

Dialing for Displays: Session Initiation Protocol for Opportunistic Augmentation

John J. Barton

Stina Nylander*

Fopefolu Folowosele †

Beverly Harrison‡

IBM Almaden Research Center
650 Harry Road, San Jose CA, USA

Abstract

Opportunistic augmentation denotes connecting a personal mobile device to another device to gain a transient advantage for the user. For example, a mobile phone user might borrow a large display and keyboard from a desktop personal computer. This uniquely ubiquitous computing activity requires effective device and service discovery as well as appropriate media usable across two or more devices. In this paper we show how Session Initiation Protocol (SIP), the call signaling protocol for Voice over IP, effectively separates discovery from media-rendering selection in opportunistic augmentation. This separation improves system flexibility while allowing users or system administrators to choose the most appropriate discovery technologies for the environment. We also describe two phone-centric discovery mechanisms and demonstrate the practicality of the system by implementation and use in a test environment.

1. Introduction

Small handheld computing and communications devices now offer comparable processor speed and storage capacity to low-end personal computing equipment. However, no handheld device has a large display and text input remains a significant challenge [1].

Pierce and Mahaney propose “opportunistic annexing”, a transient use of input/output peripherals like monitors and keyboards, to overcome these limitations

[2]. The use of “docking stations” for notebook computers is an existing example of the use-model for mobile but not handheld devices. We have varied the name slightly, using “augmentation” rather than “annexing” to suggest that the mobile device needs access to I/O resources, not total control over them. Specifically, we imagine sharing a display and keyboard across multiple mobile devices. Either way, the nomadic handheld user gains functionality while retaining mobility by leveraging ubiquity of larger, less mobile I/O devices in their environment.

In this paper we explore a phone-call like approach to opportunistic augmentation, using the Session Initiation Protocol (SIP) [3]. SIP, widely known as the IETF [3] call-initiation protocol for Voice over IP, readily adapts to initiate non-audio sessions and it has been used in numerous experimental multi-modal and multi-device settings. In SIP-based opportunistic augmentation (SIP/OA), one device is phone: a personal, mobile device with limited human interaction capabilities. The other device has better affordances for human interaction. For example, the phone could run a web distributed authoring and versioning (WEBDAV) server to allow access to files in the device through a web browser on a display device. Another example would have the phone run a “virtual desktop” and the display device would act as a remote display for that desktop environment. Other media-exchange protocols may appear in the future and they can be added to a SIP-based system without altering the SIP layer.

2. Use Model

The essential logic behind SIP as a basis for opportunistic augmentation derives from the mobile user’s

*current address Swedish Institute of Computer Science

†current address Johns Hopkins University

‡current address Intel Research Seattle WA

experience. Given a small, portable device with limited human interface capability and an available device with a larger screen and keyboard, we need a simple interaction model for connecting the two devices that people would learn quickly. A phone call clearly matches these requirements; we need only teach users that calls could be placed to computers and that non-audio sessions could result.

The basic user experience for SIP/OA will be that of a “multimedia phone call to myself”. The user “dials” the “phone number” of the display and, when the display “answers”, a session of user-interaction can proceed. Of course, there need be no dials, numbers, or audio involved in the augmentation session. The dialing procedure (discovery and binding) can be fulfilled with, for example, RFID sensing as we describe in the section “Click-a-display”. The “answer” starts a session within which a data-oriented computing activity can proceed. Even without overt voice call features, the SIP/OA user-experience is modeled on a phone-call in that there is a caller and callee, a call or session, an exchange, and finally disconnect.

SIP/OA builds on the SIP architecture and protocol by 1) adding discovery mechanisms suitable to device augmentation that transfer SIP URLs and by 2) adding session description payloads to the INVITE messages suitable for device-device interaction negotiation. With these relatively small additions we create an architecture that supports multiple, widely varied discovery solutions and supports any IP host-to-host media connection. Most important, we can do this without coupling the discovery mechanism to the particular media rendering programs. The architecture is illustrated in Fig. 1. The next two sections describe these two additions.

