毕业设计（论文）模版

**西安邮电大学**

**毕业设计（论文）**

题目： 基于Python网络爬虫的天气信息

搜索与预报系统

学院： 通信与信息工程学院

专业： 通信工程

班级： 通工1310

学生姓名： 鲍颖

学号： 03131313

导师姓名： 石薇 职称：

起止时间： 2017年2月 27日 至 2017 年 6 月 17 日

## 毕业设计（论文）声明书

本人所提交的毕业论文《基于Python网络爬虫的天气信息搜索与预报系统》是本人在指导教师指导下独立研究、写作的成果，论文中所引用他人的文献、数据、图件、资料均已明确标注；对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式注明并表示感谢。

本人完全理解《西安邮电大学本科毕业设计（论文）管理办法》的各项规定并自愿遵守。

本人深知本声明书的法律责任，违规后果由本人承担。

论文作者签名：

日期： 年 月 日

摘□□要

*（“摘要”之间空两格，采用三号字、黑体、居中，与内容空一行）*

在互联网发展的初期，用户主要通过访问一些知名的门户网站的方式去搜集自己想要的数据信息，但是随着Web的急剧发展与扩大，这种传统的获取信息的方式已经变得越来越困难了。因为互联网就像是一个非常庞大的数据库，各种各样的数据和资源以各种形式存储于网络上，在网页展示形式也非常多：文字、图片、Flash动画或者视频等等。用户试图在网页上获取一些信息并点击链接的时候，往往会收到大量外带的数据，例如弹窗推送广告、网页浮动窗口等，这些多余的信息不但容易干扰视线、使用户较难筛选出有效信息，而且某些广告弹窗可能会暗藏一些恶意链接，使用户造成不必要的损失。因此，通过网络爬虫技术来进行高效、快速的数据搜集，将可以给用户提供一个纯净、可靠地获取信息的方式。本课题旨在设计并实现一个可以主动、定向收集天气信息的运行在服务器上的Python网络爬虫程序，使得客户端在去获取信息的时候，避开大量的无效信息的干扰，使用户能够高效、快速的完成信息的获取。

关键词：网络爬虫;客户端-服务器模式;Python

×××××

*采用小四号、宋体、接排*

*小四号、黑体、顶格*

# ABSTRACT

*（采用三号字、Times New Roman字体、加黑、居中、与内容空一行）*

□□In the early days of the Internet, the user mainly through the access to some well-known portal to collect the information they want, but with the rapid development and expansion of the Web, the traditional way to obtain information has become more and more Difficult. Because the Internet is like a very large database, a variety of data and resources in a variety of forms stored in the network, in the form of web display is also very much: text, pictures, Flash animation or video, and so on. When users try to get some information on the page and click on the link, they often receive a lot of data, such as pop-up advertising, page floating window, etc., these redundant information is not only easy to interfere with the line of sight, so that users more difficult to screen out Effective information, and some advertising bounces may hide some malicious links, causing users to cause unnecessary losses. Therefore, through the web crawler technology to carry out efficient and fast data collection, will be able to give users a pure, reliable way to obtain information. The purpose of this project is to design and implement a web crawler program that can be used to collect weather information on the server, so that when users try to get information on the network, avoid a large amount of invalid information interference, so that users can Efficient and fast access to information.*（内容采用小四号Times New Roman字体）*

