Internet Engineering Task Force
INTERNET-DRAFT
draft-moyer-sip-appliances-framework-00.ps

SIP WG
S. Moyer
D. Marples
S. Tsang
J. Katz
P. Gurung
T. Cheng
A. Dutta
Telcordia Technologies Inc.

H. Schulzrinne
Columbia University

July 2000
Expires Jan 2001

Framework Draft for Networked Appliances using the Session Initiation Protocol

Status of this Memo

Abstract

This document proposes the use of SIP for Network-capable appliances. It leverages the standard SIP capabilities to directly communicate with appliances even when they are behind firewalls, NATs or other entities that prevent direct end-to-end communication. When combined with the recently proposed Instant Messaging and Presence SIP extensions these techniques become even more powerful.

Contents

1 Introduction

The next wave of the Internet is widely predicted to be the move towards the *Networked Appliance (NA)*; The Fridge that can keep an inventory of your groceries and re-order when necessary or perhaps the Alarm Clock that can co-ordinate your agenda, the weather and the road conditions to determine the correct time to wake you up. It is clear that these appliances will need to communicate amongst themselves so that, for example, the Alarm Clock can turn on the Bedroom Lamp, but the mechanism these appliances will use to communicate is far from obvious. This document provides the rationale behind the requirement for the standardization of these mechanisms and proposes extensions to the SIP architecture to meet these requirements. It further presents ways in which these new mechanisms are intended to be used.

Networked Appliances (NAs) are dedicated function consumer devices containing at least one networked processor. Examples include lamps, coffee makers and alarm clocks. Figure 1 provides an example of a home domain containing Networked Appliances.

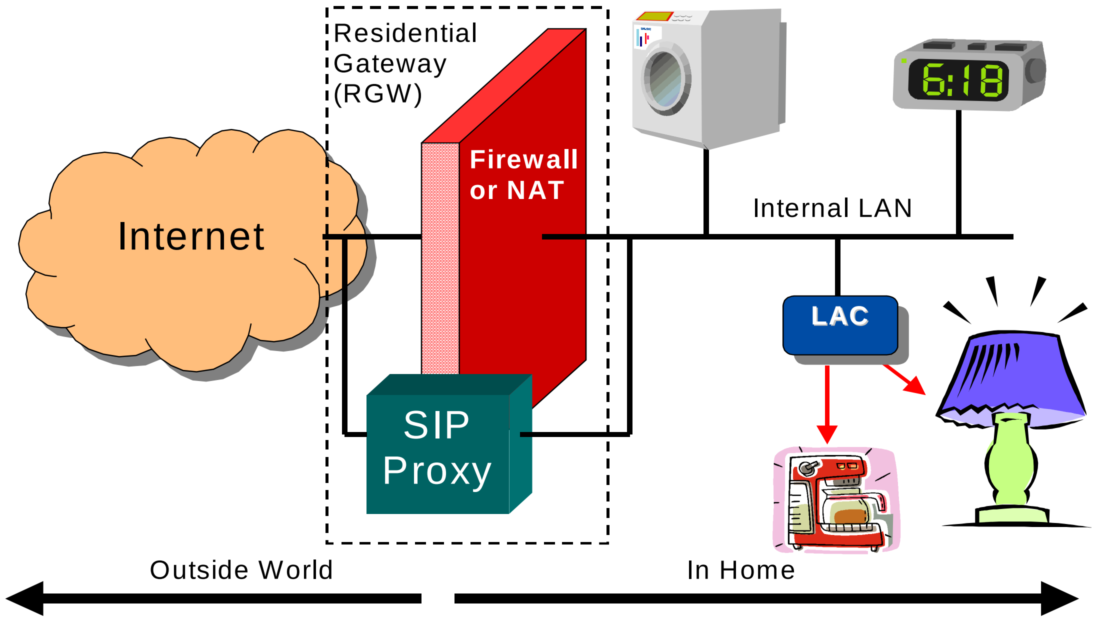

Figure 1: Example Home Domain Containing Networked Appliances

1.1 Intra-Home Communication

Appliances within a single home are not intended to use the capabilities outlined within this document for communication between themselves, but it is important to have a conceptual understanding of the in-home communication techniques that are being developed in order to understand the relationship of this document to them. For the purposes of this document, a home is considered to be a single (DNS) domain.

Inter-device communication within a single domain can be subdivided into two essential issues; the *location and identification* of the device it is desired to talk to and then the *communication* with it. Some approaches separate these two issues explicitly although most tend to merge them into a single solution. There are numerous emerging standards for in-home inter-device communication (HAVi[3], VESA Home Networking[6], JINI[5] and UPnP[2] being four obvious examples) and there are other standards explicitly dedicated to *location and identification* of services (SLP[13] and Salutation[4] for example). There seems little point in bringing new techniques into this already confused space.

Even allowing for the proliferation of intra-home communication standards, it is possible to identify a number of features that are common to each of them, as follows;

- The rules that lead appliances to contact each other will either be coded in the appliance, or possibly downloaded from network servers.
- These rules will be written in protocol specific terms such as *Send a message to controller at IP address 10.23.45.67 port 2357, turn on item number 4555*.
- The way in which the mapping from protocol specific descriptions to human user-friendly descriptions is standard dependant, and is, in general, one of the less well developed aspects of the standards.

1.2 Extra-Home communication

The development of techniques for access into the home from outside of it has not received anything like the attention that the in-home communication problem has. This is reasonable, given that one is predicated on the other.

There are a number of additional factors that need to be accommodated when considering communication outside of the home, notably;

- **Security –** In-home communication exploits a level of physical security that is lost when arbitrary access from outside of it is permitted.
- **Authentication –** The entity trying to enter into the home needs to be unambiguously identified prior to permitting access.
- **Reliability** – Because of the wide-area nature of extra-home access, there are more points of failure. The home should continue to operate independently of external systems when communication with them is lost.
- **Scaling –** there are very many homes.
- **Protocol Independence** – Although within a single home it is acceptable that many different protocols are used for inter-device communication, a much more protocol-independent approach is required for the wide area, since the exact details of the devices comprising the in-home network may not be known from the outside world.
- **Naming and Location** – Devices within the home need to be unambiguously named and their location identified from outside of it.

Techniques are being developed to act as a beach-head into the home from the outside world, most notably by the Open Services Gateway Initiative (OSGi) [1] but this still does not address the general problem of wide area access, with the considerations above taken into account. Platforms such as the OSG may serve as the basis for such a set of techniques, however.

2 The Use of SIP for Extra-home communication

This document outlines an architecture initially explicitly targeted to appliances but with more general applicability to any networked device, in which the location phase and communication (or action) phases are merged into a single activity. The requesting agent sends an instruction to perform an action on a named object in a message. The message contains the name of the object upon which the action should be performed as its address, and the action itself as the payload. This message is routed from agent to agent, resolving the name as it goes along.

For example, the command *Switch on the lamp in the master bedroom in Dave's house* is first routed to the server that knows where *Dave's house* is. Then the message is onward routed to the *Dave's house* firewall, where access control and authorization is performed. If this is successful, the message payload is then delivered to the device to perform whatever action has been requested.

We may easily observe that many of these concepts are already present in the Session Initiation Protocol (SIP). SIP performs exactly this routing by name function in the `INVITE` process. An `INVITE` is sent first to an agent, or proxy, for the name. The Proxy can rewrite the name and relay the `INVITE`, getting closer to the eventual destination for the message, delivering the payload (which is conventionally Session Description Protocol (SDP)) once it arrives. The *Location* and *Action* processes are intertwined in the same procedure. In addition, the SIP security architecture enables verification based on these high level names. On initial consideration, it appears that SIP is capable of performing the functions identified above.

There are, however, two essential differences between the capabilities of SIP and the identified requirements;

1. SIP URIs are, in practice, Internet DNS addresses.
2. The only *action* capability that the SIP `INVITE` method can perform is to set up a session with associated bearers, using SDP (or some other MIME TYPE, e.g., ISUP/QSIG).

If these differences can be addressed, SIP becomes a very practical method of communicating with appliances.

