# Simple Least Cost Routing Using TekRADIUS

This document explains how to provide simple least cost routing for small Internet Telephony Service Providers *(ITSP)* using TekRADIUS. In this example following components are used:

- **IP-to-IP Gateway.** This is a Cisco gateway device with IP-to-IP gateway software installed. It can be a Cisco 3660 or Cisco 3845 model.
- **TekRADIUS**

You can see how simple least cost routing components organized on the office LAN in the diagram below:

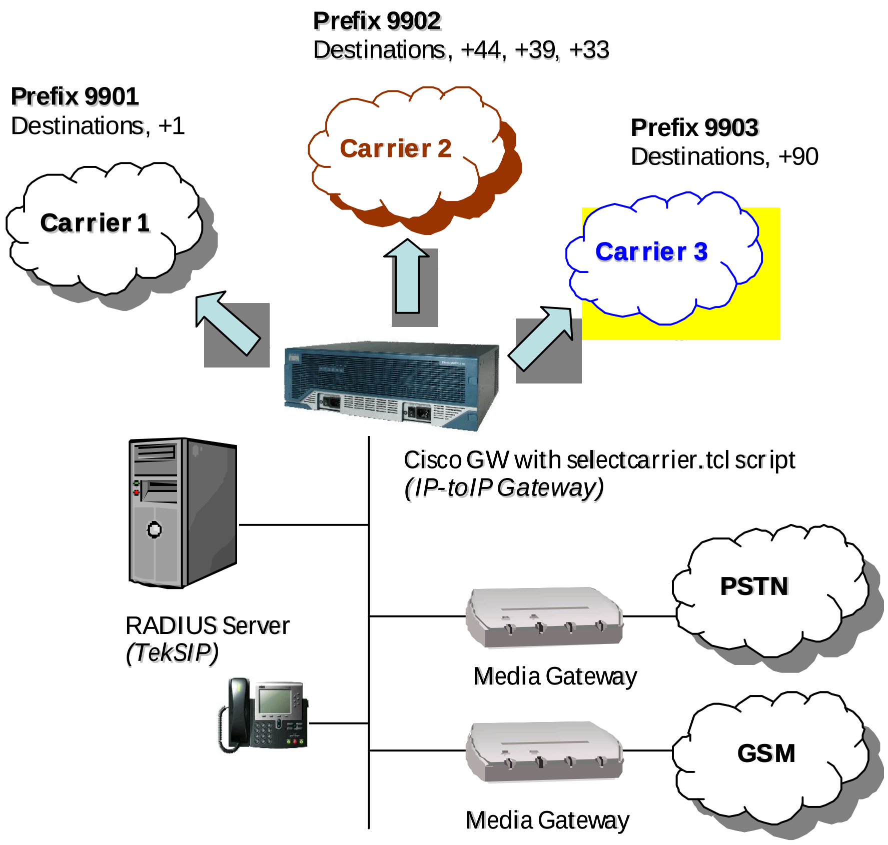


**Figure - 1.** Sample Topology

You can process VoIP calls *(H.323 or SIP)* using Cisco TCL scripting. You can authenticate and authorize incoming VoIP calls using RADIUS. It's also possible to change call parameters; you can add a prefix to incoming dialed digits *(Called-Station-Id)* for example.

Suggested work model based on authorizing incoming calls using a RADIUS server. You can return a carrier prefix form RADIUS server to Cisco Gateway if a matching Carrier found for the dialed number. After adding Carrier prefix to dialed number, Cisco Gateway can route the call to a outgoing dial-peer configured for the selected carrier.

Although you can use any RADIUS server which permits manipulation of RADIUS reply attribute, we'll show how you can accomplish simple least cost routing using TekRADIUS.

## TekRADIUS Configuration

You'll need to add two tables to TekRADIUS database manually *(TekRADIUS supports only Microsoft SQL Server)*:

Carriers table:

```
CREATE TABLE [dbo].[Carriers](
      [Carrier] [nvarchar](50) NULL,
      [Desc] [nvarchar](50) NULL
)
```

Destinations table:

```
CREATE TABLE [dbo].[Destinations](
      [Prefix] [nvarchar](50) NULL,
      [Destination] [nvarchar](255) NULL,
      [Type] [char](10) NULL,
      [Carrier] [char](10) NULL
)
```

Carriers table is used to store carrier prefixes and Destination table contains which destination is routed to which carrier. 'Desc' field in Carriers table and 'Destination' and 'Type' fields are informational fields and optional. You will need also a special function to query these tables to select carrier for a particular dialed number:

```
CREATE function [dbo].[select_carrier](@dnis nvarchar(50))
returns table
as
return select top 1 'cisco|100' as Attribute, 'carrier:' + carrier as [Value]
from destinations where prefix=substring(@dnis,1, len(prefix)) order by
len(prefix) desc
```

After creating necessary tables and the function you must enter test data to tables. In our example following data assumed entered.

