



## Technical Document

Auto Provisioning and Customization for  
IP Phone ST2030 and ST2022  
ST20XX-AutoProvisioning-V0032

November 16, 2009

ASKEY CONFIDENTIAL

# TABLE OF CONTENTS

<b>1</b>	<b>SYSTEM DESCRIPTION.....</b>	<b>3</b>
1.1	PURPOSE.....	3
1.2	INTRODUCTION.....	3
<b>2</b>	<b>AUTO PROVISIONING PROCESS.....</b>	<b>4</b>
2.1	PREREQUISITES.....	4
2.2	SYSTEM FLOW.....	4
2.3	AUTO PROVISIONING VIA PRE-CONFIGURED HTTP OR TFTP SERVER ADDRESS.....	5
2.4	AUTO PROVISIONING VIA DHCP + TFTP.....	5
2.4.1.	<i>Message Flow of SIP Phone</i> .....	6
2.4.2.	<i>Message Flow of MGCP Phone</i> .....	10
2.4.3.	<i>Syntax of Provisioning INF(TFTP supports absolute or relative path)</i> .....	15
2.4.4.	<i>Example of Provisioning Information File</i> .....	17
2.5	AUTO PROVISIONING VIA DHCP + HTTP.....	17
2.5.1.	<i>Message Flow of SIP Phone</i> .....	18
2.5.2.	<i>Message Flow of MGCP Phone</i> .....	21
2.5.3.	<i>Syntax of Provisioning INF (HTTP support relative path only)</i> .....	26
2.5.4.	<i>Example of Provisioning Information File</i> .....	29
2.6	LCD DISPLAY DURING PROVISIONING.....	30
2.7	SYNTAX OF CONFIGURATION FILE.....	30
<b>3</b>	<b>UPGRADE TRIGGERS.....</b>	<b>31</b>
3.1	INITIAL UPGRADE.....	31
3.2	PERIODICAL UPGRADES.....	31
3.3	UPGRADE TIMING.....	31
3.4	SIP NOTIFY (SIP ONLY).....	32
3.5	SECURITY CONSIDERATION.....	33
<b>4</b>	<b>VLAN PROVISION.....</b>	<b>34</b>
4.1	OVERVIEW.....	34
4.2	SUPPORT OF “FILE” FIELD IN THE DHCP HEADER FOR APS.....	34
4.3	CONFIGURING VLAN IDS VIA DHCP OPTIONS.....	34
4.4	DESCRIPTION.....	34
4.4.1.	<i>Preconditions:</i> .....	34
4.4.2.	<i>Parameters:</i> .....	34
4.4.3.	<i>DHCP options:</i> .....	35
4.5	SYSTEM FLOW:.....	35
4.5.1.	<i>PersistentVLANfromDHCP=0</i> .....	35
	INTERACTION WITH DHCP-BASED AND FILE-BASED VLAN PROVISIONING.....	35
4.5.2.	<i>PersistentVLANfromDHCP=1</i> .....	35
<b>5</b>	<b>SIP/MGCP SWAP MECHANISM.....</b>	<b>37</b>
<b>6</b>	<b>SUPPORT ENCRYPTION OF CONFIGURATION FILES IN APS.....</b>	<b>38</b>
	FEATURE ACTIVATION.....	39
A.	VIA WEBGUI.....	39
B.	VIA APS.....	40

# 1 System Description

## 1.1 Purpose

This document describes the Auto Provisioning feature of both ST2030 and ST2022 SIP (SIP phone for short) and ST2030/ST2022 MGCP (MGCP phone for short). ST20XX will refer to both ST2030 and ST2022.

## 1.2 Introduction

The Auto Provisioning feature is an auto-installation and upgrading system supported by the SIP and MGCP phones.

This system supports the following features:

- Upgrade firmware automatically, if newer FW available on server.
- Upgrade XML files for LCD display automatically, if newer files available on server (for MGCP only).
- Download melody, call waiting tone and system melody files automatically, if newer melody text files available on server (call waiting and system melody files are for SIP model).
- Support VLAN provision
- Download configuration file automatically, if configuration updated on server.

The difference between SIP and MGCP phones is that SIP phone does not need to download XML files.

The phone triggers the process by itself in the following conditions:

1. Every time the phone powers up, whether it is a warm start or a cold start.
2. Every programmable time frame if NTP available.
3. Upon reception of a "reboot" message (available with EMS only).
4. Upon reception of a specific SIP NOTIFY (see 3.4)

Three kinds of auto provisioning are supported:

- Based on DHCP and TFTP
- Based on HTTP

If the phone is in communication (off hook, ringing, voice conversation...), auto-provisioning will be launched only after the end of the communication.

In addition to auto provisioning, Customization is an efficient procedure allowing a worker to customize the phone in the warehouse or factory, before shipment to the customer.

## 2 Auto Provisioning Process

### 2.1 Prerequisites

Some prerequisites are defined as following.

1. Each firmware release has a different filename or is located at a different URL.
2. In MGCP, every time when the phone triggers this process will download the deck files.
3. There is a general configuration file with common parameters, shared by all phones. If general configuration file has changed, the filename must be changed as well. In addition, each phone also has its own configuration file which filename is the mac address preceded by the phone model "ST2030S\_" for SIP or "ST2030M\_" for MGCP, for example ST2030S\_009096A1B1C1.txt. ("ST2022S\_" or ST2022M\_" for the corresponding model). Each time the administrator modifies the configuration file, he must also change the series number (config\_sn) in this file used by phone to identify if the configuration has been updated in the server. Each MAC-specific config file has its own series number (config\_sn).
4. Provisioning Information file, which contains the locations of firmware, deck file and common config file, will be downloaded in the beginning.

### 2.2 System Flow

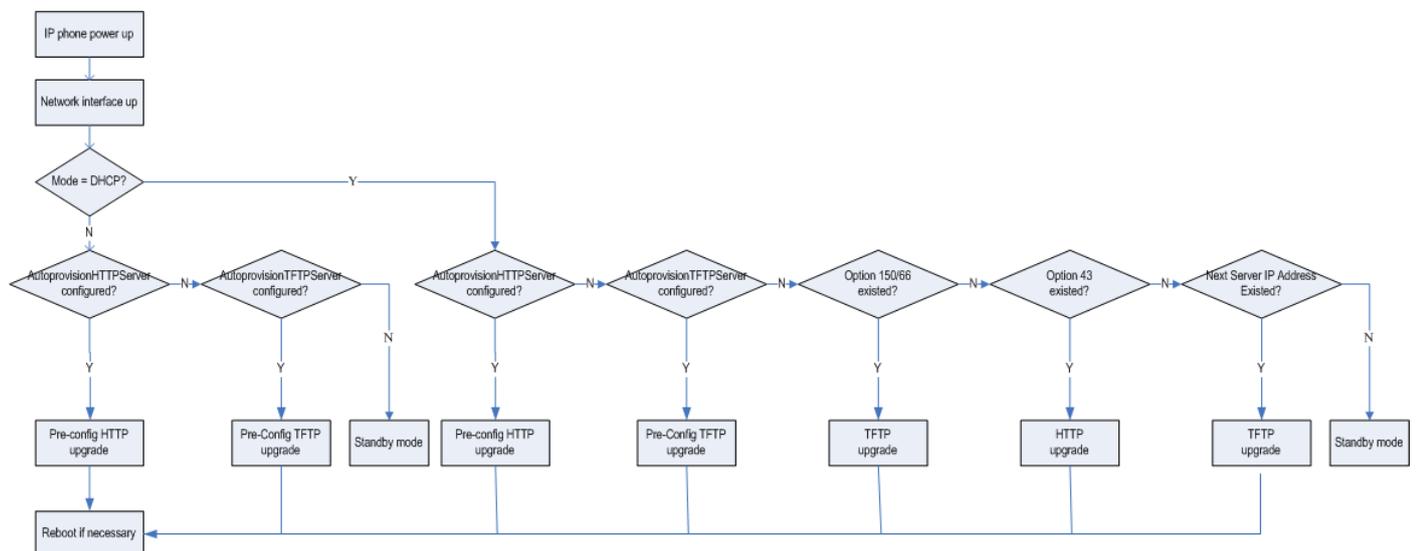


Figure 1 System Flow of Provisioning

Here is the summary of INF file order for both ST20XX(S) and ST20XX(M):

Take ST20XX(M) for example,

- Perconfigured HTTP/TFTP: The first priority is ST20XXM\_<mac\_address>.inf. If ST20XX(M) can't find this file on the server, it will look for autoprovisionconfigname instead. If both files can't be found on the server, ST20XX(M) will go back to idle mode with error message display.
- Option 150/66: The first priority is option 67.If ST20XX(M) can't find option 67 in DHCP server, it will look for ST20XXM\_<mac\_address>.inf. If this file still can't be found on the server, it will look for autoprovisionconfigname. If none of these can't be found on the server, ST20XX(M) will go back to idle mode with error message display.
- Option 43:The first priority is the file in HTTP server. If this file can't be found, ST20XX(M) will look for autoprovisionconfigname instead. If both files can't be found on the server, ST20XX(M) will go back to idle mode with error message display.
- Next Server IP Address: The first priority is option 67.If ST20XX(M) can't find option 67 in DHCP server, it will look for ST20XXM\_<mac\_address>.inf. If this file still can't be found on the server, it will look for autoprovisionconfigname. If none of these can't be found on the server, ST20XX(M) will go back to idle mode with error message display.

### **2.3 Auto Provisioning via pre-configured HTTP or TFTP Server address**

This method relies on pre-configured parameters in ST20XX. If "AutoprovisionHTTPServer" is pre-configured, ST20XX will perform HTTP upgrade using this IP address. If "AutoprovisionTFTPServer" is pre-configured, ST20XX will perform TFTP upgrade using this IP address. When both parameters are configured, HTTP will be chosen. This method has higher priority than any other APS mechanism. For the INF file, if "AutoprovisionConfigname" is pre-configured, ST20XX will look for the specific file, otherwise it will see option 67 and default INF file.

### **2.4 Auto Provisioning via DHCP + TFTP**

This method relies on TFTP for file transmission, the network addressing of phone is limited to DHCP. The DHCP server must support option 66/ 67 or "Next Server IP Address".

The Provisioning INF will be retrieved according to the information in DHCP option 66 (or 150 if present) and 67.

Option 66: The TFTP server IP address for upgrading

Option 150: The TFTP server IP address for upgrading. Used instead of option 66 if present. Either ip address or hexadecimal with 4 Byte can be accepted.

Option 67: The configuration file name in TFTP server for upgrading .Its format should only contain numeric, character and underscore. As for the maximum length is 255 characters. If option 67 is not present, but option 66 or 150 (TFTP server address) are there, then the phone will send an TFTP request to look for INF file whose name is based on IPP's MAC address with inf extension, say STST20XXS\_009096aabbcc.inf or STST20XXM\_009096aabbcc.inf. Based on the INF file content, Autoprovision will be performed. The file name of this INF is configurable.

Besides, in order to allow different models within the same network, the phone must differentiate itself from other phones. This way, the configuration file name proposed by the TFTP server can be related to a phone model. That is why the phone supports options 60 (vendor class identifier).

- Option 60 is sent in DHCP DISCOVER and REQUEST messages. Its format is as follows:

Description	Code	Length (bytes)	Note	Value
Manufacturer ID	1	7 (may vary)	Phone vendor code	Thomson
Model ID	2	6 (may vary)	Phone model	ST2030, ST2022,...
Protocol ID	3	3 (may vary)	Protocol	SIP, MGCP,...

The format of this option follows chapter 9.13 of RFC 2132:

```

Code  Len  Vendor-specific information
+-----+-----+-----+-----+-----+
| 60 |  n | i1 | i2 | ...
+-----+-----+-----+-----+

```

Each vendor-specific information field is coded as code/length/value. Since several servers have been found which cannot recognize non-string characters in this option, codes and lengths are now expressed in string format

An example is:

6	0	3	3	'01'	'07'	'T'	'h'	'o'	'm'	's'	'o'	'n'	'02'	'06'	'S'	'T'	'2'	'0'
'3'	'0'	'03'	'03'	'S'	'I'	'P'												

However if the DHCP server does not support option 60 formatted like this, it will just ignore it. It is then the responsibility of the network administrator to make sure only one model of phone is present, or all models can share the same configuration file (in case of configuration update only), or option 77 is used instead to recognize the type of phone.

