# 即时通信相关技术分享

## 网络协议

|  |  |  |
| --- | --- | --- |
| OSI中的层 | 功能 | TCP/IP协议族 |
| 应用层 | 文件传输，电子邮件，文件服务，虚拟终端 | TFTP，HTTP，SNMP，FTP，SMTP，DNS，Telnet |
| 表示层 | 数据格式化，代码转换，数据加密 | 没有协议 |
| 会话层 | 解除或建立与别的接点的联系 | 没有协议 |
| 传输层 | 提供端对端的接口 | TCP，UDP |
| 网络层 | 为数据包选择路由 | IP，ICMP，RIP，OSPF，BGP，IGMP |
| 数据链路层 | 传输有地址的帧以及错误检测功能 | SLIP，CSLIP，PPP，ARP，RARP，MTU |
| 物理层 | 以二进制数据形式在物理媒体上传输数据 | ISO2110，IEEE802。IEEE802.2 |

<http://jingyan.baidu.com/article/08b6a591e07ecc14a80922f1.html>

### TCP/IP协议栈

通常人们认为OSI模型的最上面三层（应用层、表示层和会话层）在TCP/IP组中是一个应用层。由于TCP/IP有一个相对较弱的会话层，由TCP和RTP下的打开和关闭连接组成，并且在TCP和UDP下的各种应用提供不同的端口号，这些功能能够被单个的应用程序（或者那些应用程序所使用的库）增加。与此相似的是，IP是按照将它下面的网络当作一个黑盒子的思想设计的，这样在讨论TCP/IP的时候就可以把它当作一个独立的层。

|  |  |  |
| --- | --- | --- |
| 4 | **应用层** *（OSI 5到7层）* | 例如[HTTP](http://zh.wikipedia.org/wiki/HTTP)、[FTP](http://zh.wikipedia.org/wiki/FTP)、[DNS](http://zh.wikipedia.org/wiki/DNS) *（如*[*BGP*](http://zh.wikipedia.org/wiki/BGP)*和*[*RIP*](http://zh.wikipedia.org/wiki/RIP)*这样的路由协议，尽管由于各种各样的原因它们分别运行在TCP和UDP上，仍然可以将它们看作网络层的一部分）* |
| 3 | **传输层** *（OSI 4层）* | 例如[TCP](http://zh.wikipedia.org/wiki/%E4%BC%A0%E8%BE%93%E6%8E%A7%E5%88%B6%E5%8D%8F%E8%AE%AE)、[UDP](http://zh.wikipedia.org/wiki/UDP)、[RTP](http://zh.wikipedia.org/wiki/RTP)、[SCTP](http://zh.wikipedia.org/wiki/SCTP) *（如*[*OSPF*](http://zh.wikipedia.org/wiki/OSPF)*这样的路由协议，尽管运行在IP上也可以看作是网络层的一部分）* |
| 2 | **网络互连层** *（OSI 3层）* | 对于TCP/IP来说这是[因特网协议](http://zh.wikipedia.org/wiki/%E5%9B%A0%E7%89%B9%E7%BD%91%E5%8D%8F%E8%AE%AE)（IP） *（如*[*ICMP*](http://zh.wikipedia.org/wiki/ICMP)*和*[*IGMP*](http://zh.wikipedia.org/wiki/IGMP)*这样的必须协议尽管运行在IP上，也仍然可以看作是网络互连层的一部分；*[*ARP*](http://zh.wikipedia.org/wiki/ARP)*不运行在IP上）* |
| 1 | **网络接口层** *（OSI 1和2层）* | 例如[以太网](http://zh.wikipedia.org/wiki/%E4%BB%A5%E5%A4%AA%E7%BD%91)、[Wi-Fi](http://zh.wikipedia.org/wiki/Wi-Fi)、[MPLS](http://zh.wikipedia.org/wiki/MPLS)等。 |

### 应用层

该层包括所有和应用程序协同工作，利用基础网络交换应用程序专用的数据的协议。 [应用层](http://zh.wikipedia.org/wiki/%E5%BA%94%E7%94%A8%E5%B1%82)是大多数普通与网络相关的程序为了通过网络与其他程序通信所使用的层。这个层的处理过程是应用特有的；数据从网络相关的程序以这种应用内部使用的格式进行传送，然后被编码成标准协议的格式。

一旦从应用程序来的数据被编码成一个标准的应用层协议，它将被传送到IP栈的下一层。

在传输层，应用程序最常用的是TCP或者UDP，并且服务器应用程序经常与一个[公开的端口号](http://zh.wikipedia.org/wiki/TCP/UDP%E7%AB%AF%E5%8F%A3%E5%88%97%E8%A1%A8)相联系。服务器应用程序的端口由[互联网号码分配局](http://zh.wikipedia.org/wiki/%E4%BA%92%E8%81%94%E7%BD%91%E5%8F%B7%E7%A0%81%E5%88%86%E9%85%8D%E5%B1%80)（IANA）正式地分配，但是现今一些新协议的开发者经常选择它们自己的端口号。由于在同一个系统上很少超过少数几个的服务器应用，端口冲突引起的问题很少。应用软件通常也允许用户强制性地指定端口号作为[运行](http://zh.wikipedia.org/w/index.php?title=%E8%BF%90%E8%A1%8C&action=edit&redlink=1)[参数](http://zh.wikipedia.org/wiki/%E5%8F%82%E6%95%B0)

