ICS 25.040
N 10

# 中华人民共和国国家标准化指导性技术文件

GB/Z 19582.3—2004

# 基于 Modbus 协议的工业自动化网络规范 第 3 部分：Modbus 协议在 TCP/IP 上的实现指南

**Modbus industrial automation network specification—Part 3: Modbus protocol implementation guide over TCP/IP**

2004-09-21 发布　　2005-03-01 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会　发布

中 华 人 民 共 和 国
国家标准化指导性技术文件
**基于 Modbus 协议的工业自动化网络规范**
**第 3 部分:Modbus 协议在 TCP/IP 上的**
**实 现 指 南**
GB/Z 19582.3—2004
*
中国标准出版社出版发行
北京复兴门外三里河北街 16 号
邮政编码:100045
网址 www.bzcbs.com
电话:68523946 68517548
中国标准出版社秦皇岛印刷厂印刷
各地新华书店经销
*
开本 880×1230 1/16 印张 2.5 字数 70 千字
2004 年 12 月第一版 2004 年 12 月第一次印刷
印数 1—5 000
*
书号:155066·1-21774 定价 18.00 元

# 前　言

本指导性技术文件包括两个通信规程中使用的 Modbus 应用层协议和服务规范：

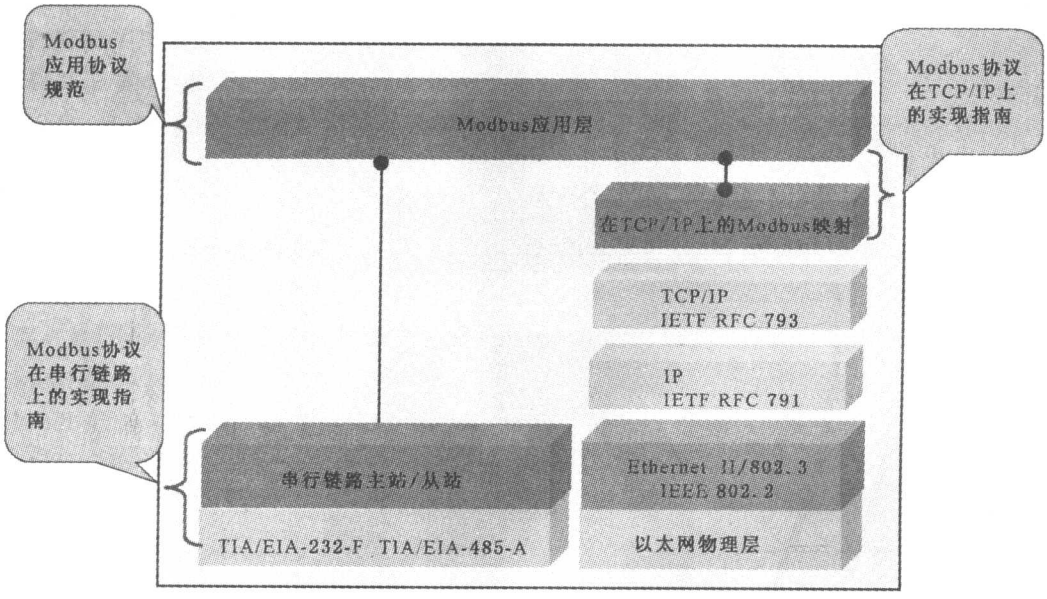
——串行链路上的 Modbus

Modbus 串行链路基于 TIA/EIA 标准：232-F 和 485-A。

——TCP/IP 上的 Modbus

Modbus TCP/IP 基于 IETF 文件：RFC793 和 RFC791。

串行链路和 TCP/IP 上的 Modbus 是根据相应 ISO 分层模型说明的两个通信规程。下图强调指出了本指导性技术文件的主要部分。深色方框表示规范，浅色方框表示已有的国际标准（TIA/EIA 和 IETF 标准）。




基于 Modbus 协议的工业自动化网络规范分为三部分。

——第 1 部分：Modbus 应用协议

——第 2 部分：Modbus 协议在串行链路上的实现指南

——第 3 部分：Modbus 协议在 TCP/IP 上的实现指南

第 1 部分描述了 Modbus 事务处理；第 2 部分提供了一个有助于开发者实现串行链路上的 Modbus 应用层的参考信息；第 3 部分提供了一个有助于开发者实现 TCP/IP 上的 Modbus 应用层的参考信息。

本部分由中国机械工业联合会提出。

本部分由全国工业过程测量和控制标准化技术委员会归口。

本部分起草单位：机械工业仪器仪表综合技术经济研究所、北京交通大学现代通信研究所、上海自动化仪表股份有限公司、施耐德电气（中国）投资有限公司、冶金工业钢铁研究总院、宝钢集团上海宝信软件股份有限公司。

本部分主要起草人：欧阳劲松、孙昕、刘铁椎、冯晓升、王勇、张荣生、丛力群、段永康。

# 目　次

# 基于 Modbus 协议的工业自动化网络规范
# 第 3 部分：Modbus 协议在 TCP/IP 上的实现指南

## 1 范围

本部分叙述了 TCP/IP 上的 Modbus 报文传输服务，提供参考信息以帮助软件开发者使用这种服务。本标准不包括 Modbus 功能码的编码内容，有关这些内容见 GB/Z 19582.1—2004《基于 Modbus 协议的工业自动化网络规范 第 1 部分：Modbus 应用协议》。

