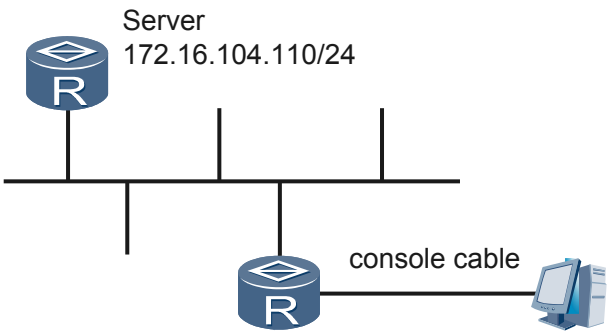
  图 1-9 设备作为 FTP 客户端

- 设备作为 FTP 服务器

  设备作为 FTP 服务器，从超级终端登录到客户端上，再从 FTP 服务器下载文件。如图 1-10 中，IP 地址为 172.16.104.110 的设备作为 FTP 服务器。

图 1-10 设备作为 FTP 服务器



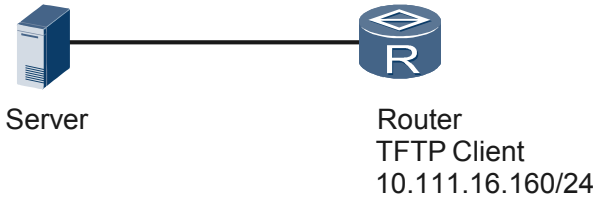
1.5.2 TFTP

使用 TFTP 上传/下载文件：

当用户需要从服务器上传/下载文件且不需要复杂的交互环境时，可以使用 TFTP 协议。设备目前只能作为 TFTP 客户端使用。

组网图如图 1-11 所示。

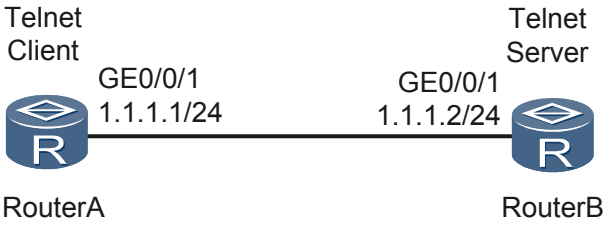
图 1-11 利用 TFTP 功能下载或上传文件



1.5.3 Telnet

Telnet 通常用在远程登录应用中，以便对本地或远端运行设备的网络设备进行配置、监控和维护。

图 1-12 Telnet 登录方式组网图



1.5.4 SSH

- 支持 STelnet 的客户端和服务器端

  STelnet 客户端基于 SSH2 协议，服务器端基于 SSHv1.x 和 SSHv2 协议。客户端和服务器之间经过协商，建立安全连接，客户端可以像操作 Telnet 一样登录服务器，之后进行配置操作，如图 1-13 所示。

  图 1-13 SSH 支持 Stelnet 协议

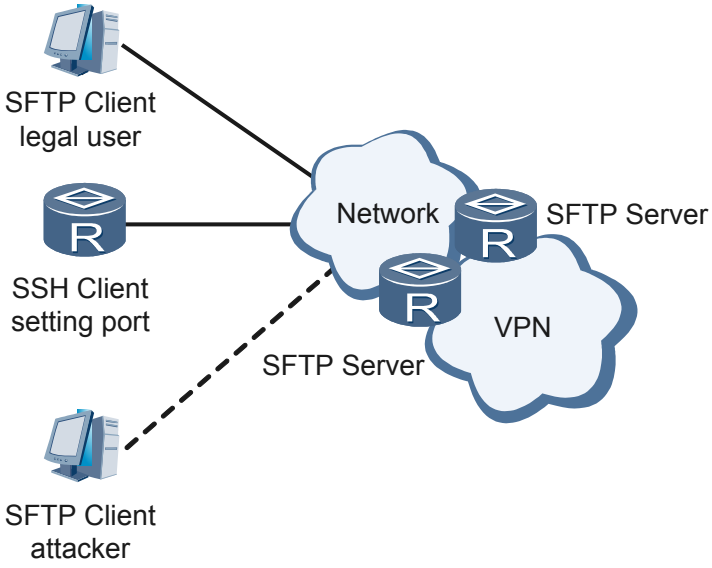
  - 设备支持 STelnet 客户端、STelnet 服务器功能：为了方便用户的使用，设备不仅提供 STelnet 服务器功能，同时也可以做为 STelnet 客户端访问其他 STelnet 服务器。
  - 支持使能/去使能 STelnet 服务器功能（默认关闭）：在不需要 STelnet 服务器情况下可以将其去使能，该功能在全局模式下配置。
- 支持 SFTP 的客户端和服务器端

