计算机网络协议交互实验指导书

1　IPv4 协议交互实验

1.1　实验目的

IPv4 协议是互联网的核心协议，它保证了网络节点（包括网络设备和主机）在网络层能够按照标准协议互相通信。IPv4 地址唯一标识了网络节点。在我们日常使用的计算机的主机协议栈中，IPv4 协议必不可少，它能够接收网络中传送给本机的分组，同时也能根据上层协议的要求将报文封装为 IPv4 分组发送出去。

本实验通过设计实现主机协议栈中的 IPv4 协议，让学生深入了解网络层协议的基本原理，学习 IPv4 协议基本的分组接收和发送流程。

1.2　实验要求

根据计算机网络实验系统所提供的上下文，对给出的 IP 报文进行字段检查，如果出错指出其出错字段或组装一个正确的 IP 报头。

1.3　实验内容

1）　实现 IPv4 分组的基本接收处理功能

对于接收到的 IPv4 分组，检查目的地址是否为本地地址，并检查 IPv4 分组头部中其它字段的合法性。在选项中选择对应的项。

2）　实现 IPv4 分组的封装发送

根据题干中给出的上下文环境，封装 IPv4 分组，使用系统提供的发送 IP 报文界面将分组发送出去。

1.4　实验帮助

在使用系统提供的发送 IP 报文界面（图 1.1 发送 IP 报文界面）时，校验和字段的初始值为零。如要计算正确的校验和则需使用 Valid Checksum 选项（图中用红色椭圆标注）。在打开该选项后，系统会自动计算 IP 头部的校验和（即时反应 IP 头部各个字段值的变化）。如需观察当前填充值下 IP 头部对应的十六进制和字符串表示，可点击 Decode 按钮，结果显示在图中蓝色方框标示的区域中。

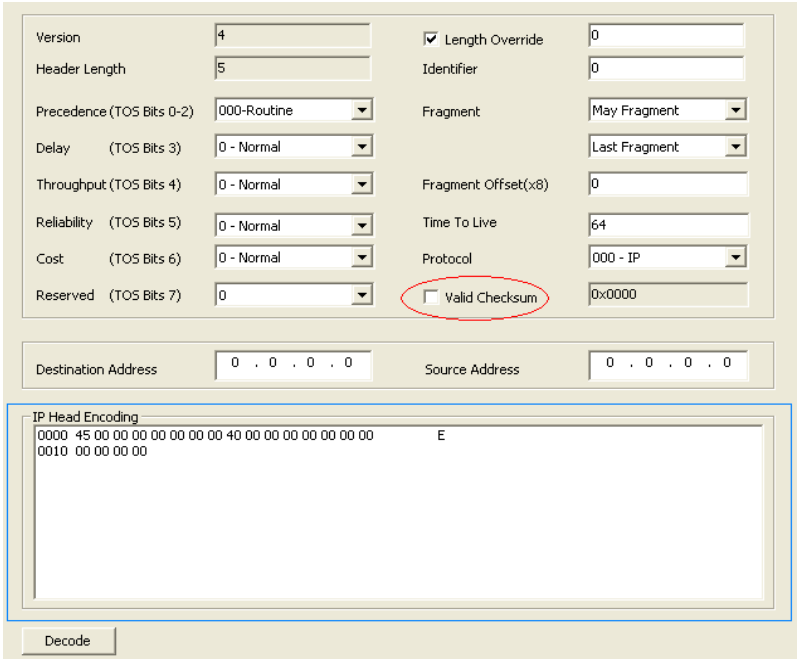

图 1.1 发送 IP 报文界面

当在检查 IP 报文时，需通过使用 Valid Checksum 选项观察校验和显示是否发生变化。如发生变化则说明该 IP 报文的校验和有错误。校验和的原始值可在蓝色框中的十六进制显示中的相应字段找到。

