计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2018,27(2):112−116 [doi: 10.15888/j.cnki.csa.006190]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

IPv6 物联网接入网关的设计实践①

张美平, 丁文才, 许友泽

(福建师范大学 数学与计算机科学学院, 福州 350007)
通讯作者: 张美平, E-mail: mpjason@fjnu.edu.cn

摘　要: 在智能农业大棚物联网系统中, 传感器节点采集的数据需通过互联网上传到远程服务器, Contiki 节点间使用 6LoWPAN, 与传统的 IPv4 不能直连, 需配置一个物联网网关. 通过移植 OpenWRT 系统至 HG255D 无线路由器, 使得该路由器成为一个小巧、实用的嵌入式 Linux 系统, 能够运行开源或自行开发的程序. 本文将介绍如何在 HG255D 路由器上实现 IPv6 物联网接入网关设计的过程. 主要涉及 OpenWRT 固件的编译、路由器配置修改、网关程序的移植, 以及效果演示.

关键词: 物联网; 网关; OpenWRT; IPv6

引用格式: 张美平,丁文才,许友泽.IPv6 物联网接入网关的设计实践.计算机系统应用,2018,27(2):112–116. http://www.c-s-a.org.cn/1003-3254/6190.html

Design and Practice of IPv6 Access Gateway for Internet of Things

ZHANG Mei-Ping, DING Wen-Cai, XU You-Ze

(Colledge of Mathematics and Computer Science, Fujian Normal University, Fuzhou 350007, China)

Abstract: In the IOT of intelligent agricultural greenhouse system, the data collected by the sensor nodes need to be uploaded to the remote server via the Internet, and 6LoWPAN is used between the Contiki nodes. It cannot connect directly with the traditional IPv4, so it is necessary to configure an Internet of Things gateway. By transplanting the OpenWRT system to HG255D router, the router can be changed into a small, practical embedded Linux system, so that they can run open source or developed program. This article introduces how to implement HG255D routers on the IPv6 Internet access gateway, mainly involving the compilation of the OpenWRT firmware, router configuration changes, the transplantation of gateway program, as well as the effect of demonstration.

Key words: Internet of Things; gateway; OpenWRT; IPv6

引言

随着科学技术的发展, 信息技术的进步, 我们正逐渐的从互联网向物联网迈进. 物联网就是物物相连的互联网, 是新一代信息技术的重要组成部分, 也是“信息化”时代的重要阶段.

物联网技术在现代农业中的应用, 对提高农业的生产管理水平效果显著. 搭建一个智能农业大棚系统模拟系统, 监测环境数据, 探究温度、湿度、光强等环境因素对农作物的影响. 采集节点采用移植了 Contiki 操作系统的传感器, 该操作系统支持 6LoWPAN 协议, 该协议是 IPv6 在无线传感网的一种实现, 为使传感器节点接入网络, 与现有的 IPv4 融合, 需要设计移植一个支持 IPv6 的嵌入式设备作为物联网接入网关.

物联网网关, 作为一个新的名词, 在未来的物联网时代将会扮演非常重要的角色, 它将成为连接物联网与互联网的纽带. 作为网关设备, 物联网网关可以实现

① 基金项目: 福建省 2016 年省级大学生创新训练项目 (201610394044)
收稿时间: 2017-04-27; 修改时间: 2017-05-19; 采用时间: 2017-05-31

物联网与通信网络以及不同类型无线传感器与执行器网络之间的协议转换设备. 既可以实现广域互联, 也可以实现局域互联[1]. 对于使用 IPv6 的无线传感器与执行器网络的设备来说, 如果要跟互联网上的设备终端进行连接, 那就必须经过协议转换才能做到, 但由于自身性能的限制, 而不能自行进行协议转换, 若在传统的无线路由器上, 结合开源无线路由操作系统 OpenWRT, 将其作为网关设备, 就能实现与互联网的无缝连接.

