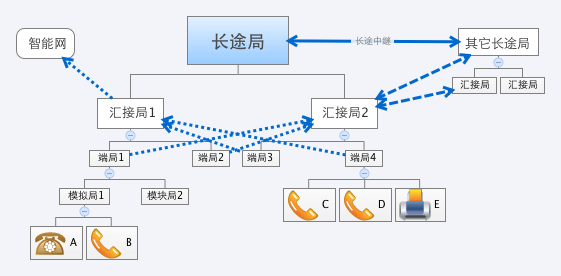
Freeswitch中文用户手册

第一章 PSTN 与 VoIP  
说起VoIP，也许大家对网络电话更熟悉一些。其英文原意是Voice Over IP，即承载于IP网上的语音通信。大家熟悉家庭用来上网的ADSL吧，也许有些人还记得前些年用过的吱吱叫的老“猫”。技术日新月异，前面的技术都是用电话线上网，现在，VoIP技术使我们可以在网上打电话，生活就是这样。  
所谓温故而知新，在了解任何东西以前，我们都最好了解一下其历史，以做到心中有数。在了解VoIP之前，我们需要先看一下PSTN，那在PSTN之前呢？  
PSTN起源  
PSTN(Public Switched Telephone Network)的全称是公共交换电话网，就是我们现在打电话所使用的电话网络。  
第一次语音传输是亚历山大·贝尔(Alexander Granham Bell)在1876年用振铃电路实现的。在那之前，普遍认为烽火台是最早的远程通信方式。其实峰火台不仅具备通信的完整要素(通信双方，通信线路及中继器)，而且还是无线通信。当时的没有电话号码，相互通话的用户之间必须有物理线路连接；并且，在同一时间只有一个用户可以讲话(半双工)。发话方通过话音的振动激励电炭精麦克风而转换成电信号，电信号传到远端后通过振动对方的扬声器发声，从而传到对方的耳朵里。  
由于每对通话的个体之间都需要单独的物理线路，如果整个电话网上有10个人，而你想要与另个9个人通话，你家就需要铺设9对电话线。同时整个电话网上就需要 10 x (10-1) / 2 = 45 对电话线。  
当电话用户数量增加的时候，为每对通话的家庭之间铺设电话线是不可能的。因此一种称为交换机（Switch）的设备诞生了。它位于整个电话网的中间用于连接每个用户，用户想打电话时先拿起电话连接到管理交换机的接线员，由接线员负责接通到对方的线路。这便是最早的电话交换网。  
由于技术的进步，电子交换机替代了人工交换机，便出现了现代意义的PSTN。随着通信网络的进一步扩大，便出现了许许多多的交换机。交换机间通过中继线（Trunk）相连。有时一个用户与另一个用户通话需要穿越多台交换机。  
后来出现了移动电话（当移动电话小到可以拿在手里的时候就开始叫“手机”），专门用于对移动电话进行交换的通信网络称移动网，而原来的程控交换网则叫固定电话网，简称固网。简单来说，移动网就是在普通固网的基础上增加了许多基站（Base Station，可以简单理解为天线），并增加了归属位置寄存器（HLR，Home Location Register）和拜访位置寄存器（VLR，Visitor Location Register），以用户记录用户的位置（在哪个天线的覆盖范围内）、支持异地漫游等。移动交换中心称之为MSC（Mobile Switch Center）。  
模拟与数字信号  
现实中的一切都是模拟的。模拟量（Analog）是连续的变化的，如温度、声音等。早期的电话网是基于模拟交换的。模拟信号对于人类交流来讲非常理想，但它很容易引入噪声。如果通话双方距离很远的话，由于信号的衰减，需要对信号进行放大。问题是信号中经常混入线路的噪音，放大信号的同时也放大的噪音，导致信噪比（信号量与噪声的比例）下降，严重时会难以分辨。  
数字（Digital）信号是不连续的（离散的）。它是按一定的时间间隔（单位时间内抽样的次数称为频率）对模拟信号进行抽样得出的一些离散值。根据抽样定理，当抽样频率是最高模拟信号频率的两倍时，就能够完全还原原来的模拟信号。  
PCM  
PCM（Pulse Code Modulattion）的全称是脉冲编码调制。它是一种通用的将模拟信号转换成以0和1表示的数字信号的方法。  
一般来说，人的声音频率范围在 300Hz ~ 3400Hz 之间， 通过滤波器对超过 4000Hz 的频率过滤出去，便得到 4000Hz 内的模拟信号。然后根据抽样定理，使用 8000Hz 进行抽样，便得到离散的数字信号。  
通过使用压缩算法（实际为压扩法，因为有的部分压缩有的是扩张的。目的是给小信号更多的比特位数以提高语音质量），可以将每一个抽样值压缩到8个比特。这样就得到 8 x 8000 = 64000bit （通常称为64kbit/s。注意，通常来说，对于二进制数，1kbit=1024bit，但此处的k=1000）的信号。通常我们就简称为64k。  
PCM通常有两种压缩方式：A律和μ律。其中北美使用μ律，我国和欧洲使用A律。这两种压缩方法很相似，都采用8bit的编码获得12bit到13bit的语音质量。但在低信噪比的情况下，μ律比A律略好。  
我国电话网结构  
  
图中主体部分为一地市级电话网的结构。通常，话机（如c）通过一对电话线连接到距离最近的交换机上，该交换机称为端局交换机（一般以区或县为单位）。端局交换机通过局间中继线连接到汇接局。为了保证安全，汇接局通常会成对出现，平常实行负荷分担，一台汇接局出现故障时与之配对的汇接局承担所有话务。长途电话需要通过长途局与其它长途局相连。但根据话务量要求，汇接局也可以直接与其它长途局开通高速直达中继。为节省用户线，在一些人口比较集中的地方（如学校、小区），端局下会再设模块局或接入网，用户则就近接入的模块局上。  
智能网一般用于实现电话卡、预付费或400/800类业务，而近几年新布署的NGN（Next Generation Network，下一代网络，一般指软交换。）则支持更灵活、更复杂的业务。  
时分复用与局间中继  
时分复用  