2 RIP 协议交互实验

2.1 实验目的

通过简单实现路由协议 RIP，深入理解计算机网络中的核心技术——路由技术，并了解计算机网络的路由转发原理。

2.2 实验要求

充分理解 RIP 协议，根据 RIP 协议的流程设计 RIP 协议的报文处理和超时处理函数。能够实现如下功能：

1） RIP 报文有效性检查
2） 处理 Request 报文
3） 处理 Response 报文
4） 路由表项超时删除
5） 路由表项定时发送

2.3 实验内容

1) 对客户端接收到的 RIP 报文进行有效性检查
对客户端接收到的 RIP 协议报文进行合法性检查，如果出错，指出错误原因；
2) 处理 Request 报文
正确解析并处理 RIP 协议的 Request 报文，并能够根据报文的内容以及本地路由表组成相应的 Response 报文，回复给 Request 报文的发送者，并实现水平分割；
3) 处理 Response 报文
正确解析并处理 RIP 协议的 Response 报文，并根据报文中携带的路由信息更新本地路由表；
4) 路由表项超时删除
处理来自系统的路由表项超时消息，并能够删除指定的路由；
5) 路由表项定时发送
实现定时对本地的路由进行广播的功能，并实现水平分割。

2.4 实验帮助

2.4.1 RIP 协议介绍

2.4.1.1 协议简述

RIP 协议采用的是距离-向量路由算法，该算法早在 Internet 的前身 ARPANET 网络中就已经被广泛采用。在 70 年代中期，Xerox 公司根据它们对互联网的研究成果提出了一套被称为 XNS（Xerox Network System）的网络协议软件。这套协议软件包含了 XNS RIP 协议，该协议就是现在所使用的 RIP 协议的最早原型。80 年代加州大学伯克利分校在开发 Unix 系统的同时，在 routed 程序中设计实现了 RIP 协议软件。routed 程序被绑定在 BSD Unix 系统中一起推出，被广泛的应用于早期网络中的机器之间交换路由信息。尽管 RIP/routed 没有非常突出的优点，但是由于 Unix 操作系统的普及，RIP/routed 也逐渐被推广出来，为许多人所接受，成为了中小型网络中最基本的路由协议/程序。

RIP 协议的 RFC 文本在 1988 年 6 月被正式推出，它综合了实际应用中许多实现版本的特点，同时为版本的兼容互通性提供了可靠的依据。由于 RIPv1 中存在着一些缺陷，再加上网络技术的发展，有必要对 RIP 版本进行相应的改进。1994 年 11 月，RFC1723 对 RIPv1 的报文结构进行了扩展，增加一些新的网络功能。1998 年 11 月，RIPv2 的标准 RFC 文本被正式提出，它在协议报文的路由表项中增加了子网掩码信息，同时增加了安全认证、不同路由协议交互等功能。

随着 OSPF、IS-IS 等域内路由协议的出现，许多人认为 RIP 协议软件已经过时。尽管 RIP 在协议性能和网络适应能力上远远落后于后来提出的路由协议，但是 RIP 仍然具有自身的特点。首先，在小型的网络环境中，从使用的网络带宽以及协议配置和管理复杂程度上看，RIP 的运行开销很小；其次，与其他路由协议相比，RIP 使用简单的距离-向量算法，实现更容易；最后，由于历史的原因，RIP 的应用范围非常广，在未来的几年中仍然会使用在各种网络环境中。因此，在路由器的设计中，RIP 协议是不可缺少的路由协议之一，RIP 协议的实现效率高低对路由器系统的路由性能起着重要的作用。

2.4.1.2 RIPv2 协议的报文结构

图 2.1　RIPv2 的报文结构

RIPv2 的报文结构如图 2.1 所示。每个报文都包括一个报文命令字段、一个报文版本字段、一个路由域字段、一个地址类字段、一个路由标记字段以及一些路由信息项（一个 RIP 报文中最多允许 25 个路由信息项），其中每个字段后括号中的数字表示该字段所占的字节数。RIP 报文的最大长度为 4+20*25=504 字节，加上 UDP 报头的 8 字节，一共是 512 字节。如果路由表的路由表项数目大于 25 时，那么就需要多个 RIP 报文来完成路由信息的传播过程。下面对报文字段进行逐一介绍：