2.1 Modifications and Extensions

To address the issues identified above, the following modifications are proposed to SIP. These modifications and extensions are described in more detail in the Internet-Drafts [7] and [8].

2.1.1 URL Changes

In SIP, the names that are found in the To: and From: fields are encoded as Universal Resource Locators (URL). Current implementations support SIP and PHONE URLs. One could define a new type of URL without changing the nature of the protocol. This allows for “user friendly” discovery of the appliance address. An example, using the service URL syntax defined in RFC2609 but without the location information (which has already been determined via the SIP routing would be:

```
slp://d=lamp,r=bedroom
```

By base64 encoding this URL (and potentially encrypting it to avoid revealing information about the types of devices contained in the domain) it is possible to structure this URL as part of a SIP URL;

```
sip:a458fauzu3k3z@stan.home.net
```

Thus, the existing structure of `<entity>@<location>` is maintained even when extended to accommodate appliances.

2.1.2 New excitation Method

SIP was initially created with call set-up in mind. It is intended for establishing a relationship, or session, between two endpoints such that ongoing bearer paths can be established between them. This structure could be generalized to cater for ‘short-lived’ connections if the connection establishment phase were removed and the payload generalized. The difference between the current way in which SIP is used and the proposed modifications is analogous in many ways to the difference between TCP and UDP or other Session/Datagram protocols.

A new method is being defined as part of the initiative to use SIP for Instant Messaging [7],[8]. This method, called `MESSAGE`, meets the requirements identified above and can carry payloads other than SDP. Any MIME type could be used as the payload of a SIP command and new MIME types could easily be defined for Action Languages for particular classes of appliances. `MESSAGE` would carry the command that is appropriate for the target appliance, such as *Turn The Light On*. The command would trigger a single response, indicative of its result, which would be carried by the standard SIP response mechanisms.

2.1.3 New Payload Type(s)

The typical MIME payload for SIP INVITE messages is SDP (Session Description Protocol). For networked appliances, a payload type that is specific for communicating with devices is required. We therefore propose a new MIME type called Device Messaging Protocol (DMP). The exact details of the DMP are still under investigation, but we believe it will be an XML-based specification that may be similar to Universal Plug 'n Play's Device Control Protocol [9].

In addition, when a device registers with a Proxy (via the REGISTER message) a description of that device needs to be conveyed. We propose to use a Device Description Protocol (DDP) to carry this information. Like the DMP, the exact details are still under development, but it also will likely be XML-based and will leverage existing work in this area.

2.1.4 Notification/Events

In addition to synchronous communication with networked appliances, a need also exists for asynchronous communications. For example, to be notified when an alarm goes off in your home, a certain temperature is reached, or when someone rings your doorbell.

The SIP instant messaging work [7],[8] defines, in addition to the MESSAGE method, two new primitives, SUBSCRIBE and NOTIFY that can be used to achieve asynchronous communications. When these two methods are used in conjunction with the proposed URL changes (specified in section 2.1.1) and the Device Messaging Protocol MIME type, then event notification from and between networked appliances is enabled.

3 Example Network Architectures

The architectures outlined in this section provide three different examples of how to support remote communication with networked devices. Note that actual implementations may use any combination of the three architectures described hereafter or something completely different. In the first, the client application is able to directly connect to and interact with the Home Domain. The second is when the client application must interact with a proxy located in the home's gateway device in order to communicate with networked devices in the home and the final architecture adds another proxy outside of the home (e.g., in a Service Provider's network) that "outsources" some of the functionality of the gateway proxy. All three of these architectures are now described.

3.1 Example 1: Direct Communication with the Home Domain

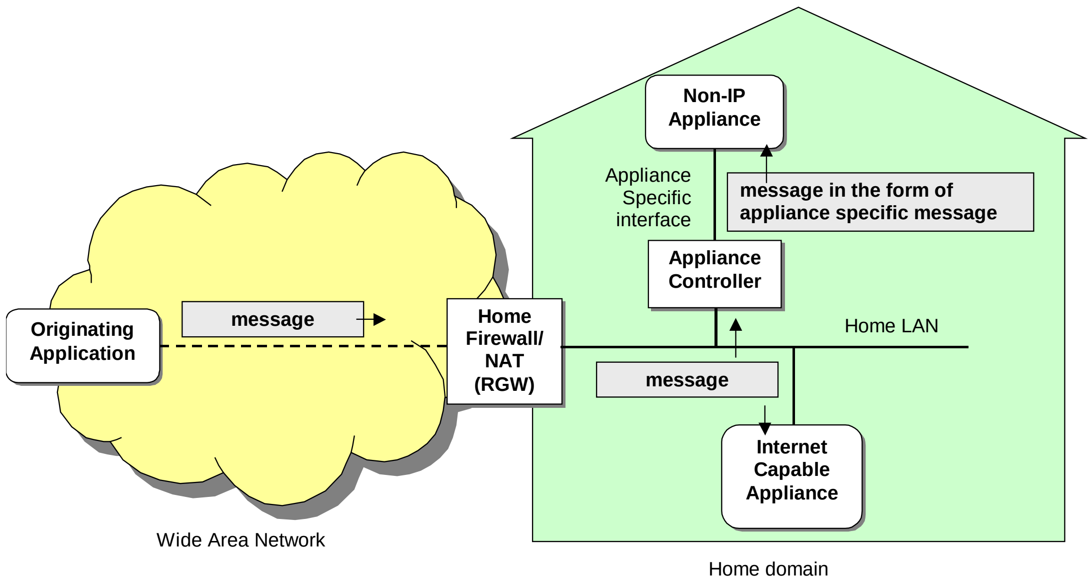

Figure 2: Network Architecture: Direct Communication with the Home Domain

The simplest architecture of the three examples allows a client application to directly connect to and interact with networked devices in the home domain. The wide area network is used to carry messages from a Client application to the Home Firewall/NAT. Once authenticated, they are allowed through the firewall. Inside the home domain messages are transported over the Home LAN(s) to the appropriate Networked Device. Devices may either be 'IP capable' (they can process the incoming messages themselves), or Non-IP-capable appliances. Non-IP-capable appliances require Appliance controllers to map IP control requests to the specific protocol and physical system that the Appliance can understand. This example is illustrated in Figure 2.

3.2 Example 2: Communication Via Gateway Proxy

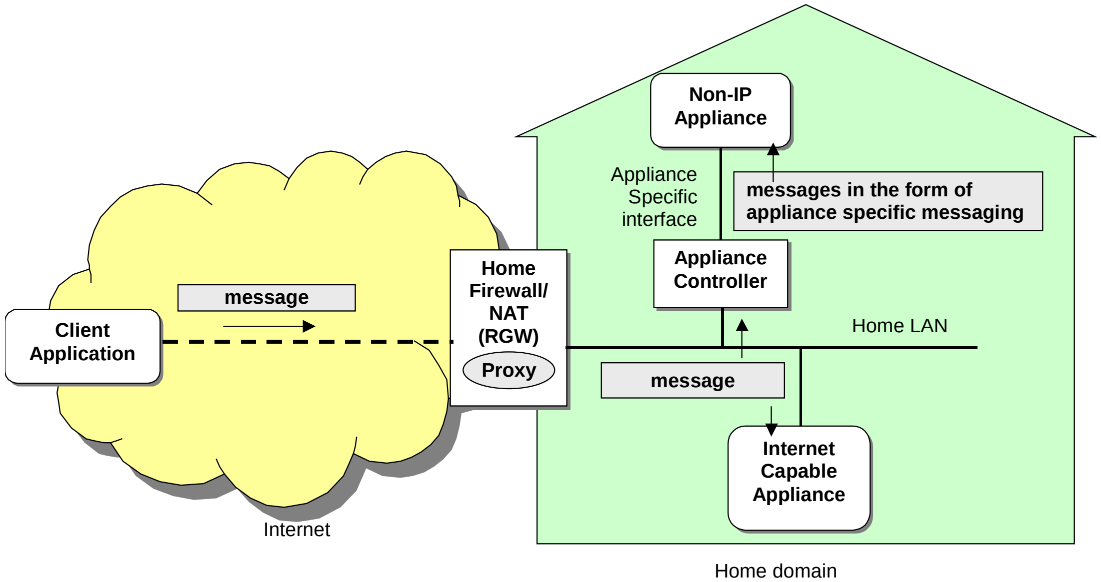

Figure 3: Network Architecture: Communication via Gateway Proxy

In many cases, it will not be possible or desirable to allow client applications to directly access and control a user's Networked Appliances. This situation can occur for a number of reasons including:

- The Appliance's IP address cannot be determined because it is behind a NAT.
- The Appliance does not have an IP address.
- It is desired to keep visibility of in-home devices to a minimum.
- The Home Firewall/NAT filters and rejects communications from unknown sources for security reasons.
- Finer-grained security is desired (i.e. authentication and access control on a per device/message basis)

In this case, control messages from Client Applications must first be sent to a 'trusted' Proxy, which has visibility into the home. This architecture is illustrated in Figure 3. All communications between the Proxy and the Home Firewall/NAT are assumed to be secure. In this case the Proxy is located in the home domain's gateway device. The proxy can provide a number of functions including:

- Authenticate and authorize each message/request.
- Address mapping/resolution for NAs within the home domain.
- Act as secure proxy to Home Firewall/NAT (RGW) for communications to the outside world.
- Provide NA mobility and tracking service.
- Provide message protocol mapping service for client applications. By taking this approach, a variety of client applications can be supported for remote controlling NAs.
- Act as charging point for services.

3.3 Example 3: Communication Via (Service Provider) Proxy

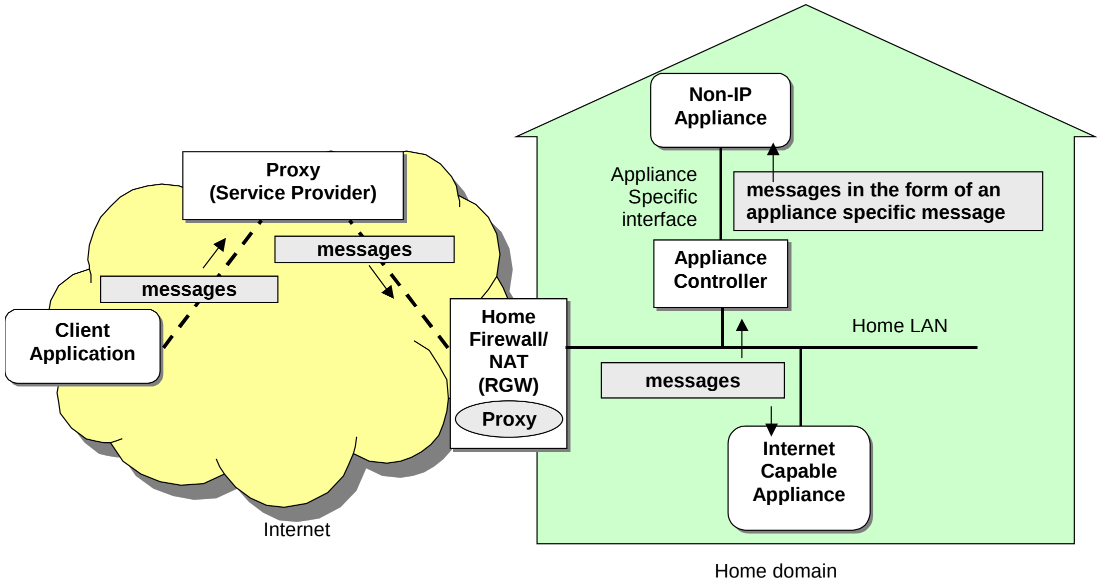

Figure 4: Network Architecture: Communication via (Service Provider) Proxy

The previous case (Proxy in the Gateway device) requires a lot of functionality in this Proxy, which may place onerous requirements on the gateway device in terms of performance, memory, etc. Since gateway devices may not have the resources required to support the Proxy functionality previously described, much of the functionality could be "outsourced" to an external proxy (in the Service Provider's network for example). This external Proxy could provide all the functionality described in the previous section and, if a secure connection (e.g., IPsec tunnel) existed between the external proxy and the gateway proxy, the gateway proxy would only be required to forward the SIP messages to the appropriate UA. The split of functionality in the gateway proxy does not have to be an "all or nothing" decision, but could be split equally (or unequally) between the two Proxies. This architecture is depicted in Figure 4.

The advantages of this approach would be;

- Administration of the SIP Proxy could be done centrally, avoiding a distributed systems issue.
- If the local link to the home were to fail, functionality would still be available through the Service Provider Proxy from the wide area, to re-direct to another home, for example.
- Configuration of the RGW is kept to a minimum, although it may still be necessary to perform some limited configuration such as the creation of an IPsec tunnel.
- The costs of making the Service Provider fault tolerant can be amortized across multiple homes.

4 Example SIP Architecture

This section describes an example of functional blocks that may be used in the SIP Architecture for supporting networked appliances. This is followed by typical ways in which these functional blocks can be realized in a practical system.

4.1 Functional/Logical Elements

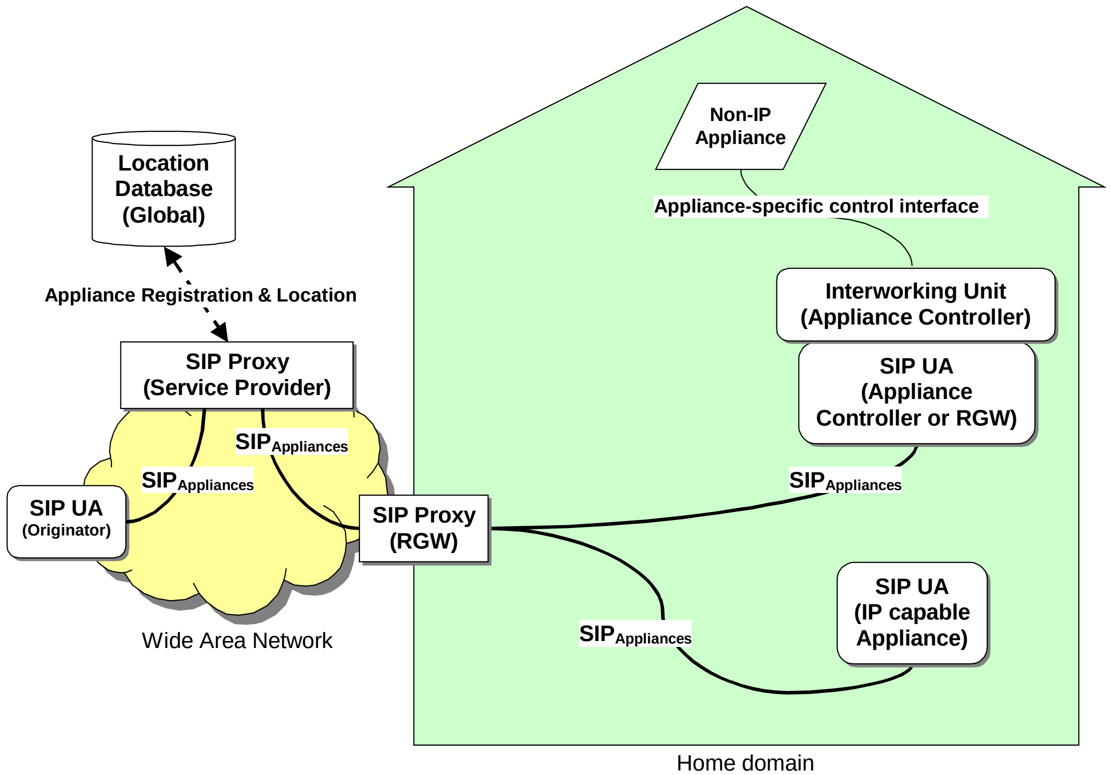

Figure 5: Example Functional Architecture for Remote Communication with Networked Appliances

This section outlines an example functional architecture for remote controlling Networked Appliances. The functional architecture (based on the Messaging via Proxy architecture of Figures 3 and 4) is illustrated in Figure 5. The functional entities can be distributed across different physical network elements (see Section 3) and this document only describes some of the possible distributions. The key functional entities are now described:

- *SIP UA (Originating domain)*: This SIP UA is used by the Originating application to generate and send Appliance messages (MESSAGE) to the SIP Proxy hosted by either the Service Provider or the Home RGW.
- *SIP Proxy (Service Provider domain)*: The SIP proxy in the Service Provider domain resolves the address of the Appliance to be communicated with (including appropriate Home domain RGW) by lookup in a Location Database. The SIP proxy forwards Appliance messages from the Client SIP UA to the SIP Proxy in the Home Domain RGW or, via a secure connection, directly to the UA in the target device.
- *Location Database (Service Provider domain)*: The Service Provider Location Database contains location information for all registered appliances within the home domains. This database is populated with information gathered by the Service Provider SIP Proxy.
- *SIP Proxy (Home Domain RGW)*: The SIP Proxy in the Home Domain RGW provides the Gateway between Appliances in the Home Domain and entities in the wide area. Other RGW functions such as Firewall and NAT will likely be co-located with the RGW SIP Proxy.
- *SIP UA (Appliance Controller/RGW)*: This SIP UA terminates SIP Appliance messages from the Originating Application SIP UA. It retrieves messaging information from the SIP message and passes this information to the Interworking Unit. This SIP UA may reside in either the RGW or the Appliance Controller. The mapping from (logical) SIP UA to Appliance Controller is 1:N.
- *Interworking Unit (Appliance Controller)*: The Interworking Unit maps the Appliance message carried in the payload of the SIP message onto the Appliance-specific protocol.

- *SIP UA (IP capable appliance)*: This SIP UA resides in IP (SIP) capable NAs. It terminates SIP Appliance Control messages from the Originating Application SIP UA, and retrieves the Appliance Control information for the Appliance application, acting on it directly without any requirement for an intervening interworking unit.
- *Non-IP Capable Appliance*: Non-IP Appliances are controlled by the Appliance Controller and require Interworking Units to communicate/interact with the Client Applications.

The key interfaces in Figure 5 are:

- SIP $_{\text{Appliances}}$: This interface represents IETF SIP [10] with the MESSAGE method for communicating with Networked Appliances.
- Appliance Registration and Location: Any appropriate database update and lookup protocol may be used to access the Location Databases. Examples of such a protocols are LDAP [11] and SLP [12].
- Appliance-Specific interfaces: There are numerous home-networking technologies currently available [1-6]. It is the function of the Interworking Unit to map from SIP to the protocols of the specific technology of the target device.

4.2 Physical Realization

A possible physical realization of the theoretical system described above, following the same organization as in Figure 5, is shown in Figure 6 below;

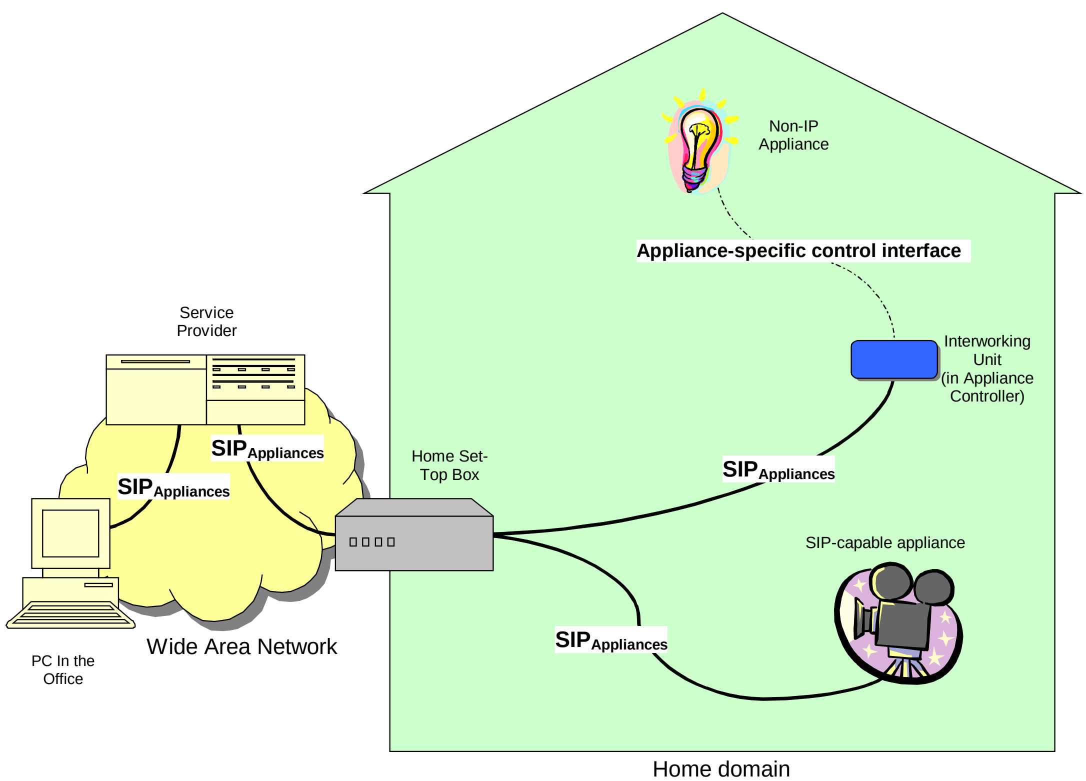

Figure 6. One Possible Physical Realization of Functional Architecture

In this diagram, the Originating SIP UA is on the PC in the user's office. They originate a message from this machine to manipulate an object within the home – perhaps a video camera or a light, for example. This message is forwarded, using standard SIP techniques, to the Service Provider who is responsible for the home, who sends it on to the Set Top Box (STB) (or RGW, or Cable Modem or ADSL Modem or whatever appropriate edge of home technology is deployed). The STB sends the message on either directly to the SIP-capable device (which will tend to be devices with higher capabilities such as video recorders and home audio-video equipment, HAVi notwithstanding), or indirectly via an Interworking Unit, which will be part of an appliance controller.

The intention with the physical realization is to avoid the user needing to be aware of the level and sophistication of the communication that is being performed on their behalf.

5 Application Scenarios

The following sub-sections provide examples of how SIP could be used for inter-domain networking of appliances. Note, not all SIP header fields (e.g. CSeq, Call-ID, and Content-length) are included in the following examples. For the sake of brevity, only the header fields of particular interest have been included. Also note that DMP has not yet been standardized and the DMP examples should be considered to be for illustration only.

We do not comment on the practical validity of any given scenario.

5.1 Simple Access to Home from Work

In this scenario the user wishes to turn on a lamp within their home from their office PC.

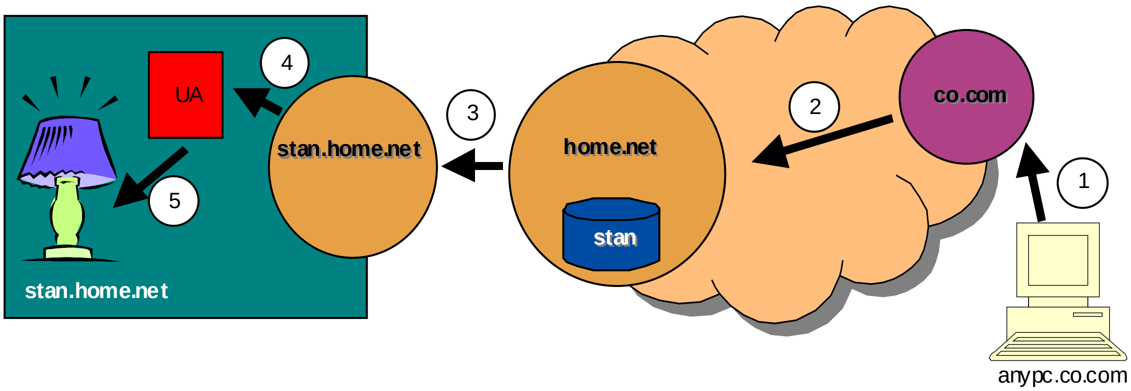

Figure 7. Simple Access to Home from Work

The SIP messages for the remote control (e.g., from the office) of a NA within the home (e.g., a lamp) are shown below. Please note that the SLP URL information will be encoded and optionally encrypted for privacy, but is shown un-encrypted between [and] for clarity in the examples in this section. In the following example we assume that the UA in stan.home.net has registered with stan.home.net and that that information has also propagated to home.net.

```
1. MESSAGE sip:[slp://d=lamp,r=bedroom,u=stanm]@home.net SIP/2.0
   From: sip:stan@co.com
   To: sip:[slp://d=lamp,r=bedroom,u=stanm]@home.net
   Via: anypc.co.com
   Content-function: render
   Content-type: application/dmp
   <command><turn>On</turn></command>
```

