



slcr

Overview

SoftCP's least cost routing application, **slcr**, is an application server that uses either SUA (rfc3868) or SIP (rfc3261) to provide telecommunications routing information to querying nodes. When using SUA, the slcr process will register with a signaling gateway process (SGP) that runs on a signaling gateway (SG) host server. The SGP is also communicating with a telecommunications switch. The slcr process may also communicate directly with a switch supporting SUA. When communicating using SIP, the slcr application responds to routing requests, but does not have any handshaking capability with the querying node.



Figure 1: SIP Network Diagram

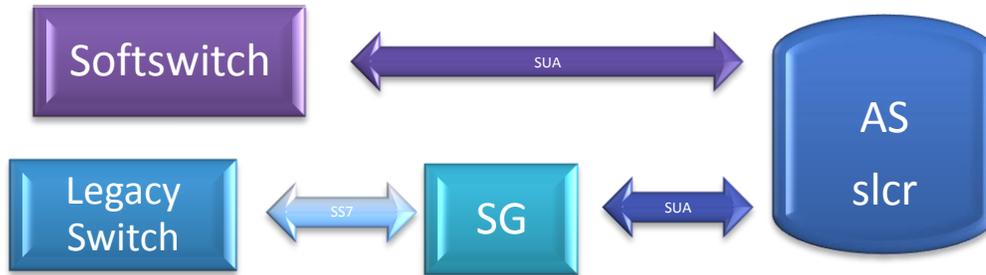


Figure 2: SUA Network Diagram

Running SLCR

The server should be equipped such that **slcr** will start upon server boot. In the event that manual intervention is required, the command usage and flag descriptions are identified here. The default directory for **slcr** and its companion management interface **slcrmgr** is `/opt/slcr/bin`.

Usage

```
>slcr [-c configuration_file] [-l license_file] [-v]
```

Flags

Flag	Description
-c	The configuration file used to start slcr. Default is slnpConfig.xml in directory /etc/opt/slcr
-l	License key file. Default is licenseKey.xml in directory /etc/opt/slcr.
-v	Prints version information for slcr.

Configuration Files

The **slcr** application reads its configuration from several XML encoded files. The main configuration file contains parameters that are globally applied to the **slcr** application. Items such as log file and CDR file locations, additional configuration file locations and global feature configuration settings are also present in this file. A complete description of all available parameters in the main configuration file appears in Appendix A.

Initialization

The slcr process first loads its main configuration file. In the main configuration file the base directory parameter (<main><baseDir>) identifies the location of the remaining configuration files. The table below shows the order that the remaining configuration files are loaded and describes the information present in each configuration file.

Parameter	Description	Mandatory/Optional
tgCarFile	Contains customer, trunk group and carrier information	Mandatory
qualityFile	Contains quality values per carrier and destination	Optional
tiersDir	Directory containing routing data files	Mandatory
areaFile	File containing information mapping destination codes to areas	Optional
lcaFile	Local calling area file mapping destination codes to local calling areas	Optional
clustersFile	Local calling area file defining locality relationships between local calling areas	Optional

The base directory parameter is used in addition to each of the file or directory parameters above unless an absolute path is specified within the parameter itself.

Application Ready State

After all of the data files have loaded. The SLCR application is ready to begin accepting queries for routing information. The application will generate an informational alarm once this condition has been reached. Also, the slcrmgr application will not connect until this application ready state has been reached.

Basic Call Flow

Overview

When a query request for least cost routing information arrives, the input data is collected from signaling. The trunk group number, called number and calling number are extracted and used to find the call jurisdiction which yields the routing tier.

Once the routing tier has been determined, the called number is used to find the preliminary list of carriers. This initial carrier list then undergoes a filtration process where any skips that have been provisioned against the incoming trunk group or its customer entity are applied. A quality evaluation follows to determine whether each carrier meets minimum standards suitable for the customer.

After the filtration process has occurred, the final list of carrier codes remains. The SLCR process collects the switch Id (SWID) and trunk group Id (TGID) values assigned to each carrier and returns them to the querying node.

Figure 3 depicts this high-level logic for the complete SLCR call flow.