通过将多个信道以时分复用的方式合并到一条电路上，可以减少局间中继线的数量。通过将32个64k的信道利用时分复用合并到一条2M（ 64k x 32 = 2.048M ，通俗来说就直接叫一个2M）电路上，称为一个E1（在北美和日本，是24个64k复用，称为T1，速率是1.544M）。在E1中，每一个信道称作一个时隙。其中，除0时隙固定传同步时钟，其它31个时隙最多可以同时支持31路电话（如果使用隨路信令，则使用第16时隙传送，这时最多支持30路电话）。  
局间中继  
这些连接交换机(局)的2M电路就称为局间中继。随着话务量的增加，交换机之间的电路越开越多，目前通常的做法是将63个2M合并到一个155M（ 2 x 63 + P = 155，其中P是电路复用的开销）的光路（光纤）上。  
信令  
用户设备（如话机）与端局交换机之间，以及交换机与交换机之间需要进行通信。这些通信所包含的信息包括（但不限于）用户、中继线状态，主、被叫号码，中继路由的选择等。我们把这些消息称为信令（Signaling）。  
用户线信令  
用户线信令是从用户话机到端局交换机之间传送的信令。对于普通的话机，线路上传送的是模拟信号，信令只能在电话线路上传送，这种信令称为带内信令。话机通过电压变化来传递摘、挂机信号；通过DTMF（Dual Tone Multi Frequency，双音多频。话机上每个数字或字母都可以发送一个低频和一个高频信号相结合的正弦波，交换机经过解码即可知道对应的话机按键）传送要拨叫的电话号码。另外，也可以通过移频键控（FSK,Frequency Shift-keying）技术支持来电显示（Caller ID或CLIP，Caller Line Identification Presentation，主叫线路识别提示）。  
与普通电话不同，ISDN（Integrated Service Digital Network，综合业务数字网）在用户线上传送的是数字信号。它的基本速率接口使用144k的2B+D信道--两个64k的B信道及一个16k的D信道。由于其信令在话路（B信道）以外的D信道传送，这种信令称为带外信令。  
实际上，2B+D的ISDN并没有发挥出它应有的作用，在国内已很少有人使用。  
局间信令  
局间信令主要在局间中继上传送。一般一条信令链路通常只占用一个64k的时隙。一条信令消息通常只有几十或上百个字节，一条64k的电路足矣容纳成千上万路电话所需要的信令。但随着技术的进步，话务量的上涨以及更多增值业务的出现，完成一次通话需要更多的信令消息，因此出现了2M速率的信令链路，即整个E1链路上全部传送信令。  
局间信令也分为带内信令和带外信令。带内信令又称为随路信令，它是在跟话路同一个2M上传送的，通常使用第16时隙。带外信令则是在独立的专门用于传送信令链路的2M中继上传送的，与带内信令相比，它更加灵活。我国的电话网络中有专门的信令网并使用7号信令（SS7, Signaling System No. 7）。  
七号信令  
SS7是目前我国使用的主要的信令方式  
用户A          a交换机        b交换机         用户B  
   |             |             |             |  
   |   摘机                
   |------------>|             |             |  
   |   拨号音                
   |<------------|             |             |  
   |   拨号            IAM          振铃      
   |------------>|------------>|------------>|  
   |   回铃音           ACM        
   |<------------|<------------|             |  
   |   通话            ANM           接听  
   |<------------|<------------|<------------|  
   |   ...       |             |             |  
   |   ...       |             |             |  
   |   挂机            REL           送催挂音     
   |------------>|------------>|------------>|  
   |                  RLC  
   |             |<------------|<------------|   
   |             |             |             |  
我们来看一次简单的固定电话的通话流程。如图。用户A摘机，与其相连的a交换机根据电压变化检测到A摘机后，即送拨号音，同时启动收号程序。A开始拨号，待a交换机号码收齐后，即查找路由，发送IAM（初始地址消息）给b交换机。b向话机B振铃，同时向a发ACM（地址全消息），a向A送回铃音。这时如果B接听电话，则b向a发送ANM（应答计费消息），A与B开始通话，同时a对A计费。通话完毕，任何一方挂机，则本端交换机（如a）向对端b发送 REL（释放消息），b向a回RLC（确认，释放完成），并向B送催挂音（啫啫啫...）。  
上面在交换机a与b之间传递的为七号信令中的TUP（Telephone User Part，电话用户部分）部分。目前，由于ISUP（ISDN User Part，ISDN用户部分）能与ISDN互联并提供比TUP更多的能力和服务，已基本取代TUP而成为我国七号信令网上主要的信令方式。  
