Solving Business Problems with OpenSIPS: Least Cost Routing

Peter Kelly / pkelly@sourcevox.com

Who we are

- VoIP and OpenSIPs software development and consultancy
- Based in UK
- Some of larger customers are
  - Localphone
    - Retail ITSP offering (VoIP accounts, apps, DIDs in UK, US, Europe, Worldwide)
    - Over 1,000,000 users
  - Magic Telecom
    - US Faclities based CLEC
  - Voxbeam
    - Wholesale, A-Z Termination, VoIP reseller
    - US CLEC
    - Terminate ~20,000,000 mins/week internationally
- We use
  - OpenSIPS
  - Asterisk
  - FreeSWITCH
  - RabbitMQ
  - Redis
  - Hadoop
  - Homer
  - Sangoma

Solving Business Problems With OpenSIPS….

… what is a business problem?

Business Problem (defined by me)
"Something that must be solved in order for a business to move forwards"

Problem (defined by The Internet)
"A perceived gap between the existing state and a desired state"

NOT (necessarily)
- "What the developers want to write"
- "What a 3rd party softwre company has written and wants the business to conform to"

Least Cost Routing

… what is Least Cost Routing?

Least Cost Routing (defined by Wikipedia)
"the process of selecting the path of outbound communications traffic based on cost"

This Businesses Definition
"route the call to the cheapest carrier"

Routing Example

Prefix	Carrier	Cost	Routing Order
447	Vodafone	0.02	2
447	BT	0.01	1 (cheapest)
1407	BT	0.03	3
1407	AT&T	0.01	1 (cheapest)
1407	Verizon	0.02	2

Easily solved using OpenSIPS module - DROUTING

DROUTING example

Gateways

```
mysql> select gwid, address, description from dr_gateways;
+------+---------+-------------+
| gwid | address | description |
+------+---------+-------------+
| 1    | 1.2.3.4 | BT          |
| 2    | 5.6.7.8 | Vodafone    |
| 3    | 9.0.1.2 | AT&T        |
| 4    | 3.4.5.6 | Verizon     |
+------+---------+-------------+
4 rows in set (0.00 sec)
```

Rules

```
mysql> select prefix, gwlist, description from dr_rules;
+--------+--------+-------------+
| prefix | gwlist | description |
+--------+--------+-------------+
| 447    | 1,2    | UK Mobile   |
| 1407   | 3,4,1  | USA Orlando |
+--------+--------+-------------+
2 rows in set (0.00 sec)
```

DROUTING example – routing call

Main (request) route

- Number dialled is 447….

```
if(!do_routing("$avp(system_id_int)", "DR_SORT")) {
        sl_send_reply("503", "Service Unavailable to that destination");
        exit;
}
```

This gives you

- OpenSIPs sets $du (destination URI) to
  - gwid1 (BT).
- Call is sent to BT

DROUTING example – call failure

Failure route

- Call has failed to BT.

```
#Do some failure checks
if(use_next_gw()) {
        #Send call to next carrier
}
```

This gives you

- OpenSIPs sets $du (destination URI) to
  - gwid2 (Vodafone).
- Call is sent to Vodafone

DROUTING example – call failure, carrier 2 is too expensive

reply route

- Call has failed to BT.

```
#Do some failure checks
if(use_next_gw()) {
        #Send call to next carrier
}
```

This gives you

- OpenSIPs sets $du (destination URI) to
  - gwid2 (Vodafone).
- Call is sent to Vodafone, and we lose money on the call

Obvious answer…!

- Don’t put loss making carriers into the list of carriers!!

- OpenSIPS has no knowledge of sell price

- Write/use external application to constantly update OpenSIPS with carriers.

So, we have solved the problem, right?:

- We use OpenSIPS using standard module (DROUTING)
- We can load in full intl. A-Z, 70,000+ prefixes
- We have 20 carriers = 20 ratesheets / week
- For every ratesheet
  - Recalculate all prefixes and load into OpenSIPS

All fine because we only have 1 sell tariff….
…. But now the business wants 2 (or 3, or 4, or n) sell tariffs.

- Gold
- Silver
- Bronze

As far as the business is concerned they still are solving the same problem

This Businesses Definition
"route the call to the cheapest carrier"

Multiple Sell Tariffs: Solution 1:

- Use DROUTING “groups” feature
  - Means n x 70,000 prefixes (no problem for OpenSIPS)
  - Means n x 20 updates to DROUTING
    - 20 ratesheets/week from carriers means updating OpenSIPs for each sell tariff

  - Quality considerations?
  - Is this really efficient, can the external app cope?

Multiple Sell Tariffs: Solution 2:

- Make OpenSIPS do the work!

- Turn to the next slide to find out how!