Option 77: User Class Identifier. Phone identifies itself in case Option 60 is not recognized by the server. The content depends on model type and protocol and will be:

Thomson <ST2030S|ST2030M|ST2022S|ST2022M>

For example, for a ST2022SIP it would be

Thomson ST2022S

### 2.4.1. Message Flow of SIP Phone

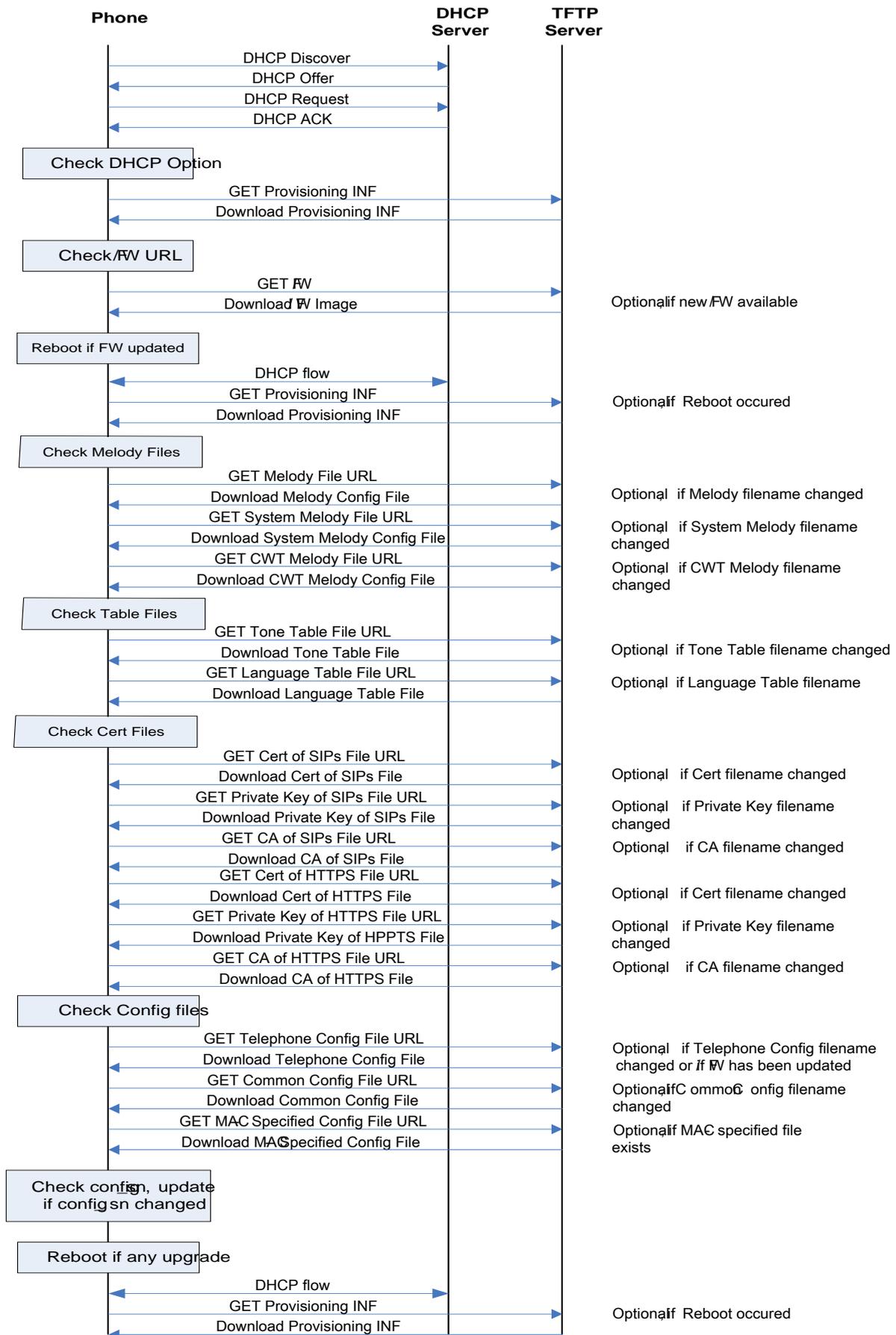


Figure 2 Message Flow of SIP Phone based on DHCP and TFTP

1. Download Provisioning INF from the TFTP server address and filename described in DHCP options.

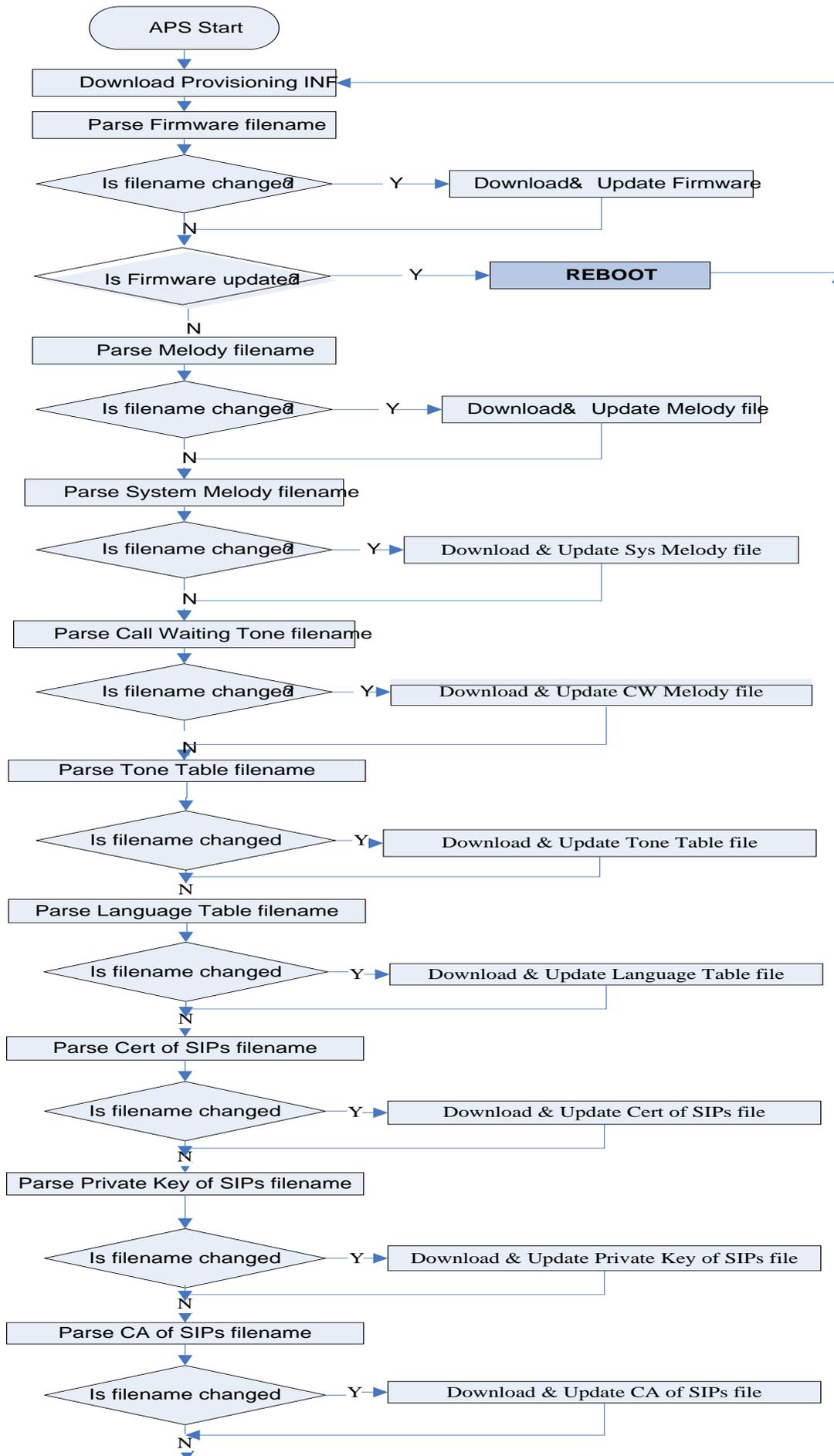
This file contains the following information (Details refers to section 2.4.3 for SIP)

- Provision mode
- Firmware filename
- Melody file name
- System Melody filename
- CWT Melody filename
- Tone Table filename
- Language Table filename
- Cert of SIPs filename
- Private Key of SIPs filename
- CA of SIPs filename
- Cert of https filename
- Private Key of https filename
- CA of https filename
- Telephone config filename
- Common config filename

Refer to section 2.4.37 for details.

2. Download firmware if firmware filename differs from last upgraded filename. This includes FW downgrading. If the new FW supports new config parameters, these new config parameters will be in the following config files (common and/or MAC-specific). They will be taken into consideration after reboot.
3. Download melody file from TFTP server if the filename is different from the previous.
4. Download system melody file from TFTP server if the filename is different from the previous.
5. Download CWT melody file from TFTP server if the filename is different from the previous.
6. Download Tone Table file from TFTP server if the filename is different from the previous.
7. Download Language Table file from TFTP server if the filename is different from the previous.
8. Download Cert of SIPs file from TFTP server if the filename is different from the previous.
9. Download Private Key of SIPs file from TFTP server if the filename is different from the previous.
10. Download CA of SIPs file from TFTP server if the filename is different from the previous.
11. Download Cert of HTTPs file from TFTP server if the filename is different from the previous.
12. Download Private Key of HTTPs file from TFTP server if the filename is different from the previous.
13. Download CA of HTTPs file from TFTP server if the filename is different from the previous.
14. Download telephone config file from TFTP server if the filename is different from the previous or if the firmware has been updated.
15. Download common config file from TFTP server if the filename is different from the previous.
16. Download MAC-specific config file from TFTP server, if it has been provided on the server. The filename is IPP's MAC address preceded by the phone model "STST20XXS\_", appended by phone automatically, for example STST20XXS\_009096A1B1C1.txt
17. Update common configuration. Update MAC-specific configuration if config\_sn differs from current one.
18. Reboot if any upgrade was successful.

Common and MAC-specific can both contain all possible parameters. As a consequence, they may provide some identical parameters. If this situation arises, the value given in MAC-specific config file will overwrite the value from the common config file.