*Carriers;*

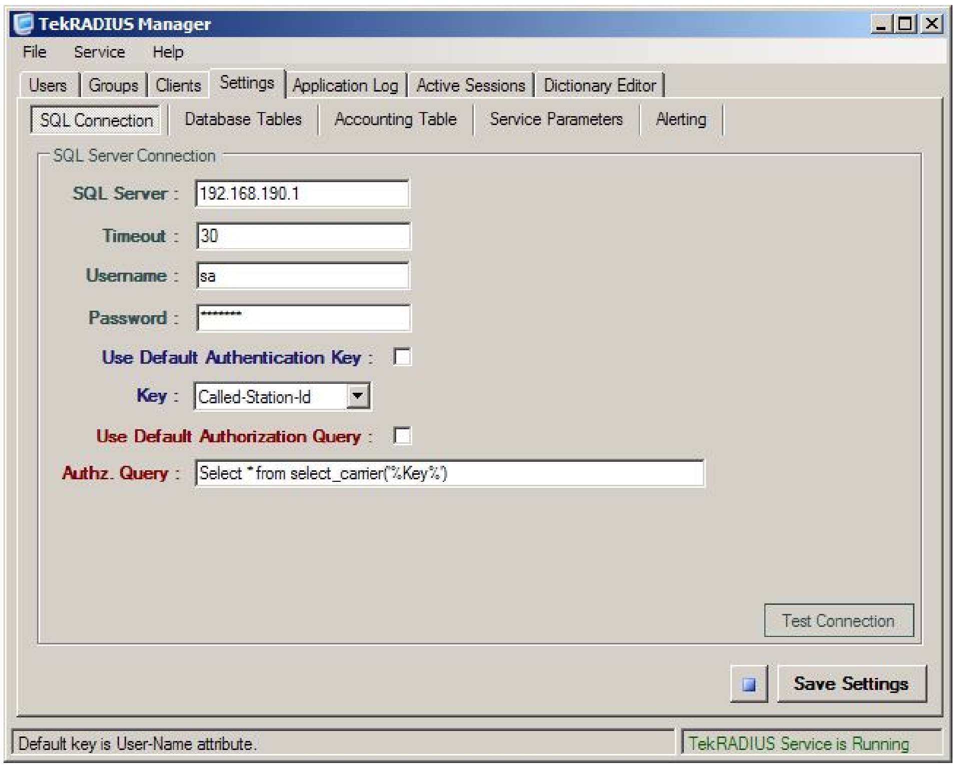
| Carrier | Desc |
|---|---|
| 9901 | Protel |
| 9902 | AMS |
| 9903 | MCI |

*Destinations;*

| Prefix | Destination | Type | Carrier |
|---|---|---|---|
| 1 | A.B.D. | FIX | 9901 |
| 44 | U.K. | FIX | 9902 |
| 33 | Italy | FIX | 9902 |
| 39 | France | FIX | 9902 |
| 90 | Turkey | FIX | 9903 |

Next step for configuring TekRADIUS is changing authorization query. Uncheck 'Use Default Authentication Key' and select 'Called-Station-Id' then 'Use Default Authorization Query' and enter new 'Authz. Query' in Settings/SQL Connection tab:

```
Select * from select_carrier('%Key%')
```