The co.com Proxy does an SRV look-up in DNS [14] for [slp://d=lamp,r=bedroom,u=stanm]@home.net to find the name of the SIP server for the destination domain and gets a value of home.net. This implies that the user/service name must be unique within the service provider's domain when an SRV record points to a service provider's Proxy.

```
2. MESSAGE sip:[slp://d=lamp,r=bedroom,u=stanm]@home.net SIP/2.0
   From: sip:stan@co.com
   To: sip:[slp://d=lamp,r=bedroom,u=stanm]@home.net
   Via: co.com
```

```
Via: anypc.co.com
Content-function: render
Content-type: application/dmp
<command><turn>On</turn></command>
```

3. MESSAGE sip:[*slp://d=lamp,r=bedroom,u=stanm*]@stan.home.net SIP/2.0
   From: sip:stan@co.com
   To: sip:[*slp://d=lamp,r=bedroom,u=stanm*]@home.net
   Via: home.net
   Via: co.com
   Via: anypc.co.com
   Content-function: render
   Content-type: application/dmp
   <command><turn>On</turn></command>

4. MESSAGE sip:[*slp://d=lamp,r=bedroom,u=stanm*]@ua.stan.home.net SIP/2.0
   From: sip:stan@co.com
   To: sip:[*slp://d=lamp,r=bedroom,u=stanm*]@home.net
   Via: stan.home.net
   Via: home.net
   Via: co.com
   Via: anypc.co.com
   Content-function: render
   Content-type: application/dmp
   <command><turn>On</turn></command>

5. <Action!!!>

5.2 Access with Re-direction

In this case the lamp from `stan.home.net` has temporarily been moved to `simon.home.net`. To accommodate the change, a re-direction is added into the home.net proxy . The SIP Messages for this scenario are shown below. In this example, we assume that the lamp from stan.home.net has been unregistered with both stan.home.net and home.net and that the UA in simon.home.net has performed the appropriate registrations.

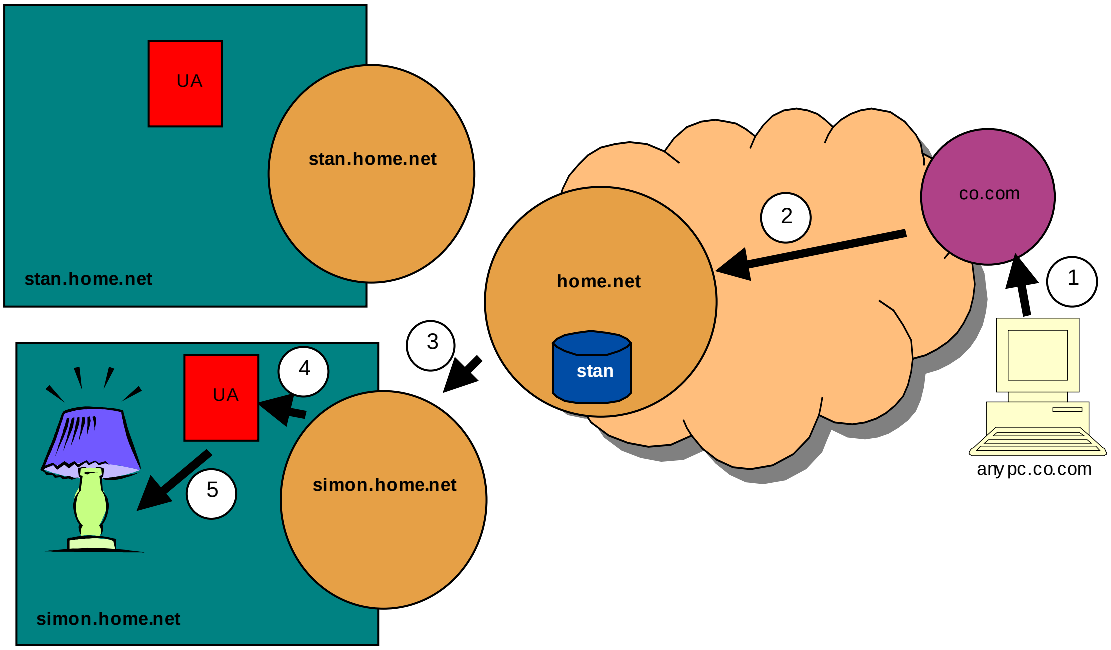

Figure 8. Access with Re-direction

1.
```
MESSAGE sip:[slp://d=lamp,r=bedroom,u=stanm]@home.net SIP/2.0
From: sip:stan@co.com
To: sip:[slp://d=lamp,r=bedroom,u=stanm]@home.net
Via: anypc.co.com
Content-function: render
Content-type: application/dmp
<command><turn>On</turn></command>
```

2.
```
MESSAGE sip:[slp://d=lamp,r=bedroom,u=stanm]@home.net SIP/2.0
From: sip:stan@co.com
To: sip:[slp://d=lamp,r=bedroom,u=stanm]@home.net
Via: co.com
Via: anypc.co.com
Content-function: render
Content-type: application/dmp
<command><turn>On</turn></command>
```

The home.net proxy does a look-up and notices that Stan's bedroom lamp is now in Simon's spare room. Therefore, the Request-URI now points to the spare room in Simon's house.

3.
```
MESSAGE sip:[slp://d=lamp,r=spareroom,u=stanm]@simon.home.net SIP/2.0
From: sip:stan@co.com
To: sip:[slp://d=lamp,r=bedroom,u=stanm]@home.net
Via: home.net
Via: co.com
Via: anypc.co.com
Content-function: render
Content-type: application/dmp
<command><turn>On</turn></command>
```

```
4. MESSAGE sip:[slp://d=lamp,r=bedroom,u=stanm]@ua.simon.home.net SIP/2.0
   From: sip:stan@co.com
   To: sip:[slp://d=lamp,r=bedroom,u=stanm]@home.net
   Via: simon.home.net
   Via: home.net
   Via: co.com
   Via: anypc.co.com
   Content-function: render
   Content-type: application/dmp
   <command><turn>On</turn></command>

5. <Action!!!>
```

5.3 Checking the Central Heating from the Office

In this scenario, Stan is at work and wants to check the temperature of the downstairs zone of his two-zone heating and cooling system in his home.[1]

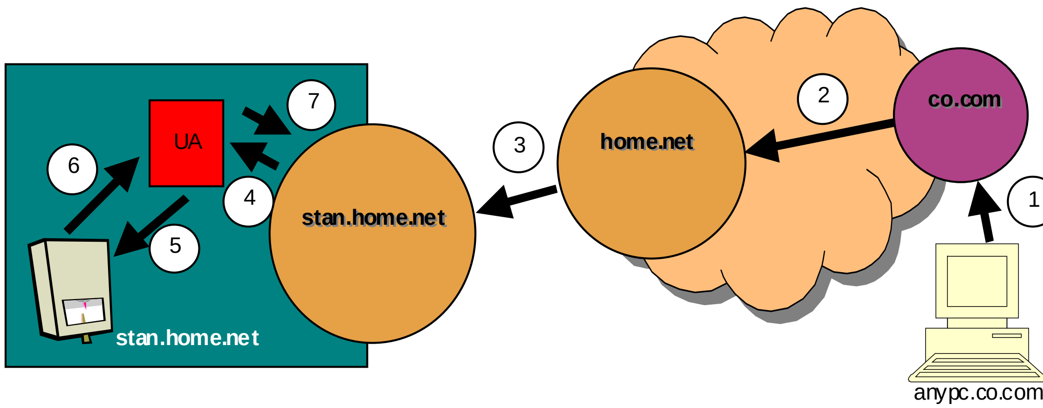

Figure 9. Checking the House Temperature from Work

```
1. MESSAGE sip:[slp://d=thermostat,r=downstairs,u=stanm]@home.net SIP/2.0
   From: sip:stan@co.com
   To: sip:[slp://d=thermostat,r=downstairs,u=stanm]@home.net
   Via: anypc.co.com
   Content-type: application/dmp
   <query>Temperature</query>

2. MESSAGE sip:[slp://d=thermostat,r=downstairs,u=stanm]@home.net SIP/2.0
   From: sip:stan@co.com
   To: sip:[slp://d=thermostat,r=downstairs,u=stanm]@home.net
   Via: co.com
   Via: anypc.co.com
   Content-type: application/dmp
   <query>Temperature</query>

3. MESSAGE sip:[slp://d=thermostat,r=downstairs,u=stanm]@stan.home.net
   SIP/2.0
   From: sip:stan@co.com
   To: sip:[slp://d=thermostat,r=downstairs,u=stanm]@home.net
   Via: home.net
```

