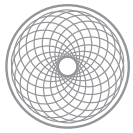
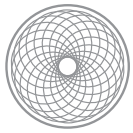


Bicom Systems

Provisioning Server proposal

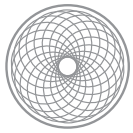
Prepared for: The Company
Prepared by: Sergej Kasumovic

2 May 2008
Proposal number: 1.0



Bicom Systems

Table of Contents



Bicom Systems

Introduction

This document is a proposal for new “Provisioning server”. Current implementation in PBXware assumes that device (be it phone, ATA, soft phone) is equal to a particular extension. The problem arises when one wants to have multiple extensions configured on one device or in even worse case, multiple extensions from multiple servers (be it PBXware, or 3rd party ones) configured on one device.

To complicate the matter even further, each device can have Phone Directory, BLF (Busy Lamp Field) lists and possibly other items customized, which in current design where extension is equal to device is not possible.

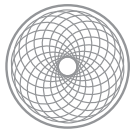
Provisioning server is meant to solve all of above problems as well as provide the full platform for provisioning of all the missing parts - such as firmwares, general device settings, different types of provisioning methods et cetera.

Implementation

Provisioning server is to be implemented as an additional application in the Sitemanager interface and it will be used as a central provisioning point by not only one, but multiple PBXware systems.

The whole server consist of:

- Web GUI
- Four different provisioning protocols:
  - FTP
  - HTTP
  - HTTPS
  - TFTP
- And possibly an LDAP server



Bicom Systems

Web GUI

Web GUI is the central and most important part of the system. Its navigation on the left side, consists of the following:

- Devices
- Device models
  - Polycom
  - Sipura/Linksys
  - Grandstream
  - Aastra
  - Cisco
  - ...
- PBXware servers

Devices

The devices list is a list of all relevant devices to be provisioned on the system. Each device has a unique ID - MAC Address. MAC address, as it is explained fully on Wikipedia page[1], consists of 12 hexadecimal characters (excluding the possible separation character ":") in which first 6 characters (or 3 bytes) represent a unique organization identifier.

Every device can have the following configured per MAC address:

- Lines (Caller ID, Extension, Registration server, Outbound proxy server, Username, Password)
- Directory list (if supported)
- BLF list (if supported)
- Other device model-specific configuration settings (codecs, SIP settings etc)

Devices list can be filtered by MAC address and possibly other relevant details (PBXware server), as well as displayed as a limited list as we assume there will be a lots of devices to be provisioned.

[1] http://en.wikipedia.org/wiki/MAC_address

Lines

Each device model is different. Lines range is usually from 1 to 12 (at maximum). Practice has shown there is a requirement that each line is potentially registered to a different server (be it PBXware or 3rd party one).

Idea is to split lines into two modes:

- a link with a particular PBXware server (defined under PBXware servers menu) and
- a direct input for a 3rd party server

In case of a direct link with PBXware server, one would configure two fields only: PBXware server ID and and an extension it is linked to. The idea is that each PBXware sends information to the Provisioning server and that it can only be stored for a particular devices and lines assigned for that server/extension. This bit is still a bit 'rough' and it is open for discussion, but the point of central storage for Provisioning server is the safest way to go (even if there are slight delays involved).

As for the 3rd party servers - since there is no control of such servers, the only way is to input the configuration bits directly (usernames, passwords, IP addresses).

The important bit about lines is to have a clear understanding of which line belongs to which server and extension. That is certainly not the case at this moment.

Directory List

Some device models are able to download directories either per MAC address or as an global directory list for all such device types. We will concentrate on the directory list per particular MAC address here. Since one device could be linked to multiple PBXware servers, it means that the list has to be populated with either one or all of them.

Either way, each PBXware server must send such information to the Provisioning server if 'Directory' option is turned on for a particular extension. That means that the Provisioning server will also hold a copy of all available extensions on a particular PBXware server.

There also has to be a way to manually input additional extensions in Directory (as SIP contacts), as well as a way to exclude extensions from Directory. Adåministrator of Provisioning server is one to decide upon such action.

Another option that is to be discussed is the availability of LDAP server. Some device models can fetch directories over LDAP and it is usually a global option (configured for all devices at once).

BLF List

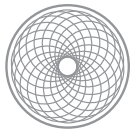
BLF list is sometimes separated from the Directory list, and sometimes it appears as a separate provisioning option. Either way data is to be pulled out from the hot copy of Directory list for linked PBXware servers, or manually inputted one.

For those who don't understand what BLF list is all about - it usually is not a part of the device itself, but rather an additional pluggable dock that can display extension statuses (on phone, not registered etc).

Other configuration

Some devices support additional configuration per each line. That configuration can include different codecs, NAT and SIP options etc. This is device specific and is to be displayed according to device model.

