Signaling for Networked Appliances

Henning Schulzrinne
Dept. of Computer Science
Internet Real-Time Laboratory (IRT)
Columbia University
New York, New York
schulzrinne@cs.columbia.edu

Panasonic (Princeton, New Jersey)

December 8, 2000

With Jonathan Rosenberg, Jonathan Lennox, Kundan Singh, Adam Roach and other participants in the SIP WG

Overview

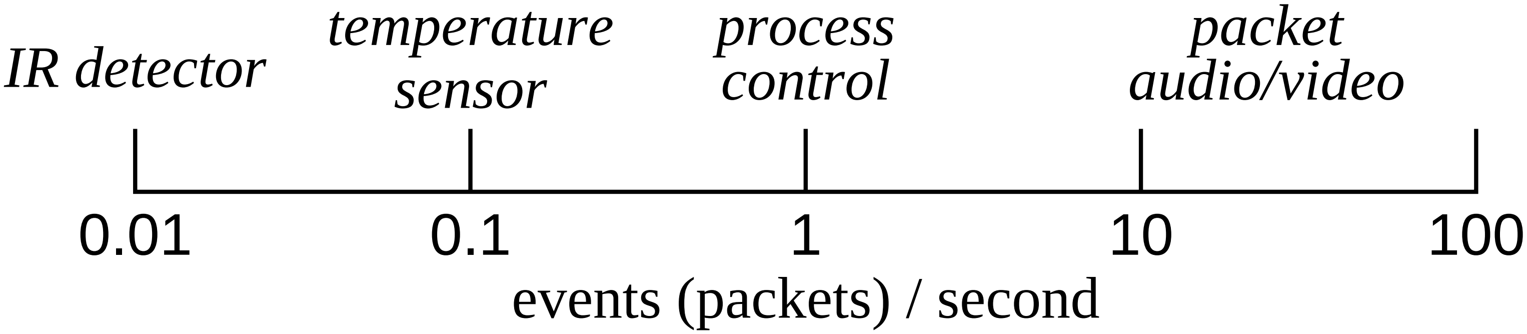
- networked appliances
- requirements for control
- the Session Initiation Protocol (SIP)
- generic event notification
- instant messaging & presence
- programming services

Networked appliances

- household devices: “home automation” ➠ light switches, thermostats, IR presence detectors, door alarms, thermometers, . . .
- entertainment systems: video cameras, CD changers, (MP3) radios, . . .
- industrial control: sensing and controlling environment, machinery
- may be built into other devices, e.g., Internet telephone

Observations

- single-valued (light-switch) to complex (CD changer) to multi-valued (temperature samples)
- both built-in and mediated (X10)
- often combined with audio/video in same system: security, industrial control, home entertainment
- notification rates vary ➠ gradual transition to continuous media



Current options for control

- OSGi (http://www.osgi.org)
- HAVi (http://www.havi.org)
- UPnP (http://www.upnp.org)
- Jini (http://www.jini.org)
- X.10 (http://www.x10.org): very low command rate, few bits/packet, bidirectional, 256 addresses/home, but very cheap
- Bluetooth (http://www.bluetooth.com)
- Salutation (http://www.salutation.org)

Requirements for control

- work both in local network and across Internet
- security ➠ don’t assume trusted network
- human-friendly naming: A10 ⟶ bedroom lamp
- integrate with continuous media
- control not just through web browser, but also master controllers ➠ not just built-in web browser
- small footprint
- language-neutral
- buy-and-use, without (network) configuration

“Mobile” appliances

- most appliances don’t walk around
- but they do move in the network: different BlueTooth base stations, lend to friend, . . .
- more importantly, notification target moves around: home, work, security monitoring station, . . .

