Application-Layer Mobility Using SIP *

Henning Schulzrinne[a]
schulzrinne@cs.columbia.edu

Elin Wedlund[b]
elin.wedlund@netinsight.se

[a]Dept. of Computer Science, Columbia University
[b]Netinsight

Supporting mobile Internet multimedia applications requires more than just the ability to maintain connectivity across subnet changes. We describe how the Session Initiation Protocol (SIP) can help provide terminal, personal, session and service mobility to applications ranging from Internet telephony to presence and instant messaging. We also briefly discuss application-layer mobility for streaming multimedia applications initiated by RTSP.

I. Introduction

A large amount of effort has been expended over the years on allowing computer and communication devices to continue communicating even when mobile. However, the vast majority of mobile communication devices today continue to be single-service cellular phones. Third generation wireless systems offer the opportunity

However, given the extremely high cost of spectrum[1], other, non-licensed, means of wireless access will likely remain attractive. Thus, system design should make it easy for devices to move between different wireless networks, depending on population density, speed of movement and propagation characteristics. Here, we describe one possible architecture that allows to support a full range of mobility options, independent of the underlying technology. We primarily focus on the provision of multimedia services, but allude briefly to "data" services. We explore how *application-layer* support is necessary to offer more than just hand-off between base stations and subnets, as well as how it can, under some circumstances, compensate for the lack of deployment of mobile IP. The protocol at the heart of this effort is the Session Initiation Protocol (SIP), an IETF-developed signaling protocol.

We begin by outlining the principles of operation of the Session Initiation Protocol (SIP) in Section II, followed by a discussion of four modes of mobility, terminal, session, personal and service mobility in Sections III through VI.

II. Review of the Session Initiation Protocol

The Session Initiation Protocol (SIP) [2, 3, 4, 5] allows two or more participants to establish a session consisting of multiple media streams. The media streams can be audio, video or any other Internet-based communication mechanism, with examples such as distributed games, shared applications, shared text editors and white boards having been demonstrated in practice. The media streams for a single user can be distributed across a set of devices, e.g., specialized audio and video network appliances in addition to a workstation. The protocol is standardized by the IETF and is being implemented by a number of vendors, primarily for Internet telephony. Recently, SIP has been extended to provide presence, event notification and instant messaging services [6, 7].

SIP end points are addressed by SIP URLs that have the form of email addresses, such as `sip:alice@example.com`. While this is not required, it appears likely that many users will re-use at least some of their email addresses as SIP URLs in the future. SIP requests contain a source address and two destination addresses, with one identifying the original, logical destination of the request (the To header), and, in the request URI in the first request line, the current destination (see Fig. 1). The current destination is derived by SIP "application-layer routers", so-called proxies, as explained below. SIP requests also indicate, in the Contact header, where future requests should be sent.

SIP defines a number of logical entities, namely user agents, redirect servers, proxy servers and registrars. User agents originate and terminate requests; examples include conferencing software or gateways to the PSTN, but also voice mail systems. Generally, user agents are the only elements where media and signaling converge. Redirect servers receive requests and return a response that indicates where the requestor should send the request next. For example, a redirect server may keep track of the user's location and then return a response indicating that location, as a list of one or more SIP or other URLs. A proxy server can be either stateful or stateless. A stateless proxy server simply forwards incoming requests to another server, without ensuring the request's reliability. A stateful proxy maintains state for a *transaction*, that is, a request and all responses that belong to that transaction. (A request typically generates one or more *provisional* responses that indicate progress and then a *final* response indicating whether the request succeeded or not.) A stateful proxy can also *fork* a request, i.e., send copies of the request to different destinations. Each such request has a different request URL, but it otherwise the same. Such forking is useful when proxy servers do not know the final destination of the request and need to try various possibilities. Proxy servers can either try a set of destinations in parallel or sequentially, or some combination.

Typically, a "SIP server" implements a redirect and proxy server, with information provided by a built-in registrar. Depending on configuration and the specific request, the server acts as either a proxy or redirect server or a registrar. Consider as an example a request from `alice@wonderland.com`

*This paper is loosely based on the paper "Mobility Support using SIP", by Elin Wedlund and Henning Schulzrinne, published in the *Second ACM/IEEE International Conference on Wireless and Mobile Multimedia (WoWMoM'99)* [1]

[1]Recent auctions have yielded above $1,000 per *potential* subscriber, before a single transmitter or router has been deployed.

addressed to bob@macrosoft.com, as shown in Fig. 1. Typically, Alice would send all requests to a designated local server at wonderland.com. That server recognizes that the request is not meant for it and forwards it to the server for the macrosoft.com domain, say sip.macrosoft.com. (The server is located based on DNS SRV [8] records.) The sip.macrosoft.com server first tries the address bob@b.macrosoft.com, based on the registration of Bob on host b.macrosoft.com. However, Bob is temporarily forwarding his calls to Carol and thus has his Internet phone issue a redirect response to the proxy server. The proxy server, without interacting with Alice, sends an invitation to Carol, which succeeds. The response contains the network address of Carol's computer, which is then used to directly exchange the acknowledgment and later tear down the call via a BYE request.

For larger signaling volumes and higher reliability, proxies can be scaled by replicating them. DNS SRV records allow the requestor to randomly distribute queries across proxies, without the need for "layer-4 routing" entities. Proxies can force subsequent requests within a session to revisit the same server by inserting a Record-Route header into the first request.