本网关是在 OpenWRT 无线路由器上部署 MQTT\MQTT-SN 协议, 以实现无线传感器与执行器网络与传统通信网络的无缝连接. 消息队列遥测传输 (Message Queuing Telemetry Transport, MQTT) 是 IBM 开发基于消息发布与订阅机制的即时通讯协议, 有可能成为物联网的重要组成部分. 该协议支持大部分的嵌入式系统平台, 几乎可以把大部分的物联网硬件和外部网络连接起来, 通常被用来当做传感器和执行器 (比如通过 Twitter 让房屋联网) 的通信协议[2]. MQTT-SN 协议是 MQTT 针对无线传感器网络 (Wireless Sensor Networks, WSN) 的拓展. MQTT-SN 的运行需要依赖于已部署 MQTT 协议的网络. MQTT-SN 协议中有两个角色: 节点 (Node) 和网关 (GW). WSN 节点通过 MQTT-SN 协议与网关交互, 网关进行协议转换之后使节点接入 MQTT 服务器. 由于 MQTT-SN 协议运行需要依赖于 MQTT 协议, 下文使用 MQTT-SN 协议泛指 MQTT\MQTT-SN 的组合[3].

1 IPv6 物联网接入网关设计

1.1 IPv6 物联网接入网关总体框架

本文提出的 IPv6 物联网接入网关将 IPv6 无线传感网与现有的 IPv4 网络相融合, 进行协议转换. 底层的物联网使用部署了 MQTT-SN 协议的 contiki 节点, 这些节点通过 IEEE 802.15.14 无线协议与边界节点通信; 边界节点则使用 PL2302 USB 转串口芯片与运行 MQTT-SN 网关的无线路由器相连; 运行 OpenWRT 系统的路由器可与 contiki 节点组建一个网络, MQTT 服务器通过网线或 Wifi 接入网关所在的局域网内; 物联网应用程序可运行在手机、电脑、开发板或其它平台, 也是通过网线或 Wifi 接入该局域网, 并与 MQTT 服务器通信, 系统结构如图 1 所示.

1.2 物理网网关硬件软件设计方案

网关选择华为 HG255D 无线路由器作为硬件平台, 其处理器为 MIPS 架构的 Ralink RT3052, 拥有 4 个 Lan 口、1 个 Wan 口、一个 USB 接口. 运行 OpenWRT 系统, 可拓展路由器的功能, 配合 contiki 物联网节点, 共同完成本设计.

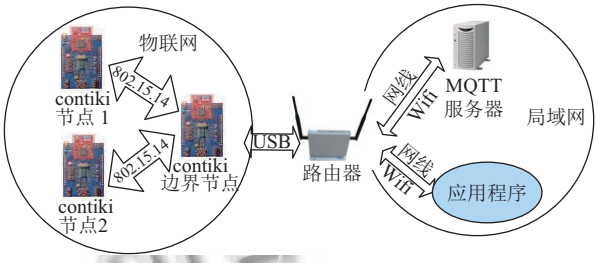

图 1 系统结构

Contiki 传感器与执行器节点的处理器为意法半导体公司的 STM32W108, 该处理器采用了 32 位 ARMCortex-M3 内核, 其特点是低功耗、处理能力强, 该微控制器提供了丰富的 GIPO 接口, 可以当作连接外部设备的控制接口, 如用于提供 LED、按键、蜂鸣器接口, 使节点具有丰富的人机交互与设备控制功能; STM32W108 提供丰富外设接口如 UART、ADC 等.

OpenWRT 是一个高度模块化、高度自动化的嵌入式 Linux 操作系统, 具有强大的网络功能和扩展能力. 同时, 还提供了 100 多个软件以供编译选择, 而且数量还在不断增加, 其中包含支持 IPv6 的相关软件. 在同类的无线路由器固件系统中, OpenWRT 在性能、稳定性、可扩展性方面的出色表现, 赢得众多开发者的支持. 是一个全功能的, 易于扩展的路由器操作系统, 成为在同类产品中最好的固件解决方案. 由于 OpenWRT 系统拥有强大的可扩展性, 可以通过交叉编译来运行 OpenWRT 官方未提供的程序, 以满足用户定制的需求. 此外, Linux 也为用户提供了众多的开源且免费的软件, 基于此, 本系统选择 OpenWRT 作为网关的操作系统.

