第十章 Parking and Paging

目录

[译者注：] Parking 直译是“停车”，Paging 直译是“传呼”（注：Paging 在本文可以翻译为广播）。我找不到合适的中文术语，所以这两个术语本章还是直接引用英文原文。但为便于读者理解，此处先举个例子说明这两个功能：请设想一个大仓库，内部安装了多部分机电话。现在有个来电打到总机，请接线员转给仓管员小 A。但是仓库很大，小 A 并不会固定待在仓库的特定位置（如某一分机附近），那么怎么办呢？接线员可以这样做：首先把这个电话转到一个内部服务号上（如 701），这就是所谓 Parking 了（确实很像停车，车位编号 701）；然后让所有电话都进入“免提自动应答”状态，充当个广播喇叭的作用，这样整个仓库都听到接线员喊话：“小 A 电话，在 701 线”，这个功能就是所谓的 Paging 了。小 A 听到后，就可以在任意一个分机上接起这个之前 Parking 到 701 上的电话。

本章集中讨论 PBX 系统的两个重要方面：Parking 一个呼叫并允许它在另外一部分机上恢复，以及 Paging 功能。利用 paging 功能，我们可以通报这个呼叫时找谁的以及如何接听这个呼叫。

在 Asterisk 中，这两个功能是彼此独立的，并且都可以单独使用。一些拥有大型仓库，或者那些员工在办公室中不断移动而不需要整天坐在办公桌旁的商业机构可以利用 paging and parking 功能在办公室中转接电话。在本章中，我们将向你展示如何在传统环境中利用 parking and paging，以及一些对这一常用功能的更现代用法。

11.1 features.conf

Asterisk 支持所有现代 PBX 系统的通用特性。许多这些特性都需要一些可选得配置参数。*features.conf* 就是 Asterisk 中用于修改或定义特性参数的文件。

基于 DTMF 的特性

在 *features.conf* 里定义的许多参数只有当呼叫已经通过 Dialplan 中的 Dial()或 Queue()应用程序伴随 K,k,H,h,T,t,W,w,X,x 等参数被成功桥接之后才能生效。用这种方法使用的特性称为基于 DTMF 的特性（意思是它们不可以通过 SIP 消息使用，而只能通过语音通道的信号音才能触发）。[注 1]

SIP channel 的呼叫转移（例如从一部 SIP 电话机上）可以被 SIP 话机自身的功能处理，不会受到 *features.conf* 中的任何定义影响。

11.1.1 The [general] section

在 *features.conf* 的 [general] 部分，你可以定义一些选项微调 Asterisk 中 park 和 transfer 的一些行为。这些选项参见表格 1

表格 1 features.conf [general] section

Option	Value/Example	Note
parkext	700	Sets the default extension used to park calls.
parkpos	701-720	Sets the range of extensions used as the parking lot. Parked calls may be retrieved by dialing the numbers

		in this range.
context	parkedcalls	Sets the default dialplan context where the parking extension and the parking lot extensions are created.
parkinghints	no	Enables/disables automatic creation of dialplan hints for the parking lot extensions so that phones can subscribe to the state of extensions in the parking lot. The default is no.
parkingtime	45	Specifies the number of seconds a call will wait in the parking lot before timing out.
comebacktoorigin	yes	Configures the handling of timed-out parked calls. For more information on the behavior of this option, see the sidebar titled “Handling Timed-Out Parked Calls with the comebacktoorigin Option” in this chapter.
courtesytone	beep	Specifies the sound file to be played to the parked caller when the parked call is retrieved from the parking lot.
parkedplay	caller	Indicates which side of the call to play the courtesytone to when a parked call is picked up. Valid options include callee, caller, both, or no. The default is no.
parkedcalltransfers	caller	Controls which side of a call has the ability to execute a DTMF-based transfer in the call that results from picking up a parked call. Valid options include callee, caller, both, or no. The default is no.
parkedcallreparking	caller	Controls which side of a call has the ability to execute a DTMF-based park in the call that results from picking up a parked call.[注2] Valid options include callee, caller, both, or no. The default is no.
parkedcallhangup	caller	Controls which side of a call has the ability to execute a DTMF-based hangup in the call that results from picking up a parked call. Valid options include callee, caller, both, or no. The default is no.
parkedcakkrecording	caller	Controls which side of a call has the ability to initiate a DTMF-based one touch recording in the call that results from picking up a parked call. Valid options include callee, caller, both, or no. The default is no.
parkeddynamic	yes	Enables the dynamic creation of parking lots in the dialplan. The channel variables PARKINGDYNAMIC, PARKINGDYNCONTEXT, and PAR KINGDYNPOS need to be set.
adsipark	yes	Passes ADSI information regarding the parked call back to the originating set.
findslot	next	Configures the parking slot selection behavior. See ??? for more details.