SIP user agents typically register their current network location with their local registrar. The registrar is found either by simply sending a multicast REGISTER message to a well-known multicast address, contacting the home domain registrar or using the Service Location Protocol [9]. (The home domain registrar is the registrar in the domain corresponding to the user's SIP URL, located via the DNS SRV records.)

The routing of requests can be influenced by logic executing on proxy or redirect servers [10], in conjunction with indications restricting the choice of destination in requests [11].

SIP requests and responses consist of a text header and a MIME body, very similar to the format of HTTP requests. One major difference is that SIP requests can use any transport protocol, including UDP, with user agents and stateful proxies ensuring request reliability via retransmission for unreliable protocols. SIP defines an extensible set of request methods, currently including in the base specification INVITE to initiate a session, ACK to confirm a session establishment, BYE to terminate a session, OPTIONS to determine capabilities and CANCEL to terminate a session that has not been established yet. The session itself is typically described using the Session Description Protocol [12] that lists media stream addresses, ports and the encodings supported.

Session components and characteristics can be re-negotiated in mid-session, allowing the addition, modification and deletion of media streams or applications. Also, a third-party [13] mechanism allows a user to request that the other participant in a session issue a request to a third party. This mechanism is used to transfer a session and is described in more detail in Section IV.

Recently, SIP has been extended to support presence and instant messaging services [6]. In this model, presence and its generalization, event notification, is seen as the dual of session initiation or signaling. While session initiation queries the potential session members whether they are available and willing to join the session, presence has users notify potential session partners about their availability changes. In many systems, the two modes will be used together, with presence signaling that a friend is available to talk and then using SIP INVITE requests to negotiate the means of communications and establish the actual communication session. In addition, "connectionless" or "datagram" messaging is often useful, as evidenced by the popularity of instant messaging services and the GSM short message service (SMS). To enable such services, SIP only needs to add a MESSAGE method [14].

As we will see below, using the MESSAGE mechanism is particularly useful for mobile end systems since each message is routed independently.

III. Terminal Mobility

Terminal mobility allows a device to move between IP subnets, while continuing to be reachable for incoming requests and maintaining sessions across subnet changes. A subset of terminal mobility, being able to be reached for *new* sessions after subnet changes, requires only DHCP and dynamic DNS.

III.A. Limitations of Mobile IP

Particularly for delay-sensitive multimedia applications, mobile Ip4 has some limitations (also alluded to in Section 2 of [15]), including triangle routing, triangle registration, encapsulation overhead and need for home addresses.

In particular, in its basic form [16], all packets are routed through the home agent, resulting in triangular routing. Binding updates [17] shortcut the data path, but still require that binding updates are tunneled through the home agent, adding hand-off delays. Also, they require changes in operating system of the correspondent host, including authentication mechanisms.

Regardless of the data path, mobile IP encapsulation adds between 8 or 12 [18] and 20 [19] bytes of overhead. Even without explicit encapsulation, IPv6 [20] has to carry an additional 16-byte address, the Home Address destination option. Packet header overhead is particularly significant for low bitrate packet voice where payloads tend to be very short, e.g., a small multiple of 10 bytes for the G.729 8 kb/s codec [21].

To obtain the benefits of IPv4 and IPv6 mobility, a user needs to have a permanent home IP address and needs to convince his ISP to offer home agent services. For IPv4, most "consumer" devices do not have a fixed IP address but rather acquire one dynamically via DHCP only when logged in. Even for cable modems and DSL connections, residential customers generally get only one static IP address. (In many cases, even these permanently-connected devices still get only a DHCP-assigned address to discourage the use of the home PC as a web or Napster server.) Thus, a customer is at the mercy of his ISP to obtain mobility services. With application-layer mobility, the customer has more options. Services similar to the various web-based email services could easily provide SIP mobility. (Indeed, it appears plausible that many users will want to choose a home SIP address that is independent of their transport service provider and may even choose an address identical to their web-based email address.)

The SIP registration mechanism can be considered as the application-layer equivalent of the mobile IP registration mechanism [16]. However, while mobile IP binds a permanent IP address identifying a host to a temporary care-of address,