2 IPv6 物联网接入网关工作原理

2.1 虚拟网卡 tun0 及 tunslip6

在物联网中, 运行 contiki 系统的无线传感器与执行器网络的节点通过 contiki 边界节点与外界通信, 基于 6LoWPAN 的物联网使用 IPv6 协议, 因此在路由器上需有一个 IPv6 的网络接口与之建立通信. 在路由运行 contiki 自带的 tunslip6 程序, 该程序执行后会建立一个名为“tun0”的虚拟网卡. Tunslip6 的作用是建立

RPL 边界节点与 tun0 两者之间相互通信的桥梁, 给边界路由器赋予 IPv6 网络地址, 例如 aaaa: : 1/64, 它的原理是在主机上为 tun0 虚拟网卡配置网络参数 (虚拟网卡设备位于/dev/net/tun0, 可以直接用程序读写该设备, tun0 建立在数据链路层, 所以读写内容为 IP 数据包), 边界路由器 (border-router) 与主机之间通信使用的是 slip 协议, slip 协议为一个串口 IP 协议, tunslip6 程序中包含读串口的代码, 可将读到的内容可以写到 tun0 设备上, tunslip6 程序中还包含读 tun0 的代码, 可以将 tun0 读到的内容通过 slip 协议封装后写到串口, 其原理可由图 2 表示.

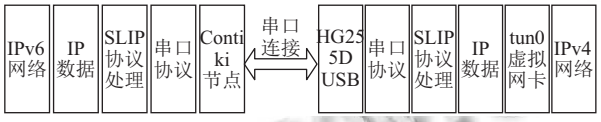

图 2 tunslip6 原理

2.2 MQTT 以及 MQTT-SN

MQTT 协议是为大量计算能力有限, 且工作在低带宽、不可靠的网络的远程传感器和控制设备通讯而设计的协议. 使用消息发布/订阅模式, 提供一对多的消息发布, 使用 TCP/IP 提供网络连接, 小型传输, 开销很小 (固定长度的头部是 2 字节), 协议交换最小化, 以降低网络流量, 使用 Last Will 和 Testament 特性通知有关各方客户端异常中断的机制以及“至多一次”、“至少一次”、“只有一次”三种消息发布服务质量.

在 MQTT 协议中有发布者、订阅者和代理 (broker) 三种角色, 消息由发布者通过 topic 发布, 由订阅者对感兴趣的 topic 进行订阅, 一个发布者可以对应多个订阅者, 一个订阅者也可以订阅多个 topic, 另外订阅者也可以是发布者, 发布者也可以是订阅者, 这样就实现了“M2M”(Machine to Machine) 的通信. Broker 类似于信差的角色, 主要作用就是接收信件并投递给订阅信件的人. 如果订阅者不在线, 也就是说没有 connected to the broker, 那么消息会保留, 等订阅者在线时推送. 图 3 为 MQTT 协议消息订阅发布模型.

图 3 MQTT 订阅发布模型

MQTT 是基于 TCP/IP 协议的, 在无线传感器网络中, 可以采用 MQTT-SN (MQTT for Sensor Networks) 协议进行补充, 它是为了资源受限硬件如传感器设计的, 能够通过 IEEE 802.15.4 发送 UDP 数据包, 再通过 MQTT-SN 网关与 MQTT broker 建立连接. MQTT-SN 的网关程序名为“mipsmqttsn”, 负责处理 tun0 上的数据, 然后由 tun0 转发处理这些数据. 在运行 mipsmqttsn 程序时, 需要 MQTT 服务器的 IPv4 地址作为参数. 应用程序则与 MQTT 服务器直接通信. 其工作原理可由图 5 表示, 图 4 为物联网网关设计系统结构示意图.