本部分描述了 Modbus 报文传输服务的实现。其目的是促进在使用 Modbus 报文传输服务的设备之间的互操作。

本部分主要由三部分组成：

——在 TCP/IP 上的 Modbus 协议概述；

——Modbus 客户机和服务器实现的功能描述；

——针对一个 Modbus 实现实例的对象模型建议的实现准则。

## 2 客户机/服务器模型

Modbus 报文传输服务提供设备之间的客户机/服务器通信，这些设备联接在一个 Ethernet（以太网）TCP/IP 网络上（见图 1）。

这个客户机/服务器模型基于 4 种报文类型：

——Modbus 请求

——Modbus 证实

——Modbus 指示

——Modbus 响应

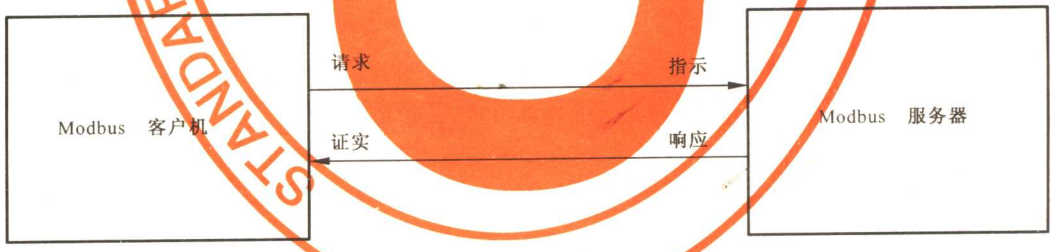



图 1 Modbus 客户机/服务器模型

Modbus 请求是客户机在网络上发送用来启动事务处理的报文；

Modbus 指示是服务器侧接收的请求报文；

Modbus 响应是服务器发送的响应报文；

Modbus 证实是在客户机侧接收的响应报文。

Modbus 报文传输服务（客户机/服务器模型）用于实时信息交换：

——在两个设备应用程序之间；

——在设备应用和其他设备之间；

——在 HMI/SCADA 应用程序和设备之间；

——在一个 PC 和一个提供在线服务的设备程序之间。

## 3 规范性引用文件

下列文件中的条款通过 GB/Z 19582 本部分的引用而成为本部分的条款。凡是注日期的引用文件，其随后所有的修改单(不包括勘误的内容)或修订版均不适用于本部分，然而，鼓励根据本部分达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件，其最新版本适用于本部分。

GB/Z 19582.1—2004 基于 Modbus 协议的工业自动化网络规范 第 1 部分：Modbus 应用协议

RFC 1122 Requirements for Internet Hosts—Communication Layers.

## 4 缩略语

| | |
|---|---|
| ADU(Application Data Unit) | 应用数据单元 |
| IETF(Internet Engineering Task Force) | 互联网工程工作组 |
| IP(Internet Protocol) | 互联网协议 |
| MAC(Medium Access Control) | 介质访问控制 |
| MB(MODBUS) | Modbus |
| MBAP(MODBUS Application Protocol) | Modbus 应用协议 |
| PDU(Protocol Data Unit) | 协议数据单元 |
| PLC(Programmable Logic Controller) | 可编程序逻辑控制器 |
| TCP(Transport Control Protocol) | 传输控制协议 |
| BSD(Berkeley Software Distribution) | 伯克利软件发布 |
| MSL(Maximum Segment Lifetime) | 最大段寿命 |
| SCADA(Supervisory Control and Data Acquisition) | 数据采集和监控 |

## 5 背景概要

### 5.1 协议描述

#### 5.1.1 总体通信结构

见图 2～图 3。

Modbus TCP/IP 的通信系统可以包括不同类型的设备：

——连接至 TCP/IP 网络的 Modbus TCP/IP 客户机和服务器设备；

——互连设备，例如：在 TCP/IP 网络和串行链路子网之间互连的网桥、路由器或网关，该子网允许将 Modbus 串行链路客户机和服务器终端设备连接起来。

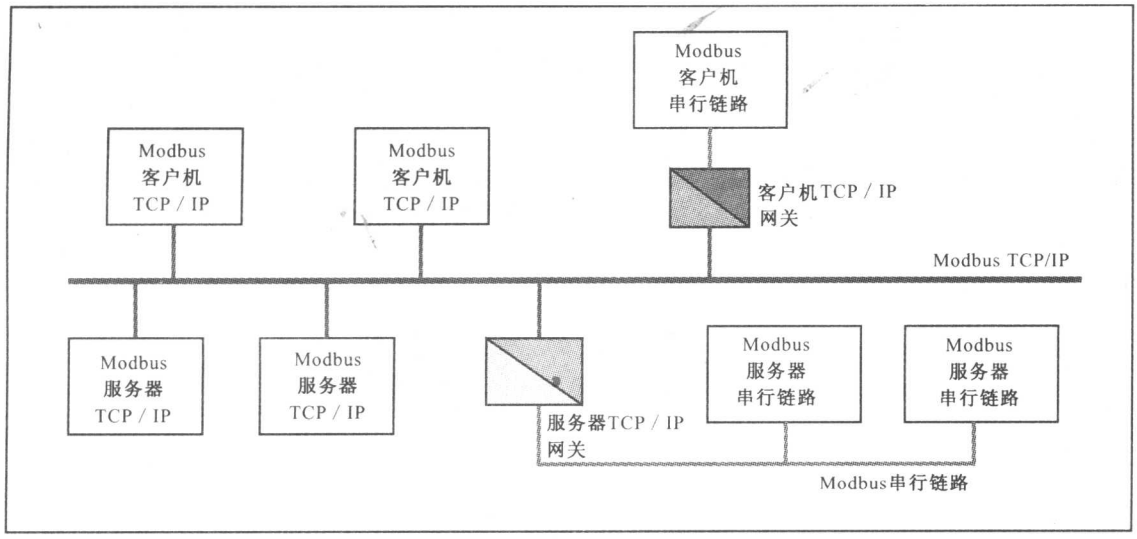



图 2 Modbus TCP/IP 通信结构

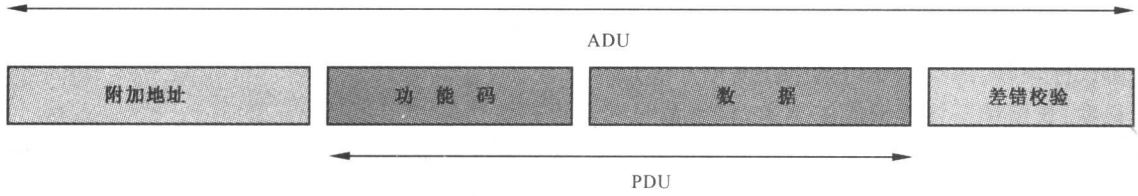



**图 3 通用 Modbus 帧**

Modbus 协议定义了一个与基础通信层无关的简单协议数据单元(PDU)。特定总线或网络上的 Modbus 协议映射能够在应用数据单元(ADU)上引入一些附加域。

启动 Modbus 事务处理的客户机建立 Modbus 应用数据单元。这个功能码向服务器指示执行何种操作。

### 5.1.2 TCP/IP 上的 Modbus 应用数据单元

见图 4。

本节描述了 Modbus TCP/IP 网络上进行的 Modbus 请求或响应的封装。

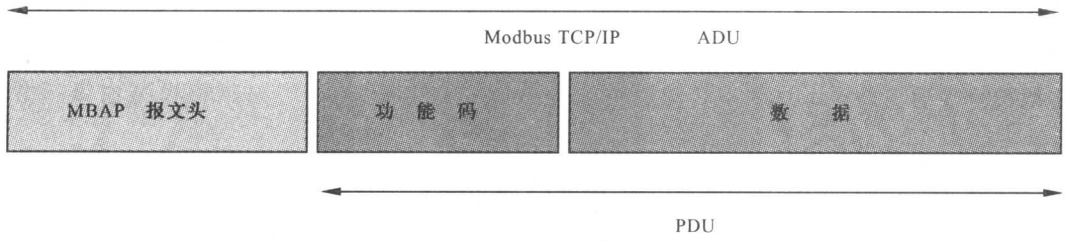



**图 4 TCP/IP 上的 Modbus 的请求/响应**

在 TCP/IP 上使用一种专用报文头来识别 Modbus 应用数据单元。将这种报文头称为 MBAP 报文头(Modbus 应用协议报文头)。

与串行链路上使用的 Modbus RTU 应用数据单元相比,这种报文头有一些区别:

——用 MBAP 报文头中的单字节"单元标识符"取代 Modbus 串行链路上通常使用的 Modbus 从站地址域。这个"单元标识符"用于设备的通信,这些设备使用单个 IP 地址支持多个独立 Modbus 终端单元,例如:网桥、路由器和网关。

——用接收者可以验证报文结束的方式设计所有的 Modbus 请求和响应。对于 Modbus PDU 有固定长度的功能码来说,仅功能码就足够了。对于在请求或响应中传输一个可变数据量的功能码来说,数据域包括字节数。

——通过 TCP 传输 Modbus 时,即使将报文分成多个信息包来传输,需在 MBAP 报文头上传输附加长度信息,以便接收者能识别报文边界。显式和隐式长度规则的存在以及 CRC—32 差错校验码的使用(在以太网上),使未检出的请求或响应报文的差错降至极低。

### 5.1.3 MBAP 报文头描述

MBAP 报文头包括的各个域,见表 1:

**表 1**

| 域 | 长度 | 描述 | 客户机 | 服务器 |
|---|---|---|---|---|
| 事务处理标识符 | 2 字节 | Modbus 请求/响应事务处理的识别码 | 客户机启动 | 服务器从接收的请求中重新复制 |
| 协议标识符 | 2 字节 | 0=Modbus 协议 | 客户机启动 | 服务器从接收的请求中重新复制 |
| 长度 | 2 字节 | 随后字节的数量 | 客户机启动(请求) | 服务器启动(响应) |
| 单元标识符 | 1 字节 | 串行链路或其他总线上连接的远程从站的识别 | 客户机启动 | 服务器从接收的请求中重新复制 |

报文头长度为 7 个字节：

事务处理标识符：用于事务处理配对。在响应中，Modbus 服务器复制请求的事务处理标识符。

协议标识符：用于系统内的多路复用。通过值 0 识别 Modbus 协议。

长度：长度域是接续域的字节数，包括单元标识符和数据域。

单元标识符：此域用于系统内路选择。典型地用于通过以太网 TCP-IP 网络和 Modbus 串行链路之间的网关对 Modbus 从站的通信。Modbus 客户机在请求中设置这个域，服务器必须在响应中用相同的值返回这个域。

