基于开源代码与嵌入式 Xscale 270 的无线 Mesh 网络实验床①

Wireless Mesh Network Testbed Based on Open Source Code and Embeded Xscale 270

张美平　许　力　(福建师范大学 网络安全与密码技术福建省高校重点实验室 福建 福州 350007)

摘　要： 提出了一种应用嵌入式 Xscale 处理器与嵌入式 Linux、IEEE 802.11b 无线网卡设计无线 Mesh 网络实验床平台的方法，应用 Liod-s 嵌入式开发板设计并实现了无线 Mesh 网络实验床 FJNU-Mesh。由于嵌入式 linux 的开源特性，该实验床能大大促进无线 Mesh 网络的理论研究与应用开发工作。

关键词： 嵌入式 Linux IEEE 802.11b 无线 Mesh 网络 实验床

1　无线Mesh网络简介

无线 Mesh 网络是一种新型的无线网络架构，它的核心指导思想是让网络中的每个节点都可以发送和接收信号，无线 Mesh 网络也称为“多跳(multi-hop)”网络，它是一种与传统无线网络(WLAN)完全不同的新型无线网络技术。在传统的无线局域网中，每个客户端均通过一条与 AP 相连的无线链路来访问网络，用户如果要进行相互通信的话，必须首先访问一个固定的接入点(AP)，这种网络结构被称为单跳网络。而在无线 Mesh 网络中，任何无线设备节点都可以同时作为 AP 和路由器，网络中的每个节点都可以发送和接收信号，每个节点都可以与一个或者多个对等节点进行直接通信。无线 Mesh 网络技术的出现，代表着无线网络技术的又一大跨越，有极为广阔的应用前景，传统的 WLAN 一直存在的可伸缩性低和健壮性差等诸多问题由此迎刃而解。

无线 Mesh 网络由 Mesh 路由器(Mesh Router)、Mesh 客户端(Mesh Client)构成。Mesh 路由器可以是普通 PC，也可以是专用的嵌入式系统，如 ARM 等。位于网络边缘的 Mesh 无线路由器将兼具有因特网网关的功能。Mesh 路由器除了具有传统无线路由器中网关/网桥功能的路由能力外，还包含特殊的路由功能以支持 Mesh 组网。客户节点可以是笔记本电脑、PDA、Wi-Fi 手机、RFID 阅读器和无线传感器或控制器等；客户节点按照功能可以分为两类：一类只作为普通终端接入网络，不具有转发信息的功能；另一类既具有普通节点的接入功能，又具有路由和信息转发功能，即兼具了无线路由器的功能。

按照结构层次，无线 Mesh 网络可以分为平面结构、多级结构和混合结构。图 1 为多级结构的无线 Mesh 网络。

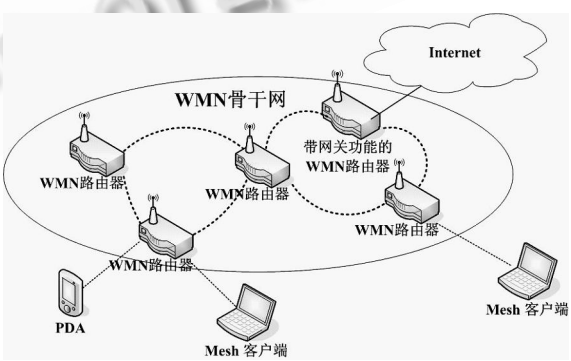

图 1　多级结构的无线 Mesh 网络

福建师范大学网络安全与密码技术福建省高校重点实验室，应用嵌入式 Xscale 处理器、嵌入式 linux

① 基金项目:国家自然科学基金(60502047);福建省教育厅基金(JA07043)
收稿时间:2008-12-28

与开源 Mesh 网络路由协议 OLSR[1]，实现了基于嵌入式 Linux 操作系统的无线 Mesh 路由器，完成了无线 Mesh 网络实验床(FJNU-Mesh)的建设，为进一步进行无线 Mesh 网络的理论研究与应用开发打下了坚实的基础。

2 FJNU-Mesh实验床体系结构

FJNU-Mesh 实验平台使用的硬件平台为 Liod-s 嵌入式开发板，其结构框图如图 2 所示。该平台使用 Intel Xscale PXA270 处理器、主频运行于 520MHz、采用 32M Flash 与 64M SDRAM。Liod-s 平台配备了众多的外围接口，提供了 SPI 接口、USB 接口、CF 接口、MMC/SD 接口，通过 LAN91C111 扩展以太网接口，还通过 UCB1400 提供触摸屏接口。