parkedmusicclass	default	Specifies the class to be used for the music on hold played to a parked caller. A music class set in the dialplan using the CHANNEL(musicclass) dialplan function will override this setting
transferdigittimeout	3	Sets the number of seconds to wait for each digit from the caller executinga transfer.
xfersound	beep	Specifies the sound to be played to indicate that an attended transfer is complete.
xferfailsound	beeperr	Specifies the sound to be played to indicate that an attended transfer has failed to complete.
pickupexten	*8	Configures the extension used for call pickup.
pickupsound	beep	Specifies the sound to be played to indicate a successful call pickup attempt. No sound is played by default.
pickupfailsound	beeperr	Specifies the sound to be played to indicate a failed call pickup attempt. No sound is played by default.
featuredigittimeout	1000	Sets the number of milliseconds to wait in between digits pressed during a bridged call when matching against DTMF activated call features.
atxfernoanswertimeout	15	Configures the number of seconds to wait for the target of an attended transfer to answer before considering the attempt timed out.
atxferdropcall	no	Configures behavior of attended transfer call handling when the transferer hangs up before the transfer is complete and the transfer fails. By default, this option is set to no and a call will be originated to attempt to connect the transferee back to the caller that initiated the transfer. If set to yes, the call will be dropped after the transfer fails.
atxferloopdelay	10	Sets the number of seconds to wait in between callback retries if atx ferdropcall is set to no.
atxfercallbackretries	2	Sets the number of callback attempts to make if atxferdropcall is set to no. By default, this is set to 2 callback attempts.

根据 comebacktoorigin 选项处理超时 Parked Calls

这个选项配置当 parked call 超时情况下的行为。**comebacktoorigin** 可能的取值包括：

yes*(default)*

当 parked call 超时后，Asterisk 会尝试将呼叫转回发起这个 parked call 的 channel。如果这个 channel 不再可用，Asterisk 会挂断这个呼叫。

no

当你希望对于超时的 parked call 执行一些客户化的 dialplan 功能时，可以使用这个选项。这个 parked call 会在超后被转给 dialplan 中的特定位置，在那里会完美的处理这个呼叫（这可能是简单的把这个呼叫转给另一部分机，或者按某种排序做查找）。

你也许需要顾及这样一些 parked calls，即无法把这些 parked call 返回到发起它们的 channel。举例来说，当发起这个 parked call 的 channel 同时也是另一个系统的 trunk 时，将没有足够的信息把这个 parked call 返回到另一个系统中的正确用户。超时后实际要求的处理会比 **comebacktoorigin=yes** 能够处理的情况要复杂得多。

当 **comebacktooriginal=no** 时超时的 parked calls 会被转到 **parkedcallstimeout** context 处理。

Dialplan（和 contexts）的细节参见第六章。

转回的目标分机由 parked 这个呼叫的 channel 名构成。例如，如果名为 **0004F2040808** 的 SIP peer parked 了这个呼叫，则转回的目标分机是 **SIP_0004F2040808**。

如果这个分机不存在，这个呼叫会被转到 **parkedcallstimeout** context 中的 **s** 分机。最后，如果 **parkedcallstimeout** 的 **s** 分机也不存在，呼叫会被转到 **default** context 的 **s** 分机。

另外，对于配置 comebacktooriginal=no 的呼叫，Asterisk 会在 park-dial context 下创建一个 SIP_0004F2040808 分机。这个分机用于利用 Dial() 拔打 SIP/0004F2040808。[注3]

11.1.2 The [featuremap] Section

在这部分可以定义特定的 DTMF 序列，它们将触发连接的 Channel 上的不同特性，通过 Dial() 或 Queue() 应用程序的选项。请参见

Option	Value/Example	Notes	Dial()/Queue() Flags
blindxfer	#1	Invokes a blind (unsupervised) transfer	T , t
disconect	*0	Hangs up the call	H , h

automon	*1	Starts recording of the current call using the Monitor() application (pressing this key sequence a second time stops the recording)	W , w
atxfer	*2	Performs an automated transfer	T , t
parkcall	#72	Parks a call	K , k
automixmon	*3	Starts recording of the current call using the MixMonitor() application (pressing this key sequence again stops the recording)	X , x

