

Unleashing the Power of Wearable Devices in a SIP Infrastructure

Arup Acharya

IBM T.J. Watson Research Center
19 Skyline Drive, Hawthorne
New York, NY 10532

arup@us.ibm.com

Stefan Berger

IBM T.J. Watson Research Center
19 Skyline Drive, Hawthorne
New York, NY 10532

stefanb@us.ibm.com

Chandra Narayanaswami

IBM T.J. Watson Research Center
19 Skyline Drive, Hawthorne
New York, NY 10532

chandras@us.ibm.com

Abstract

The Session Initiation Protocol (SIP) has been widely adopted for Instant Messaging (IM) and VoIP telephony both by the enterprise and in service provider systems. Till now, SIP functionality has been accessible to end-users primarily via computing platforms such as an IM client on a laptop or on communication platforms such as an IP Phone or a cell phone. We show that wearable devices such as the IBM Linux Watch (WatchPad™) can play a powerful role as a control device in a SIP infrastructure, given their easily accessible, always available, and visible user-interfaces. We have designed and prototyped several key control applications, using a SIP User Agent on the WatchPad, such as the ability to initiate and route incoming and outgoing phone calls, receive simple queries as Instant Messages, and call initiation based on address books residing on wearable devices. We discuss other examples such as video conferencing and the authentication layer for work flow. Our investigation and implementation shows that wearable devices can plug an important gap and significantly enhance user experience.

1. Introduction

The deployment of SIP (Session Initiation Protocol) [11] in enterprise networks gives users significantly greater flexibility to use and control commonly used communication technologies. SIP is today primarily used for VoIP, Instant Messaging and presence detection. Several other uses are also being explored. SIP has been adopted by popular products such as Microsoft's Windows Messenger and IBM's Lotus Sametime. A significant reason for the accelerated acceptance of SIP is that the infrastructure allows regular phones to interoperate with VoIP phones and thus does not create *islands of communication*.

An interesting feature of SIP is the ability to separate the control and media portions of a connection. In fact, they may take place at different devices. Thus SIP allows a level of indirection and *late binding* of the media device. This is a powerful concept and has the potential to do for communications what pointers did for programming by separating data from addresses and virtual memory did for large programs by creating virtual addresses to overcome limits of real addresses.

Since the ability to exercise control places very few requirements on end devices, it is possible to make wearable devices with a TCP/IP stack part of the SIP infrastructure. In this paper we argue that wearable computers are ideally suited for setting up sessions because of their availability and ability to get the user's attention in varied situations. Wearable computers also contain information that can be used to personalize and improve the user experience. Similarly stationary devices are better suited to play the media because they have fewer constraints in terms of power, size, networking, etc. The above balance is unlikely to change because it is based on human behavior patterns and basic physics.

For our work, we have chosen IBM's Linux WatchPad [7] as a wearable device to experiment with controlling the SIP infrastructure. A cell phone that offers an open platform that can be modified is another possibility. We have implemented and tested several scenarios. The results of our experiments are very encouraging and show how network equipped wearable devices can be used in every-day situations.

The contributions of this paper are: (a) an architecture for use of a wearable device as a SIP-enabled control device, (b) use of SIP to tie the control and media components supported by different devices within the network, (c) novel application scenarios highlighting the separation of functionality between control and media and (d) working prototype of a SIP-enabled wearable device that implements the scenarios.

2. SIP background

SIP is a control protocol that allows creation, modification and termination of sessions with one or more participants. SIP is used for voice and video calls either for point-to-point or multiparty sessions. It is independent of the media transport which for example, typically uses RTP over UDP [10]. SIP is also used for Instant Messaging and presence detection [8]. SIP allows multiple end-points to establish media sessions with each other: it supports locating the end-points, establishing the session and then, after the media session has been completed, terminating the session. In recent times, SIP has gained widespread acceptance and deployment among wireline service providers for introducing new services such as VoIP, within the enterprise for Instant Messaging and collaboration and for push-to-talk service amongst mobile carriers. Industry acceptance of SIP as the protocol of choice for converged communications over IP networks is thus highly likely.

As shown in Figure 1, a SIP infrastructure consists of *user agents*, *registration servers*, *location servers* and *SIP proxies* deployed across a network. A *user agent* is a SIP endpoint that controls session setup and media transfer. User agents are identified by SIP URIs, which is a unique HTTP-like URI of the form *sip:user@domain*. All user agents REGISTER with a SIP *registrar server* (which can be co-located with a SIP proxy) with their IP address (see Figure 1). The mapping of a URI to the IP address of a device registered by the user is done using intermediate SIP proxies, *location* and *redirect servers* as part of the session setup process.

Details of the SIP protocol can be found in [11]. SIP defines a set of messages, such as INVITE, REFER etc., as shown in Figure 2, to setup sessions between user agents. These messages are routed through SIP proxies that are deployed in the network. DNS SRV records help in finding SIP proxies responsible for the destination domain.