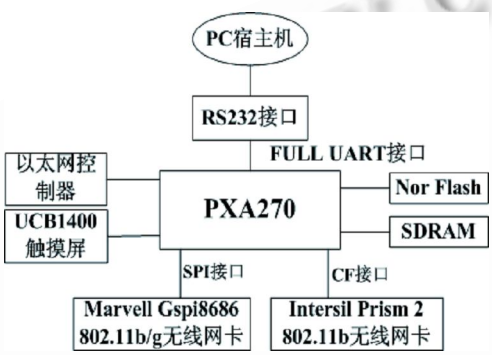

图 2　FJNU-Mesh 路由器节点硬件结构

FJNU-Mesh路由器使用两个IEEE802.11 的无线网卡，其中通过 SPI 接口外接了 Marvell GSPI8686 的 IEEE802.11b/g 网卡，通过该网卡实现与其他 Mesh 路由器间的 Ad-Hoc 多跳通信，使用工业级开源无线 Mesh 路由协议 OLSR[1]实现路由表的维护与更新。同时通过 PCMCIA/CF 接口外接了 Intersil Prism 2 CF 接口的 IEEE 802.11b 无线网卡，并应用开源的 HostAP 无线网卡驱动程序，来实现 Mesh 路由器的 IEEE 802.11 Access Point 功能，允许 Mesh client 通过该 AP 接入到无线 Mesh 网络中。图 3 为该 Mesh 路由器的软件体系结构图。

3 FJNU-Mesh实验床的设计与实现

FJNU-Mesh 实验床系统的设计包括嵌入式系统的内核编译与文件系统的建立、CF 接口的 IEEE802.11b 无线网卡的 HostAP 驱动与配置、SPI 接口 Marvell GSPI8686 网卡驱动，OLSR 路由协议的移植与编译，启动实验床的 shell 脚本设计等几个方面。

Mesh路由器需要同时与其他主干路由器通信又需要接入Mesh客户	使用双无线网卡 SPI接口GSPI8686 CF接口Interlsl Prlsm2
Mesh客户端接入	HostAP(Master模式)
路由器间多跳路由	OLSR(Ad-Hoc模式)
嵌入式Linux-2.6.9	
基于Xscale270的Liod-s嵌入式开发平台	

图 3　FJNU-Mesh 路由器软件体系结构

3.1 CF 接口无线网卡 HostAP 驱动移植

FJNU-Mesh 路由器节点使用的 Liod-s 开发版提供的 linux-2.6.9_270-s 的内核，该内核提供了 PCMCIA/CF 接口、SPI 接口、但没有提供 CF 接口的无线网卡的 HostAP 驱动，需要移植 HostAP 驱动。

从 HostAP[2]的官方网站下载 HostAP 的驱动程序 hostap-driver-0.4.9.tar.gz，由于 hostap 0.4.9 需要内核使用 Wireless Extensions v17，而 linux-2.6.9_270-s 内核中的 Wireless Extensions 版本为 WEV16，需要给内核打上 WEV17 的补丁，可以从 Wireless Tools for Linux[3]网站下载 Wireless Extensions v17 for linux-2.6.9 的补丁文件 iw269_we17-12.diff。假设 Liod-s 内核解压后目录为/root/work/liod-s/linux-2.6.9_270-s，下载后的补丁及其驱动文件均放在/root/work/liod-s/中，按以下步骤添加 linux 内核驱动补丁：

(1) 安装 HostAP 驱动。

cd　/root/work/liod-s/;

tar　zxvf　hostap-driver-0.4.9.tar.gz;

cd　linux-2.6.9_270-s;

patch-p1< ../hostap-driver-0.4.9/kernel-patches/hostap-linux-2.6.2.patch;

cp　../hostap-driver-0.4.9/driver/modules/hostap*.[ch]　drivers/net/wireless;

说明：以上步骤分别完成进入 liod-s 工作目录、解压 hostap 驱动、给内核打上 hostap 驱动配置补丁、复制相应的驱动文件到内核无线网络设备的驱动目录中。

(2) 安装 Wireless Extensions v17 for linux-2.6.9 的补丁。

cd /root/work/liod-s/linux-2.6.9_270-s;

patch -p1 < ../iw269_we17-12.diff;