默认的 blindxfer 和 disconnect 特性码分别是 # 和 * 。通常，你可能想要修改它们的默认值，因为它们可能用于你希望做的其它事（例如，如果你在你的 **Dial()**命令中使用了 Tt 选项，每次你按下 # 时都会发起一个 transfer）。

11.1.3 The [applicationmap] Section

features.conf 的这部分用于映射 DTMF 码到 dialplan 应用程序。主叫将进入 hold 状态，直到相应的应用程序完全执行完。

定义一个应用程序映射的语法如下所示（必须被写在同一行，换行是不允许的）[注4]：

<FeatureName> => *<DTMF_sequence>*,*<ActivateOn>*[/*<ActivatedBy>*]

,*<Application>*([*<AppArguments>*])[,*MOH_Class*]

你需要做下述事宜：

1. 给定映射一个名字，从而可以通过使用 channel 变量 **DYNAMIC_FEATURES** 来使能它；
2. 定义一个 DTMF 序列来激活这个特性（我们建议至少使用两个按键）；
3. 定义这个特性在哪个 channel 上起作用，以及（这是可选项）哪一方可以激活这个特性（默认是双方都可以激活）；
4. 给出这个映射激活的应用程序的名字，以及相关参数；
5. 给这个特性分配一个呼叫保留音乐组（MOH, Music On Hold），这样对端用户就会在应用程序执行时听到这个音乐。如果你没有定义 MOH 组，对端用户则听不到任何声音。

下面是触发一个 AGI 脚本的应用程序映射的例子：

```
agi_test => *6,self/callee,AGI(agi-test.agi),default
```

由于通过应用程序映射执行的应用程序是在 PBX 内核程序之外的，所以你不能执行任何触发 dialplan 的应用程序（例如 Goto()，Macro()，Background()，等等）。如果你希望利用应用程序映射执行一些外部处理（包括执行 dialplan 代码），你需要通过 AGI()或 System()应用程序来触发外部应用程序。要特别指出的是，如果你希望利用应用程序映射做一些复杂的工作，你需要非常小心的测试，因为并不是所有事情都会像你期望的那样执行。

为了使用一个应用程序映射，你必须在 dialplan 中通过在 Dial()命令执行前设置 DYNAMIC_FEATURES 变量来声明它。在变量名前使用双下划线是为了保证应用程序映射在呼叫生命周期内的两个 channels 中都有效。举例：

```
exten => 101,n,Set(__DYNAMIC_FEATURES=agi_test)
exten => 101,n,Dial(SIP/0000FFFF0002)
```

如果你需要在一个呼叫中允许多于一个的应用程序映射，你需要使用 # 做为多个映射名之间的分隔符：

```
Set(__DYNAMIC_FEATURES=agi_test#my_other_map)
```

使用 # 而不是简单的使用逗号作为分隔符的原因是，老版本的 Set()对于逗号的解释与新近的版本不同，而应用程序映射的语法一直没有升级。

不要忘记在修改了 *features.conf* 文件后重现加载 features module：

```
*CLI> features reload
```

你可以通过 CLI 命令 *features show* 来确认你的修改是否生效。请确保你在将你设计的应用程序映射交付客户前测试过它们！

继承 Channel 变量

Channel 变量总是与最初设置它们的 channel 关联，并且一旦这个 channel 被呼转则变量不再有效。

为了允许 channel 变量能够跟随这个 channel 在系统内呼转，必须使用 channel 变量继承。有两种修饰符可以让 channel 变量能够跟随 channel：单下划线和双下划线。

单下划线使 channel 变量可以再一次呼转中被继承，而在其后的呼转中将不再有效。如果你使用双下划线，channel 变量将在这个 channel 的整个生命周期中被继承。

设置 channel 变量继承只需要简单的在 channel 名前增加一个单下划线或双下划线前缀。不过 channel 变量仍然采用不加前缀之前的方法来引用（例如，不要尝试用带下划线的变量名来读取变量的值）。