Architecture proposal

- integrate into common architecture for Internet-wide notification and messaging

  ➠ new basic internet service:

Asynchronous messaging with pickup	SMTP + POP/IMAP
Data retrieval	HTTP, ftp, tftp
Export computer UI	telnet, ssh, X11, vnc
Synchronous messaging	**SIP**

The largest signaling network is not running SS7

- AT&T: 280 million calls a day
- AOL: 110 million emails/day, total about 18 billion/day
- total $>$ 1 billion instant messages a day (AOL: 500 million)
- signaling effort of call $\approx$ IM

Session Initiation Protocol (SIP)

- IETF standards-track protocol (RFC 2543)
- request-response (message) protocol
- runs over UDP, TCP, SCTP, . . .
- message header + MIME body
- now widely used for Internet telephony: phones, gateways, soft clients, proxy servers, . . .

Example SIP exchange (simplified)

```
INVITE   sip:bob@pc42.macrosoft.com  SIP/2.0
To:      Alice <sip:alice@wonderland.com>
From:    Bob <sip:bob@macrosoft.com>
Call−ID: 17548xw@wonderland.com
CSeq:    1 INVITE

v=0 ...
c=IN IP4  128.59.16.1
m=audio  47192 RTP/AVP 0
```

SIP header

session description

```
SIP/2.0 200 OK
To:      Alice <sip:alice@wonderland.com>
From:    Bob <sip:bob@macrosoft.com>
Call−ID: 17548xw@wonderland.com
CSeq:    1 INVITE

v=0 ...
c=IN IP4  152.1.2.4
m=audio  16922 RTP/AVP 0
```

SIP features

Naming: user@host or device@home

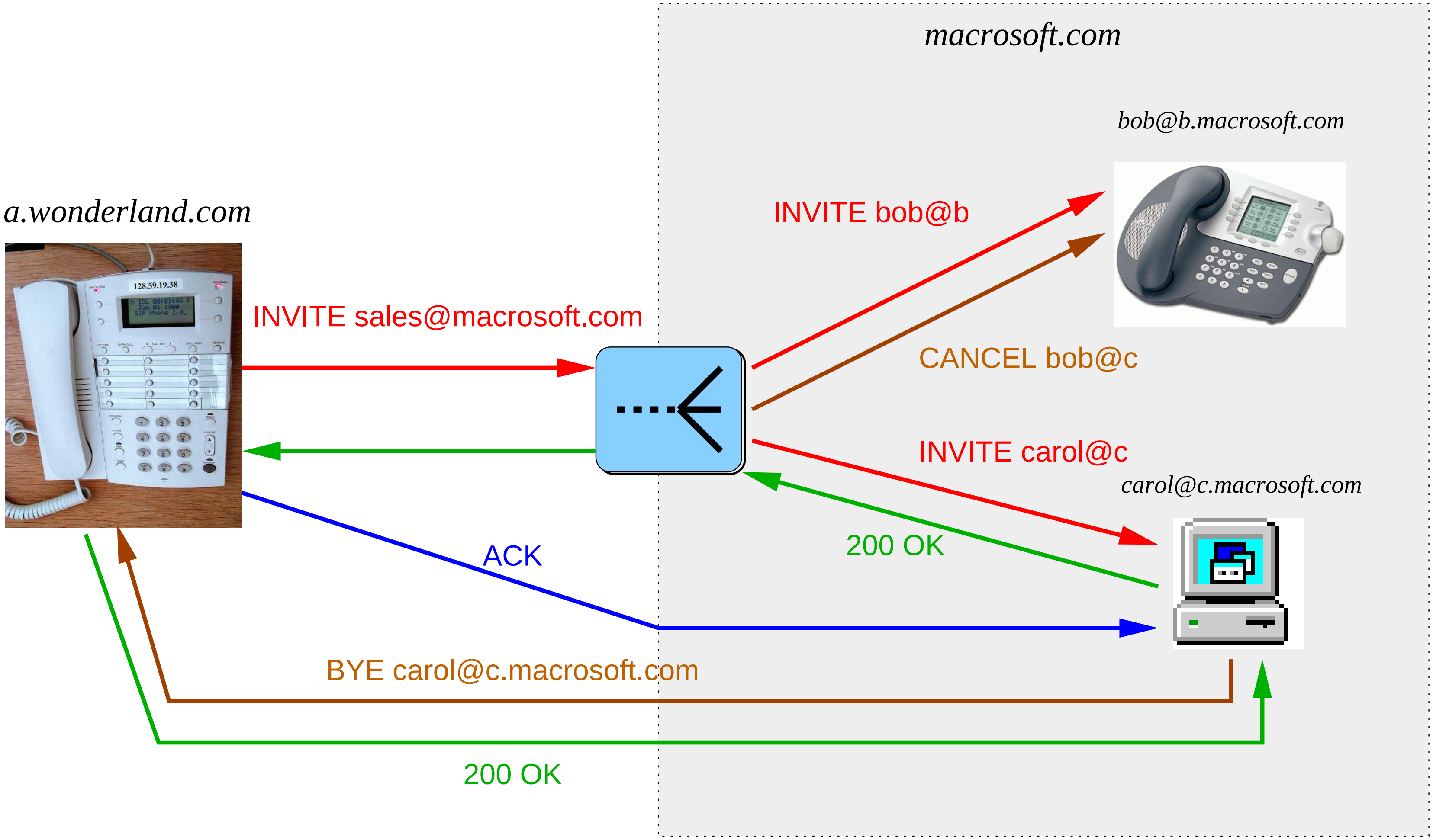
Security: authenticate callers, encrypt content ➠ basic, digest, PGP, S/MIME

