

AdHoc 网络中语音传输的分析及性能测试*

杜智敏^{1,2}, 谢维波^{1,2}, 杨大力^{1,2}

(1. 华侨大学 计算机科学与技术学院, 福建 厦门 361021;

2. 华侨大学 厦门软件园嵌入式技术开放实验室, 福建 厦门 361008)

摘要: 基于 AdHoc 网络的语音通信研发需要对 AdHoc 网络路由机制、语音通信特性等有深入理解。阐述了 AdHoc 网络中语音通信模型, 网络机制、性能评估分析。最后对 AdHoc 网络及其上的语音通信性能进行评估分析。

关键词: 语音通信; AdHoc; 网络机制

中图分类号: TP311

文献标识码: A

文章编号: 1674-7720(2012)10-0049-03

Analyzing and performance evaluation of voice transmission over AdHoc network

Du Zhimin^{1,2}, Xie Weibo^{1,2}, Yang Dali^{1,2}

(1. College of Computer Science & Technology, Huaqiao University, Xiamen 361021, China;

2. Open Laboratory of Embedded Technology, Xiamen Software Park, Huaqiao University, Xiamen 361008, China)

Abstract: To develop voice communication application over AdHoc network, it is required that the developers deeply understand the routing mechanism and the characteristic of voice transmission in ad hoc network. The thesis illustrates the basic voice communication model, additionally carries out exhaustive analysis and discussion upon adhoc network mechanism and its performance, then give a performance evaluation of voice transmission over AdHoc network.

Key words: voice transmission; AdHoc; network mechanism

随着无线通信技术和嵌入式技术的飞速发展, AdHoc 网络技术及其上的多媒体应用成为一个研究热点。由于实验条件等限制, 目前针对 AdHoc 网络的研究大都基于 NS (Network Simulator)、GloMoSim 等仿真工具, 研究成果往往理论有余, 而实践稍显不足。为了深入研究 AdHoc 网络中的语音通信技术, 构建实际 AdHoc 网络, 对该实际网络环境及其上语音通信性能的评估分析研究就显得十分必要。

1 语音传输模型

语音在 AdHoc 网络中传输的基本模型如图 1 所示。音频设备采集到的模拟语音信号必须经过采样量化等处理手段转换成数字信号, 然后经过压缩并打包成某种网络数据包格式, 之后在 AdHoc 网络中进行传输。到达

目的端系统后首先经过网络协议栈解包, 得到压缩过的数字语音数据, 之后经过解压缩等处理过程, 得到原始语音数据。

决定语音质量的主要因素是编解码器的选择, 其次是网络传输性能的影响。网络延时和丢包对语音通信质量造成的影响在 AdHoc 网络中一直是个技术瓶颈。

2 AdHoc 网络模型

IEEE 802.11 作为无线通信的官方标准仅仅考虑到了单跳的网络, 因此很有必要为多跳无线网络开发相应的路由协议。这些路由协议决定如何通过其他节点对目的节点不在传输域内的信息进行转发。

大量针对不同路由协议的模拟已经通过不同的模拟器进行了模拟。但是模拟器建立的是一种理想模型,



* 基金项目: 厦门市重点产学研项目 (厦经技[2009]233-03), 福建省自然科学基金项目 (2010J01334)

网络与通信 Network and Communication

并不能将实际情况中的所有因素考虑在内,所以其结果同样是一种理想结果,并不能作为商业应用的参考。

这里使用 AODV(AdHoc on-Demand Distance Vector)^[1-2] 协议对 AdHoc 网络的实时通信性能进行测试。AODV 借鉴了 DSR 协议的路由发现和维护概念,以及 DSDV 协议的序列号和发送周期性 Hello 包的机制。其主要路由机制如下:

(1)路由发现机制:当一个节点要与另外一个节点通信时,它首先查看自己的路由表,看到达该目的节点的路由是否存在,若不存在,则初始化一个路由发现过程(AODV 创建并广播一个 RREQ 控制包,并设置一个定时器来等待回应数据包 RREP)。所有收到 RREQ 控制包的节点都会比较该包的 ID,看是否以前曾经收到过。若非第一次收到,则直接丢弃该控制包。

(2)路由维护机制:当一个路由建立好后,就需要源节点来对其进行维护。节点的移动只会影响到途经该节点的路由,并不对网络造成全局影响。源节点在通信过程中的移动会造成它与当前使用路由的下一跳之间的连接断开,此时,它将会重新广播一个 RREQ 包来发现路由。如果中间节点与其下一跳之间的连接断开,它会发送一个 RERR 包给其(当前路由)前驱节点,并将到达该目的节点的路由标记为无效(被标记为无效的路由将于一段固定时间之后被丢弃)。其前驱节点收到 RERR 包后,同样将该条路由标记为无效,并将该 RERR 发送给自己的前驱节点。

