

Cisco Voice Gateways

PacNOG7 VoIP Workshop
PagoPago, June 2010

Jonny Martin - jonny@jonnynet.net

Voice Gateways

- Any device with one or more TDM PSTN interfaces on them
 - TDM - Time Division Multiplexing (i.e. traditional telephony)
 - PSTN - Public Switched Telephone Network
 - To be really useful, gateways also need an IP interface on them
- Many vendors, we'll concentrate on Cisco IOS based voice gateways
- Both analog and digital interfaces, we'll look at the more common ones

Interface Types - Digital

- ISDN primary rate circuits (there are others, but we will look at ISDN)
- E1 (primarily used in Europe and Oceania)
 - 2 Mbit/s bearer
 - 32x 64kbit/s channels. 30 for voice, 1 for signalling (timeslot 16), 1 framing
- T1 (primarily used in North America)
 - 1.5 Mbit/s bearer
 - 24x 64kbit/s channels. 23 for voice, 1 for signalling (timeslot 24)
- Common interfaces for ISP dial-in, PBX to carrier trunks, etc.

Interface Types - Digital

- Basic Rate ISDN
 - 144kbit/s bearer
 - 2x 64kbit/s channels + 1x 16kbit/s signalling channel
- 2B + D
 - B channels = 64kbit/s voice/data channels
 - D channel(s) = signalling data channels

Interface Types - Analog

- Only really two types:
- FXO interface - plugs into your telco (Foreign eXchange central Office)
 - uses FXS signalling!
- FXS interface - plugs into a telephone. e.g. ATAs (Foreign eXchange Station)
 - uses FXO signalling!
- Uses analog signalling, limited to one DDI per line
- Signalling is generally more ambiguous and harder to work with than digital signalling

AS5300 / AS5350 / AS 5400

- Multi-port E1/T1 access servers
- Popular ISP dial-in boxes
- 5300 - can be used for VoIP when loaded with DSP cards
- 5350/5400 has universal ports - modem or VoIP
- Dial-up ISPs often well placed to provide VoIP services
 - POPs in many locations, with the right hardware!



IOS Voice Configuration

- For VoIP we need to configure:
 - voice-port - the voice 'interface'
 - FXS / FXO - e.g. voice-port 1/0/0
 - E1/T1 signalling channel - e.g. voice-port 1/0:D
 - dial-peer - tells the gateway how to connect voice ports to VoIP call legs
- For E1/T1 links we also need to configure the physical bearer
 - controller E1 / controller T1
 - interface serial 0:15 (the signalling timeslot for an E1, 0:23 for T1)

E1 Configuration

```
! This configuration works with Telecom NZ E1 circuits
!
isdn switch-type primary-net5
!
controller E1 0
  clock source line primary
  pri-group timeslots 1-10,16      ! note, timeslots count from 1.
  description Link to Telecom
!
!
interface Serial0:15              ! note, serial channels count from 0.
  no ip address
  isdn switch-type primary-net5
  isdn incoming-voice modem       ! treats incoming calls as modem or voice
!                                  ! rather than data
!
voice-port 0:D
  echo-cancel coverage 64
  cptone NZ                        ! returns NZ progress tones
  bearer-cap Speech
!
```


T1 Configuration

```
!  
isdn switch-type primary-ni  
!  
!  
controller T1 1/0  
  framing esf  
  linecode b8zs  
  pri-group timeslots 1-24  
!  
!  
interface Serial1/0:23  
  no ip address  
  encapsulation hdlc  
  isdn switch-type primary-ni  
  isdn incoming-voice modem  
!  
!  
voice-port 1/0:D  
  echo-cancel coverage 64  
  ! default cptone is US  
!
```

FXS / FXO Configuration

```
! Some useful settings
!
voice-port 1/0/0
  no comfort-noise           ! needs 'no vad' on VoIP dial-peer
  cptone NZ
  timeouts interdigit 3     ! timeout when gathering dialled digits
  description Analog phone line
!

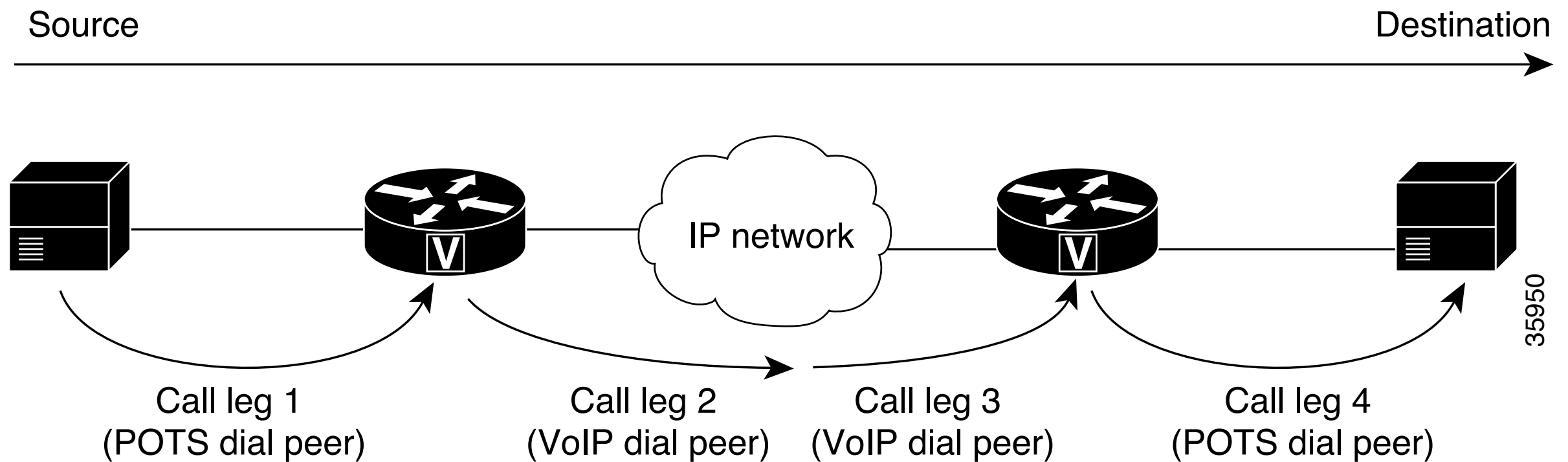
! Or, if you're just having a play, the defaults will work:
!
voice-port 1/0/1
!
```

