Internet Engineering Task Force
INTERNET-DRAFT
schulzrinne-sip-config-00.ps

SIP WG
Schulzrinne
Columbia U.

December 28, 2000
Expires: June 2001

Configuring IP Telephony End Systems

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

To view the list Internet-Draft Shadow Directories, see http://www.ietf.org/shadow.html.

Copyright Notice

Copyright (c) The Internet Society (2000). All Rights Reserved.

Abstract

IP telephony devices are likely to be simple systems that are going to be deployed in large numbers. This document discusses alternatives for configuring such devices that is independent of the vendor.

1 Introduction

SIP [1] user agents need to be configured with a variety of SIP- and application-related parameters. In many environments, thousands of such devices are likely to be deployed, thus making automated configuration management a necessity. Automating configuration is particularly important for these devices since many are likely to have only a limited user interface, such as a twelve-key keypad and a two-line text display, making manual configuration tedious, error-prone and time consuming. These devices may also be severely limited in their computational capabilities and protocol support. Also, devices are likely to have a limited amount of non-volatile memory. Thus, these devices are in many ways similar to the diskless workstations that motivated the development of protocols like BOOTP and DHCP. However, unlike the environments envisioned in the zero-configuration scenarios [2], large Internet telephony deployments generally have an existing infrastructure of DHCP and other servers.

Here, we are not concerned with configuring basic IP information, such as

- network address
- network mask,
- host name,
- DNS servers,
- time zone (DHCP 88)

- time servers, or
- default routers.

All of these are readily configured via DHCP [3].

Also, the SIP outbound proxy can be configured via SIP DHCP [4].

Configuration information can be roughly divided into four areas: SIP information, user interface, RTP and company-specific information. Some of this information is specific to a particular device or a particular SIP user (labeled as "ind."), while other information is the same for all devices in an organization ("org.").

	Value	for
SIP information includes	default Route destinations	org.
	values for Organization header	org.
	values for display name	ind.
	request timeout parameters	org.
	request initial retransmit interval	org.
	registration authentication secrets	ind.
	request authentication secrets	ind.
	voicemail forwarding SIP URLs for busy	ind./org.
	voicemail forwarding SIP URLs for no answer	ind./org.
	no-answer ring durations before forwarding	ind./org.
	non-SIP Contact information	ind./org.
	911 configuration: PSAP	org.
	911 configuration: location	ind.
	ring tones, volume	ind.

Note that not all of this information needs to be configured in the end system. A script providing call handling information may subsume busy and no-answer handling instructions. An outgoing proxy may add Organization headers. Non-SIP URLs may be configured through a non-SIP interface, such as a web page or database, accessible to the registrar.

User information includes

- the definition of speed-dial buttons;
- call log (for callbacks);
- user interface language.

RTP information includes

- the default media to advertise,
- the default audio or video codec,
- the default packetization interval,
- the use of silence suppression,
- the use of audio sidetone,
- the transmission of touch tones as either encoded waveforms or as RFC 2833 [5] payloads.

In addition, end systems may want to retrieve address books, either in full or as updates to information available locally. LDIF [6] appears to be sufficient for that purpose, although the transport mechanism remains to be determined, with tftp and SIP (using TCP), described below, as feasible options.

2 Design Choices

There are a number of existing alternatives for conveying configuration information to end systems. However, unlike for routers and workstations, some SIP user agents are likely to be severely constrained in their ability to run a multitude of protocols, due to very limited memory space. Some devices may not support TCP, for example, as it is not needed for other services on such devices.

Mechanisms need to be initiated by the end system, since configuration information needs to be restored immediately after a reboot.

2.1 SNMP

SNMP [7] has the advantage of being able to use UDP, but it is unsuitable since actions are initiated by a network management system, not the system to be managed. Also, implementation complexity is a concern.

2.2 DHCP

The Dynamic Host Configuration Protocol [3, 8] is a natural choice since end systems already need to support it to obtain IP-level configuration information. DHCP also requires only UDP and has modest implementation complexity. The main disadvantages are practical. In some systems, it is difficult for an application to gain access to DHCP information. This, however, is not likely to be a major issue for embedded systems. Also, all DHCP information generally resides on a single server, so it requires the ability of those administering SIP end systems to gain access to the DHCP server database. Information in DHCP databases can be tailored to a particular MAC address or user identification string. The latter allows, for example, to retrieve information associated with a particular user rather than a particular device.

DHCP options are numbered. The space of available DHCP options, only 255 entries, is nearly exhausted, so that any use of DHCP for the dozen or more configuration options likely to be needed will have to be sub-options. However, care must be taken to avoid complicated data structures, as typical DHCP servers only support very simple data structures, such as strings, integers and network addresses.

If users “log in” to a device, the DHCP query would have to be re-done once the user identify is known. Generally, DHCP appears designed for device-level configuration, not user-level configuration.

DHCP provides limited security, but lower-layer security mechanisms such as IPsec can be employed for confidentiality and authentication.

DHCP options are limited to 255 bytes. Since DHCP packets are transmitted via UDP prior to MTU discovery, the overall length is limited to 576 characters, although clients may negotiate the use of larger DHCP messages [3].