通过注册的 502 端口上的 TCP 接收所有 Modbus/TCP ADU。

注：用最高有效字节在低地址存储的方式编码不同域。

### 5.2 Modbus 功能码描述

在 GB/Z 19582.1—2004 中详细说明了 Modbus 应用层协议上使用的标准功能码。

## 6 功能描述

本标准提供的 Modbus 组件结构是一个既包含 Modbus 客户机又包含 Modbus 服务器组件的通用模型，适用于任何设备。

有些设备可能仅提供服务器或客户机组件。

6.1 给出一个有关 Modbus 报文传输服务组件结构的简要概述，并且对结构模型内每一个组件进行描述。

### 6.1 Modbus 组件结构模型

见图 5～图 7 和表 2。

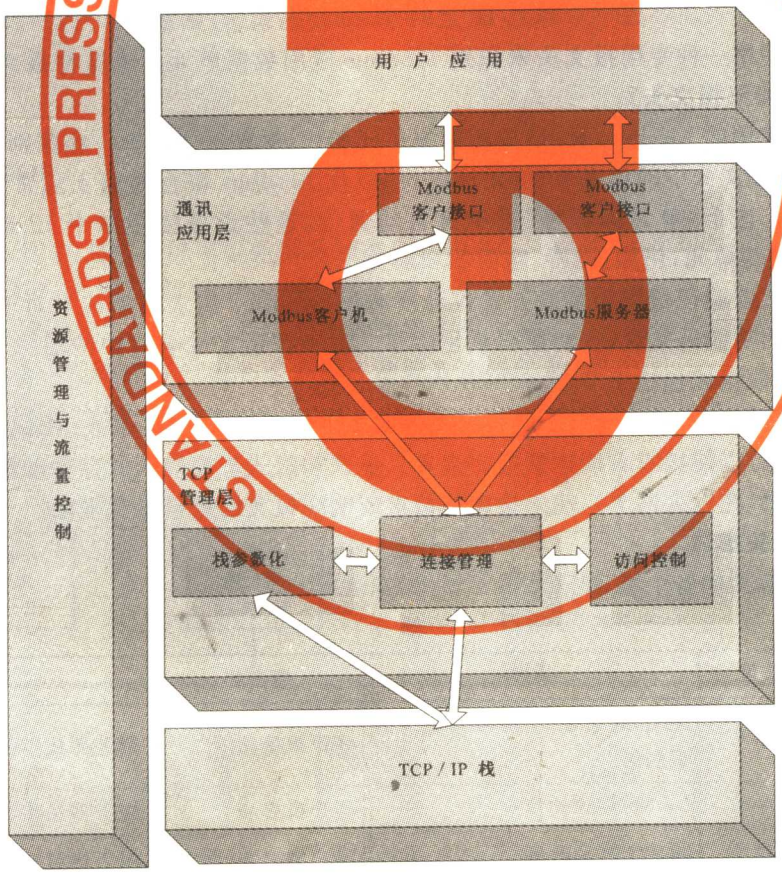



图 5 Modbus 报文传输服务概念结构

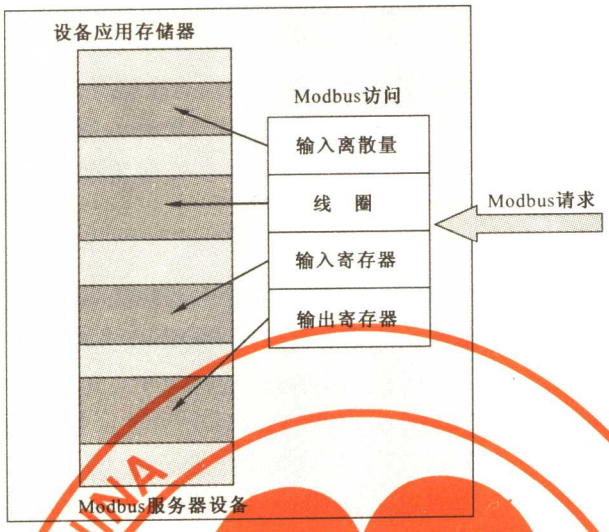



图 6　独立数据块的 Modbus 数据模型

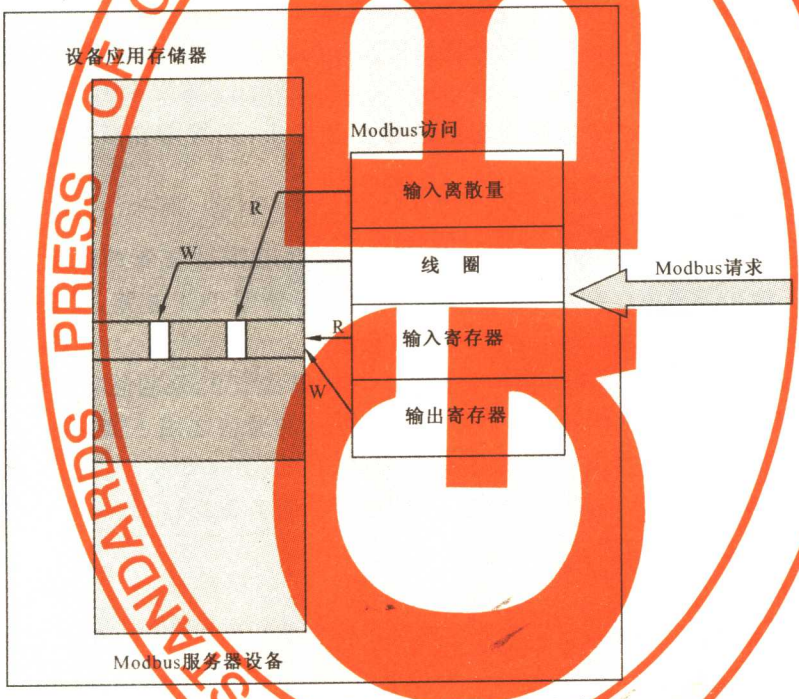



图 7　单个数据块的 Modbus 数据模型

表 2

| 基本表 | 对象类型 | 访问类型 | 注　　释 |
|---|---|---|---|
| 离散量输入 | 单个位 | 只读 | I/O 系统可提供这种类型数据 |
| 线圈 | 单个位 | 读写 | 通过应用程序可改变这种类型数据 |
| 输入寄存器 | 16 位字 | 只读 | I/O 系统可提供这种类型数据 |
| 保持寄存器 | 16 位字 | 读写 | 通过应用程序可改变这种类型数据 |

### 6.1.1　通信应用层

一个 Modbus 设备可以提供一个客户机和/或服务器 Modbus 接口。

可提供一个 Modbus 后端接口，间接地允许对用户应用对象的访问。

此接口由四个区域组成：离散量输入、离散量输出(线圈)、输入寄存器和输出寄存器。必须进行这

个接口与用户应用数据之间的映射(本地实现)。

——Modbus 客户机

Modbus 客户机允许用户应用显式控制与远程设备的信息交换。Modbus 客户机根据用户应用向 Modbus 客户机接口发送的要求中所包含的参数来建立一个 Modbus 请求。

Modbus 客户机使用一个 Modbus 的事务处理,该事务处理管理包括对 Modbus 证实的等待和处理。

——Modbus 客户机接口

Modbus 客户机接口提供一个接口,使得用户应用能够生成各类 Modbus 服务的请求,该服务包括对 Modbus 应用对象的访问。在本规范中没有详细地描述 Modbus 客户机接口(API),仅给出实现模型中的实例。

——Modbus 服务器

在收到一个 Modbus 请求以后,模块激活一个本地操作进行读、写、或完成其他操作。这些操作的处理对应用程序开发人员来说都是透明的。Modbus 服务器的主要功能是等待来自 TCP502 口的 Modbus 请求,处理这一请求,然后根据设备的状况生成一个 Modbus 应答。

——Modbus 后端接口

Modbus 后端接口是一个从 Modbus 服务器到定义应用对象的用户应用之间的接口。

### 6.1.2 TCP 管理层

报文传输服务的主要功能之一是管理通信的建立和结束,及管理在所建立的 TCP 连接上的数据流。

——连接管理

在客户机和服务器的 Modbus 模块之间的通信需要使用 TCP 连接管理模块。它负责全面管理报文传输 TCP 连接。连接管理中存在两种可能:用户应用自身管理 TCP 连接,或全部由这个模块进行连接管理,因此对用户应用是透明的。后一种方案灵活性较差。

TCP 502 端口的监听是为 Modbus 通信保留的。在缺省状态下,强制监听这个端口。然而,有些市场上的产品或应用可能需要其他端口作为 TCP 上 Modbus 的通信之用。例如:楼宇控制。为此,特别建议:客户机和服务器均应向用户提供对 TCP 端口号进行 Modbus 参数配置的可能性。重要的是:即使在某一个特定的应用中为 Modbus 服务配置了其他 TCP 服务器端口,除一些特定应用端口外,TCP 服务器 502 端口必须仍然是可用的。

——访问控制模块

在某些至关重要的场合,必须禁止无关的主机对设备内部数据的访问。这既是需要的安全模式,也是在需要时实现安全处理的原因。

### 6.1.3 TCP/IP 栈层

可以对 TCP/IP 的栈进行参数配置,以适用对产品或系统的不同的特定约束进行数据流控制、地址管理和连接管理。一般说来,BSD 套接字接口被用来管理 TCP 连接。

### 6.1.4 资源管理和数据流控制

为了平衡 Modbus 客户机与服务器之间进出报文传输的数据流,在 Modbus 报文传输栈的所有各层均设置了数据流控制机制。资源管理和数据流控制模块首先是基于 TCP 内部数据流控制,加上数据链路层的某些数据流控制,以及用户应用层的数据流控制。

