

Shared Line Appearances

Russell Bryant
Software Engineer
Digium, Inc.

March 14, 2007

Contents

1	Introduction	1
2	Configuration	2
2.1	Summary	2
2.2	Dialplan	2
2.3	Trunks	2
2.4	Stations	4
3	Configuration Examples	5
3.1	Basic SLA	5
3.2	SLA and Voicemail	6
4	Call Handling	9
4.1	Summary	9
4.2	Station goes off hook (not ringing)	9
4.3	Station goes off hook (ringing)	9
4.4	Line button on a station is pressed	9

1 Introduction

The "SLA" functionality in Asterisk is intended to allow a setup that emulates a simple key system. It uses the various abstraction layers already built

into Asterisk to emulate key system functionality across various devices, including IP channels.

2 Configuration

2.1 Summary

An SLA system is built up of virtual trunks and stations mapped to real Asterisk devices. The configuration for all of this is done in three different files: `extensions.conf`, `sla.conf`, and the channel specific configuration file such as `sip.conf` or `zapata.conf`.

2.2 Dialplan

The SLA implementation can automatically generate the dialplan necessary for basic operation if the "autocontext" option is set for trunks and stations in `sla.conf`. However, for reference, here is an automatically generated dialplan to help with custom building of the dialplan to include other features, such as voicemail (3.2).

However, note that there is a little bit of additional configuration needed if the trunk is an IP channel. This is discussed in the section on trunks (2.3).

There are extensions for incoming calls on a specific trunk, which execute the SLATrunk application, as well as incoming calls from a station, which execute SLAStation. Note that there are multiple extensions for incoming calls from a station. This is because the SLA system has to know whether the phone just went off hook, or if the user pressed a specific line button.

Also note that there is a hint for every line on every station. This lets the SLA system control each individual light on every phone to ensure that it shows the correct state of the line. The phones must subscribe to the state of each of their line appearances.

Please refer to the examples section for full dialplan samples for SLA.

2.3 Trunks

An SLA trunk is a mapping between a virtual trunk and a real Asterisk device. This device may be an analog FXO line, or something like a SIP trunk. A trunk must be configured in two places. First, configure the device

itself in the channel specific configuration file such as zapata.conf or sip.conf. Once the trunk is configured, then map it to an SLA trunk in sla.conf.

```
[line1]
type=trunk
device=Zap/1
```

Be sure to configure the trunk's context to be the same one that is set for the "autocontext" option in sla.conf if automatic dialplan configuration is used. This would be done in the regular device entry in zapata.conf, sip.conf, etc. Note that the automatic dialplan generation creates the SLATrunk() extension at extension 's'. This is perfect for Zap channels that are FXO trunks, for example. However, it may not be good enough for an IP trunk, since the call coming in over the trunk may specify an actual number.

If the dialplan is being built manually, ensure that calls coming in on a trunk execute the SLATrunk() application with an argument of the trunk name, as shown in the dialplan example before.

IP trunks can be used, but they require some additional configuration to work.

For this example, let's say we have a SIP trunk called "mytrunk" that is going to be used as line4. Furthermore, when calls come in on this trunk, they are going to say that they are calling the number "12564286000". Also, let's say that the numbers that are valid for calling out this trunk are NANP numbers, of the form _1NXXNXXXXXX.

In sip.conf, there would be an entry for [mytrunk]. For [mytrunk], set context=line4.

```
[line4]
type=trunk
device=Local/disa@line4_outbound
```

```
[line4]
exten => 12564286000,1,SLATrunk(line4)
```

```
[line4_outbound]
exten => disa,1,Disa(no-password|line4_outbound)
exten => _1NXXNXXXXXX,1,Dial(SIP/${EXTEN}@mytrunk)
```

So, when a station picks up their phone and connects to line 4, they are connected to the local dialplan. The Disa application plays dialtone to the phone and collects digits until it matches an extension. In this case, once the phone dials a number like 12565551212, the call will proceed out the SIP trunk.