- **命令字段：**表示 RIP 报文的类型，目前 RIP 只支持两种报文类型，分别是请求报文（request 1）和响应（response 2）报文。
- **版本字段：**表示 RIP 报文的版本信息，RIPv2 报文中此字段为 2。
- **路由域字段：**是一个选路守护程序的标识符，它指出了这个数据报的所有者。在一个 Unix 实现中，它可以是选路守护程序的进程号。该域允许管理者在单个路由器上运行多个 RIP 实例，每个实例在一个选路域内运行。
- **地址类字段：**表示路由信息所属的地址族，目前 RIP 中规定此字段必须为 2，表示使用 IP 地址族。
- **IP 地址字段：**表示路由信息对应的目的地 IP 地址，可以是网络地址、子网地址以及主机地址。

- **子网掩码字段：**应用于 IP 地址产生非主机部分地址，为 0 时表示不包括子网掩码部分，使得 RIP 能够适应更多的环境。
- **下一站 IP 地址字段：**下一驿站，可以对使用多路由协议的网络环境下的路由进行优化。
- **度量值字段：**表示从本路由器到达目的地的距离，目前 RIP 将路由路径上经过的路由器数作为距离度量值。

一般来说，RIP 发送的请求报文和响应报文都符合图 2.1 的报文结构格式，但是当需要发送请求对方路由器全部路由表信息的请求报文时，RIP 使用另一种报文结构，此报文结构中路由信息项的地址族标识符字段为 0，目的地址字段为 0，距离度量字段为 16。

2.4.1.3 RIP 协议的基本特点

协议规定，RIP 协议使用 UDP 的 520 端口进行路由信息的交互，交互的 RIP 信息报文主要是两种类型：请求（request）报文和响应（response）报文。请求报文用来向相邻运行 RIP 的路由器请求路由信息，响应报文用来发送本地路由器的路由信息。RIP 协议使用距离-向量路由算法，因此发送的路由信息可以用序偶<vector, distance>来表示，在实际报文中，vector 用路由的目的地址 address 表示，而 distance 用该路由的距离度量值 metric 表示，metric 值规定为从本机到达目的网络路径上经过的路由器数目，metric 的有效值为 1 到 16，其中 16 表示网络不可到达，可见 RIP 协议运行的网络规模是有限的。

当系统启动时，RIP 协议处理模块在所有 RIP 配置运行的接口处发出 request 报文，然后 RIP 协议就进入了循环等待状态，等待外部 RIP 协议报文（包括请求报文和响应报文）的到来；而接收到 request 报文的路由器则应当发出包含它们路由表信息的 response 报文。

当发出请求的路由器接收到一个 response 报文后，它会逐一处理收到的路由表项内容。如果报文中的表项为新的路由表项，那么就会向路由表加入该表项。如果该报文表项已经在路由表中存在，那么首先判断这个收到的路由更新信息是哪个路由器发送过来的。如果就是这个表项的源路由器（即当初发送相应路由信息从而导致这个路由表项的路由器），则无论该现有表项的距离度量值（metric）如何，都需要更新该表项；如果不是，那么只有当更新表项的 metric 值小于路由表中相应表项 metric 值时才需要替代原来的表项。

此外，为了保证路由的有效性，RIP 协议规定：每隔 30 秒，重新广播一次路由信息；若连续三次没有收到 RIP 广播的路由信息，则相应的路由信息失效。

2.4.1.4 水平分割

水平分割是一种避免路由环路的出现和加快路由汇聚的技术。由于路由器可能收到它自己发送的路由信息，而这种信息是无用的，**水平分割技术不反向通告任何从终端收到的路由更新信息，**而只通告那些不会由于计数到无穷而清除的路由。

