

基于 Linux 平台的 Ad - Hoc 实验床的搭建

Construction of Ad - Hoc Test Bed Based on Linux

徐梦然 徐 乾 (中国科学院 计算机网络信息中心 北京 100080)
(中国科学院研究生院 北京 100049)

摘 要: 本文根据工程实践,介绍了在 Linux 平台下 Ad - hoc 实验床的搭建过程。文章首先介绍了在 Linux 平台下无线设备的安装与配置,进而介绍了当前 Ad - hoc 路由协议的实现情况,最后对 Ad - hoc 实验床的调试过程以及调试结果进行了说明和分析。

关键词: Ad - hoc Linux 无线设备

1 引言

Ad - hoc 网络领域是当前研究的热点,很多项目都与此有关。近几年 Ad - hoc 相关领域的学术论文也有上千篇。为了更好的测试研究结果以及最终构建出能够满足某种应用需求的 Ad - hoc 网络,搭建一个 Ad - hoc 实验床是很有必要的。

另外,linux 平台因其优越的开放性与可靠性,已被广大计算机网络领域的研究人员所使用。因此,所搭建的 Ad - hoc 实验床应该尽可能的基于 Linux 平台。

本文将根据在 Linux 平台下搭建 Ad - hoc 实验床中所获得的实践经验,对搭建过程中的各个环节进行详细阐述。文章的主要内容包括:Linux 平台下无线设备的安装与配置;Ad - hoc 路由协议;Ad - hoc 实验床的调试与结果分析。

2 Linux 平台下无线设备的安装与配置

《Linux Wireless LAN Howto》是 Linux 社团在官方页面上提供的一个指南。它主要介绍了关于在 Linux 平台下安装配置无线网络的技术。

下面依据这份指南,并结合实践过程中所遇到的具体情况对 Linux 平台下无线设备的使用进行介绍。

2.1 Linux 平台下无线设备的安装

在《Linux Wireless LAN Howto》中,作者将大量的篇幅用在了介绍无线设备以及相应的驱动上。可见,这一部分在 Linux 的无线应用方面是非常重要的。之所以这样的一个主要原因是大多数的无线设备芯片制

造商不支持 Linux 平台。

为解决这方面的问题,linux 开源社区的研究人员采取了一些方法。他们通过各种方式获得相应芯片的参数,有的研究人员采用了较为复杂的反编译(reverse engineer)技术。在这些研究人员的努力下,某些主流芯片制造商的产品已经能够在 Linux 下被驱动起来。尽管如此,到目前为止还有一部分无线产品不能够被 Linux 平台支持。另外,即使某些无线产品已经可以在 Linux 平台下使用,但是在 Linux 下的驱动还不能够支持这些产品所具有的全部功能。

下面根据搭建过程中的实践经验介绍 Linux 平台下的无线设备和驱动(由于搭建过程中使用到的无线设备仅是无线网卡,因此下面介绍的无线设备专指无线网卡)。

搭建此 Ad - hoc 实验床总共用到五块无线网卡。具体情况如表 1 所示:

表 1 无线网卡型号及对应的芯片制造商

型号	芯片制造商
英特尔 PRO/无线 2200BG	Intel(IPW2200)
英特尔 PRO/无线 2915ABG	Intel(IPW2915)
D - Link DWL G520	Atheros
D - Link DWL 120 +	TI
D - Link DWL - G122 rev. C1	Ralink

下面具体介绍以上五块网卡的安装情况:

- 英特尔 PRO/无线系列

2200BG 和 2915ABG 是属于 Intel 迅驰技术的无线产品。它们常见于具有 Intel 迅驰技术的笔记本计算机中。它们的芯片制造商是 Intel 公司。

Intel 公司支持 Linux 社区,对所生产的无线设备提供 Linux 平台下的驱动。

当前的 Linux 2.6 版内核装有 Intel 公司提供的驱动。如果需要使用这些驱动,应该重新编译内核,从而将相应的驱动程序编译进内核或者编译成可加载模块形式。另一种简便的方法是到 Intel 公司的官方网页上直接下载相应型号的驱动。对于相关的安装与配置操作,可以查看驱动安装包中的帮助文件。