2.3 ACAP

The Application Configuration Access Protocol [9] is a line-oriented protocol, similar in flavor to POP or IMAP, that allows to retrieve configuration variables. For example,

```
C: A046 SEARCH "/addressbook/" DEPTH 3 RETURN ("addressbook.Alias"
```

```
        "addressbook.Email" "addressbook.List") OR NOT EQUAL
        "addressbook.Email" "i;octet" NIL NOT EQUAL
        "addressbook.List" "i;octet" NIL
S: A046 ENTRY "/addressbook/user/joe/A0345" "fred"
        "fred@stone.org" NIL
S: A046 ENTRY "/addressbook/user/fred/A0537" "joe" "joe@stone.org"
        NIL
S: A046 ENTRY "/addressbook/group/Dinosaur Operators/A423"
        "saurians" NIL "1"
S: A046 MODTIME "19970728105252"
S: A046 OK "SEARCH completed"
```

The protocol requires a reliable stream transport such as TCP. It's main use appears to be in the configuration of email clients. Parameters are named and allow textual and binary values of nearly unlimited length.

ACAP can use TLS [10] to secure its communication channel against eavesdropping, modification and unauthorized use. SASL is used for authentication.

2.4 tftp

The Trivial File Transfer Protocol (tftp) [11] is a simple UDP-based file retrieval protocol which has been commonly used to load boot code into memory. Thus, it is likely to be needed by many implementations. Tftp provides weak security, as it is designed to serve public information.

Configuration information could be returned as a text or binary file, e.g., as lines of parameter-value text or XML.

The address of the tftp server can be configured via DHCP (Option 66).

2.5 SIP

SIP REGISTER responses can contain message bodies, using similar configuration parameter formats as the options described for tftp. This approach has the advantage that the information is readily available to the application, but does require registrar software modifications.

To avoid additional configuration, the SIP UA can use multicast registrations to locate the appropriate registrar or it could attempt to use the outbound proxy.

2.6 Others

Other solutions, such as LDAP or SQL, are theoretically feasible but likely to be excessively complex or insufficiently standardized as protocols.

3 Suggested Approach

Among the solutions suggested above, DHCP and SIP-REGISTER appear to be the easiest to implement. In DHCP, a means for packing sub-options needs to be found to circumvent the code-space constraints. A simple, but somewhat kludgey, mechanism packs XML or lines of parameter-value pairs into the information element.

4 Information Elements

As a strawman, we propose a simple hierarchical data streams with text values. This can either be encoded into XML, as in

```
<par name="sip.busy.timeout">10</>
<par name="acme.par17" encoding=base64>84kz8Z37...</>
```

or in an LDIF-like format, where base64-encoded parameters are marked with a double colon:

```
sip.busy.timeout: 10
sip.busy.url: sip:foo.com
acme.par17:: base-64 encoded string
```

The LDIF-like format appears simpler to parse, but a specialized XML parser for a small number of elements seems feasible even for low-functionality end systems.

5 Security Considerations

Some configuration information, such as speed-dial configuration or voice-mail URLs, may be considered sensitive information and thus should not be disclosed to unauthorized users even within the same organization.

6 Authors' Addresses

Henning Schulzrinne
Dept. of Computer Science
Columbia University
1214 Amsterdam Avenue
New York, NY 10027
USA
electronic mail: schulzrinne@cs.columbia.edu

References

[1] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, "SIP: session initiation protocol," Request for Comments 2543, Internet Engineering Task Force, Mar. 1999.

[2] M. Hattig, "ZeroConf requirements," Internet Draft, Internet Engineering Task Force, Sept. 2000. Work in progress.

[3] R. Droms, "Dynamic host configuration protocol," Request for Comments 2131, Internet Engineering Task Force, Mar. 1997.

[4] G. Nair and H. Schulzrinne, “DHCP option for SIP servers,” Internet Draft, Internet Engineering Task Force, Apr. 2000. Work in progress.

[5] H. Schulzrinne and S. Petrack, “RTP payload for DTMF digits, telephony tones and telephony signals,” Request for Comments 2833, Internet Engineering Task Force, May 2000.

[6] G. Good, “The LDAP data interchange format (LDIF) - technical specification,” Request for Comments 2849, Internet Engineering Task Force, June 2000.

[7] K. McCloghrie, D. Perkins, and J. Schoenwaelder, “Structure of management information version 2 (smiv2),” Request for Comments 2578, Internet Engineering Task Force, Apr. 1999.

[8] S. Alexander and R. Droms, “DHCP options and BOOTP vendor extensions,” Request for Comments 2132, Internet Engineering Task Force, Mar. 1997.

[9] C. Newman and J. G. Myers, “ACAP – application configuration access protocol,” Request for Comments 2244, Internet Engineering Task Force, Nov. 1997.

[10] C. Newman, “Using TLS with IMAP, POP3 and ACAP,” Request for Comments 2595, Internet Engineering Task Force, June 1999.

[11] G. Malkin and A. Harkin, “TFTP blocksize option,” Request for Comments 2348, Internet Engineering Task Force, May 1998.

Full Copyright Statement

Copyright (c) The Internet Society (2000). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an ”AS IS” basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.