(3) 修改 PCMCIA Card Services pcmcia_cs 的配置文件，实现自动驱动 CF 网卡。

在嵌入式 linux 中，CF 接口设备驱动程序的加载通过 pcmcia_cs 软件包来实现。liod-s 开发板提供的文件系统镜像 rootfs 中提供了 pcmcia_cs 的软件包。为了能在开机启动 pcmcia 服务时能自动使用 hostap 来驱动该 CF 接口的无线网卡或开发板运行过程中接上 CF 无线网卡后就能自动识别到网卡的型号并向内核启动加载对应的 CF 无线网卡 hostap 驱动，需要修改 pcmcia_cs 的配置文件 hostap_cs.conf，该文件位于 rootfs 的/etc/pcmcia/目录中。

修改 hostap_cs.conf 添加对应的驱动配置，需要了解系统使用的 CF 无线网卡的 ID 号。先把 CF 无线网卡接入到开发板，使用 cardctl ident 命令找出无线网卡的相关信息及其对应 ID 号，然后在根据网卡信息修改 hostap_cs.conf 文件。本实验床使用的 CF 网卡的信息为以下内容：

Socket 0:

product info: "Intersil", "PRISM Freedom PCMCIA Adapter", "ISL37100P", "Eval-RevA"

manfid: 0x000b, 0x7100

function: 6 (network)

即本系统使用的 CF 接口的无线网卡为 ID 为 0x000b,0x7100，其型号为 ISL37100P，根据该网卡信息，把以下内容添加到 hostap_cs.conf 文件中，从而 pcmaia_cs 能识别到该设备并自动加载 hostap_cs.ko 驱动模块。

```
card "Wireless CompactFlash Card"
  version "Intersil","PRISM Freedom PCMCIA Adapter", "ISL37100P", "Eval-RevA"
  manfid 0x000b,0x7100
  bind "hostap_cs"
```

3.2 Marvell GSPI8686 无线网卡的驱动

Liod-s 开发板提供的 linux 2.6.9 驱动程序中，提供了 Marvell GSPI8686 无线 IEEE 802.11b/g 网卡的驱动，复制对应的驱动程序源代码，修改编译配置文件 Makefile，主要修改对应内核源代码目录的设置，设置参数 KERNELDIR=/root/work/liod-s/linux-2.6.9_270-s，执行 make 命令编译后产生 gspi.ko、gspi8xxx.ko。其中 gspi.ko 为 spi 接口的驱动，gspi8xxx.ko 为 IEEE802.11b/g 芯片的驱动。把编译后的 gspi.ko、gspi8xxx.ko，以及驱动程序 FwImage 目录下的两个芯片固件驱动 gspi8686.bin、helper_gspi.bin 4 个文件复制到开发板的文件系统 rootfs 目录的/wifi 子目录中，并在该目录下建立一个 wifi.sh 的 shell 脚本文件，该脚本文件的作用是驱动 gspi8686 的无线网卡，内容如下：

```
insmod gspi.ko
insmod gspi8xxx.ko helper_name=/wifi/helper_gspi.bin fw_name=/wifi/gspi8686.bin
```

3.3 嵌入式 linux 内核编译

内核是整个实验床 Mesh 路由器 Linux 操作系统的核心，在配置内核的过程(make menuconfig 步骤)中需要配置 SPI 接口驱动、PCMCIA 接口驱动、Networking support、Packet Socket 、Netfilter、Ip_Queue、Netlink device(以上这几项均选择编译进核心)、CF 接口无线网卡的 HostAP 驱动(该项选择编译为模块)。

按以下步骤完成配置 linux-2.6.9 的内核，编译内核与模块。

```
cd /root/work/liod-s/linux-2.6.9_270-s;
make mrproper;
make clean;
make xsbase270edr_defconfigs;
make menuconfig;
make zImage;
make modules;
make modules_install;
```