3 Stages to consider for our new business problem:

1. Handle 3 sell tariffs
    1. Bronze
    2. Silver
    3. Gold

2. Take into account carrier qualities

3. Still do LCR

What we are going to do:

1. Tell OpenSIPS about the Sell Tariffs
   (so we can automatically check if a call is loss making)

2. Tell OpenSIPS about the carrier qualities
   (so we can make sure we select the right carrier for the requested sell tariff)

3. Still do LCR, but in a more clever way!

4. Let OpenSIPS do the hard work

Stage 1: Get the Sell Tariffs into OpenSIPS

- In previous slides OpenSIPS had no awareness of sell price (it was “dumb”)
- Now OpenSIPS needs to know
  - Could use DROUTING module
    - Facility to have more than 1 prefix “group”
  - Could use DIALPLAN module
    - Facility to look up values based on a key
      - Key = prefix
      - Value = price

Stage 1: Get the Sell Tariffs into OpenSIPS

- DROUTING is better to match prefixes so we use this.

GROUP	Prefix	Sell Price
1 (GOLD)	447	0.05
1 (GOLD)	1407	0.10
2 (SILVER)	447	0.03
2 (SILVER)	1407	0.07
3 (BRONZE)	447	0.02
3 (BRONZE)	1407	0.05

Stage 1: Get the Sell Tariffs into OpenSIPS

- What the dr_rules DB looks like

```
mysql> select * from dr_rules WHERE groupid IN(1,2,3);
+--------+---------+--------+---------+----------+---------+--------+-------+-------------+
| ruleid | groupid | prefix | timerec | priority | routeid | gwlist | attrs | description |
+--------+---------+--------+---------+----------+---------+--------+-------+-------------+
|      3 | 1       | 447    |         |       10 |         | 1      | 5     | Gold 447    |
|      4 | 1       | 1407   |         |       10 |         | 1      | 10    | Gold 1407   |
|      5 | 2       | 447    |         |       10 |         | 1      | 3     | Silver 447  |
|      6 | 2       | 1407   |         |       10 |         | 1      | 7     | Silver 1407 |
|      7 | 3       | 447    |         |       10 |         | 1      | 2     | Bronze 447  |
|      8 | 3       | 1407   |         |       10 |         | 1      | 5     | Bronze 1407 |
+--------+---------+--------+---------+----------+---------+--------+-------+-------------+
6 rows in set (0.00 sec)
```

- Note the use of attrs column to hold the sell price
  - 5 = 5 cents

Stage 1: Get the Sell Tariffs into OpenSIPS

- How does the script work.

```
modparam("drouting", "rule_attrs_avp", '$avp(dr_rule_attrs)')

...

#By some mechanism we know this customer is GOLD
#(maybe by prefix?)
$avp(system_id_int) = 1;

#Save the initial $ru (As drouting will alter it)
$var(orig_ru) = $ru;

if(!do_routing("$avp(system_id_int)", "DR_SORT")) {
        #There is no sell price
        sl_send_reply("503", "Service Unavailable to that destination");
        exit;
}

#We have our sell price now, put back the changes
# do_routing() made
$ru = $var(orig_ru);
resetdsturi();

#Now we have our sell price in the attribute
xlog("--[$ci] Sell price for ru[$ru] is [$avp(dr_rule_attrs)]");
```

Stage 2: Get Carrier Qualities into OpenSIPS

- We now have 3 “sets” of carriers…
  - One “set” per Sell Product
- Why store 3 sets when you can store 1?
- DROUTING already contains a comma separated list of carriers.
- Let’s use that and use the attributes column again

Stage 2: Get Carrier Qualities into OpenSIPS

- What we put in DROUTING

```
mysql> select groupid, prefix, gwlist, attrs, description from dr_rules
where groupid=10;
+---------+--------+--------+--------------------+-------------+
| groupid | prefix | gwlist | attrs              | description |
+---------+--------+--------+--------------------+-------------+
| 10      | 447    | 1,2    | 1,1,G:2,2,S        | UK Mobile   |
| 10      | 1407   | 3,4,1  | 3,1,G:4,2,B:1,3,G | USA Orlando |
+---------+--------+--------+--------------------+-------------+
2 rows in set (0.00 sec)
```

- attrs are in a format OpenSIPS can understand and parse

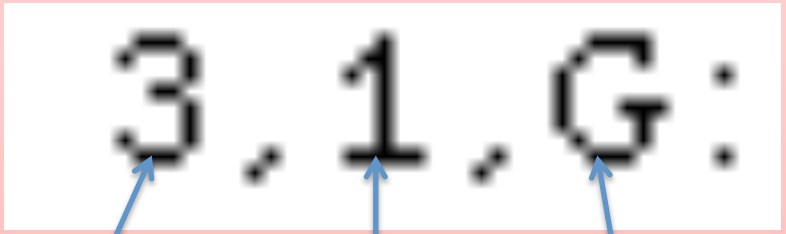
Stage 2: Get Carrier Qualities into OpenSIPS