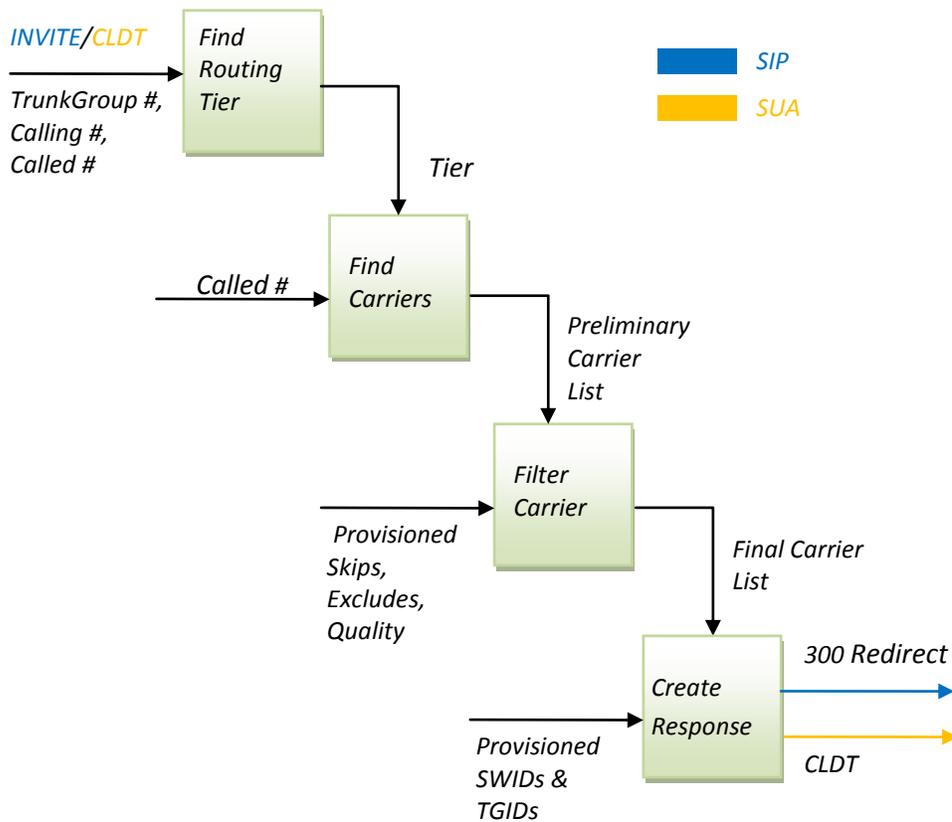


Figure 3: High-Level SLCR Logic Flow

Input Data Collection

SLCR supports both SIP and SUA interfaces to allow communication with many of today's telecommunications network nodes. Once the input data has been collected the logic flow through SLCR remains the same regardless of the interface type. This section describes the message formats expected for each interface type.

SIP

When using SIP, the application expects an incoming INVITE. The INVITE message is searched for an optional trunk group number. When present the **trunk group number** must be in front of the **called number**

in the RequestURI header. A pound character (#) should delimit the trunk group number and called number. A **local routing number** (LRN) may also be present within the RequestURI option parameters. When present, the LRN becomes part of the input data and is later used to find the preliminary carrier list.

The called number is obtained from the RequestURI header as well, while the **calling number** is extracted from the 'From' header. Both the calling and called numbers should be in E.164 format.

Example SIP INVITE message:

```
INVITE sip:40000001#13034241234;npdi;rn=3032330000@10.10.10.10:5070 SIP/2.0
Via: SIP/2.0/UDP 10.10.10.9:5060;branch=z9hG4bK-12913-1-0
From: sipp <sip:12146987300@10.10.10.9:5060>;tag=1
To: <sip:40000001#13034241234@10.10.10.10:5070>
Call-ID: 1-12913@10.10.10.9
CSeq: 1 INVITE
Contact: sip:12146987300@10.10.10.9:5060
Max-Forwards: 70
Subject: Performance Test
Content-Type: application/sdp
Content-Length: 133
(3)
v=0
o=user1 53655765 2353687637 IN IP4 10.10.10.9
s=-
c=IN IP4 10.32.202.54
t=0 0
m=audio 6000 RTP/AVP 0
a=rtpmap:0 PCMU/8000
```

SUA

When SUA is used as the communication protocol and the application has successfully registered with a signaling gateway, the slcr process will respond to connectionless data (CLDT) messages. SLCR expects the Data parameter within the CLDT message to contain a TCAP message. The TCAP message should contain an AINO.2 query message with trigger type 'InfoAnalyzed'. The 'InfoAnalyzed' message must contain a called number within the 'CalledPartyId' parameter. The trunk group identifier should be present within the userId parameter. A calling number may be present in a CallingPartyId parameter. Refer to Telcordia's GR-1299-CORE Switch – Service Control Point (SCP)/Adjunct Interface for additional encoding information.

Find Routing Tier

A routing tier is a concept within SLCR that creates a pointer into the routing table. A routing tier represents one jurisdiction within a routing product. A routing product is the set of routing tiers that are assigned to one or more trunk groups. Each trunk group selects routing tiers individually, so the routing

product is merely conceptual. Routing product is mentioned here only to further convey the definition of routing tier.