其中，make xsbase270edr_defconfigs 为开发板内核提供的一个默认配置；在执行完 make zImage 后，在 /root/work/liod-s/linux-2.6.9_270-s/arch/arm/boot/ 目录中有编译后的内核文件 zImage；执行完 make modules_install 后在宿主机系统的/lib/ modules/目录中存在一个目录 2.6.9-

intc1 的子目录，该目录就是开发板的内核模块目录，需要把该目录复制到到开发板文件系统镜像对应的/lib/modules 目录中。

3.4 Mesh 路由协议 olsrd 的编译与配置

OLSR[1](Optimized Link State Routing Protocol 优化链路状态路由算法)是由 IETF MANET 工作组提出的一种先应式的链路状态路由协议，是一种基于 LS(link state)的表驱动路由协议。OLSR 是相当成熟的适用于无线 Mesh 网络的路由协议，它通过采用 MPR(multipoint relays)技术，减少了洪泛过程中不必要的消息传递，减少了开销，从而提高了效率。可以从 http://sourceforge.net/projects/olsrd/下载到最新的协议源码 olsrd-0.5.6.tar.gz，编译时只需要指定交叉编译链 CC=arm-linux-gcc 即可编译出适用于嵌入式 linux 的守护进程 olsrd。

要使 mesh 路由器中的 olsrd 能正确工作，需要配置 olsrd 的配置文件 oldrd.conf，该文件主要定义了 mesh 路由器中参与无线 mesh 骨干网络间通信的接口、IP 协议版本、需要参与交换到其他到 mesh 路由器的本地非 mesh 接口网络地址或主机地址信息。一个简单的 olsrd.conf 的文件如下:

```
DebugLevel  0
IpVersion     4
FIBMetric     "flat"
Hna4 {
192.168.21.0     255.255.255.0
}
Interface "eth1" {
AutoDetectChanges   yes
Ip4Broadcast        255.255.255.255
}
```

该配置定义了当前 mesh 路由器的地址协议为 ipv4，本地网段 192.168.24.0 将交换到其他的 mesh 路由器的路由表，eth1 接口为该 mesh 路由器与其他 mesh 路由器多跳互连的无线网络接口。

3.5 Mesh 路由器的启动脚本

为了使 liod-s 开发板启动后实现 mesh 路由器的功能，需要依次完成以下几个步骤:

(1) 启动配置无线 CF 网卡为 AP 接入。

由于配置好了 pcmcis_cs 的相关配置，CF 接口的无线网卡能自动驱动，不需要手动启动该网卡(该网卡启动后设备号识别为 wlan0)，所以只需要配置该无线网卡的 IP 地址、设置工作模式为 master、工作 channel、该 AP 所处的 essid。

(2) 启动配置 gspi8686 无线网卡为 mesh 接入。

运行/wifi/wifi.sh 启动脚本启动 gspi8686 的无线网卡(该网卡启动后设备号识别为 eth1)，需要配置该无线网卡的 IP 地址，设置工作模式为 ad-hoc，工作 channel，该 AP 所处的 essid。在该接口上启动 mesh 路由进程 olsrd。

(3) 在 Mesh 路由器上配置 olsrd.conf，并启动 olsrd 路由协议守护进程

为了使各节点能快速启动 mesh 路由功能，把上述各个步骤步骤编写成一个运行脚本/sbin/olsr.sh，为了让系统开机能自动执行启动脚本，可以把命令添加到/etc/rc.d/rc.local 文件，使之开机自动执行 olsr.sh 启动脚本。下面列出某一个节点的启动脚本。

```
ifconfig    wlan0   192.168.21.1
iwconfig    wlan0   mode   master
iwconfig    wlan0   channel   1
iwconfig    essid   mesh-ap1
/wifi/wifi.sh              #启动 gspi8686 网卡#
sleep 10    #使系统暂停 10m，等待网卡启动#
ifconfig    eth0    down     #关闭 eth0 接口#
ifconfig    eth1 192.168.3.11
iwconfig    eth1 mode ad-hoc
iwconfig    eth1 essid mesh-olsr
iwconfig    eth1    channel   6
olsrd #根据/etc/olsrd.conf 的配置启动 olsrd#
```

3.6 Mesh 路由器 flash 的烧写

把各个 Mesh 路由器的相关的配置文件与相关的配置参数设置好后制作出的各个节点的 rootfs.img，编译的内核 zImage 通过 tftp、bootloader 下载烧写的开发板 flash 中。

4 FJNU-Mesh实验床运行测试

采用基于 Liod-s 平台搭建了 4 个 Mesh 路由节点组成的测试网络，拓扑结构如图 4 所示，其中 Mesh 路由器的 eth1 无线网络接口用于与其他 Mesh 路由器进行数据回传，所有的节点的 eth1 接口均配置在