• D - Link DWL G520

D - Link DWL G520 是 D - Link 公司生产的新一代无线网卡。它可以支持 108Mbps 的 802.11a 标准。DWL G520 的芯片由 Atheros 公司制造。

Atheros 公司与 Intel 公司一样,提供对 Linux 平台的支持。通过 <http://madwifi.org/wiki>,可以获得稳定版本的驱动。

• D - Link DWL 120 +

D - Link DWL 120 + 是 D - Link 公司生产的早期无线网卡产品。它支持 22Mbps 的 802.11b + 标准。DWL 120 + 的芯片由 TI 公司生产。

TI 公司不提供 Linux 平台下该公司产品的技术支持。因此,早期在 Linux 平台下不能使用 TI 公司的产品。此后,在 2002 年底,一个二进制版本的 Linux 驱动出现在网络中。尽管如此,该驱动没有官方授权,没有清晰的合法性,并且,它只支持某些特定版本的 Linux 内核。

为解决这些问题,Linux 平台下的研究人员采用复杂的反编译技术,开发出了第一个版本的开源驱动 - ACX100。但是,由于采用反编译技术,所开发的相应驱动可能因为芯片的某些特性不确定而导致功能的不完善。尽管如此,在使用此网卡的过程中,没有出现问题。

另外,在安装此网卡驱动时需要注意,所加载的固件(firmware)在帮助文档中的命名规则较复杂,容易出错。简便的方法是即时查看系统的日志信息,按照日志中提供的线索修改固件的名称。

• D - Link DWL - G122 rev. C1

D - Link DWL - G122 rev. C1 也是 D - Link 公司生产

的无线设备。它支持 54Mbps 的 802.11g 标准。DWL - G122 rev. C1 的芯片由 Ralink 公司生产。

Ralink 公司支持 Linux 社区。在相应的开源社区或者 Ralink 公司提供驱动下载的官方页面中可以获得相应版本的驱动。一般情况下,通过以上两种渠道获得的相应版本驱动支持相同的功能。尽管如此,也存在一些特殊情况。DWL - G122 rev. C1 属于这一情况。在开源社区 <http://rt2x00.serialmonkey.com> 所获得的驱动版本不支持 Ad - hoc 模式,而在官方支持页面中得到的驱动可以支持。

另外,在下载驱动安装包时需要注意,由于芯片的主板型号不同,DWL - G122 rev. C1 与 DWL - G122 rev. B1 的驱动安装包不是同一个。DWL - G122 rev. C1 的驱动包是 RT73_Linux_STA_Drv1.0.3.6,而 DWL - G122 rev. B1 的驱动包是 RT25USB - SRC - V2.0.8.0。

除此之外,在编译安装驱动模块前,需要在驱动安装包的 `rtmp_def.h` 文件中第 810 行左右(具体位置参看相应处代码)添加一行代码。代码如下所示:

```
{USB_DEVICE(0x07d1,0x3c03)}, /* D - Link  
*/ \
```

上面介绍的是在搭建 Ad - hoc 实验床的过程中安装相应无线网卡的情况。为获取更多关于无线设备的信息以及在 Linux 下支持的具体情况请参看 <http://linux-wless.passys.nl>。这个网页提供了多种型号无线网卡的相关信息。

2.2 Linux 下无线设备的配置

Linux 下配置无线设备最主要的工具是“Wireless Extensions for Linux”。

“Wireless Extensions for Linux”是一个针对于 Linux 操作系统的无线网络 API,由 Jean Tourrilhes 在十年前开发。它的主要用途是在 Linux 下对无线设备进行配置,获得有关无线设备的统计信息。当前主流版本的 Linux 默认将其编译进内核。

因为“Wireless Extensions for Linux”的配置命令类似于有线网络的配置命令,因此它被称为“Extensions”。“Wireless Extensions for Linux”主要提供了一个用户接口以及三个相关的配置命令,分别是:`/proc/net/wireless`, `iwconfig`, `iwlist` 和 `iwpriv`。下面分别进行介绍:

- `/proc/net/wireless`

/proc/net/wireless 是一个 /proc 条目。它给出系统中每个无线接口详细的统计信息。它实际上是 /proc/net/dev 的克隆。通过 cat /proc/net/wireless 命令可以查看相应的统计信息。这个接口主要用于编程中。

• iwconfig

iwconfig 是配置无线设备最常用的命令之一。通过 iwconfig 命令可以配置设备最重要的几个参数。图 1 显示了使用 iwconfig 命令的情况。

```
[root@xmr ~]# iwconfig ath0 essid xmr
[root@xmr ~]# iwconfig
lo        no wireless extensions.

eth0     no wireless extensions.

wifi0    no wireless extensions.

sit0     no wireless extensions.

ath0     IEEE 802.11b  ESSID:"xmr"
Mode:Ad-Hoc  Channel:0  Cell: Not-Associated
Bit Rate:0 kb/s  Tx-Power:0 dBm  Sensitivity=0/3
Retry:off  RTS thr:off  Fragment thr:off
Encryption key:off
Power Management:off
Link Quality=0/94  Signal level=-95 dBm  Noise level=-95 dBm
Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
Tx excessive retries:0  Invalid misc:0  Missed beacon:0
```

图 1 iwconfig 命令截图

从图 1 中可以看到通过 iwconfig 命令得到的网络接口设备的参数统计信息。这些统计信息来源于 /proc/net/wireless。

图 1 中示出的多项条目中,需要关注的是 ESSID 和 Mode。

ESSID 是无线网络的逻辑 ID。通过它可以区分在同一信道中的不同无线网络。

Mode 是无线网络的模式。在无线网络中常用的两种模式分别是 Managed 和 Ad-Hoc。其中,Managed 模式即 AP 模式。

配置这两个参数可以仿照图 1 中第一行的形式。如果需要配置其它参数,也可以利用这种方式。但一般情况下不需要。这主要是因为对于其它参数,网络设备会自动选择。因此,指定了 ESSID,一般的网络设备会自动配置好相应的参数以连接到指定的网络中。

• iwlist

iwlist 也是配置无线设备常用的命令。一般情况下,使用这一命令查看当前所在区域内无线网络的状态以及系统内无线设备信道的情况。图 2 显示了使用 iwlist 命令的情况。

态以及系统内无线设备信道的情况。图 2 显示了使用 iwlist 命令的情况。

```
[root@xmr ~]# iwlist ath0 scan
ath0     Scan completed :
          Cell 01 - Address: 00:0E:D7:4F:C4:50
          ESSID:"CSTNET-2"
          Mode:Master
          Frequency:2.437 GHz (Channel 6)
          Quality=22/94  Signal level=-73 dBm  Noise level=-95 dBm
          Encryption key:off
          Bit Rates:1 Mb/s; 2 Mb/s; 5.5 Mb/s; 6 Mb/s; 9 Mb/s
                   11 Mb/s; 12 Mb/s; 18 Mb/s; 24 Mb/s; 36 Mb/s
                   48 Mb/s; 54 Mb/s
          Extra:bcn_int=100
```

图 2 iwlist 命令截图

从图 2 中可以看到通过 iwlist 命令发现了当前区域内有一个 ESSID 为“CSTNET-2”的 AP 模式的无线网络。为连接到这个网络,可以利用已介绍的 iwconfig 命令将相应设备的 ESSID 设置成“CSTNET-2”。

* iwpriv

iwpriv 命令也比较常用。这一命令的具体使用需要参考设备驱动包中的说明文档。不同的无线设备芯片生产厂家对于 iwpriv 命令各种参数的定义是不同的。根据命令名中的“priv”,也可以理解这一情况。

另外需要说明的是,在配置不同厂商生产的无线设备时,需要参看设备驱动包中提供的说明材料。尽管无线设备厂商支持“Wireless Extensions for Linux”这一配置工具,但也存在一些情况,需要使用厂商提供的特定工具才可以配置无线设备。例如,在配置 D-Link DWL G520 这块网卡时,通过“iwconfig ath0 mode ad-hoc”不能将此设备配置成 Ad-hoc 模式,而需要通过厂商提供的“wlanconfig ath0 create wlanmode wifi0 wlanmode adhoc”这一命令。这些特有的配置工具及命令一般在相应的帮助文件中有所说明。