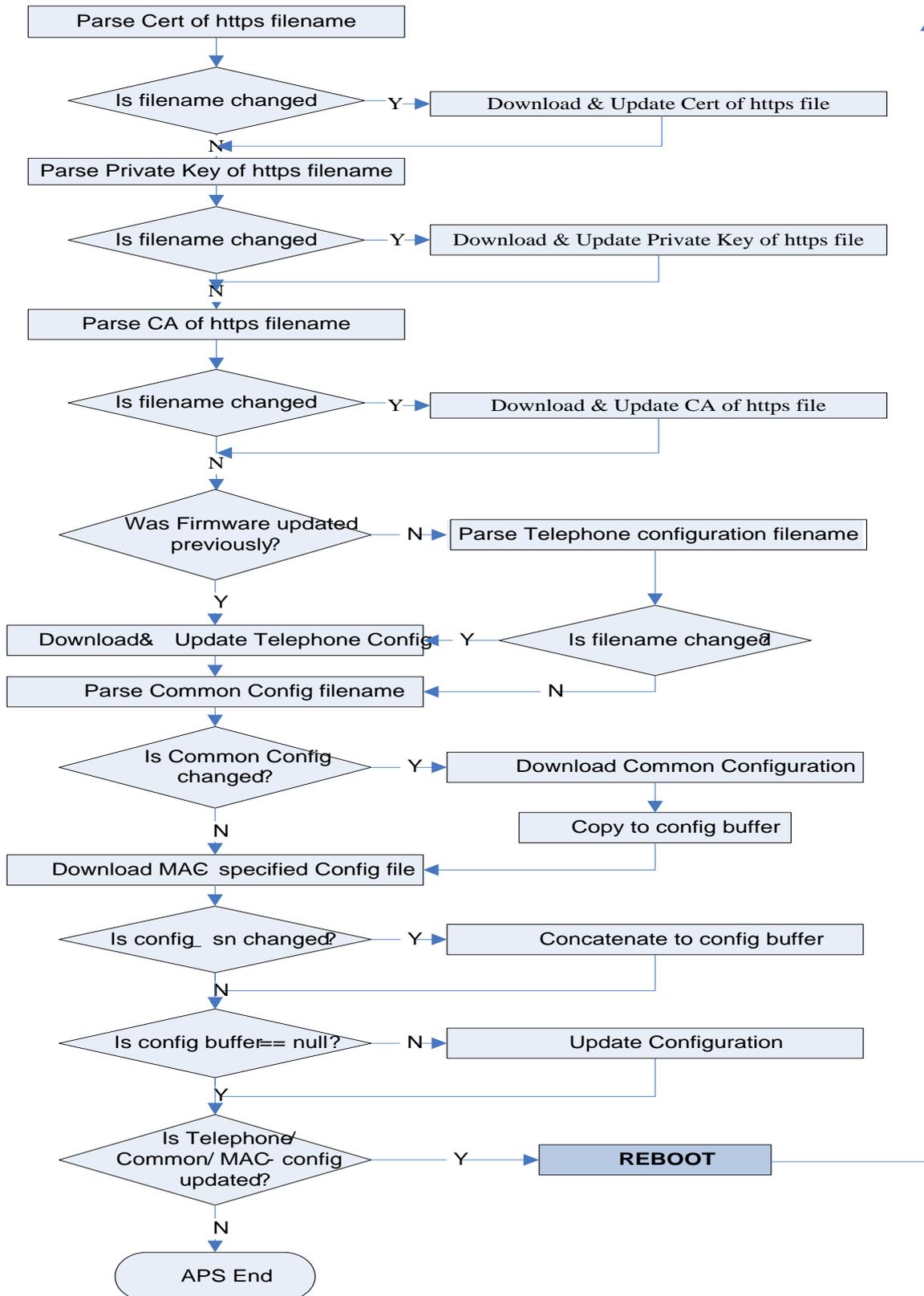


Figure 3 Version checking of SIP Auto provisioning via DHCP + TFTP

2.4.2. Message Flow of MGCP Phone

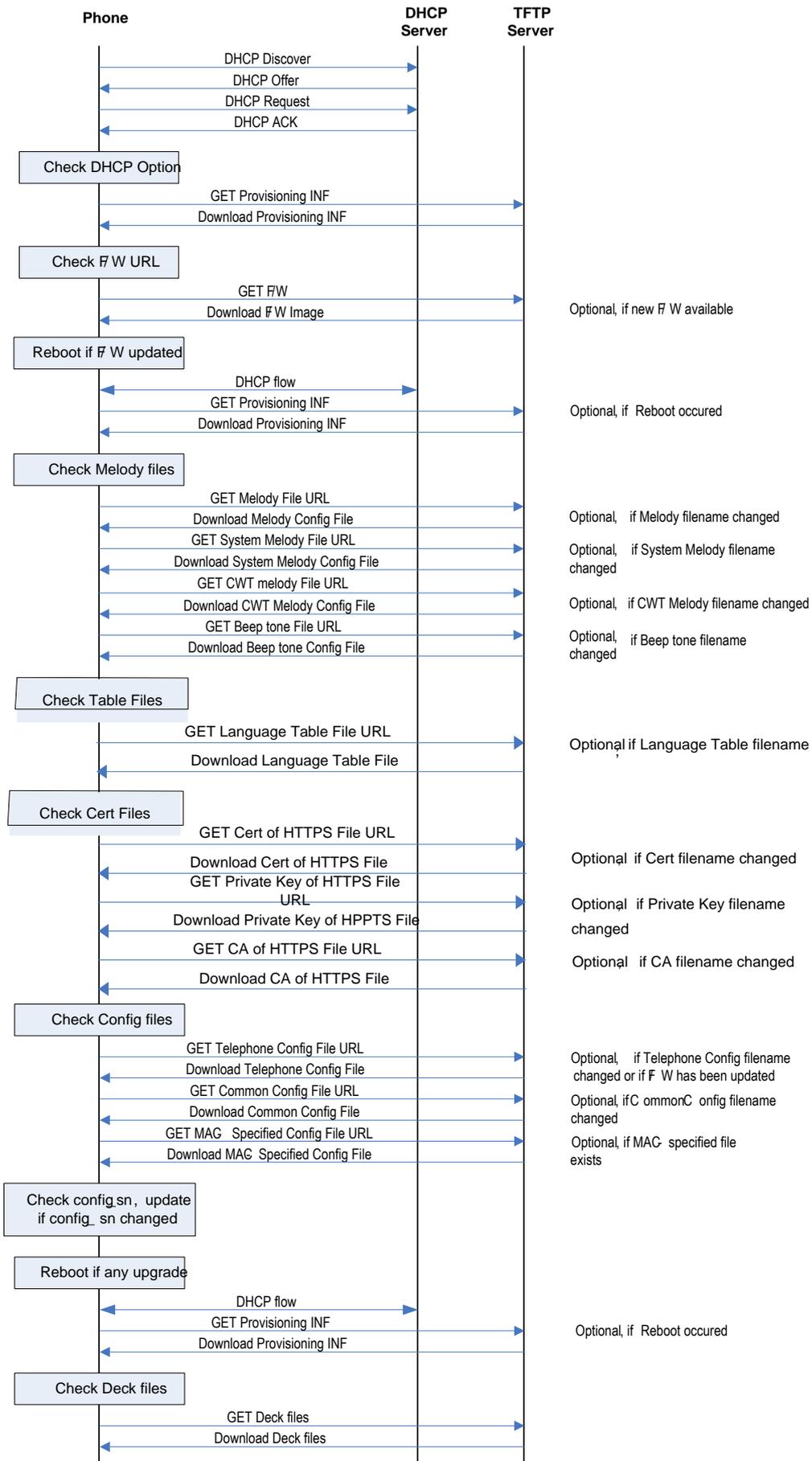


Figure 4 Message Flow of MGCP Phone based on DHCP and TFTP

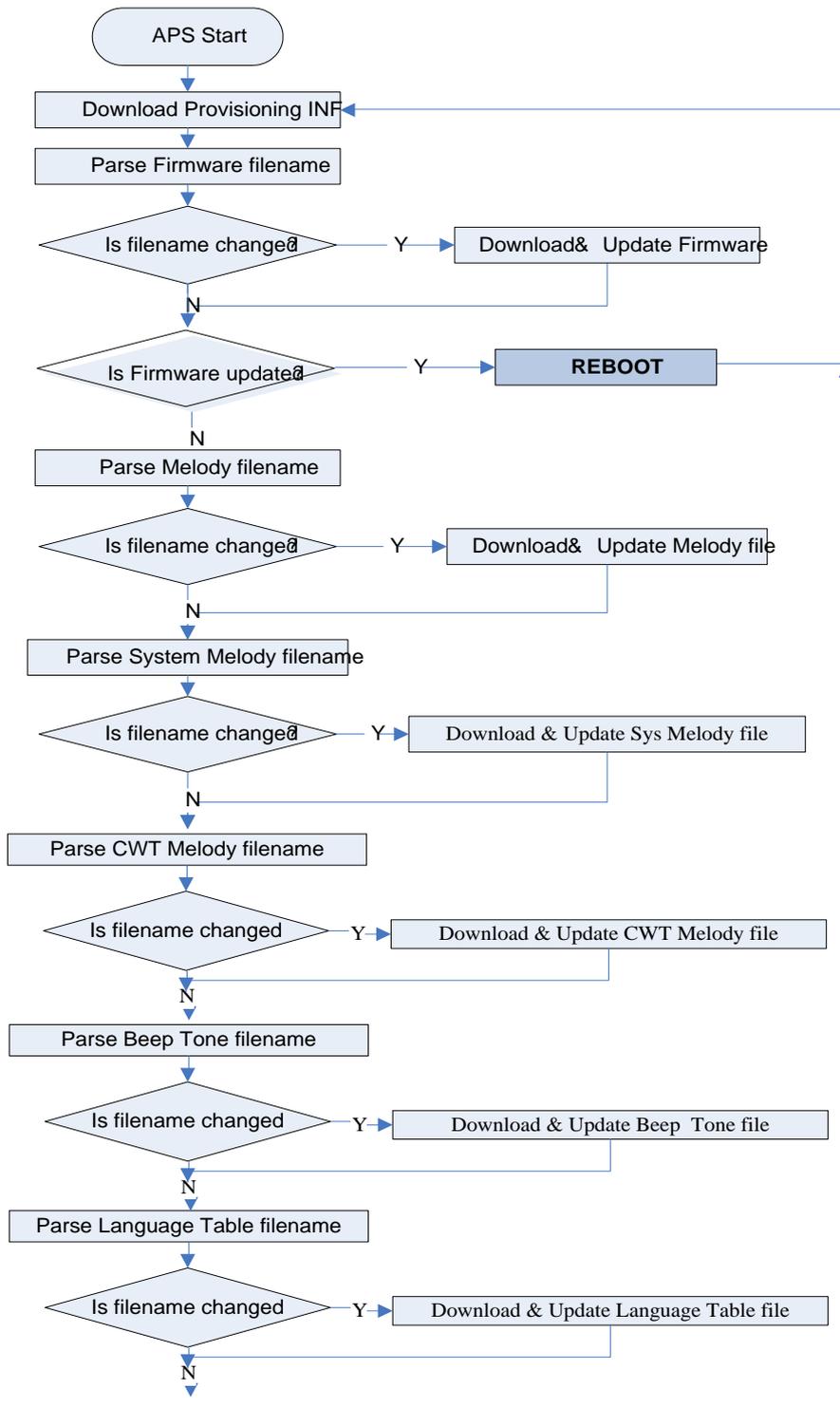
1. Download Provisioning INF from the TFTP server address and filename described in DHCP options.

This file contains the following information (Details refers to section 2.4.3)

- Provision mode
- Firmware filename
- Deck File filename
- Melody file filename
- System melody file filename
- CWT melody file filename
- Beep Tone file filename
- Language Table filename
- Cert of HTTPs filename
- Private Key of HTTPs filename
- CA of HTTPs filename
- Telephone config filename
- Common config filename

Refer to section 2.4.3 for details.

2. Download firmware if firmware filename differs from last upgraded filename. This includes FW downgrading. If the new FW supports new config parameters, these new config parameters will be in the following config files (common and/or MAC-specific). Reboot if the firmware has been updated.
3. Download melody file from TFTP server if the filename is different from the previous.
4. Download system melody file from TFTP server if the filename is different from the previous.
5. Download CWT melody file from TFTP server if the filename is different from the previous.
6. Download Beep Tone file from TFTP server if the filename is different from the previous.
7. Download CWT melody file from TFTP server if the filename is different from the previous.
8. Download Language Table file from TFTP server if the filename is different from the previous.
9. Download Cert of HTTPs file from TFTP server if the filename is different from the previous.
10. Download Private Key of HTTPs file from TFTP server if the filename is different from the previous.
11. Download Cert of HTTPs file from TFTP server if the filename is different from the previous.
12. Download telephone config file from TFTP server if the filename is different from the previous or if the firmware has been updated.
13. Download common config file from TFTP server if the filename is different from the previous.
14. Download MAC-specific config file from TFTP server, if it has been provided on the server. The filename is IPP's MAC address preceded by the phone model "STST20XXM\_", appended by phone automatically, for example STST20XXM\_009096A1B1C1.txt
15. Update common configuration. Update MAC-specific configuration if config\_sn differs from current one.
16. Reboot if any upgrade was successful.
17. Download deck files.