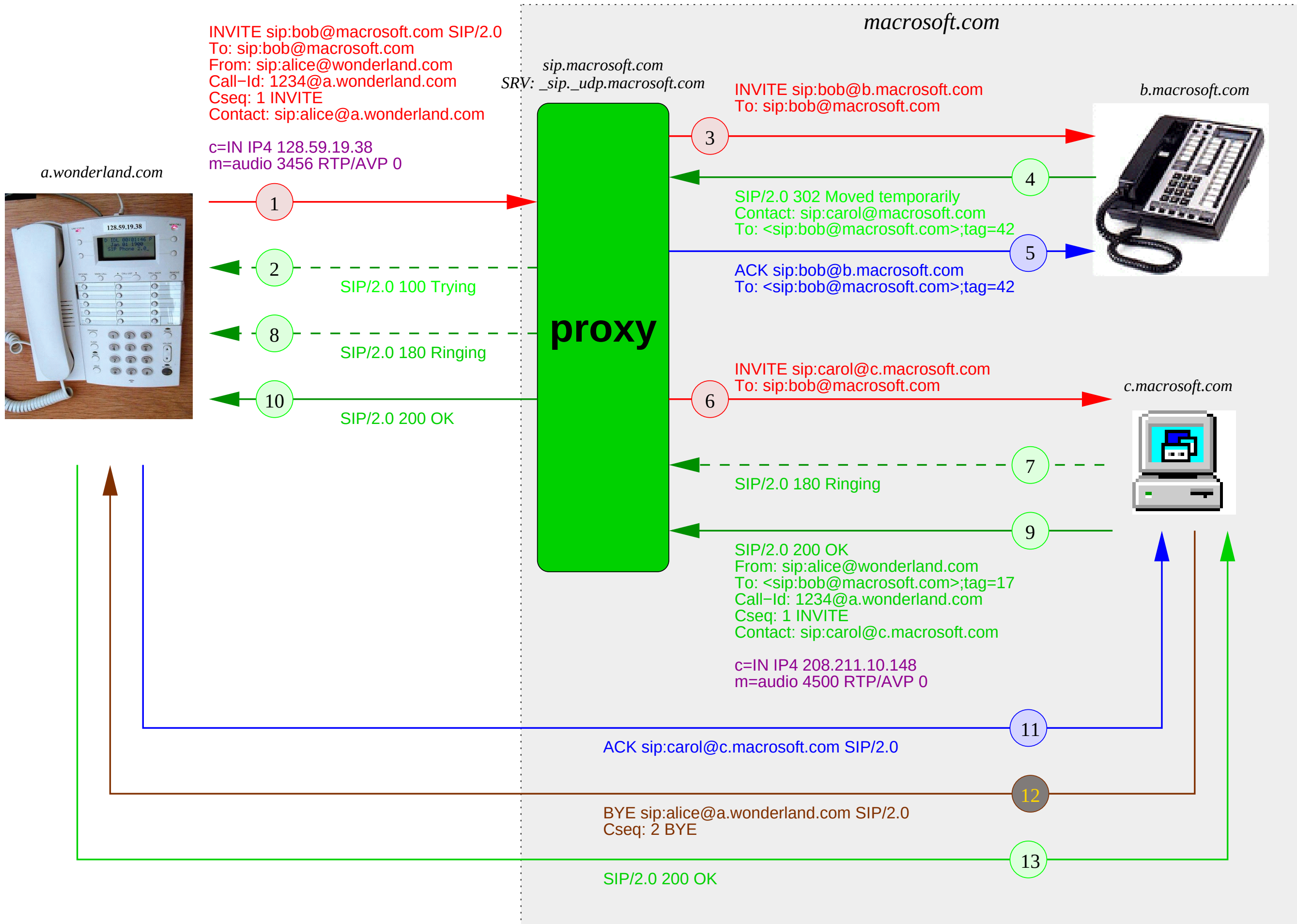

Figure 1: Example of SIP session setup

SIP binds a user-level identifier to a temporary IP address or host name.

III.B. SIP Support for Terminal Mobility

Using SIP for mobility trades generality for ease of deployment. SIP-based mobility is less suitable, as discussed below, for TCP-based applications, but does not require to add capabilities to existing operating system nor the installation of home agents or dynamic DNS updates [22] in the user's ISP.[2] While this may change with the deployment of third generation IP-based networks at least for the wireless portion of the Internet, very few Internet users can avail themselves of IP mobility services for the next few years. Terminal mobility impacts SIP at three stages, pre-call, mid-call and to recover from network partitions, as described below.

III.B.1. Pre-Call Mobility

The easiest part of SIP mobility is the pre-call mobility, where the mobile host (MH) acquires a new address prior to receiving or making a call. The MH simply re-registers with its "home" registrar each time it obtains a new IP address. The only difficult part there is the ability to detect, at the application layer, when the IP address has changed. In our implementation, the client simply polls the OS every few seconds, but the ability to have applications subscribe to be notified of such changes would be preferable.

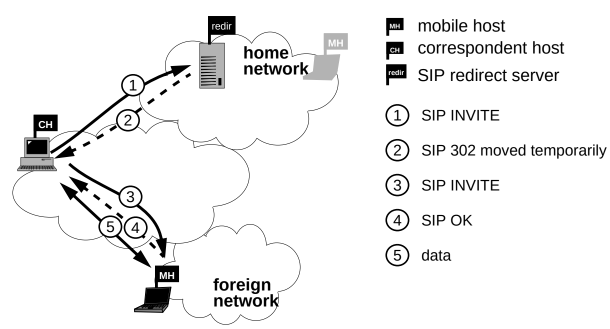

Figure 2: SIP-based pre-call location

Paging, for MH power conservation, can also be implemented in SIP. One approach assigns each device within a domain a unique, scoped IP multicast address. The domain proxy then forwards the INVITE request to that multicast address. Paging with increasing scope requires a bit more work.

[2]Dynamic DNS updates [22] are needed if mobile devices that acquire addresses dynamically in the home network are to be reached as "servers", e.g., called for a VoIP call, rather than just acting as clients.

We assume that proxies are organized hierarchically, e.g., with a proxy for each wireless network, region, cell cluster and base station. In that case, the proxy that has the most recent registration for the current MH location starts the paging operation within its scope if it does not receive a provisional response to the INVITE request. (The SIP registration update mechanism ensures that the call request is routed to the most recent location.) If there is no answer within a short time interval, the proxy reports back a failure to its upstream proxy, i.e., the proxy closer to the caller. The upstream proxy can then multicast the INVITE with a larger scope.

III.B.2. Mid-Call Mobility

For mid-call mobility, the moving MH sends another INVITE request to the correspondent host (CH), without going through any intermediate SIP proxies. (A SIP proxy will be traversed if, during the initial call setup, it has requested to be part of future signaling messages by inserting a Record-Route header.) This INVITE request contains an updated session description with the new IP address. Thus, the location update takes one one-way delay after the application in the MH recognizes that it has acquired a new IP address. For wideband access, the delay is probably equal to the propagation delay plus a few milliseconds, but narrowband systems may impose delays of several tens of milliseconds.