3. Phone-centric Discovery Mechanisms

The first addition to SIP is augmentable device address discovery for phones. We built prototypes for two quite different approaches. Dial-a-display uses the SIP/OA metaphor literally: the user keys a phone number into their handheld device to provide the URI for SIP INVITE. (See Fig. 2.) The use model is unusual compared to a phone call in two regards: first the call is to a nearby device, not one far away; second, the call is answered by the same human that placed the call,

just on a different device. Despite these oddities, the user interface could be straight forward as it is driven by user actions. As with voice calls, SIP/OA calls can be added to user’s “speed dial” settings.

Click-a-display simplifies the user experience by storing the equivalent of the phone number in an electronic tag. (See Fig. 3). The tag can be on the handheld or on the display. For a tag reader on the display, the identification tag would be mounted in or on the handheld device. A user would apply the handheld to the reader, the display computer would resolve the tag-id to a URL [6] which would then be used as a URI in SIP INVITE. We used RFID tags on our SIP phone. For optical tags, existing cameras on phones can be harnessed for this mode of discovery [7], with the tag on the display and the caller being the phone.

4. “Application” Sessions

The second ingredient in SIP/OA is media description suitable for opportunistic augmentation. The SIP standard [3] relies on the Session Description Protocol, or SDP [11] for specification of potential media sessions. Typically the SIP INVITE message from a caller to a callee will contain in its message body an SDP-formatted message. SDP in turn relies on IANA MIME to specify media types. In Voice over IP (VoIP) uses, the media types in the SDP body are audio and other information in the body details the acceptable attributes for audio communications. RFC 3264 details the communications pattern of SIP messages that allows selection of compatible media and attributes between caller and callee [11].

For opportunistic augmentation we need user agents that support media type “application”. Some existing SIP user agents support video media types, but we did not find one supporting media type `application`. Therefore we extended a SIP softphone to add support for media types “application/http” to support web browser/server interaction and “application/x-vnc” to support the VNC remote desktop protocol. The extension required adding new media descriptions to the SDP body sent by the SIP UAC, new “offer/answer” code [12] to compare the session request to the capabilities of the recipient, and new media-handler code for the new media types.

Since SDP evolved for use in Internet telephony, the

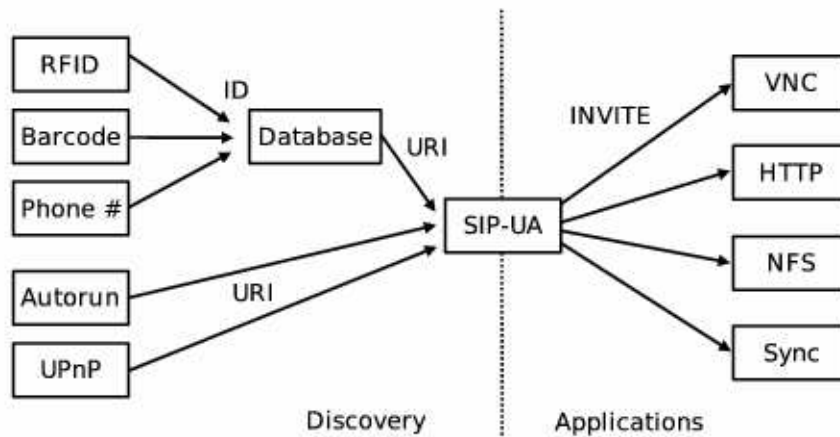


Figure 1. SIP as an architectural separation between discovery and media renderers in opportunistic augmentation. On the left side each box represents a discovery approach: RFID and Barcode mean identification technologies as describe for “click-a-display” “Phone #” refers to “dial a display” in the text. These discovery mechanisms can use a database to convert numbers into SIP URIs. Autorun means using stored files in the mobile device with special names recognized by the augmenting device; UPnP stands in for various network protocol based device and service discovery solutions. In the middle is the SIP User Agent (UA), addressed by discovery, and configured with information on the available media rendering applications. On the right side are media-renderers including VNC, web browsing where the server could be in the mobile device (HTTP), file editing where the files are served from the mobile device (NFS), and file content synchronization (sync). Discovery leads to a SIP interaction that then negotiates a matching application protocol. The SIP/OA architecture allows flexibility on either side mediated by an Internet standard communications protocol.

