

# 基于嵌入式 Linux 的无线自组织网络实验床设计<sup>①</sup>

## Design of Wireless Ad Hoc Network Testbed Based on Embedded Linux

张美平 许力 沈金波 (福建师范大学 网络安全与密码技术实验室 福建 福州 350007)

**摘要:** 目前对于无线自组织网络的研究一般使用模拟仿真软件如 NS-2、OPNET 等来进行实验。仿真模拟的方法较适合路由协议算法设计方面的研究;而对于一些网络安全、入侵检测、应用层协议等一些需要实际数据传输与实际算法运行的研究,使用仿真软件就显得力不从心。本文提出了一种应用 Xscale 处理器与嵌入式 Linux、开源路由软件 aodv-uu,设计并实现了一个基于 IEEE 802.11b 的无线自组织网络实验床系统。该实验床能大大促进无线自组织网络的研究工作。

**关键词:** 嵌入式 无线自组织网络 实验床

### 1 无线自组织网络实验床简介

对于无线自组织网络的研究目前一般使用模拟仿真的方式来进行,如使用 NS-2、OPNET 等软件来进行仿真研究,仿真模拟的方法较适合于路由协议算法设计方面的研究;而对于一些网络安全、入侵检测、应用层协议等一些需要实际数据传输与实际算法运行的研究,使用仿真软件就显得力不从心,所以设计一种能真实体现其无线自组织网络的特性、具备实际数据传输能力、并具有方便编程接口的实验床平台就显得特别重要。

嵌入式 Linux 是一种可运行于嵌入式处理器的开源操作系统,能运行于多种嵌入式处理器,同时具备方便的编程接口与广泛的外设驱动能力。使用嵌入式 Linux 与 ARM 处理器设计实现的无线自组织网络实验床平台将在一定程度上促进无线自组织网络在网络安全与应用层协议等方面的研究工作。

## 2 Lioid XSBASE270 平台与 aodv-uu 简介

### 2.1 Lioid XSBASE270 平台简介

本无线自组织网络实验床系统使用的硬件平台为 Lioid XSBASE270,其功能结构框图如图 1 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。Lioid XS-

Base270 采用核心板加底板的设计思想,平台配备了众多的外围接口,提供了 USB 接口、CF 接口、MMC/SD 接口,同时还通过 LAN91C111 扩展以太网接口,还通过 UCB1400 提供触摸屏接口。通过 CF 接口,可以外接 802.11b 无线网卡。

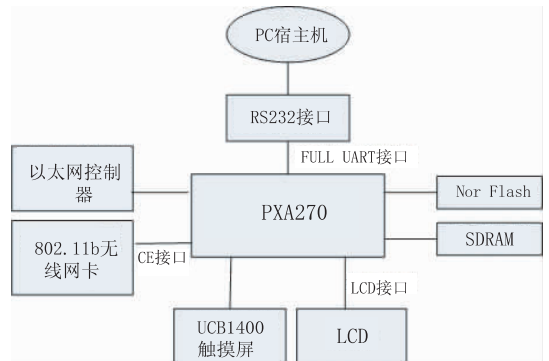


图 1 Xscale PXA270 系统结构

### 2.2 aodv-uu 简介

aodv-uu<sup>[1]</sup>是瑞典乌普萨拉大学与爱立信公司基于 GPL 联合发布的一种无线自组织网络路由协议,其协议运行机制如图 2 所示。

aodv-uu 采用一个运行用户空间的守护进程 aodvd 与一个运行于内核空间的驱动模块 kaodv.o 相结合的方式来实现路由协议的各项路由触发事件与路由实现逻辑。用户空间守护进程 aodvd 使用 linux 提

① 基金项目:福建省教育厅基金资助项目(JA07043)