2.4.2 实验拓扑

客户端软件模拟一个网络中的路由器，在其中 2 个接口运行 RIP 协议，接口编号为 1 和 2，每个接口均与其他路由器连接，通过 RIP 协议交互路由信息。

2.4.3 实验界面

界面上最多会有 4 个标签页:点击"Rip 报文展示"标签可以查看接收到的 RIP 报文;点击"路由表"标签可以查看/编辑本地路由表;点击"发送 RIP(接口 1)"标签可以封装 RIP 报文并从接口 1 发送;点击"发送 RIP(接口 2)"标签可以封装 RIP 报文并从接口 2 发送。其具体界面如下所示：

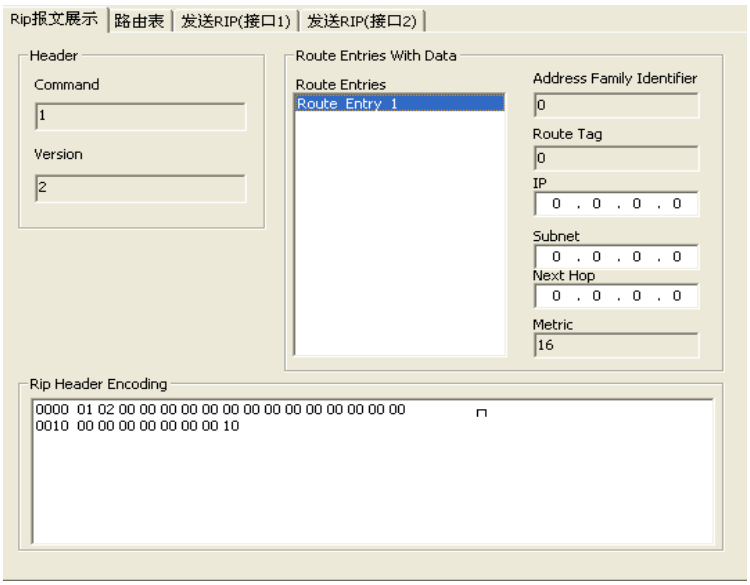

图 2.2　Rip 报文展示

图 2.3　路由表

如果想删除一条路由信息，可以在表格中选中该条路由，然后点击删除按钮，即可删除该条路由；如果想添加一条路由信息，可以先点击添加按钮，然后在表格下面的信息编辑区域填入相关的路有信息；如果想修改一条路由信息，可以先在表格中选中该条路由，然后在信息编辑区域编辑相关的信息。

可以使用发送 RIP(接口 1)/ 发送 RIP(接口 2)封装相应的 RIP 报文，并将其以相应的接口发送出去。如图 2.4 和图 2.5 所示：

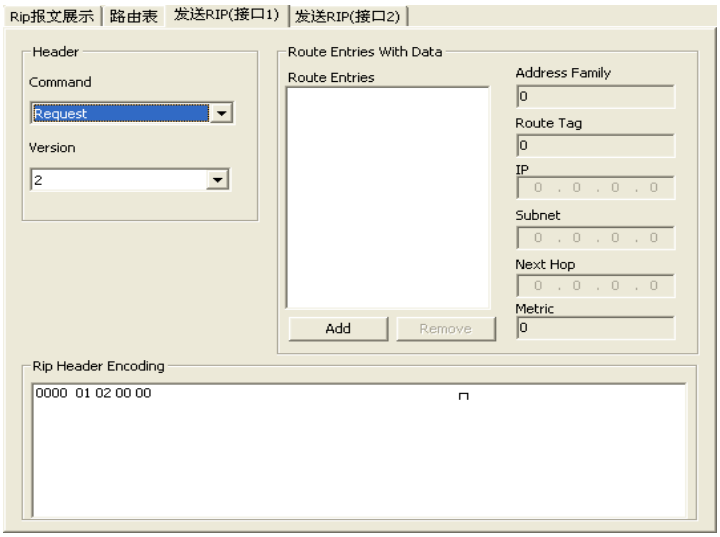

图 2.4　发送 RIP(接口 1)