电路交换与分组交换  
VoIP  
维基百科上是这样说的:  
IP电话(简称VoIP，源自英语Voice over Internet Protocol；又名宽带电话或网络电话)是一种透过互联网或其他使用IP技术的网络，来实现新型的电话通讯。过去IP电话主要应用在大型公司的内联网内，技术人员可以复用同一个网络提供数据及语音服务，除了简化管理，更可提高生产力。随着互联网日渐普及，以及跨境通讯数量大幅飙升，IP电话亦被应用在长途电话业务上。由于世界各主要大城市的通信公司竞争日剧，以及各国电信相关法令松绑，IP电话也开始应用于固网通信，其低通话成本、低建设成本、易扩充性及日渐优良化的通话质量等主要特点，被目前国际电信企业看成是传统电信业务的有力竞争者。详细内容参见维基百科上的 IP 电话。  
目前，VoIP呼叫控制协议主要有SIP、H323，以及MGCP与H.248/MEGACO等。H323是由ITU-T（国际电信联盟）定义的多媒体信息如何在分组交换网络上承载的建议书。它是一个相当复杂的协议，使用起来很不灵活。而SIP则是IETF（互联网工程任务组）开发的（RFC3261），它是一种类似HTTP的基于文本的协议，很容易实现和扩展，被普遍认为是VoIP信令的未来。

第二章 FreeSWITCH 初步  
什么是 FreeSWITCH ？  
FreeSWITCH 是一个开源的电话交换平台，它具有很强的可伸缩性--从一个简单的软电话客户端到运营商级的软交换设备几乎无所不能。能原生地运行于Windows、 Max OS X、Linux、BSD 及 solaris 等诸多32/64位平台。可以用作一个简单的交换引擎、一个PBX，一个媒体网关或媒体支持IVR的服务器等。它支持SIP、H323、Skype、Google Talk等协议，并能很容易地与各种开源的PBX系统如sipXecs、Call Weaver、Bayonne、YATE及Asterisk等通信。 FreeSWITCH 遵循RFC并支持很多高级的SIP特性，如 presence、BLF、SLA以及TCP、TLS和sRTP等。它也可以用作一个SBC进行透明的SIP代理（proxy）以支持其它媒体如T.38 等。FreeSWITCH 支持宽带及窄带语音编码，电话会议桥可同时支持8、12、16、24、32及48kHZ的语音. 而在传统的电话网络中，要做到三方通话或多方通话需要通过专门的芯片来处理，其它像预付费，彩铃等业务在PSTN网络中都需要依靠智能网(IN)才能实现，而且配置起来相当不灵活。  
快速体验  
FreeSWITCH 的功能确实非常丰富和强大，在进一步学习之前我们先来做一个完整的体验。FreeSWITCH 默认的配置是一个SOHO PBX(家用电话小交换机)，那么我们本章的目标就是从0安装，实现分机互拨电话，测试各种功能，并通过添加一个SIP-PSTN网关拨打PSTN电话。这样，即使你没有任何使用经验，你也应该能顺利走完本章，从而建立一个直接的认识。在体验过程中，你会遇到一点稍微复杂的配置，如果不能完全理解，也不用担心，我们在后面会详细的介绍。当然，如果你是一个很有经验的 FreeSWITCH 用户，那么大可跳过本章。  
安装FreeSWITCH基本系统  
在本文写作时，最新的版本1.0.5pre10，但说不定当你读到时1.0.5正式版已经发布了。FreeSWITCH支持32位及64位的 Linux、 MacOSX、BSD、Solaris、Windows等众多平台。某些平台上有编译好的安装包，但本人强烈建议从源代码进行安装，因为 FreeSWITCH 更新非常快，而已编译好的版本通常都比较旧。你可以下载源码包，也可以直接从SVN仓库中取得最新的代码。与其它项目不同的是，其SVN主干 (trunk)代码通常比稳定的发布版更稳定。而且，当你需要技术支持时，开发人员也通常建议你先升级到SVN中最新的代码，再看是不是仍有问题。  
Windows用户可以直接下载安装文件 <http://files.freeswitch.org/windows_installer/freeswitch-1.0.4.exe> （再提醒一下，版本比较旧代，如果从源代码安装的话，需要Visual Studio 2008）。安装完成执行 c:\freeswitch\freeswitch.exe 便可启动，其配置文件都在c:\freeswitch\conf\。  
以下假定你使用 Linux 平台，并假定你有 Linux 的基本知识。如何从头安装 Linux 超出了本书的范围，而且，你也可以很容易的从网上找到这些资料。一般来说，任何发行套件都是可以的，但是，有些发行套件的内核、文件系统、编译环境，LibC 版本会有一些问题。所以，如果你在遇到问题后想获得社区支持，最好选择一种大家都熟悉的发行套件。FreeSWITCH 开发者使用的平台是 CentOS 5.2/5.3（CentOS 5.4 上会有一些问题，并不总是版本越高越好），社区中也有许多人在使用 Ubuntu 和 Debian，如果你想用于生产环境，建议使用 LTS（Long Term Support） 的版本，即 Ubuntu8.04/10.04 或 Debian Stable。在安装之前，我们需要先准备一些环境(FreeSWITCH 可以以普通用户权限运行，但为了简单起见，以下所有操作均用 root 执行）：  
CentOS:  
yum install -y subversion autoconf automake libtool gcc-c++ ncurses-devel make  
Ubuntu:  
apt-get -y install build-essential subversion automake autoconf wget libtool libncurses5-dev  