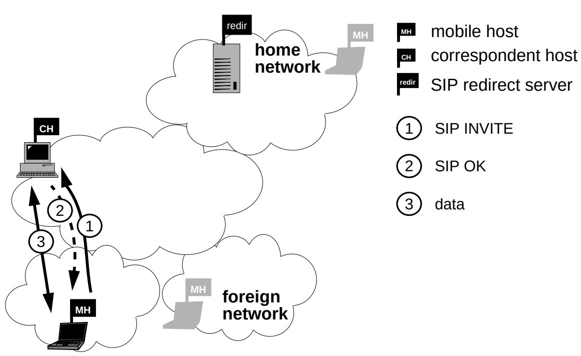

Figure 3: SIP-based hand-off in mid-call

Faster hand-off can be achieved by mechanisms that are similar to the micromobility approaches [23, 24, 25, 26, 27]. Here, the MH advertises not its own address as the media destination, but rather that of the proxy or an RTP translator [28] affiliated with the proxy. Alternatively, the proxy can rewrite the network address in the session description contained in the INVITE requests, so that this mechanism does not necessarily require end system support. The RTP translator intercepts the media packets and directs them to the current location of the MH. It can also buffer media packets, to transmit them to the new location after hand-offs. (Any duplicate packets are discovered at the RTP layer.) In addition, such a translator may also be useful for transcoding media to a lower bandwidth or adding forward error correction. The SIP proxy could, for example, use Megaco [29] or MGCP [30]. In an alternate implementation, SIP signaling is split, terminating from the perspective of the caller at a server in the network. That server then signals the MH.

Insertion of an RTP translator reduces hand-off delay to the one-way delay between the MH and the first proxy that is shared between the old and new location.

Soft hand-off at the IP level is difficult to support at the application layer. The CH would have to send two data streams to both the old and new IP address. The RTP translator approach can, however, also help here.

Hand-offs need to be secured to avoid that an intruder sends a message redirecting the media stream to a location where it can conveniently listen in. SIP supports three authentication mechanisms, namely HTTP-style basic and digest authentication that use shared secrets and PGP, using public-key cryptography. A very rudimentary protection against intruders stealing calls is also provided by the random call identifier. Thus, the intruder would have to be able to monitor the packet exchanges between the parties.

While undesirable from an efficiency perspective, triangle routing in mobile IP has one advantage in that it hides the current network location of the terminal which may also allow inferences about the rough physical location of the terminal. SIP-based mobility does not, as the IP address of the terminal is disclosed in the session description. If SIP terminals desire to hide their IP address, they need to find an anonymizer service. An anonymizer service is a back-to-back SIP UA that terminates media streams. However, a calling terminal could choose an anonymizer that is "mid-way" between the two parties, thus, improving performance. This anonymizer service can be combined with the micromobility system described above.

III.B.3. Network Partitions

If the network partition lasts less than about thirty seconds, SIP will recover without further mechanisms, as it retransmits the request if there is no answer. If the network partition lasts longer, updates may be lost and the other host may also have moved. In that case, to rendezvous again, each side should address the SIP INVITE request to the canonical address, the home proxy of the other side. Recovery from such partitions can be done automatically if the user agents implement the SIP session timer mechanism [31] that automatically causes a refresh of the session at user-configurable intervals.

III.C. Hierarchical Registration

By default, registrations are sent to the "home" registrar, as explained earlier. Thus, any location change causes a SIP REGISTER request and response to be sent. Although SIP signaling is likely to use a higher-speed network than most traditional SS7/MAP-based networks, this signaling traffic is still undesirable. Within SIP, registrations can be proxied just like other requests, as shown in Fig. 4. In the figure, Alice, with a home in New York, visits California. Each time she moves, she sends a REGISTER request towards her home registrar, through the outbound proxy in California. For the first REGISTER, originating in San Francisco, the outbound proxy makes a note of the registration and then forwards the request to the normal home registrar, after modifying the Contact in the registration to point to it rather than Alice's mobile host. After Alice travels to Los Angeles, the REGISTER update hits the same registrar (CA). It recognizes that Alice is already in California and does not forward the request. A call from anywhere first reaches the NY proxy server, which forwards the request to the CA proxy server, which in turn forwards it to Alice's MH.

The mechanism described here works whether the ISP in California is the same has Alice home ISP or not. While only a single level of indirection is shown, the ISP in California could nest this as deeply as desired, with a hierarchy of "outbound proxies".

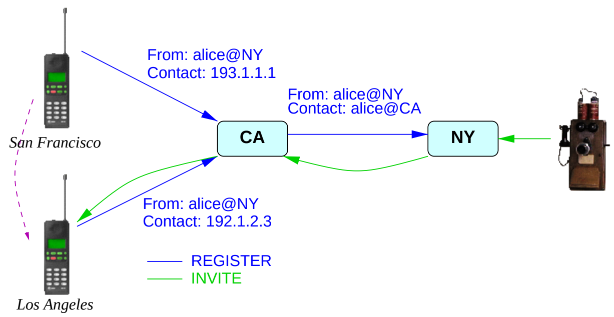

Figure 4: Hierarchical registration in SIP

III.D. RTP Issues

Unlike TCP, RTP does not use the IP address to maintain associations between end systems. Rather, end systems are identified by randomly chosen 32-bit identifiers, the SSRC. However, IP addresses are used to detect *collisions*, where two end systems accidentally pick the same SSRC value. The current specification suggests keeping packets from the old IP address; however, for mobility, it is preferable to keep packets from the new address.