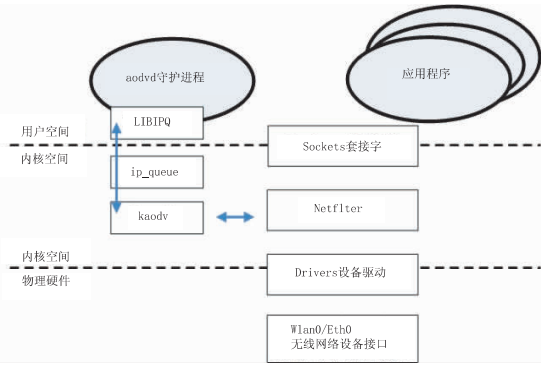


图 2 aodv-uu 的体系结构

供的虚拟接口 Netlink 通过 iq\_queue 提供的 LIBIPQ 库与运行于内核空间的模块 kaodv.o 进行通信。

内核模块 kaodv.o 应用 Netfilter 提供的钩子,编写对应的用户自定义代码来重定向数据分组,接受从本机 (NF\_IP\_LOCAL\_OUT)、从别的机器 (NF\_IP\_PRE\_ROUTING)、还有所有发送到其他机器 (NF\_IP\_POST\_ROUTING) 的分组。应用 iq\_queue 把这些数据包排队发送到用户空间的路由守护进程 aodvd。

Netfilter 是由 linux TCP/IP 协议栈内部许多钩子点的组成一组过滤子系统,其钩子的位置关系如图 3 所示。Netfilter 允许用户自定义内核模块来注册并实现这些钩子的回调函数,Netfilter 中的钩子可以把网络分组重定向到用户自定义的代码,在自定义的代码中能为用户空间的守护进程检测、丢弃、修改或者排队这些分组。

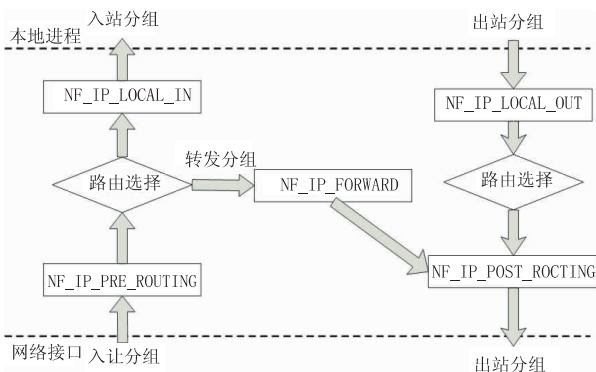


图 3 netfilter 钩子函数在协议栈中的位置关系

路由守护进程 aodvd 主要实现路由协议的实现逻辑,使用 libipq 来对每个包做控制决定,从而来决定何时需要触发相应的路由事件。

### 3 系统设计

整个实验床系统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 802.11b 无线网卡的驱动、aodv-uu 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。

#### 3.1 linux 内核编译

内核是整个实验床节点的操作系统的核心,在编译内核的过程中需要配置 CF 接口无线网卡的驱动,同时还要添加 aodv-uu 内核模块所需要的 Netfilter、Ip\_Queue、Netlink device 等模块的驱动,按如下的步骤进行内核的配置与编译:

- (1) 解压 Liod 开发板提供的内核源代码压缩包 linux-2.4.21-51Board\_DVK.tar.gz,假定内核解压后目录为/work/liod-aodv/linux-2.4.21-51Board\_DVK;
- (2) 进入到内核目录,整理内核源代码;  
make mrproper
- (3) 修改内核源码的配置文件 Makefile,设置好交叉编译器参数 CROSS\_COMPILE 与处理器体系结构参数 ARCH。其中 CROSS\_COMPILE 为系统安装 arm-linux-gcc 的路径及其编译器的前缀;
- (4) 载入针对 Liod 开发板平台预先设置的配置文件 arch/arm/def-configs/xsbase270;
- (5) 进入内核配置窗口进行内核配置,配置结束后保存退出,配置结果将保存在.config 文件中;

make menuconfig

在内核配置过程中需要注意以下几个问题,由于 aodv-uu 的内核模块 kaodv.o 需要使用到 Netfilter、Netlink、Ip\_Queue 等功能。在内核配置过程中需要把以下几项网络功能编译进内核。在内核配置菜单 Networking options - - - > 的子菜单中选中 [\*] Netlink device emulation 与 [\*] Network packet filtering (replace ipchains),在 Networking options - - - > IP: Netfilter Configuration - - - > 配置菜单中选中 [\*] Userspace queueing via NETLINK,这些项目均配置为编译到内核。