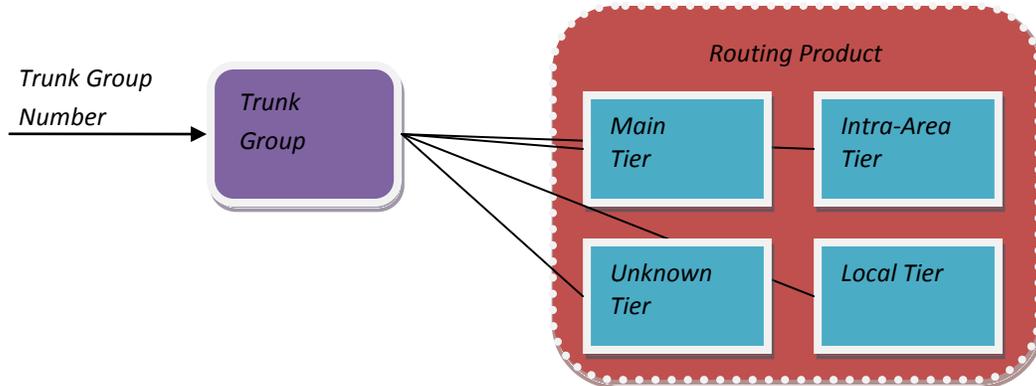


Figure 4: Routing Tiers

Determining Jurisdiction

Using the called number and calling number when available, jurisdiction can be determined. Note that even when an LRN is present in the call data, the called number is used for jurisdiction determination. The first check is to determine whether the call is of local jurisdiction. An intra-area check follows, which also determines whether the call is of unknown jurisdiction. If the jurisdiction is not found with the local or intra-area check, then the call is deemed to be inter-area and will use the main routing tier.

Local Jurisdiction

The main configuration file can define two local area data files. The first file (*lcaFile*) defines a cluster for each NPANXX in the North American Numbering Plan (NANP), along with business and residential calling plan numbers. The next file *clustersFile* defines the relationships between originating and terminating clusters based on the calling plan numbers.

```

<localArea>
  <lcaFile>LCA.TXT</lcaFile>
  <clustersFile>LCA_CLUSTERS.TXT</clustersFile>
</localArea>
  
```

If the `<main><localArea><lcaFile>` and `<main><localArea><clustersFile>` parameters are defined, the calling number is present, and the trunk group definition contains a local tier, then the local jurisdiction check occurs. The local jurisdiction check first finds the cluster to which the calling number belongs, along with the residential and business calling plan numbers. Next the cluster for the called number is determined. If a relationship exists between the calling cluster and the called cluster for either the residential or business calling plan numbers, then the call is considered to be of local jurisdiction.

Intra-Area Jurisdiction

The main configuration file may also define an area file that determines whether a call originates and terminates within the same region. It is expected that the file containing the area information that has fixed positioned entries. An NPANXX value is expected in positions 33 through 38. The letter 'A' is expected in position 39 and positions 78 and 79 contain the area identity code.

If the `<main><area><file>` parameter is defined and either the intra-Area tier or unknown Tier is defined on the trunk group, then the intra-Area jurisdiction check will occur. The intra-area jurisdiction check determines whether the calling and called numbers are members of the same area. If so, the call is considered to be of intra-area jurisdiction.

Unknown Jurisdiction

During the intra-area jurisdiction check if the calling number is not part of a defined area, or if the calling number length is not equal to the value defined in the main configuration file in the `<main><normalizedLength>` parameter, then the call is considered to be of unknown jurisdiction.

Inter-Area Jurisdiction

If the call jurisdiction has not been determined after both the local and intra-area jurisdiction checks have completed, then the call is considered to be of inter-area jurisdiction. The main routing tier is used for inter-area jurisdiction calls.

International Calling Numbers

By default, calls with calling numbers that have a country code that is different than the `<main><localCountryCode>` parameter in the main configuration file are international calling numbers. Calls with international calling numbers will route using the trunk group's main tier unless parameter `<main><area><intlTier>` is set to 'Unknown' in the main configuration file. In which case, the unknown tier will be used.

Trunk Group Definitions

Once the jurisdiction has been determined, it is used in combination with provisioned data to determine the routing tier. The trunk group definitions are loaded during application startup from the file specified in `<main><tgCarFile>`. Trunk groups may also be created and their parameters modified using the **slcrmgr** operations management interface. An example trunk group definition from the `tgCarFile` appears below.

```
<trunkGroup id="5678">
  <tier>GLDE</tier>
  <intraAreaTier>GLDA</intraAreaTier>
  <unknownTier>GLDU</unknownTier>
  <localTier>GLDL</localTier>
  <skips>PMX,ANT,KWC</skips>
  <customer>CRKT</customer>
</trunkGroup>
```

The id number specified in the trunk group must match the trunk group number received in signaling. As a result trunk group identifiers must contain only numeric digits.

The trunk group's main, intra-area, unknown and local tier definitions are defined using parameters `<tier>`, `<intraAreaTier>`, `<unknownTier>`, and `<localTier>`, respectively. After the call jurisdiction has been

determined, the trunk group number from the signaling data and the call jurisdiction combine to determine the routing tier. This is a simple lookup within the applications cache memory.

The skips and customer parameters are discussed in the carrier filtering section.

Find Carriers

Once the routing tier has been determined, the next phase of processing finds the preliminary routing carriers that will be used for egress routing. The provisioned routing tier and either the local routing number, if present, or the called number are used to search the routing tiers for the preliminary carrier list.