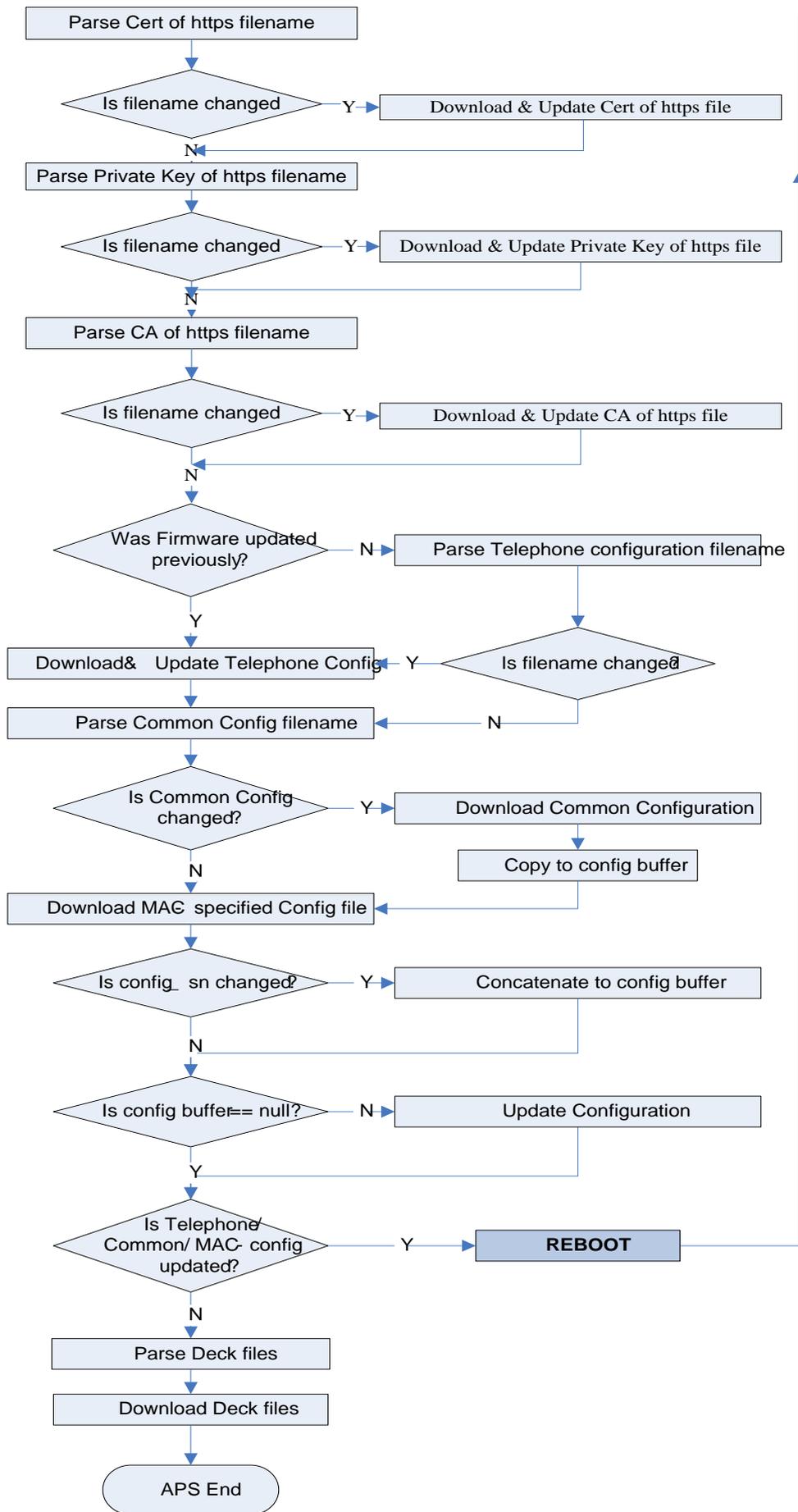


Figure 5 Version checking of MGCP Auto provisioning via DHCP + TFTP

### 2.4.3. Syntax of Provisioning INF(TFTP supports absolute or relative path)

#### ◆ [provision\_mode] Section

This section specifies if the phone is processing Auto-Provisioning or Customization.

If this section is not present, Auto-Provisioning is done by default. The default value is auto\_provisioning.

- provision\_mode

**Syntax:**

provision\_mode=auto\_provisioning|customization

**Description:**

Indicator of provision mode (auto-provisioning or customization)

**Example:**

Provision\_mode=auto\_provisioning

#### ◆ [application] Section

This section contains application firmware location.

- fw\_filename

**Syntax:**

fw\_filename=[path\filename]

**Description:**

File name of application firmware (can be an absolute or relative path)

**Parameter Range:**

Maximum to 255 characters

**Example:**

fw\_filename= C:\ip\_phones\mgcp\st2030\v100.zz

#### ◆ [deckfile] Section (For MGCP only)

This section contains location of deck file used for LCD display.

- deckfile

**Syntax:**

deckfile=[filename]

**Description:**

Filename of deck files used for LCD display (can be an absolute or relative path).

**Parameter Range:**

Maximum to 255 characters

**Example:**

deckfile= .\ip\_phones\mgcp\st2030\main.thd

#### ◆ [melody] Section

- Melody file

**Syntax:**

melodies=[filename]

**Description:**

Filename of melody files used for ringer melody (can be an absolute or relative path).

**Parameter Range:**

Maximum to 255 characters

**Example:**

melodies= .\ip\_phones\mgcp\st2030\ melody.txt

- System melody file

**Syntax:**

system\_melodies=[filename]

**Description:**

Filename of system melody files used for SIP-INFO (can be an absolute or relative path).

**Parameter Range:**

Maximum to 255 characters

**Example:**

system\_melodies= .\ip\_phones\sip\st2030\ sysmelody.txt

- CWT melody file

**Syntax:**

call\_waiting\_tone=[filename]

**Description:**

Filename of CWT melody files used for call waiting (can be an absolute or relative path).

**Parameter Range:**

Maximum to 255 characters

**Example:**

call\_waiting\_tone= .\ip\_phones\sip\st2030\CWT.txt

- Beep Tone file(for MGCP only)

**Syntax:**

beep\_tone=[filename]

**Description:**

Filename of Beep Tone files used for supervision (can be an absolute or relative path).

**Parameter Range:**

Maximum to 255 characters

**Example:**

call\_waiting\_tone= .\ip\_phones\sip\st2030\BT.txt

◆ **[config] Section**

This section contains configuration file location.

- common\_config

**Syntax:**

common\_config=[filename]

**Description:**

filename of common config file (can be an absolute or relative path)

**Parameter Range:**

Maximum to 255 characters

**Example:**

common\_config= C:\ip\_phones\mgcp\st2030\st2030s.txt

- telcfg

**Syntax:**

telcfg=[filename]

**Description:**

filename of telephone config file (can be an absolute or relative path)

**Parameter Range:**

Maximum to 255 characters

**Example:**

telcfg= .\ip\_phones\mgcp\st2030\st2030s\_v100\_cfg.txt

- telcfg

**Syntax:**

language\_table=[filename]

**Description:**

filename of language\_table file (can be an absolute or relative path)

**Parameter Range:**

Maximum to 255 characters

**Example:**

language\_table= .\ip\_phones\mgcp\st2030\LanTbl\_\_ST2030.thd

**2.4.4. Example of Provisioning Information File**

```
[application]
fw_filename= C:\ip_phones\mgcp\st2030\v100.zz

[deckfile]
deckfile=.\ip_phones\mgcp\decks\st2030\main.thd
deckfile=.\ip_phones\mgcp\decks\st2030\renvoi.thd
deckfile=.\ip_phones\mgcp\decks\st2030\config.thd
deckfile=.\ip_phones\mgcp\decks\st2030\abrege.thd
deckfile=.\ip_phones\mgcp\decks\st2030\call.thd
deckfile=.\ip_phones\mgcp\decks\st2030\log.thd
deckfile=.\ip_phones\mgcp\decks\st2030\supervision.thd
deckfile=.\ip_phones\mgcp\decks\st2030\filtrage.thd
deckfile=.\ip_phones\mgcp\decks\st2030\register.thd

[config]
common_config=st2030m.txt
telcfg=st2030m_v100_cfg.txt
melodies=ringtone_033006.txt
system_melodies=ringtone_040406.txt
call_waiting_tone=ringtone_040406CWT.txt
tone_table=ToneTable.txt
language_table=LangTable.txt
certAPSfilename=cert\001291C2DF29.crt
pkeyAPSfilename=cert\001291C2DF29.key
calistAPSfilename=cert\cacert.pem
```

(Note: tone table download is only supported for SIP, shown here to give an example only)

ST20XX will look for the Mac-specific config file in the TFTP root directory.

**2.5 Auto Provisioning via DHCP + HTTP**

This method relies on HTTP for file transmission, the network addressing of phone is not limited, supports under static IP, DHCP, or PPPoE environment.

The Provisioning INF will be retrieved according to the information in DHCP option 43.

The Provisioning Information file contains the URL of firmware, deck file (for MGCP) and configuration files. Maximum length of this URL is 255 characters.

DHCP option 43 (vendor specific information) is used for providing the URL of the Provisioning Information file. Its format follows RFC2132 chapter 8.4, containing one single field which is the URL:

```
Code  Len  Vendor-specific information
+----+-----+-----+-----+----
| 43 |  n  |  i1 |  i2 | ...
+----+-----+-----+-----+----
```

2.5.1. Message Flow of SIP Phone

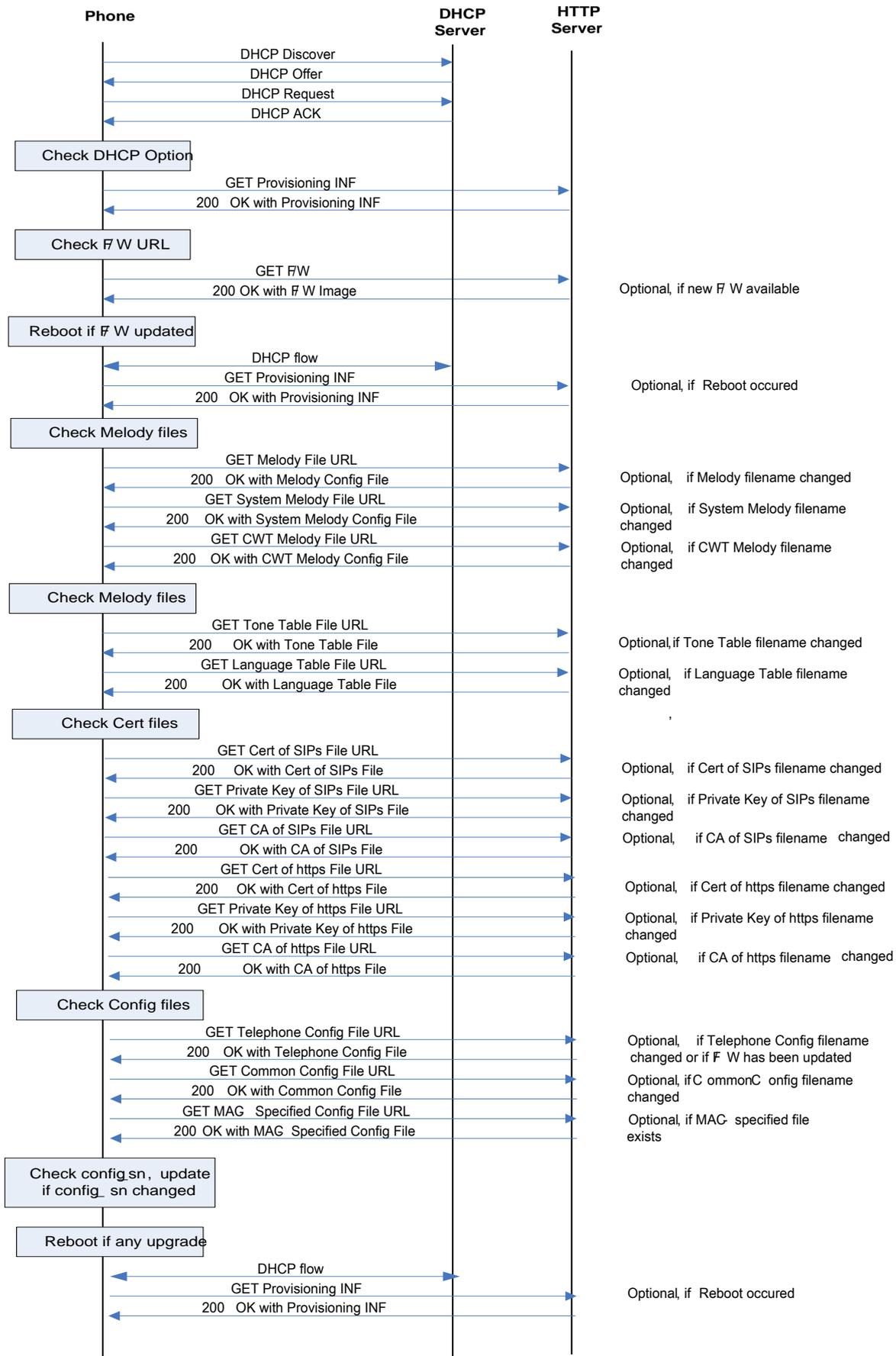


Figure 6 Message Flow of SIP Phone based on DHCP and HTTP

1. Download Provisioning INF from the HTTP server using the URL described in DHCP option 43.

This file contains the following information (Details refers to section 2.5.3)