以下三种安装方式任选其一，默认安装位置在/usr/local/freeswitch。安装过程中会下载源代码目录，请保留，以便以后升级及安装配置其它组件。  
解压缩源码包安装:  
wget <http://latest.freeswitch.org/freeswitch-1.0.5-latest.tar.gz>  
tar xvzf freeswitch-1.0.5-latest.tar.gz  
cd freeswitch-1.0.5  
./configure  
make install  
这是在在 Linux 上从源代码安装软件的标准过程。首先第 1 行下载最新的源代码，第 2 行解压缩，第 4 行配置编译环境，第 5 行编译安装。  
从 SVN 仓库安装：  
FreeSWITCH 使用 Subversion 管理源代码，从代码库安装能让你永远使用最新的版本。  
svn co <http://svn.freeswitch.org/svn/freeswitch/trunk> freeswitch  
cd freeswitch  
./bootstrap.sh  
./configure  
make install  
与上一种方法不同的是，直接从代码库安装需要执行一个 bootstrap.sh 以初始化一些环境。  
最快安装（推荐）  
wget <http://www.freeswitch.org/eg/Makefile> && make install  
以上命令会下载一个 Makefile，然后使用 make 执行安装过程。安装过程中它会从 SVN 仓库中获取代码，实际上执行的操作跟上一种安装方式相同。  
安装声音文件  
在以下例子中我们需要一些声音文件，而安装这些声音文件也异常简单。你只需在源代码目录中执行：  
make sounds-install  
make moh-install  
以下高质量的声音文件可选择安装。FreeSWITCH支持8、16、32及48kHz的语音，很少有其它电话系统支持如此多的抽样频率（普通电话是8K，更高频率意味着更好的通话质量）。  
make cd-sounds-install  
make cd-moh-install  
安装完成后，会显示一个有用的帮助，  
+---------- FreeSWITCH install Complete ----------+  
+ FreeSWITCH has been successfully installed.     +  
+                                                 +  
+       Install sounds:                           +  
+       (uhd-sounds includes hd-sounds, sounds)   +  
+       (hd-sounds includes sounds)               +  
+       ------------------------------------      +  
+                make cd-sounds-install           +  
+                make cd-moh-install              +  
+                                                 +  
+                make uhd-sounds-install          +  
+                make uhd-moh-install             +  
+                                                 +  
+                make hd-sounds-install           +  
+                make hd-moh-install              +  
+                                                 +  
+                make sounds-install              +  
+                make moh-install                 +  
+                                                 +  
+       Install non english sounds:               +  
+       replace XX with language                  +  
+       (ru : Russian)                            +  
+       ------------------------------------      +  
+                make cd-sounds-XX-install        +  
+                make uhd-sounds-XX-install       +  
+                make hd-sounds-XX-install        +  