Forking: multiple destinations with same name, ACD

Content-neutral: any attachment, multi-part

Extensible: common base, negotiate features, add headers

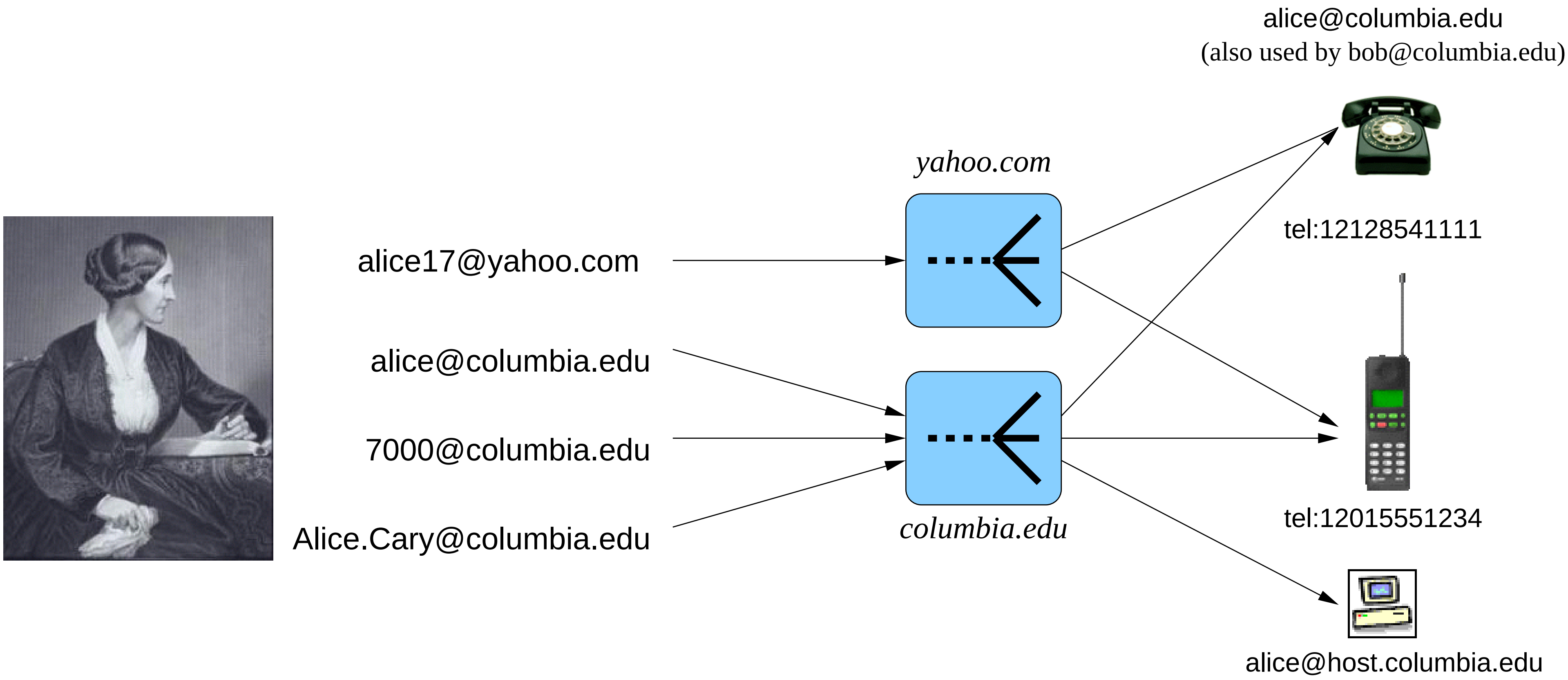
SIP forking


macrosoft.com
bob@b.macrosoft.com
a.wonderland.com
INVITE bob@b
INVITE sales@macrosoft.com
CANCEL bob@c
INVITE carol@c
carol@c.macrosoft.com
200 OK
ACK
BYE carol@c.macrosoft.com
200 OK

SIP mobility

terminal	cross-provider	REGISTER, re-INVITE
personal	different terminals, same address	REGISTER
service	different terminals, same services	upload
session	move sessions across terminals	REFER

SIP personal mobility



alice@columbia.edu
(also used by bob@columbia.edu)