Tier Definition

Parameter `<main>tierDir` in the main configuration file defines the directory where tier load files reside. Any file containing extension `'xml'` that is present in this directory will be loaded into memory when the application starts. Tier load files contain many entries with destination code information and carrier lists separated by commas. A portion of an example tier file appears below.

```

<?xml version="1.0"?>
<!-- SoftCP LCR Routes -->

<LCR>
  <tier id="GLDL">
    <country id="1">
      <code id="201007"><list>CVM,VZY,NV2,PA3,GXG,TN8,SCP</list></code>
      <code id="201032"><list>VZY,NV2,TN8,SCP,CVM,GO2,NBG,GRX,PMT,PAE,L3X</list></code>
      <code id="201040"><list>CVM,BBM,VZY,PAE,GO2,NV1,SCP,L3X,PMI</list></code>
      <code id="201200"><list>MMP,PMI,ANT,KWK,NB2,CVM,PA3</list></code>
      <code id="2012009999"><list>PMI,ANT,MMP,KWK,CVM</list></code>
      <code id="201202"><list>GXG,80T,ATM,GNZ</list></code>
      <code id="201203"><list>KWK,BBM,ANT,VZY,GO2,PMT,TN8,SMP,ZSS,GXG</list></code>
      ... # Routing information truncated.
    </country>
    <country id="44">
      <code id="7"><list>PA3,80T,TN8,SCP</list></code>
      <code id="default"><list>VZY,NV2,TN8,SCP,PMT,PAE,L3X</list></code>
    </country>
    <country id="default">
      <code id="default"><list>TN8,SCP,XOT,PAE,L3X</list></code>
    </country>
  </tier>
</LCR>

```

Diagram annotations:

- Tier Identity**: points to `<tier id="GLDL">`
- Country Code "1" Routing**: points to `<country id="1">`
- Destination Code**: points to `<code id="201007">`
- Preliminary Carrier List**: points to `<list>CVM,VZY,NV2,PA3,GXG,TN8,SCP</list>`
- Country Code "44" Routing**: points to `<country id="44">`
- Default routing for Country Code "44"**: points to `<code id="default">` within the `<country id="44">` block.
- Default routing for all other countries**: points to `<code id="default">` within the `<country id="default">` block.

Figure 5: Example Tier Load File

Tier Identity

The tier identity is the first component of the tier load file. The tier identity can contain alphanumeric characters of any length. It is recommended that the tier name be kept as short as possible, however, for better memory utilization. More than one tier definition may appear within a tier load file.

Country Identity

Routing for one or more country codes may exist within a routing tier. A country id of "default" also allows the user to define a preliminary carrier list for all countries that are not specifically defined within the tier.

Destination Code

Within each country code are several entries defining preliminary carrier lists for specific dial string prefixes. When searching for digit matches using the called number or LRN, the application will find the best match that appears within the routing table.

A default destination code may appear within each country similar to the default country code entry. The default destination entry will provide a carrier list from the default entry if no list was found that matched any of the called number string.

Carrier List

Each destination code defined in the routing tier will have an associated comma delimited carrier list. Carriers may be of any length. Carrier codes must contain at least one alphabetic carrier, however, since list entries containing all digits are treated as cost elements for the previous carrier in the list. This carrier list becomes the preliminary carrier list that is delivered to the carrier filtering phase of the routing logic.

Inherit Tier

An inherit tier may be assigned to the tier that functions as a fallback routing tier. The inherit tier is not present in the example above. If routing is not found using the derived routing tier and an inherit tier is defined, a second route search will occur using the inherit tier. Only one level of tier inheritance occurs. If a tier that is used as an inherit tier contains an inherit tier, the second inherit tier will not be searched.

The inherit tier is defined within the tier load file as follows:

```
<?xml version="1.0"?>
<!-- SoftCP LCR Routes -->

<LCR>
  <tier id="SILVER">
    <inheritTier>GOLD</inheritTier>
    <country id="1">
      ...
    </tier>
  </LCR>
```

Carrier Filtering

The result of the 'find carriers' phase of the routing logic should be a list of carrier ids. The 'carrier filtering' routing logic phase will filter unwanted carriers based on one of many filtering mechanisms. This release of SLCR contains skip, quality and tier exclude filtering.

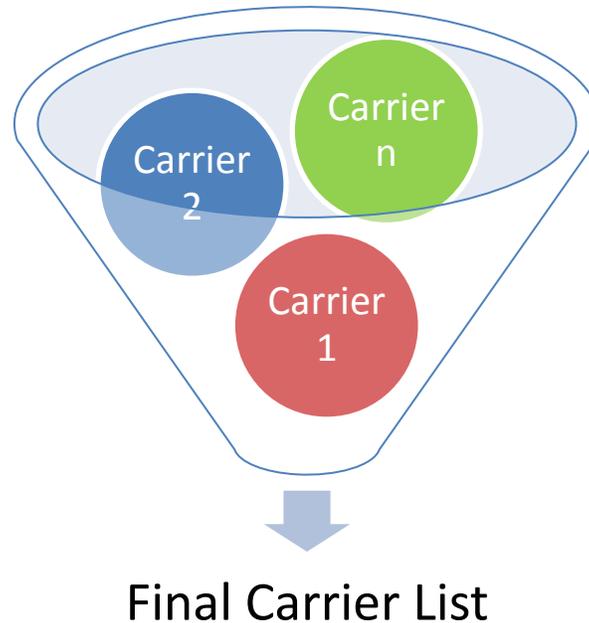


Figure 6: Carrier Filtering

Carrier Skips

Carrier skips can be defined at either the trunk group or customer entity. Up to this point, the customer entity has not been discussed. Like the trunk group, the customer entities are defined in the *tgCarFile*. They may also be created and/or modified using *slcrmgr*. The customer entity is referenced by the trunk group entity. Multiple trunk groups can point to a single customer.