RTP could also be used as an indicator of mobility. As soon as an RTP packet with a known SSRC and a new IP address arrives, the receiving host redirects its media stream to that new address. This is likely to be somewhat faster, since it does not have to wait until the application becomes aware of its change of IP address. However, it fails badly if a collision occurs and opens an obvious method for an intruder to redirect an existing phone call. However, if the RTP stream is encrypted, the receiver can check the validity of the packet and easily rule out collisions or attempts to steal the call.

III.E. Hand-over Performance

A complete timeline for the SIP-based hand-over mechanism is shown in Fig. 5, using an IEEE 802.11 network as an example. The timeline shows all packets exchanged when the MH enters radio range of a new 802.11 base station. Typical beacon intervals are around 80 to 100 ms. The figure ignores a potentially large delay component DHCP [32], namely if the DHCP server attempts to use ICMP echo requests to determine if the new address has already been assigned. It also ignores any AAA (Authorization, Authentication, Accounting) delays that would be incurred on inter-domain hand-offs as these will strongly depend on the architecture chosen.

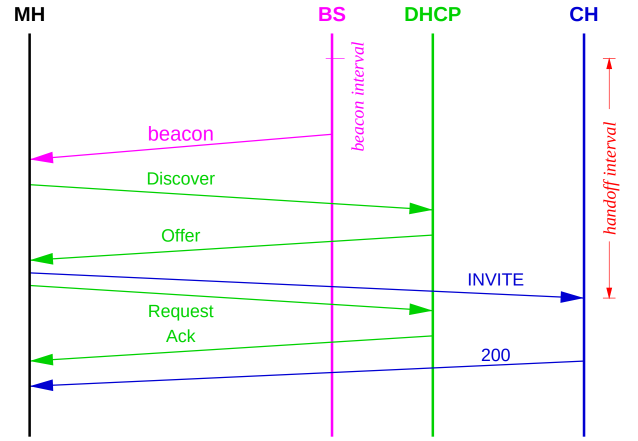

Figure 5: Timing for SIP-controlled terminal hand-off in IEEE 802.11

III.F. Supporting TCP-based Applications

Providing terminal mobility for TCP-based applications is only difficult if applications are to maintain TCP connections across subnet changes. For many mobile client applications with short-lived connections, only DHCP is required. For example, HTTP clients can minimize the chance that a connection is interrupted by a subnet change by inserting a Connection: close header into an HTTP/1.1 [33] request. Alternatively, applications can implement application-layer restart and recovery capabilities, for example, using the ftp SIZE and REST (restart) facility [34] or the HTTP Range header. (HTTP application-layer retry only works for idempotent requests.) Application-layer restart is likely to be useful in mobile environment in any event, given the higher likelihood of disconnects.

Overall, while application-layer recovery is useful and can overcome the lack mobile IP support, it is unlikely that all protocols and protocol implementations will support it any time soon.

The Stream Control Transmission Protocol (SCTP, [35]), a new reliable transport protocol developed recently within the IETF, may be able to avoid the need to maintain a constant destination IP address by using its multi-homing feature to allow for mid-session subnet changes. It remains to be explored whether this is feasible.

III.G. Streaming Multimedia Applications

Streaming media sessions are commonly controlled by RTSP [36] (Fig 6). RTSP is used to create sessions between a media server and a media client. Media delivery can be either via unicast or multicast. For unicast, RTSP also has a built-in mechanism for application-layer mobility. In that case, the client simply sends a SETUP request to the server during the session, with the new IP address. It can include an indication of the last packet received to ensure a smooth transition. Since the playout delay for streaming media is generally longer than for interactive sessions, hand-off delays are not as likely to cause disruptions, even without hand-off optimizations.

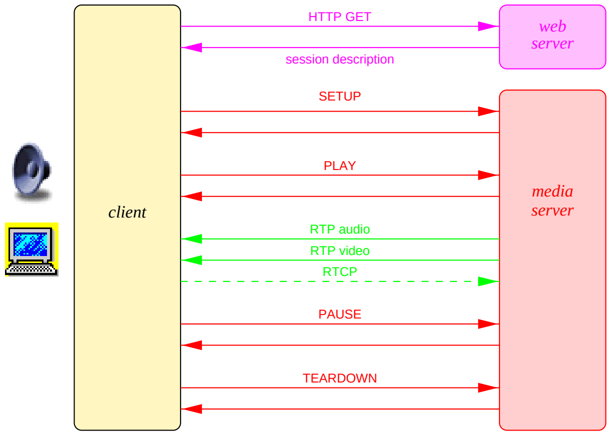

Figure 6: RTSP protocol session

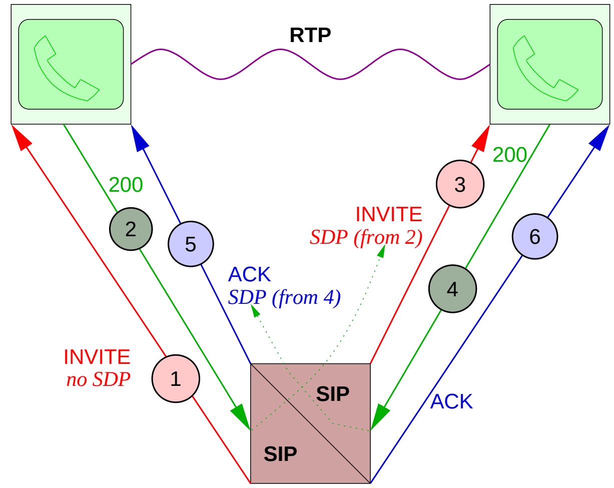

Figure 7: Session mobility using third-party call control

IV. Session Mobility

Session mobility allows a user to maintain a media session even while changing terminals. For example, a caller may want continue a session begun on a mobile device on the desktop PC when entering her office. A user may also want to move parts of a session, e.g., if he has specialized devices for audio and video, such as a video projector, video wall or speakerphone.