media description for applications is not worked out in detail. However its basis in IANA MIME types makes extensions orthogonal to the audio and video descriptions. Fig. 4 shows a example SDP document. Our additions in the last two lines offer a web server at port 8080.

To answer this offer, we apply RFC 2534, iterating over each of the media descriptions (lines starting with `m=`) stanzas and comparing the media types and parameters to capabilities of the answering user agent. Any failures to match are marked with port 0 in the answer SDP body. We invented the attributes “server” and “client” and interpret them using the rules for “sendonly” and “recvonly” respectively. The attributes designed for audio and video are not clear when applied to application protocols. Applications tend to be bi-directional so the attribute needs only specify which side will listen and which will talk to start the protocol.

Thus “server” and “client” would be clearer.

After the offer/answer phase, we launch a media stream. If this is an application media type, we need to launch or configure a corresponding application. This action will be specific to the user agent operating system and application preferences. Just like web browsers launch external applications to handle media types they do not support directly, our extended SIP softphones launch external applications. We use the same strategy of a locally configured table that maps MIME types to applications.

5. Implementation

To demonstrate separation of discovery and application selection via SIP we implemented a SIP “softphone” to support “application” MIME media types[10], configured these phones to launch one of

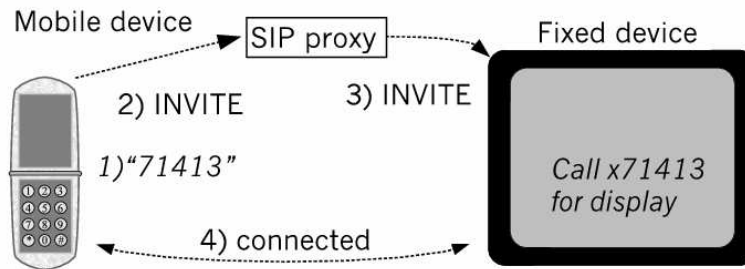


Figure 2. Schematic for Dial-a-display Opportunistic Augmentation with SIP. 1) User reads the phone number of the display and keys it into the mobile device. 2) The mobile device issues INVITE message to the SIP proxy. 3) The INVITE message is forward to the display. 4) The remaining SIP protocol messages establish a session.

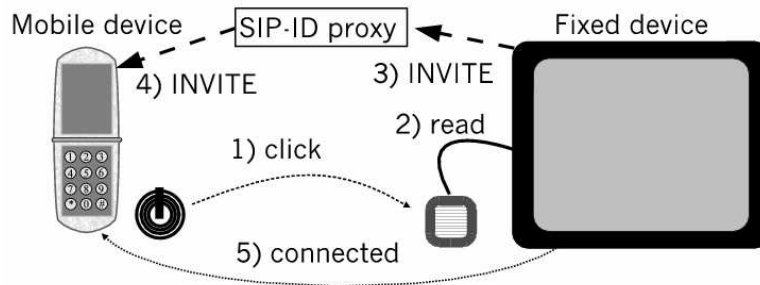


Figure 3. Schematic for Click-a-display Opportunistic Augmentation with SIP. 1) An RFID tag is placed within range of the tag reader for the display. 2) The reader sends the ID value to the SIP UA representing the display. 3) The display's SIP UA issues an INVITE message to a URI created from the ID value and sends it to the SIP proxy. 4) The INVITE message arrives at the mobile device. 5) After the human answers, the remaining SIP protocol messages establish a session.

```
v=0
o=ThinkLink 0 0 IN IP4 9.1.115.99
s=-
c=IN IP4 9.1.115.99
t=0 0
m=audio 22224 RTP/AVP 0 3 4 5 6 8 15 18
a=recvonly
m=video 22222 RTP/AVP 26 34 31
a=recvonly
m=application 8080 TCP http
a=server
```