- ***Matching keys***

gwlist	attrs
1,2	1,1,G:2,2,S
3,4,1	3,1,G:4,2,B:1,3,G

Gateway 3
(0th element in list)

Attributes for
gateway 3



Gateway 3
(check)

Cost price
= 1 cent

Gold
Quality

Stage 2: Get Carrier Qualities into OpenSIPS

- OpenSIPS understands the attributes string

```
3,1,G:4,2,B:1,3,G
```

```
$(avp(dr_rule_attrs){s.select,0,:});
```
= 3,1,G

```
$(avp(dr_rule_attrs){s.select,1,:});
```
= 4,2,B

```
$(avp(dr_rule_attrs){s.select,2,:});
```
= 1,3,G

Stage 3: Still do LCR

- We can do this now:
  - Sell Prices In OpenSIPS
  - Cost prices in OpenSIPs
  - Qualities in OpenSIPs
  - Discard any “non matching” gateways as the calls are routed
  - Cost price > sell price
  - Quality does not fit

Stage 1+2+3: DROUTING now looks like this:

```
mysql> select groupid, prefix, gwlist, attrs, description
from dr_rules where groupid IN (1,2,3);
+---------+--------+--------+-------+-------------+
| groupid | prefix | gwlist | attrs | description |
+---------+--------+--------+-------+-------------+
| 1       | 447    | 1      | 5     | Gold 447    |
| 1       | 1407   | 1      | 10    | Gold 1407   |
| 2       | 447    | 1      | 3     | Silver 447  |
| 2       | 1407   | 1      | 7     | Silver 1407 |
| 3       | 447    | 1      | 2     | Bronze 447  |
| 3       | 1407   | 1      | 5     | Bronze 1407 |
+---------+--------+--------+-------+-------------+
6 rows in set (0.00 sec)
```

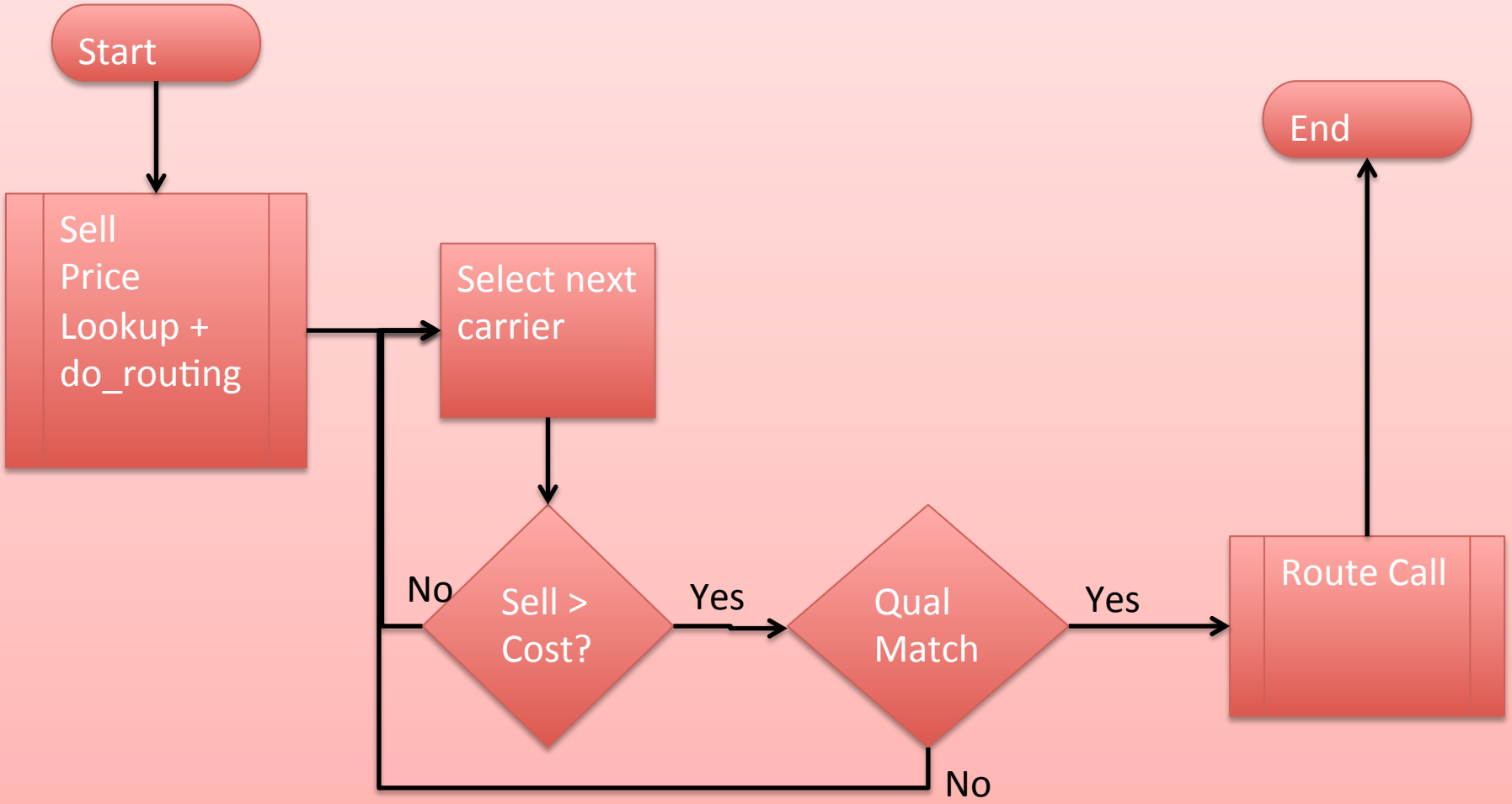
```
mysql> select groupid, prefix, gwlist, attrs, description from dr_rules
where groupid=10;
+---------+--------+--------+-------------------+-------------+
| groupid | prefix | gwlist | attrs             | description |
+---------+--------+--------+-------------------+-------------+
| 10      | 447    | 1,2    | 1,1,G:2,2,S       | UK Mobile   |
| 10      | 1407   | 3,4,1  | 3,1,G:4,2,B:1,3,G | USA Orlando |
+---------+--------+--------+-------------------+-------------+
2 rows in set (0.00 sec)
```