下面是设置单次呼转继承的 channel 变量的例子：

```
exten => example,1,Set(_MyVariable=thisValue)
```

下面是设置永久呼转继承的 channel 变量的例子：

```
exten => example,1,Set(__MyVariable=thisValue)
```

要读取 channel 变量的值，不要使用下划线：

```
exten => example,1,Verbose(1,Value of MyVariable is: ${MyVariable})
```

11.1.4 Application Map Grouping

如果你有许多 features 需要对特定的 context 或 extension 使用，你可以在 application map grouping 中把这样几个 features 组成一组。因此，当用 DYNAMIC_FEATURES 变量指定组名时相当于设置了组中所有的 features 到映射中。

这种 application map groupings 增加在 features.conf 的最后。每个组指定一个名字，其后跟随相关的特性列表。

```
[shifteight]
unpauseMonitor => *1        ; custom key mapping
pauseMonitor => *2          ; custom key mapping
agi_test =>                 ; no custom key mapping
```

如果你希望在 application map grouping 中为某个 feature 指定自定义的按键映射，简单的在 => 输入你希望按键映射就可以。如果你没有指定按键映射，则该特性就会使用默认的按键映射（在[featuremap]部分定义）。不管你是否指定自定义按键映射，=> 操作符都是必须的。

在 dialplan 中，你可以用 Set()来设置这个 application map grouping：

```
Set(__DYNAMIC_FEATURES=shifteight)      ; use the double underscore if you want to ensure
                                        ; both call legs have the variable assigned.
```

11.1.5 Parking Lots

Parking lot 允许呼叫被保留在系统中而不与任何 extension 关联。然后这个呼叫还可以被知道对应 park code 的任何人恢复。这个特性通常与吸顶式广播系统一起使用。由于这个原因，它一般被称为 park-and-page。然而，需要注意的是实际上 parking 和 paging 功能是独立的。

为了在 Asterisk 中 park 一个呼叫，你需要将呼叫呼转到 feature code 指定的 parking 号码，它通过 *features.conf* 文件中的 **parkext** 来配置，默认是 700：

```
parkext => 700          ; What extension to dial to park (all parking lots)
```

你必须等待这个呼转操作完成，直到你从系统得到一个 parking 恢复号码，否则你将无法恢复这个呼叫。默认的呼转恢复号码，用 *features.conf* 中 **parkpos** 指定，编码范围是 701-720：

```
parkpos => 701-720 ; What extensions to park calls on (defafult parking lot)
```

一旦呼叫被 parked，任何人都可以通过拨打指定给这个呼叫的恢复号码（parkpos）来恢复通话。然后这个呼叫就会被桥接到拨打了恢复号码的 channel。

有两种常用方法来定义怎么分配恢复号码。这可以通过 *features.conf* 中的 **findslot** 来定义。默认的方法（**findslot => first**）总是使用最小的空闲号码。第二种方法（**findslot => next**）循环使用恢复号码并每次指定下一个，当最后一个恢复号码使用后怎回到第一个恢复号码。具体选择哪种方法取决于你的 parking 系统有多忙。如果你很少使用 parking，**findslot** 的默认值 **first** 是最佳选择（人们将会总是使用同一个恢复号码）。如果你的 parking 系统很繁忙（例如，当用于一个汽车销售店时），那么每个后继的 parking 呼叫采用下一个恢复号码会更好，因为你经常会遇到同时有多个 parking 呼叫。用户需要仔细听用的是哪个恢复号码（而不是总是 **701**），而且有时会发生意外的恢复了错误的呼叫的情况。

如果你使用了 parking 功能，你大概还需要一个方法宣布这个 parked 呼叫，从而可以让接听人知道如何恢复这个呼叫。当然你也可以跑到大厅大喊“Bob，有你的电话在 701 线！”，更专业的做法是使用 paging 系统（更正式的名字是公共广播系统），

11.2 吸顶式和“下巴底下式”Paging （a.k.a. 公共广播）

在许多 PBX 系统中，都允许用户通过电话把他的声音发送到公共广播系统中。这一般要求拨打一个 feature code 或者 extension 和公共广播资源建立连接，然后通过电话机的手柄讲话，就可以广播到所有相连的公共广播设备上。通常，这是一种包括一个连接吸顶式喇叭的放大器的外接广播设备。然而，通过办公电话的免提喇叭的广播系统也非常流行（主要是出于成本原因）。如果你预算足够（或者已经有了一个吸顶式广播系统），吸顶式广播系统一般来说要更好一些，但是基于办公电话的广播系统（a.k.a. “下巴底下式”广播系统）在大多数环境也工作的很好。更常见的情况是混合的广播系统，在这种方案中，举例来说，基于办公电话的广播系统被用于办公室环境，而基于吸顶式喇叭的广播系统被用于仓库，走廊，以及一些公共区域（食堂，接待处等）。