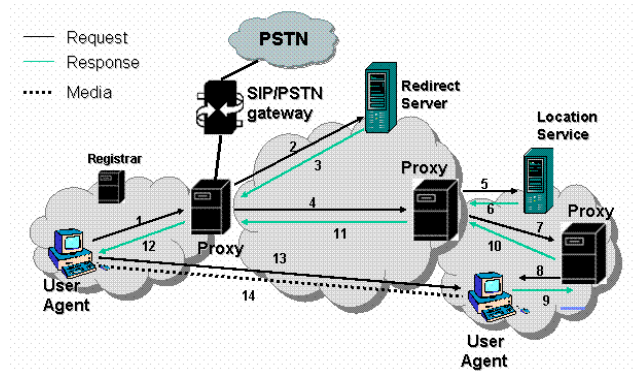


Figure 1: SIP architecture

All requests from an originating user agent such as an INVITE are routed by the proxy to an appropriate

destination user agent based on the destination SIP URI included in the INVITE message. Proxies may query location and redirect servers to determine the current bindings of the SIP URI. Signaling messages are exchanged between user agents, proxies and redirect/location servers to locate the appropriate endpoints for media exchange. For reasons of scalability, multiple proxies are used to distribute the signaling load [4]. A session is setup between two user agents through SIP signaling messages comprising of an INVITE, an OK response and an ACK to the response [11]. This is shown in Figure 2 where the call setup is followed by media exchange using RTP. The session is torn down through an exchange of BYE and OK messages.

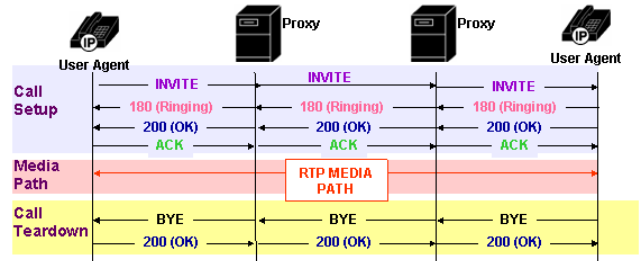


Figure 2: SIP Call setup and Media Path

SIP distinguishes between the process of session establishment and the actual session. A basic tenet of SIP is the *separation of signaling (control) from media*. Signaling messages are usually routed through the proxies while the media path is end-to-end. The session setup messages like INVITE contain user parameters using Session Description Protocol (SDP) [3] in the message body. SDP provides information about the session such as parameters for media type, transport protocol, IP addresses and port numbers of endpoints. The IP address and port numbers exchanged through SDP is used for the actual data transmission (*media path*) for the session. Any of these parameters can be changed during an ongoing session through a RE-INVITE message, which is identical to the INVITE message except that it can occur within an existing session. In addition, a user agent can *transfer* an existing session by using a REFER message. This message instructs the other endpoint of an existing session to initiate an INVITE/OK/ACK exchange with a third user agent and terminate the existing session (with the sender of the REFER message). Extensions to SIP for presence detection and Instant Messaging (SIMPLE) support SUBSCRIBE/NOTIFY mechanisms [8], in which user agents subscribe to certain events at another user agent and can be notified whenever that event occurs. SIMPLE provides support for Instant Messaging through the introduction of a new message, called MESSAGE. More recently, the Message Session Relay Protocol (MSRP) [1] has been proposed as an alternate method for Instant Messaging that, unlike SIMPLE, is session-based.

An interoperability function with the public-switched telephony network (PSTN), defined [14], allows calls to be made between a PSTN phone and a SIP phone. A SIP/PSTN gateway is responsible for converting between SIP and PSTN signaling on the control path as well as media conversion between circuit-switched voice in the PSTN network and packet voice in the SIP/ IP network. A SIP device can be addressed from a PSTN phone by assigning a PSTN number to the SIP device and setting up appropriate call routing entries at the SIP/PSTN gateway. Similarly, a PSTN number can be easily mapped to a corresponding SIP URI and thus a PSTN device can be addressed by a SIP device as well. Besides interoperability, there are some known issues with respect to firewall traversal when SIP signaling and associated media crosses domain boundaries. However, multiple solutions [12] have been put forward to address the issue.

The separation of SIP signaling or control from media exchange makes it an attractive choice for use with wearable devices. It allows a wearable device to initiate a session but not require it to handle the media flow. Once the session has been established, it can move the session to another device which is better equipped to handle the media. The ability to interoperate between SIP and PSTN networks implies that a SIP-enabled wearable control device can control not only SIP devices but also PSTN and cellular phones, which are ubiquitous in home and offices.

3. Roles for Mobile and Stationary Devices

In this section we show why the symbiosis between mobile/wearable devices and stationary devices is compelling.

3.1. Why use wearable devices for control?

Wearable devices have two significant advantages over traditional devices. First, a wearable device is always available and offers a familiar, known interface to the user and also contains the user's personal information (e.g., calendar, address book). In contrast a stationary phone, say in a conference room, may only offer a keypad to dial out a number and contain no personalized information. Second, wearable devices are also very good at getting the user's attention to an incoming request by using audio or tactile means to alert the user subtly.

