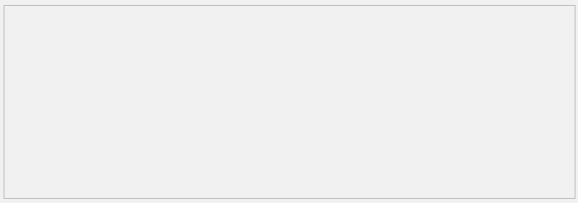
PBX对接R2收号限制导致不通案例

李勇 2006-05-25 发表

PBX对接R2收号限制导致不通案例

1、 组网：



Pbx内线号码：8007，出局后加前缀0351，变成03518007。
Cisco3640带fxs语音模块，电话号码为03551000。
Cisco3640通过以太网e0接我们的r3640e的e0，r3640e通过v1e1接pbx程控交换机，交换机在接一部电话8007。

2、 问题描述：

从pbx的8007打电话到cisco的03551000正常，但是从cisco下接的03551000电话拨号03518007打到pbx 的8007，不通无振铃音。

3、 分析过程：

配置见后面附件。

当cisco下接的电话拨打pbx上电话时，从deb信息来看语音的数据包到了路由器，排除网络不通的情况，而且路由器也处理了语音数据，有h225协议交互信息，而且也发送了CallProceeding消息，但是协议协商不起来，Send ReleaseComplete Msg when receive CCRelease Msg，问题应该出现在3640e和pbx相接的v1e1线路协商上，详情见下面的deb信息。

故障时的debug信息：

R2:channel 1 recv RCV msg R2_OFFHOOK

R2:tk(0:0) channel 1 open MFC

R2:tk(0:0) channel 1 send dl message: sig-val 3, direction 2 src-id 8

R2:Tk(0:0) State --- Occupy TK circuit, wait SetupAck

R2:tk(0:0) channel 1 Received dl msg sig DTE_11

R2:tk(0:0) channel 1 send MFC sig 0x8

R2:Send called number --- 8 *******被叫第一位号码8*******

R2:Tk(0:0) State --- Sending called number

R2:tk(0:0) recv MFC:mode:1, register-signal 1 *****正常时的数值******

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:tk(0:0) channel 1 send MFC sig 0xa

R2:Send called number --- 10 ********被叫第二位号码0******

R2:Tk(0:0) State --- Sending called number

R2:tk(0:0) recv MFC:mode:1, register-signal 6 *****数值改变*****

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:Recieve A6. ********要求送主叫号码*****

R2:tk(0:0) channel 1 send MFC sig 0x1

R2:Tk(0:0) State --- Sending caller number

R2:tk(0:0) recv MFC:mode:1, register-signal 1

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:tk(0:0) channel 1 send MFC sig 0xa *********主叫第一位号码0 *********

R2:Tk(0:0) State --- Sending caller number

R2:tk(0:0) recv MFC:mode:1, register-signal 1

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:tk(0:0) channel 1 send MFC sig 0x3 *******主叫第二位号码3*******

R2:Tk(0:0) State --- Sending caller number

R2:tk(0:0) recv MFC:mode:1, register-signal 1

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:tk(0:0) channel 1 send MFC sig 0x5 ******主叫第三位号码5 *****

R2:Tk(0:0) State --- Sending caller number

R2:tk(0:0) recv MFC:mode:1, register-signal 1

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:tk(0:0) channel 1 send MFC sig 0x5 主叫第四位号码5

R2:Tk(0:0) State --- Sending caller number

R2:tk(0:0) recv MFC:mode:1, register-signal 1

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:tk(0:0) channel 1 send MFC sig 0x1　　主叫第五位号码1

R2:Tk(0:0) State --- Sending caller number

R2:tk(0:0) recv MFC:mode:1, register-signal 1

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:tk(0:0) channel 1 send MFC sig 0xa　　主叫第三六位号码0

R2:Tk(0:0) State --- Sending caller number

R2:tk(0:0) recv MFC:mode:1, register-signal 1

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:tk(0:0) channel 1 send MFC sig 0xa　　主叫第七位号码0