连结外部的客户端程序通常使用系统分配的一个随机端口号。监听一个端口并且通过服务器将那个端口发送到应用的另外一个副本以建立对等连结（如[IRC](http://zh.wikipedia.org/wiki/IRC)上的[dcc](http://zh.wikipedia.org/w/index.php?title=Dcc&action=edit&redlink=1)文件传输）的应用也可以使用一个随机端口，但是应用程序通常允许定义一个特定的端口范围的规范以允许端口能够通过实现[网络地址转换](http://zh.wikipedia.org/wiki/%E7%BD%91%E7%BB%9C%E5%9C%B0%E5%9D%80%E8%BD%AC%E6%8D%A2)（NAT）的路由器映射到内部。

每一个应用层（[TCP/IP参考模型](http://zh.wikipedia.org/wiki/TCP/IP%E5%8F%82%E8%80%83%E6%A8%A1%E5%9E%8B)的最高层）协议一般都会使用到两个传输层协议之一： 面向连接的[TCP传输控制协议](http://zh.wikipedia.org/wiki/%E4%BC%A0%E8%BE%93%E6%8E%A7%E5%88%B6%E5%8D%8F%E8%AE%AE)和无连接的包传输的[UDP](http://zh.wikipedia.org/wiki/UDP)[用户数据报文协议](http://zh.wikipedia.org/wiki/%E7%94%A8%E6%88%B7%E6%95%B0%E6%8D%AE%E6%8A%A5%E6%96%87%E5%8D%8F%E8%AE%AE)。

常用的应用层协议有：

运行在[TCP](http://zh.wikipedia.org/wiki/%E4%BC%A0%E8%BE%93%E6%8E%A7%E5%88%B6%E5%8D%8F%E8%AE%AE)协议上的协议：

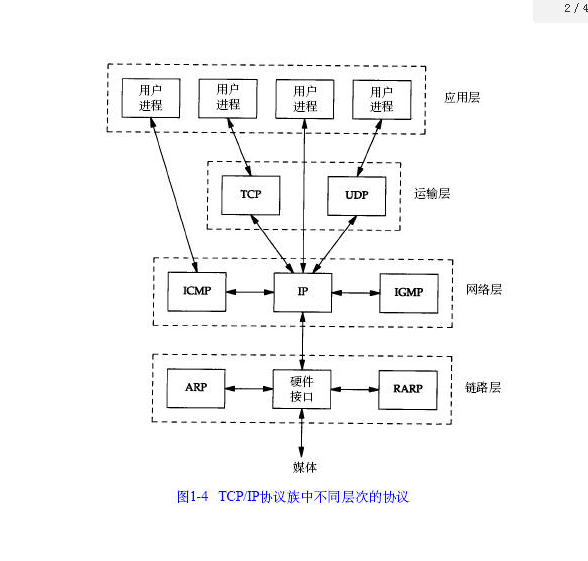
* [HTTP](http://zh.wikipedia.org/wiki/HTTP)（Hypertext Transfer Protocol，超文本传输协议），主要用于普通浏览。
* [HTTPS](http://zh.wikipedia.org/wiki/HTTPS)（Hypertext Transfer Protocol over Secure Socket Layer, or HTTP over SSL，安全超文本传输协议）,HTTP协议的安全版本。
* [FTP](http://zh.wikipedia.org/wiki/FTP)（File Transfer Protocol，文件传输协议），由名知义，用于文件传输。
* [POP3](http://zh.wikipedia.org/wiki/POP3)（Post Office Protocol, version 3，邮局协议），收邮件用。
* [SMTP](http://zh.wikipedia.org/wiki/SMTP)（Simple Mail Transfer Protocol，简单邮件传输协议），用来发送电子邮件。
* [TELNET](http://zh.wikipedia.org/wiki/TELNET)（Teletype over the Network，网络电传），通过一个终端（terminal）登陆到网络。
* [SSH](http://zh.wikipedia.org/wiki/SSH)（Secure Shell，用于替代安全性差的[TELNET](http://zh.wikipedia.org/wiki/TELNET)），用于加密安全登陆用。

运行在[UDP](http://zh.wikipedia.org/wiki/UDP)协议上的协议：

* [BOOTP](http://zh.wikipedia.org/wiki/BOOTP)（Boot Protocol，启动协议），应用于无盘设备。
* [NTP](http://zh.wikipedia.org/wiki/NTP)（Network Time Protocol，网络时间协议），用于网络同步。