On the other hand, though wearable devices may support media, they are typically limited in battery power. Wearable devices also have lower signal quality compared to wired counterparts. Thus, one of the goals of our work is to make full use of such devices for control purposes only, whenever practical.

In our architecture, the media input/output is accomplished through one of the devices in the user's

vicinity rather than through the wearable device. The primary role of the wearable device is to setup a session with called parties and then transfer the media stream(s) to one or more adjacent devices. In a corporate setting, it is getting easier to find devices in the neighborhood that are capable of handling media to different degrees (e.g., a laptop may be able to display video and play audio while a phone may handle audio streams alone). The just in time *binding* of the media device (that is done when the session is established) makes it possible to take into account the location and situational context of the user. The user does not have to fill in preferences or policies ahead of time that may be invalid at the time the session is initiated.

The wearable device can supply any authentication information necessary to set up the session. In addition it can also negotiate parameters of a session: for a media device with a limited screen size, this feature could be utilized to restrict the type of content that the device is willing to receive, e.g., an instant message not exceeding 100 characters. The wearable device can also log the time, duration, destination party, and other details for the session. The log can later be used to retrieve the media from servers storing the media.

A wearable device can be useful as a companion device to laptop computers, especially when the laptop's display is busy (e.g., a presentation in progress) or out of the user's visual field. When the laptop's display is not available, notification of incoming calls or IMs, including the caller's identification, can be forwarded to the wearable device's display and allow the user to respond with a limited set of options via the device's input means. These options vary with the context and can be tailored to specific applications.

For the above reasons, the control part, which is less power/memory/compute intensive, is best left to a wearable device (and yet leverages user-specific information stored on the device such as an address book) while the media components are best handled by tethered devices commonly found within an office environment or in some upscale homes. The control glue to tie in all the pieces is provided by the wearable device.

A wearable personal device such as the WatchPad or a pendant also has a few advantages even over popular devices like cell phones or personal digital assistants. A watch presents a *glanceable* and *easily accessible* user interface. In contrast, a cell phone must be pulled out of the pocket to identify the caller. A watch or pendant is thus less obtrusive than a cell phone and has a lower user response time to react to incoming calls or messages. Therefore, the WatchPad or pendant can be used in two modes: either (a) as a first-class control device in the SIP network or (b) as a front-end to a cell-phone or a PDA, where *the watch's own screen acts as a proxy for the cell-phone or PDA's user interface*.

The key challenge in integrating a SIP-enabled wearable device is to determine how to partition functionality of existing and new applications between the watch and existing SIP devices, map control functionality onto the wearable device without exceeding the memory or processing budget of the watch and yet enhance the overall user experience.

3.2. Stationary devices for media exchange

Stationary devices in the infrastructure can be superior in many ways for actual I/O for the media. Stationary devices have access to electrical power, and can easily have more computing capacity and have larger speakers, larger and brighter displays, and larger and more secure storage systems. These properties also allow the users to record the media portions of the session for later retrieval and search. The data gathered can be linked to other applications such as email and voicemail systems. Higher quality audio output such as stereo, Dolby Surround Sound and Dolby Pro Logic IIX, etc., are possible with these systems and are more difficult to provide on mobile platforms. Some audio systems are also able to provide directed sound. Systems such as the IBM EveryWhere Display [6] can provide large steerable projectors to show the graphics or video. This is not possible with small displays on mobile systems. Moreover, stationary devices typically can have faster and cheaper network connections, allowing higher frame rates in graphics and video applications, and can play compressed media streams in real time. Stationary devices are often found in conference rooms and other public places where several people can gather to listen to or see the media. Finally these systems are able to encrypt and decrypt data strongly and offer a higher level of security.

Another advantage of stationary devices is the integration with corporate infrastructure and with other applications. For example, a recorded conversation can be converted to text and distributed via an email application. This would be much harder to do if the conversation was recorded on a mobile device. Similarly a recorded video can be attached to corporate databases for later retrieval and review. In other cases, using translation technologies on the stationary device, one party could use voice while the other uses text.

However, stationary devices are not as good as wearable devices for getting the user's attention since they may be away from the user, may not offer privacy in announcing the caller, have their speakers turned off, etc. They also usually do not have tactile mechanisms to subtly alert the user of an incoming call.

4. Testbed architecture and devices

4.1. WatchPad Background

A wrist watch is in many ways the ultimate wearable device. The IBM WatchPad [7] is a wrist watch that runs Linux. The most recent version (the one we use) has 8MB of dynamic RAM for execution of programs and 16MB of flash memory. It uses an ARM7 processor that can run at speeds between 18 and 74 MHz. The watch has 320x240 monochrome LCD display and a Bluetooth interface that gives it access to the Internet using TCP/IP. Further, it has a touch-screen, three buttons on the front, a stem switch, and a built-in beeper and vibrator to attract a user's attention, and a fingerprint sensor for user authentication.

One significant challenge when building applications for wearable devices is the design of simple and intuitive interaction mechanisms through self-explanatory graphical, auditory and tactile means. The vibration device can be programmed to generate different tactile patterns (waveforms) for different types of communication and notify the recipient subtly. For example a 4 second *on* and a 4 second *off* pattern could indicate a voice call and a 4 second *on* and a 2 second *off* pattern an instant message.

