

Instant Messaging and Presence for SIP Enabled Networked Appliances

Arjun Roychowdhury
Hughes Software Systems
India
archow@hss.hns.com

Stan Moyer
Telcordia Technologies
USA
stanm@research.telcordia.com

Abstract - There has been much interest in enabling networked appliances (NAs) to communicate with and control each other over the Internet. Although there are numerous networking technologies which allow networked appliances to operate within a single home environment or domain, there is currently no support for control of these devices from the Internet, or for interworking the various home networking technologies. The ability to provide such support will radically enhance the ability to provide exciting new services. The new age of interworking will converge both general devices (like PDAs, laptops) and specialized devices (like phones, washing machines) into one date network, enabling services to the end customer that were unthinkable a few years ago. This paper discusses the use of the Session Initiation Protocol (SIP) to provide wide area networking for appliances. In the first half it describes the motivation behind using SIP for such appliances. Following this, we illustrate how services can be provided to the consumer using Instant Messaging and Presence in the networked appliance domain.

A. INTRODUCTION

Network enabling home appliances is not a new field. For a few years now, various companies and bodies have come out with a multitude of technologies which enable networked appliances (NA) to talk to each other. The simplest of the lot is X.10, which allows any user to control his home devices using an X.10 controller which can send electric pulses and put on/put off or change voltage level of certain appliances in and around the house. Then came more complicated and feature rich protocols like Jini [1], UPnP [14], HAVi [3], and other such protocols which added a lot more of capabilities for appliance communication. These existing protocols are mainly designed to work in a single home environment (i.e. local area network) and do not scale for Internet-wide communications and are lacking in the necessary security mechanisms. Home appliance interworking has most been restricted to research labs and limited customer deployment. The primary reasons for the above are:

Interoperability – It is imperative that the underlying protocol used is hidden from the consumer. He should be able to buy an Internet alarm clock from his local store and buy a coffee machine from somewhere else and have the alarm clock start the coffee maker at

exactly 8:00 am every day without worrying about whether they both speak the same protocol.

Scalability – The protocol(s) in question must be able to scale across large areas. If Robert goes on a vacation to India and realizes that he has forgotten to program his VCR to record every episode of Friends, he should be able to do so without worrying about whether the underlying protocol can handle the distance. The protocol should be able to traverse multiple hops from the source to the destination, similar to IP routing and in that process not load in-between networks (thus protocols which rely solely on broadcast and multicast would be a bad idea)

Security – This is probably the single most important issue of interworking appliances. The breaches of security that can occur here are of much greater consequence than simple data piracy or data corruption. Human lives are of concern here. By breaking into a home network, it becomes possible to control every appliance that was connected to produce disastrous results. The protocol under consideration not only must be able to provide strong encryption and authentication mechanisms but it must also allow a user to selectively allow or disallow appliance sharing, based on his discretion.

Limited Services – How much bandwidth is available for home appliance communication? How much more bandwidth is required for wide area appliance communication? In the past, a significant amount of work has gone into trying to make the protocol as efficient as possible to work within strict bandwidth constraints. A lot more concentration has gone into optimization rather than feature provisions. As a result, the services provided were limited in nature and possibly not exciting enough for the consumer to consider very seriously. However, now with broadband technologies coming into the mainstream, this is less of a concern.

Currently, SIP has emerged as a protocol of choice in the VoIP space as an enabler for next-generation services. Based on HTTP, SIP is a lightweight protocol that brings along with it a host of web enabled services

to internet telephony — a space which until now was dominated by heavier and more complicated protocols that made next generation service deployment more challenging. We believe that using SIP for network appliances would enable service providers to quickly and easily introduce more value added services to end consumers.

This paper describes an approach based on SIP for application-layer communications with networked appliances. The paper is a summary of recent Internet Drafts [1][5][7][12] related to this topic (many of which the authors co-wrote). The next section provides a definition of a networked appliance. Then advantages for using SIP

B. WHAT IS A NETWORKED APPLIANCE ?

Before we proceed, let us first describe what we mean by a *networked appliance*:

Networked appliances (NAs) are regarded as networked devices which share the following characteristics:

1 *The ability to interact with the physical environment (either through sensors or actuators);*

Limited (restricted) general computational power, though the devices may possess high-computation power for specific tasks such as image processing, or speech recognition.