+                make sounds-XX-install           +  
+                                                 +  
+       Upgrade to latest:                        +  
+       ----------------------------------        +  
+                make current                     +  
+                                                 +  
+       Rebuild all:                              +  
+       ----------------------------------        +  
+                make sure                        +  
+                                                 +  
+       Install/Re-install default config:        +  
+       ----------------------------------        +  
+                make samples                     +  
+                                                 +  
+       Additional resources:                     +  
+       ----------------------------------        +  
+       [http://www.freeswitch.org](http://www.freeswitch.org/)                 +  
+       [http://wiki.freeswitch.org](http://wiki.freeswitch.org/)                +  
+       [http://jira.freeswitch.org](http://jira.freeswitch.org/)                +  
+       [http://lists.freeswitch.org](http://lists.freeswitch.org/)               +  
+                                                 +  
+       irc.freenode.net / #freeswitch            +  
+-------------------------------------------------+  
至此，已经安装完了。在Unix类操作系统上，其默认的安装位置是/usr/local/freeswtich，下文所述的路径全部相对于该路径。两个常用的命令是 bin/freeswitch 和 bin/fs\_cli，为了便于使用，建议将这两个命令做符号链接放到你的搜索路径中，如：  
ln -sf /usr/local/freeswitch/bin/freeswitch /usr/local/bin/  
ln -sf /usr/local/freeswitch/bin/fs\_cli /usr/local/bin/  
当然，如果 /usr/local/bin 不在你的搜索路径中，可以把上面 /usr/local/bin 换成 /usr/bin/。 另外你也可以修改你的PATH环境变量以包含该路径。  
接下来就应该可以启动了，通过在终端中执行freeswitch命令(如果你已做符号链接的话，否则要执行/usr/local /freeswitch/bin/freeswitch)可以将其启动到前台，启动过程中会有许多log输出，第一次启动时会有一些错误和警告，可以不用理会。启动完成后会进入到系统控制台(以下称称FS-Con)。并显示类似的提示符“freeswitch@internal>”(以下简作 “FS> ”)。通过在FS-Con中输入shutdown命令可以关闭FreeSWITCH。  
如果您想将FreeSWITCH启动到后台(daemon，服务模式)，可以使用freeswitch -nc (No console)。后台模式没有控制台，如果这时想控制FreeSWITCH，可以使用客户端软件fs\_cli连接。注意，在fs\_cli中需要使用 fsctl shutdown 命令关闭 FreeSWITCH。当然，也可以直接在 Linux 提示符下通过 freeswitch -stop 命令关闭。如果不想退出 FreeSWITCH 服务，只退出fs\_cli客户端，则需要输入 /exit，或Ctrl + D，或者，直接关掉终端窗口。  
连接SIP软电话  
FreeSWITCH最典型的应用是作为一个服务器(它实际上是一个背靠背的用户代理，B2BUA)，并用电话客户端软件（一般叫软电话）连接到它。虽然 FreeSWITCH 支持 IAX、H323、Skype、Gtalk 等众多通信协议，但其最主要的协议还是 SIP。支持SIP的软电话有很多，最常用的是 X-Lite 和 Zoiper。这两款软电话都支持 Linux、MacOSX 和 Windows平台，免费使用但是不开源。在 Linux 上你还可以使用 ekiga 软电话。  