The customer entity contains only a small set of parameters. An example customer definition is shown below:

```
<customer id="CRKT">
  <minQuality>70</minQuality>
  <skips>GZX,SCP</skips>
</customer>
```

The trunk group definition that was shown earlier contained two parameters that apply to carrier skip filtering.

```
<trunkGroup id="5678">
  ...
  <skips>PMX,ANT,KWC</skips>
  <customer>CRKT</customer>
</trunkGroup>
```

The *skips* parameter defined against the trunk group is a comma delimited list of carrier Ids. Any of the carriers present in the trunk group *skips* list that appear in the preliminary carrier list are removed and will not be part of the final carrier list.

The trunk group definition may optionally reference a customer identity. If a customer identity is defined, the referenced customer skips list is obtained. If any carriers from the preliminary carrier list are appear in the customer skips list, they are eliminated from the final carrier list.

Using our example trunk group '5678' if a preliminary routing list contained the following carriers:

ANT,GZX,PMX,SCP,XOT,KWC

The final carrier list would contain only XOT since ANT, PMX and KWC were skipped by the trunk group and GZX and SCP were skipped by customer CRKT.

Quality Filtering

Please contact SoftCP for more information regarding this feature.

Tier Excludes

During the 'find carriers' section of the routing logic, the inherit tier attribute of the routing tier was mentioned. Carrier definitions can include exclude tiers that allow the carrier to remove itself from the route list if a specific tier was used to find it.

The intent of this feature is to allow multiple routing products to exist within the same tier definition. The feature can allow gold, silver, bronze style routing within one routing product. The highest quality carriers will allow themselves to remain in the carrier list if the gold tier was the routing tier determined during the 'find tiers' logic phase, but if the silver or bronze tier was the original tier, the carrier would eliminate itself. Similarly, a silver level carrier would eliminate itself if the bronze tier were the original tier.

The following definition exemplifies how a simple gold, silver, bronze routing scheme could be accomplished.

```
<?xml version="1.0"?>
<!-- SoftCP LCR TgCar File -->

<LCR>
  <customer id="CUST">
  </customer>
  <trunkGroup id="40000001">
  ...
  </trunkGroup>
  <carrier id="GCOM">
    <name>Golden Communications</name>
    <swid>5000</swid>
    <tgid>1000</tgid>
    <excludeTiers>SLVR,BRNZ</excludeTiers>
  </carrier>
  <carrier id="STEL">
    <name>Silver Telecom</name>
    <swid>4000</swid>
    <tgid>1000</tgid>
```

```

</carrier>
<carrier id="BNET">
  <name>Bronze Networks</name>
  <swid>3000</swid>
  <tgid>1000</tgid>
</carrier>

<?xml version="1.0"?>
<!-- SoftCP LCR Routes -->

<LCR>
  <tier id="GOLD">
    <country id="default">
      <code id="default"><list>GCOM,STEL,BNET</list></code>
    </country>
  </tier>
  <tier id="SLVR">
    <inheritTier>GOLD</inheritTier>
  </tier>
  <tier id="BRNZ">
    <inheritTier>GOLD</inheritTier>
  </tier>
</LCR>

```

Skip Out Carrier

Please contact SoftCP for more information regarding this feature.

Response Creation

Once the final carrier list has been determined, a response needs to be created and returned to the querying node. Based on the interface type being used, there are many differences in the way the response message needs to be constructed. Two different SIP implementations are supported in this release in addition to supporting AINO.2 over SUA.

Carrier Definitions

Carriers and carrier lists have been mentioned during the earlier routing sections. The attributes assigned to each carrier only play a role when formulating the response message; therefore, the discussion of carrier definitions first appears here. The carrier definitions reside in the `tgCar` file referenced in the main configuration file. Some example carrier definitions appeared in the discussion about excluded tiers. The Golden Communications carrier definition is shown again below. This time with an added `host` attribute.

```

<carrier id="GCOM">
  <name>Golden Communications</name>
  <swid>5000</swid>
  <tgid>1000</tgid>
  <host>10.41.235.21</host>
  <excludeTiers>SLVR,BRNZ</excludeTiers>
</carrier>

```

Each carrier value can be any number of alphanumeric characters; however, the user is encouraged to keep the number of characters to a minimum (3 is suggested) so that memory utilization can be kept to a minimum. Another restriction on the carrier id is that it must have at least one alphabetic character. A carrier value containing all digits is reserved to indicate a cost value for completing the call to the carrier.