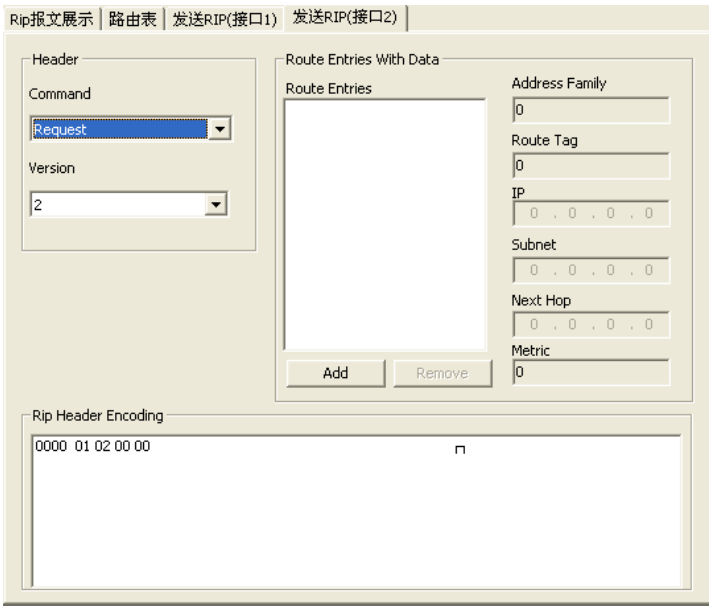

图 2.5　发送 RIP(接口 2)

3　IPv6 协议交互实验

3.1　实验目的

现有的互联网是在 IPv4 协议的基础上运行。IPv6 是下一版本的互联网协议，它的提出最初是因为随着互联网的迅速发展，IPv4 定义的有限地址空间将被耗尽，地址空间的不足必将影响互联网的进一步发展。为了扩大地址空间，拟通过 IPv6 重新定义地址空间。IPv4 采用 32 位地址长度，只有大约 43 亿个地址，估计在 2005～2010 年间将被分配完毕，而 IPv6 采用 128 位地址长度，几乎可以不受限制地提供地址。

本实验通过设计实现主机协议栈中的 IPv6 协议，让学生深入了解网络层协议的基本原理，学习 IPv6 协议基本的分组接收和发送流程。

3.2　实验要求

根据计算机网络实验系统所提供的上下文，对给出的 IP v6 报文进行字段检查，如果出

错指出其出错字段或组装一个正确的 IP v6 报头。

3.3 实验内容

1） 实现 IPv4 v6 分组的基本接收处理功能

对于接收到的 IP v6 分组，检查目的地址是否为本地地址，并检查 IP v6 分组头部中其它字段的合法性。在选项中选择对应的项。

2） 实现 IP v6 分组的封装发送

根据题干中给出的上下文环境，封装 IP v6 分组，使用系统提供的发送 IP v6 报文界面将分组发送出去。

3.4 实验帮助

界面上会显示接收到的 IPv6 报头(接收部分)或要填充的 IPv6 报头(发送部分),其内容展示分为字段意义显示和 16 进制显示,在字段意义显示部分可以看到接收到 IP v6 报头或要发送 IPv6 报头各个字段的数据值,在 16 进制显示部分会展示相应的 IPv6 报头的 16 进制显示。