yahoo.com

tel:12128541111

alice17@yahoo.com

alice@columbia.edu

7000@columbia.edu

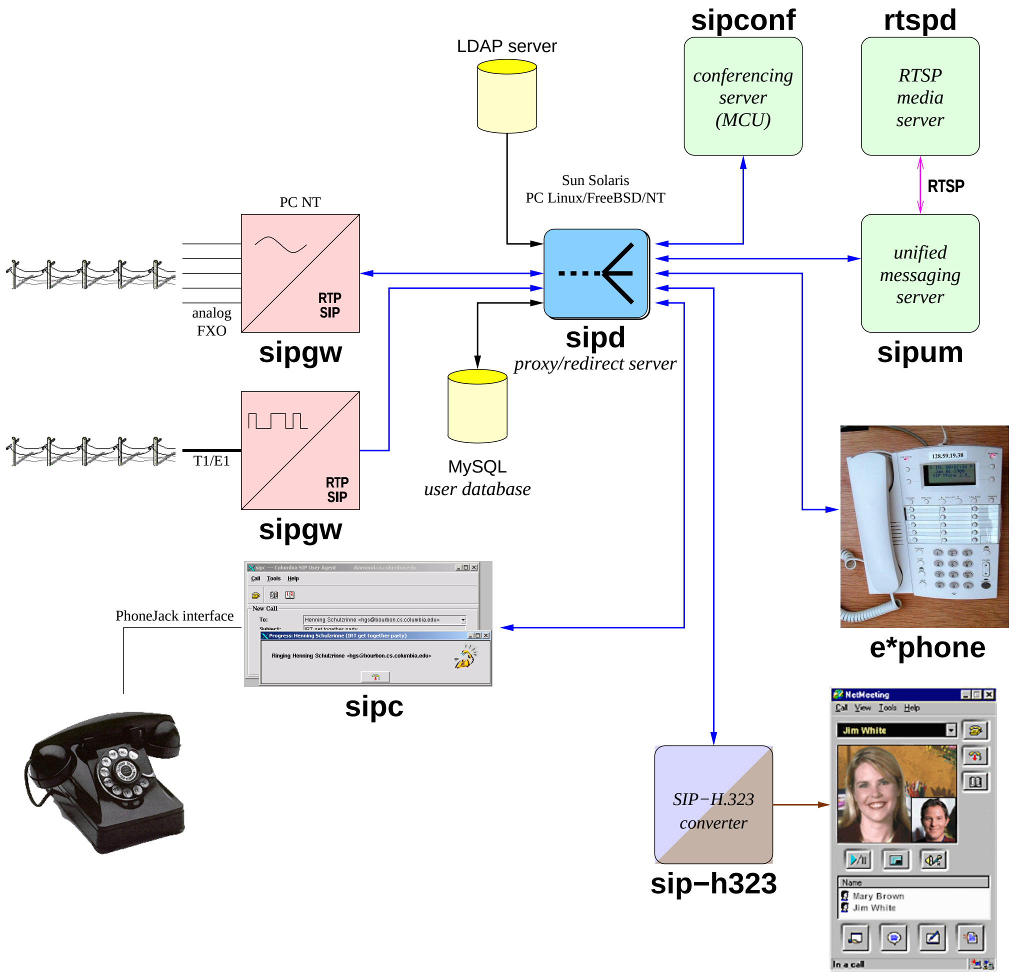
Alice.Cary@columbia.edu

columbia.edu

tel:12015551234

alice@host.columbia.edu

Example SIP system

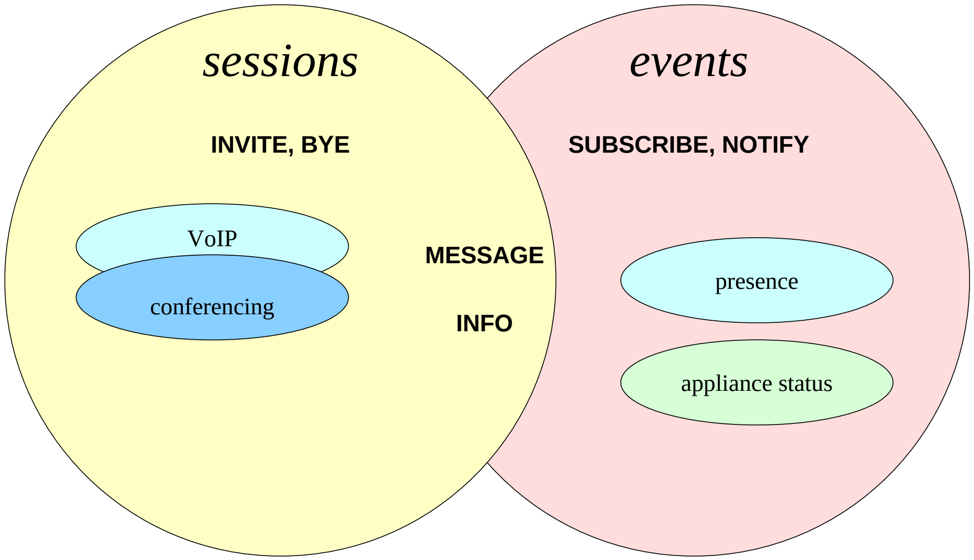


Invisible Internet telephony

VoIP technology will appear in . . .

- Internet appliances
- home security cameras, web cams
- 3G mobile terminals
- fire alarms
- chat/IM tools
- interactive multiplayer games
- 3D worlds: proximity triggers call

Signaling and events



Signaling: “do this” (push) – Events: “this just happened”