[1] He has been trying to determine the right combination of upstairs and downstairs thermostat settings to get the house to the desired temperature. Stan is an engineer.

```
Via: co.com
Via: anypc.co.com
Content-type: application/dmp
<query>Temperature</query>
```

4. MESSAGE sip:[*slp://d=thermostat,r=downstairs,u=stanm*]@ua.stan.home.net SIP/2.0
From: sip:stan@co.com
To: sip:[*slp://d=thermostat,r=downstairs,u=stanm*]@home.net
Via: stan.home.net
Via: home.net
Via: co.com
Via: anypc.co.com
Content-type: application/dmp
<query>Temperature</query>

5. <Query!!!>

6. Current temperature reading returned

7. 200 stan@co.com
From: sip:[*slp://d=thermostat,r=downstairs,u=stanm*]@home.net
To: sip:stan@co.com
Via: stan.home.net
Via: home.net
Via: co.com
Via: anypc.co.com
Content-type: application/dmp
<temperature>65F</temperature>
(this message is forwarded to request originator on anypc.co.com)

The main differences between this example and the example in section 5.1 are:

- a different body to issue a query for the temperature instead of a command to turn on the light
- the removal of the Content-function header field since “render” is the default value for this header field in a MESSAGE method (this is optional and it could have been left in).

After Stan receives the temperature, he could issue another command to set the desired temperature. This would use a similar `MESSAGE` with a different body content.

5.4 Answering the Front Doorbell from the Car

Stan is riding with Dave in Dave’s car and remembers that he was expecting a service person to come and fix the dishwasher and he does not have his Web phone. He asks to borrow Dave’s phone and sends a message to his service provider to notify him if someone “rings” the doorbell.[2] When the service person “rings” the doorbell (and authenticates themselves with their ID badge), a message is sent to Dave’s Web phone for Stan indicating that the service person is at the front door. After verifying that it is indeed a person from the right company, Stan issues a command to unlock the front door and let the person in.

[2] We assume that Stan has to enter some authentication code that will be attached to the message to verify that it is Stan and not Dave that is requesting this. Dave is an engineer too.

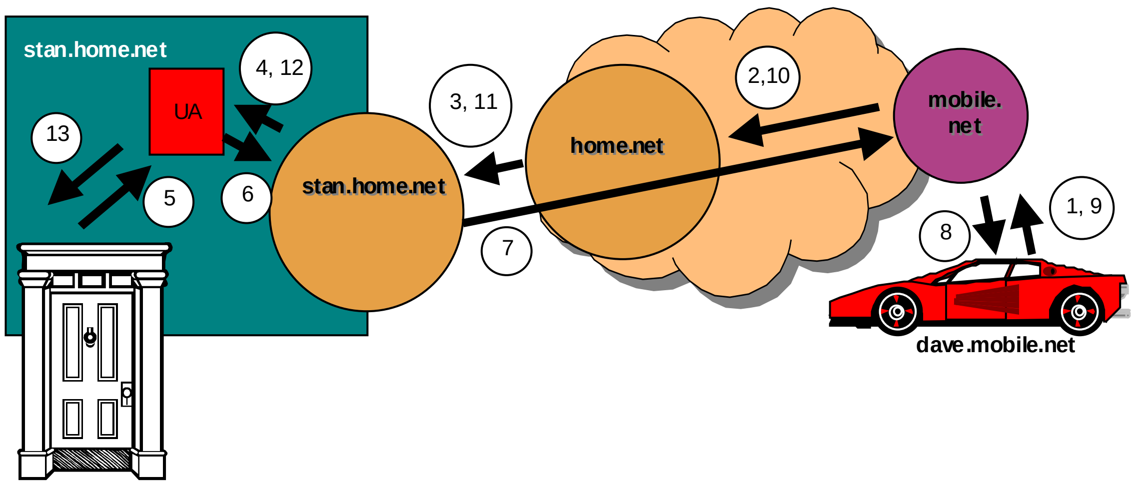

Figure 10. Answering the Front Door from a Car

1. SUBSCRIBE sip:[*slp://d=door,r=front,u=stanm*]@home.net SIP/2.0
   From: sip:stanm@dave.mobile.net
   To: sip:[*slp://d=door,r=front,u=stanm*]@home.net
   Via: dave.mobile.net
   Content-type: application/dmp
   <event>ring</event>

2. SUBSCRIBE sip:[*slp://d=door,r=front,u=stanm*]@home.net SIP/2.0
   From: sip:stanm@dave.mobile.net
   To: sip:[*slp://d=door,r=front,u=stanm*]@home.net
   Via: mobile.net
   Via: dave.mobile.net
   Content-type: application/dmp
   <event>ring</event>

3. SUBSCRIBE sip:[*slp://d=door,r=front,u=stanm*]@stan.home.net SIP/2.0
   From: sip:stanm@dave.mobile.net
   To: sip:[*slp://d=door,r=front,u=stanm*]@home.net
   Via: home.net
   Via: mobile.net
   Via: dave.mobile.net
   Content-type: application/dmp
   <event>ring</event>

4. SUBSCRIBE sip:[*slp://d=door,r=front,u=stanm*]@ua.stan.home.net SIP/2.0
   From: sip:stanm@dave.mobile.net
   To: sip:[*slp://d=door,r=front,u=stanm*]@home.net
   Via: stan.home.net
   Via: home.net
   Via: mobile.net
   Via: dave.mobile.net
   Content-type: application/dmp
   <event>ring</event>

5. (Doorbell Rings! Credentials established.)

```
6. NOTIFY stanm@dave.mobile.net SIP/2.0
   From: sip:[slp://d=door,r=front,u=stanm]@stan.home.net
   To: stanm@dave.mobile.net
   Via: ua.stan.home.net
   Content-type: application/dmp
   <event>ring</event>
   <identity>Maytag Repairman</identity>

7. NOTIFY stanm@mobile.net SIP/2.0
   From: sip:[slp://d=door,r=front,u=stanm]@stan.home.net
   To: stanm@dave.mobile.net
   Via: stan.home.net
   Via: ua.stan.home.net
   Content-type: application/dmp
   <event>ring</event>
   <identity>Maytag Repairman</identity>

8.  NOTIFY stanm@dave.mobile.net SIP/2.0
   From: sip:[slp://d=door,r=front,u=stanm]@stan.home.net
   To: stanm@dave.mobile.net
   Via: mobile.net
   Via: stan.home.net
   Via: ua.stan.home.net
   Content-type: application/dmp
   <event>ring</event>
   <identity>Maytag Repairman</identity>

9. (User alerted and decides to unlock the door)
   MESSAGE sip:[slp://d=door,r=front,u=stanm]@home.net SIP/2.0
   From: sip:stan@dave.mobile.net
   To: sip:[slp://d=door,r=front,u=stanm]@home.net
   Via: dave.mobile.net
   Content-type: application/dmp
   <command>unlock</command>

10. MESSAGE sip:[slp://d=door,r=front,u=stanm]@home.net SIP/2.0
   From: sip:stan@dave.mobile.net
   To: sip:[slp://d=door,r=front,u=stanm]@home.net
   Via: mobile.net
   Via: dave.mobile.net
   Content-type: application/dmp
   <command>unlock</command>

11. MESSAGE sip:[slp://d=door,r=front,u=stanm]@stan.home.net SIP/2.0
   From: sip:stan@dave.mobile.net
   To: sip:[slp://d=door,r=front,u=stanm]@home.net
   Via: home.net
   Via: mobile.net
   Via: dave.mobile.net
   Content-type: application/dmp
   <command>unlock</command>

12. MESSAGE sip:[slp://d=door,r=front,u=stanm]@ua.stan.home.net SIP/2.0
   From: sip:stan@dave.mobile.net
   To: sip: sip:[slp://d=door,r=front,u=stanm]@home.net
   Via: stan.home.net
   Via: home.net
```

```
Via: mobile.net
Via: dave.mobile.net
Content-type: application/dmp
<command>unlock</command>
```