**Key words：** Web Crawler;Client - server mode;Python

×××××

*采用小四号、Times New Roman字体、接排*

*小四号、Times New Roman、加黑、顶格*

目录

*（三号、黑体、居中、目录两字空两格、与正文空一行）*

第一章 *引言……………………………………………………………*×

1.1选题背景与意义………………………………………………………………×

1.2网络爬虫技术发展现状**………………………………………………………**×

1.3 C/S架构下的软件开发…………………………………**………………**×

1.4 本文章节安排**………………………………………………………………**×

……………

第二章 *网络爬虫技术…………………………………………………*×

2.1 网络爬虫的基本架构及工作流程**…………………………………………**×

2.2 网络爬虫的搜索策略**…………………………………………**×

2.2.1 深度优先搜索策略**………………………………………………………**×

2.2.2 广度优先搜索策略**………………………………………………………**×

2.2.3 聚焦搜索策略**……………………………………………………………**×

2.3 网络爬虫伪装用户**…………………………………………**×

2.3.1 IP代理**…………………………………………**×

2.3.2 User Agent**……………………………………**×

2.3.3 Cookie**……………………………………**×

2.4 几种开源爬虫框架的对比测试**……………………………………………**×

2.5 网络爬虫的几种应用**……………………………………………**×

………………

第三章 *基于客户端/服务器模式下的软件开发……………………*×

3.1 客户端/服务器模式介绍**……………………………………………………**×

3.1.1 什么是C/S架构**…………………………………………………………**×

3.1.2 C/S架构与B/S架构的区别**……………………………………………**×

3.2 客户端与服务器的通信**……………………………………………………**×

3.2.1 客户端与服务端的通信过程及原理**……………………………………**×

3.2.2 底层网络协议**…………………………………………**×

3.2.3 I/O多路复用**………………………………………**×

3.3 linux服务器端程序开发**……………………………………**×

3.3.2 代码版本管理**……………………………………**×

3.3.3 多线程支持**……………………………………**×

………………

1. *基于网络爬虫的天气搜索工具的设计与实现………………*×

4.1 软件架构设计

4.2 数据爬取**………………………………………………**×

4.2.1 网站结构的分析**……………………………………………………**×

4.2.2 Spider的设计实现**………………………………………**×

4.2.3 Spider的伪装与使用代理IP**………………………………………**×

4.2.4 使用多线程提升爬虫性能**………………………………………**×

4.2.5 Crontab实现定时数据爬取**………………………………………**×

4.3 数据存储**……………………**×

4.3.1 数据存储需求分析**………………………**×

4.3.2 数据库结构设计**………………………**×

4.4 服务器程序的设计实现**………………………**×

4.4.1 使用网络套接字与Client通信**………………………**×

4.4.2 Server验证登录用户合法性**………………………**×

4.3.3 后端数据分析与查询**………………………**×

4.3.4 使用IO多路复用提升Server性能**………………………**×

4.5 Client的设计与实现**………………………**×

4.5.1 使用Qt进行GUI开发**………………………**×

4.5.2 用户登录模块设计与实现**………………………**×

4.5.3 数据收发模块设计与实现**………………………**×

4.5.4 数据分析与曲线绘制**………………………**×

4.6 **………………………**×

………………

结束语*（四号、宋体）………………………………………*………×

致谢*（四号、宋体）……………………………………………………*×

参考文献*（四号、宋体）………………………………………*………×

附录*（四号、宋体）……………………………………………*……**…**×

第一章□□引言*（居中、小三号、黑体）*

1.1选题背景与意义*（四号、黑体、顶格）*

正文开始标注页眉，宋体五号居中

□□随着时代的进步与科技的发展，互联网已经在不知不觉间蔓延、扩散到了我们生活空间的每一个角落。近年来被社会谈论的非常热门的一些科技词汇，比如大数据、云计算、物联网等，几乎都和互联网的繁荣发展息息相关。互联网具有的多维性、自由平等性、虚拟交互性、以及最重要的海量性与全球性，改变了生产信息与传播信息的方式，进而改变了人类的工作方式、生活方式、生产方式甚至是思维方式。人们已逐渐习惯通过互联网去获取自己需要的各种信息（浏览新闻网站或者在线观看视频），或者是通过互联网去发布一些自己愿意公开的信息（发微博、写说说），而不是通过曾经的那些传统渠道（报纸、杂志），传统渠道在各方面所占的份额也在逐渐降低，这证明人们对信息的获取方式的态度也在转变。

□□然而这种情景的背后，有我们需要迫切关注的一点，那就是每天都有EB级别的新数据涌入互联网这个庞大的数据库，那么随着数据的不断累积，摆在我们家面前的的就是一个拥有海量数据的、无比庞大的数据库，各种各样的信息全部沉淀、堆积在这个庞大的数据库之中。这就意味着，当用户想要从互联网上接收信息的时候，不一定能够快速、准确的获取到自己真正想要的信息，而且这些信息中或许掺杂了非常多的无用数据，甚至于是包含了恶意链接的、会造成用不必要损失的内容。因此，能够从网络中定向的、准确的、高效的提取用户需要的信息，才可以真正的提升用户的体验。本课题以使用频率较高的天气搜索功能为起点，通过网络爬虫技术与C/S软件架构模型，设计并实现了一套能够高效获取互联网天气数据并分析的的软件，可以精确、快速的获取天气信息，降低用户浪费在数据分辨与筛选上的时间，为用户提供一个纯净、可靠的信息获取来源。