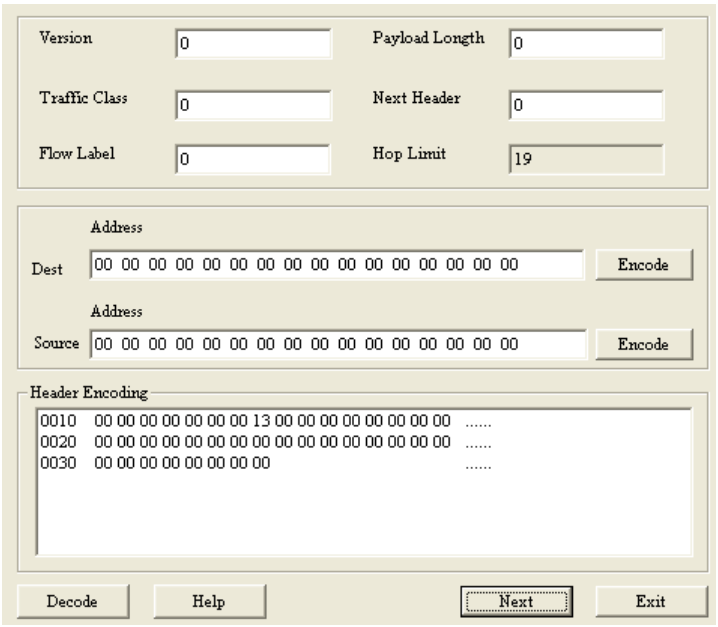

图 3.1 IPv6 报文发送界面

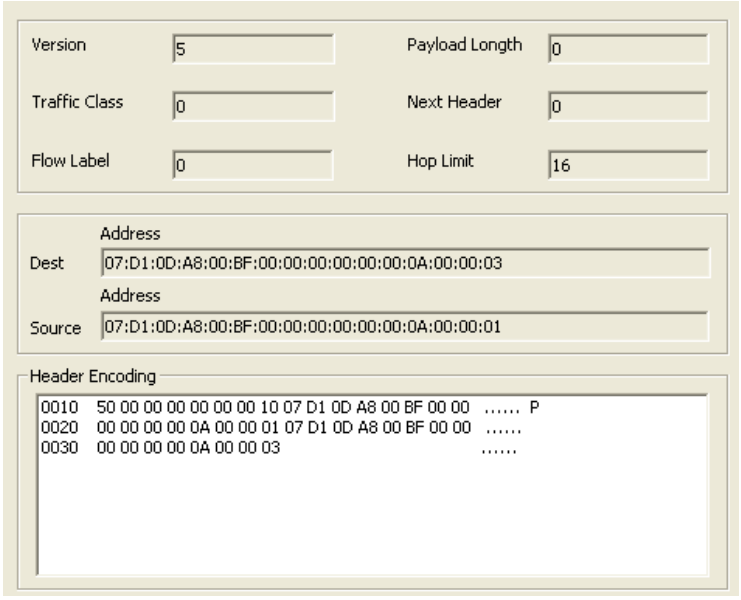

图 3.2 IPv6 报文接收界面

4 TCP 协议交互实验

4.1 实验目的

传输层是互联网协议栈的核心层次之一，它的任务是在源节点和目的节点间提供端到端的、高效的数据传输功能。TCP 协议是主要的传输层协议，它为两个任意处理速率的、使用不可靠 IP 连接的节点之间，提供了可靠的、具有流量控制和拥塞控制的、端到端的数据传输服务。TCP 协议不同于 IP 协议，它是有状态的，这也使其成为互联网协议栈中最复杂的协议之一。网络上多数的应用程序都是基于 TCP 协议的，如 HTTP、FTP 等。本实验的主要目的是学习和了解 TCP 协议的原理和设计实现的机制。

TCP 协议中的状态控制机制和拥塞控制算法是协议的核心部分。TCP 协议的复杂性主要源于它是一个有状态的协议，需要进行状态的维护和变迁。有限状态机可以很好的从逻辑上表示 TCP 协议的处理过程，理解和实现 TCP 协议状态机是本实验的重点内容。另外，由于在网络层不能保证分组顺序到达，因而在传输层要处理分组的乱序问题。只有在某个序号之前的所有分组都收到了，才能够将它们一起提交给应用层协议做进一步的处理。

