

## **Comdasy** FAQ: SIP

Comdasy AG  
Rüdesheimer Str. 7  
D-80686 München  
Tel.: +49.89.5484333-0  
Fax: +49.89.5484333-29  
support@comdasy.com  
<http://www.comdasy.com>

## **Disclaimer**

We have taken all possible care to ensure that this manual contains correct, accurate information. However, the manufacturer cannot assume liability for any possible errors. In addition, the manufacturer cannot guarantee that the hardware will meet the purpose you require.

Comdasys reserves the right to make changes according to technical progress at any time. Brand names may be registered trademarks and must be treated as such.

© Copyright 2005-2008  
Comdasys AG  
80686 München, Germany

All rights reserved. No part of this manual may be reproduced, processed or distributed in any form (print, photocopy, microfilm or any other process) or processed by an electronic system without prior written permission from the manufacturer.

# Inhaltsverzeichnis

1. Multiple SIP Ports.....	4
1.1. Introduction.....	4
1.2. Firewall rule.....	4
2. Survivability Mode Traps.....	4
2.1. Introduction.....	4
2.2. Creating a Sample Trap.....	4
2.3. Creating a Sample Trap.....	5
3. Fixing VIA Header.....	5
3.1. Introduction.....	5
3.2. Firewall rule.....	5
4. Using DNS SRV.....	6
4.1. Introduction.....	6

# 1. Multiple SIP Ports

## 1.1. Introduction

In some cases it's desired to have a Convergence accept SIP messages on multiple ports. The easiest way to accomplish this is by using the firewall to redirect the packets in question.

## 1.2. Firewall rule

Convergences are by default able to do this redirection. The easiest way is to create or edit the file `/etc/sysconfig/firewall.custom` and add the following rules (the port number 30000 to 30255 of course have to be adjusted to the respective needs):

```
iptables -t nat -I PREROUTING -p udp --dport \  
30000:30255 -j REDIRECT --to-port 5060  
iptables -t nat -I PREROUTING -p tcp --dport \  
30000:30255 -j REDIRECT --to-port 5060
```

Afterwards you need to call `applyfirewall.sh` or press the *Apply changes* button in the web GUI.

# 2. Survivability Mode Traps

## 2.1. Introduction

In some cases it is desirable to have the Convergence send traps for switching from/to survivability mode. A single trap will then be sent for every mode change.

## 2.2. Creating a Sample Trap

We will need two traps so make this function sensible which we will `/usr/share/snmp/mibs` (we will name ours `TRAP-SURV-MIB.txt`):

```
TRAP-SURV-MIB DEFINITIONS ::= BEGIN  
    IMPORTS ucdExperimental FROM UCD-SNMP-MIB;  
  
    survtraps OBJECT IDENTIFIER ::= { ucdExperimental 990 }  
  
    surv-enter TRAP-TYPE  
        STATUS current  
        ENTERPRISE survtraps  
        VARIABLES { sysLocation }  
        DESCRIPTION "Enter Survivability Mode"  
        ::= 17  
  
    surv-exit TRAP-TYPE  
        STATUS current  
        ENTERPRISE survtraps
```

```
VARIABLES { sysLocation }
DESCRIPTION "Exit Survivability Mode"
::= 18
```

END

## 2.3. Creating a Sample Trap

The traps will be initiated by 2 hook scripts that you can add into the /etc/customscripts directory. The first would be the survivability-enter script. The HOSTTONOTIFY method has to be set to the destination for your trap. As always, the \ means that the content should be a single line that has been formatted here for better readability.

```
#!/bin/bash

HOSTTONOTIFY=10.0.0.197

snmptrap -v 1 -c public $HOSTTONOTIFY \
  TRAP-SURV-MIB::survtraps \
  ' 6 17 ' SNMPv2-MIB::sysLocation.0 s \
  "Survivability Mode Enter"
```

You have to make sure that this script has execute permissions. If not use chmod +x survivability-enter to set them. The second script would be /etc/customscripts/survivability-leave. As before, the HOSTTONOTIFY variable should be set to the destination for your traps.

```
#!/bin/bash

HOSTTONOTIFY=10.0.0.197

snmptrap -v 1 -c public $HOSTTONOTIFY \
  TRAP-SURV-MIB::survtraps \
  ' 6 18 ' SNMPv2-MIB::sysLocation.0 s \
  "Survivability Mode Exited"
```

## 3. Fixing VIA Header

### 3.1. Introduction

In some cases it's desirable to have the answer to SIP requests to be sent to a certain IP. The one thing that is responsible for doing this is the record route SIP header. There is a field in the WebGUI of the SIP Proxy that allows you to preset this record route header to a specified IP address.