The watch does not offer a big screen to display a lot of information, which led us to implement a rather simple GUI. The limited input capabilities of the device through a stem switch and three buttons confines the amount of information that a user can enter. However, for our problem, these limitations on wearable devices are offset by the advantage of having them always available.

When the watch is powered-off, an alternate device such as a user agent on a laptop may register the same URI as the watch so that incoming session setup requests or IMs are not missed when the watch is powered off. To conserve power on the WatchPad the external Bluetooth communications chip is turned off as often and for as long as possible. This helps us to preserve more power than what would be possible with Bluetooth power saving modes. As a consequence, our SIP-enabled applications need to check for network connectivity on a regular basis. Once IP connectivity has been made available, SIP registration is performed and subscription requests are sent. To prevent too frequent registration updates on the one side and stale information on the server side on the other side, we chose registration times of 10 minutes. This keeps communication requirements at a moderate level. Other power optimizations discussed in [5] can be applied as well.

4.2. System architecture

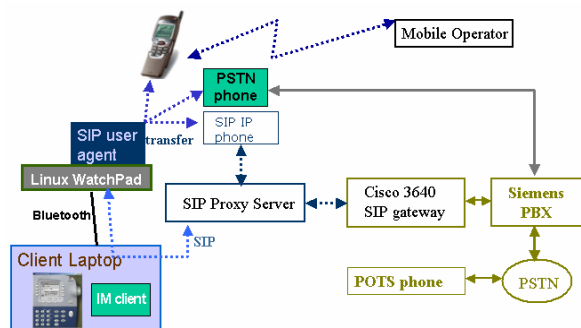


Figure 3: SIP Environment

The WatchPad is augmented with a SIP stack enabling it to operate as a User Agent. It registers with the SIP proxy with its own SIP URI. A typical SIP network environment is shown in Figure 3. A laptop computer serves as Bluetooth base-station and router into the wireline network in our system. In addition to SIP enabled devices, there are also PSTN devices such as phones and cell phones in the vicinity of the WatchPad. The PSTN landline phones interconnect with the SIP network via a SIP/PBX gateway which translates signaling and media between PSTN and SIP/VoIP. In other words, the PSTN devices can be addressed from the SIP infrastructure, e.g., a PSTN phone with a number 123-456-1111 inside a company say, *talk.domain*, can be addressed as `sip:1234561111@talk.domain.com`. This is equally true for a cellular phone which unlike the PSTN phone is maintained not by the corporate IT network but by an external mobile operator. However, the connectivity to the cell phone is still provided by the PSTN gateway¹. In other words, the WatchPad has access and control over three classes of devices, computing platforms such as laptops and PDAs which are also SIP endpoints, SIP IP phones and PSTN phones both landline and cellular.

To enable SIP on the watch we have used the GNU OpenSIP library (oSIP) [16] as well as the user agent library from the LinPhone project [17]. The user agent library offers methods for registration of end-devices with the SIP server in our network and itself uses the oSIP library. It provides the basic functionality for call establishment and tear down including SDP parameter negotiation for RTP audio and video exchange. Besides that, it takes care of the dialog maintenance that is required by the SIP protocol. All functionality that was not provided by the library, such as support for instant messaging, pub/sub and presence detection, has been added during our experiments. We enhanced the application shell on the watch [9] through a module that

offers an interface for the user to establish voice calls or receive an IM. Its task is to handle the interaction with the underlying SIP user agent and coordinate user inputs.

With the above SIP functionality supported on the WatchPad, the watch behaves just like any other SIP device in the infrastructure, except that we explicitly do not support any media input/output. After the WatchPad registers itself with a SIP proxy using a SIP URI containing an embedded PSTN number, it becomes addressable by any device, SIP or PSTN (due to the interoperability function described in section 2) and is itself able to control such devices including redirecting VoIP media sessions to user-selected devices. In the next few subsections we describe various application scenarios where the WatchPad controls how sessions are initiated and different devices are brought into the session to support various media types.

5. Scenarios and implementation

5.1. Control for voice based applications

For our first set of scenarios we use the WatchPad to control voice communications. Note that the WatchPad never acts as the source or sink of VoIP media (RTP packets) and thus its role is to setup the session and then orchestrate the flow of RTP media packets to the right device.

5.1.1. Call Initiation

A wearable device like the WatchPad can store a user's address book and is thus a natural device to initiate calls. In this scenario, the WatchPad is used to set up a call between two phones. One phone is in the user's vicinity and the other phone belongs to the party the user wishes to call.

Our application allows the selection of the destination party from the address book. It also requires the user to select a device within the user's vicinity from a list. Although the list of nearby phone is currently static, these devices could be determined dynamically using known service discovery mechanisms. Figure 4 shows a snapshot of the user interface on the WatchPad while call-initiation is in progress.