强烈建议在同一局域网上的其它机器上安装软电话，并确保麦克风和耳机可以正常工作 。当然，如果你没有多余的机器做这个实验，那么你也可以在同一台机器上安装。只是需要注意，软电话不要占用 UDP 5060 端口，因为 FreeSWITCH 默认要使用该端口，这是新手常会遇到的一个问题。你可以通过先启动 FreeSWITCH 再启动软电话来避免该问题，另外有些软电话允许你修改本地监听端口。  
通过输入以下命令可以知道 FreeSWITCH 监听在哪个IP地址上，记住这个 IP 地址(:5060以前的部分)，下面要用到：  
netstat -an | grep 5060  
FreeSWITCH 默认配置了 1000 ~ 1019 共 20 个用户，你可以随便选择一个用户进行配置：  
在 X-Lite 上点右键，选 Sip Account Settings...，点Add添加一个账号，填入以下参数(Zoiper 可参照配置)：  
Display Name: 1000  
User name: 1000  
Password: 1234  
Authorization user name: 1000  
Domain: 你的IP地址，就是刚才你记住的那个  
其它都使用默认设置，点 OK 就可以了。然后点 Close 关闭 Sip Account 设置窗口。这时 X-Lite 将自动向 FreeSWITCH 注册。注册成功后会显示"Ready. Your username is 1000"，另外，左侧的“拨打电话”（Dial）按钮会变成绿色的。如下图。  
激动人心的时刻就要来了。输入“9999”按回车（或按绿色拨打电话按钮），就应该能听到保持音乐(MOH, Music on Hold)。如果听不到也不要气馁，看一下 FS-Con 中有没有提示什么错误。如果有“File Not Found”之类的提示，多半是声音文件没有安装，重新查看 make moh-install 是否有错误。接下来，可以依次试试拨打以下号码：  
------------------  
号码        |   说明  
----------------------  
9999      |   保持音乐  
9996      |   echo，回音测试   
9992      |   info，在控制台上显示呼叫参数  
9888      |   FreeSWITCH电话会议，每周召开  
5900      |   呼叫挂起  
5901      |   接听挂起的呼叫  
5000      |   示例IVR  
4000      |   听取语音信箱  
33xx      |   电话会议，48K(其中xx可为00-99，下同)  
32xx      |   电话会议，32K  
31xx      |   电话会议，16K  
30xx      |   电话会议，8K  
2000-2002 |   呼叫组  
1000-1019 |   默认分机号  
表一： 默认号码及说明  
另外，也许你想尝试注册另外一个SIP用户并在两者间通话。最好是在同一个局域网中的另外一台机器上启动另一个 X-Lite ，并使用 1001 注册，注册完毕后就可以在 1000 上呼叫 1001，或在 1001 上呼叫 1000 。当然，你仍然可以在同一台机器上做这件事（比方说用Zoiper注册为1001），需要注意的是，由于你机器上只有一个声卡，两者可能会争用声音设备。特别是在Linux上，有些软件会独占声音设备。如果同时也有一个USB接口的耳机，那就可以设置不同的软件使用不同的声音设备。  
配置简介  
FreeSWITCH配置文件默认放在 conf/， 它由一系列XML配置文件组成。最顶层的文件是freeswitch.xml，系统启动时它依次装入其它一些XML文件并最终组成一个大的XML文件。  
文件                                |    说明  
---------------------------------------------------  
vars.xml                          | 一些常用变量  
dialplan/default.xml              | 缺省的拨号计划  
directory/default/\*.xml           | SIP用户，每用户一个文件  
sip\_profiles/internal.xml         | 一个SIP profile，或称作一个SIP-UA，监听在本地IP及端口5060，一般供内网用户使用  
sip\_profiles/externa.xml          | 另一个SIP-UA，用作外部连接，端口5080  
autoload\_configs/modules.conf.xml | 配置当FreeSWITCH启动时自动装载哪些模块  
添加一个新的SIP用户  
FreeSWITCH默认设置了20个用户(1000-1019)，如果你需要更多的用户，或者想通过添加一个用户来学习FreeSWITCH配置，只需要简单执行以下三步：  
    \* 在 conf/directory/default/ 增加一个用户配置文件  
    \* 修改拨号计划(Dialplan)使其它用户可以呼叫到它  
    \* 重新加载配置使其生效  
如果想添加用户Jack，分机号是1234。只需要到 conf/directory/default 目录下，将 1000.xml 拷贝到 1234.xml。打开1234.xml，将所有1000都改为1234。并把 effective\_caller\_id\_name 的值改为 Jack，然后存盘退出。如：  