  攻击者没有正确的私钥和密码，无法通过服务器的认证，并且攻击者无法获得其他用户和服务器之间的会话密钥，因此后续服务器和指定客户端的通讯报文只有指定客户端和服务器才能解密。即使攻击者窃听到通讯报文，也不能解密，实现了网络数据传输的安全性。

  SFTP 是基于 SSH2 的安全文件传输协议。SSH2 支持两种认证方式：密码认证和 RSA 认证。合法用户通过客户端登录时，输入正确的用户名以及对应的密码和私钥，通过服务器的验证。此时用户可以像使用 FTP 一样使用，实现对网络文件的远程传输管理，而系统会对用户的数据采用协商出来的会话密钥对数据加密。

  - 支持 SFTP 客户端、SFTP 服务器功能：为了方便用户的使用，设备不仅提供 SFTP 服务器功能，同时也可以做为 SFTP 客户端访问其他 SFTP 服务器。
  - 支持使能/去使能 SFTP 服务器功能（默认关闭）：在不需要 SFTP 服务器情况下可以将其去使能，该功能在全局模式下配置。
  - 支持对应用户的 SFTP 访问默认目录设定：对于不同的用户，服务器允许访问的文件目录不同。用户只能访问 SFTP 服务设定目录，因此通过对应用户的 SFTP 访问默认目录设定实现不同用户文件隔离。

图 1-14 SSH 支持 SFTP 协议

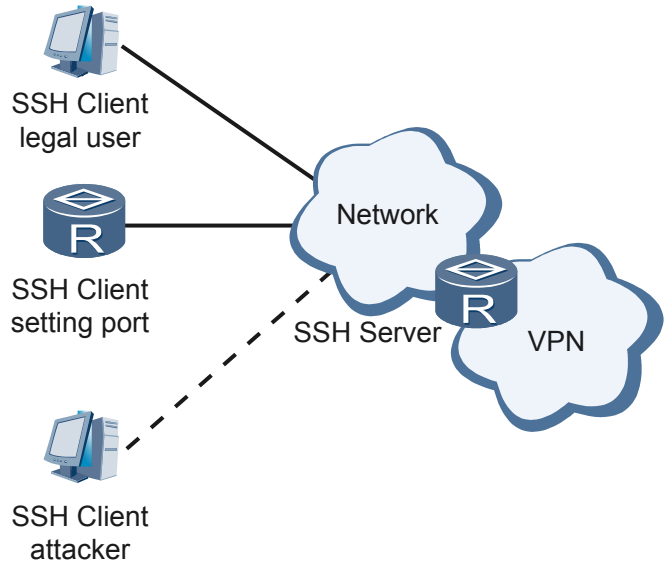


- 私网访问

  设备支持 STelnet 客户端、SFTP 客户端，因此可以建立基于 VPN 的 Socket 连接，在公网设备实现如下访问：

  - STelnet 客户端访问私网 SSH 服务器
  - SFTP 客户端访问私网 SSH 服务器

图 1-15 SSH 支持私网访问



- SSH 服务器支持其他端口访问

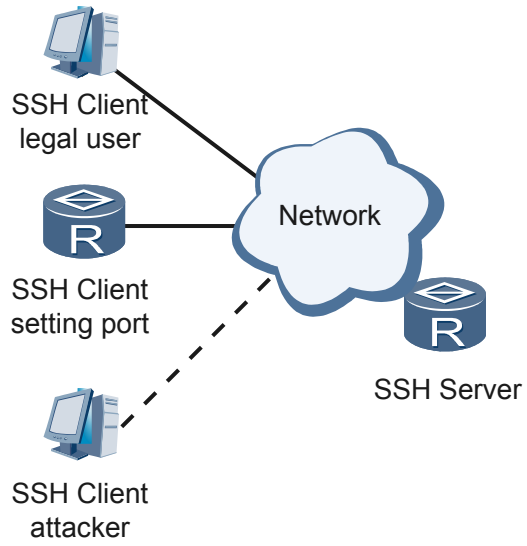
  SSH 协议的标准监听端口号为 22，如果攻击者不断访问标准端口，将致带宽和服务器性能的下降，导致其他正常用户无法访问。

通过设定 SSH 服务器端的监听端口号为其他端口号，攻击者不知道 SSH 监听端口号的更改，仍然发送标准端口号 22 的 Socket 连接，SSH 服务器检测发现请求连接端口号不是监听的端口号，就不建立 Socket 连接。