Dial Peers

- Basic building block on Cisco voice gateways, the dial-peer
- All calls consists of at least two call legs:
 - Originating device to originating gateway (POTS)
 - Originating gateway to IP network (VoIP)
 - ...and/or
 - IP network to destination gateway
 - Destination gateway to destination device

Dial Peers ...ctd

- Most hardware will also allow TDM switching, i.e. POTS to POTS
 - But not typically VoIP media proxying (i.e. no VoIP-VoIP)



Dial Peer Syntax

```
! POTS dial peer
!  
dial-peer voice tag pots  
  destination-pattern number  
  port voiceport#  
  other configurable options  
!
```

```
! VoIP dial peer  
!  
dial-peer voice tag voip  
  destination-pattern number  
  session target data address  
  other configurable options  
!
```

```
! Destination pattern = E.164 number (i.e. a telephone number)
```

Dial Peer Matching

- When a call is made, IOS will select the appropriate dial-peer for an outbound leg depending on call direction
 - voip --> pots
 - pots --> voip
- Longest match for *destination-pattern* is chosen
- If multiple longest matches exist, the dial-peer with the lowest *preference* will be chosen

Example POTS Dial Peers

```
! Outbound send-everything-to-the-pstn POTS dial-peer:
!
dial-peer voice 1 pots
  destination-pattern T           ! T = digit timeout, i.e. any string of digits
  direct-inward-dial             ! allow incoming calls from the POTS port also
  port 0:D
!

! Only send numbers prefixed with 021 out the POTS port:
!
dial-peer voice 1 pots
  destination-pattern 021T       ! T = digit timeout, i.e. any string of digits
  direct-inward-dial
  port 1:D
!

! Only send seven digit numbers prefixed by 04
!
dial-peer voice 1 pots
  destination-pattern 04.....    ! . = a single digit
  direct-inward-dial
  port 2:D
!
```


Failover Routing

- Failover routing is achieved by ‘*hunting*’ on busy, no answer, and a myriad of other causes
- Works for both *pots* and *voip* dial-peers
- Use *preference* to step through dial-peers
 - 0 is best and the default, 9 is worst
- Use *huntstop* on the ‘last’ dial-peer
- Often used in conjunction with *translation-patterns* to ensure correct dial string for different trunks

Failover Example

```
! Incoming POTS calls first try one VoIP server, then failover to another  
! if that server doesn't answer or is busy
```

```
!  
voice hunt user-busy  
voice hunt no-answer  
!  
dial-peer voice 49896411 voip  
  destination-pattern 4989641  
  session protocol sipv2  
  session target ipv4:a.b.c.1  
  dtmf-relay rtp-nte  
  codec g711ulaw  
!  
dial-peer voice 49896412 voip  
  huntstop  
  preference 1  
  destination-pattern 4989641  
  session protocol sipv2  
  session target ipv4:a.b.c.2  
  dtmf-relay rtp-nte  
  codec g711ulaw  
!
```

Translation Patterns

- Used to translate called and calling numbers
- Uses basic translation rules to prepend / strip digits, translate one number into a completely different number
- Some basic examples...

Translation Pattern Examples

```
! strip 644 from the start of the number for numbers starting 6442 - 6449
```

```
!
```

```
translation-rule 100
```

```
Rule 2 ^6442..... 2
```

```
Rule 3 ^6443..... 3
```

```
Rule 4 ^6444..... 4
```

```
Rule 5 ^6445..... 5
```

```
Rule 6 ^6446..... 6
```

```
Rule 7 ^6447..... 7
```

```
Rule 8 ^6448..... 8
```

```
Rule 9 ^6449..... 9
```

```
!
```

```
! Prefix 04 to the beginning of any number
```

```
!
```

```
translation-rule 101
```

```
Rule 1 ^.% 04
```

Translation Pattern Examples ...ctd

```
! translate any number to 0212304323
!
translation-rule 120
  Rule 1 any 0212304323

! Normalise numbers into a standard format
!
translation-rule 150
  Rule 1 ^644498..... 498      ! 6444981234  --> 4981234
  Rule 2 ^04498..... 498      !  044981234  --> 4981234
  Rule 3 ^00644498..... 498   ! 006444981234 --> 4981234
!
```

Apply the Translation Pattern

```
!  
dial-peer voice 44989560 voip  
  destination-pattern 4989560  
  translate-outgoing calling 100           ! translated the CALLING number  
  translate-outgoing called 200          ! translate the CALLED number  
  session protocol sipv2  
  session target ipv4:203.114.148.130  
  dtmf-relay rtp-nte  
  codec g711ulaw  
  no vad  
!
```