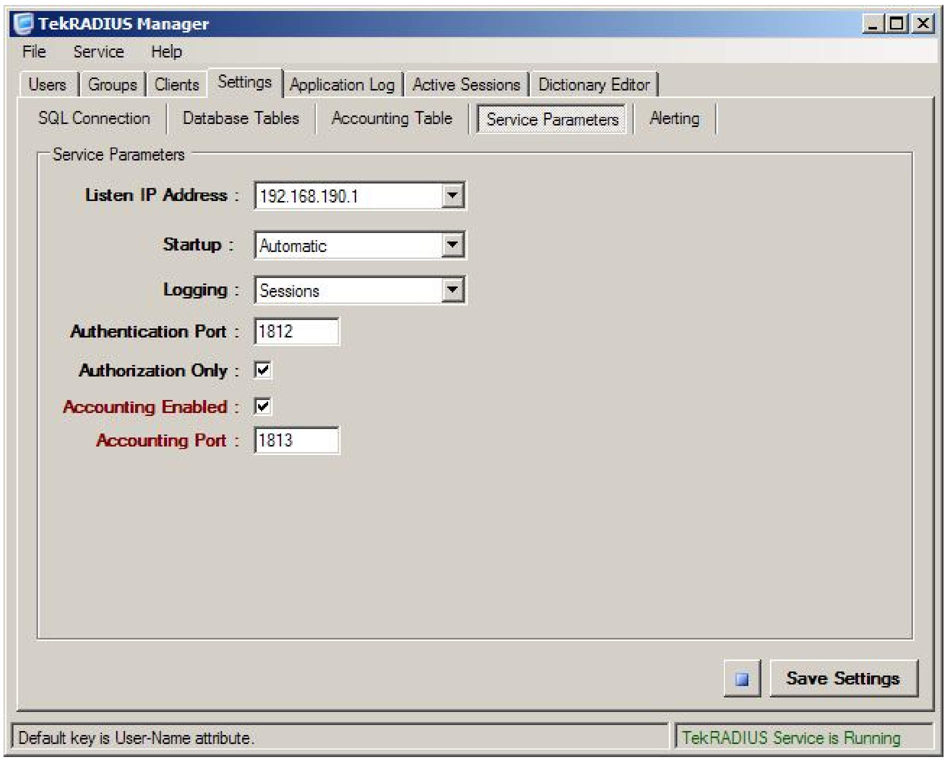

TekRADIUS Manager
File Service Help
Users Groups Clients Settings Application Log Active Sessions Dictionary Editor
SQL Connection Database Tables Accounting Table Service Parameters Alerting
SQL Server Connection
Test Connection
Save Settings

SQL Server : 192.168.190.1
Timeout : 30
Username : sa
Password : *******
Use Default Authentication Key : ☐
Key : Called-Station-Id
Use Default Authorization Query : ☐
Authz. Query : Select * from select_carrier('%Key%')

Default key is User-Name attribute.
TekRADIUS Service is Running

**Figure - 2.** TekRADIUS Settings/SQL Connection tab

Final step is configuring TekRADIUS to run in authorization only mode. Check 'Authorization Only' option in Settings/Service Parameters tab:


TekRADIUS Manager
File Service Help
Users Groups Clients Settings Application Log Active Sessions Dictionary Editor
SQL Connection Database Tables Accounting Table Service Parameters Alerting
Service Parameters
Save Settings

Listen IP Address : 192.168.190.1
Startup : Automatic
Logging : Sessions
Authentication Port : 1812
Authorization Only : ✓
Accounting Enabled : ✓
Accounting Port : 1813

Default key is User-Name attribute.
TekRADIUS Service is Running

**Figure - 3.** TekRADIUS Settings/Service Parameters tab

Do not forget to define you Cisco gateway as a RADIUS client in Clients tab. Re-start TekRADIUS after all settings done and saved.

**Cisco Gateway Configuration**

You need to have IP-to-IP Gateway software installed on Cisco Gateway. You need enter following configuration to communicate with RADIUS server:

```
aaa new-model
!
aaa authentication login default line
aaa authentication login h323 group radius
aaa authorization exec h323 group radius
! Add following line if you like to also get accounting data for VoIP sessions
aaa accounting connection h323 start-stop group radius
!
radius-server host 192.168.190.1 auth-port 1812 acct-port 1813 key secret
```

Configure outgoing dial peers for outgoing carriers. SIP version 2 is used as VoIP protocol in our example:

```
dial-peer voice 50 voip
 description –- Carrier 1 - out --
 destination-pattern 9901T
 session protocol sipv2
 session target ipv4:10.10.1.1
 codec transparent
!
dial-peer voice 60 voip
 description –- Carrier 2 - out --
 destination-pattern 9902T
 session protocol sipv2
 session target ipv4:10.10.2.1
 codec transparent
!
dial-peer voice 70 voip
 description –- Carrier 3 - out --
 destination-pattern 9903T
 session protocol sipv2
 session target ipv4:10.10.30.1
 codec transparent
```