R2:Tk(0:0) State --- Sending caller number

R2:tk(0:0) recv MFC:mode:1, register-signal 1

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:tk(0:0) channel 1 send MFC sig 0xa　　主叫第八位号码0

R2:Tk(0:0) State --- Sending caller number

R2:tk(0:0) recv MFC:mode:1, register-signal 1

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:tk(0:0) channel 1 send MFC sig 0xf　　主叫号码结束标识

R2:Tk(0:0) State --- Sending called number　　****要求送被叫号码****

R2:tk(0:0) recv MFC:mode:1, register-signal 1

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:tk(0:0) channel 1 send MFC sig 0xa　　第三位被叫号码0前面已经送了二位

R2:Send called number --- 10

R2:Tk(0:0) State --- Sending called number

R2:tk(0:0) recv MFC:mode:1, register-signal 1

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:tk(0:0) channel 1 send MFC sig 0x7

R2:Send called number --- 7　　第三位被叫号码7

R2:Tk(0:0) State --- Sending called number　　******* 继续请求送被叫号码*******

R2:tk(0:0) recv MFC:mode:1, register-signal 1

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:tk(0:0) channel 1 send MFC sig 0x0　　但是已经送完，发出号码送完提示

R2:All of called number have been send completely!

R2:Tk(0:0) State --- Sending called number　　但是被叫号码没有达到pbx要求的位数，继续要求送被叫号码，而网关没有任何提示，只能等待超时。

R2:TR_TKO_SEND_CALLED_NUM time out!, pArg=0x0

R2:tk(0:0) channel 1 close MFC　　pbx认为号码接受完毕，关闭接受协议进程。

R2: tk(0:0)TR_TKO_SEND_CALLED_NUM Time out

R2:Tk(0:0) State --- idle

R2: channel 1 send rcv msg: prim-id: R2_RELEASE　　网关等待一定时间没有收到提示，认为目的不可达，发送主动拆链，
callId 36, service type 0
param[0] 3, param[1] 0, param[2] 0, param[3] 0
DNIS: , ANI

R2:tk(0:0) channel 1 send dl message: sig-val 11, direction 2 src-id 8

R2:Tk(0:0) State --- caller onhook firstly, wait called onhook
IPP_225 [65535]: Send ReleaseComplete Msg when receive CCRelease Msg

R2:tk(0:0) channel 1 Received dl msg sig DTE_10

R2: tk(0:0)receive line-signal RELEASE-GUARD

R2:Tk(0:0) State --- idle

R2: channel 1 send rcv msg: prim-id: R2_RELEASE
callId 36, service type 0
param[0] 3, param[1] 0, param[2] 0, param[3] 0
DNIS: , ANI
IPP_225 [65535]: Receive ReleaseComplete Msg crv=32774
reason=128+255,FacilityLen=0
[yuyin_router]
[yuyin_router]

由以上信息可以断定pbx接受被叫号码的位数是不是有限制？通过下面的debug信息就可以清楚的解释了。

将3640e上的语音配置match-template 0351。。。。 改成match-template 。。。。。。。。。debug信息如下：

注意这次的主叫号码变为12345678。

R2:channel 1 recv RCV msg R2_OFFHOOK

R2:tk(0:0) channel 1 open MFC

R2:tk(0:0) channel 1 send dl message: sig-val 3, direction 2 src-id 8

R2:Tk(0:0) State --- Occupy TK circuit, wait SetupAck

R2:tk(0:0) channel 1 Received dl msg sig DTE_11

R2:tk(0:0) channel 1 send MFC sig 0xa

R2:Send called number --- 10

R2:Tk(0:0) State --- Sending called number

R2:tk(0:0) recv MFC:mode:1, register-signal 1

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:tk(0:0) channel 1 send MFC sig 0x3

R2:Send called number --- 3

R2:Tk(0:0) State --- Sending called number

R2:tk(0:0) recv MFC:mode:1, register-signal 6

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:Recieve A6.