Stage 1+2+3: Summary:

OpenSIPs now knows about:

- All 3 Sell Tariffs
  - Prefixes
  - Sell Prices
- All carriers
  - Prefixes
  - Cost Price
  - Quality
- Now all we need to do is write the code!

Stage 1+2+3: Call Flow



```
route[RT_LCR_MAIN] {

        #Extract sell pricing

        #This gives us our first carrier
        do_routing(...);

        $avp(lcr_count) = 0;

        if(route(RT_LCR)) {
                #Route call
        }
}


route[RT_LCR] {

        $var(lcr_match) = 0;

        while($var(lcr_match) == 0) {

                #Retrieve attrs
                $var(lcr_entry) = $(avp(dr_rule_attrs){s.select,$avp(lcr_count),:});

                if(cost > sell || cost_quality < sell_quality) {
                        use_next_gw();
                        $avp(lcr_count) = $avp(lcr_count) + 1;
                } else {
                        $var(lcr_match) = 1;
                }
        }

        if($var(lcr_match) == 1) {
                return(1);
        } else {
                return(-1);
        }
}
```

```
failure_route[FAIL_PSTN] {


        #Checks on some final codes
        # e.g. 486, 404

        if($T_reply_code == "486"...) {
                exit;
        }

        #Otherwise we can continue LCR

        #$avp(dr_rules_attrs) still exists and is at the current iteration
        #Our gateway list is still accessed using DROUTING and is at the current iteration

        if(route(RT_LCR)) {
                #Route call
        }
}
```

We are using DROUTING, also using the powers of OpenSIPs scripting

- While loops
- Function (route) calls
  - This can be done recursively too
- Counters
- Iterating through arrays

- Applying real business logic to a SIP call as it is routed

We have demonstrated:

- Basic system in place for LCR
- Takes into account business scenarios
  - Supports Multiple Sell Tariffs
  - Negotiates carrier attributes dynamically
- Easily modified to handle more complex scenarios
  - Guaranteed CLI?
  - Dialer Traffic?
- Could even give users custom prices for individual prefixes
- Could even allow loss making calls to a specified percentage

- Data size in OpenSIPS stays relatively static
- Processing by external system stays static
  - Simply reorder carriers by cost when ratesheet arrives
  - No need for 20 x n processing
- When complexity is ramped up makes the old way almost impossible in terms of data size

One more example:

- NPANXX Jurisdictional routing
- 900,000+ prefixes in NPANXX list
  - LRN Dip
  - LNP Porting
  - 5 jurisdictions
- Use OpenSIPS to choose Jurisdiction
  - Then apply least cost routing

Conclusion:

- It is possible to solve great problems if you explore outside the static modules
- Applies to OpenSIPS especially due to the power of the scripting language
- Make OpenSIPS perform your business logic for you
  - More efficient than using external systems
  - Creating real business specific solutions

Questions?

pkelly@sourcevox.com
www.sourcevox.com