同一个 IP 网段 192.168.3.0，并且工作于 ad-hoc 模式；采用了 hostap 驱动的 wlan0 接口，工作于 master 模式，允许 Mesh 客户机 AP 接入。

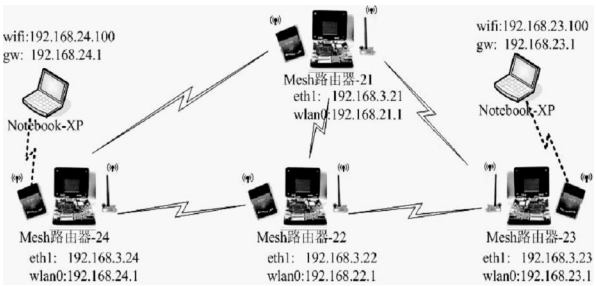

图 4　测试实验床拓扑结构图

由于 IEEE 802.11b 只能提供 11 个频段，其中只有频段 1、6、11 可以同时使用且彼此不会相互干扰。为了防止节点间无线频段的干扰，用于 ap 接入的 wlan0 分别使用 channel 1、channel 11 频段，而用于与其他 mesh 路由器数据回传的 eth1 接口使用 channel 6。

按实验测试拓扑图的地址规划分别配置各个节点的 olsr.sh、olsrd.conf，然后分别在各个节点启动 olsrd。各节点的 olsrd 协同完成该 Mesh 网络的路由表更新。分别在各个节点使用 route 查看路由表，图 5、图 6 分别为测试网络中各 Mesh 路由器节点的路由表。

图 5 为 Mesh 路由器-21 的路由表，从该节点出发到达其他的的 mesh 路由器，其他的网段的下一跳地址均为 192.168.3.22，即从 Mesh 路由器-22 出发可以到达网络中的其他网段。

```
[root@liod-s-21 ~]# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.3.23    192.168.3.22    255.255.255.255 UGH   2      0        0 eth1
192.168.3.22    *               255.255.255.255 UH    2      0        0 eth1
192.168.3.24    192.168.3.22    255.255.255.255 UGH   2      0        0 eth1
192.168.23.0    192.168.3.22    255.255.255.0   UG    2      0        0 eth1
192.168.22.0    192.168.3.22    255.255.255.0   UG    2      0        0 eth1
192.168.21.0    *               255.255.255.0   U     0      0        0 wlan0
192.168.3.0     *               255.255.255.0   U     0      0        0 eth1
192.168.24.0    192.168.3.22    255.255.255.0   UG    2      0        0 eth1
127.0.0.0       *               255.0.0.0       U     0      0        0 lo
```

图 5　Mesh-路由器-21 节点的路由表

```
[root@liod-s-23 ~]# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.3.21    192.168.3.22    255.255.255.255 UGH   2      0        0 eth1
192.168.3.22    *               255.255.255.255 UH    2      0        0 eth1
192.168.3.24    *               255.255.255.255 UH    2      0        0 eth1
192.168.23.0    *               255.255.255.0   U     0      0        0 wlan0
192.168.22.0    192.168.3.22    255.255.255.0   UG    2      0        0 eth1
192.168.21.0    192.168.3.22    255.255.255.0   UG    2      0        0 eth1
192.168.3.0     *               255.255.255.0   U     0      0        0 eth1
192.168.24.0    192.168.3.24    255.255.255.0   UG    2      0        0 eth1
127.0.0.0       *               255.0.0.0       U     0      0        0 lo
```

图 6　Mesh-路由器-23 节点的路由表

5　结语

本方案设计的无线 Mesh 实验平台 FJNU-Mesh，能通过多个 liod 节点动态组网，目前改实验床平台能稳定运行。基于嵌入式 Xscale270、嵌入式 linux 与开源 olsr 设计的设计的无线 mesh 网络，由于其开源特性能能极大地方便研究人员在该平台上进行无线 Mesh 网络的应用开发与协议研究。

参考文献

1 A wireless mesh routing daemon. http://www.olsr.org, [2008-05-31].

2 Host AP driver for Intersil Prism2/2.5/3 wireless LAN cards.http:// hostap.epitest.fi/,[2008-06-28].

3 Wireless tools for linux.http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html,[2008-05-24].

4 郑灵翔.嵌入式系统设计与应用开发.北京:北京航空航天大学出版, 2006.