3 Ad-hoc 路由协议

当前被设计出的 Ad-hoc 路由协议有几十种。其中一些主流的并且已通过编码实现的也有十几种。为测试实验床,需要从这些版本中选定一个。

最终,根据 MANET 工作组提供的 RFC 文档和 AODV@ IETF project,选定使用 AODV 协议的两个实现版本 AODV-UU 和 WINDOWS-AODV 对搭建的 Ad-hoc 实验床进行测试。

3.1 MANET 和 AODV

MANET (MoBie Ad-hoc Networks) 是 IETF 为研究 Ad-hoc 网络的路由协议专门成立的工作组。到目前为止, MANET 总共发布了五个 RFC 文档, 分别是: Routing Protocol Performance Issues and Evaluation Considerations (RFC 2501), Ad Hoc On Demand Distance Vector (AODV) Routing (RFC 3561), Optimized Link State Routing Protocol (RFC 3626), Topology Dissemination Based on Reverse-Path Forwarding (TBRPF) (RFC 3684), 以及 The Dynamic Source Routing Protocol (DSR) for MoBie Ad Hoc Networks for IPv4 (RFC 4728)。其中, RFC2501 是一个评判 Ad-hoc 路由协议的标准, 而其余四个 RFC 是真正的 Ad-hoc 路由协议。

在已发布的这四个路由协议中, AODV 协议是最早的。并且多数实现 AODV 协议的软件比较完善。所以确定选用 AODV 协议测试实验床。

3.2 AODV@ IETF project

在选择具体的 AODV 协议实现软件时, 参考了 AODV@ IETF project。AODV@ IETF project 是由 UCSB (University of California, Santa Barbara) 和 Intel R&D 的两个实验室 MOMENT 和 NMSL 共同承担的。此项目的目标是建立一个大规模的公众使用的 AODV Ad-hoc 网络。

在此项目的网站中提供了一些实现 AODV 路由协议软件的链接, 其中被推荐使用的是 KERNEL-AODV (NIST Implementation) 和 AODV-UU (Uppsala University Implementation)。这两种版本都与 RFC3561 兼容。

尽管 KERNEL-AODV 是完全内核态的实现, 但是由于它的版本只更新到 2004 年 7 月, 并且不支持 Linux 2.6 版内核, 所以没有选用它。而是选用了另一版本 AODV-UU。不过, AODV-UU 不是全内核态的, 因此性能上会表现不佳。

除此之外, AODV@ IETF project 的网站上也提供了 Windows 环境下的 AODV 实现链接。因为所搭建的实验床中也包括一台 Windows 系统的节点, 所以另外选择了 AODV 在 Windows 下的实现软件 WINDOWS-AODV。

4 Ad-hoc 实验床的调试与结果分析

试验环境为一个长 100 米, 宽 30 米的办公区域。在该区域内搭建一个由两台笔记本和三台台式机组成

的 Ad-hoc 实验床, 并在此实验床上实现跨越多个节点的多跳文件传输应用。下面介绍调试的过程以及对于实验结果的分析。

4.1 调试过程

组成实验床的设备主要包括两台 IBM T43 笔记本, 两台 Acer 品牌台式机以及一台 Dell 品牌台式机。各设备具体的配置情况如表 2 所示:

表 2 Ad-hoc 实验床设备配置表

名称	无线网卡型号	操作系统版本
IBM T43-1	英特尔 PRO/无线 2200BG	WindowsXP SP2
IBM T43-2	英特尔 PRO/无线 2915ABG	ubuntu 6.06 (内核: 2.6.17.8)
Acer PC-1	D-Link DWL G520	Fedora Core 5 (2.6.16.26)
Acer PC-2	D-Link DWL 120+	ubuntu 5.10 (2.6.17.8)
Dell PC	DWL-G122 rev. C1	Fedora Core 5 (2.6.16.26)