The carrier definition allows for a name description field to provide additional information about the carrier. The swid and tgid attributes define the switch Id and trunk group Id values that together create a unique identity used to convey the carrier value using certain protocols. The host attribute defines the host value in a Contact header when signaling using rfc3261 SIP – Session Initiation Protocol.

The swid, tgid and host parameters can be modified in real time using **slcrmgr**. The name and excludeTier attributes are fixed at load time.

SIP Responses

As mentioned, this product release supports two different SIP implementations. The implementations vary only in the response handling. The first implementation is labeled rfc3261 SIP. When using rfc3261 SIP the carrier host address will be returned in the Contact header of a 300 Redirect Request message. One contact will appear in the Contact header for each carrier in the final carrier list. If no carrier host address is defined, then a default host address value (1.1.1.1) will be returned, which will likely cause a routing failure or increased post-dial delay.

The other SIP implementation interacts with Sonus Networks Policy Server Engine (PSX). Please contact SoftCP for more information regarding this SIP implementation.

For both SIP implementation types as many as 10 contacts can be returned in the response. Additional carriers in the final carrier list are discarded.

Determining SIP Response Types

The SIP response type to be used can be defined at two levels. It can be defined globally within the main configuration file using parameter <main><sip><responseType>. The two valid values for this parameter are 'rfc3261' and 'Sonus'. If no value is specified, 'rfc3261' is used.

The SIP response type may also be defined at the customer entity. If a customer response type is defined, it takes precedence over the value specified in the main configuration file. The customer entity definition allows SLCR to interact with multiple switch types while sharing a single routing table.

SUA Responses

As mentioned, the SUA implementation uses AINO.2 over TCAP and SUA. AINO.2 uses three parameters to convey switch Id and trunk group Id values back to the querying node. These parameters are primaryTrunkGroup, alternateTrunkGroup and secondAlternateTrunkGroup within an AnalyzeRoute response message. As a result AINO.2 has a limit to the number of carriers that can be conveyed in one response. To combat this limit, multiple analyze route messages can be sent if the TCAP mode is set to conversation instead of query and response.

Setting <main><useConversation> to value 'true' in the main configuration file will cause all TCAP communication between SLCR and the softswitch or signaling gateway to be in conversational mode. In conversational mode, the first three carrier's switch Id and trunk group id values are populated in the primaryTrunkGroup, alternateTrunkGroup and secondAlternateTrunkGroup, respectively. The remaining carriers from the final carrier list are stored. When subsequent querying messages arrive in the form of NetoworkBusy event messages, the remaining carrier switch Id and trunk group Id values are populated in groups of three.

Using conversational mode allows a theoretically infinite number of carriers to be conveyed to the querying node. Network conditions will limit the practical number of carriers that can be returned.

When the query/response TCAP mode is used, only the first three carriers can be returned to the querying node. Remaining carriers in the final list are discarded. There are reasons to use query/response mode. The first is the capability of other network nodes. Some softswitches and signaling gateways only support query/response mode. Another significant reason is performance. While only a small performance impact is observed when using conversational mode with SLCR, conversational mode can have a significant performance impact on other network nodes.

Trunk Group Id from Jurisdiction Feature

Please contact SoftCP for more information regarding this feature.

Negative Responses

If no carriers were found or are available in the final carrier list, SLCR will return a negative response to the querying node. When using SIP, this negative response is a '503 No Route to Destination' message. Returning the '503' message allows the querying node to try an alternate routing engine or use its internal routing logic.

If using SUA, a Continue message is returned to the querying node. This will allow the node to continue its own internal processing.

Operational Handling

This section describes information and tools that are available to the operator to observe and influence the behavior of SLCR.

Call Detail Records

If a call detail record (CDR) directory is defined in the main configuration file (parameter <main><cdr><directory>), CDRs will be recorded in the directory specified for each query received. Two other parameters, <main><cdr><size> and <main><cdr><time> define the size and age, respectively that a CDR file may reach before a new CDR file is created.

CDR file names contain the name of the host server, an asp identifier and a six-digit sequence number separated by underscores (_). The asp identifier is always 100 when SIP is used as the communication interface. It can be specified by the user when SUA is in use. When the conditions are met for creating a new CDR file, the next available sequence number is obtained based on the existing files within the CDR directory.

One CDR is created for each query sent to SLCR. The information present in each record depends on the signaling interface employed. The tables below show the information present within each record for each signaling type.

SIP CDR Information

Position	Field Name	Description
1	Query Received Time	Time stamp for the query arrival.
2	Query Time	The time required to complete the query (sec).
3	Call Id	The Call Id header information from a SIP query.
4	Response Code	The response code returned to the querying node (300 or 503).
5	Trunk Group Number	The trunk group number received from the querying node.
6	Tier	The tier that was used to find the initial carrier list.
7	Calling Number	The calling number used for routing.
8	Called Number	The called number received from the querying node.
9	LRN	The Local Routing Number received from the querying node.
10	Jurisdiction	The call jurisdiction determined using the called and calling numbers.
11	Carrier List	The final carrier list to be returned.
12	Country Code	The country code from which the preliminary carrier list was found.
13	Destination Code	The destination code from which the preliminary carrier list was found.
14	Skipped Carriers	Carriers that were removed from the preliminary carrier list.
15	Filter Types	Bitmask containing the filtering types that caused carriers to be removed (Bit 0=Trunk Group Skip, Bit 1=Customer Skip, Bit 2=Quality Filtering, Bit 3=Tier Exclusion, Bit 4=Unknown Carrier).