In general, NAs will be groups of sensors and/or actuators with a limited computational power, and networking capability. Examples of NAs include: controllable lamps, alarm clocks, SIP phones, washing machines, door bells, cameras, temperature sensors, sprinklers, and garage doors.

C. THE SIP ADVANTAGE

The following attributes of SIP make it an ideal choice for use with appliances.

Scalability – SIP is a very scalable protocol. It works well in both LAN and WAN conditions. It does not rely solely on multicast/broadcast technologies to reach a destination endpoint. It has a strong concept of **routing** (similar to HTTP routing) which enables a packet to traverse from source to destination using intermediate existing routes, hopping from one node to another till it reaches its final destination. Further SIP can work on both UDP and TCP which allows SIP based servers to scale well. This coupled with the fact that SIP allows various definitions of servers, ranging from stateless proxies (which simply forward all messages without

keeping state) to call stateful proxies (that keep state of all SIP based calls that route through them) enables service providers to select the ideal combination they need based on network load and services to be provided.

Simplicity – An important attribute for any protocol is that it must be simple to provide value added services. The protocol should enable service providers to rapidly deploy new services without getting lost in the mires of implementation. This has a direct relation to how flexible and simple a protocol is. SIP is very lightweight and is text based – implying that both message communication and formulation are simple and non-intensive in nature.

Flexibility – In SIP it is very simple to add extensions to support new features. The protocol is defined in a way that any provider can define extensions easily to the existing grammar set to add features which may not exist in the core SIP specification yet. Most importantly, the protocol also specifies mechanisms which can ensure that any new extension does not break an existing SIP aware node that may be in route and which does not understand this new extension (For example, proxies are supposed to transparently pass bodies across. Further, if it does not understand any header, it should pass it along to the next node)

Registration – In SIP it is not necessary that a calling device needs to know exactly where to locate the called device. A device can REGISTER its current location with its registrar. Bob simply specifies that he wants to talk to Peter. The exact present location of Peter will be resolved by the proxy in conjunction with the registrar to ensure that the message reaches Peter wherever he is now. Location independence is important for appliances too.

Personal Mobility – Related to the point above, SIP provides the concept of personal mobility at no extra cost. Again, this is a key service enable for appliances too. Consider an example of Bob trying to remotely control his VCR to record a program at 7pm. He does not know that his wife has moved the VCR from his den to their bedroom that very day in order to re-organize the home décor. He simply sends the request to `vcr@home.bob.net` and the residential proxy server in his house knows that `vcr@home.bob.net` is now an alias to point to `vcr@bedroom.home.bob.net`. This happened when the VCR was shifted – it automatically registered with the new location with its local registrar.

Security – SIP provides both authentication and encryption using schemes such as basic, digest and PGP to provide end-to-end security. Security is an important aspect of appliance interworking for both safety and privacy reasons.

Transport Independence – SIP does not rely on any fixed transport protocol. That essentially means that SIP messages can traverse through various heterogeneous networks. So if Bob wanted to control his home VCR from office, he did not have to bother about the fact that this SIP message might have actually traversed various intermediate networks based on TCP, UDP, SCTP and other such transport protocols. This is important since NAs may not have much CPU power and little memory and therefore only implement UDP (and not TCP).

Event Notification – SIP has recently been extended to introduce SUBSCRIBE and NOTIFY messages which enable elements to “subscribe” to certain events and be notified when they occur. For example, Bob could also tell his office laundry washing machine “Start tumble washing, and let me know when you finish so that I can come on down and pick up my clothes”. Event notification forms a very important part of Instant Messaging and Presence, which we discuss later.

Addressing – SIP uses URIs for addressing. This may be SIP URLs, HTTP URLs, Tel URL or other schemes that map into the generic URI grammar. This means that we can introduce any new addressing scheme like SLP or others by mapping it either to an existing scheme of URLs (see [7] for mapping SLP URLs to SIP URLs). We believe that the URI scheme is an extremely generic and flexible addressing mechanism that can encompass a very wide range of addressing requirements, that may well be the case for appliances.