2.4 Stations

An SLA station is a mapping between a virtual station and a real Asterisk device. Currently, the only channel driver that has all of the features necessary to support an SLA environment is `chan_sip`. So, to configure a SIP phone to use as a station, you must configure `sla.conf` and `sip.conf`.

```
[station1]
type=station
device=SIP/station1
trunk=line1
trunk=line2
```

Here are some hints on configuring a SIP phone for use with SLA:

1. Add the SIP channel as a `[station]` in `sla.conf`.
2. Configure the phone in `sip.conf`. If automatic dialplan configuration was used by enabling the "autocontext" option in `sla.conf`, then this entry in `sip.conf` should have the same context setting.
3. On the phone itself, there are various things that must be configured to make everything work correctly:

Let's say this phone is called "station1" in `sla.conf`, and it uses trunks named "line1" and "line2".

- (a) Two line buttons must be configured to subscribe to the state of the following extensions: - `station1_line1` - `station1_line2`
- (b) The line appearance buttons should be configured to dial the extensions that they are subscribed to when they are pressed.
- (c) If you would like the phone to automatically connect to a trunk when it is taken off hook, then the phone should be automatically configured to dial "station1" when it is taken off hook.

3 Configuration Examples

3.1 Basic SLA

This is an example of the most basic SLA setup. It uses the automatic dialplan generation so the configuration is minimal.

sla.conf:

```
[line1]
type=trunk
device=Zap/1
autocontext=line1

[line2]
type=trunk
device=Zap/2
autocontext=line2

[station](!)
type=station
trunk=line1
trunk=line2
autocontext=sla_stations

[station1](station)
device=SIP/station1

[station2](station)
device=SIP/station2

[station3](station)
device=SIP/station3
```

With this configuration, the dialplan is generated automatically. The first zap channel should have its context set to "line1" and the second should be set to "line2" in zapata.conf. In sip.conf, station1, station2, and station3 should all have their context set to "sla_stations".

For reference, here is the automatically generated dialplan for this situation:

```
[line1]
exten => s,1,SLATrunk(line1)

[line2]
exten => s,2,SLATrunk(line2)

[sla_stations]
exten => station1,1,SLAStation(station1)
exten => station1_line1,hint,SLA:station1_line1
exten => station1_line1,1,SLAStation(station1_line1)
exten => station1_line2,hint,SLA:station1_line2
exten => station1_line2,1,SLAStation(station1_line2)

exten => station2,1,SLAStation(station2)
exten => station2_line1,hint,SLA:station2_line1
exten => station2_line1,1,SLAStation(station2_line1)
exten => station2_line2,hint,SLA:station2_line2
exten => station2_line2,1,SLAStation(station2_line2)

exten => station3,1,SLAStation(station3)
exten => station3_line1,hint,SLA:station3_line1
exten => station3_line1,1,SLAStation(station3_line1)
exten => station3_line2,hint,SLA:station3_line2
exten => station3_line2,1,SLAStation(station3_line2)
```

3.2 SLA and Voicemail

This is an example of how you could set up a single voicemail box for the phone system. The voicemail box number used in this example is 1234, which would be configured in voicemail.conf.

For this example, assume that there are 2 trunks and 3 stations. The trunks are Zap/1 and Zap/2. The stations are SIP/station1, SIP/station2, and SIP/station3.

In zapata.conf, channel 1 has context=line1 and channel 2 has context=line2.

In sip.conf, all three stations are configured with context=sla_stations.