在 Asterisk 中，Page()应用程序用于广播。这个应用程序简单的用一组 channels 作为参数，同时呼叫每一个 channels，然后，当它们应答后，将每一个都放入会议室。需要注意的是，很明显，*paging 系统能工作的一个条件式每个目标 channel 都可以自动应答来电*并能用某种方式把语音输出到免提喇叭（换句话说，如果所有被叫电话只是振铃的话，**Page()**无法工作）。

所以，虽然 Page()应用程序本身的使用非常简单，但让所有的目标 channel 都正确的处理来电广播却要更麻烦一些。我们将简要说明如下。

Page()应用程序有三个参数，分别是需要广播的 channel 组，选项，以及超时时间：

```
exten => *724,1,Page(${ChannelsToPage},i,120)
```

选项给 Page()的工作方式提供了一些灵活性，但是大部分这些配置都与目标设备如何处理来电连接有关。我们将在下面的部分研究配置设备接收广播的不同方法。

Option	Description	Discussion
d	Enables full-duplex audio	Sometimes referred to as “talkback paging,” the use of this option implies that the equipment that receives the page has the ability to transmit audio back at the same time as it is receiving audio. Generally, you would not want to use this unless you had a specific need for it.
i	Ignores attempts to forward the call	You would normally want this option enabled.
q	Does not play beep to caller (quiet mode)	Normally you won’t use this, but if you have an external amplifier that provides its own tone, you may want to set this option.
r	Records the page into a file	If you intended on using the same page multiple times in the future, you could record the page and then use it again later by triggering it using Originate() or using the A(x) option to Page().
s	Dials a channel only if the device state	This option is likely only useful (and reliable)

	is NOT_INUSE	on SIP-bound channels, and even so may not work if a single line is allowed multiple calls on it. Therefore, don’t rely on this option in all cases.
A(x)	Plays announcement x to all participants	You could use a previously recorded file to be played over the paging system. If you combined this with Originate() and Record(), you could implement a delayed paging system.
n	Does not play announcement simultaneously to caller (implies A(x))	By default the system will play the paged audio to both the caller and the callee. If this option is enabled, the paged audio will not be played to the caller (the person paging).

由于 Page()的工作方式，它是非常消耗资源的。我们怎么强调这一点都不过分。仔细阅读本书，我们将讨论如何保证 paging 不会在一个产品环境中导致性能问题（如果没有正确的设计，这几乎是必然的）。

11.2.1发送你的广播

如我们之前所述，Page()本身是非常简单的。窍门在于如何把它们组织到一起。广播可以发送到不同类型的 channels 中，而且它们都需要不同的配置。

11.2.1.1 外部广播系统

如果建筑中已经安装了公共广播系统，通常的做法是连接电话系统到一个外部放大器并通过一个呼叫发送广播到公共广播系统中去。做到这一点的一种方法是在你的服务器上安装一个声卡连接到外部放大器，然后把呼叫发送到名为 Console/DSP 的 channel，但是这是假设你服务器上的声卡驱动工作正常并且正确的设置了音量。另一种，更简单，也更可靠的连接外部广播设备的方法是使用某种带有 FXS 接口的设备（例如 ATA），该设备被连接到诸如 Bogen UTI1 [注5] 这样的广播设备接口上，这样就连接到外部广播放大器[注6] 上了。

在你的 dialplan 中，向一个外部放大器发送广播的操作就是简单的 Dial()到连接着广播设备的设备上。举例来说，如果你有一个 ATA 在 *sip.conf* 中配置为**[PagingATA]**，并且你将这个 ATA 连接到 UTI1，你可以这样来执行广播：

```
exten => *724,1,Verbose(2,Paging to external amplifier)          ; note the '*' in the
                                                                 ; extension is part of
                                                                 ; what you actually dial
    same => n,Set(PageDevice=SIP/PagingATA)
    same => n,Page(${PageDevice},i,120)
```

