Minimal CLI OpenSAND setup v6.0 DEV

DEV version

Overview

The objective of this example OpenSAND deployment example is to provide a very simple setup, focused on step-by-step approach. All commands are run manually on the command-line interface, from network configuration to binary startup.

This section will guide you through the steps required:

1. Network configuration
2. Simulation configuration
3. Run OpenSAND entities

Platform description

This is an example of how to setup the network to emulate a simple SATCOM System using OpenSAND. The topology of the considered platform is described in the figure below. It includes a satellite (SAT), a Gateway (GW0) and a Satellite Terminal (ST1). Once this setup is done, refer to IP mode or Ethernet mode following the interconnection method selected.
The OpenSAND network is a overlay network (Platform Network, 192.168.63.0/24) running on top of the Emulation network (192.168.18.0/24). It uses TAP devices to implement the DVB stack over the Emulation network. The Platform network will show characteristics similar to a real SATCOM network using DVB-S2/RCS2 protocols.

**Network Configuration**

It is assumed that the networking configuration for the Emulation Network is already done. So, we focus here on the OpenSAND network.

The setup will follow the steps below, on the ST and GW0 entities:

1. Create the TAP interface, on which the OpenSAND binary will read/write packets
2. Create the OpenSAND Bridge: since OpenSAND is a TAP interface it does not have an IP stack.
   - Linking the tap interface to a bridge allows to use the kernel IP stack.
3. Attach the OpenSAND interface to the bridge.
4. Configure the IP addresses of the OpenSAND/Satcom network
The satellite does not need any specific networking configuration.

Creation of the TAP interface:
$ ip tuntap add mode tap opensand_tap

Create the OpenSAND bridge:
$ ip link add name opensand_br type bridge

Attach the OpenSAND interface to the bridge:
$ ip link set dev opensand_tap master opensand_br

And set interfaces up:
$ ip link set opensand_br up
$ ip link set opensand_tap up

The opensand bridges should have been assigned a MAC address @opensand_br. You can see this address with:
$ ip addr show dev opensand_br
Remember the ST and GW bridges addresses since they will be used in the next configuration step.

OpenSAND Emulation Configuration

Each entity need to access their configurations files in /etc/opensand so they know which parameters of the simulation they need to load. In this example, only the necessary modification for a working example are show.

First, get the reference scenario configuration files, located in the folder /usr/share/opensand. Now, several files can be edited:

- topology.conf to specify how entities are interconnected;
- core_global.conf to specify generic parameters of the simulation;
- the plugins folder;
- gw0: Gateway parameters
- st1: Satellite Terminal parameters

The example provided emulates a classical DVB-S2/RCS2 setup, and no particular modification of the configuration files is needed. However, the following sections of topology.conf need to be edited: First, add the bridge addresses of the terminal (id 1) and gateway (id 0) to the sarp section, replacing with the bridge addresses from the previous section:

topology.conf

```xml
<sarp>
  <!-- The default destination terminal if no one is found, -1 for none -->
  <default>-1</default>
  <ethernet>
    [...]  
  </ethernet>
  <!-- MAC for terminal -->
  <terminal_eth mac="00:00:00:00:02" tal_id="1"/>
  <terminal_eth mac="00:00:00:00:00:01" tal_id="0"/>
</sarp>
```
Second, modify the addresses in the sat_carrier section accordingly:

```
topology.conf

<sat_carrier>
  <spot id="1" gw="0">
    <carrier id="0" type="ctrl_out" ip_address="239.137.194.221" port="55000">
      <!-- SAT to GW/ST control (CR, TBTP, SoF) -->
      <ip_multicast="true"/>
    </carrier id="1" type="ctrl_in" ip_address="192.168.18.15" port="55001">
      <!-- GW/ST to SAT control (CR, TBTP, SoF) -->
      <ip_multicast="false"/>
    </carrier id="2" type="logon_out" ip_address="192.168.18.42" port="55002">
      <!-- SAT to GW Logon -->
      <ip_multicast="false"/>
    </carrier id="3" type="logon_in" ip_address="192.168.18.15" port="55003">
      <!-- ST to SAT Logon -->
      <ip_multicast="false"/>
    </carrier id="4" type="data_out_st" ip_address="239.137.194.222" port="55004">
      <ip_multicast="true"/>
    </carrier id="5" type="data_in_st" ip_address="192.168.18.15" port="55005">
      <!-- ST to SAT Data -->
      <ip_multicast="false"/>
    </carrier id="6" type="data_out_gw" ip_address="192.168.18.42" port="55006">
      <!-- SAT to GW Data -->
      <ip_multicast="false"/>
    </carrier id="7" type="data_in_gw" ip_address="192.168.18.15" port="55007">
      <!-- GW to SAT Data -->
      <ip_multicast="false"/>
    </carrier id="8" type="logon_in" ip_address="192.168.18.15" port="55008">
    </carrier id="9" type="ctrl_in" ip_address="192.168.18.15" port="55009">
    </carrier id="10" type="ctrl_out" ip_address="192.168.18.15" port="55010">
    </carrier id="11" type="logon_out" ip_address="192.168.18.15" port="55011">
    </carrier id="12" type="logon_in" ip_address="192.168.18.15" port="55012">
    </carrier id="13" type="data_out_st" ip_address="192.168.18.15" port="55013">
    </carrier id="14" type="data_in_st" ip_address="192.168.18.15" port="55014">
    </carrier id="15" type="data_out_gw" ip_address="192.168.18.15" port="55015">
    </carrier id="16" type="data_in_gw" ip_address="192.168.18.15" port="55016">
  </spot>
</sat_carrier>
```