1.2网络爬虫技术发展现状

传统的网络爬虫技术，主要是用来爬取静态的web页面，基于socket的httpclient功能简单，性能强大，特别是在多线程、高并发的情境下，从而被开发者青睐，特别是搜索引擎中，若抓取的是静态页面，httpclient是非常适合这种情景的。但是随着AJAX/web2.0技术的兴起，很多网页都使用这种新技术改写成了动态网页，所以，如何抓取AJAX等动态页面，就成了搜索引擎紧急需要解决的问题。因为AJAX颠覆了传统的纯HTTP请求/响应的协议机制，当遇到AJAX加载的信息、需要JavaScript渲染的时候，httpclient就无法正常工作了。所以若爬虫程序还是按照传统的下载网页、分析文档、提取链接的机制的话，是无法抓取到AJAX页面的有效数据的。AJAX采用的是JavaScript驱动的异步请求/响应机制，传统的爬虫采用的是xpath或正则表达式这种技术来进行内容的提取，而缺乏对JavaScript语义上的理解，是无法模拟出触发JS的异步调用并解析返回的异步回调逻辑和内容。而且，在AJAX的应用中，JS会对DOM结构进行修改与变动，甚至于页面的所有内容都是通过JavaScript动态绘制的，这种动态页面生成技术与DOM结构相对不变的静态页面相差甚多。所以从此处可以看出，传统的爬虫程序是基于HTTP协议驱动的，用来爬取静态页面足够了，但是对于新型的AJAX这样的技术，爬虫引擎的核心必须是基于事件驱动的。

若是爬虫程序的可行性得到了解决，那么其面临的最大问题就是爬虫性能的提升和突破网站对爬虫的封锁。大部分情况下，采用多线程高并发、高频率进行数据抓取是可行的，但是这有一个前提，那就是网站没有采用任何反爬虫措施，例如验证码、防火墙、限制访问频率等等。但更多的时候，真正有价值的信息，一定是伴随着严格的反爬取策略的，一旦网站检测出当前请求者是爬虫程序而非真正的用户，那么最严格的情况下，有可能会直接将当前IP地址加入黑名单，一旦被加入黑名单，则意味着当前运行爬虫程序的服务器再也无法访问这个网站，更别提爬取数据了。所以，由此诞生出很多反封锁的办法，例如维护一个代理IP池来防止网站封IP，但是这也带来了新的问题，那就是IP池是否稳定，和切换IP的速度等等，这些都是开发出一款稳定的、功能完善的网络爬虫而无法避免、需要考虑的问题。

1.3 C/S架构下的软件开发

1.4 本文章节安排

本文章节具体安排如下：

1. ：引言部分。主要用于介绍背景知识，包括网络爬虫技术发展现状，以及客户端/服务器架构下的软件技术的发展现状。
2. ：网络爬虫技术。对网络爬虫技术进行系统的阐述，介绍网络爬虫的几种搜索策略，以及网络爬虫如何模拟用户行为，并对多个开源爬虫框架进行了对比测试。

第三章：基于客户端/服务器模式下的软件开发，阐述什么是C/S架构，解释其通信流程，与使用的底层网络协议，以及如何使用IO复用提升服务器性能。

第四章：基于网络爬虫的天气搜索工具的设计与实现。主要介绍天气搜索工具的设计步骤与实现细节，并对爬取结果进行处理与分析。

第二章□□网络爬虫技术*（居中、小三号、黑体）*

2.1网络爬虫的基本架构及工作流程*（四号、黑体、顶格）*

网络爬虫（Web crawler）是一种能够自动提取网页信息程序的统称。从功能上来讲，爬虫程序一般分为数据采集、数据处理、数据存储三个模块。网络爬虫的定义有狭义与广义之分，广义爬虫的定义为：所有能够利用http协议搜索web文档的软件都称之为网络爬虫。狭义爬虫的定义 为：利用标准的http 协议，根据超级链接和Web 文档检索的方法遍历万维网信息空间的软件程序。