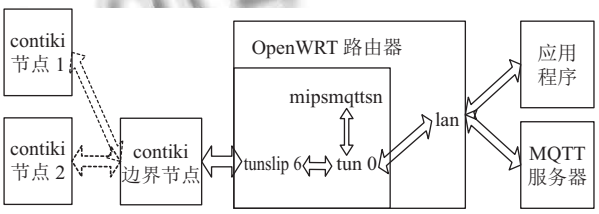

图 4 系统结构

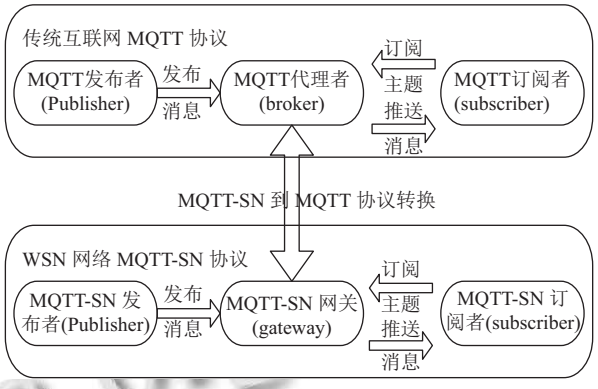

图 5 MQTT-SN 订阅发布交互模型

3 系统运行测试

为了测试该设计, 我们搭建一个简单的测试环境, 逻辑如图 6 所示.

其中, MQTT 服务器运行在 Windows 系统, 应用程序运行在 Linux 系统中; contiki 节点连接光敏传感器, 路由器为运行 OpenWRT 的物联网网关. Contiki 光照节点捕获光照数据后, 将所获取的光照数据以 IPv6 数据包形式通过无线射频发送给边界节点, 边界节点通过串口将数据包传输到物联网网关, 网关进行协议转换, 转换为 IPv4 数据包, 发布到 MQTT 服务器, 运行在 Linux 中的 QT 程序向 MQTT 服务器订阅该 topic, 服务器收到订阅请求后, 向订阅者推送订阅消息, 组网测试实物连接如图 7 所示.

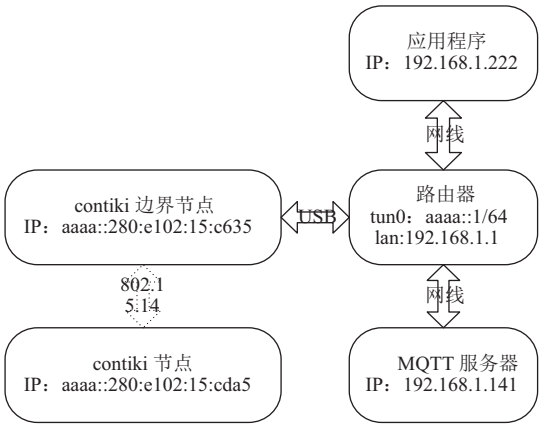

图 6 测试环境逻辑框图

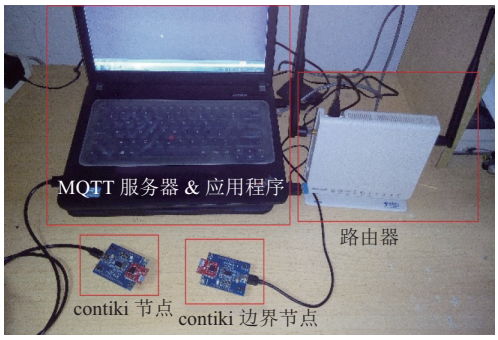

图 7 测试环境实物图

在该测试系统中, contiki 节点采集光敏传感器的数据, 并向服务器发布数据; contiki 边界节点则负责 contiki 节点与 tun0 网卡间的相互通信.

测试步骤:

1) 启动 MQTT 服务器 apache-activemq, 运行“apache-activemq-5.9.0\bin\win64”目录中的 activemq.bat 批处理文件, 即可启动服务器. 该服务器的 IP 地址即为 PC 的 IP 地址, 通过开始→运行→cmd→ipconfig 来查看 PC 的 IP 地址. 此时, IP 地址为 192.168.1.141.