SUA CDR Information

Position	Field Name	Description
----------	------------	-------------

1	<i>Thread Id</i>	<i>SUA thread used for the query.</i>
2	<i>Query Received Time</i>	<i>Time stamp for the query arrival.</i>
3	<i>Call Id</i>	<i>The TCAP dialog Id received.</i>
4	<i>Received AIN Data</i>	<i>The signaled information received in AIN format.</i>
5	<i>Query End Time</i>	<i>Time stamp when the response was sent.</i>
6	<i>AIN Response Data</i>	<i>The signaled information returned to the querying node in AIN format.</i>

Operations Management Interface – *slcrmgr*

SLCR has a companion application, *slcrmgr*, that allows users to view and alter various items within the application's memory. Please refer to the *Slcrmgr User's Guide* for more information.

Logging

The main configuration file contains parameters `<main><log><filename>` and `<main><log><level>` which define the location and logging level at which events are recorded in the system log file. The log location is relative to the directory specified in parameter `<main><baseDir>`. When SLCR starts, if the log file name specified already exists, the existing file is renamed with a time stamp extension added to the file name.

There are 5 levels at which system events are recorded. The table below defines each event level. Higher event levels inherit the logging information of lower log levels; therefore, logging level 4 is the most verbose. The logging level may be altered via *slcrmgr*.

System Log Levels

Level	Name	Description
0	<i>ERROR</i>	<i>System errors and warnings are recorded.</i>
1	<i>MGMT WRITE</i>	<i>User changes to the in memory database are recorded (Recommended production setting).</i>
2	<i>MGMT READ</i>	<i>User commands that do not alter the in memory database are recorded.</i>
3	<i>LOGIC_TRACE</i>	<i>Query logic flow events are recorded.</i>
4	<i>SIGNAL_TRACE</i>	<i>Signaling Messages are recorded.</i>

Single Call Trace

When SIP is used as the signaling interface, single call tracing is allowed regardless of the current logging level setting. If string 'trace' is present in the option portion of the RequestURI header, the logging level will be set to value 4 for the duration of the call query. All other ongoing call queries will record logging information based on the current logging level setting.

Example Call Trace INVITE Message:

```
INVITE sip:40000001#13034241234;npdi;trace@10.32.231.16:5070 SIP/2.0
Via: SIP/2.0/UDP 10.32.202.54:5060;branch=z9hG4bK-15478-1-0
From: sipp <sip:12146987300@10.32.202.54:5060>;tag=1
To: <sip:40000001#13034241234@10.32.231.16:5070>
```

Call-ID: 1-15478@10.32.202.54
 CSeq: 1 INVITE
 Contact: sip:12146987300@10.32.202.54:5060
 Max-Forwards: 70
 Subject: Performance Test
 Content-Type: application/sdp
 Content-Length: 133

v=0
 o=user1 53655765 2353687637 IN IP4 10.32.202.54
 s=-
 c=IN IP4 10.32.202.54
 t=0 0
 m=audio 6000 RTP/AVP 0
 a=rtpmap:0 PCMU/8000

Alarm Handling

SLCR generates syslog events when conditions arise where it is deemed that an operator should be alerted. Each generated event will use syslog facility 'daemon'. The severity level used depends on the event. A unique code will be generated for each event. The code contains two class digits separated by a dash and three digits indicating the event identifier. The first class digit indicates the protocol in use where zero (0) indicates SIP and one (1) indicates SUA. The second class digit identifies the logic region in which the event occurred.

The table below identifies each event and severity level.

Event Code	Severity	Description
00-000	Info	Application Ready Event indicating that the application is ready to begin receiving queries.
00-001	Error	No Base Directory Defined Parameter <main><baseDir> has not been defined in the main configuration file.
00-002	Error	No Interface (SUA/SIP) Defined Neither parameter <main><useSip> nor <main><useSua> was defined in the main configuration file.
x1-001	Error	Undefined Tier Attempted by Trunk Group The trunk group specified attempted to use a tier that is not present in the application's memory.
x3-003	Error	Skip Out Carrier Reference Undefined The Skip Out Carrier feature is in use, but the skip out carrier is not present in memory. The first occurrence of this event creates an error as does every 1000 th occurrence afterward.
x4-003	Error	Cannot Write to Socket The application is not able to respond to the querying node.
x5-001	Warning	SLCR Exiting on Management Command The application received a command to exit.

Appendix A. The Main Configuration File

The main configuration file contains pointers to all data files that are loaded upon application start up. It also provides location information for log and CDR files. In addition, any globally applied features are defined in this file. The table below provides an explanation of each parameter in the file.