网络爬虫的功能是根据已有URL去爬取对应的网站，并从得到的html文档中提取出设计者需要的信息和新的URL,然后就可以进行递归的爬取。所以说，网络爬虫的本质就是模拟用户使用浏览器发出http请求，虽然浏览器和网络爬虫是两种不同的客户端程序，但都用相同的方式去获取网页：客户端程序连接到DNS服务器上，DNS服务器将传给它的域名翻译成web服务器的IP地址，然后客户端通过IP地址和端口去向web服务器发起http请求，http协议是OSI七层模型中的应用层协议，它使用的底层协议分别是TCP（传输层协议）和IP（网络层协议）。连接建立之后，客户端的请求被web服务器收到，服务器会将相应的结果返回给客户端，并且一般来讲，HTTP协议默认的端口是80端口，是一个知名端口。

但是浏览器和爬虫程序还有有着本质的区别的，因为浏览器需要用户的点击来完成操作http请求，而爬虫程序是自动完成http请求的，所以爬虫程序需要有一整套完善的架构才能完成全部的工作任务。

在网络爬虫的系统框架中，主要过程结构由控制器，解析器，资源库三部分。控制器（也称调度器）的主要工作是负责给多线程中的各个爬虫线程分配工作任务。解析器的主要工作是根据url队列中的网页地址下载对应网页，并进行页面的分析处理，主要是将一些JS脚本标签、CSS代码内容、HTML标签等内容处理掉，爬虫的基本工作是由解析器完成的。资源库是用来存放下载到的网页资源的，将解析器解析出的数据保存起来，一般都采用大型的数据库存储，并对其建立索引。

下面是爬虫的工作流程图与步骤解析。



爬虫技术已经发展了十几年，整体框架上已经相对成熟，下面是通用的爬虫框架工作流程：

1）首先从所要爬取的网站中挑选一些合适的网页，将这些网页的url作为初始url（也成为种子url）

2）将初始url放到一个待抓取url的队列中，之后调度器将从这个队列中取出url进行网页下载。

3）爬虫程序从待抓取url队列中一次读取url，并将url通过DNS解析转换为web服务器IP地址

4）再由网页下载器根据IP地址和url中域名后的相对路径，发起http请求，下载目标网页

5）已经下载的网页，爬虫程序会将这个网页的url信息存入到一个已抓取url队列中，这个队列中记录着所有已经被下载器下载过的网页的url,通过这种机制，可以防止重复多次抓取同一个网页。

6）爬虫程序会对下载好的html文档进行分析，将其中的url链接提取出来，使用Bloom Fliter之类比较高效的算法，测试已爬取队列中是否存在这个url，如果不在，则说明这个一个未被爬取的url，那么就将它放入带爬取队列中。

7）调度器再次从带爬取队列中取出url进行爬取，如此形成一个完整的循环，直到待爬取队列为空结束。

2.1网络爬虫的搜索策略

网络爬虫是从一个或若干个初始网页URL开始，获取初始网页上的URL，在抓取网页的过程中，不断的从当前页面上抽取新的URL放入带爬取队列，重复这个过程直到爬取完全部页面，或满足系统设置的一定停止条件。因为这种数据采集过程像是一个爬虫或蜘蛛在网络上漫游，所以它才被称为网络爬虫系统或网络蜘蛛系统。然而由于网络资源太过于庞大复杂，无数网页资源之间的关系盘根错节，就算是再精明的爬虫、蜘蛛，可能也会在这网络森林中迷失方向，所以我们需要给爬虫从网页上爬取数据的过程设置一些搜索策略，以便更好的进行数据搜集工作。

网络爬虫通过请求网站站点上的HTML文档访问某一站点，遍历web空间，不断从一个站点爬到另一个站点，搜索与分析网站内包含的信息，对信息建立索引，再将索引关系加入到网页数据库中，并利用HTML语言的标记结构来搜索信息及获取指向其他网站的URL地址，通过这种方式，可以完全不依赖用户的干预实现在网络上的自动爬取。