13. `<Unlock!!!>`

5.5 Establishing a Session with a Networked Appliance

The previous application scenarios all involved session-less communications. It may be necessary to establish sessions with networked appliances. For example, consider an Internet Alarm Clock service where your wake-up time is adjusted based on current weather, road, and traffic conditions. If the network-based alarm clock service provider adjusts your wake-up time, you would want to know why. So, the it would be convenient for the alarm clock service provider to establish an audio session with your alarm clock and "play" a message containing the reason(s) for the adjusted wake-up time (and maybe include the current weather, top news stories, etc.). The message flow for this scenario (depicted in Figure 11) would be as follows:

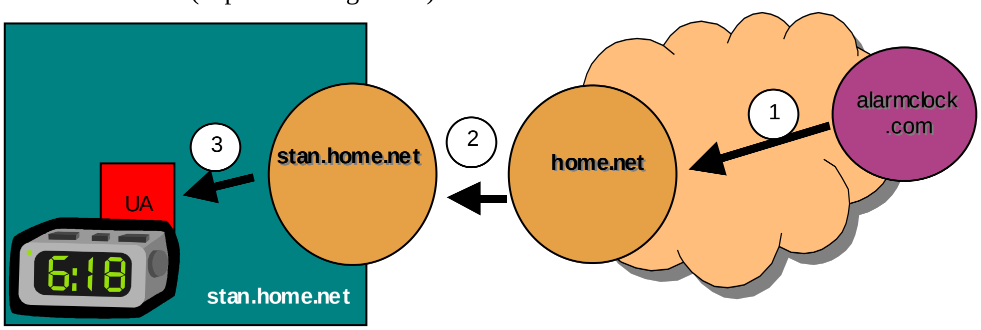

Figure 11. Network-based Alarm Clock Service SIP Message Flows

1. `INVITE sip:[`*`slp://d=lamp,r=bedroom`*`]@home.net SIP/2.0`
   `From: sip:announcement@alarmclock.com`
   `To: sip:[`*`slp://d=lamp,r=bedroom`*`]@stan.home.net`
   `Via: alarmclock.com`
   `Content-type: application/sdp`
   [SDP Parameters for uni-directional RTP stream]

2. `INVITE sip:[`*`slp://d=lamp,r=bedroom`*`]@stan.home.net SIP/2.0`
   `From: sip:announcement@alarmclock.com`
   `To: sip:[`*`slp://d=lamp,r=bedroom`*`]@stan.home.net`
   `Via: home.net`
   `Via: alarmclock.com`
   `Content-type: application/sdp`
   [SDP Parameters for uni-directional RTP stream]

3. `INVITE sip:[`*`slp://d=lamp,r=bedroom`*`]@ua.stan.home.net SIP/2.0`
   `From: sip: announcement@alarmclock.com`
   `To: sip: sip:[`*`slp://d=lamp,r=bedroom`*`]@stan.home.net`
   `Via: stan.home.net`
   `Via: home.net`
   `Via: alarmclock.com`
   `Content-type: application/sdp`
   [SDP Parameters for uni-directional RTP stream]

A response is then returned to the alarm clock service provider with the alarm clock's RTP parameters and an audio RTP stream is initiated (sent to the alarm clock).

6 Security Considerations

Security is a primary concern, especially since this system is intended for deployment in individual user's homes. We briefly outline here the security threats that are applicable to the protocol described herein, and which must be considered in any implementation of this protocol. We also discuss some possible approaches to deal with the security issues raised.

We point out that there are security concerns raised by this document, which are, however, outside the scope of the present discussion. In this discussion, *we only consider security of the underlying SIP protocol* (as it relates to networked appliances). We also assume (physical) security of the user's home; in other words, our analysis assumes that only on the Internet can an adversary eavesdrop on SIP messages, forge SIP messages, and modify SIP messages.

6.1 Security Threats

6.1.1 Importance of Security

Security concerns must be paramount when designing a system which allows remote access to a home. A forged (generic) SIP message will usually be no more than an annoyance, but a forged command to turn on an appliance within someone else's home is a potential disaster. Therefore, methods for verifying authenticity MUST be provided in any system that allows remote home access. In particular, all (SIP) communication to the home MUST be authenticated by the home RGW/firewall, and verified to have originated from either an authorized individual (in the case of direct communication from user to the home, as in Figure 3) or an authorized Service Provider Proxy (in the case of communication via proxy, as in Figure 4). In the second case, the Service Provider Proxy also MUST have verified that communication originated from an individual authorized to communicate with the home.

Privacy will be a concern for many users, particularly since messages contain information about the existence and use of appliances within their home. Thus, implementations MUST provide methods for private (i.e., encrypted) communication. However, users may choose to opt out (but they MUST NOT be allowed to opt out of checking proper authentication, as detailed above).

Security of message responses is important as well. These responses may eventually contain status information (i.e., the current temperature of the house, and faked 100 and 200 response messages can also cause problems.

6.1.2 Privacy

In general, a user will not want a passive eavesdropper to be able to determine the content of a message. This applies not only to the body of the message (which will contain the command to be executed), but also to header fields which may leak information about the devices one owns. For example, the To: header field will contain a URL of the addressed entity which, as proposed in Section **Error! Reference source not found.**, will indicate the device type and location. A user may not want anyone to know whether he owns a television, and he certainly will not want anyone to know the room in which the television is located.

If the underlying architecture being implemented provides direct control of the home domain, no intermediate proxies need be trusted (with respect to privacy) because appropriate fields can be suitably encrypted. However, if the underlying architecture is Communication via Proxy (Figure 4), an assumption of trust of the intervening Service Provider Proxy is inevitable.

REGISTER messages may also require encryption, if registration takes place in a network outside the home (as it would in the case of Communication via Proxy (Figure 4)).

6.1.3 Authentication

From the user's point of view, an even more important concern is proper authentication of SIP messages. Note that messages in either direction (from user to home or from home to user) require authentication. The authentication requirement on messages from user to home is obvious, since these are messages which will effect certain actions inside the home. However, 100 and 200 response messages from the home to the user must also be authenticated, lest an adversary insert a fake Acknowledge or Confirmed message when in fact the original message was never received. Also, responses may eventually include status information, such as the temperature of the house or whether the alarm system is turned on.

In addition to authentication of MESSAGE messages, REGISTER messages may also potentially require authentication. However, if registration is done with the home RGW (as would be the case when direct communication (Figure 3) is assumed), cryptographic solutions are not necessary (due to the physical security of the home network). When registration takes place in an outside network (as when Communication via Proxy is used), these messages must be authenticated.

Authentication of messages will prevent fabrication, modification, and masquerading. An issue not directly resolved by authentication is replay attacks. Replay attacks can be defended against in two ways. One simple way to do this is to check for repeated messages; this can be done, for example, by checking the Timestamp: or Cseq: fields against previously stored messages. However, there is a limit to the number of previously used Timestamps can be stored, and this leaves open the possibility of replaying a (very) old message. A more robust solution is to check for old messages by comparing the Timestamp: field to the current system time. Note that the Timestamp: field cannot be maliciously modified because of the assumed message authentication being used. In this case, however, some synchronization of clocks is assumed

6.2 Solutions

Methods for achieving privacy and authentication for (generic) SIP messages appear in [10] and, in general, these methods apply to the case of addressing NAs as well. However, we highlight a few important differences between general SIP security and the specific case of remote home access.