请注意为了这个系统能正常工作，你必须在 *sip.conf* 中将你的 ATA 作为一个 SIP 设备注册，在这个例子中我们命名这个设备为**[PagingATA]**。你可以用任何你喜欢的名字命名这个设备（例如，通常我们用 MAC 地址作为 SIP 设备的名字），但是无论如何它不是一个用户电话机，

使用一个有别于其它设备的名字是有帮助的。

如果你在系统中使用了一张 FXS 板卡并连接到 UNI1 上，你可以 **Dial()**到这个 channel 来代替 FXS port：

```
same => n,Dial(DAHDI/25)
```

UTI1 应答这个呼叫并且打开连接到广播系统的 channel；然后你就可以发表广播了。

11.2.1.2 基于电话机的广播系统

基于电话机的广播系统第一次变得流行是在按键式电话系统中，在那里办公电话机的免提喇叭被用作穷人的公共广播系统。大部分 SIP 电话机都具备免提自动接听呼叫的能力，这就实现了基于电话机广播系统的要求。除此而外，还需要同时把语音送往多余一部电话机。Asterisk 使用它内置的会议引擎处理这些内部细节。你可以通过使用 **Page()**应用程序实现这一目的。

像 **Dial()**一样，**Page()**应用程序可以处理多个 channels。由于你通常希望 Page()一次给多个话机（或者全部的话机）发信号，你可能只能采用如下所示的冗长设备字符串：

```
Page(SIP/SET1&SIP/SET2&SIP/SET3&SIP/SET4&SIP/SET5&SIP/SET6&SIP/SET7&...
```

当超过一定规模后，你的 Asterisk 系统将无法向更多的话机发送广播。例如，如果一个办公室有 200 个电话机，利用 SIP 来广播每一个设备是不可能的；你 Asterisk 服务器的流量和 CPU 负荷会变得过重。在这种情况下，你应该考虑或者采用多播广播或者外部广播系统。

或许基于 SIP 的广播系统的最棘手部分是这样一件事，你必须告诉每一部话机它必须自动应答，但是不同的 SIP 话机制造商使用不同的 SIP 信息来实现这一点。所以，依赖于你所使用的电话型号，用于实现给予 SIP 话机的广播命令也会不同。举例如下：

- For Aastra:

```
exten => *724,1,Verbose(2,Paging to Aastra sets)
    same => n,SIPAddHeader(Alert-Info: info=alert-autoanswer)
    same => n,Set(PageDevice=SIP/00085D000000)
    same => n,Page(${PageDevice},i)
```

- For Polycom:

```
exten => *724,1,Verbose(2,Paging to Polycom sets)
    same => n,SIPAddHeader(Alert-Info: Ring Answer)
    same => n,Set(PageDevice=SIP/0004F2000000)
    same => n,Page(${PageDevice},i)
```

- For Snom:

```
exten => *724,1,Verbose(2,Paging to Snom sets)
    same => n,Set(VXML_URL=intercom=true)
    ; replace 'domain.com' with the domain of your system
    same => n,SIPAddHeader(Call-Info: sip:domain.com\;answer-after=0)
```

```
same => n,Set(PageDevice=SIP/000413000000)
same => n,Page(${PageDevice},i)
```

- For Cisco SPA (the former Linksys phones, not the 79XX series):

```
exten => *724,1,Verbose(2,Paging to Cisco SPA sets -- but not Cisco 79XX sets)
same => n,SIPAddHeader(Call-Info:\;answer-after=0) ; Cisco SPA phones
same => n,Set(PageDevice=SIP/0004F2000000)
same => n,Page(${PageDevice},i)
```

估计你已经发现，如果在你的环境中混合了不同的电话机，会发生什么？你如何控制哪种信息头发送给哪种电话机？[注7]
无论你从哪方面考虑，这都不好实现。

幸运的是，许多 SIP 话机都支持 IP 组播，这是一种向多个话机发送广播的好得多的办法（请仔细阅读）。然而，如果在你的系统中只有少量电话机并且都来自于同一个供应商，基于 SIP 的广播将是最简单的方法，所以我们并不想完全把你吓跑。

11.2.1.3 通过MulticastRTP channel实现多播广播

如果你认证对待通过话机实现广播，并且你有超过 100 个话机，你需要考虑 IP 多播。IP 多播的概念已经存在很长时间了[注8]，但它并没有被广泛应用。然而，它是在一个地点实现广播系统的理想方法。