2.1.1 深度优先搜索策略*（四号、黑体、顶格）*

深度优先搜索策略所遵循的原理就是尽可能的“往深处走”，从起点开始尽可能深的搜索图。在计算机科学的图论中，深度优先就是对于一个节点来说，如果这个节点还有以它为起点而且尚未被探索完的边，那么就继续从这个节点探索下去，直到这个节点的所有边都已经被探寻过，搜索将回溯到上层节点。这一过程一直持续到所有从源节点开始的可以到达的节点都被探索完毕为止。但是深度优先搜索策略可能会使爬虫陷入一个网站系统内部，不利于搜索比较靠近网站首页的网页信息，而且有的时候可能会导致爬虫陷入过深的递归搜索中导致爬虫出不来的情况。

2.1.2 广度优先搜索策略*（四号、黑体、顶格）*

广度优先搜索算法是最便捷的搜索算法之一，同时这种算法也是很多重要的图论算法的原型。例如Prime最小生成树算法与Dijkstra有向图的单源最短路径算法，都采用了类似广度优先搜索类似的思想。广度优先搜索算法的思想是沿着树的宽度遍历树的节点，遍历一个节点，并将其所有的子节点加入队列中，按“层”遍历。在爬虫程序的具体的实现中，就是将新下载的网页包含的URL链接直接追加到带爬取URL队列的末尾。

有实验表明，这种策略的实际效果很好，虽然看似死板与机械，但实际上的网页抓取顺序基本上是按照网页的重要性进行排序。之所以如此，有研究人员认为：如果某个网页入度很高（包含很多入链），那么更有可能被广度优先遍历策略早早的抓到，从某种意义上讲，入链个数也体现了一个网页的重要性，即实际上广度优先遍历策略也隐含了一些网页优先级假设。

2.1.3 聚焦搜索策略*（四号、黑体、顶格）*

聚焦搜索策略的出现，主要是因为随着互联网的不断发展与不断扩大，网页信息开始呈指级增长，那么传统的搜索策略在面对如此海量与庞大的网页信息会显得力不从心，因为其中掺杂了大量的与想要获取的信息关联性较低的或毫无关联内容，面对这种情况，定向抓取相关网页资源的聚焦爬虫便应运而生。聚焦爬虫的爬行策略只挑出所需要的特定主题的页面。聚焦爬虫会给它所下载下来的网页进行评价并排序，排序越靠前的，会越先被访问，这种策略会保证爬虫能优先跟踪那些最有可能包含目标页面链接的页面，决定这种搜索策略的关键是如何评价链接价值，或者成为链接价值的计算方法。

不同的链接价值评价方法计算出的结果是不同的, 那么最终得出的链接的“重要程度”的排序也不同， 从而决定了不同的搜索策略。由于url链接包含于网页页面之中，而通常具有较高价值的网页页面包含的url链接也具有较高的价值，因而对链接价值的评价有时也转换为对页面价值的评价. 这种策略通常运用在专业搜索引擎中，因为这种搜索引擎只关心某一特定主题的页面。但有的时候，也会因此会漏掉一些评价不高，但是可能确实包含了有用信息的相关网页，所以不能盲目使用聚焦搜索策略，还应针对问题具体分析，才能得到最好的抓取效果。

2.3网络爬虫伪装用户

随着互联网的发展，网络爬虫已经成了很普及的网络技术，一个内容驱动型的网站，容易受到网络爬虫的青睐与光顾，这是不可避免的。一些智能的搜索引擎的爬虫的爬取频率是比较合理的，对网站资源的占用与消耗都较低，但是仍旧有很多比较糟糕的爬虫，肆无忌惮的并发几十上百请求来进行网页的循环重复抓取，这种网络爬虫对网站资源的消耗是十分巨大的，特别是对一些中小型网站来说，很容易就会造成网站访问压力增大，导致网站访问速度慢，甚至无法访问。

所以，一般来说，网站从三个方面进行反爬虫：用户请求的报文头部（Headers），用户行为，网站的数据的加载方式。从用户请求的Headers反爬虫是最常见的网站反爬虫策略，很多网站都会对报头Headers的User-Agent部分进行检测，还有一部分网站会对Referer部分进行检测（一些资源网站的防盗链就是通过检测Referer来进行的）。用户行为检测是有些网站是通过检测用户行为来判断是否是爬虫的，例如一个IP地址短时间内多次访问了同一页面，或者同一用户账户短时间内多次进行了相同的操作，那么很有可能会被认为是网络爬虫而被采取相应措施。有些网站的信息必须登录后才能获取到，例如新浪微博，不登录账号，只能看到大V的前十条微博，想要看到全部内容，网站要求用户必须处于登录态，也即需要Cookie验证，才能获取到相关数据。