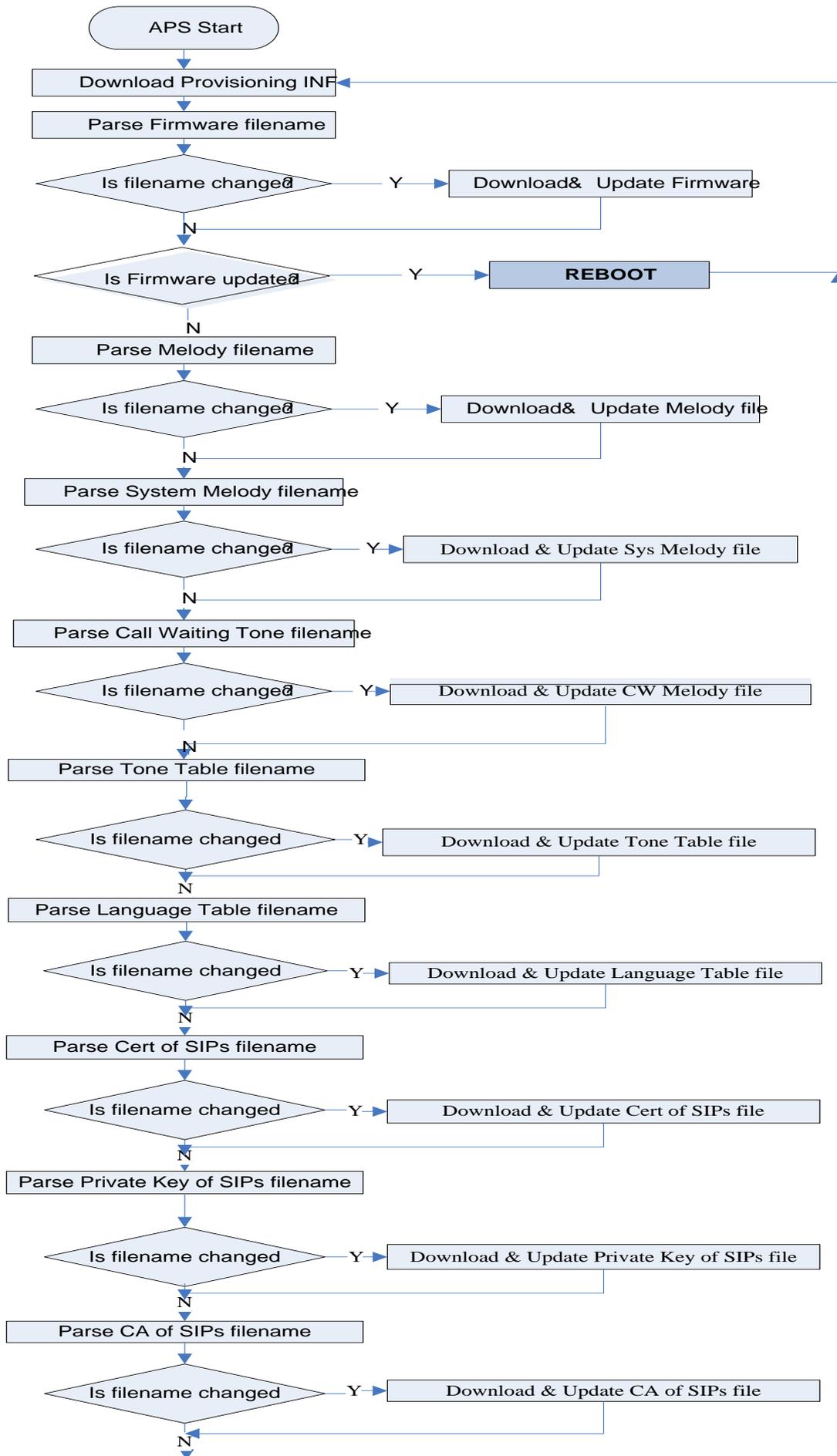
- Provision mode
- Firmware URL
- Melody URL
- System Melody URL
- CWT Melody URL
- Tone Table URL
- Language Table URL
- Private Key of SIPsURL
- CA of SIPs URL
- Cert of https URL
- Private Key of https URL
- CA of https URL
- Common config URL
- Telephone config URL
- MAC-specified config URL.

Refer to section 2.7 for details.

2. Upgrade firmware if firmware URL differs from last upgraded URL. Reboot if the firmware has been updated.
3. Download melody file from HTTP server if the filename is different from the previous.
4. Download system melody file from HTTP server if the filename is different from the previous.
5. Download CWT melody file from HTTP server if the filename is different from the previous.
6. Download Tone Table file from HTTP server if the filename is different from the previous.
7. Download Language Table file from HTTP server if the filename is different from the previous.
8. Download Cert of SIPs file from HTTP server if the filename is different from the previous.
9. Download Private Key of SIPs file from HTTP server if the filename is different from the previous.
10. Download CA of SIPs file from HTTP server if the filename is different from the previous.
11. Download Cert of https file from HTTP server if the filename is different from the previous.
12. Download Private Key of https file from HTTP server if the filename is different from the previous.
13. Download CA of https file from HTTP server if the filename is different from the previous.
14. Download telephone config file from HTTP server if the filename is different from the previous or if the firmware has been updated.
15. Download common config file from HTTP server if the filename is different from the previous.
16. Download MAC-specified config file from the MAC-specified config URL (obtained in upgrade information file) followed by config filename (mac address preceded by the phone model "ST20XXs\_", appended by phone automatically), if it has been provided on the server.

For example, if the URL in Provisioning INF is `http://upgrade.company.com/ST20XXs/config/`, phone will attempt get configuration from `http://upgrade.company.com/ST20XXs/config/STST20XXS_009096A1B1C1.txt`

17. Update common configuration. Update MAC-specific configuration if `config_sn` differs from current one.
18. Reboot if any upgrading successfully.



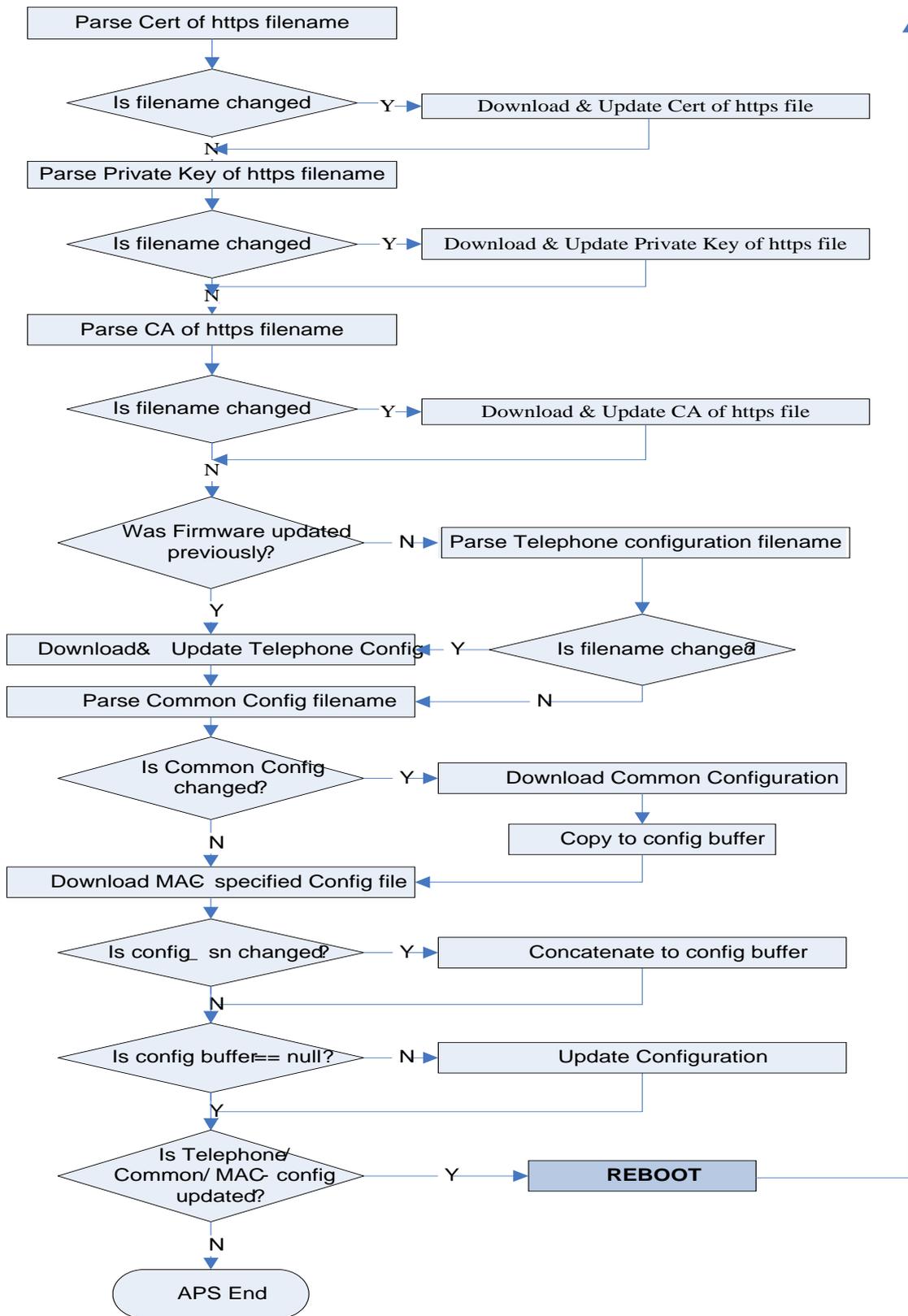


Figure 7 Version checking of SIP Auto provisioning via DHCP + HTTP

**2.5.2. Message Flow of MGCP Phone**

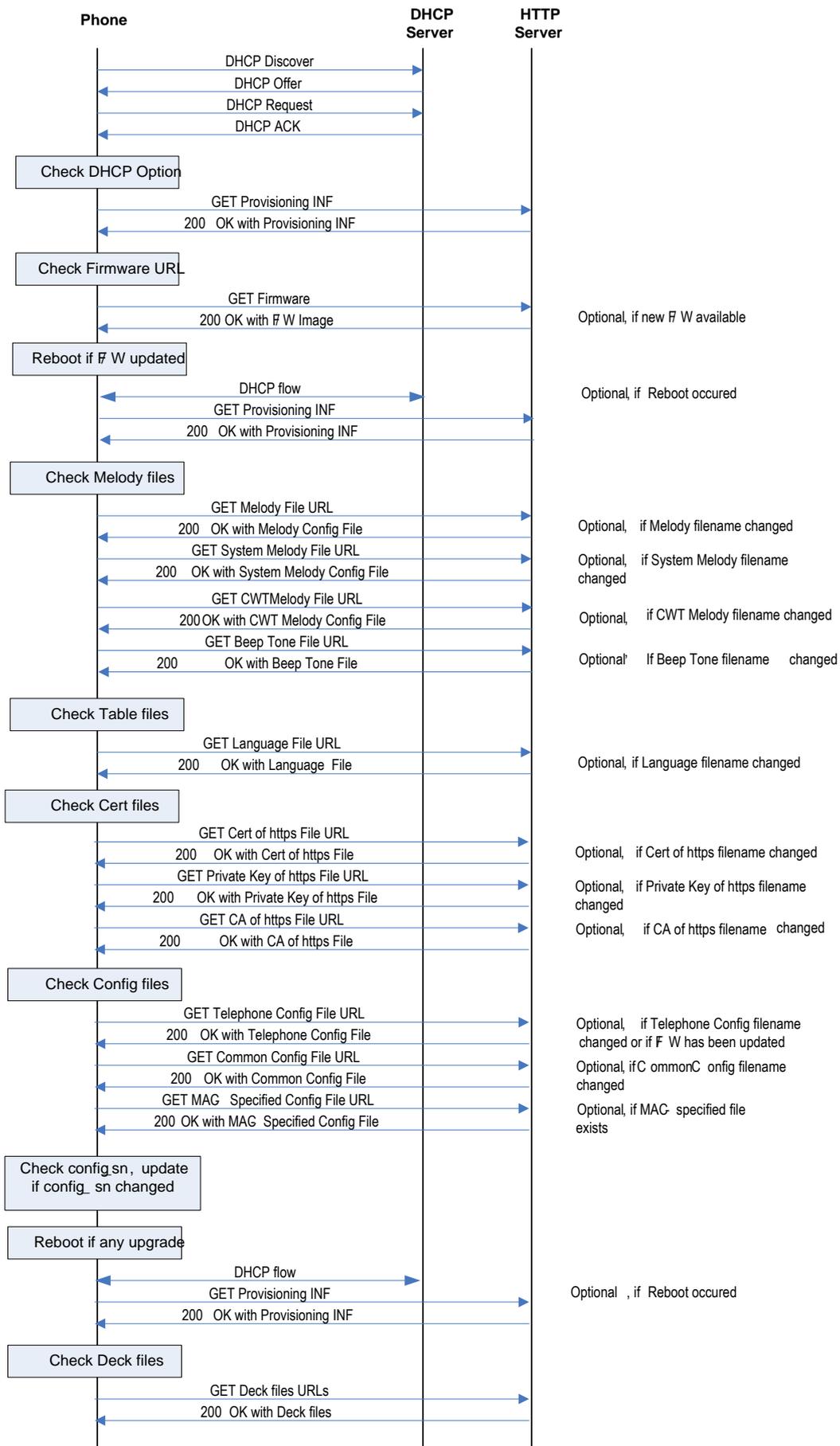
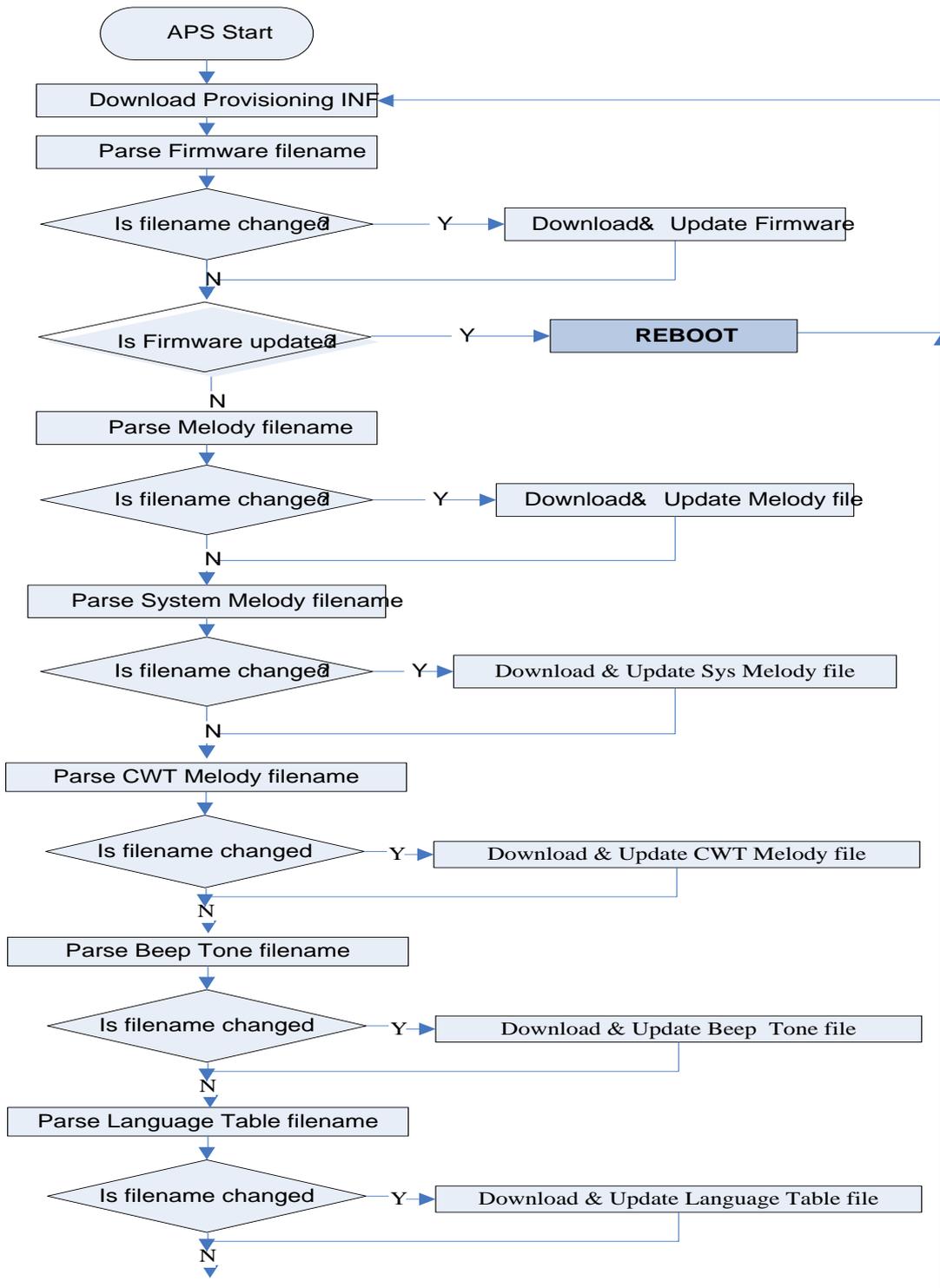