<variable name="effective\_caller\_id\_name" value="Jack"/>  
接下来，打开 conf/dialplan/default.xml，找到 <condition field="destionation\_number"  
expression="^(10[01][0-9])$"> 一行，改为 <condition field="destionation\_number" expression="^(10[01][0-9]|1234)$">。熟悉正则表达式的人应该知道，“^(10[01][0-9])$”匹配被叫号码1000-1019。因此我们修改之后的表达式就多匹配了一个1234。FreeSWITCH使用Perl兼容的正则表达式(PCRE)。  
现在，回到FS-Con，或启动fs\_cli，执行 reloadxml 命令或按快捷键F6，使新的配置生效。  
找到刚才注册为1001的那个软电话(或启动一个新的，如果你有足够的机器的话)，把1001都改为1234然后重新注册，则可以与1000相互进行拨打测试了。如果没有多台机器，在同一台机器上运行多个软电话可能有冲突，这时，也可以直接进在FreeSWITCH控制台上使用命令进行测试：  
FS> sofia status profile internal  (显示多少用户已注册）  
FS> originate sofia/profile/internal/1000 &echo  (拨打1000并执行echo程序）  
FS> originate user/1000 &echo  (同上）  
FS> originate sofia/profile/internal/1000 9999    (相当于在软电话1000上拨打9999)  
FS> originate sofia/profile/internal/1000 9999 XML default   (同上)  
其中，echo() 程序一个很简单的程序，它只是将你说话的内容原样再放给你听，在测试时很有用，在本书中，我们会经常用它来测试。  
FreeSWITCH用作软电话  
FreeSWITCH也可以简单的用作一个软电话，如X-Lite. 虽然相比而言比配置X-Lite略微麻烦一些，但你会从中得到更多好处：FreeSWITCH是开源的，更强大、灵活。关键是它是目前我所知道的唯一支持CELT高清通话的软电话。  
FreeSWITCH使用mod\_portaudio支持你本地的声音设备。该模块默认是不编译的。到你的源代码树下，执行：  
make mod\_portaudio  
make mod\_portaudio-install  
其它的模块也可以依照上面的方式进行重新编译和安装。然后到FS-Con中，执行:  
FS> load mod\_portaudio  
如果得到“Cannot find an input device”之类的错误可能是你的声卡驱动有问题。如果是提示“+OK”就是成功了，接着执行：  
FS> pa devlist  
API CALL [pa(devlist)] output:  
0;Built-in Microphone;2;0;  
1;Built-in Speaker;0;2;r  
2;Built-in Headphone;0;2;  
3;Logitech USB Headset;0;2;o  
4;Logitech USB Headset;1;0;i  
以上是在我笔记本上的输出，它列出了所有的声音设备。其中，3和4最后的“o”和“i”分别代表声音输出(out)和输入(in)设备。在你的电脑上可能不一样，如果你想选择其它设备，可以使用命令：  
FS> pa indev #0  
FS> pa outdev #2  
以上命令会选择我电脑上内置的麦克风和耳机。  
接下来你就可以有一个可以用命令行控制的软电话了，酷吧？  
FS> pa looptest    (回路测试，echo)  
FS> pa call 9999  
FS> pa call 1000  
FS> pa hangup  
如上所示，你可以呼叫刚才试过的所有号码。现在假设想从SIP分机1000呼叫到你，那需要修改拨号计划(Dialplan)。用你喜欢的编辑器编辑以下文件放到conf/dialplan/default/portaudio.xml  
<include>  
  <extension name="call me">  
    <condition field="destination\_number" expression="^(me|12345678)$">  
      <action application="bridge" data="portaudio"/>  
    </condition>  
  </extension>  
</include>  
然后，在FS-Con中按“F6”或输入以下命令使之生效：  
FS> reloadxml  
在分机1000上呼叫“me”或“12345678”(你肯定想为自己选择一个更酷的号码)，然后在FS-Con上应该能看到类似“[DEBUG] mod\_portaudio.c:268 BRRRRING! BRRRRING! call 1”的输出（如果看不到的话按“F8”能得到详细的Log），这说明你的软电话在振铃。多打几个回车，然后输入“pa answer”就可以接听电话了。“pa hangup”可以挂断电话。  