Figure 5 shows that the call establishment is completely orchestrated by the watch. The watch first sets up a connection with the device selected by the user (say for example, the user's office phone). This is achieved through the INVITE/OK/ACK exchange of SIP messages. At this point, instead of sending RTP packets, the watch

¹ Charges incurred in the two situations could be different, e.g. the call to a cell-phone could incur a PSTN toll charge.



Figure 4: Placing a call using the watch

initiates a call transfer by *logically* sending a REFER message to the office phone² containing the SIP URI of the called party. This causes the office phone to setup a session with the called party through an INVITE/OK/ACK message exchange and then start exchanging media. The net effect of this control exchange is that the call is initiated from the watch, the office phone rings and when the user picks up this phone, the destination party is called and media flows between the office phone and the called party. The watch is then completely taken out of the control loop.

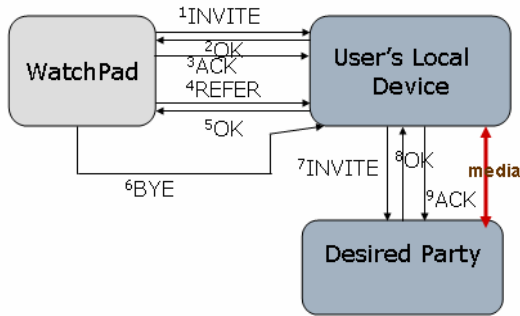


Figure 5: Watch places call, redirection to outside telephone

A practical enhancement of the calendar application on the watch could make SIP URIs with information regarding multi-party conference calls available. The watch can then be used to dial into a teleconference and provide the right password. In this scenario the watch would also use a nearby phone to initiate the call. In practice we find that this simple application is very useful because it avoids a trip to the calendar application on the

² Note that the office phone can be a standard POTS phone as long as it is connected to the SIP network through a SIP/PSTN gateway, as mentioned in Section 4.2. It is the gateway that handles the SIP messages and translates them to appropriate PSTN signaling.

laptop to determine the right information to dial, keeps calls on time, and also reduces frequency of misdialing.

Measurement	183 Session Progress
Watch	2.06s
Watch simulator on PC	1.68s
Softphone	1.39s

Table 1: Delay Measurements

We compared the initial call setup times of our application when run on the watch against times when run in the watch simulator on a PC or when using a Java Softphone implementation (both Ethernet-connected). We measured the times it took to receive a response from the SIP Proxy in our network following the initial INVITE. In this case we received a 183 Session in Progress, which is similar to the 180 response in Figure 2. Our results are shown in Table 1. The measurements were conducted in slightly different ways, though. On the Softphone we used a packet sniffer to measure the times whereas on the watch we measured times at the application level.

Our measurements indicate that delays are within acceptable level and from our experience we can say that delays in call establishment from the watch are hardly noticeable. The delay that we see on the watch is mostly related to the delay introduced by the Bluetooth interface and slower processing of the message.

5.1.2. Incoming Call Redirection

A second useful scenario for the WatchPad is to allow the user to control the responses to incoming calls. The WatchPad's audio beeper and vibration device can quickly get the user's attention. Once the user's attention is obtained, he can be presented with the call details and also a list of options for responding to the call.

The user-agent on the watch registers with the SIP proxy and is associated with a SIP URI (with an associated PSTN number for non-SIP users to call). There are two modes of possible operation: (a) the WatchPad is the contact number/address, i.e., this is the "phone number" advertised to all parties, or (b) the user may have a cell phone that is used as the contact point, but the cell-phone is augmented with a SIP user agent that automatically transfers all incoming calls to the WatchPad, either directly because there is a direct Bluetooth link between the two or through an intervening network of the mobile operator and a SIP/PBX gateway as depicted in the architecture diagram (Figure 3). For simplicity of description, we will assume that the WatchPad's address is used as the contact address.

When a notification about an incoming call is received, our application audibly notifies the user and takes over

the display. As shown in Figure 6, the name of the calling party is displayed on the screen along with a menu of pre-configured options that allow the user to respond to the call in several different ways:

- The user can redirect the call to a pre-defined set of phone numbers (cell phone, administrative assistant, backup, etc.),
- reject the call with a busy response, or
- redirect the call to a voicemail server with a message indicating that the user will be back later.



Figure 6: WatchPad receiving phone call

We believe the ability to respond to calls with even canned messages (such as call me later or call me at a certain time) or to be able to redirect the call in a non-obtrusive manner is much better than the approach today with cell phone where there are only two choices – ignore the call or rush to take it. The pre-stored options could be customized based on the caller.

The user has to roll the stem switch to select one of the

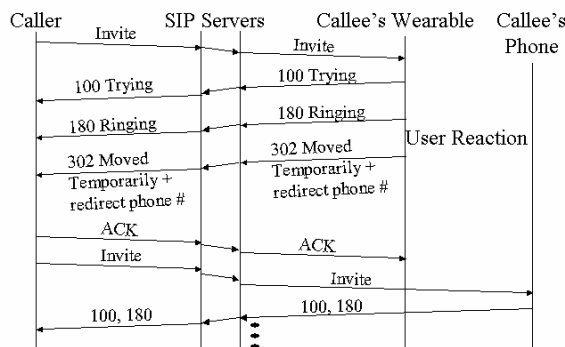


Figure 7: Call flow for call redirection on incoming call

options. Depending on the selection, the underlying SIP user agent generates a response message with the appropriate status code. If for example the user selects the 'reject' choice, the response message will be a '603 Decline'.