Figure 8 Message Flow of MGCP Phone based on DHCP and HTTP

1. Download upgrade information file from the HTTP server using the URL described in DHCP option 43.

This file contains the following information (Details refers to section 2.5.3):

- Provision mode
  - Firmware URL
  - Deck File URL
  - Melody file URL
  - CWT melody file URL
  - Beep Tone file URL
  - Language Table file URL
  - Cert of HTTPs file URL
  - Private Key of HTTPs file URL
  - CA of HTTPs file URL
  - Common config URL
  - Telephone config URL
  - MAC-specified config URL.
2. Upgrade firmware if firmware URL differs from last upgraded URL. Reboot if the firmware has been updated.
  3. Download melody file form HTTP server.
  4. Download system melody file form HTTP server.
  5. Download CWT melody file form HTTP server.
  6. Download Beep Tone file form HTTP server.
  7. Download Language Table form HTTP server.
  8. Download Cert of HTTPs form HTTP server.
  9. Download Private Key of HTTPs form HTTP server.
  10. Download CA of HTTPs form HTTP server.
  11. Download telephone config file from HTTP server if the filename is different from the previous or if the firmware has been updated.
  12. Download common config file from HTTP server if the filename is different from the previous.
  13. Download MAC-specified config file from the MAC-specified config URL (obtained in upgrade information file) followed by config filename (mac address preceded by the phone model "ST20XXm\_", appended by phone automatically), if it has been provided on the server.  
  
For example, if the URL in Provisioning INF is `http://upgrade.company.com/ST20XXm/config/`, phone will attempt get configuration from `http://upgrade.company.com/ST20XXm/config/STST20XXM_009096A1B1C1.txt`
  14. Update common configuration. Update MAC-specific configuration if `config_sn` differs from current one.
  15. Reboot if any upgrading successful.
  16. Download deck files



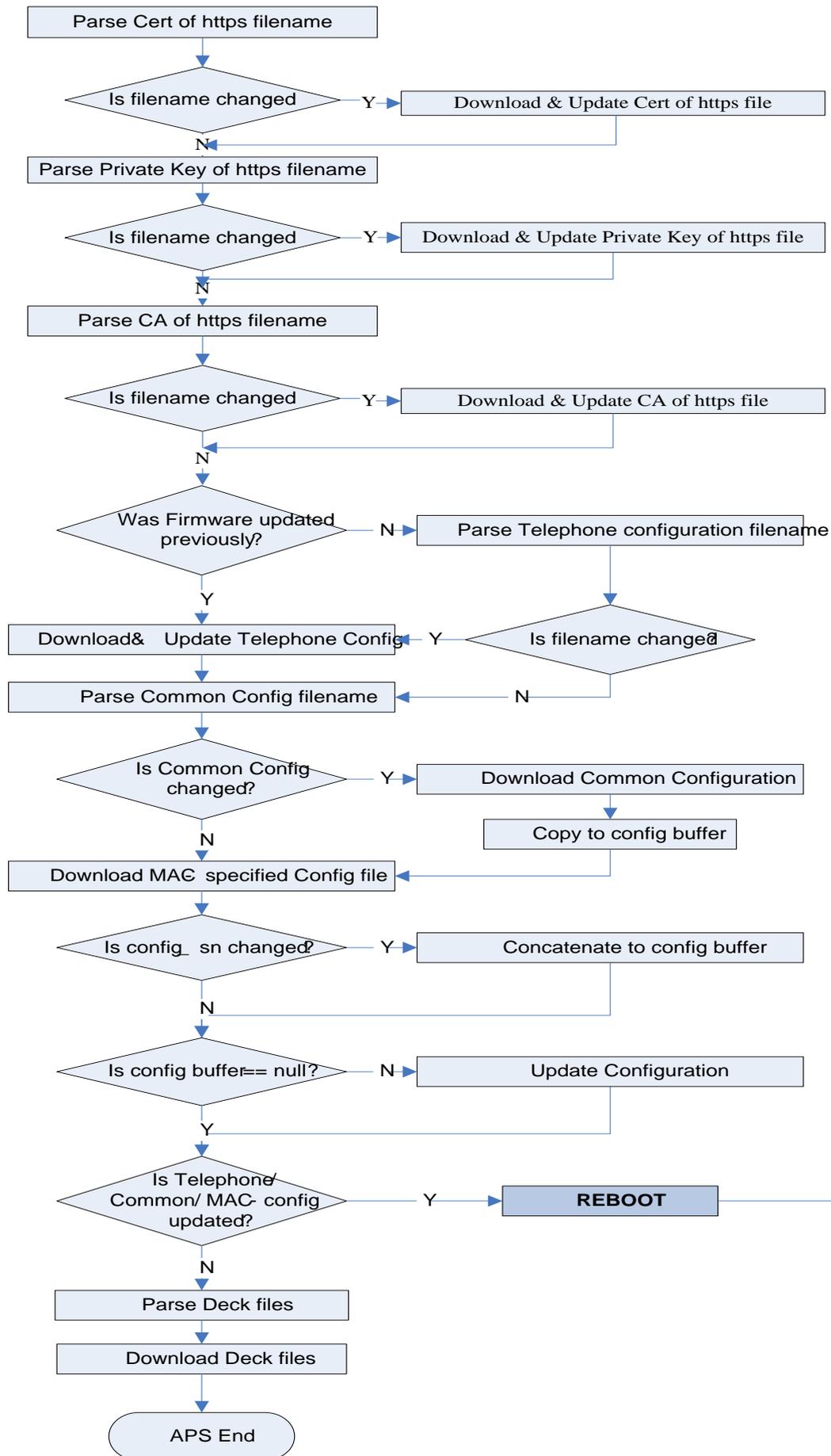


Figure 9 Version checking of MGCP Auto provisioning via DHCP + HTTP

### 2.5.3. Syntax of Provisioning INF (HTTP support relative path only)

#### ◆ [provision\_mode] Section

This section specifies if the phone is processing Auto-Provisioning or Customization.

- provision\_mode

**Syntax:**

provision\_mode=auto\_provisioning|customization

**Description:**

Indicator of provisioning mode (auto-provisioning or customization)

**Example:**

provision\_mode=auto\_provisioning

#### ◆ [application] Section

This section contains application firmware location.

- fwurl

**Syntax:**

fwurl=[url]

**Description:**

URL of application firmware

**Parameter Range:**

Maximum to 255 characters

**Example:**

fwurl=http://upgrade.company.com/st2030m/v100.zz

#### ◆ [deckfile] Section (For MGCP only)

This section contains location of deck file used for LCD display.

- deckfile

**Syntax:**

deckfile=[url]

**Description:**

URL of deck files used for LCD display.

**Parameter Range:**

Maximum to 255 characters

**Example:**

deckfile=http://upgrade.company.com/st2030m/main.thd

---

**◆ [melody] Section**

- Melody file

**Syntax:**

melodies=[filename]

**Description:**

Filename of melody files used for ringer melody (relative path only).

**Parameter Range:**

Maximum to 255 characters

**Example:**

melodies= <http://upgrade.company.com/st2030m/melody.txt>

- System melody file

**Syntax:**

system\_melodies=[filename]

**Description:**

Filename of system melody files used for SIP-INFO (relative path only).

**Parameter Range:**

Maximum to 255 characters

**Example:**

system\_melodies= <http://upgrade.company.com/st2030m/sysmelody.txt>

- CWT melody file

**Syntax:**

call\_waiting\_tone=[filename]

**Description:**

Filename of CWT melody files used for call waiting (relative path only).

**Parameter Range:**

Maximum to 255 characters

**Example:**

call\_waiting\_tone= <http://upgrade.company.com/st2030s/CWT.txt>

- Beep Tone file

**Syntax:**

beep\_tone=[filename]

**Description:**

Filename of Beep Tone files used for supervision (relative path only).

**Parameter Range:**

Maximum to 255 characters

**Example:**

beep\_tone = http://upgrade.company.com/st2030s/BT.txt

- Language table file

**Syntax:**

language\_table=[filename]

**Description:**

Filename of Language Table used for extra language upload (relative path only).

**Parameter Range:**

Maximum to 255 characters

**Example:**

language\_table= http://upgrade.company.com/st2030s/languagex.txt

- Tone table file (for SIP only)

**Syntax:**

tone\_table=[filename]

**Description:**

Filename of Tone Table used for extra country tones upload (relative path only).

**Parameter Range:**

Maximum to 255 characters

**Example:**

tone\_table= http://upgrade.company.com/st2030s/countryx.txt

◆ **[config] Section**

This section contains configuration file location.