只有合法的用户采用 SSH 服务器设定的非标准监听端口才能建立 Socket 连接，进行 SSH 协议的版本号协商、算法协商及会话密钥生成、认证、会话请求和会话阶段等过程。

SSH 服务可以应用在网络中的中间交换设备、边缘设备上，可以实现用户对设备的安全访问和管理。

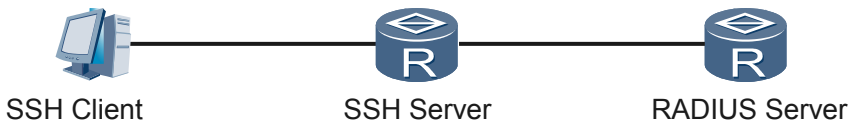
图 1-16 SSH 服务器支持其他端口访问



- 支持 RADIUS

  SSH 的如果使用密码认证，与 FTP、Telnet 一样，也是调用 AAA 提供的接口。在 AAA 中配置用户认证为 RADIUS 方式，当 SSH 认证启动时，SSH 服务器将 SSH 客户端的认证信息（用户名、密码）发送给 RADIUS 服务器（兼容 TACACS 服务器）进行认证；RADIUS 服务器认证该用户，将认证结果（成功、失败，如果成功还包含用户等级）返回给 SSH 服务器。SSH 服务器根据认证结果决定是否允许 SSH 客户端建立连接。

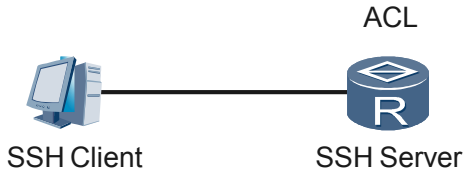
图 1-17 SSH 支持 RADIUS



- 支持 ACL 应用

  ACL 是地址访问控制列表。通过 ACL 对 SSH 类型的用户界面限制呼入呼出权限，防止一些非法地址的用户进行 TCP 连接，避免其进入 SSH 协商，借此提高 SSH 服务器安全性。

图 1-18 SSH 支持 ACL 应用



1.6 术语与缩略语

术语

术语	解释
FTP	File Transfer Protocol—文件传输协议，在 TCP/IP 协议族中属于应用层协议，主要向用户提供本地和远程主机之间的文件传输。FTP 协议基于相应的文件系统实现。
TFTP	Trivial File Transfer Protocol—简单文件传输协议。
Telnet	Telecommunication network protocol—电信网络协议。在 TCP/IP 协议族中属于应用层协议。Telnet 提供了一种通过终端远程登录到服务器的方式，呈现一个交互式操作界面。
NVT	Network Virtual Terminal—网络虚拟终端。是一种双向的虚拟设备，连接的双方都必需把它们各自的物理终端同 NVT 之间进行转换。Telnet 协议可以工作在任何主机（例如，任何操作系统）或任何终端之间正是由于使用了统一的 NVT。
SSH	Secure Shell—安全外壳协议，SSH 使用多种加密方式和认证方式，解决了以上传统服务的数据加密、身份认证问题。SSH 成熟的公钥/私钥体系，为客户端和服务端之间的会话提供加密通道，解决了数据（包括口令）在网络上以明文传输的不安全问题。SSH 还支持 CA、smart 卡等多种认证方式，解决了身份认证问题和克服了 man-in-the-middle 攻击等不安全因素。
SFTP	Secure File Tranfer Protocol—基于 SSH 协议的上层应用，提供安全的文件传输功能。
Stelnet	Secure Shell Telnet—基于 SSH 协议的上层应用，提供安全的登录操作功能。

缩略语

缩略语	英文全称	中文全称
FTP	File Transfer Protocol	文件传输协议
TFTP	Trivial File Transfer Protocol	简单文件传输协议。

缩略语	英文全称	中文全称
Telnet	Telecommunication network protocol	电信网络协议
NVT	Network Virtual Terminal	网络虚拟终端
SSH	Secure Shell l	安全外壳协议
SFTP	Secure File Tranfer Protocol	安全文件传输协议
AES	Advanced Encryption Standard	高级加密标准。
ACL	Access control list	访问控制列表
MAC	Message Authentication Code	消息认证码
RSA	Revest，Shamir and Adleman	RSA 加密系统（三个人的名字）
TACACS	Terminal Access Controller Access Control System	终端访问控制器控制系统
VPN	Virtual Private Network	虚拟专用网
TTY	Terminal controller（A/S or SA）	异步口或同异步口的统称
CON	Console，Primary terminal line	主控制台接口
AAA	Authentication，Authorization，Accounting	认证、授权、计费
VRP	Versatile router platform	通用路由平台
NAP	Neighbor Access Protocol	邻居访问协议