Figure 7 shows the call flow diagram for this scenario. Since a user needs some time to decide on how to deal with the call, the "Moved Temporarily" response is

generated after the user makes a choice (which in this case, is to take the call from a nearby phone). Thus, the incoming call received at the WatchPad is redirected to a phone by use of the "Moved Temporarily" SIP response; this response specifies a second SIP URI (phone number) that the caller can use to retry the call setup.

5.1.3. Moving a call from a landline to a cell phone

In the previous two scenarios, the WatchPad relinquished control once the session was established and the media flow was setup with an external device. However, there are situations where it is advantageous for the WatchPad to remain in control during the lifetime of the session. As an example, consider the case where the WatchPad initially sets up a voice call with an office phone but then the user may need to go on the road prior to termination of the call. In such situations, it would be useful if the watch could orchestrate transfer of the call to another device, perhaps a cell phone, without interrupting the ongoing call.

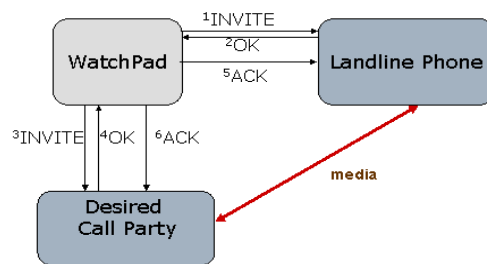


Figure 8: Call setup with WatchPad remaining in control loop

In order to facilitate the above scenario, an alternate SIP mechanism called a Back2back User Agent (B2BUA) [11] must be used on the device. In this case, the watch initiates an INVITE/OK exchange with a PSTN phone (but not sent an ACK) and receives an IP address/port number from the phone for sending the media (Figure 8). It then sends this information in the INVITE/OK/ACK exchange with the called party and in return receives the IP address/port number of the called party for media transfer. At this point, the watch sends the IP address/port number of the called party via the ACK message to the PSTN phone. Consequently, the media is still exchanged directly between the landline phone and the called party but all signaling goes through the WatchPad.

When the call needs to be moved to the user's cell phone, as shown in Figure 9, the watch breaks the current call leg to the PSTN phone (through a BYE/OK exchange, initiates a INVITE/OK/ACK exchange with a new device such as cell-phone and then informs the called party of a change in the IP address/port number of the other endpoint (i.e., the cell-phone). The media then flows between the new device and the called party.

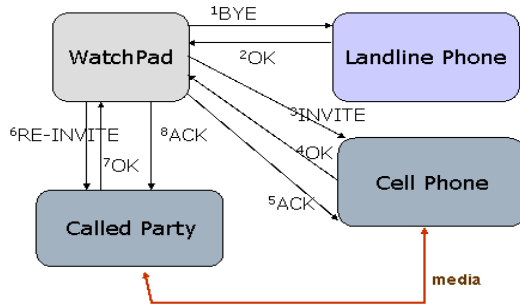


Figure 9: WatchPad as control device to initiate call transfer

5.2. Controlling Video Sessions

SIP is independent of the type of media exchanged once the session has been set up. SDP is used to inform the session endpoints of the type of media, which is opaque to the SIP signaling per se. Thus, the mechanisms described in section 5.1 for orchestrating voice sessions can also be used for video sessions. For example, the WatchPad may initiate a video session between two laptop computers. In another example, the WatchPad may initiate a session between two laptop computers and then switch the session to the user's wearable device with video capabilities when the user has to move. Invitations to join video conferences can also be sent to the WatchPad. The information necessary for joining the conference, i.e., IP addresses, etc., can be sent to the watch directly. The user can then connect to the video session with a few operations on the watch.

5.3. Controlling text exchanges - IM Client

Besides e-mail, instant messaging has become one of the most popular ways to communicate. In fact, in corporate settings IM has sped up the communication process significantly and is preferred over email for quick message exchanges. Since SIP has gained IM functionality through recent extensions to the protocol and the fact that the SIP stack is already on the device, support of instant messaging is a logical addition for wearable devices.

To support instant messaging, we adapted the OpenSIP stack for being able to exchange messages with the NIST SIP Instant Messaging client [18] and presence server. This meant extending the stack to handle incoming SUBSCRIBE messages and being able to send NOTIFY message with presence information (cpim-pidf[13]) content in the body (Figure 10). This allows the watch to register its availability as an Instant Messaging device with the presence server and peer client. Further, we extended our SIP application with a messaging client and added missing functionality to the user agent library such as support for the MESSAGE method. MESSAGE is used

for transporting Instant Messages in the SIP Instant Messaging and Presence Leveraging Extensions (SIMPLE) [8] model. When an IM is received by the watch, the application switches into instant messaging mode and displays the sender's name, address and recent message on the display.