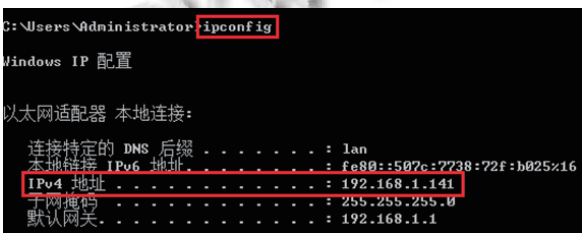

图 8 MQTT 服务器 IP 地址

2) 查看无线路由器是否挂载边界节点, 若成功挂载上 contiki 的 RPL 边界节点, 则会在/dev/目录下找到名为“ttyUSB0”的设备文件.

3) 运行 tunslip6 命令生成 tun0 网卡:

tunslip6 -s/dev/ttyUSB0 aaaa::1/64 &

运行 tunslip6, 该程序使用 tun/tap 网络协议使 USB 口可以作为虚拟网接口使用, 在这里是建立起 Contiki 边界节点与 tun0 之间的相互通信, 结果如图 9 所示.

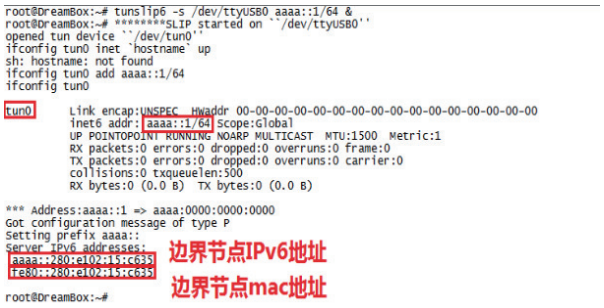

图 9 运行 tunslip6

启动成功后, 建立了 tun0 的接口, tun0 节点的 ip 为 aaaa::1/64, 程序与边界节点交互, 根据 tun0 的网段与边界节点自身的 mac 地址, 获得边界节点的 IPv6 地址为“aaaa::280:e102:15:c635”.

4) 成功启动 tunslip6 程序后, 即可运行 MQTT-SN 协议的网关程序. 在路由器的终端输入命令 mips-mqttsn-h “MQTT 服务器 IP 地址”, 结果如图 10 所示.

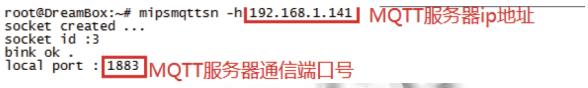

图 10 运行 mipsmqttsn

5) 接入 contiki 节点, 给 contiki 节点上电, 它将自动与边界节点组成 6LoWPAN 网络, 根据边界节点所处网段与节点自身 mac 地址, 获得一个与边界节点处于同一网段的 IPv6 地址“aaaa::280:e102:15:cda5”, 并开始向 MQTT 服务器发布光照强度的数据.

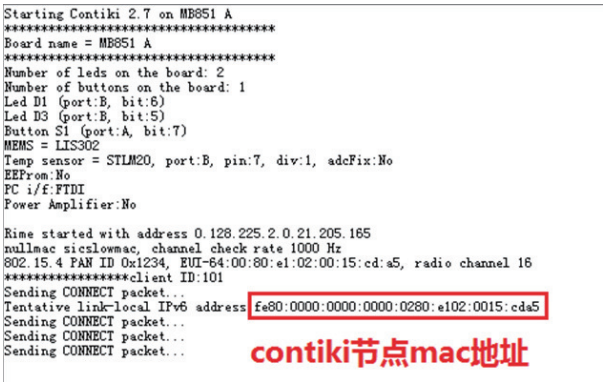

图 11 Contiki 节点串口信息

6) 在 Linux 虚拟机中运行一个 Qt 数据采集程序, 该程序向 MQTT 服务器订阅光照强度的数据, 如图 12 所示.

图 12 Qt 应用程序界面

结果显示, 该 Qt 程序能从 MQTT 服务器上获取相应的数据.