6.2.1 Security Using a Shared Secret

For general SIP security, some form of public-key technology must be employed to provide security [10]. In the case of remote access to NAs within the home, however, shared secrets can be used to provide privacy and authentication. The are two primary reasons for this difference: first, general SIP communication can potentially occur between any two parties, while in the case of remote access to the home a one-to-one (or few-to-one) correspondence exists between authorized users and the homes to which they will be communicating. Second, general SIP communication frequently occurs between parties who have had no prior contact, and therefore no opportunity to generate a shared secret. In the case of home access, however, users will have the opportunity to designate a shared secret for use in their communication with the home. The secret may be shared either with the home RGW/firewall (in the case of direct communication from user to the home, as in Figure 1) or with the Service Provider Proxy (in the case of Communication via Proxy, as in Figure 4).

In general, secret-key methods are preferable to public-key methods due to both their higher level of security and increased efficiency.

Note that in some cases, public-key methods may be preferable. It may be advantageous to provide implementations for both. Implementation details will depend on outside factors including the requirements of the Service Provider, initial installation, billing, record keeping, supported remote access methods, and future upgradability.

6.2.2 End-to-End Encryption

The SIP RFC [10] describes two methods of achieving privacy: encrypt end-to-end or hop-by-hop. In the particular setting of remote access to the home, we recommend that only end-to-end encryption be used. End-to-end encryption is certainly more efficient, and if the user and the home RGW/firewall (or Service Provider Proxy) share a secret key, there is no need to rely on hop-by-hop encryption. Furthermore, hop-by-hop encryption requires trust in every proxy along the message path, while end-to-end encryption only requires trust in the final UA which performs the decryption (either the home RGW/firewall or the Service Provider Proxy).

We distinguish two cases. First, as in Figure 3, we may have direct communication from the user to the home where decryption and verification of authenticity are done by the home RGW/firewall. In this case, the original message from the user can be encrypted and authenticated using the secret key shared by the user and the home. In the second case, as in Figure 4, communication is via a Service Provider Proxy. In this scenario, the message from the user is first encrypted and authenticated using a key shared between the user and the Service Provider Proxy. Upon receipt of the message, the Service Provider Proxy verifies the authenticity of the message and decrypts. Then, the message is authenticated and encrypted using a key shared between the Service Provider Proxy and the home and forwarded to the home (note that this step may also be handled by the establishment of a secure IPSec tunnel between the Service Provider Proxy and the home). The forwarded message is authenticated (as having come from the Service Provider Proxy) and decrypted by the home RGW/firewall before being allowed inside the home. (Although technically this is no longer end-to-end encryption, we prefer to think of it that way since there will be many intermediate proxies along the path from user to Service Provider Proxy, and the message is not re-encrypted for each hop.)

6.2.3 Encryption of the To: field

One major difference between this document and [10] is that the To: header field now contains potentially sensitive information (such as device names and locations) which should be encrypted. The body of the message (and appropriate header fields) should be encrypted as detailed in [10] (although possibly using private-key technology). Encryption of the To: field should take place separately from encryption of the body of the message. Since the entire contents of the To: field cannot be encrypted (this information is used for routing), only the portion to the left of the "@" (the entity information) should be encrypted.

At first glance, this might appear problematic since routing is done based on entity information contained in the To: field. We show that this problem is easily avoided. Indeed, routing is done based on two components of the To: field: the entity name (appearing to the left of the "@") and the location (appearing to the right of the "@"). Information about the location component (typically domain-names) is available to every proxy in the network. On the other hand, information about specific entities is (typically) only available to a select few proxies (in particular, the home RGW/firewall when assuming direct communication from user to the home, or the Service Provider Proxy when assuming communication via proxy). Thus, for most proxies, routing will be based solely on the location component of the To: field, and they therefore have no need to examine the entity component. On the other hand, proxies that do need to see the contents of the entity component will have the decryption key available to them (since the encryption was done with the appropriate shared key). Thus, routing will proceed via the location component until the message reaches a proxy that has access to information concerning specific devices within that domain. This proxy, by construction, will also have access to the correct key for decrypting (and authenticating) the message. Upon decrypting the message, and in particular the entity component of the To: field, the proxy can correctly route the message using this additional information.

7 Conclusion

We have shown how SIP, with the newly proposed MESSAGE, SUBSCRIBE, and NOTIFY messages, plus the new MIME types, and new mechanism for encoding service information in the To: field can provide the support necessary for communication with networked appliances from the wide area. This proposal enables leveraging the existing SIP infrastructure and capabilities (e.g., hop-by-hop routing and security) for a new problem domain — networked appliances.

8 Acknowledgements

The author's would also like to thank and acknowledge the contributions of Steven Ungar, Mike Little and Christian Huitema, without whose input this work would not have been possible.

A. Author's Addresses

Stanley Moyer e-mail: stanm@research.telcordia.com
Dave Marples e-mail: dmarples@research.telcordia.com
Simon Tsang e-mail: stsang@research.telcordia.com
Jonathan Katz e-mail: jkatz@research.telcordia.com
Provin Gurung e-mail: pgurung@research.telcordia.com
Thanh Cheng e-mail: thanh@research.telcordia.com

Ashutosh Dutta e-mail: adutta@research.telcordia.com

All of the above are at;
Telcordia Technologies Inc.
445 South Street
Moristown, NJ 07960, USA.

Henning Schulzrinne
Department of Computer Science
Columbia University
M/S 0401
1214 Amsterdam Avenue
New York, NY 10027-7003
e-mail:hgs@cs.columbia.edu

B. References

[1] OSGi, www.osgi.org.

[2] UPnP, www.upnp.org.

[3] HAVi, www.havi.org.

[4] Salutation, www.salutation.org.

[5] JINI, www.jini.org.

[6] VESA Home Networking, www.vesa.org.

[7] J. Rosenberg, et al., “A Protocol for Instant Messaging based on SIP,” Internet Draft draft-rosenberg-impp-im-00.txt, June 6, 2000.

[8] J. Rosenberg, et al., “A Protocol for Presence based on SIP,” Internet Draft draft-rosenberg-impp-presence-00.txt, June 6, 2000.

[9] UPnP Device Control Protocol, www.upnp.org.

[10] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, "SIP: session initiation protocol," Request for Comments (Proposed Standard) 2543, Internet Engineering Task Force, March 1999.

[11] T. Howes, and M. Smith, "The LDAP URL Format", Request For Comments (Proposed Standard) 2255, December 1997.

[12] E. Guttman, C. Perkins, J. Veizades, and M. Day, "Service Location Protocol, Version 2", Request For Comments 2608, June 1999.

[13] E. Guttman, C. Perkins, and J. Kemp, "Service Templates and Service:Schemes", Request for Comments 2609, June 1999.

[14] A. Gulbrandsen, P. Vixie, L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)," Request for Comments 2782, February 2000.

C. Acronyms and Abbreviations

ADSL	Asynchronous Digital Subscriber Line
CHAP	Challenge-Handshake Authentication Protocol
DDP	Device Description Protocol
DMP	Device Messaging Protocol
DNS	Domain Name Service

HAVi	Home Audio/Video Interoperability
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPSec	IP Security
JINI	JINI Is Not Initials
LAN	Local Area Network
MIME	Multi-purpose Internet Mail Extension [or Multimedia ....]
NA	Networked Appliance
NAT	Network Address Translator
OSGi	Open Services Gateway Initiative
PGP	Pretty Good Privacy
RGW	Residential Gateway
SDP	Session Description Protocol
SIP	Session Initiation Protocol
SLP	Service Location Protocol
TCP	Transmission Control Protocol
UA	(SIP) User Agent
UDP	User Datagram Protocol
UPnP	Universal Plug and Play
URL	Universal Resource Locator
VESA	Video Electronics Standards Association
XML	Extensible Markup Language