Figure 4. An example SDP body with application media formats.

two different applications, and connected two different discovery mechanisms to the phones. Running this softphone on a mobile laptop models a futuristic mobile phone; running this softphone on a desktop PC models a display that can augment such a phone. We also worked with a commercially available VoIP phone[14] taking the role of the mobile device, to verify the basic communications elements of our softphone, to understand issues with commercial implementations, and for more realistic user trials. These phones do not support the mime-type “application” so we placed code in our softphone to simulate the reception or transmission of the extended data for the user trials.

For SIP/OA to be minimally practical, users must

1) be able to connect phones to displays and 2) obtain significant benefit from the augmentation. Using a commercial SIP phone [14], we conducted a 12-person user study to verify that SIP/OA met these criteria. The details of this study are reported separately [15]. The three tests relevant here were for click-a-display, dial-a-display, and—for comparison—manual device connection by looking up the phone’s IP address and typing it into a web-browser address bar.

We found the click-a-display (mean 24 ± 1 sec std error) and dial-a-display (28 ± 2 sec) technology allows people to connect the phone to the display with very little training. These connect times were faster than manual device connection (54 ± 7 sec). Also the variation in connection time as measured by the standard deviation of the times was highest when users typed the phone’s IP address in to the web browser (24 sec), lowest for click-a-display (4 sec) and dial-a-display (8 sec). Therefore we claim to meet the first minimum criteria above.

The second criterion, benefit, depends upon the user’s needs and skills. For our study we had users enter values in the phone’s address book. This task can be done with the phone’s keypad through multitap text entry. After augmentation they can use the larger, two-handed keyboard and large screen. We timed users finding the phone-book using the keypad as a fourth “connection method” for comparison. Access to the phone-book was fast on the keypad, 13 ± 8 sec, but text entry was three times slower than computer keyboard. (These results are consistent with other studies in the field [16, 1] on text-entry speed). Our study showed that users gain from the use of a computer keyboard and screen over the phone’s keypad and screen after as little as 15 characters.

6. Related Work

Our work provides a different concrete architecture for the concepts of opportunistic annexing explored by Pierce and Mahaney [2]. Rather than imagine that the mobile device and the augmenting device as part of one overall application, we treat these devices as part of multi-device, multimedia session modeled after multi-party, multimedia work designed for the Internet. This divides the sets of problems into discovery, connection, and media rendering. Complex applica-

tions can be built with Internet multimedia protocols, but simple ones like browsing files in a mobile phone can be used without additional research. Our architecture addresses only the connection and leaves all questions of interface and user actions after the connection up to the media rendering programs.

Want et al. use a web-browsing metaphor in their personal server work [4]. They demonstrate a handheld device with little or no user interface, and thus no ability to “dial”. The discovery in their personal server relies on proximity wireless strength with the user interaction driven completely from the augmenting device. Presumably users will imagine their device emitting an invisible aura that is picked up by all nearby displays. These displays may either actively notice every user as they come near – creating a privacy issue – or they may await a user signal to search out nearby handhelds – creating a usability challenge since the user must wait for discovery then correctly select their device from all those within radio range. In addition, disconnection depends upon the highly variable decay of radio strength.

Two well-known ways to connect small devices to computers could be used for opportunistic augmentation: docking and network broadcast discovery. Docking could be used for phone augmentation connections. We expect this connection approach be useful in user-controlled environments where the handheld is stored for recharging, for example in a home application.

Network-based discovery is widely used for small groups of fixed devices and it has been adapted for wireless devices [5]. As the number of devices and networks increase, we expect the user experience to decline because selection from a long list of potential matches becomes onerous. Rekimoto [18] showed how to use human button-press timings for discovery to avoid this problem. All of these solutions fail if the two devices are on different subnets.