具体的调试过程如下:

首先将整个网络搭建起来。经过一段时间, 待网络稳定之后, 将安装在 Acer PC-1 和 IBM T43-1 两个节点上的 ftp 服务器启动。而后, 运行各节点的 ftp 终端软件, 分别与两个 ftp 服务器相连, 上传和下载文件, 观察文件传输的情况。同时, 在 Linux 系统节点上运行 ping-R 命令, 观察相应节点的路由情况。另外, 还需要随时察看 AODV-UU 和 WINDOWS_AODV 显示的整个网络的路由情况。

在实验床的调试过程中遇到以下几个问题:

* IBM T43-1 与 Acer PC-1 不能连通

将 IBM T43-1 的无线网卡英特尔 PRO/无线 2200BG 和 Acer PC-1 的无线网卡 D-Link DWL G520 分别在 Windows 和 Linux 下配置后启动, 两个节点不能互相 ping 通。但是如果存在第三个节点完成配置并启动, 前述两个节点就可以 ping 通。

* D-Link DWL 120+ 与 D-Link DWL G520 冲突

在安装配置各节点的无线网卡时, 如果先将 Acer PC-2 上的 D-Link DWL 120+ 启动, 那么 Acer PC-1 上的 D-Link DWL G520 就无法完成相关的配置。但是, 如果卸掉 D-Link DWL 120+, 先配置启动 D-Link DWL G520 就没有问题。

以上两个问题产生的原因都应该来源于 Linux 对于无线设备支持的不完善。

* Fedora Core 5 系统的节点不能 ping

在实验床上运行 AODV 路由协议之后,原本能够 ping 通其它节点的 Fedora Core 5 系统的两个节点不能够再 ping 通其它节点。解决的办法是关掉这两个节点上自带的防火墙。

4.2 实验结果分析

尽管最终在实验床上实现了多跳的文件传输应用,但是实验结果不是十分令人满意。在进行文件传输应用中,实验床的丢包率较高,难以保证大文件的传输成功。

从截取的实验现象中发现,传输大文件的多次尝试都失败了。大文件在传输开始时比较正常,但经过一段时间便会与服务器断开连接。

产生上述问题的原因应该来源于以下几个方面:一是实验床运行的环境不稳定,这与无线通信的本质特性有关;二是实验床上跑的数据量过大使得 AODV 协议不能承受;三是 AODV 协议本身或者是实现 AODV 协议的软件,即 AODV-UU 和 WINDOWS-AODV 的问题。

5 结束语

本文根据工程实践,对基于 Linux 平台搭建 Ad-hoc 实验床的各方面情况进行了系统介绍。希望这些内容对从事相关领域的研究人员有所帮助。

另外,还需要提醒正着手准备搭建 Ad-hoc 实验床的工作人员以下两点:

首先,搭建这样一个实验床,在不是特别了解各种无线产品的性能以及具体的使用情况之下还是应该选用“同样”的节点。这里的“同样”是指:同型号同版本的操作系统;同厂家同型号的无线网卡;同版本的无线网卡驱动;同版本的 Ad-hoc 路由协议软件。由于在上述几个方面中没有形成同一的标准,所以在实验床的搭建和使用过程中可能会遇到不兼容问题。

其次,对于 Ad-hoc 实验床的搭建,最好还是选用可移动的设备,如 PDA。尽管若干台的台式机和笔记本计算机是需要的,但这些设备不应该是 Ad-hoc 网络的主角。众所周知,Ad-hoc 网络是可移动的。这里的“可移动性”是指在一定速度下的移动,网络仍会保持连通。但是只利用台式机和笔记本计算机是体验不到这种巨大的优越性的。

参考文献

- 1 Jean Tourrilhes. Linux Wireless LAN Howto. 2005. <http://www.tldp.org/>
- 2 郑相全等. 无线自组网技术实用教程. 北京:清华大学出版社,2004.
- 3 师若鸣,姜中华,刘在强等. 红帽 Fedora Core 2 系统配置与管理. 北京:清华大学出版社,2005.