(3)Hello 包机制:在规定时间内,如果没有数据包传输,节点就会广播发送 Hello 消息给其邻居节点来对本地连接进行更新。Hello 消息的内容包括节点的 IP 地址及当前序列号等。由于该信息不需要被转发,它们的 TTL 值都为 1。

3 AODV 协议对网络性能的影响

(1)端到端延时:由于 AODV 为被动路由协议,所以在第一次数据传输时,如果没有有效路由,则会启动发现路由机制,于是引入了发现路由延时。发现路由延时的大小主要取决于端到端需要经过的跳数,当然不同的 AODV 实现版本之间的发现路由延时也会有所差别。

(2)路由变化延时:该延时是指从当前工作路由失效到重新发现并启用新的有效路由之间的时间。该延时取决于 $ALLOWED_HELLO_LOSS \times HELLO_INTERVAL$ 。可以通过调整这两个参数来降低该延时。 $ALLOWED_HELLO_LOSS$ 默认为 2,如果调整为 1,由于无线信道的不稳定,时不时信道状况不好会导致 Hello 包的丢失,由于 $ALLOWED_HELLO_LOSS$ 调整为 1,于是造成了路由协议认为链路已损坏,而实际链路并未断开的状况的出现。

分析图 2 所示的情况。假设 $ALLOWED_HELLO_LOSS$ 的默认值为 2,则 AODV 路由协议在 $2 \times HELLO_INTERVAL$ 未收到 Hello 包时就会认为该链路已经断开。此时有两种极端情况:



图 2 链路失效检测时间

①节点 1 在收到第 2 个 Hello 包之后,实际链路断开,而此时 AODV 路由协议由于收到了 Hello 包,认为链路依然有效。

②在节点 1 收到第 2 个 Hello 包之后一个 $HELLO_INTERVAL$ 时,实际链路断开,而 AODV 协议并不认为链路断开,而是会发送第二个 Hello 包来确认链路状况。由于链路失效时间不可预知,可认为其是介于 $HELLO_INTERVAL$ 和 $2 \times HELLO_INTERVAL$ 之间的一个随机数。由于相对路由变化时间来说路由发现时间可以忽略,可对其定义如下:

$$RCLatency = T_{detect} + RDLatency \approx T_{detect}$$

4 性能评估测试

4.1 测试环境构建

本文使用 4 台安装 Linux 系统的笔记本作为实验设备。路由协议使用开源的 `aodv-uu-0.9.6.tar.gz`^[3],应用程序为自己编写的双工语音通信程序(使用 `udp` 进行语音通信,由语音录制、语音压缩、语音解压缩、语音播放 4 个线程构成)。测试环境构建流程为:

(1)配置无线网卡(以 laptop1 为例)

```
#ifconfig wlan0 192.168.2.5 up
#iwconfig wlan0 essid test mode ad-hoc channel 1 rate 11M
```

设置完网卡后,可以使用 `ifup wlan0` 来查看网卡是否设置成功。

(2)安装路由协议

```
#tar -xvf aodv-uu-0.9.6.tar.gz
#cd aodv-uu-0.9.6
#make & make install
```

(3)构建网络拓扑

将 4 台笔记本直线摆放在实验桌面上,IP 地址分配依次为 192.168.2.5~192.168.2.8(4 节点结构,3 节点网络拓扑与此类似)。

由于笔记本自带无线网卡传输范围比较大,laptop1 和 laptop4 可以直接通信,而不需经过 laptop2 和 laptop3,而室内空间也有限,所以为实现 laptop1→laptop2→laptop3→laptop4 多跳效果,这里需要使用 Linux 系统的防火墙工具 `iptables`^[4]。其命令格式为:

```
Iptables -A INPUT -p ALL -m mac --mac laptop_mac -j DROP
```

其中 `laptop_mac` 为要屏蔽的 laptop 的 mac 地址。

为实现 laptop1 经过 laptop2,再经过 laptop3,最后到

表 1 3 个节点丢包率和延时

时隙/s	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.5	2.0	5
丢包率/%	0	1	1	2	2	2	2	3	4	4	4	5	58
时迟/ms	3.022	3.102	3.102	3.159	3.126	3.252	2.972	3.085	3.152	3.151	3.034	2.832	2.761

表 2 4 个节点丢包率和延时

时隙/s	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.5	2.0	5
丢包率/%	1	3	6	8	6	7	9	9	12	11	17	27	67
时迟/ms	5.097	5.140	5.441	5.513	5.349	5.232	5.304	5.508	5.631	5.452	5.431	5.738	5.419