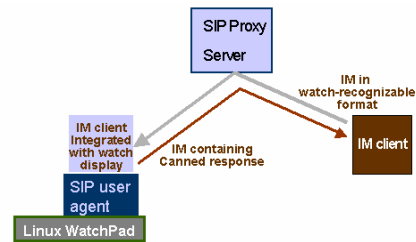


Figure 10: Instant Messaging client on the WatchPad

The limited input capabilities of the watch make it difficult to design a user interface for sending elaborate responses. We addressed this problem by allowing simple 'yes' and 'no' replies by pressing either the left or right button. The functionality of the middle button has been set to allow the selection of a reply from a list of predefined sentences. Further, the watch user's communication partner automatically receives notification of the limited input capabilities of the device as an initial response message. The recipient should honor this by formulating his own messages such that they can lead to simple responses. In the future the presence mechanism on IM systems can add another state indicating that the user is available but is using a wearable device with limited capabilities.

It is worthwhile to point out here that at first sight, supporting IMs on the WatchPad may appear to be a departure from our stated goal of exploiting the separation of control and media offered by SIP. In the SIMPLE model [8], which is the current model for supporting IMs in the SIP architecture, the payload of an IM is carried as a signaling message, i.e., the "control" message itself carries the "media" which is the IM payload in this case. A similar argument also applied to the scenario described in the next section where the WatchPad is used to receive subscription information via a SIP-based publish-subscribe system. There too, we are using the WatchPad as a means for receiving immediate notifications when certain events become true (in which the user has expressed an interest). Thus, in both these scenarios, while SIP may not be used strictly for control purposes (as interpreted through a strict definition of the protocol), we believe that these scenarios still represent 'control' applications in terms of the information provided to the user.

As with the voice example, ongoing IM sessions can also be transferred from one device to another, say from the WatchPad while in the corridor to the PC when back

in the office. Repair service providers could make use of this technology. For example, if a busy technician receives an instant message regarding a problem, he can redirect it to someone who is less busy.

5.4. Wearables and Pub/Sub Functionality of SIP

Another area of SIP-based functionality that the WatchPad can use is a subscription service to a publish-subscribe system. SIP provides built-in pub/sub functionality through its methods SUBSCRIBE and NOTIFY [2,8]. Currently this functionality is being used in SIMPLE for a client to subscribe to another user's current presence status and willingness to exchange messages. The functionality of pub/sub can be applied to other areas as well, such as for example periodic retrieval of personalized information such as weather conditions or forecasts or updates on stock prices.

The WatchPad is uniquely suited for user-display of pub/sub information. Each subscription to a service can be represented on the watch's screen as an icon, and a change in the subscribed parameter's value can be visually indicated by a change in the icon, e.g., a weather service can be represented by icons for sunny, rainy etc., and the current icon on display immediately indicates the current weather to the user. Just as cell phones show the antenna signal strength and the battery strength on the screen, information like weather or traffic can be shown on the watch face to make the information available at a glance.

We have prototyped a weather information

```
<?xml version="1.0" encoding="UTF-8"?>
<weather xmlns="xyz:ns:weather">
  <subscription>
    <location>
      <country>US</country>
      <zip>98765</zip>
    </location>
    <temperature>1</temperature>
    <winddir>30</winddir>
    <condition>*</condition>
  </subscription>
</weather>
```

Figure 11: Weather information subscription package

subscription package to request updates from our information server. Whenever the current weather condition deviates from the last update by a specified quantity, the server sends an update notification to the client. The pub/sub mechanism of SIP provides the advantage of not having to poll the server for continuous updates. The number of updates the watch receives can be adjusted through choosing the granularity in weather changes that will trigger notification. The subscription package for weather information can be seen in Figure 11.

It is sent in the body of a SUBSCRIBE message where the 'Event' header line indicates 'weather'. The updates section of the XML document defines at what deviation the client would like to receive updates. In this case updates are requested for every degree Celsius and every 30 degrees change in wind-direction and any change in weather condition (currently not provided by the server).

To retrieve real-world information, our server queries a public web service [15] every 10 minutes for weather information (temperature only) and then notifies all interested parties.

6. Future Work

6.1. Workflow, Sensor Data Orchestration

In earlier sections, we described scenarios where the WatchPad was used to receive IMs and/or notifications. These scenarios become even more appealing when they are coupled with enterprise workflow systems, such as processing of travel expenses within a department which typically requires approval from managers as well as HR personnel. When an exception is encountered in such systems, an email is sent to the originator of the workflow asking for clarification. Depending on when the originator responds to the email, this may incur a delay in the workflow. This delay can be alleviated if instead the SIP URI of the originator's WatchPad is registered with the workflow system and an IM is sent to the WatchPad when an exception is encountered. As described earlier, this IM may offer a set of preconfigured responses which would allow the originator to respond quickly to the exception, thus cutting down on processing times of enterprise workflow systems. The same ideas can be tried for orchestrating sensor data traffic to data composition and analysis engines.