Asterisk 有一个 channel（chan_multicast_rtp）是设计用于创建 RTP 多播的。然后这个流会被不同的电话订阅，结果是当声音出现在多播流上时，电话机会将这个声音通过免提喇叭播放出来。

因为 MulticastRTP 是一个 channel 驱动，所以它没有独立的应用程序。但是你可以在 dialplan 中任何可以使用 channel 的地方使用它。在我们的例子中，我们将使用 Page()应用程序来初始化我们的多播。

为了使用多播 channel，你只要像处理任何其它 channel 一样简单的发送呼叫给它就可以。语法如下：

```
MulticastRTP/<type>/<ip address:port>[/<linksys address:port>]
```

其中 type 可以是 basic 或 linksys。当 type 是 basic 时的语法是：

```
exten => *723,1,Page(MulticastRTP/basic/239.0.0.1:1234)
```

并不是所有的设备都支持 IP 多播，但是我们已经在 Snom [注9]，Linksys/Cisco，Avaya 等设备上进行了测试，它们都能正常工作[注10]。

在 Cisco SPA 电话机上使用 IP 多播广播

在 Cisco SPA 电话机上实现 IP 多播广播特性的方法有些奇怪，但是一旦我们配置好，它也能很好的工作。窍门在于你在电话机上配置的多播地址并不是 Page()中发送的地址，而更像一个信令 channel。

我们发现的窍门是，你完全可以把这个地址当作多播地址一样只用，但是要使用不同的 IP 端口号。

Dialplan 看起来是这样的：

```
exten => *724,1,Page(MulticastRTP/linksys/239.0.0.1:1234/239.0.0.1:6061)
```

在 SPA 电话机上，你需登录到它的 WEB 管理界面的 SIP 页面。在这一页非常靠下的地方，你会找到一个名为 *Linksys Key System Parameters* 的部分。你需要配置下列参数：

- Linksys Key System: Yes
- Multicast Address: 239.0.0.1:6061

请注意你在这里指定的多播地址是你在 Asterisk 的 MulticastRTP channel 中定义的第二个地址（在我们的例子中，这一个使用端口 6061）。

请注意当你的环境中混合由 SPA 电话机（过去是 Linksys，现在是 Cisco）和其它型号的电话机时，你可以像这个例子中一样书写 **Page()**命令。其它电话机将使用第一个地址，就像你在命令中用 **basic** 替代 **linksys** 一样。

11.2.1.4 VoIP广播适配器

最近，市场上出现了一种基于 VoIP 的广播喇叭。这种设备在 dialplan 中的处理与连接到 UTI1 上的 SIP ATA 完全相同，但是它们可以被与吸顶式喇叭相同的方法安装。由于它们是自动应答的，所以不需要像对于 SIP 话机那样向它们发送额外的信息。

对于小型应用（可能只需要不到六个喇叭），这种设备可能是不划算的。然而，对任何更大的系统（或者安装在复杂环境的系统，例如仓库或停车场）来说，你将在远低于通过模拟接口（FXS）连接到传统电话系统的传统模拟公共广播系统的成本的情况下获得更好的性能。

我们不知道这类设备是否支持多播。如果你计划大量使用这种设备，请记住这一点。

11.2.1.5 组合广播

在许多组织中，可能既需要基于话机的广播系统也需要外部广播系统。作为一个例子，一家工厂可能希望在办公区使用基于话机的广播系统，而在车间和仓库使用吸顶式喇叭的广播系统。从 Asterisk 的观点看，这非常容易实现。当你调用 Page()应用程序时，你可以简单的指定你希望广播的不同资源，并利用 & 字符分隔开，这样它们就将都被包含在 **Page()**应用程