The dial-a-display discovery method resembles the most common early method for connecting computers: using IP addresses directly in media rendering applications. For example, URLs containing IP addresses can be typed directly into web browsers. The multimedia handling by web browsers is similar to the SIP approach we use. However, the SIP approach fits with the predominant handheld device, the phone; phone

numbers naturally work well with phone keypads; SIP already works with NAT and similar multi-network issues; SIP allows either device to initiate the connection.

7. Conclusion

Our contribution includes 1) an architectural element for ubiquitous computing system that decouples opportunistic augmentation into discovery and session phases, 2) application of an emerging Internet Protocol – SIP – to an important ubiquitous computing paradigm, opportunistic augmentation, 3) demonstrating how existing or proposed discovery techniques would work with SIP in our approach, 4) elaboration and implementation of two light-weight discovery techniques “dial-a-display” [2] and “click-a-display” [9].

References

- [1] P.-O. Kristensson and S. Zhai. SHARK2: A large vocabulary shorthand writing system for pen-based computers. *CHI Letters* 6(2), 2004.
- [2] J. S. Pierce and H. E. Mahaney. Opportunistic Annexing for Handheld Devices: Opportunities and Challenges. *Human-Computer Interaction Consortium (HCIC)* 2004, Fraser, CO
- [3] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol, *IETF Request for Comments: 3261*. June 2002
- [4] R. Want, T. Pering, G. Danneels, M. Kumar, M. Sundar, and J. Light. The Personal Server: Changing the Way We Think about Ubiquitous Computing. *UbiComp 2002* Göteborg, Sweden, September 29 - October 1, 2002. G. Borriello, L.E. Holmquist (Eds.).
- [5] E. Guttman. Autoconfiguration for IP Networking: Enabling Local Communication. *IEEE Internet Computing* 5(3):81-86, May 2001.
- [6] T. Kindberg Implementing physical hyperlinks using ubiquitous identifier resolution. *Proc. of the 11th Intl Conf. World Wide Web*, 2002
- [7] I. Haritaoglu. InfoScope : Link from Real World to Digital Information Space. *UbiComp 2001*: 247-255
- [8] D. Scott, R. Sharp, A. Madhavapeddy, and E. Upton. Using Visual Tags to Bypass Bluetooth Device Discovery. *Mobile Comput. and Comm. Rev.*, 9(1):41, Jan. 2005
- [9] R. Want, K. P. Fishkin, A. Gujar, B. L. Harrison. Bridging physical and virtual worlds with electronic tags *Proc. SIGCHI conference on Human factors in computing systems* p 370 - 377 1999.
- [10] Our softphone implementation used the NIST-JAIN SIP stack available from <http://www-x.antd.nist.gov/proj/iptel/>
- [11] M. Handley and V. Jacobson. SDP: Session Description Protocol *IETF RFC 2327*.
- [12] J. Rosenberg H. Schulzrinne. An Offer/Answer Model with Session Description Protocol (SDP). *IETF RFC 3264*
- [13] SIP Express Router (SER) <http://iptel.org>
- [14] ZyXEL 2000 Prestige W
- [15] J. J. Barton, F. Folowosele, S. Nylander, M. Houston and S. Zhai, “Connection Time for Strange Devices” submitted to WMCSA 2006.
- [16] C. L. James and K. M. Reischel. Text Input for mobile devices: comparing model prediction to actual performance. *Proc. of the SIGCHI conference on Human factors in computing systems*, 2001
- [17] A. Acharya, S. Berger and C. Narayanaswami. Unleashing the power of Wearable Devices in a SIP infrastructure. *Third IEEE Int’l Con. on Pervasive Computing and Communications (Per-Com)* 2005
- [18] J. Rekimoto, Y. Ayatsuka and M. Kohno. Sync-Tap: An Interaction Technique for Mobile Networking. *MOBILE HCI* 2003.