Integration With Existing SIP service mechanisms – using SIP as a protocol enables appliances to exploit the same service rich infrastructure that SIP provides. For example, Peter could send a request saying “*Could any printer print my document please ?*” to his office proxy which could then fork this request to the various printers in the office and then allot the job to any available printer.

Session based and Non-session based communication – If Bob wants to remotely ask his close circuit TV to start transmitting the video capture in room 28 of the supermarket, he essentially needs to set up a regular receive-only session with the circuit TV which then starts to stream video to his browser QuickTime plug-in. This is no different from the regular SIP INVITE session establishment. On the other hand, Dilbert may want to simply send a command to his internet alarm clock to set a time for wakeup and forget about it. No session is required for this – this is no different from the Instant Messaging concepts in SIP. SIP allows us to communicate using both session based and non-session based transactions, which is ideal for appliances.

Business Models – Finally, since SIP and the HTTP are so similar in nature, using SIP would introduce the same possibilities of different service models that the Browser-Server technology have had success in introducing to the consumer market – that of different business models for products that were not possible without the Internet – the ASP and pay per use models. In fact, these model are already being experimented in home appliances with Electrolux introducing their “pay-per-use” washing machines (See [16])

D. INSTANT MESSAGING AND PRESENCE

Now that we have established a foundation of why SIP seems to be an appropriate protocol for appliances, we shall proceed to illustrate how one can use concepts of IM to enable services for NAs.

The authors assume that the reader is aware of the Instant Messaging extensions for SIP and their model of operation (for details, please see [8]) The applicable extensions are a new SIP message type, MESSAGE [9], and the use of the previously proposed new message types SUBSCRIBE and NOTIFY [10][11].

Basic Communication

Most often, the act of asking a device to perform a certain operation consists of sending it a message with the command embedded in the body of the message. The requisites for such an operation are: a) Sending the command to the destination using any available route, and b) Receiving an acknowledgement that the destination has received the message.

Hence, simple communication essentially consists of a 2 way transaction exchange. For the purpose of such communication, we have defined an extension to SIP methods called DO (see [5]) which allows the sender to encode a message in its body using an XML like format. This message is proxies across intermediate hops and finally reaches the destination. Using SIP processing rules, the response too traces back the same path and reaches the source. *Note*, The operation of the DO command is strikingly similar to the MESSAGE command defined in [9]. The reason for the separation of commands was mainly due to concerns within the SIP community of overloading the MESSAGE command too far. A disagreement exists as to whether instant messaging a person is equivalent in semantics to instant messaging a device. Since creating another message type does not add additional complexity for SIP Proxy or User Agent implementations, it was agreed to use a

different message type. This is still a debatable issue – however the intent of DO remains the same as that of MESSAGE and using MESSAGE instead of DO would not change any of the concepts discussed in this paper.

An example:

Bob, sitting in his office wants to start the self diagnostic check in his new internet enabled car. In case of any damage detected by the car microprocessor, he wants to be informed so that he can set up an appointment with the repairman.

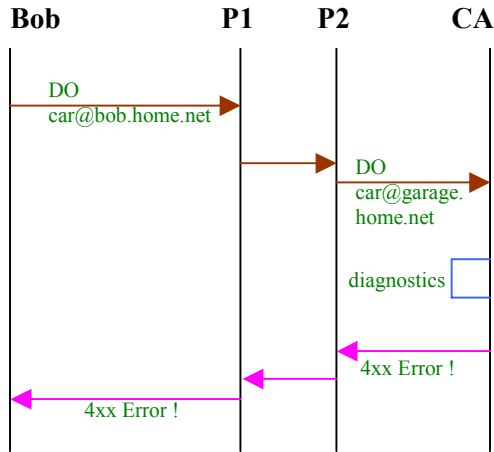


Figure 1: Example of IM

The illustration above shows a very simple case of how IM forms the basic block for communication to appliances. Here, Bob sends a DO message to his car parked in the garage. The office proxy (P1) recognizes the domain to be bob.home.net and forwards it to the proxy in his house. The home proxy (P2) now looks up the registration database and sees that the car is now inside the garage. It proxies this request to the correct location. The internet enable microprocessor in the car, on receiving this message proceeds to run the self diagnostics and sees that its wheel alignment needs to be corrected. It now reports this diagnostic error in a 4xx message back to Bob who can then proceed to take corrective action. (Note: in case the diagnostic process takes time, it can also return a 100 trying response which is not shown in the illustration).