其他：

* [DNS](http://zh.wikipedia.org/wiki/%E5%9F%9F%E5%90%8D%E7%B3%BB%E7%BB%9F)（Domain Name Service，域名服务），用于完成地址查找，邮件转发等工作（运行在[TCP](http://zh.wikipedia.org/wiki/%E4%BC%A0%E8%BE%93%E6%8E%A7%E5%88%B6%E5%8D%8F%E8%AE%AE)和[UDP](http://zh.wikipedia.org/wiki/UDP)协议上）。
* [ECHO](http://zh.wikipedia.org/w/index.php?title=ECHO&action=edit&redlink=1)（Echo Protocol，回绕协议），用于查错及测量应答时间（运行在[TCP](http://zh.wikipedia.org/wiki/%E4%BC%A0%E8%BE%93%E6%8E%A7%E5%88%B6%E5%8D%8F%E8%AE%AE)和[UDP](http://zh.wikipedia.org/wiki/UDP)协议上）。
* [SNMP](http://zh.wikipedia.org/wiki/SNMP)（Simple Network Management Protocol，简单网络管理协议），用于网络信息的收集和网络管理。
* [DHCP](http://zh.wikipedia.org/wiki/DHCP)（Dynamic Host Configuration Protocol，动态主机配置协议），动态配置IP地址。
* [ARP](http://zh.wikipedia.org/wiki/ARP)（Address Resolution Protocol，地址解析协议），用于动态解析以太网硬件的地址。



TCP/IP（TransmissionControlProtocol/InternetProtocol的简写，中文译名为传输控制协议/互联网络协议）协议是Internet最基本的协议，简单地说，就是由底层的IP协议和TCP协议组成的。TCP/IP协议的开发工作始于70年代，是用于[互联网](http://www.baike.com/wiki/%E4%BA%92%E8%81%94%E7%BD%91)的第一套协议。

TCP/IP的通讯协议

简要介绍一下TCP/IP的内部结构，TCP/IP协议组之所以流行，部分原因是因为它可以用在各种各样的信道和底层协议（例如T1和X.25、以太网以及RS-232串行接口）之上。确切地说，TCP/IP协议是一组包括TCP协议和IP协议，[UDP](http://www.baike.com/wiki/UDP)（UserDatagramProtocol）协议、[ICMP](http://www.baike.com/wiki/ICMP)（InternetControlMessageProtocol）协议和其他一些协议的协议组

## Tomcat源码

官方源码

<http://tomcat.apache.org/svn.html>

### 编译Tomcat的源码

1：从svn 检出源代码  [http://svn.apache.org/repos/asf/tomcat](http://svn.eu.apache.org/repos/asf/tomcat/trunk/)

假设位于：E:\workspaces\MyEclipse\_9\TOMCAT\_7\_0\_19

2：编辑：

 build.properties.default  
把base.path=/usr/share/java   
改成base.path=E:/workspaces/MyEclipse\_9/TOMCAT\_7\_0\_19

**3:**用ant编译Tomcat7

在myeclipse中右击build.xml ---> run build

开始编译：

--------------------------------------------------

BUILD SUCCESSFUL   
Total time: 5 minutes 16 seconds   
**第三步**：  
修改E:/。。/output/build/conf/server.xml   [注意并不是根目录下的/conf下的]

【运行org.apache.catalina.startup.Bootstrap时候，则修改conf的server.xml.. 】

 将

**注**意**:目录结构： 反斜杠 / 或者 \\**

**<Connector** port="8080" protocol="HTTP/1.1"    connectionTimeout="20000"      redirectPort="8443" **/>**

改成

**<Connector** port="8080" protocol="org.apache.coyote.http11.Http11NioProtocol" connectionTimeout="20000"

               redirectPort="8443" **/>**

7.运行:   
。。。。。/output/build/bin>startup

注意： 如果直接双击 startup.bat  需要 配置 catalina\_home 环境变量

【此时的catalina\_home为E:/workspaces/MyEclipse\_9/TOMCAT\_7\_0\_19/o**utput/build** 】

[Http://localhost:8080/excmples/](http://localhost:8080/excmples/)

.....................

  ok  tomcat 检出并编译成功

 然后接下来是运行tomcat 源代码 ：

 新建一个 java 工程 【注意不是web工程 】 test

 然后把 trunk 中的 java 下的源文件 导入test 下的src

 然后把 /output/build   下的 conf    webapps  俩文件夹copy到

 test 的根目录下 : 目录结构如下

 test --

       -- src

       -- conf

       -- webapps

 其他的非必须。。。。。。。。

 运行 org.apache.catalina.startup.Bootstrap    搞定。。。。

=====================

  其中需要四个jar

1：  ant.jar  ：  下载 Ant    找到  lib  目录

 2：  eclipse :org.eclipse.jtd.core\_xxxx.jar   :  去 eclipse 官网 下载 ：

 3：  jaxrpc.jar    |     wsdl4j-1.5.1.jar

 去 apache 下载 axis-1.4   ： 解压  找到 lib  目录

 注意： axis-1.4  与 其他的版本 尤其是 1.5.x 由很大的不同

 其他版本中 不存在 jaxrpc.jar