由于本系统使用 ID 为 0x015b:0x0002 的 802.11b CF 接口无线网卡,该无线网卡使用 orinico\_cs 的驱动程序。在 XSBase270 系统中 CF 接口通过 PCMCIA 接口

电路驱动,在内核配置菜单中选中对应的 wireless 网卡驱动。在内核配置主菜单 Network device support --> Wireless LAN (non-hamradio) --> 的子菜单中选中 [M]Hermes chipset 802.11b support (Orinoco/Prism2/Symbol) 与 [M]Hermes PCMCIA card support,这两项均配置为编译成模块驱动形式。

(6) 开始编译内核与模块;

```
make dep
make zImage
```

其中 make zImage 执行成功后将在 arch/arm/boot/ 目录下产生 zImage 文件,这个就是新编译好的内核文件。

(7) 编译模块与安装模块;

```
make modules
make modules_install
```

### 3.2 aodv-uu 的移植与编译

路由协议是无线自组织网络实验平台的一个重要组成部分,路由协议 aodv-uu 的移植与编译过程如下:

(1) 下载 aodv-uu 到 linux 系统并解压到目录/work/liod-aodv/aodv-uu-0.9.3/;

(2) 修改 aodv-uu 的编译配置文件 Makefile,主要是修改其内核源码目录等参数;

把/work/liod-aodv/aodv-uu-0.9.3/Makefile 与/work/liod-aodv/aodv-uu-0.9.3/lnx/Makefile 文件中的以下参数根据 liod 平台的内核源码目录进行修改:

```
KERNEL_DIR = /work/liod-aodv/linux-2.4.21-51Board_DVK
```

(3) 编译运行于 arm 体系结构处理器的用户空间守护进程 aodvd 与内核模块 kaodv.o;

```
make arm
```

(4) 安装 aodv-uu 的内核模块 kaodv.o 与用户空间守护进程 aodvd;

编译成功后会在/work/liod-aodv/aodv-uu-0.9.3/产生 aodv 的用户空间守护进程 aodvd 与内核模块文件 kaodv.o,需要把 kaodv.o 安装到内核模块的系统安装目录,同时把用户空间守护进程 aodvd 复制到平台的根文件系统 rootfs 的目录中。

```
mkdir /lib/modules/2.4.21-rmk1-pxa1-xs-
```

```
base270/aodv
```

```
cp /work/liod-aodv/aodv-uu-0.9.3/kaodv.o
/lib/modules/2.4.21-rmk1-pxa1-xsbase270/aodv
```

### 3.3 嵌入式 linux 文件系统 rootfs 的构建

(1) 使用 depmod.pl 生成出正确的模块依赖性文件 modules.dep;

当安装模块命令 make modules\_install 成功执行后,会把新编译的内核模块安装到/lib/modules/2.4.21-rmk1-pxa-xsbase270 目录中,但 linux 2.4 提供的模块依赖性创建工具 depmod 不能正常识别 ARM 体系结构处理器使用的内核模块文件的格式,所以该步骤不能正确产生模块依赖性列表文件 modules.dep 文件,若没有正确的 modules.dep 文件,则内核不能实现自动加载模块文件,而需要每次启动系统后手动把各个的内核模块的相应依赖模块都使用 insmod 添加到内核。

要生成正确的 modules.dep 文件,可以使用开源嵌入式根文件系统软件 busybox[2] 提供的 depmod.pl 文件,depmod.pl 文件可以从 busybox-0.60.3.tar.gz 文件的 scripts 目录中提取。