### 3.2. Firewall rule

Convergences will by default autodetect the IP address of the interface the SIP message is sent out on. This IP address will then be recorded in the VIA header. In some cases, for example if an interface has multiple IP addresses, some type of NAT is performed through custom firewall rules, etc. this detection will however fail. For those, it is possible to manually specify the IP address inserted into the VIA header as follows. Simply insert the following two lines into

/etc/ser/ser.cfg before the route block.

```
advertised_address=192.168.x.x  
advertised_port=5060
```

Afterwards you need to call restart the SIP Proxy with `rcser restart`.

## 4. Using DNS SRV

### 4.1. Introduction

DNS SRV is used for making service based lookups. Most SIP devices like phones and gateways support it. It can be either used for failover configurations or load balancing. Instead, of directly looking up a hostname, we look up any "SRV" DNS records for a given service, e.g. SIP. These will have a priority, weight, and server name as the value part. The priority is used to pick one of several servers (this gives us failover), and then use the weight to pick among servers of the same priority (this gives us load balancing). In order to configure this, it is necessary to appropriately edit the zone file.

SRV records (specified in RFC2782) are in the form of

```
"_Service._Proto.Name TTL Class SRV Priority Weight Port Target"
```

For example,

```
"_sip._udp.foo.bar 43200 IN SRV 10 10 5060 sipserver.foo.bar."
```

The service is SIP.

The transport is UDP. Other values could be TCP, SCTP or TLS.

The cache lifetime is 12 hours (43,200 seconds.) This could be any positive signed 32 bit integer.

The class is IN (this is always true.)

The record type is SRV.

The priority is 10. With multiple SRV records the priority determines the proxy query order. Lower values are queried first.

The weight is 10. With multiple SRV records of similar priority, the weight determines proportionally how often a proxy is queried. Higher values are queried more often. So, a weight of 20 would be queried twice as often as one of 10. A weight of 30 would be queried three times as often as one of 10.

The port is 5060.

The proxy server FQDN is sipserver.foo.bar and as is required in DNS the FQDN is terminated with a dot.

An example DNS implementation with redundant proxy servers might look like this:

```
foo.bar IN SOA ns.foo.bar. root.foo.bar. (  
    2003032001  
    10800  
    3600  
    604800  
    86400 )
```

```

foo.bar.           43200 IN NS      ns.foo.bar.
;
ns.foo.bar.       43200 IN A      10.0.0.20
sipserver1.foo.bar. 43200 IN A      10.0.0.21
sipserver2.foo.bar. 43200 IN A      10.0.0.22
;
_sip._udp.foo.bar. 43200 IN SRV 0 0 5060 sipserver1.foo.bar.
_sip._udp.foo.bar. 43200 IN SRV 1 0 5060 sipserver2.foo.bar.
_sip._tcp.foo.bar. 43200 IN SRV 0 4 5060 sipserver1.foo.bar.
_sip._tcp.foo.bar. 43200 IN SRV 0 2 5060 sipserver2.foo.bar.
_sips._tcp.foo.bar. 43200 IN SRV 0 0 5060 sipserver1.foo.bar.
_sips._tcp.foo.bar. 43200 IN SRV 0 0 5060 sipserver2.foo.bar.

```

In this configuration UDP SIP requests will always be sent to sipserver1 because the priority value is lower than sipserver2. Should that request fail, sipserver2 would be queried. In effect, sipserver2 is an emergency backup to sipserver1.

Two TCP SIP requests will be sent to sipserver1 for each one sent to sipserver2 because the weight for sipserver1 is twice that for sipserver2. In this case sipserver1 and sipserver2 are both being queried and load balanced. A weight such as this might be used where one machine is significantly more powerful than another.

Secure SIP requests will be equally load balanced between the two servers.

A SIP UA wanting to initiate a call to sip:bigcheese@foo.bar will first send a DNS SRV lookup to foo.bar. With a successful return, the SIP URI may be re-written to sip:bigcheese@sipserver1.foo.bar. Absent information on bigu's preference for transport, the calling UA will choose the transport it prefers among the responses it gets.

In this example the calling UA could not use SCTP as a transport. The available choices are UDP, TCP, TLS or allow the call to fail.