Following TCL script is used for authorizing incoming VoIP calls *(call_route_app.1.1.tcl)*:

```
# Script Locked by: Yasin KAPLAN
# Script Version: 1.1.0.0
# Script Name: call_route_app
#------------------------------------------------------------------
# Dec 13th 2007, Yasin KAPLAN
#
# Copyright (c) 2007 by Yasin KAPLAN
# All rights reserved.
#------------------------------------------------------------------
#
# Description:
#
# Call routing for VoIP session using RADIUS
#
#------------------------------------------------------------------
#

proc init_perCallVars { } {
    global disconnect_cause

    set disconnect_cause 0
}

proc act_Setup { } {
    global account
    global pin
    global destination
    global i_destination

    init_perCallVars

    leg setupack leg_incoming

    set destination [infotag get leg_dnis]
    set ani  [infotag get leg_ani]

    set account [infotag get leg_username leg_incoming]
    set pin [infotag get leg_password leg_incoming]
    set i_destination [string trimleft $destination 0]
    aaa authorize $account $pin $ani $i_destination leg_incoming
}

proc act_Authorized { } {
    global account
    global destination
    global creditTime
    global carrier
    global disconnect_cause
    global f_destination
    global ivrIn

    set callInfo(accountNum) $account

    set status [infotag get evt_status]
    puts "\n aaa authorize Status=$status"

     if { $status == "ao_000" } {
        set ivrIn [infotag get aaa_avpair h323-ivr-in]
        set callInfo(accountNum) $account
        set prefix [split $ivrIn ":"]
        set f_destination "[lindex $prefix 1]$destination"
        leg setup $f_destination callInfo leg_incoming
     } else {
        # You can add a default prefix in place of disconnecting the call...
        set disconnect_cause di_021
        act_SendCauseCode
        fsm setstate CALLDISCONNECT
     }
}

proc act_CallSetupDone { } {
    global creditTime
    global disconnect_cause

    set status [infotag get evt_status]

    puts "\t\t******* act_CallSetupDone: $status"

    if {$status == "ls_000"} {
       return
    } else {
      set disconnect_cause [infotag get evt_last_disconnect_cause]
      act_SendCauseCode
      fsm setstate CALLDISCONNECT
    }
}

proc act_SendCauseCode { } {
    global disconnect_cause
    puts "\t\t***** DISCONNECT CAUSE CODE: $disconnect_cause"
    set cause [split $disconnect_cause "_"]
    leg disconnect leg_incoming [lindex $cause 1]
}

proc act_ConnectionDestroy { } {
    global disconnect_cause

    set disconnect_cause [infotag get evt_last_disconnect_cause]
    connection destroy con_all
}

proc act_Cleanup { } {
    call close
}

requiredversion 2.0

#----------------------------------
#   State Machine
#----------------------------------

  set fsm(any_state,ev_disconnected)           "act_Cleanup           same_state"
  set fsm(CALL_INIT,ev_setup_indication)       "act_Setup             AUTHORIZE"
  set fsm(AUTHORIZE,ev_authorize_done)         "act_Authorized        PLACECALL"
  set fsm(PLACECALL,ev_setup_done)             "act_CallSetupDone     CALLACTIVE"
  set fsm(CALLACTIVE,ev_leg_timer)             "act_ConnectionDestroy CONNDESTROY"
  set fsm(CALLACTIVE,ev_disconnected)          "act_ConnectionDestroy CONNDESTROY"
  set fsm(CONNDESTROY,ev_destroy_done)         "act_SendCauseCode     CALLDISCONNECT"
  set fsm(CALLDISCONNECT,ev_disconnected)      "act_Cleanup           same_state"
  set fsm(CALLDISCONNECT,ev_destroy_done)      "act_Cleanup           same_state"
  set fsm(CALLDISCONNECT,ev_disconnect_done)   "act_Cleanup           same_state"
  set fsm(CALLDISCONNECT,ev_leg_timer)         "act_Cleanup           same_state"

  fsm define fsm CALL_INIT
```

Enter following configuration to define this script as an application:

```
application
 service carrier_routing flash:call_route_app.1.1.tcl
```

And finally add a dial peer for incoming VoIP calls:

```
dial-peer voice 900 voip
 description -- Carrier Routing --
 service carrier_routing
 session protocol sipv2
```

## References

1. Cisco IP-to-IP Gateway Configuration
   http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/vvfax_c/callc_c/h323_c/ipipgw/ipgw.htm
2. Cisco TCL Scripting
   http://www.cisco.com/en/US/products/sw/iosswrel/ps5207/products_feature_guide09186a00801a75a7.html
3. TekRADIUS
   http://www.tekradius.com/
4. Microsoft SQL Server
   http://www.microsoft.com/sql/default.mspx