Instant Messaging forms the very basic block for appliance communication. Other examples of IM for devices include querying your thermostat for the current temperature, sending a command to your microwave to start thawing the steak or asking your VCR at home to start recording you favorite program while you are sitting in the office trying to finish off your presentation for that important meeting the next day. This allows

both control of devices (e.g., “turn on”) and querying (e.g., “what is the temperature in the living room?”).

In all of the above, what we really need is a simple two way transaction that can traverse multiple hops and routes to where you want it to and have the results routed back to you. Isn’t IM then, just what we need ?

IM combined with Presence

We now show some powerful examples of how IM and presence combined can provide enhanced services.

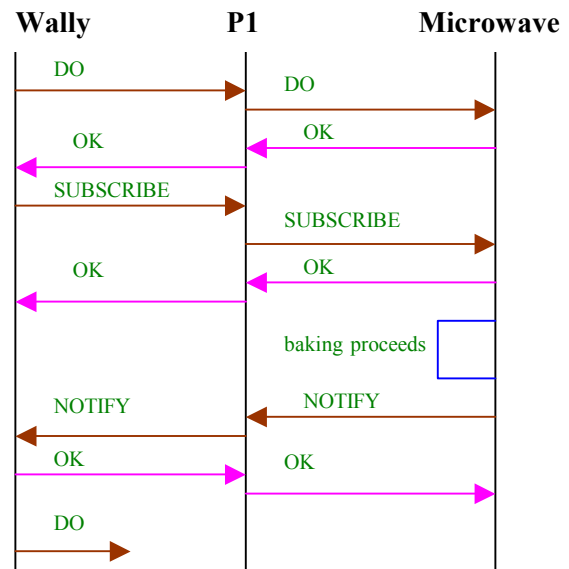


Figure 2: IM and presence

Here, Wally sends an instant message from his PC browser to his microwave asking it to start grilling the chicken at power level 5 until it is well done. He encapsulates this in a DO command body. The microwave responds by saying that the process has begun. This is a time consuming operation. Wally wants to be notified when the microwave finishes this process. So, he sends a SUBSCRIBE message to the microwave, asking it to notify him when the grilling gets done. In IM terminology, Wally is the SUBSCRIBER and the microwave is the PRESENTITY. The microwave responds to the request by sending a 200 OK. Now, Wally does not need to poll the microwave anymore. He proceeds to carry on with his usual work, while the microwave grills away. Some time later, the microwave completes the process and looks up the subscribers who have subscribed to this change of state. It realizes that wally@home.net was a subscriber and proceeds to send a notification message to this URI

about this state change. This is an asynchronous message that Wally receives on his browser through a popup which says “*Microwave has finished grilling. time taken was 2 minutes*”. Wally now proceeds to send the next set of commands to the microwave.

This is a very powerful form of communication and very useful for communications with certain types of devices. For example, notification of events that occur in the home such as a security alarm being triggered or a doorbell being pressed. Presence forms a very important part of appliance communication. It provides a mechanism for different appliances to subscribe to state information of other appliances and be notified asynchronously when any event occurs.

E. EXTENSIONS REQUIRED

As mentioned previously, SIP as specified in RFC 2543 does not support Instant Messaging for networked appliances. Some extensions/modifications are required and are described in this section.

A new method, DO [5] is required to support simple, datagram-style messaging and the SUBSCRIBE and NOTIFY extension [10] are required for asynchronous event notifications.

To better meet the naming and addressing requirements for NAs, a modified SIP URL addressing scheme is required. A structured device naming scheme (e.g. Service Location Protocol (SLP) URL) is encoded to the left of “@” sign in the To: field. This may then be encrypted to ensure privacy. For example:

`[d=lamp,r=bedroom,u=stsang]@home.net`
 where the information in the square brackets would be BASE-64 encoded (to be compliant with addressing conventions) and (optionally) encrypted for privacy.

In addition, a new payload type, specific to devices, is required. A new MIME type, called Device Messaging Protocol (DMP) [12], is proposed to carry the information required to excite NAs and which can carry responses back to the originator. There is no reason that other payloads (e.g., SOAP [13]) could not be carried too (either as another MIME type or as part of the DMP).

