SIP Routing - Headers

VIA headers

Route headers

SIP Routing - VIA Headers

When a UAC sends a request it adds a VIA header to the message

Each proxy adds a VIA before passing the request to the next node

SIP Routing - Route Headers

Route headers force routing thru a proxy

Future requests follow Route headers

SIP Routing - Recap

Requests follow Route headers

Responses follow VIA headers

Routing Modules

Carrier Route (carrierroute)
Load Balancer (load_balancer)
Dispatcher (dispatcher)
Dynamic Routing (drouting)

Carrier Route Module

Routing, Load balancing, Blacklisting

Not installed by default

No web GUI

Carrier Route Module

Performs longest prefix matching

Strips prefix, adds prefix / suffix

Probabilities to load balance

Longest Prefix Matching

Algorithm used by routers to select a routing table entry

Adapted to the telecom world to select a (least cost) routing table entry

Load Balancer Module

Provides routing based on load

Configurable via the control panel

Single database table

Load Balancer Module

Single function call: load_balance()

No support for altering prefix/suffix

Load counters are in memory only

Dispatcher Module

Computes hash over parts of the request and selects an address from the destination list

8 different hashing algorithms

CallerID, From URI, To URI, Request URI, …

Dispatcher Module

Configuration from file or DB

Out of service destinations detected by probing

Does not strip or append prefix/suffix

Dynamic Routing Module

Multiple criteria used to select destination

Prefix, Caller, Group, Time, Priority

Configuration from DB only, 4 tables - groups, rules, carriers, gateways

Dynamic Routing Module

do_routing() uses groups, rules, carriers, gateways

route_to_carrier uses carriers, gateways

route_to_gateway uses gateways

OpenSIPS Control Panel - GUI

Dispatcher

Dynamic Routing

Load Balancer

Possible Use Cases - YMMV

Dispatcher - routing to carriers or backend FreeSWITCH cluster

Dynamic Routing - routing to carriers

Load Balancer - routing to carriers or backend FreeSWITCH clusters