6.2. Wearables as Universal Remote Controls?

An interesting emerging scenario could be for the wearable device to directly control the operation of other devices with which it orchestrates the media delivery. For example the wearable device could control the volume or sound quality settings on the media device by sending commands marked up in XML. Other examples could include pausing the media, rewinding, etc. This functionality can be achieved by enhancing the recently proposed MSRP protocol [1] which is designed for session-based instant messaging. The idea is that unlike SIMPLE [8], endpoints first setup a session and then exchange messages directly between the endpoints on the "media path". We envision that these messages could represent commands and responses between the endpoints as long as the command set of the controlled device is made known to the controller. This could for example be

done by initially sending a message that contains a list of commands and explanation, perhaps with XML schemas, of their intended usage which can be displayed on the controller's screen. By appropriate use of tags in this message, the text that represents commands can be highlighted and made selectable. This area needs to be explored more fully in the future.

6.3. Session authentication

Although the watch is equipped with a fingerprint sensor to authenticate users, we have not yet applied it in our scenarios. Future enhancements could involve owner verification for releasing the password for SIP device registration and call establishment. This would prevent others from misusing the system if they gain access to someone's personal device.

6.4. Other issues

Low power technologies that implement only signaling components of SIP would be worth investigating. Systems optimizations to further reduce delays for session-setup would be useful. Charging mechanisms for mixing and matching PSTN devices with VOIP devices have to be developed. Discovery mechanisms to find media devices around the user and also mechanisms to select them through simple pointing need to be developed further. Security issues may also need to be resolved.

7. Conclusions

We have shown through several examples that wearable devices can expand the power of SIP significantly and provide a better user experience. Wearable devices can be used quite effectively to initiate communication through multiple media and switch the end devices that play the media when required. Wearable devices also present a very convenient interface for determining the originator of media streams directed towards the user. The user can then decide not only whether to pick up the media stream but also where to play the stream. We gave specific examples with telephone calls, instant messaging, and publish/subscribe applications. To our knowledge this is one of the earliest, if not the first, investigations to determine the role of wearable devices in this space. Our implementations and testing of several examples on the WatchPad confirm our thinking.

Energy conservation is a major issue for mobile devices. Our approach of transferring calls to stationary devices when possible even if mobile devices are capable of playing the media allows them to reserve their energy for handling the control portions.

At present, limitations in existing mobile and wearable devices prevent continuous connectivity to the network. However, we expect this problem to be solved with advances in 3G networks and advances in power management technology.

SIP is clearly a very powerful technology that is making serious impact on the way we communicate. We have shown that wearable devices can make that impact even more satisfying.

References

1. B. Campbell et al. *The Message Session Relay Protocol*. SIMPLE Working Group Internet-Draft, Jan 2004.
2. B. Campbell, editor. *Session Initiation Protocol (SIP) Extension for Instant Messaging*. RFC 3428.IETF, Dec 2002
3. Handley, M. and V. Jacobson, *SDP: Session Description Protocol*, RFC 2327, IETF Apr 1998.
4. W. Jiang, J. Lennox, H. Schulzrinne and K. Singh. *Towards Junking the PBX: Deploying IP Telephony*. NOSSDAV 2001, pp 177-185.
5. N. Kamijoh, T. Inoue, C. M. Olsen, M. T. Raghunath, C. Narayanaswami, *Energy trade-offs in the IBM Wristwatch computer*. IEEE International Symposium on Wearable Computing, ISWC 2001, pp 133-140.
6. C. Pinhanez, *The Everywhere Displays Projector: A Device to Create Ubiquitous Graphical Interfaces*, Proc. of Ubiquitous Computing 2001 (UbiComp'01), Atlanta, Georgia, September 2001, pp 315-331.
7. C. Narayanaswami et al., *IBM's Linux Watch: The Challenge of Miniaturization*. IEEE Computer, Vol. 35, pp 33-41, Jan 2002.
8. A. Niemi et al., *Session Initiation Protocol (SIP) Extension for Presence Publication*. SIMPLE Working Group Internet-Draft, June 2003.
9. M. T. Raghunath, C. Narayanaswami, *User Interfaces for Applications on a Wrist Watch*, Journal of Personal and Ubiquitous Computing, (2002) 6:17-30.
10. H. Schulzrinne et al. *RTP: A Transport Protocol for Real-Time Applications*. RFC 1889.IETF, Jan 1996.
11. J. Rosenberg et al. *SIP: Session Initiation Protocol*. RFC 3261. IETF, June 2002.
12. J. Rosenberg, G. Camarillo. *Examples of Network Address Translation (NAT) and Firewall Traversal for the Session Initiation Protocol (SIP)*. Internet Draft (work in progress), Dec 2003.
13. H. Sugano et al. *Presence Information Data Format*, draft-ietf-impp-cpim-pidf-08.txt. Internet Draft, IETF May 2003.
14. A. Vemuri and J. Peterson. *Session Initiation Protocol for Telephones (SIP-T): Context and Architectures*. RFC 3372, IETF Sept 2002.
15. Publicly accessible web services, www.xmethods.net
16. The GNU oSIP library. <http://www.gnu.org/software/osip>
17. <http://www.linphone.org>
18. National Institute for Standards and Technology, <http://snad.ncsl.nist.gov>