If device model does not support ‘General settings’ configuration (look under ‘Device models’ section), it is worth looking into option where all such ‘other configuration’ could be administered from the ‘General settings’ under ‘Device models’ section and then linked to relevant devices. That would avoid duplicate work if they all share the same settings.



Bicom Systems

Device models

Device models is another important part of the interface. Each device model, apart from MAC address-specific configuration consist of global configuration and default factory configuration for all devices. Possible list would include the most used ones: Polycom, Cisco, Sipura, Grandstream etc.

Each device model consist of several different provisioning objects:

- General settings
- Firmware
- Global directory list
- List of additional files available for download

General settings

Configuration for this varies from model to model. Some device models do not have such configuration (Grandstream), some rely on it heavily (Polycom). The important bit about global settings is that it influences provisioning at a great level. Time zone, Daylight saving time, SIP/NAT settings, Directory settings, Volume settings, Ringtones are among the ones that are affected.

It is important that all such options are presented in visually appearing and understandable fashion. This at some point will mean that there will have to be a great deal of work with device models that use cryptic names (Linksys and Grandstream are the affected ones). Examples would be P23489, 32489 or similar.

Another problem is that options change with firmware and some device models (such as Polycom), allow the configuration to be firmware specific.

My proposal is to implement 'General settings' for the Polycom first and to follow the document[2] which is available for Download from the Polycom page.

It's aim is to provide multiple 'General settings' as defined in above document, with nice and clean interface which would produce a static XML configuration as an end result.

Firmware

Each device model must be able to download relevant firmware. Firmware size depends on device and best way to upload is probably via some kind of FTP access (since we will provide FTP provisioning protocol anyway).

Some device models (like Polycom) download additional files which define which firmware is to be downloaded. Such administration is to be possible from this section.

[2] http://www.polycom.com/common/documents/support/setup_maintenance/products/voice/SoundPointIP_SoundStationIP_AdminGuide_SIP3_0_Eng_Rev_A.pdf

Global Directory List

Some device models (notably Polycom) have support for global directory. Due to the fact that many devices will connect from different PBXware, it is relatively questionable if this should be used in such instances.

However, this does have an impact if everyone is supposed to share a single directory. LDAP again pops into mind, however this is to be discussed.

Additional files

Ringtones, Images, Fonts are among the files that specific device models can download as well. All of these files are to be uploaded via FTP or similar mechanism and they will be listed here. We assume that the provisioning server will separate files in directories for each particular device model. Those can then be symbolically linked to the root of publicly accessible provisioning directory.

PBXware servers

PBXware servers are not to be discussed in great detail at this moment. Essentially, it is a list of valid PBXware servers that can connect to the Provisioning server with specific, randomly chosen, username and password.

This will make sure only authorized servers can push the data to the Provisioning server.

Provisioning protocols

Four different protocols are available for provisioning:

- FTP
- HTTP
- HTTPS
- TFTP

However, only one of them can be served from a DHCP server. We will assume that all devices will come to provisioning server, by being supplied a DHCP option 66 or relevant as part of their boot-up/DHCP request process.

There is another method that can be used apart from the DHCP one. Some devices (namely Polycom, Cisco) support CDP (Cisco Discovery Protocol)[3], which can be used as a daemon[4] on the Linux platform to advertise a specific Voice over IP network together with a relevant DHCP server. However, this method is rather complicated and it involves having switches with VLAN tagging support. It is to be discussed though, whether this variant should be supported.

Once devices hits the provisioning server, data can be either server dynamically or statically. This also means we will have to look into our own TFTP/FTP implementation.

Let us discuss the difference between dynamic and static method of file serving.

[3] http://en.wikipedia.org/wiki/Cisco_Discovery_Protocol

[4] http://lcdpd.sourceforge.net/

Dynamic file serving

Dynamic file serving is basically a solution based on mod_rewrite[5] functionality of Apache/lighttpd. Such functionality provides one to link all requests to a certain script, which will upon request validation, decide what kind of output should be served to the end user device.

What this means to us is that we can dynamically generate MAC address-specific configuration at the request time, rather than making sure it is re-generated every time certain change is applied (this is how it works in the PBXware at the moment). This however comes with a certain performance hit, but MAC-specific files are usually small in size.

Some kind of caching mechanism is to be implemented to avoid constant document generation from scratch.

Static file serving

Static file serving, on the other hand, is meant to be used for large, heavy files. Those files include firmwares, ringtones, fonts as well as general configuration for devices (such as Polycom one, which as an end result can be in range of couple hundred of kilobytes).

These files are either not changed that much or are strict binary files and they fit into this particular purpose nicely.

[5] http://trac.lighttpd.net/trac/wiki/Docs:ModRewrite

Conclusion

My proposal is to try to apply all above to one particular manufacturer and to implement as much options as possible. This however required direct access to all such devices in the office for present and future testing. However, we must focus on being as ‘general purpose’ as possible, in order to support other manufacturers as well.

Provisioning Diagram Sample