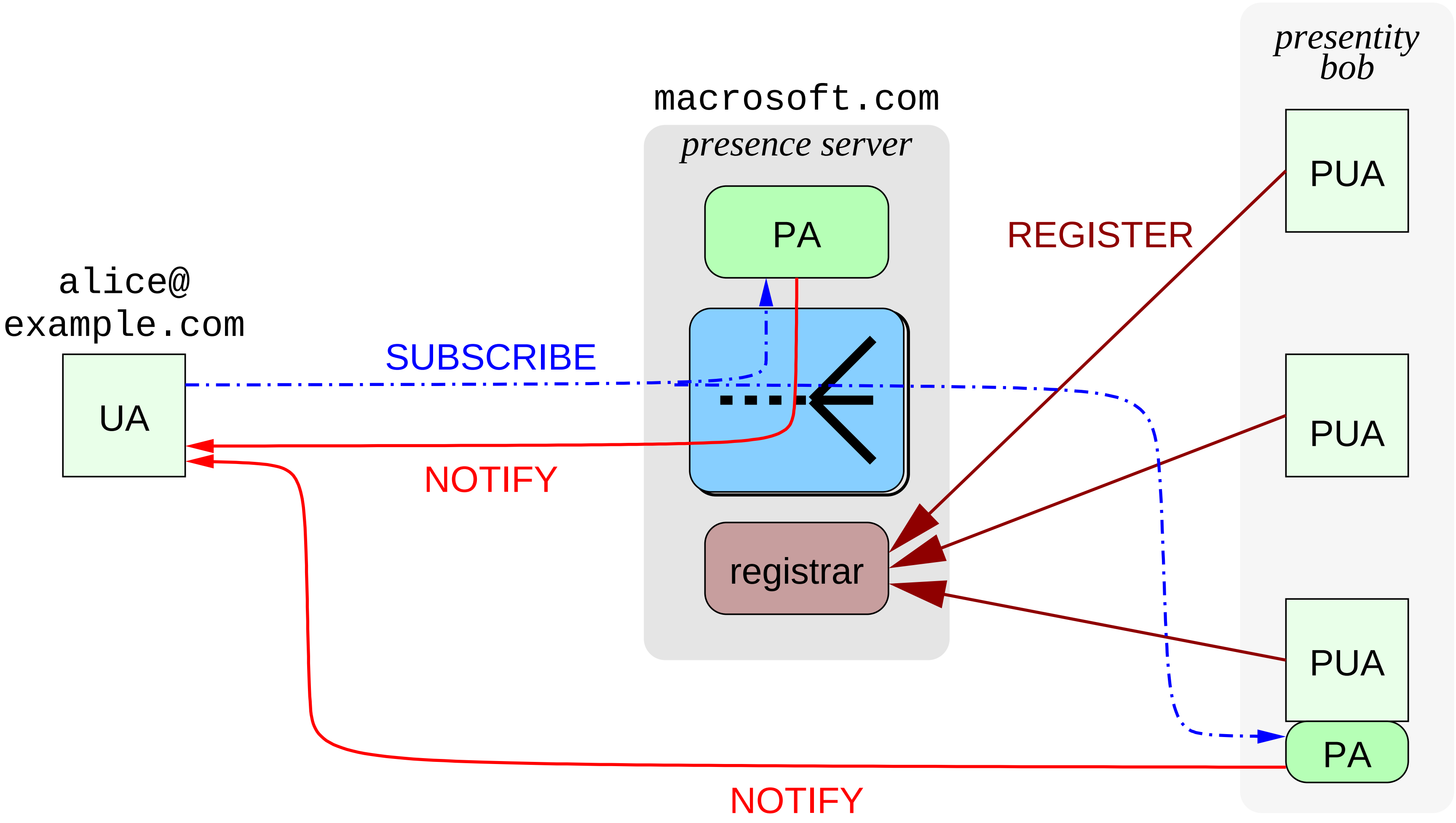
Commonalities between signaling and events

- presence is just a special case of events: “Alice just logged in” ≈ “temperature in boiler exceeds 300° F”
- need to *locate* mobile end points
- may need to find several different destinations (“forking”)
- same addressing for users
- presence often precursor to calls
- may replace call back and call waiting
- likely to be found in same devices
- events already in VoIP: message alert, call events