那么，当我们需要爬取这个网站上的数据的时候，这些策略我们应该如何去应对呢？其实我们可以把爬虫程序完全伪装成为用户的正常浏览行为，从而躲避大部分的反爬虫策略。通过以下几种策略，可以将爬虫进行伪装，从而防止被网站的反爬虫机制发现。当然，访问频率，并发数目这些是无法逃避的，需要有足够的耐心和足够多的机器、有足够多的代理IP，那总是可以抓取到足够多的信息的。

2.3.1 User Agent

从用户请求的报头Headers进行分析，是最常见的反爬虫策略。很多网站都会对Headers的User-Agent部分进行检测，看是否存在User-Agent或者User-Agent是否合法，因为正常用户通过浏览器进行网页访问，一定是有User-Agent的。还有一部分网站会对Referer部分进行检测（一些资源网站的防盗链就是检测Referer）。如果遇到了这一类反爬虫机制，我们可以直接在爬虫中添加Headers，将浏览器的User-Agent部分复制到爬虫的报头Headers中；或者直接将Referer值修改为目标网站的域名。

2.3.2 IP代理

一般来说，网站会针对用户行为进行反爬虫。比如说同一IP地址，短时间内多次尝试访问同一类型的页面，或者同一用户账号短时间内多次进行了相同操作，若发生这些情况，那么很有可能会被认为是网络爬虫。大多数网站的反爬策略都是前一种，对于这种情况，使用代理IP就可以很好的解决问题，可以专门维护一个代理IP池，有了大量的代理ip之后，我们可以每对网站发几次请求 就更换一个ip地址，更换IP的方法，在requests或者urllib2中很容易实现，，这样可以很简单绕过第一种反爬虫机制。对于第二种限制同一账号相同操作次数过多的情况，可以在每次发出请求后，随机的间隔几秒，再进行下一次请求的发出。也可以发出几次请求之后，主动地进行退出登录，然后重新登陆，继续发出请求来绕过同一账户，短时间内进行相同请求的容易被反爬虫的限制，但单一账号如果操作次数过多，也很容易被发现，这种时候，最好可以维护一个账号池，将多个账号放入其中，可以达到更好的效果。

2.3.3 Cookie

一般来说，网站会

结束语

*（三号、黑体、居中、与正文空一行）*

致谢

*（三号、黑体、居中、致谢两字空两格、与正文空一行）*

参考文献（三号、黑体、顶格）

[1] [凤祥云](http://xueshu.baidu.com/s?wd=author%3A%28%E5%87%A4%E7%A5%A5%E4%BA%91%29%20&tn=SE_baiduxueshu_c1gjeupa&ie=utf-8&sc_f_para=sc_hilight%3Dperson" \t "_blank)，[孙海艳](http://xueshu.baidu.com/s?wd=author%3A%28%E5%AD%99%E6%B5%B7%E8%89%B3%29%20&tn=SE_baiduxueshu_c1gjeupa&ie=utf-8&sc_f_para=sc_hilight%3Dperson" \t "_blank)，[张万臣](http://xueshu.baidu.com/s?wd=author%3A%28%E5%BC%A0%E4%B8%87%E8%87%A3%29%20&tn=SE_baiduxueshu_c1gjeupa&ie=utf-8&sc_f_para=sc_hilight%3Dperson" \t "_blank).[基于光纤通信技术的物联网传感器系统](http://xueshu.baidu.com/s?wd=paperuri%3A%28c77a9549c24b8365681bc6195aed1b3c%29&filter=sc_long_sign&tn=SE_xueshusource_2kduw22v&sc_vurl=http%3A%2F%2Fwww.cqvip.com%2FQK%2F91041X%2F201607%2F669514878.html&ie=utf-8&sc_us=7665783302043157044" \t "_blank)[J].激光杂志, 2016(7):131-134．