F. HOW DOES SIP SOLVE THE PROBLEM?

In the introduction, we listed four problems hindering the adoption of networked appliances — these problems are:

- Interoperability
- Security
- Scalability
- Limited Services (i.e., Functionality)

In this section we describe how SIP (and SIP Instant Messaging) address each of these four issues.

Interoperability

The SIP for networked appliances solution enables communication with devices in the local area without concern for the type of local device communication protocol being used. For example, given an architecture like that pictured in Figure 3, there are several types of local device communication protocols depicted in use — UPnP, HAVi, X.10, and SIP. This architecture assumes that an appliance controller exists to provide the necessary protocol translation/interworking.

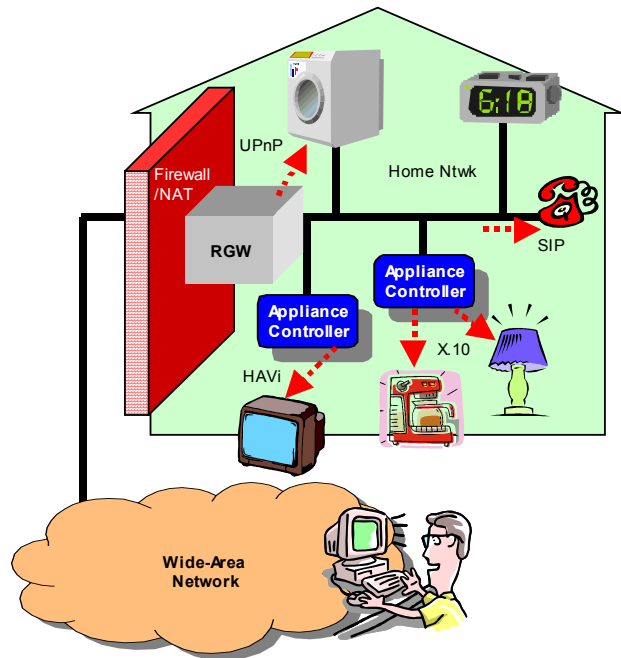


Figure 3. Example Home Networked Appliance Architecture

SIP for networked appliances uses SIP to communicate from the wide area network into the local home domain where the SIP message is then translated to the appropriate local device communication protocol (if necessary). An example of this is shown in Figure 4. The proposed Device Messaging Protocol payload is XML-based and defines a superset of all NA communication protocols — therefore it can be translated by the SIP UA associated with the NA(s) from DMP to the specific protocol for the NA.

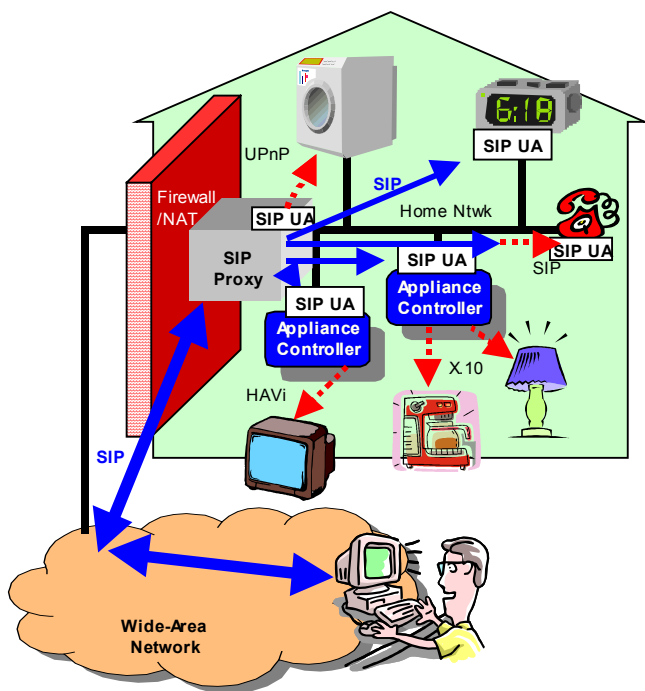


Figure 4. Use of SIP for Networked Appliances

Security

SIP for networked appliances utilizes the inherent capability of SIP to provide the security required for communication with NAs from the wide area. SIP provides an authentication mechanism (PGP is specified in the standard and Internet Drafts exist for supporting CHAP and Radius servers). If all SIP messages entering the local home domain are authenticated then an authorization check can be performed before the requested operation is executed. In addition, SIP provides a means for encrypting most portions of the message (including the payload) for privacy. However, the address is not encrypted, therefore, if the appliance name and/or description is included in the address, then that portion of the address must be encrypted (as previously described).