## 6.2 TCP 连接管理

### 6.2.1 连接管理模块

#### 6.2.1.1 总体描述

Modbus 通信需要建立客户机与服务器之间的 TCP 连接,TCP 连接管理操作见图 8。

连接的建立可以由用户应用模块显式激活,也可以由 TCP 连接管理模块自动激活。

在第一种情况下，用户应用模块必须提供应用程序接口，以便完全管理连接。这种方式为应用开发人员提供了灵活性，但需要 TCP/IP 机制方面的专长。

在第二种情况下，对仅发送和接收 Modbus 报文的用户应用来说，TCP 连接管理是完全隐含的。TCP 连接管理模块负责在需要时建立新的 TCP 连接。

TCP 客户机和服务器连接数量的定义不属于本部分的范围（在本标准中采用 $n$）。根据设备能力，TCP 连接的数量可能不同。

实现规则：

a) 如果没有显式用户需求，建议采用自动的 TCP 连接管理。

b) 建议：保持与远程设备的连接，而不要在每次 Modbus/TCP 事务处理时打开和关闭连接。

注：Modbus 客户必须能够接收来自服务器的关闭请求，并关闭连接。当需要时，连接可以被重新打开。

c) 建议：一个 Modbus 客户机要打开与远程 Modbus 服务器的最低限度的 TCP 连接（同一 IP 地址）。最好的选择是一个应用建立一个连接。

d) 几个 Modbus 事务处理可以在同一个 TCP 连接上被同时激活。

注：如果以此方式，Modbus 事务处理标识符必须被用作唯一地识别请求与响应的匹配。

e) 在两个远程 Modbus 设备（一个客户机和一个服务器）之间双向通信的情况下，有必要为客户机数据流和服务器数据流分别建立连接。

f) 一个 TCP 帧只能传送一个 Modbus ADU。建议：不要在同一个 TCP PDU 中发送多个请求或应答。

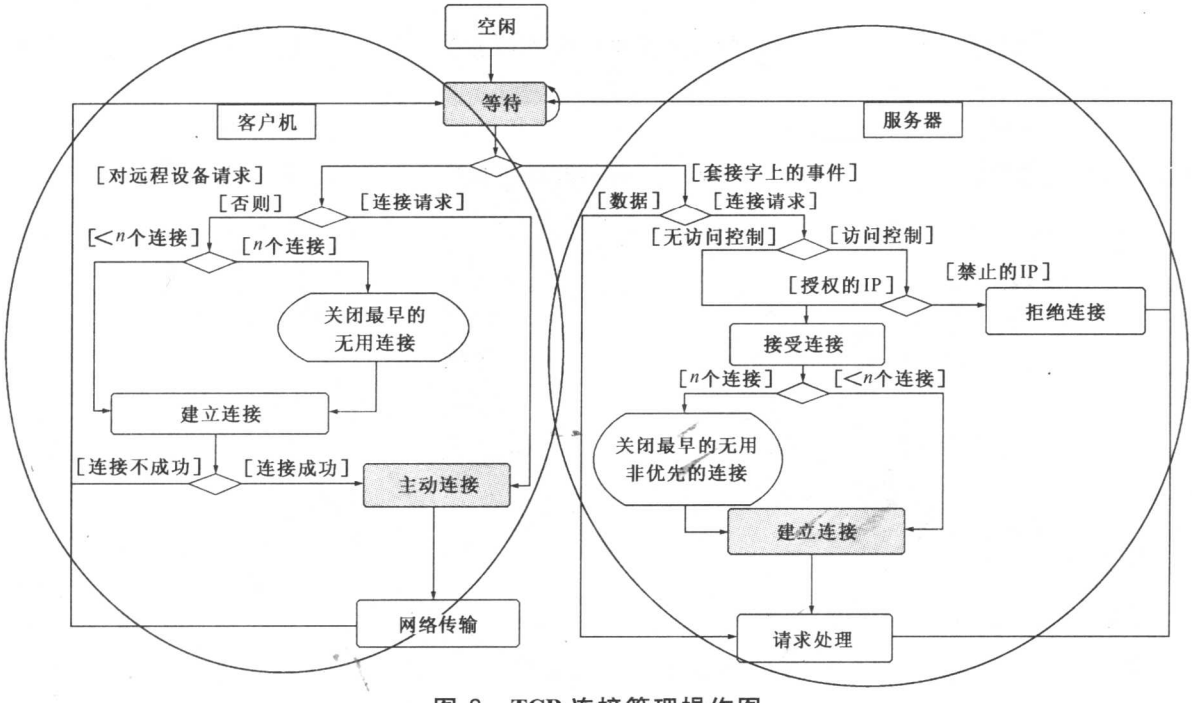



**图 8　TCP 连接管理操作图**

——显式的 TCP 连接管理

用户应用模块负责管理所有的 TCP 连接：主动的和被动的建立及结束连接……。对客户机与服务器间所有的连接进行这种管理。BSD 套接字接口用在用户应用模块中来管理 TCP 连接。这种方案提供了完全的灵活性，但也意味着应用开发人员要具备充分的有关 TCP 的知识。

考虑到设备的能力和需求，必须限定客户机与服务器间连接的配置数量。

——自动 TCP 连接管理

TCP 连接管理对用户应用模块是完全透明的。连接管理模块可以接受足够数量的客户机/服务器

连接。在超过所授权数量的连接时，必须有一种实现机制。在这种情况下建议：关闭最早建立的不使用的连接。

当收到第一个来自远程客户机或本地用户应用的数据包时，就建立了与远程对象的连接。如果网络提出终止或本地设备决定终止，此连接将被关闭。在接收连接请求时，访问控制选项可用来禁止未授权客户访问设备的可能性。

TCP 连接管理模块采用栈接口（通常 BSD 套接字接口）来与 TCP/IP 栈进行通信。

为了保持系统需求与服务器资源之间的兼容，TCP 管理将保持两个连接库。

——第一个库（优先连接库）由那些从不被本地主动关闭的连接组成。必须提供一个配置来建立这个库。实现的原理是将这个库的每一个可能的连接与一个特定的 IP 地址联系起来。具有这个 IP 地址的设备被称为“标记的”。任何一个被“标记的”设备的新的连接请求必须被接收，并从优先连接库中取出。还有必要设置允许每个远程设备最多建立连接的数量，以避免同一设备使用优先连接库中所有的连接。

——第二个库（非优先连接库）包括了非标记设备的连接。这里采用的规则是：当有来自非标记设备的新的连接请求，以及库中没有连接可用时，关闭早些时候建立的连接。

可有选择地提供一个配置来分配每个库中可用连接的数量。然而（非强制性的），如果需要，设计人员可在设计期间设置连接的数量。

#### 6.2.1.2 连接管理描述

——连接建立

见图 9。

Modbus 报文传输服务必须在 502 端口上提供一个监听套接字，允许接收新的连接和与其他设备交换数据。

当报文传输服务需要与远程服务器交换数据时，它必须与远程 502 端口建立一个新的客户机连接，以便于远距离地交换数据。本地端口必须高于 1024，并且对每个客户机的连接各不相同。