4 总结与教训

在选择物联网网关的嵌入式设备与操作系统的过程中, 我们曾考虑使用各种 ARM 嵌入式开发板, 我们分别尝试过运行 Arm-Linux 的 Tiny210、Mini2440 开发板, 最后选择了支持带 USB 接口并支持 OpenWRT 的无线路由器. 因为要使 ARM 开发板支持 IPv6, 就需要重新编译内核、新增相关模块如 tun/tap、IPv6 以及各种硬件驱动, 如果希望该网关提供无线热点功能, 还需要增加并驱动无线网卡, 配置并交叉编译 HostAP, 步骤繁琐复杂. 而提供 USB 接口的无线路由器可以方便连接 contiki IPv6 边界路由节点, 并且 OpenWRT 提供官方固件, 各项内核模块以及相关硬件驱动 (如 USB 串口驱动) 均可通过“opkg install xxx”命令在线安装 (opkg 是一个轻量快速的套件管理系统, 目前已成为 Opensource 界嵌入式系统标准. 常用于路由、交换机等嵌入式设备中, 用来管理软件包的安装升级与下载). 更重要的是无线路由器本身就板载无线 Wifi 模块, 并完美驱动可作为无线热点 AP 使用, 此外 OpenWRT 提供的 WEB 配置界面方便用户配置各种网络参数.

在选择 MQTT 服务器方面, 当前流行的开源 MQTT 代理服务器 (MQTT Broker) 的实现有: Mosquitto、HiveMQ、Apache ActiveMQ、RabbitMQ、Apollo、RSMB 等. MQTT 客户端也有不同操作系统和编程语言的实现, 流行的客户端库有: Eclipse Paho(支持 C、C++、Java、JavaScript、Python、Go、C#), M2MQTT (C#), FusesourceMQTTClient(Java), MQTT.js (javascript), libmosquitto(c/c++) 等等, 我们使用了 Apache ActiveMQ 作为 MQTT 服务器; 由于 contiki 的应用程序使用 C 语言开发, 在 MQTT 客户端的开发也采用了基于 Mosquitto 库来开发客户端.

本设计是为解决物联网中无线传感器与执行器网络与传统互联网的接入网关问题, 通过部署 MQTT-SN 协议, 使无线传感器与执行器网络的各节点具备订阅/发布的功能, 通过移植 MQTT-SN 协议的网关程序到 OpenWRT 系统上, 并与 MQTT 服务器相连, 再加入自行开发的 C/C++ Linux 应用程序, 即可构成一个简单、实用的 IPv6 物联网应用系统.

无线传感与执行器网络的各节点采用 IPv6 协议, 打破了传统的 IPv4 物联网格局, 增大了地址量、增强了安全性. 外部终端通过有线无线均可接入, 使底层物联网的控制变得更方便, 具有一定的实际应用意义与推广价值.

但本次的设计也存在一些不足, 如未将 MQTT 服务器架设到 Internet 上, 以实现真正意义上的远程控制物联网, 如果将 MQTT 服务器架设到 Internet 上, 将能实现远程推送服务, 增强物联网应用系统的实用功能.

参考文献

1 互联网. 物联网网关总结. http://wenku.baidu.com/link?url=R847CuPuArEW4ErfZwkV_yYoLBfVyjjhdgvK0sZGvignP6wYik-GiG_DQ6TNES011Hx8wNn-N-CyLAbBfS1KRArSntR1nr549XYjoBJQwwe. [2015-01-29].

2 互联网. MQTT. http://baike.baidu.com/link?url=DEeX5HzgZIO1xUTy2w5VjX2e__WcmkRmQxr7iXg_prTVM-1dU0JRtdPu8HbsHAXMtW2sVaOiDdRr9BzIWf9Uw_. [2016-08-01].

3 陈旖, 张美平, 许力. WSN 应用层协议 MQTT-SN 与 CoAP 的剖析与改进. 计算机系统应用, 2015, 24(2): 229–234.