- common\_config

**Syntax:**

common\_config=[URL]

**Description:**

Location of common config file

**Parameter Range:**

Maximum to 255 characters

**Example:**

common\_config=http://upgrade.company.com/st2030m/config/st2030m.txt

Phone will POST to http://upgrade.company.com/st2030m/config/st2030m.txt

- telcfg

**Syntax:**

telcfg=[URL]

**Description:**

Location of telephone config file

**Parameter Range:**

Maximum to 255 characters

**Example:**

telcfg=http://upgrade.company.com/st2030m/config/st2030m\_telcfg.txt

Phone will POST to http://upgrade.company.com/st2030m/config/st2030m\_telcfg.txt

- config

**Syntax:**

common\_config=[directory]

**Description:**

Location of config file

**Parameter Range:**

Maximum to 255 characters

**Example:**

config=http://upgrade.company.com/st2030m/config/

Phone will POST to <http://upgrade.company.com/st2030m/config/>

**2.5.4. Example of Provisioning Information File**

```
[application]
fwurl=http://upgrade.company.com/st2030m/v100.zz

[deckfile]
deckfile=http://upgrade.company.com/st2030m/main.thd
deckfile=http://upgrade.company.com/st2030m/renvoi.thd
deckfile=http://upgrade.company.com/st2030m/config.thd
deckfile=http://upgrade.company.com/st2030m/abrege.thd
deckfile=http://upgrade.company.com/st2030m/call.thd
deckfile=http://upgrade.company.com/st2030m/log.thd
deckfile=http://upgrade.company.com/st2030m/supervision.thd
deckfile=http://upgrade.company.com/st2030m/filtrage.thd
deckfile=http://upgrade.company.com/st2030m/register.thd

[config]
common_config= http://upgrade.company.com/st2030m/config/st2030m_common.txt
telcfg=http://upgrade.company.com/st2030m/config/st2030m_telcfg.txt
config=http://upgrade.company.com/st2030m/config/
melodies=http://10.1.29.198/ringtone_033006.txt
```

system\_melodies=http://10.1.29.198/ringtone\_040406.txt  
 call\_waiting\_tone=http://10.1.29.198/ringtone\_040406CWT.txt

(Note: call waiting tone download supported only for SIP, shown here to give an example only)

## 2.6 LCD Display during Provisioning

During provisioning, the phone is totally blocked (no reaction in case of any user action).

The phone indicates its upgrading state by System LED, and by displaying:  
 The system LED will be in Red Fast Blinking state during provisioning.

Line	LCD Message	Description
3	Provisioning...	during firmware, Configuration (and Deck file for MGCP) download
4	Please Wait...	during firmware, Configuration (and Deck file for MGCP) download

In any case of failure, the System LED blinks fast and an error message is displayed:

Line	LCD Message	Description
3	Wrong Firmware	If the firmware file name or URL is not as expected, or If the firmware used belongs to another phone (e.g. ST2022).
3	Wrong Com Config	If the configuration file name or URL is not as expected
3	Wrong Tel Config	If the tel config file name or URL is not as expected
3	Wrong Melody Config	If the melody file name or URL is not as expected
3	Server Unreachable	TFTP or HTTP server is unreachable.
3	Gen File Not Found	INF file is not found in server
3	Firmware Not Found	Firmware is not found in server
3	Melody Cfg Not Found	Melody file is not found in server
3	Com Config Not Found	Common Config is not found in server
3	Tel Config Not Found	Telephone Config is not found in server
3	Deck File Not Found	Deck Files are not found in server
3	Corrupted Firmware	If the chosen firmware file is invalid or corrupted
3	Corrupted Config	If the parameter is not correct
3	Corrupted Deck Files	The deck file name is not correct
3	Corrupted Melody file	If the melody file format is not correct
3	XML Syntax Error	Syntax Error in the Deck Files
4	Still using vxx.xx	xx.xx is the old FW version
5	Please contact admin	

If server unreachable or Gen File Not Found is the error case, IP Phone will retry 3 times and the duration is 30 seconds. On the other word, IP Phone will display the error message after 90 seconds.

The rest error messages will be displayed once the server or IP Phone detects unusual.

The LCD display and LED will return to normal when:

1. Any user operation (off hook, keypad operation)
2. Any incoming call

## 2.7 Syntax of Configuration File

Please refer to "ST20XXM\_Config File Syntax\_Vxxxx.pdf" for MGCP configuration parameters and "ST20XXS\_Config File Syntax\_Vxxxx.pdf" for SIP configuration parameters. These files can be found in the Release Package where you got this document.

## 3 Upgrade Triggers

### 3.1 Initial Upgrade

Every time when the phone powers up whenever cold start or warm start, this process is triggered.

The upgrade will not be immediate but it will take place randomly between boot-up time T0 and T0+'Upgrade time span'. 'Upgrade time span' is a configurable parameter from 0 to 255 (unit = minutes). 0 is the default value. This is to avoid server and network overload in case of power failure.

### 3.2 Periodical Upgrades

The phone will periodically connect to the server to check the new firmware, deck file and configuration availability within a certain time frame.

The periodicity is configurable. Unit is number of days, from 0 to 255 (0 means "no periodic upgrade"). Of course, day 0 corresponds to the day of the last reboot.

Administrator can setup a start time and an upgrade time span. Start time is in HH:MM format (00:00 to 23:59, or 00:00 to 11:59 am/pm according global time format). Upgrade time span is from 0 to 255 minutes. The process will be triggered randomly between 'Start time' and 'End time = Start time + Upgrade time span' according to the following two cases:

- NTP server available

Phone will refer to the NTP time and select a random timing between the start and end time to trigger this process. When the phone attempts to trigger this process, it will check the states of user operation. If the phone is not IDLE, auto-provisioning is held and will be launched only after the phone is back to IDLE mode.

- NTP server unavailable

Phone will never trigger this process.

### 3.3 Upgrade Timing

The upgrading takes effect after phone rebooting. The phone will not be rebooted until idle, however the phone is robust to power failure in during reboot and upgrading.

For example: If user set up,

No	Current NTP Time	AutoprovisionTimeDays	Autoprovisionstarttime	AutoprovisionTimeSpan	Explanation
1	17:00	0	00:00	00	APS will start immediately after ST20XX reboots. APS will be performed after ST20XX reboots again,
2	17:00	0	15:00	00	APS will start immediately after ST20XX reboots. APS will be performed after ST20XX reboots again,
3	17:00	0	15:00	10	APS will strat in ten minitues after ST20XX

					reboots. APS will be performed after ST20XX reboots again,
4	17:00	1	15:00	00	APS will start immediately after ST20XX reboots. APS will be performed at 15:00 two days later. The duration between current time and start time can't be less than 24 hrs.
5	08:00	1	15:00	20	APS will start within 20 minutes after ST20XX reboots. APS will be performed at 15:00 on the next day.

AutoprovisionTimeSpan has higher priority than SIP NOTIFY. It means ST20XX won't start APS right away when ST20XX receives SIP NOTIFY message within Timespan.

If the user would like to set up Starttime, he must be aware of the difference between current time and start time which can't be less than 24 hrs. Otherwise, APS will be postponed on the next day. (example 4 and 5)

### 3.4 SIP NOTIFY (SIP only)

The phone launches provisioning upon reception of a SIP NOTIFY :

```
NOTIFY sip: CLI@IP_ADDRESS_SIP_SERVER:5060;transport=UDP SIP/2.0
Call-ID: 03945c5ecd70248d6e76b8f0e6c64a13@10.107.111.189
CSeq: 460015 NOTIFY
From: <sip:webadmin@operator.com>;tag=8542
To: <sip: CLI@operator.com:5060>
Via: SIP/2.0/UDP
82.91.55.123:5061;branch=z9hg4bk917de6b406ebc8d056971e03991fdde6
Max-Forwards: 70
Contact: <sip:82.91.55.123:5061;transport=UDP>
Event: check-sync
Content-Length: 0
```

The phone may reboot or not, whether an upgrade is actually performed or not (i.e. a new FW or configuration is available).

Timespan will be taken into account, i.e, provisioning will not start immediately if there is a nonzero value for AutoprovisionTimeSpan.

For the basic NOTIFY-checksync behavior, the phone needs to have TFTP/HTTP servers and .inf file preconfigured.

To allow usage of DHCP parameters also, and at the same time provide a way to remotely reboot the phone, an additional parameter is recognized in the incoming SIP NOTIFY Event header: "reboot". Possible values are "true" and "false"

Thus,

```
Event: check-sync
or
Event: check-sync;reboot=false
```

Will cause the basic behaviour explained above, whereas

```
Event: check-sync;reboot=true
```

Will cause a reboot of the phone and the start of a new APS process, either using DHCP or preconfiguration depending on phone settings, and taking into account the corresponding AutoProvisionTimeSpan value.

There is a security issue: a cracker may broadcast such packets to the network frequently and cause the phones to be in provisioning mode or even reboot constantly. Strategy to overcome this problem is:

**-The usage of this feature is configurable via a "Use SIP NOTIFY to launch provisioning" parameter. Default value is 'No'.**

- Starting at version x.61, **it is possible to configure the address of the trusted NOTIFY source** via the parameter

```
AuthNotifyCheckSync=<ip|fqdn>
```

within the [sip] section of the APS files. Please see Config File Syntax document v27 (or newer) for more details. Default value is 0.0.0.0, which means all sources are accepted. Packets coming from not trusted ip's will be ignored.

### **3.5 Security Consideration**

HTTPS is now supported. For this, https urls have to be used.

## 4 VLAN provision

### 4.1 Overview

The aim of this section is to detail the changes on the provisioning process of Thomson ST20XX IP phone. These changes are requested in order to support advanced provisioning of the VLAN configuration, and to broaden the compatibility with routers not supporting option 67.

### 4.2 Support of “file” field in the DHCP header for APS

Not always DHCP servers allow configuring the Option 67 to provide the IP phone with the name of the .inf information file. This is the case of DHCP servers on board of Cisco Routers (depending on the IOS version). On the other hand, all servers allow setting the file field, as it is included in the basic BOOTP protocol header.

It is required by the ST20XXM to use the “file” field, in addition to the DHCP option 67, to retrieve the information filename from the DHCP replies. 67 have the priority on “file”.

### 4.3 Configuring VLAN IDs via DHCP options

In some deployments TFTP/HTTP servers are also within the voice VLAN.

The objective of this feature is to dynamically provision the phone with its VLAN id's without the need for an APS or setting the phones one by one.

Using DHCP custom options (code greater than 128), the DHCP protocol allows configuring the VLAN IDs in a very straightforward way.

The process described here is independent from APS being enabled or disabled.

### 4.4 Description

#### 4.4.1. Preconditions:

Phone in DHCP mode

#### 4.4.2. Parameters:

VLAN parameter must be set to 1 by either the config files (common or mac\_specified) or by manual setting. This will enable the phone to fill the Parameter Request List (option 55) by adding the new options requested.

The parameter PersistentVLANfromDHCP determines the behavior of the VLAN parameters got via DHCP:

- PersistentVLANfromDHCP=0: Dynamic behavior(default setting).
- PersistentVLANfromDHCP=1: Persistent behavior.

### 4.4.3. DHCP options:

The parameters to provision are:

- VLAN enabling flag (1 byte) in option 190
- Voice VLAN ID (2 bytes) in option 191
- Data VLAN ID (2 bytes) in option 192

A VLAN configuration via DHCP is said valid when all three parameters are provided and the three have a valid value.

## 4.5 System flow:

### 4.5.1. PersistentVLANfromDHCP=0

System boot

If valid DHCP(VLAN) options are present, phone will apply them.

Then, with VLAN settings already active, it will follow normal APS process (if enabled), based on rest of DHCP provided options or preconfigured values, with the restrictions mentioned below.

If DHCP(VLAN) options are not present or the values are not valid, it will follow normal APS process (if enabled).

DHCP provisioned VLAN values are only applicable until the next reboot. DHCP renewals will not yield changes to VLAN settings.

### Interaction with DHCP-based and file-based VLAN provisioning

DHCP provisioned VLAN values have priority over APS VLAN values during their lifetime.

That is, a valid DHCP VLAN provisioning will force the phone to ignore (not to overwrite) the values contained in the configuration files.

In case the ST2030 has received the VLAN configuration via DHCP, it doesn't overwrite any of them with any default value in case they are missing in the config file.

That is, it doesn't use default values but DHCP provided ones if the config file did not contain VLAN parameters.

### 4.5.2. PersistentVLANfromDHCP=1

- If the value is 1, then the behavior must be static. In this case once the phone got its VLAN parameters from the DHCP, it stores them in the flash memory and will use them after each restart even if the DHCP does not give those parameters again.

The only way to change the VLAN settings would then be:

Reset to default procedure

Change in VLAN parameters provided by the DHCP server

VLAN settings provided in the config file don't override the VLAN settings given by the DHCP, but they are stored in case they are needed later (due to a change in PersistentVLANfromDHCP, for example, or in case DHCP server options are never provided).

To summarize, in case PersistentVLANfromDHCP = 1, at boot time the discovery/request will use the latest DHCP parameter used (if any).

- In case DHCP VLAN options are NOT provided, the phone will continue to use the same old DHCP-based VLAN id.
- In case DHCP VLAN options are NOT provided, and old DHCP-based VLAN id are not available (because they have never been provided before, for example), the phone will use the parameters obtained via APS or manual configuration
- In case DHCP VLAN options are provided and valid, the phone will use the new DHCP-based VLAN ids

As an example, this would be an APS process in an MGCP set where the phone was initially using VLAN id set **Tag1s** (Tag1s being acquired via whatever configuring means). It is assumed in this example that:

- either Telconf, or Common or MAC file contain vlan id parameters **Tag3s** and include **PersistentVLANfromDHCP=1**
- the files are applied (change in name, or config\_sn updated)

**Reboot** + [**Tag1s** DHCP(**Tag4s**)] + [**Tag4s** TFTP(**Tag3s**)] + (APS finish & Reboot) + [**Tag4s** DHCP(**Tag4s**)] + [**Tag4s** TFTP(**Tag3s**)] + [**Tag4s** -MGCP] + ..