复制该 depmod.pl 文件到内核源码主目录中。运行 depmod.pl 脚本,并把生成的 modules.dep 文件覆盖模块目录/lib/modules/2.4.21-rmk1-pxa1-xsbase270 中的 modules.dep 文件:

```
./depmod.pl -F System.map -b /lib/modules/
2.4.21-rmk1-pxa1-xsbase270 > modules.dep
cp modules.dep /lib/modules/2.4.21-rmk1-
pxa1-xsbase270
```

(2) 把新生成的内核模块文件目录/lib/modules/2.4.21-rmk1-pxa1-xsbase270 替换原 liod 系统提供的 rootfs 中的模块目录;

(3) 把 aodv 用户空间守护进程 aodvd 程序添加到 rootfs 文件系统的 sbin 目录、使用 jffs2 的文件系统创建工具 mkrootfs.jffs2 建立 jffs2 文件系统;

(4) 修改 PCMCIA Card Services 的配置文件 config,该文件位于 rootfs 的/etc/pcmcia/目录,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 无线网卡驱动;

需要先把 CF 无线网卡接入到开发板,启动后使用 cardctl ident 命令找出无线网卡对应 ID 号,然后在根据

该 ID 号修改 config 文件。根据本系统使用的 CF 接口的无线网卡,把以下内容添加到 config 文件中:

```
device "orinoco_cs"
class "network" module "orinoco_cs"
card "Intersil PRISM2 11 Mbps Wireless Adapter"
manfid 0x0156, 0x0002
bind "orinoco_cs"
```

(5) 把新编译好的内核文件 zImage、新生成的文件系统 rootfs.img 复制到 tftp 服务器的主目录,通过 bootloader 下载并烧写到开发板。

### 4 无线自组织网络实验床测试

接入 CF 接口无线网卡,启动开发板,本系统使用的无线网卡使用 orinoco\_cs 的驱动程序,CF 接口的无线网卡被内核识别为 eth1 的设备号。使用 iwconfig 与 ifconfig 配置好各个节点无线网卡的相关参数,在各个节点启动 aodvd 路由协议,各节点通过 aodv 协议动态组成网络。

#### 4.1 线性链路拓扑测试

使用 4 个开发平台组成一个线性链路的网络,其拓扑结构如图 4 所示。每个节点间隔 50 米,均接入一个 CF 接口的 802.11b 无线网卡。

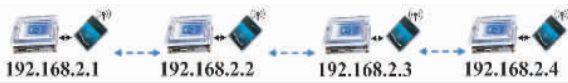


图 4 4 个节点组成线性拓扑

使用以下 shell 脚本来配置其无线网卡参数并启动 aodvd 路由守护进程,启动 aodvd 后,4 个节点组成一个自组织网络:

```
ifconfig eth1 192.168.2.N netmask 255.255.255.0 (各节点 N 分别为 1、2、3、4)
iwconfig eth1 mode ad-hoc
iwconfig eth1 essid adhoc-testbed
route add default dev eth1
aodvd -i eth1 -r 3 -l -d
```

分别在每个节点使用 ping 命令测试到其他节点的连通性,节点启用 aodv 的路由发现过程,各节点协同完成该路由查找过程。如在节点 192.168.2.1 中 ping 192.168.2.4 后在每个节点使用 route 命令查看各节点的路由表,图 5 为线性链路的最后一个节点的路由表。

路由表。

```
[root@S1Board ~]# route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
192.168.2.1 192.168.2.3 255.255.255.255 UGH 2 0 0 eth1
192.168.2.3 * 255.255.255.255 UH 1 0 0 eth1
192.168.2.2 192.168.2.3 255.255.255.255 UGH 2 0 0 eth1
192.168.2.0 * 255.255.255.0 U 0 0 0 eth1
127.0.0.0 * 255.0.0.0 U 0 0 0 lo
default * 0.0.0.0 U 0 0 0 eth1
```