序创建的会议中。

11.2.1.6 混合在一起应用

这时候，你应该有了一张你希望广播的不同 channels 的列表。由于 **Page()**可以同时向多个 channel 发信号，我们建议首先定义一个包含所有 channels 列表的全局变量，然后再调用 **Page()**应用程序：

```
[global]
MULTICAST=MulticastRTP/linksys/239.0.0.1:1234
;MULTICAST=MulticastRTP/linksys/239.0.0.1:1234/239.0.0.1:6061      ; if you have SPA
                                                                   ; (Linksys/Cisco)
                                                                   ; phones
BOGEN=SIP/ATAforPaging  ; This assumes an ATA in your sip.conf file named
                        ; [ATAforPaging]
;BOGEN=DAHDI/25         ; We could do this too, assuming we have an analog
                        ; FXS card at DAHDI channel 25
PAGELIST=${MULTICAST}&${BOGEN}      ; All of these variable names are arbitrary.
                                    ; Asterisk doesn't care what you call these strings
[page_context]  ; You don't need a page context, so long as the extension you
                ; assign to paging is dialable by your sets
exten => *724,1,Page(${PAGELIST},i,120)
```

依赖于不同的硬件，这个例子提供了几种可能配置。虽然并不严格要求必须定义 **PAGELIST** 变量，但我们发现定义该变量有助于简化管理多种广播资源，特别是在配置和测试阶段。

我们为了这个例子创建了一个用于广播的 context。为了让它工作，你或者用 **include** 将这个 context 包含到你的分机进入 dialplan 的 context 中，或者利用 Goto()跳转过来（例如，**Goto(page_context,*724,1)**）。另一个选择是，你可以在每个需要用到广播的 context 中写死一个 extension 用于这个目的。

11.2.2分区广播

分区广播在像汽车卖场这样的地方非常流行，在这种地方，新车销售部门和二手车销售部门可能都需要用到广播系统，但并不需要听到其他部门的广播信息。

在分区广播应用中，当一个人发送广播时需要选择她希望向哪个区广播。一般会采用类似 PAM2000 这样的分区广播控制器来允许发送喜好到不同的分区：Page()应用程序发送信号到分区控制器，分区控制器应答，然后一个额外的数字被发送来选择广播会发送到哪个区。大部分分区控制器允许向所有分区广播，以及向部分分区广播（例如，想新车及二手车销售部门广播）。

你也可以在 dialplan 中使用独立的 extensions 来分隔 ATAs（或者一组电话机），但是这么做

被证明比简单购买一个分区控制器带来更大的复杂性和更高的费用。分区控制器并不需要特别的不同技术，但是它需要有关 dialplan 和硬件的更多思考和计划。

11.3 结论

在本章中，我们研究了 *features.conf* 文件，它包含了很多功能，包括使能基于 DTMF 的呼叫转移，使能通话过程中的录音，为一个或多个公司配置 parking 号码。我们也学习了向办公室广播信息的不同方法，包括传统的吸顶式广播系统和利用办公电话的多播广播系统。研究这些利用现代方法实现传统的 parking 和 paging 功能的方法有助于展示 Asterisk 可以提供的灵活性。

注释：

注1. 是的，我们了解 SIP INFO 消息事实上是 SIP 消息，从技术上说并不是语音通道的一部分。但需要指出的是，在呼叫中你不能通过你的 SIP 电话机上的“transder”或“park”按键来使用这些特性。你必须通过发送 DTMF 来使用这些特性。

注2. 多读两遍，这是有意义的。

注3. 我们希望你了解，实际的 extension（分机）命名与 parked 这个呼叫的 channel 名相关，但不会真的是 SIP_0004F2040808（除非 Leif 把他实验室的 Polycom 电话卖给你了）。

注4. 虽然也可以采用一些灵活的语法（你可以从示例文件中找到例子），但是我们的例子都是采用我们推荐的语法书写的，因为它是最符合典型的 dialplan 语法的。

注5. Bogen UTI1 接口是非常有用的，因为它可以处理各种各样的输入和输出连接器，它几乎能保证我们能毫不费力的将你的电话系统连接到任何外部广播设备上，无论它有多老或多少见。

注6. 在本书中，我们假设外部广播设备已经安装并且是与老式电话系统连接在一起的。

注7. 提示：在这里 local channel 将是你的朋友。

注8. IP 多播有专用的 D 类地址，从 224.0.0.0 到 239.255.255.255（但是请在使用多播地址前仔细研究 IP 多播）。部分多播地址是私有的，部分多播地址是公共的，以及部分多播地址是用来设计实现与你所希望的不同目的的。关于多播地址的更多信息，请参考 http://en.wikipedia.org/wiki/IP_multicast#IP_multicast_addressing_assignments.

注9. 非常大声，并且无法调整增益。

注10.迄今为止，我们可以说 Polycom 设备不支持多播。我们确定无法找到办法来实现。

I