IPv4 or IPv6 mobility does not directly support such session mobility. It can be approximated at the IPv6 layer via anycast [37, 38], but it is not clear that this is sufficiently flexible and reliable as it requires coordinated dynamic address assignment and hand-off. Also, the new end application would need an application mechanism to find out the session parameters.

Session mobility using SIP can be supported in at least three ways. In the simplest approach, end systems that are to receive and send a media stream are somehow configured by the primary end system, which then conveys their IP addresses and ports to the other party using a new INVITE request. One mechanism for such configuration could be MGCP [30] or Megaco [29]. There are two better solutions, namely third-party call control [39] or the REFER mechanism, shown in Fig. 7 and 8, respectively. In our examples, we assume that Alice is in a session with Bob at a mobile host (`bob@mobile`) and wants to move the session to Bob at a fixed host (`bob@fixed`).

In third-party call control, the original session participant (Bob) sends an INVITE request to `box@fixed`, the new session destination, indicating the session parameters, such as IP address, of the remote session participant, Alice. `Bob@mobile` also sends Alice the session description generated by `Box@fixed`, so that Alice sends media streams to bob@fixed instead of `bob@mobile`. Bob could also split the session into components, with different receivers for each media type. This approach has the disadvantage that the original session participant has to remain involved in the session, as it will be contacted to change or terminate the session.

A cleaner solution explicitly transfers the session to the new destination. Here, `bob@mobile` simply sends a REFER request to Alice, indicating that she should contact `bob@fixed`. Alice then negotiates a session with `bob@fixed` using the regular INVITE exchange. (Alternatively, Bob can also send a REFER request to `bob@fixed`, asking her to invite Alice.) If the session is to be split across two participants, Bob has to invite both, say, `bob@audio` and `bob@video`.

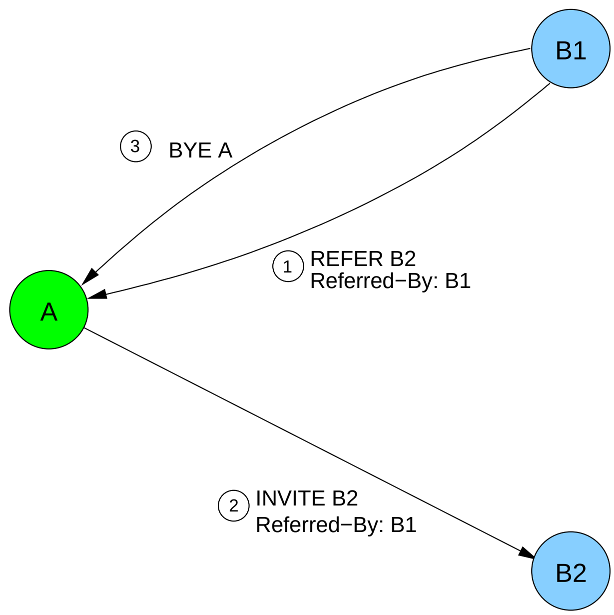

Figure 8: Session mobility using call transfer

V. Personal Mobility

Personal mobility allows to address a single user located at different terminals by the same logical address [40]. Both 1-to-n (one address, many potential terminals) and m-to-1 (many addresses reaching one terminal) mappings are useful, as illustrated in Fig. 9. For example, user `alice` may want to be reachable via a traditional PSTN phone, a PC and a wireless device. She may use these devices either at the same time or alternate between them. Using SIP forking proxies, Alice can be reached at any of the devices via the same name, making her device choice transparent to third parties. Also, it appears likely that, just as for email, users will advertise different addresses for different purposes, e.g., for private and professional contacts. In addition, telephone numbers may well serve as an additional alias for a while, particularly since many of the SIP-based Internet telephones still only have a twelve-button keypad.

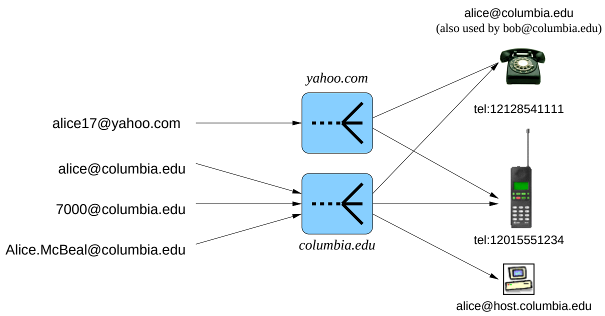

Figure 9: Example of personal mobility

One practical problem is that registrars need to be able to recognize different devices as belonging to the same person. In our current implementation of a SIP proxy server (sipd), we use a number of heuristics so that registrations from `alice@columbia.edu`, `7000@columbia.edu` and `Alice.McBeal@columbia.edu` are all part of the same logical entity.