图 5 线性测试拓扑中 192.168.2.4 节点的路由表

```
[root@node2 ~]# route
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
192.168.1.100 192.168.2.1 255.255.255.255 UGH 1 0 0 eth1
192.168.2.1 * 255.255.255.255 UH 1 0 0 eth1
192.168.2.3 * 255.255.255.255 UH 1 0 0 eth1
192.168.2.4 * 255.255.255.255 UH 1 0 0 eth1
192.168.2.0 * 255.255.255.0 U 0 0 0 eth1
127.0.0.0 * 255.0.0.0 U 0 0 0 lo
default * 0.0.0.0 U 0 0 0 eth1
```

图 6 星形测试拓扑中 192.168.2.2 节点的路由表

图 5 为线性拓扑中节点 192.168.2.4 的路由表,从 192.168.2.4 出发到达 192.168.2.1、192.168.2.2 的下一跳是 192.168.2.3,而从 192.168.2.4 出发到达 192.168.2.3? 却是直接通过 eth1 发往下一跳节点即可。

#### 4.2 自组织网络实验床与有线网络 LAN 互连

搭建如图 7 所示的星形拓扑网络,其中 ad hoc 网关节点通过 eth0 与有线网络 LAN 的 192.168.1.0 网段互连,配置 LAN 主机的网关地址为 192.168.1.1; 网关节点还通过 CF 无线网卡与其他的节点通过 aodv-uu 互连。

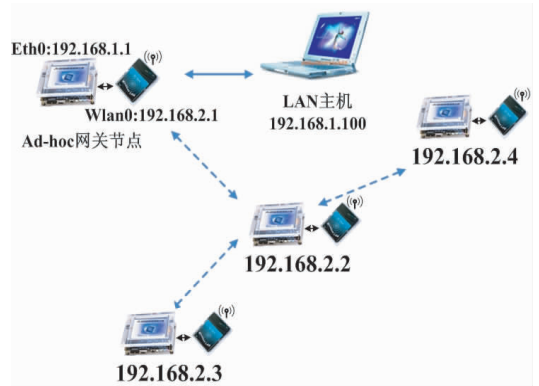


图 7 ad hoc 网络与 LAN 互连

网关节点配置有线网络接口的地址为 192.168.1.1,CF 接口无线网卡的地址为 192.168.2.1,在网关节点中启动 aodvd 网关模式。其启动的 shell 脚本为:

```
ifconfig eth0 192.168.1.1 netmask 255.255.255.0 up
```

```
ifconfig eth1 192.168.2.1 netmask 255.255.255.
```

```
0 up
```

```
iwconfig eth1 mode ad-hoc
```

```
iwconfig eth1 essid adhoc-testbed
```

```
aodvd -i eth1 -r 3 -l -w -d
```

其他节点:关闭 eth0 接口,启动无线网络接口、配置其无线网卡参数与 IP 地址,启动 aodvd,添加默认路由,其启动脚本同线性链路节点启动一致。

分别在每个节点使用 ping 命令测试其他节点的连通性,并使用 ping 命令访问有线网络中的主机 192.168.1.100,然后使用 route 命令查看当前节点的路由表。

图 6 为星形拓扑的中心节点 192.168.2.2 的路由表,到达其他三个节点 192.168.2.1、192.168.2.3、192.168.2.4 均是直接通过 eth1 发往下一跳节点即到。Ad hoc 中的各节点到达有线网络节点 192.168.1.100 则需要通过下一跳 192.168.2.1 才能到达。

## 5 结束语

本方案设计的无线自组织网络实验床能够通过多个 liod 节点自组织组网并正常运行。目前在该网络上能正常运行 ftp、telnet 等基本的网络服务。使用该自组织网络实验床可以方便进行网络安全算法、入侵检测方面的实验测试,同时也可进行一些应用层软件的测试,这也是今后研究的主要工作。

## 参考文献

- 1 Ad-hoc On-demand Distance Vector Routing - For real world and simulation. <http://core.it.uu.se/core/index.php/AODV-UU>.
- 2 BusyBox: The Swiss Army Knife of Embedded Linux. <http://www.busybox.net>.
- 3 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.