R2:tk(0:0) channel 1 send MFC sig 0x1

R2:Tk(0:0) State --- Sending caller number

R2:tk(0:0) recv MFC:mode:1, register-signal 1

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:tk(0:0) channel 1 send MFC sig 0x1

R2:Tk(0:0) State --- Sending caller number

R2:tk(0:0) recv MFC:mode:1, register-signal 1

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:tk(0:0) channel 1 send MFC sig 0x2

R2:Tk(0:0) State --- Sending caller number

R2:tk(0:0) recv MFC:mode:1, register-signal 1

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:tk(0:0) channel 1 send MFC sig 0x3

R2:Tk(0:0) State --- Sending caller number

R2:tk(0:0) recv MFC:mode:1, register-signal 1

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:tk(0:0) channel 1 send MFC sig 0x4

R2:Tk(0:0) State --- Sending caller number

R2:tk(0:0) recv MFC:mode:1, register-signal 1

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:tk(0:0) channel 1 send MFC sig 0x5

R2:Tk(0:0) State --- Sending caller number

R2:tk(0:0) recv MFC:mode:1, register-signal 1

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:tk(0:0) channel 1 send MFC sig 0x6

R2:Tk(0:0) State --- Sending caller number

R2:tk(0:0) recv MFC:mode:1, register-signal 1

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:tk(0:0) channel 1 send MFC sig 0x7

R2:Tk(0:0) State --- Sending caller number

R2:tk(0:0) recv MFC:mode:1, register-signal 1

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:tk(0:0) channel 1 send MFC sig 0x8

R2:Tk(0:0) State --- Sending caller number

R2:tk(0:0) recv MFC:mode:1, register-signal 1

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:tk(0:0) channel 1 send MFC sig 0xf

R2:Tk(0:0) State --- Sending called number

R2:tk(0:0) recv MFC:mode:1, register-signal 1

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:tk(0:0) channel 1 send MFC sig 0x5

R2:Send called number --- 5　　接受的被叫第三位号码5

R2:Tk(0:0) State --- Sending called number

R2:tk(0:0) recv MFC:mode:1, register-signal 1

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:tk(0:0) channel 1 send MFC sig 0x1

R2:Send called number --- 1　　接受的被叫第四位号码1

R2:Tk(0:0) State --- Sending called number

R2:tk(0:0) recv MFC:mode:1, register-signal 1

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:tk(0:0) channel 1 send MFC sig 0x8

R2:Send called number --- 8　　接受的被叫第五位号码8

R2:Tk(0:0) State --- Sending called number

R2:tk(0:0) recv MFC:mode:1, register-signal 1

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:tk(0:0) channel 1 send MFC sig 0xa　　接受的被叫第六位号码0

R2:Send called number --- 10

R2:Tk(0:0) State --- Sending called number

R2:tk(0:0) recv MFC:mode:1, register-signal 1

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:tk(0:0) channel 1 send MFC sig 0xa

R2:Send called number --- 10　　接受的被叫第七位号码0

R2:Tk(0:0) State --- Sending called number

R2:tk(0:0) recv MFC:mode:1, register-signal 3　此寄存器数值改变

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:Recieve A3.　　认为号码接受完毕。此时还有第八位7没有接受。交换机发出号码收够信息，转至B组信号

R2:tk(0:0) channel 1 send MFC sig 0x3

R2:Send KD=3.　　路由器发出主叫用户类别

R2:Tk(0:0) State --- wait KB　　KB信号是表示被叫用户状态的信号，起证实KD信号和控制接续的作用

R2:tk(0:0) recv MFC:mode:1, register-signal 5

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:Recieve KB=5.　　交换机通知被叫用户忙，这是因为接受的号码与原来配置的号码匹配不上。

R2: tk(0:0)The called was busy(KB = 5)