本实验要求学生能够运用 TCP 报头正确的建立和拆除连接。

4.2 实验要求

根据计算机网络实验系统所提供的上下文，实现 TCP 建立连接和主动释放连接的过程。

4.3 实验内容

实验内容主要包括：

1) 实现 TCP 三次握手建立连接

   根据题目中给出上下文环境，封装 TCP 报文，与目标主机建立起 TCP 连接。

2) 实现 TCP 主动释放连接

   根据题目中给出的上下文环境，释放在上一步中建立起来的 TCP 连接。

4.4 实验帮助

界面上会显示接收到的 TCP 报头(接收部分)或要填充的 TCP 报头(发送部分),其内容展示分为字段意义显示和 16 进制显示,在字段意义显示部分可以看到接收到 TCP 报头或要发送 TCP 报头各个字段的数据值,在 16 进制显示部分会展示相应的 TCP 报头的 16 进制显示。其中校验和会自动计算，源端口和目的端口已经确定不能更改。

学生需要注意的是，由于建立连接和释放连接具有时序关系，如果在实验过程中有一步出错就会直接显示最终实验结果—错误，而不会在错误的基础上继续下一步。所以学生需要在出错后重新开始本实验。

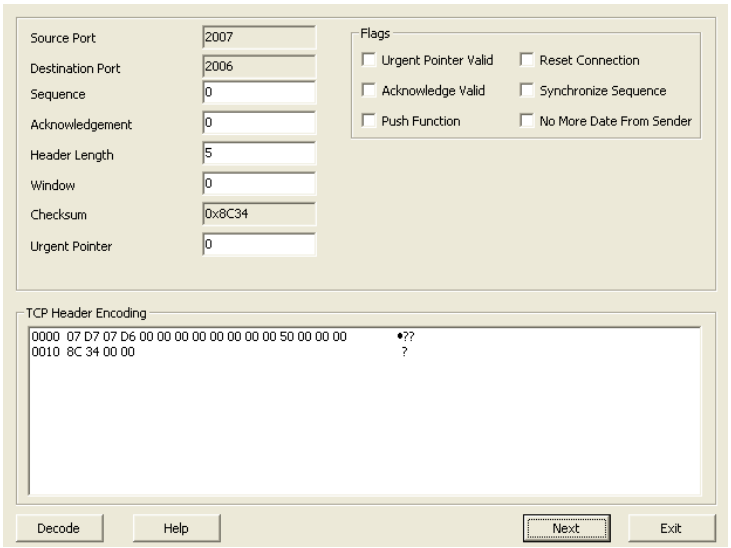

图 4.1　TCP 报文发送界面

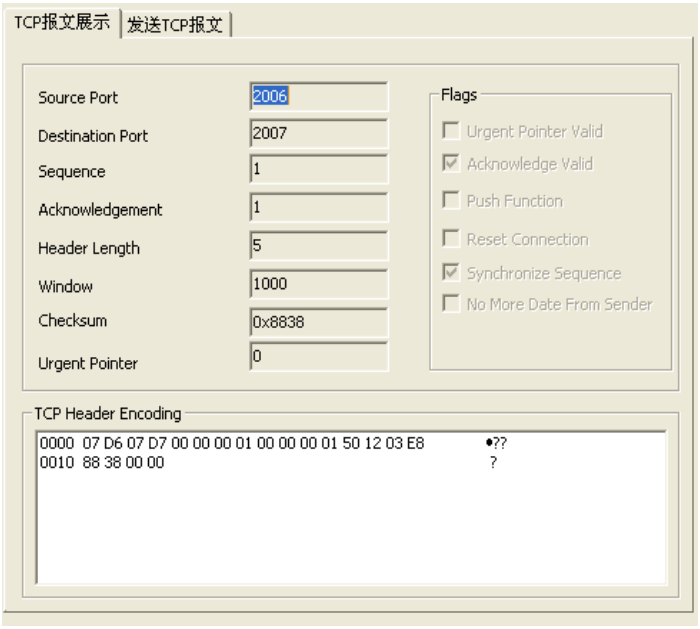

图 4.2　TCP 报文接收界面