当然，你肯定希望在振铃时能听到真正的振铃音而不是看什么BRRRRRING。好办，选择一个好听一声音文件(.wav格式)，编辑conf/autoload\_configs/portaudio.conf.xml，修改下面一行：  
<param name="ring-file" value="/home/your\_name/your\_ring\_file.wav"/>  
然后重新加载模块：  
FS> reloadxml  
FS> reload mod\_portaudio  
再打打试试，看是否能听到振铃音了？  
如果你用不惯字符界面，可以看一下FreeSWITCH-Air([http://www.freeswitch.org.cn](http://www.freeswitch.org.cn/) /download)，它为 FreeSWITCH 提供一个简洁的软电话的图形界面。另外，如果你需要高清通话，除需要设置相关的语音编解码器(codec)外，你还需要有一幅好的耳机才能达到最好的效果。本人使用的是一款USB耳机。  
配置SIP网关拨打外部电话  
如果你在某个运营商拥有SIP账号，你就可以配置上拨打外部电话了。该SIP账号（或提供该账号的设备）在 FreeSWITCH 中称为SIP网关（Gateway）。添加一个网关只需要在 conf/sip\_profiles/external/ 创建一个XML文件，名字可以随便起，如gw1.xml。  
<gateway name="gw1">   
    <param name="realm" value="SIP服务器地址，可以是IP或IP:端口号"/>  
    <param name="username" value="SIP用户名"/>  
    <param name="password" value="密码"/>  
    <param name="register" value="true" />  
</gateway>  
如果你的SIP网关还需要其它参数，可以参阅同目录下的 example.xml，但一般来说上述参数就够了。你可以重启 FreeSWITCH，或者执行以下命令使用之生效。  
FS> sofia profile external rescan reloadxml  
然后显示一下状态：  
FS> sofia status  
如果显示 gateway gw1 的状态是 REGED ，则表明正确的注册到了网关上。你可以先用命令试一下网关是否工作正常：  
FS> originate sofia/gateway/gw1/xxxxxx &echo()  
以上命令会通过网关 gw1 呼叫号码 xxxxxx（可能是你的手机号），被叫号码接听电话后，FreeSWITCH 会执行 echo() 程序，你应该能听到自己的回音。  
从某一分机上呼出  
如果网关测试正常，你就可以配置从你的SIP软电话或portaudio呼出了。由于我们是把 FreeSWITCH 当作 PBX 用，我们需要选一个出局字冠。常见的 PBX 一般是内部拨小号，打外部电话就需要加拨 0 或先拨 9 。当然，这是你自己的交换机，你可以用任何你喜欢的数字（甚至是字母）。 继续修改拨号计划，创建新XML文件： conf/dialplan/default/call\_out.xml :  
<include>  
  <extension name="call out">  
    <condition field="destination\_number" expression="^0(\d+)$">  
      <action application="bridge" data="sofia/gateway/gw1/$1"/>  
    </condition>  
  </extension>  
</include>  
其中，(\d+)为正则表达式，匹配 0 后面的所有数字并存到变量 $1 中。然后通过 bridge 程序通过网关 gw1 打出该号码。当然，建立该XML后需要在Fs-Con中执行 reloadxml 使用之生效。  
呼入电话处理。  
如果你的 SIP 网关支持呼入，那么你需要知道呼入的 DID 。 DID的全称是 Direct Inbound Dial，即直接呼入。一般来说，呼入的 DID 就是你的 SIP 号码，如果你不知道，也没关系，后面你会学会如何得到。 编辑以下XML文件放到 conf/dialplan/public/my\_did.xml  
<include>  
  <extension name="public\_did">  
    <condition field="destination\_number" expression="^(你的DID)$">  
      <action application="transfer" data="1000 XML default"/>  
    </condition>  
  </extension>  
</include>  
reloadxml 使之生效。上述配置会将来话直接转接到分机 1000 上。在后面的章节你会学到如何更灵活的处理呼入电话，如转接到语音菜单或语音信箱等。  
小结  
其实本章涵盖了从安装、配置到调试、使用的相当多的内容，如果你能顺利走到这儿，你肯定对 FreeSWITCH 已经受不释手了。如果你卡在了某处，或某些功能未能实现，也不是你的错，主要是因为 FreeSWITCH 博大精深，我不能在短短的一章内把所有的方面解释清楚。在后面的章节中，你会学到更多的基本概念、更加深入地了解 FreeSWITCH 的哲学，学到更多的调试技术和技巧，解决任何问题都会是小菜一碟了。