**图 9 Modbus TCP/IP 连接建立**

如果客户机与服务器的连接数量大于授权的连接数量，则最早建立的无用的连接被关闭。激活访问控制机制用来检查远程客户机的 IP 地址是否是经过授权的。如果未经授权，将拒绝新的连接。

——Modbus 数据传送

一个 Modbus 请求必须在已经被打开的正确的 TCP 连接上发送。远程设备的 IP 地址用于寻找所建立的 TCP 连接。在与同一个远程设备建立多个连接时，必须选择其中一个连接用于发送 Modbus 报文，可以采取不同的选择策略，例如：最早的连接、第一个连接。在 Modbus 通信的全过程中，连接必须始终保持打开。一个客户机可以向一个服务器启动多个事务处理，而不必等待先前的事物处理结束。

——连接关闭

当客户机与服务器间的 Modbus 通信结束时，客户机必须关闭用于通信的连接。

#### 6.2.2 操作模式对 TCP 连接的影响

某些操作模式(两操作端点之间通信断开、一个端点的故障和重新启动……)会对 TCP 连接产生影响。在没有接收到来自另一侧的确认时,一个连接可被视为关闭或异常终止,称这种连接为半打开的连接。

本章描述各种主要操作模式的特性。这种描述基于在连接的两端采用了保持连接 TCP 机制的假设(见 6.3.2)。

##### 6.2.2.1 两操作端之间通信断开

通信断开的原因可以是服务器侧以太网连接电缆断开。预期的特性是:

——如果在连接上没有正在发送数据包:

如果通信断开持续的时间短于"保持连接"计时器的值,将察觉不到通信断开。如果通信断开时间超过"保持连接"计时器的值,将一个错误返回到 TCP 连接层,由其复位连接。

——如果在连接断开的前后发送一些数据包:

TCP 重新传输算法(Jacobson 算法、Karn 算法以及指数补偿算法,参见 6.3.2)被激活。这可以导致在"保持连接"计时器终止之前由栈的 TCP 层复位连接。

##### 6.2.2.2 服务器端的故障和重新启动

在服务器故障和重新启动以后,客户机端处于"半打开"连接状态。预期的特性是:

——如果在半打开的连接上没有发送数据包:

只要"保持连接"计时器还在计时中,从客户机端看,连接是半打开的。之后,将返回一个错误到 TCP 管理层,由其复位连接。

——如果在半打开的连接上发送一些数据包:

服务器在不再存在的连接上接收数据。栈的 TCP 层发送一个复位指令来关闭客户机端的半打开的连接。

##### 6.2.2.3 客户机端的故障和重新启动

在客户机故障和重新启动以后,服务器侧处于"半打开"连接状态。预期的特性是:

——如果在半打开的连接上没有发送数据包:

只要保持连接计时器还在计时中,从服务器端看,这种 TCP 半打开连接被认为是打开的。之后,将返回一个错误到该 TCP 管理层,由其复位连接。

——如果在保持连接计时器完成计时前,客户机打开一个新的连接:

必须分析两种情况:

1) 所打开的连接与服务器侧半打开的连接具有相同的特性(相同的源和目的端口、相同的源和目的 IP 地址),所以,在连接建立超时后(伯克利实现的多数情况下为 75 ms),TCP 栈层将不能打开连接。为了避免在较长超时时间内不能进行通信,建议:在客户机端重新启动后,确保使用与原有连接不同的源端口号建立连接。

2) 所打开的连接与服务器侧半打开的连接具有不同的特性(不同的源端口和相同的目的端口、相同的源和目的 IP 地址),所以,在 TCP 栈层上打开连接,并向服务器侧的 TCP 管理层发送信号。

如果服务器侧 TCP 管理层仅支持一个远程客户机 IP 地址的连接,那么可以关闭原来的半打开的连接,使用新的连接。

如果服务器侧 TCP 管理层支持多个远程客户机 IP 地址的连接,那么新的连接保持打开状态,原来的连接也保持半打开状态,直到"保持连接"计时器计时结束,此时,将返回一个错误到 TCP 管理层。之后,TCP 管理层将能够复位原有的连接。

#### 6.2.3 访问控制模块

这个模块的目的是检查每一个新的连接,对照一个合法授权的远程 IP 地址表,它可以授权或禁止一个远程客户机的 TCP 连接。

在至关重要的场合，应用开发人员需要选择访问控制模块来保证网络的访问。在这种情况下，需要对每个远程 IP 授权或禁止访问。用户需提供一个 IP 地址的表，并特别注明每个 IP 地址是否合法授权。在缺省情况下，在安全模式中，用户未配置的 IP 地址均被禁止。所以，借助于访问控制模式，关闭来自未知的 IP 地址的访问连接。

### 6.3 TCP/IP 栈的使用

见图 10。

TCP/IP 栈提供了一个接口，用来管理连接、发送和接收数据，还可以进行某些参数配置，以使得栈的特性适应于设备或系统的限制。

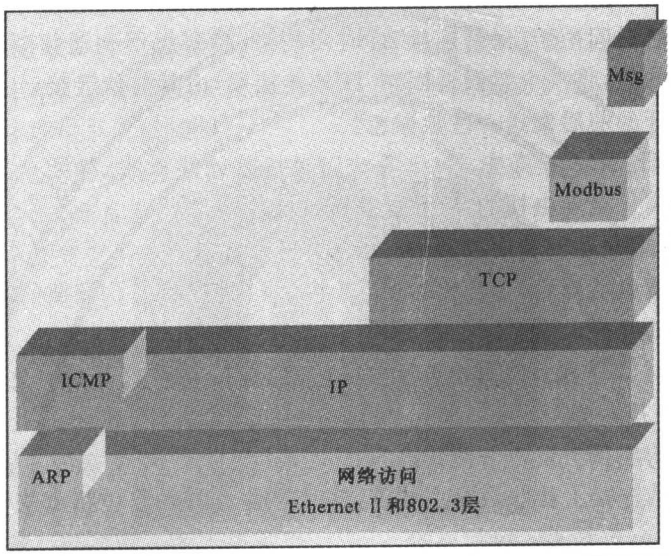



**图 10 Modbus TCP/IP 通讯栈**

本章的目的是给出有关栈接口的综述，以及一些与栈的参数配置有关的信息。综述的主要内容是 Modbus 报文传输所使用的一些特性。

有关更多的信息，建议阅读 RFC 1122，这个 RFC 1122 为厂商和开发商提供了互联网通信软件的指南。RFC 1122 详述了一个连接到互联网的主机必须采用的标准协议，以及一组明确的需求和选项。