Where the meaning of those expressions is:

[**Tag1s** DHCP(**Tag4s**)] --> the DHCP handshake was tagged with vlan parameter set Tag1s, and DHCP server provided Tag4s in options 190, 191, 192

[**Tag4s** TFTP(**Tag3s**)] -->TFTP request was tagged with Tag4s, and config files provided via TFTP contained Tag3s

As an additional example, this would be an APS process in an MGCP set starting from default settings, where the DHCP server is not providing vlan options, and APS is providing VLAN parameters. It is assumed in this example that:

- either Telconf, or Common or MAC file contain vlan id parameters **Tag3s** and include **PersistentVLANfromDHCP=1**
- the files are applied (change in name, or config\_sn updated)

**Reboot** + [Untagged DHCP(No tags)] + [Untagged TFTP(**Tag3s**)] + (APS finish & Reboot) + [**Tag3s** DHCP(No tags)] + [**Tag3s** TFTP(**Tag3s**)] + [**Tag3s** -MGCP] + ..

And the next reboot would be, assuming that APS process does not yield any additional change to be applied:

**Reboot** + [**Tag3s** DHCP(No tags)] + [**Tag3s** TFTP(**Tag3s**)] + (APS finish & Reboot) + [**Tag3s** DHCP(No tags)] + [**Tag3s** TFTP(**Tag3s**)] + [**Tag3s** -MGCP] + ..

Where the meaning of those expressions is:

[Untagged DHCP(No tags)]--> the DHCP handshake was untagged and DHCP server did not provide options 190, 191, 192, or they were not valid

[**Tag3s** TFTP(**Tag3s**)] -->TFTP request was tagged with Tag3s, and config files provided via TFTP contained Tag3s

[**Tag3s** DHCP(No tags)] --> the DHCP handshake was tagged with Tag3s, and DHCP server did not provide options 190, 191, 192, or they were not valid

For any configuration the VLAN settings shown by the phone through the proper menu reflect the real value applied.

## **5 SIP/MGCP SWAP MECHANISM**

Please refer “Thomson ST20xx SIP-MGCP swap procedureEd2.pdf” for detail.

## 6 Support Encryption of Configuration files in APS

Starting from x.67.2, the transfer of encrypted configuration files is supported to prevent data exposure during APS.

### Encrypted files support:

- Common configuration file (Comconfig file)
- MAC configuration file (MacConfig file)

### Encryption method:

Symmetric method, advance encryption standard (AES) 128bit

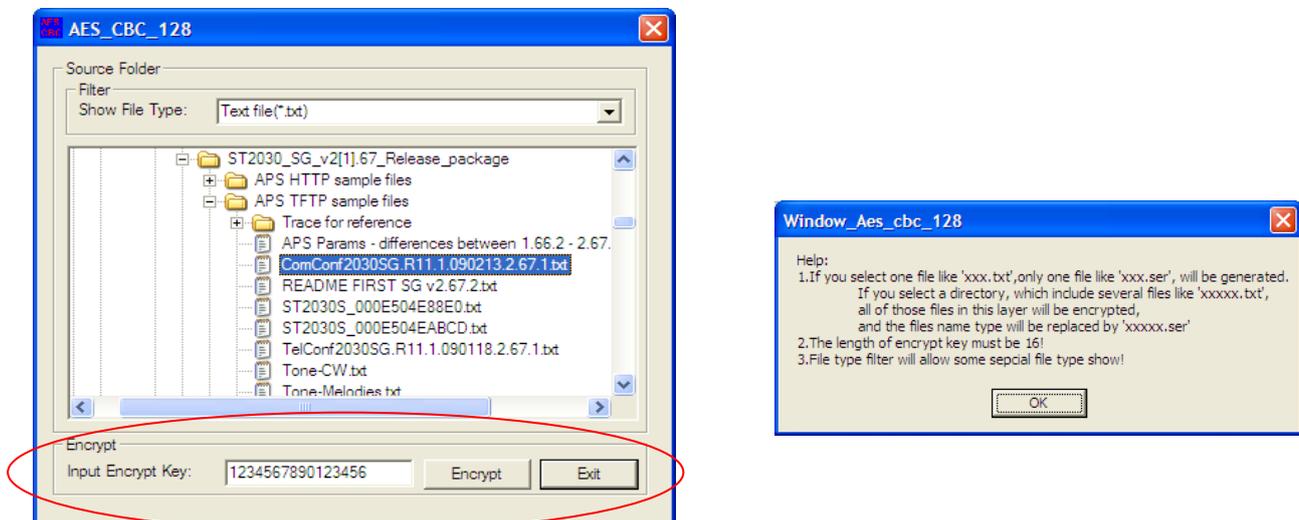
### Encryption/Decryption tools:

- Name: Window\_Aes\_cbc\_128.exe (including in release package/ Decryption tool folder)
- Press F1 for get HELP window

### In Window Platform

File name: Window\_Aes\_cbc\_128.exe in ../Encryption Tool folder.

Select the Configuration file and input a 16-digits key for encryption. Press [Encrypt] to start the process. Press 'F1' to get help window for more information.



### In Linux Platform:

Filename: Linux\_aes\_cbc\_128\_bin.tar.gz ../Encryption Tool folder.

- tar zxvf FileName.tar.gz
- cd FileName
- ./aes\_cbc\_128 <key> <source file or directory> <destination file or directory>
- '\* .ser' file will be generated

```

root@localhost:/tmp/Encryptool
File Edit View Terminal Tabs Help
[root@localhost Encryptool]# ls
ComConf2030SG.R11.1.090213.2.67.1_test.txt Linux_aes_cbc_128_bin.tar.gz
[root@localhost Encryptool]# tar zxvf Linux_aes_cbc_128_bin.tar.gz
aes_cbc_128
[root@localhost Encryptool]# ls
aes_cbc_128 ComConf2030SG.R11.1.090213.2.67.1_test.txt Linux_aes_cbc_128_bin.tar.gz
[root@localhost Encryptool]# ./aes_cbc_128
./aes_cbc_128 <key> <input file or directory> <decrypty out file or directory>
[root@localhost Encryptool]# ./aes_cbc_128 1234567890123456 ComConf2030SG.R11.1.090213.2.67.1_test.txt ComConf2030S
Encrypt file ComConf2030SG.R11.1.090213.2.67.1_test.txt succesfull!
[root@localhost Encryptool]# ls
aes_cbc_128 ComConf2030SG.R11.1.090213.2.67.1_test.ser ComConf2030SG.R11.1.090213.2.67.1_test.txt Linux_aes_cbc_
[root@localhost Encryptool]# ls -l
total 860
-rwxrwxr-x 1 501 501 559713 2009-02-26 15:35 aes_cbc_128
-rw----- 1 root root 16128 2009-03-04 19:05 ComConf2030SG.R11.1.090213.2.67.1_test.ser
-rwxr-xr-x 1 root root 16114 2009-03-02 14:41 ComConf2030SG.R11.1.090213.2.67.1_test.txt
-rwxr-xr-x 1 root root 261249 2009-02-26 15:40 Linux_aes_cbc_128_bin.tar.gz
[root@localhost Encryptool]#

```

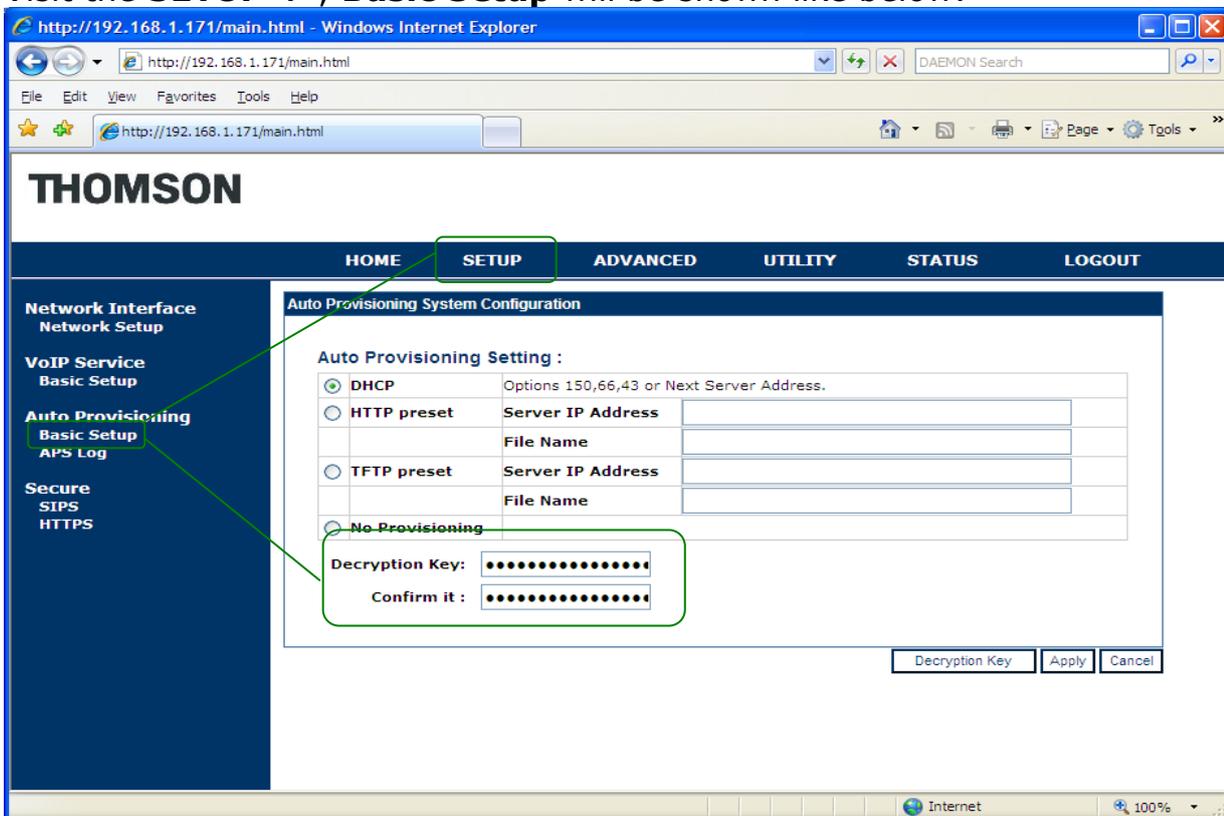
**Things to know:**

The ComConfig/ MacConfig file should be encrypted with symmetric key offline before doing APS. The decryption will be handled on the phone by using the same key.

**Feature Activation**

**A. Via WebGui**

Visit the **SETUP** → , **Basic Setup** will be shown like below:



Input the decryption key (16 digits) in the field and confirm field. If the key is different with the key for confirmation, an error message will prompt you to input

the key again.

Press the [**Decryption key**] to save the key settings in the phone.  
 → Phone reboot.

If the key doesn't match with the key for encrypted files during APS, the files are not allowed to download to the phone.

You are recommended to check the **APS log** after doing APS to make sure its process is running correctly.

**APS log (if Decryption key Match):**

```

Begin Common Config download...
AES decrypt common config file successfully!
CommonConfig: ComConf2030SG.R11.1.090213.2.67.1.ser download successfully!
Begin Mac config download...
MacConfig: ST2030S_00147F00DFE4.ser download successfully!
AES decrypt mac file successfully!
Serial number is the same!
Begin upgrading config file...
Check config file syntax
Check config file syntax successfully!
Upgrading config file successfully!
    
```

**APS log (if Decryption key not-Match):**

```

Begin Common Config download...
AES decrypt common config file successfully!
CommonConfig: ComConf2030SG.R11.1.090213.2.67.1.ser download successfully!
Begin Mac config download...
MacConfig: ST2030S_00147F00DFE4.ser download successfully!
AES decrypt mac file successfully!
Error: Parse config file failed or config_sn tag not found!
Begin upgrading config file...
Check config file syntax
Error: Config file format error
    
```

On the **MMI** of the phone, you also be noted that the **Provision Error!** displaying on the screen when the .ser is failed to download.

**B. Via APS**

APS in [autoprovision] section of both Common and Specific-MAC config files with the new parameter:

Decryption\_Key=Th0mson2\$8s8@9z! (Default)

```

[autoprovision]
...
Decryption_Key=Th0mson2$8s8@9z!
...
    
```