Now, upload files [TODO].

## IP mode

It is the default interconnection method. The network topology corresponding to this mode is the one presented in the figure above. In this mode, IP protocol is used for communications between gateway and ST. In order to do so, we assign IP addresses to the opensand bridges, by issuing the following commands:

On GW0:

```
$ ip address add 192.168.63.254/24 dev opensand_br
```

On ST1:

```
$ ip address add 192.168.63.1/24 dev opensand_br
```
If in your topology, there are LANs behind ST1 and GW0, additional configuration can be necessary, such as enable ip forwarding. If so, you have to setup routing over the satellite network by configuring routing on STs and GW0.

You also need to tell opensand that it will carry IP packets by setting the component `lan_adaptation_schemes` in core.conf as follows:

```
core.conf

<global>
  <!-- LAN adaptation, header compression/suppression schemes -->
  <lan_adaptation_schemes>
    <lan_scheme pos="0" proto="IP"/>
  </lan_adaptation_schemes>
</global>
```

**Ethernet mode**

This is an example of how to setup the network to emulate a simple SATCOM System in **Ethernet mode** using OpenSAND. The topology of the considered platform is described in the figure below as well as IPv4 addresses. It includes a satellite (SAT), a Gateway (GW0) and a Satellite Terminal (ST1). Work-stations server (SRV) and client (CLT) belonging to distant terrestrial networks, are on the same IP subnet and communicate over satellite segment.

![Ethernet mode diagram](image)

In this mode, Ethernet protocol is used for communications between gateway and satellite terminal, and LANs behind ST1 and GW0 are in a same subnet. In order to do so, we add the network interface of each LAN (ens5 in this case) to opensand bridge, by issuing the following commands:

On GW0 and ST1

```
sip link set dev ens5 master opensand_br
```

In this setup, additional configuration is needed in topology.conf since OpenSAND does not implement yet learning of MAC addresses. For each workstation connected behind ST1 and GW0, a
sarp must be filled to bind this MAC address with its corresponding terminal ID. This process is needed in order for the satellite carriers to work properly.

topology.conf

```xml
<sarp>

    <!-- The default destination terminal if no one is found, -1 for none -->
    <default>-1</default>
    <ethernet>
        [...]
        <!-- MAC for terminal -->
        <terminal_eth mac="01:02:03:00:00:00:02" tal_id="1"/>
        <terminal_eth mac="01:02:03:00:00:00:01" tal_id="0"/>
    </ethernet>
</sarp>
```

where 01:02:03:00:00:01 and 01:02:03:00:00:02 are the MAC addresses of the workstations SRV and CLT respectively.

You also need to tell opensand that it will carry Ethernet frames by setting the component lan_adaptation_schemes in core.conf as follows:

core.conf

```xml
<global>
    <!-- LAN adaptation, header compression/suppression schemes -->
    <lan_adaptation_schemes>
        <lan_scheme pos="0" proto="ETH"/>
    </lan_adaptation_schemes>
</global>
```

Starting entities

In this setup, all statistics and logs will be directed to a local directory in ~/opensand_simulation and to a remote collector (ip address 192.168.18.15, stat port 5361, log port 5362).

First, start the satellite:

```bash
opensand sat -a 192.168.18.15 -c /etc/opensand -r 192.168.18.15 -s 5361 -l 5362 -f ~/opensand_simulation
```

Start the Gateway:

```bash
opensand gw -a 192.168.18.42 -c /etc/opensand -t opensand tap -i 0 -r 192.168.18.15 -s 5361 -l 5362 -f ~/opensand_simulation
```

Start the Satellite Terminal:

```bash
opensand st -a 192.168.18.1 -c /etc/opensand -t opensand_tap -i 1 -f ~/opensand_simulation -r 192.168.18.15 -s 5361 -l 5362 -f ~/opensand_simulation
```