VI. Service Mobility

Service mobility allows users to maintain access to their services even while moving or changing devices and network service providers. In a voice-over-IP environment, simple services that users will likely want to maintain include their speed dial lists, address books, call logs, media preferences, buddy lists and incoming call handling instructions. Call handling instructions could be encoded in a variety of format, with the Call Processing Language (CPL) [10, 41, 42] as one possible portable and system-independent format. It should also be possible to update these service definitions from any terminal, without having to then explicitly synchronize them. (Unlike in the typical PDA scenario, it is less clear which device holds the "master" copy.)

One solution for service mobility is to have the user carry this information with him, either as a PDA [43] or as a memory chip (e.g., CompactFlash). Certainly, even a basic 8 MB CompactFlash card should have enough memory for most user configurations. (The SIM smart card in GSM mobile phones is a simple predecessor of this approach, with about 16 to 32 kB of EEPROM.) However, even with local storage, updates made on any one of the user's end systems still needs to propagate to the other devices, even if the device performing the update and the other devices are never in the same place. This requires network storage.

Formats for many data elements such as speed dial lists or user interface configurations remain to be standardized, but SIP offers a basic mechanism for synchronizing this type of service data across servers. The architecture is predicated on having a "home" server, associated with the user's address. For example, a user identified as `alice@wonderland.com` would use the designated SIP server for the `wonderland.com` domain to store service information. Since it is likely that users will maintain several different SIP identities, similar to the number of email addresses, only the user's end system can be used to propagate service information among all domains.

SIP applications register with a registrar generally about once an hour, or whenever the network address changes. Registration conveys three pieces of information to the registrar: the current network address, properties of the device (e.g., languages spoken, personal vs. business use, media supported and minimum call priority) and one or more user configuration elements.

For example, a user agent currently located at `host42.example.com` whose owner speaks English, Spanish and German, can handle audio, video and a chat application, has full-duplex capability and only wants to receive urgent calls includes

```
Contact: Carol <sip:carol@example.com>
   ;language="en,es,de"
   ;media="audio,video,application/chat"
   ;duplex="full"
   ;priority="urgent"
```

A SIP user agent also uploads its timestamped version of the configuration information [44]. The server either updates its own version or returns a more recent copy in the registration response. However, this assumes that there is a single server handling services for a given user. With registration proxying, updates of service configuration in the mobile terminal or the "home" server will not necessarily be propagated expeditiously. It may be sufficient, however, to simply have the regular, hourly registration updates go to the home server.

We envision that a terminal uses PDAs [43], physical tokens such as the i-button or biometrics to recognize their users.

VII. Conclusion

Application-layer mobility can either partially replace or complement network-layer mobility. We have tried to show that for interactive sessions, SIP-based mobility can be used to provide all common forms of mobility, including terminal, personal and service mobility. However, for terminal mobility, an IPv6-based solution is likely to be preferable, as it applies to all IP-based applications, rather than just Internet telephony and conferencing. In the absence of home agents, however, SIP-based mobility can provide mobility services to the most important current mobile application, telephony.

References

[1] E. Wedlund and H. Schulzrinne, "Mobility support using SIP," in *Second ACM/IEEE International Conference on Wireless and Mobile Multimedia (WoWMoM'99)*, (Seattle, Washington), Aug. 1999.

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

[3] H. Schulzrinne and J. Rosenberg, "Internet telephony: Architecture and protocols – an IETF perspective," *Computer Networks and ISDN Systems*, Vol. 31, pp. 237–255, Feb. 1999.

[4] J. Rosenberg and H. Schulzrinne, "The IETF internet telephony architecture and protocols," *IEEE Network*, Vol. 13, pp. 18–23, May/June 1999.

[5] H. Schulzrinne and J. Rosenberg, "The session initiation protocol: Internet-centric signaling," *IEEE Communications Magazine*, Vol. 38, Oct. 2000.

[6] J. Rosenberg *et al.*, "SIP extensions for presence," Internet Draft, Internet Engineering Task Force, June 2000. Work in progress.

[7] D. Marples, "Naming and accessing internet appliances using extensions to the session initiation protocol," in *Proc. of SIP 2000 Conference and Exhibition*, (Paris, France), May 2000.

[8] A. Gulbrandsen, P. Vixie, and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)," Request for Comments 2782, Internet Engineering Task Force, Feb. 2000.

[9] J. Kempf and J. Rosenberg, "Finding a SIP server with SLP," Internet Draft, Internet Engineering Task Force, Feb. 2000. Work in progress.

[10] J. Rosenberg, J. Lennox, and H. Schulzrinne, "Programming internet telephony services," *IEEE Network*, Vol. 13, pp. 42–49, May/June 1999.

[11] H. Schulzrinne and J. Rosenberg, "SIP caller preferences and callee capabilities," Internet Draft, Internet Engineering Task Force, July 2000. Work in progress.

[12] M. Handley and V. Jacobson, "SDP: session description protocol," Request for Comments 2327, Internet Engineering Task Force, Apr. 1998.

[13] R. Sparks, "SIP call control," Internet Draft, Internet Engineering Task Force, July 2000. Work in progress.

[14] J. Rosenberg *et al.*, "SIP extensions for instant messaging," Internet Draft, Internet Engineering Task Force, June 2000. Work in progress.

[15] D. Johnson and C. Perkins, "Mobility support in IPv6," Internet Draft, Internet Engineering Task Force, Apr. 2000. Work in progress.