Scalability

As described in section C, the SIP protocol is very scalable. This property is achieved by its use of routing proxies in the network and the fact that no (or little) state is kept in the network.

Limited Services

SIP is an extensible protocol — it allows new method/message types, new types/forms of addresses, and any kind of MIME body type. Through these capabilities, extensions were designed to enable the SIP for networked appliances solution to communicate with NAs in four different ways:

Control (e.g., “turn on the coffee maker”)

Query (e.g., “how much milk is left in the refrigerator?”)

Event Notification (e.g., “tell me when my fire alarm goes off”)

Multimedia Session (e.g., “View the babysitter cam.”)

Currently those are the only four types of communication modes identified for NAs and SIP supports them all. If other modes are required in the future, SIP’s extensibility should accommodate the necessary modifications.

G. FUTURE WORK

In order to produce an inter-operable and open framework, detailed designs and specifications of the SIP message payloads for device control and device registration will be produced. The relevance of, and relationship to, other protocols — such as SOAP [11] and UPnP [14] — are being assessed and will be included into the framework described in this paper if appropriate. The possibility of interworking the SIP based service portability framework with the Open Services Gateway Initiative (OSGi [15]) framework is under investigation and the roles and capabilities that can be provided by the home gateway will also be evaluated.

H. SUMMARY

We have shown how SIP with Instant Messaging extensions supports application-layer communication with networked appliances. Many reasons for why SIP is a good solution for this problem were described. However, the real advantage to using SIP is that since SIP phones are essentially networked appliances too, SIP Instant Messaging for NAs can leverage the same SIP infrastructure deployed for Internet Telephony. Leveraging the same infrastructure greatly eases the operational (e.g., deployment, administration, and management) burden for networked appliance-based services.

Acknowledgements

The authors would like to thank S. Tsang, D. Marples, J. Katz, P. Gurung, T. Cheng, A. Dutta, S. Khurana and H. Schulzrinne for their important contributions in defining the usage of SIP with networked appliances.

References

- [1] JINI, www.jini.org.
- [2] UPnP, www.upnp.org.
- [3] HAVi, www.havi.org.
- [4] Tsang, Moyer, Marples, Schulzrinne, and Chowdhury, "Requirements for Networked Appliances: Wide- Area Access, Control, and Interworking ", Internet Draft draft-tsang-appliances-reqs-01.txt, September 2000
- [5] Tsang, Moyer, Marples, Schulzrinne, and Chowdhury, "SIP Extensions for Communicating with Networked Appliances," Internet Draft draft-tsang-sip-appliances-do-00.txt, November, 2000
- [6] 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.
- [7] Moyer, et al, "Framework Draft for Networked Appliances using the Session Initiation Protocol", Internet Draft draft-moyer-sip-appliances-framework-01.txt, November, 2000
- [8] Mark Day, J. Rosenberg "A Model for Presence and Instant Messaging", Internet Draft draft-ietf-impp-model-01.txt
- [9] J. Rosenberg, et al, "SIP Extensions for Instant Messaging", Internet Draft draft-rosenberg-impp-im-00.txt, June, 2000
- [10] A. Roach, "Event Notification in SIP," Internet Draft draft-roach-sip-subscribe-notify-02.txt, November, 2000.
- [11] J. Rosenberg, et al, "SIP Extensions for Presence", Internet Draft draft-rosenberg-impp-presence-00.txt, June, 2000
- [12] Khurana, Gurung, and Dutta, "Device Message Protocol (DMP): An XML based format for Wide Area Communication with Networked Appliances," Internet Draft, draft-khurana-dmp-appliances-00.txt, November 2000.
- [13] N. Deason, "SIP and SOAP", Internet Draft draft-deason-sip-soap-00.txt, June 2000.
- [14] Universal Plug and Play, <http://www.upnp.org>
- [15] Open Services Gateway Initiative, <http://www.osgi.org>