栈接口一般是基于本部分中描述的 BSD 接口。

#### 6.3.1 BSD 套接字接口的应用

注：有些 TCP/IP 栈从性能考虑提出其他类型的接口。Modbus 客户机或服务器可以使用这些特定的接口，但是在本标准中对这种使用不做描述。

一个套接字是一个通信端点，它是通信中的基本构成块。通过套接字发送和接收数据来执行一个 Modbus 通信。TCP/IP 库仅提供了使用 TCP 和提供基于连接的通信服务的流套接字。

socket()函数用来创建套接字。返回的一个套接字号被创建者用来访问该套接字。套接字创建时没有地址(IP 地址和端口号)。直到一个端口被绑定到该套接字时，方可接收数据。

bind()函数用来绑定一个端口号到套接字。bind()函数在套接字与所指定的端口号之间建立一种联系。

为了初始化一个连接，客户机必须发送 connect()函数来指定套接字号、远程 IP 地址和远程监听端口号(主动连接建立)。

为了完成连接，服务器必须发送 accept()函数来指定先前在 listen()调用中所指定的套接字号(被动连接建立)。一个新的套接字被创建，并具有与初始套接字相同的特性。这个新的套接字连接到客户机的套接字，而将套接字号返回到服务器。于是，释放初始套接字，以便为其他欲与该服务器连接的客户机使用。

在 TCP 连接建立以后，数据即可被传送。将 send()和 recv()函数专门地设计成与已经连接的套接字一起使用。

setsockopt()函数允许套接字的创建者将套接字与选项联合使用。这些选项修改了套接字的操作特征。在 6.3.2 给出这些选项的描述。

select()函数允许编程人员测试所有套接字上的事件。

shutdown()函数允许套接字的使用者来终止 send()和/或 recv()。

一旦不再需要套接字，通过使用 close()函数来放弃套接字的描述符。

图 11 给出了客户机与服务器间的完整的 Modbus 通信过程。客户机建立一个连接，向服务器发送 3 个 Modbus 请求，而不等待第一个请求的响应到来。在收到所有的响应后，客户机正常地关闭连接。

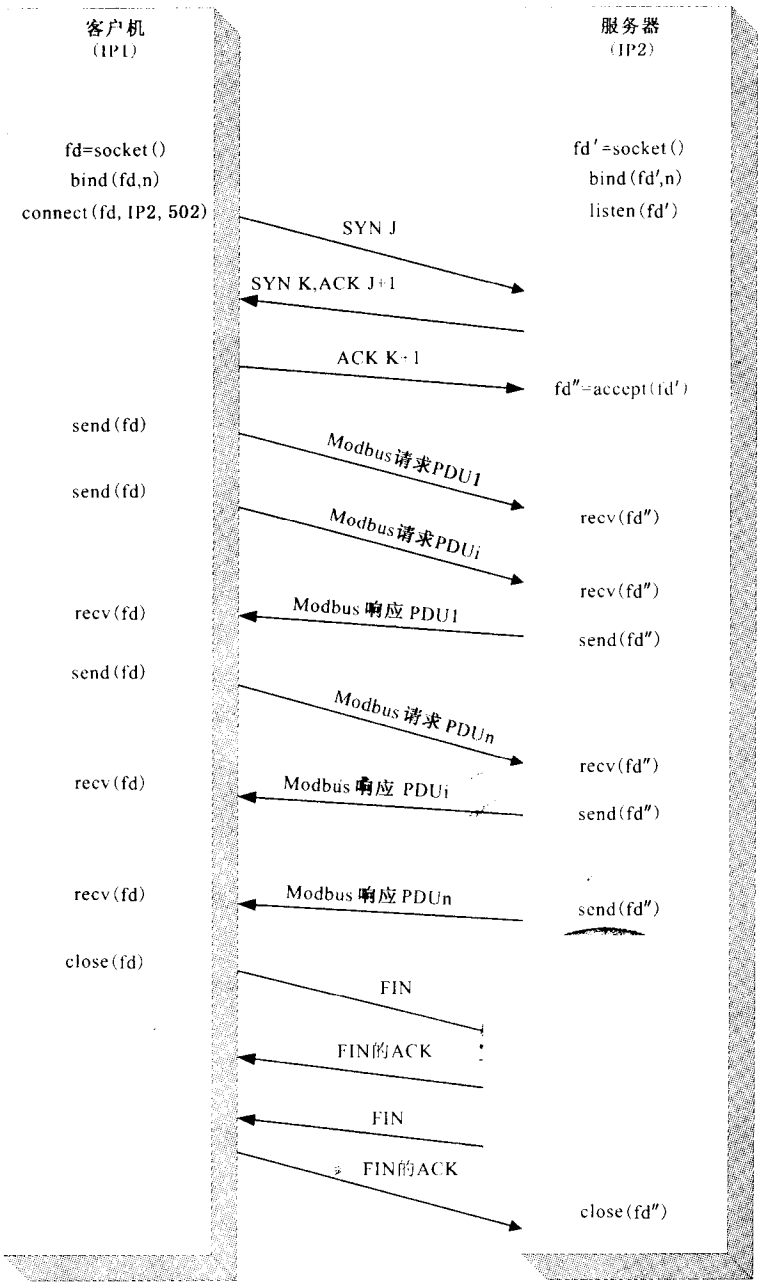



**图 11 Modbus 信息交换**

### 6.3.2 TCP 层参数配置

可以调整 TCP/IP 栈的一些参数以使得其特性适应产品或系统的限制。TCP 层的下列参数可以进行调整：

——每个连接的参数

SO-RCVBUF，SO-SNDBUF：

这些参数允许为发送和接收用套接字接口设定高限位。可以通过调整这些参数来实现流量控制管理。接收缓冲区的大小是每个连接通告窗口的最大值。为了提高性能，必须增加套接字缓冲区的大小。否则，这些值必须小于内部驱动器的资源，以便在内部驱动器的资源耗尽之前关闭 TCP 窗口。

接收缓冲区大小取决于 TCP 窗口大小、TCP 最大段的大小和接收输入帧所需的时间。由于最大段的大小为 300 个字节(一个 Modbus 请求需要最大 256 字节＋MBAP 报文头)，如果需要 3 个帧进行缓存，可将套接字缓冲区大小调整为 900 字节。为了满足最大需求和最好的预定时间，可以增加 TCP 窗口的大小。