[16] C. Perkins, "IP mobility support," Request for Comments 2002, Internet Engineering Task Force, Oct. 1996.

[17] C. Perkins and D. Johnson, "Route optimization in mobile IP," Internet Draft, Internet Engineering Task Force, Feb. 2000. Work in progress.

[18] C. Perkins, "Minimal encapsulation within IP," Request for Comments 2004, Internet Engineering Task Force, Oct. 1996.

[19] C. Perkins, "IP encapsulation within IP," Request for Comments 2003, Internet Engineering Task Force, Oct. 1996.

[20] S. Deering and R. Hinden, "Internet protocol, version 6 (ipv6) specification," Request for Comments 1883, Internet Engineering Task Force, Dec. 1995.

[21] H. Schulzrinne, "RTP profile for audio and video conferences with minimal control," Request for Comments 1890, Internet Engineering Task Force, Jan. 1996.

[22] P. Vixie, Ed., S. Thomson, Y. Rekhter, and J. Bound, "Dynamic updates in the domain name system (DNS UPDATE)," Request for Comments 2136, Internet Engineering Task Force, Apr. 1997.

[23] A. G. Valk, "Cellular IP: A new approach to internet host mobility," *ACM Computer Communication Review*, Vol. 29, pp. 50–65, Jan. 1999.

[24] R. Ramjee, T. L. Porta, S. Thuel, K. Varadhan, and S.-Y. Wang, "HAWAII: A domain-based approach for supporting mobility in wide-area wireless networks,," in *International Conference on Network Protocols (ICNP)*, (Toronto, Canada), Nov. 1999.

[25] S. Das, A. Misra, P. Agrawal, and S. K. Das, "TeleMIP: Telecommunications-enhanced mobile IP architecture for fast intradomain mobility," *IEEE Personal Communications Magazine*, Vol. 7, pp. 50–58, Aug. 2000.

[26] A. T. Campbell, J. Gomez, S. Kim, A. Valk, C.-Y. Wan, and Z. R. Turnyi, "Design, implementation, and evaluation of cellular IP," *IEEE Personal Communications Magazine*, Vol. 7, pp. 42–49, Aug. 2000.

[27] R. Ramjee, T. F. LaPorta, L. Salgarelli, S. Thuel, K. Varadhan, and L. Li, "Ip-based access network infrastructure for next-generation wireless networks," *IEEE Personal Communications Magazine*, Vol. 7, pp. 34–41, Aug. 2000.

[28] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: a transport protocol for real-time applications," Request for Comments 1889, Internet Engineering Task Force, Jan. 1996.

[29] F. Cuervo, N. Greene, C. Huitema, A. Rayhan, B. Rosen, and J. Segers, "Megaco protocol 0.8," Request for Comments 2885, Internet Engineering Task Force, Aug. 2000.

[30] M. Arango, A. Dugan, I. Elliott, C. Huitema, and S. Pickett, "Media gateway control protocol (MGCP) version 1.0," Request for Comments 2705, Internet Engineering Task Force, Oct. 1999.

[31] S. Donovan and J. Rosenberg, "SIP session timer," Internet Draft, Internet Engineering Task Force, July 2000. Work in progress.

[32] J.-O. Vatn and G. Q. M. Jr., "The effect of using co-located care-of addresses on macro handover latency," in *Proc. of 14th Nordic Tele-traffic Seminar*, (Technical University of Denmark, Lyngby, Denmark), Aug. 1998.

[33] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext transfer protocol – HTTP/1.1," Request for Comments 2616, Internet Engineering Task Force, June 1999.

[34] R. Elz and P. Hethmon, "Extensions to FTP," Internet Draft, Internet Engineering Task Force, July 2000. Work in progress.

[35] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxon, "Stream control transmission protocol," Internet Draft, Internet Engineering Task Force, July 2000. Work in progress.

[36] H. Schulzrinne, A. Rao, and R. Lanphier, "Real time streaming protocol (RTSP)," Request for Comments 2326, Internet Engineering Task Force, Apr. 1998.

[37] C. Partridge, T. Mendez, and W. Milliken, "Host anycasting service," Request for Comments 1546, Internet Engineering Task Force, Nov. 1993.

[38] R. Hinden and S. Deering, "IP version 6 addressing architecture," Request for Comments 2373, Internet Engineering Task Force, July 1998.

[39] J. Rosenberg, H. Schulzrinne, and J. Peterson, "Third party call control in SIP," Internet Draft, Internet Engineering Task Force, Mar. 2000. Work in progress.

[40] H. Schulzrinne, "Personal mobility for multimedia services in the Internet," in *European Workshop on Interactive Distributed Multimedia Systems and Services (IDMS)*, (Berlin, Germany), Mar. 1996.

[41] J. Lennox and H. Schulzrinne, "Call processing language framework and requirements," Internet Draft, Internet Engineering Task Force, Jan. 2000. Work in progress.

[42] J. Lennox and H. Schulzrinne, "CPL: a language for user control of internet telephony services," Internet Draft, Internet Engineering Task Force, Mar. 1999. Work in progress.

[43] I. Dalgic, M. Borella, R. Dean, J. Grabiec, J. Mahler, G. Schuster, and I. Sidhu, "True number portability and advanced call screening in a SIP-based IP telephony system," white paper, 3Com, Santa Clara, California, Mar. 2000.

[44] J. Lennox and H. Schulzrinne, "Transporting user control information in SIP REGISTER payloads," Internet Draft, Internet Engineering Task Force, Mar. 1999. Work in progress.