达 laptop4 的多跳效果, 需要让 laptop1 屏蔽掉其到 laptop3 和 laptop4 的直接通路。在其上执行命令(其中以 # 开头的表示在超级用户权限执行的命令):

```
# Iptables -A INPUT -p ALL -m mac --mac
laptop3_mac -j DROP
# Iptables -A INPUT -p ALL -m mac --mac
laptop4_mac -j DROP
```

以同样的方式让 laptop2 屏蔽掉其到 laptop4 的直接通路, 让 laptop3 屏蔽掉其到 laptop1 的直接通路, 让 laptop4 屏蔽掉其到 laptop1 和 laptop2 的直接通路。

4.2 性能测试方案设计及测试结果分析

(1) 路由发现时间

按照上面步骤设置好无线网卡后, 启动 AODV 路由协议:

```
#!/aodvd -r 1 -l
```

使用 ping 命令从 laptop1 发送数据包到 laptop3 (和 laptop4), 此时 AODV 会启动发现路由过程, 并建立路由表。路由发现时间可由 AODV 路由协议的日志中得出 (rreq_create 启动到 aodv_socket_process_packet 对 RREP 包进行处理)。

测试可得, 3 个节点路由发现时间为 6 ms, 4 个节点路由发现时间为 10 ms。相对数据包发送延时来说, 基本可以忽略。

(2) AdHoc 网络丢包率和延时

使用 ping 命令向目的节点发送 1 000 个默认大小(56 B)的数据包。命令为: ping -c 1000 -i interval 192.168.2.8。其中 interval 为发送两次 icmp 数据包之间的时间间隔。

网络丢包率和延时测试结果如表 1、表 2 所示。

从测试结果可知, 当发包时隙为 5 s 时, 丢包率暴增到一半以上。这是 AODV 路由协议作用的结果, 正常情况下应该是发包频率越慢, 丢包率越低。由于 AODV 路由协议维护有效路由有个定时器: ACTIVE_ROUTE_TIMEOUT (默认为 3 000 ms) 和 DELETE_PERIOD。DELETE_PERIOD 计算公式^[5]为:

$$\text{DELETE_PERIOD} = 5 \times \max(\text{ACTIVE_ROUTE_TIMEOUT}, \text{ALLOWED_HELLO_LOSS} \times \text{HELLO_INTERVAL})$$

而 ALLOWED_HELLO_LOSS 为 2, HELLO_INTERVAL 为 1 000 ms, 于是 DELETE_PERIOD 默认为 3 000 ms。即

AODV 协议每隔 3 s 会删除在这段时间内都未使用的路由, 而发送数据包时隙为 5 s, 由于上次使用的路由已被删除, 于是每次发送都需要重新发现路由, 最终导致了严重的网络丢包率。

(3) 语音通信丢包率测试

在 laptop1、laptop3 (3 节点网络) 和 laptop4 (4 节点网络) 运行语音通信程序, 并在程序中增加数据包计数器来统计丢包率。

对比表 3 所示丢包率测试结果和 icmp 包的丢包率测试结果, 可以看出 3 个节点的网络中音频数据传输类似时隙为 0.1~0.2 之间的 icmp 包的传输, 但是 4 个节点时相当于时隙为 0.6 的 icmp 包的传输。该结果间接表示 4 个节点之间的语音传输延时大大增加, 这也是无线 AdHoc 网络的一大技术瓶颈。

表 3 语音传输丢包率

节点数量	测试时间/s	发包总数	收包总数	丢包总数	丢包率/%
3 个节点	≈ 180	21 592	21 287	305	1.41
4 个节点	≈ 164	20 416	18 868	1 548	7.58

本文深入阐述了 AdHoc 网络中的语音传输模型和 AdHoc 基本模型, 详细分析了 AODV 路由协议对 AdHoc 网络性能的影响以及语音评价标准。最后实际搭建了 AdHoc 网络, 并对其通信性能及其上的语音通信性能进行评估测试。本文旨在帮助广大研究人员对 AdHoc 网络性能有更深入的理解。

参考文献

- [1] AODV[DB/OL]. <http://moment.cs.ucsb.edu/AODV/>.
- [2] PERKINS C, ROYER E. AdHoc on-Demand distance vector routing-RFC3561[C]. 2003: 1-38.
- [3] AODV-UU[S/OL]. <http://sourceforge.net/projects/aodvu/>.
- [4] Netfilter/Iptables[S/OL]. <http://www.netfilter.org>.
- [5] Lu Yi, Zhong Yuhui, BHARGAVA B. Packet loss in mobile AdHoc networks[R]. 2003.

(收稿日期: 2011-12-25)

作者简介:

杜智敏, 男, 1987年生, 硕士研究生, 主要研究方向: 嵌入式技术。

谢维波, 男, 1964年生, 教授, 博士, 硕士生导师, 主要研究方向: 嵌入式技术、数字信号处理。