[2] [迪内希·钱德拉·维玛](http://xueshu.baidu.com/s?wd=author%3A%28%E8%BF%AA%E5%86%85%E5%B8%8C%C2%B7%E9%92%B1%E5%BE%B7%E6%8B%89%C2%B7%E7%BB%B4%E7%8E%9B%29%20&tn=SE_baiduxueshu_c1gjeupa&ie=utf-8&sc_f_para=sc_hilight%3Dperson" \t "_blank)，[帕利德·维玛](http://xueshu.baidu.com/s?wd=author%3A%28%E5%B8%95%E5%88%A9%E5%BE%B7%C2%B7%E7%BB%B4%E7%8E%9B%29%20&tn=SE_baiduxueshu_c1gjeupa&ie=utf-8&sc_f_para=sc_hilight%3Dperson" \t "_blank).大数据爆炸时代的移动通信技术与应用[M]. 郎为民,译．北京：机械工业出版社, 2016：20-30．

[3] [刘俊文](http://xueshu.baidu.com/s?wd=author%3A%28%E5%88%98%E4%BF%8A%E6%96%87%29%20&tn=SE_baiduxueshu_c1gjeupa&ie=utf-8&sc_f_para=sc_hilight%3Dperson" \t "_blank)，[赵子岩](http://xueshu.baidu.com/s?wd=author%3A%28%E8%B5%B5%E5%AD%90%E5%B2%A9%29%20&tn=SE_baiduxueshu_c1gjeupa&ie=utf-8&sc_f_para=sc_hilight%3Dperson" \t "_blank)，[徐慧明](http://xueshu.baidu.com/s?wd=author%3A%28%E5%BE%90%E6%85%A7%E6%98%8E%29%20&tn=SE_baiduxueshu_c1gjeupa&ie=utf-8&sc_f_para=sc_hilight%3Dperson" \t "_blank)，[张素香](http://xueshu.baidu.com/s?wd=author%3A%28%E5%BC%A0%E7%B4%A0%E9%A6%99%29%20&tn=SE_baiduxueshu_c1gjeupa&ie=utf-8&sc_f_para=sc_hilight%3Dperson" \t "_blank)．[量子通信技术在电力信息系统保密传输中的应用](http://xueshu.baidu.com/s?wd=paperuri%3A%2807b58d40f1334061b898700237944390%29&filter=sc_long_sign&tn=SE_xueshusource_2kduw22v&sc_vurl=http%3A%2F%2Fcpfd.cnki.com.cn%2FArticle%2FCPFDTOTAL-YDDX201609001069.htm&ie=utf-8&sc_us=2010724533587303457" \t "_blank)[C] [电力行业信息化年会](http://xueshu.baidu.com/usercenter/data/journal?cmd=jump&wd=confuri%3A%285b41c4bb46127d22%29%20%E7%94%B5%E5%8A%9B%E8%A1%8C%E4%B8%9A%E4%BF%A1%E6%81%AF%E5%8C%96%E5%B9%B4%E4%BC%9A&tn=SE_baiduxueshu_c1gjeupa&ie=utf-8&sc_f_para=sc_hilight%3Dpublish&sort=sc_cited" \t "_blank" \o "电力行业信息化年会), 2016

[4] DL/T5344-2006,电力光纤通信工程验收规范[S].2006.

[5] 姜锡洲.一种温热外敷药制备方案：中国,881056073[P].1989-07-26.

[6] 王明亮.关于中国学术期刊标准化数据库系统工程的进展[EB/OL]. (1998-08-16)[1998-10-04].http：//[www.cajcd](http://www.cajcd).edu.cn/pub/wml.tex/980810-2.html.

[7] 丁文祥.数字革命与竞争国际化[N].中国青年报,2000-11-20（15）.

[8] 张志祥.间断动力系统的随机扰动及其在守恒律方程中的应用[D].北京：北京大学数理学院,1998.

[9] World Health Organization.Factors regulating the immune response:report of WHO Scientific Group[R].Geneva:WHO,1970.

*（以上，如果需要两行的，第二行文字要位于序号的后边，与第一行文字对齐。中文的用五号宋体，外文的用五号Times New Roman字体。）*

附录X

*(采用三号字、黑体、顶格，与内容空一行，X表示A，B，C等)*

（小四号宋体，行距1.25倍）×××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××××