When the stations pick up their phones to dial, they are allowed to dial NANP numbers for outbound calls, or 8500 for checking voicemail.

sla.conf:

```
[line1]
type=trunk
device=Local/disa@line1_outbound
```

```
[line2]
type=trunk
device=Local/disa@line2_outbound
```

```
[station](!)
type=station
trunk=line1
trunk=line2
```

```
[station1](station)
device=SIP/station1
```

```
[station2](station)
device=SIP/station2
```

```
[station3](station)
device=SIP/station3
```

extensions.conf:

```
[macro-slaline]
exten => s,1,SLATrunk(${ARG1})
exten => s,n,Goto(s-${SLATRUNK_STATUS}|1)
exten => s-FAILURE,1,VoiceMail(1234|u)
exten => s-UNANSWERED,1,VoiceMail(1234|u)
```

```
[line1]
exten => s,1,Macro(slaline|line1)
```

```
[line2]
```

```

exten => s,2,Macro(slaline|line2)

[line1_outbound]
exten => disa,1,Disa(no-password|line1_outbound)
exten => _1NXXNXXXXXX,1,Dial(Zap/1/${EXTEN})
exten => 8500,1,VoicemailMain(1234)

[line2_outbound]
exten => disa,1,Disa(no-password|line2_outbound)
exten => _1NXXNXXXXXX,1,Dial(Zap/2/${EXTEN})
exten => 8500,1,VoicemailMain(1234)

[sla_stations]

exten => station1,1,SLAStation(station1)
exten => station1_line1,hint,SLA:station1_line1
exten => station1_line1,1,SLAStation(station1_line1)
exten => station1_line2,hint,SLA:station1_line2
exten => station1_line2,1,SLAStation(station1_line2)

exten => station2,1,SLAStation(station2)
exten => station2_line1,hint,SLA:station2_line1
exten => station2_line1,1,SLAStation(station2_line1)
exten => station2_line2,hint,SLA:station2_line2
exten => station2_line2,1,SLAStation(station2_line2)

exten => station3,1,SLAStation(station3)
exten => station3_line1,hint,SLA:station3_line1
exten => station3_line1,1,SLAStation(station3_line1)
exten => station3_line2,hint,SLA:station3_line2
exten => station3_line2,1,SLAStation(station3_line2)

```


4 Call Handling

4.1 Summary

This section is intended to describe how Asterisk handles calls inside of the SLA system so that it is clear what behavior is expected.

4.2 Station goes off hook (not ringing)

When a station goes off hook, it should initiate a call to Asterisk with the extension that indicates that the phone went off hook without specifying a specific line. In the examples in this document, for the station named "station1", this extension is simply named, "station1".

Asterisk will attempt to connect this station to the first available trunk that is not in use. Asterisk will check the trunks in the order that they were specified in the station entry in sla.conf. If all trunks are in use, the call will be denied.

If Asterisk is able to acquire an idle trunk for this station, then trunk is connected to the station and the station will hear dialtone. The station can then proceed to dial a number to call. As soon as a trunk is acquired, all appearances of this line on stations will show that the line is in use.

4.3 Station goes off hook (ringing)

When a station goes off hook while it is ringing, it should simply answer the call that had been initiated to it to make it ring. Once the station has answered, Asterisk will figure out which trunk to connect it to. It will connect it to the highest priority trunk that is currently ringing. Trunk priority is determined by the order that the trunks are listed in the station entry in sla.conf.

4.4 Line button on a station is pressed

When a line button is pressed on a station, the station should initiate a call to Asterisk with the extension that indicates which line button was pressed. In the examples given in this document, for a station named "station1" and a trunk named "line1", the extension would be "station1_line1".

If the specified trunk is not in use, then the station will be connected to it and will hear dialtone. All appearances of this trunk will then show that it is now in use.

If the specified trunk is on hold by this station, then this station will be reconnected to the trunk. The line appearance for this trunk on this station will now show in use. If this was the only station that had the call on hold, then all appearances of this trunk will now show that it is in use. Otherwise, all stations that are not currently connected to this trunk will show it on hold.

If the specified trunk is on hold by a different station, then this station will be connected to the trunk only if the trunk itself and the station(s) that have it on hold do not have private hold enabled. If connected, the appearance of this trunk on this station will then show in use. All stations that are not currently connected to this trunk will show it on hold.