R2:Tk(0:0) State --- idle

R2:tk(0:0) channel 1 close MFC

R2: channel 1 send rcv msg: prim-id: R2_RELEASE 释放连路
callId 50, service type 0
param[0] 17, param[1] 0, param[2] 0, param[3] 0
DNIS: , ANI

R2:tk(0:0) channel 1 send dl message: sig-val 11, direction 2 src-id 8

Voice_CallRecord_Will_Full
IPP_225 [65535]: Send ReleaseComplete Msg when receive CCRelease Msg

R2:tk(0:0) channel 1 Received dl msg sig DTE_10

R2:call in tk(0:0) was terminated 2
IPP_225 [65535]: Receive ReleaseComplete Msg crv=32786 reason=128+255,FacilityLen=0

将pbx交换机的中继参数修改成接受四位号码，而不是原来的7位号码，接通正常，debug信息如下：

前面相同信息略去：

R2:Tk(0:0) State --- Sending called number

R2:tk(0:0) recv MFC:mode:1, register-signal 1

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:tk(0:0) channel 1 send MFC sig 0xa

R2:Send called number --- 10　　接受第三位被叫号码0

R2:Tk(0:0) State --- Sending called number

R2:tk(0:0) recv MFC:mode:1, register-signal 1

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:tk(0:0) channel 1 send MFC sig 0x7

R2:Send called number --- 7　　接受第四位被叫号码7

R2:Tk(0:0) State --- Sending called number

R2:tk(0:0) recv MFC:mode:1, register-signal 3　寄存器数值改变，号码接受完毕。

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:Recieve A3.

R2:tk(0:0) channel 1 send MFC sig 0x3

R2:Send KD=3.

R2:Tk(0:0) State --- wait KB

R2:tk(0:0) recv MFC:mode:1, register-signal 1

R2:tk(0:0) channel 1 send MFC sig 0x0

R2:tk(0:0) recv MFC:mode:1, register-signal 0

R2:Recieve KB=1.　　检测到被叫用户空闲

R2: channel 2 send rcv msg: prim-id: R2_ALERT 向pbx发送振铃信息。
callId 65, service type 0
param[0] 1, param[1] 31, param[2] 0, param[3] 0
DNIS: , ANI

R2:tk(0:0) channel 1 close MFC

R2:Tk(0:0) State --- called alerting　向对端发送振铃音
IPP_225 [1]: Send Alerting Msg when receive CCAlerting Msg
IPP_245 [1]: Send H245 TCS Req Msg when accept a 245 socket
IPP_245 [1]: Receive H245 TCS Request Msg, Code=10
IPP_245 [1]: Send H245 TCS Ack Msg when receive TCS Req Msg
IPP_245 [1]: Receive H245 MSD Request Msg, Type=60, Number=8778
IPP_245 [1]: Terminal Type is big, idle Send MSD Master Rsp
IPP_245 [1]: Send H245 OLC request when receive MSD request, Code=10
IPP_245 [1]: Receive H245 TCS Ack Msg
IPP_245 [1]: Receive H245 MSD Ack Msg, Type=slave
IPP_245 [1]: Receive H245 OLC Request Msg, Code=10
IPP_245 [1]: Send H245 OLC Ack Msg when receive OLC Req Msg
IPP_245 [1]: Receive H245 OLC Ack Msg, dst0=19332@10.16.0.254, dst1=19333@10.16.0.254

R2:tk(0:0) channel 1 Received dl msg sig DTE_01

R2: channel 1 send rcv msg: prim-id: R2_ACTIVE
callId 65, service type 0
param[0] 2, param[1] 0, param[2] 0, param[3] 0

DNIS: 8007, ANI 12345678

R2:Tk(0:0) State --- wait RCV active ackownledge

总 结

由此可以看出，当pbx走e1语音中继线时，可以通过设置限制接受多少位电话号码，在我们做类似调试的时候注意一下就可以了，不行通过deb voic r2 all 可以清晰看到相关信息。另外这次调试搞到一份pbx程控交换机的说明书，发过来大家可以借鉴一下。