TCP-NODELAY：

通常，小报文包在局域网(LAN)上的传输不会产生问题，因为多数局域网是不拥堵的，但是，这些小报文包在广域网上将会造成拥堵。一个称为“NAGLE 算法”的简单方案是：收集少量的数据，当前面报文的 TCP 确认到达时再用单个段进行发送。

为了获得更好的实时特性，建议：将少量的数据直接发送，而不要试图将其收集到一个段内再发送。这就是为什么建议强制 TCP-NODELAY 选项，这个选项禁止在客户机和服务器连接上采用“NAGLE 算法”。

SO-REUSEADDR：

当 Modbus 服务器关闭一个由远程客户启动的 TCP 连接时，在这个连接处于“时间等待”状态(2 个 MSL：最大段寿命)的过程中，该连接所用的本地端口号不能被再次用来打开一个新的连接。

建议：为每个客户机和服务器连接规定该 SO-REUSEADDR 选项，以旁路这个限制。此选项允许该进程为自身分配一个端口号，该端口号是在 2 个 MSL 期间内等待客户机并监听套接字的连接的一部分。

SO-KEEPALIVE：

在 TCP/IP 协议缺省状态下，没有数据通过空闲的 TCP 连接发送。因此，如果在 TCP 连接端上没有进程发送数据，在 2 个 TCP 模块间就不交换任何数据。这就是假设客户机应用或服务器应用均采用定时器来检测连接的非激活性，以便关闭连接。

建议：在客户机与服务器连接两端均采用 KEEPALIVE 选项，以便轮询另一端来得知对方是否故障并死机，或故障并重新启动。

然而，必须注意，采用 KEEPALIVE 可能引起一个非常良好的连接在瞬间故障时通信中断，如果保持连接的定时器定时的时间太短，将占用不必要的网络带宽。

——整个 TCP 层的参数

建立 TCP 连接超时：

多数伯克利推出的系统将建立新连接的时限设定为 75 s，这个缺省值应该适应于实时的应用限制。

保持连接参数：

连接的缺省空闲时间是 2 h。超过此空闲时间将触发一个保持连接试探过程。第一个保持连接试探后，在最大次数内每隔 75 s 发送一个试探，直到收到对试探的响应为止。

在一个空闲连接上发出保持连接试探的最大数是 8 次。如果发出最大试探次数之后而没有收到应答，TCP 向应用发出一个错误信号，由应用来决定关闭连接。

超时与重发参数：

如果检测到一个 TCP 报文包丢失，将重发此报文包。检测丢失的方法之一是管理重发超时

(RTO),如果没有收到来自远程端的确认,超时终止。

TCP 进行 RTO 的动态评估。为此,在发送每个非重发的报文包后测量往返时间(RTT)。往返时间(RTT)是指报文包到达远程设备并从远程设备获得一个确认所用的时间。一个连接的往返时间是动态计算的,然而,如果 TCP 不能在 3 s 内获得 RTT 的估算,那么,就设定 RTT 的缺省值为 3 s。

如果已经估算出 RTO,它将被用于下一个报文包的发送。如果在估算的 RTO 终止之前没有收到该报文包的确认,启用指数补偿算法。在一个特定的时间段内,允许相同报文包最大次数的重发。之后,如果收不到确认,连接终止。

可以对某些栈设置连接终止之前重发的最大次数和重发的最长时间。

在 TCP 标准中定义了一些重发算法:

——Jacobson RTO 估算算法用来估算重发超时(RTO);

——Karn 算法指出,在重发段,不应进行 RTO 估算;

——指数补偿算法定义:对于时间上限为 64 s 的每一次重发,加倍重发超时;

——快速重发算法允许在收到 3 个重复确认之后进行重发。考虑这个算法是因为:在 LAN 上,可能会导致报文丢失的检测快于等待 RTO 终止的检测。

在 Modbus 实现中,推荐使用这些算法。

### 6.3.3 IP 层的参数配置

#### 6.3.3.1 IP 参数

下列参数必须在 Modbus 实现的 IP 层进行配置:

——本地 IP 地址:IP 地址可以是 A、B 或 C 类中的一种。

——子网掩码:可基于各种原因,将 IP 网络划分成子网:使用不同的物理介质(例如:以太网、广域网等)、更有效地使用网络地址、以及控制网络流量的能力。子网掩码必须与本地 IP 地址的类型相一致。

——缺省网关:缺省网关的 IP 地址必须与本地 IP 地址在同一子网内。禁止使用 0.0.0.0 的值。如果没有定义网关,那么此值可设为 127.0.0.1 或本地 IP 地址。

注:Modbus 报文传输服务在 IP 层上不要求分段功能。

应该利用本地 IP 地址、子网掩码和省缺网关(不同于 0.0.0.0)配置本地 IP 端点。

## 6.4 通信应用层

### 6.4.1 Modbus 客户机

见图 12。

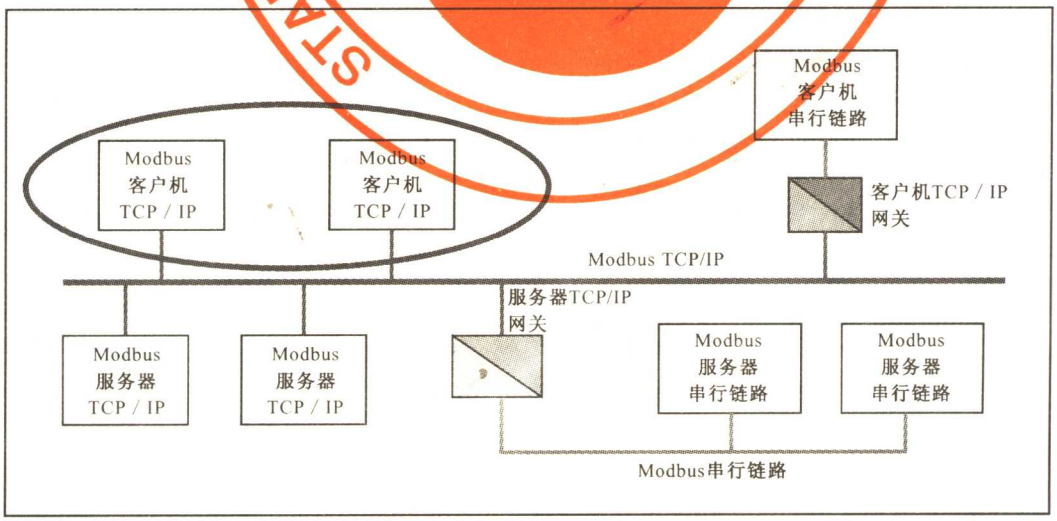



图 12 Modbus 客户机单元