Configuration Data			
Parameter			Description
<i>main</i>			
	<i>baseDir</i>		Defines the base directory name for the slcr application. All configuration file locations are relative to the base directory unless an explicit path is specified.
	<i>useSip</i>		Specifying a value of 'true' causes slcr to instantiate a SIP thread to process routing queries.
	<i>useSua</i>		Specifying a value of 'true' causes slcr to instantiate an SUA thread to process routing queries.
	<i>useConversation</i>		Specifying a value of 'true' causes slcr to use conversation mode instead of query/response mode for TCAP dialogs.
	<i>tgCarFile</i>		Defines the interface configuration file that contains customer, trunk group and carrier information.
	<i>qualityFile</i>		Defines the quality value configuration file.
	<i>tierDir</i>		Defines the location of the directory where tier routing data files are stored.
	<i>area</i>		
		<i>File</i>	Defines the area definition configuration file.
		<i>Digits</i>	Defines the number of digits of the calling and called numbers that should be used to determine area membership.
		<i>intlTier</i>	Defines whether the main or unknown tier is used when a calling number is of international jurisdiction [Main Unknown], DEFAULT=Main.
	<i>localArea</i>		
		<i>lcaFile</i>	Defines the local calling area cluster definition file.

		<i>clustersFile</i>	<i>Defines the local calling area cluster relationship file.</i>
	<i>log</i>		
		<i>filename</i>	<i>Defines the file name for logging events. This file must be unique for every slcr application running on the host server.</i>
		<i>level</i>	<i>Defines the level [0-4] for which logging events will be written to the log file. Log level 4 is a debug level and is the most verbose.</i>
	<i>cdr</i>		
		<i>directory</i>	<i>Defines the directory where CDR files will be written. If this parameter is omitted, no CDR information will be written.</i>
		<i>size</i>	<i>Defines the maximum allowable size of the CDR file before a new file will be created. Default value is 33554432 (32MB).</i>
		<i>time</i>	<i>Defines the maximum time (sec) allowed before a new CDR file is generated. Default value is 3600.</i>
	<i>management</i>		
		<i>port</i>	<i>Identifies the port to be used for the operator management interface. This value must be unique for each slcr application running on the host server.</i>
	<i>sip</i>		
		<i>port</i>	<i>Defines the SIP port to be used to listen for routing requests. DEFAULT=5060.</i>
		<i>responseType</i>	<i>Determines the SIP response type used. Valid values are 'rfc3261' (default) and 'Sonus'.</i>
<i>gateway</i>			
	<i>Primary</i>		
		<i>host</i>	<i>Defines the host name or IP address of the primary SG when using SUA.</i>
		<i>opc</i>	<i>Defines the originating point code that will be used to register with the signaling gateway. The slcr process will receive queries destined for this point code combined with the subsystem number specified below.</i>

		<i>ssn</i>	<i>Defines the subsystem number that will be used to register with the signaling gateway. The slcr process will receive queries destined for this subsystem number combined with the point code specified above.</i>
		<i>aspId</i>	<i>This parameter assigns an ASP Id to the process. It is used during the registration phase with the SG. The primary ASP Id is also used as part of the CDR file name. DEFAULT=100.</i>
		<i>trafficMode</i>	<i>The traffic mode type to be requested for the specified opc and ssn. Allowed values: 1 – Override 2 – LoadShare [DEFAULT] Consult signaling gateway documentation for traffic mode behavior.</i>
	<i>Secondary</i>		<i>All parameters for the secondary signaling gateway are the same as for the primary. Note that the definition of a secondary signaling gateway is optional.</i>
	<i>localCountryCode</i>		<i>Defines the country code used as the domestic country code.</i>
	<i>normalizedLength</i>		<i>Defines the normalized length of national calling and called numbers.</i>
	<i>tgidFromJurisdiction</i>		<i>Please contact SoftCP for more information regarding this feature.</i>
	<i>skipOutCarrier</i>		<i>Please contact SoftCP for more information regarding this feature.</i>

Example configuration file:

```

<?xml version="1.0"?>
<!-- SoftCP LCR Configuration -->

<LCR>
  <main>
    <baseDir>/export/home/scpuser/SLCR/bin</baseDir>
    <useSip>true</useSip>
    <sip>
      <port>5070</port>
      <threads>2</threads>
    </sip>
    <tgCarFile>slcrTgCar.xml</tgCarFile>
    <tierDir>tiers</tierDir>
    <localArea>
      <lcaFile>LCA.TXT</lcaFile>
      <clustersFile>LCA_CLUSTERS.TXT</clustersFile>
    </localArea>
    <area>
      <file>LERG6.DAT</file>
    </area>
  </main>
</LCR>

```

```
<digits>6</digits>
<intlTier>Unknown</intlTier>
</area>
<log>
  <filename>slcr.log</filename>
  <level>4</level>
</log>
<cdr>
  <directory>/export/home/scpuser/CDR</directory>
  <size>33554452</size>
  <time>600</time>
</cdr>
<management>
  <port>5334</port>
</management>
<localCountryCode>1</localCountryCode>
<normalizedLength>10</normalizedLength>
<tgidFromJurisdiction>true</tgidFromJurisdiction>
</main>
</LCR>
```