SIP as a presence & event platform

- requires minimal extensions to SIP: SUBSCRIBE to ask to be alerted, NOTIFY when event occurs
- MESSAGE for sending text messages (“IM”)
- with forking, can easily register MESSAGE recorder
- true “chat” is voice (+ video)
- services such as reaching mobile phone while in meeting
- types of events:
  - inside existing call leg
  - within call, but outside call leg
  - unrelated to call leg

SIP presence architecture



SIP presence components

Presentity: logical entity being subscribe to, e.g., alice@wonderland.com, with several agents

Registrar: receives REGISTER requests

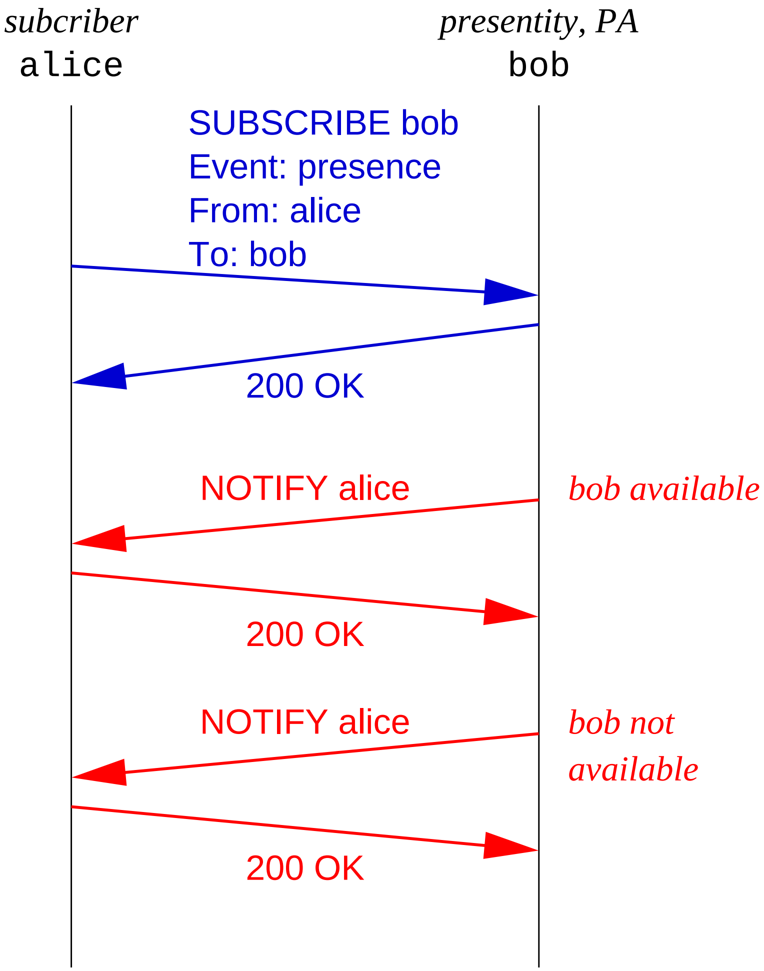
Presence user agent (PUA): generates REGISTER, but no SUBSCRIBE or NOTIFY ➠ any non-presence-aware SIP software

Presence agent: receive SUBSCRIBE, generate NOTIFY

Presence server: SIP proxy + PA

Presence client: SIP UA + PA

SIP presence protocol



SIP SUBSCRIBE example

```
SUBSCRIBE sip:bob@macrosoft.com SIP/2.0
Event: presence
To: sip:bob@macrosoft.com
From: sip:user@example.com
Contact: sip:user@userpc.example.com
Call-ID: knsd08alas9dy@3.4.5.6
CSeq: 1 SUBSCRIBE
Expires: 3600
Content-Length: 0
```

- Forked to all PUAs that have REGISTERed with method SUBSCRIBE.
- 200 (OK) response contains current state.

SIP NOTIFY example

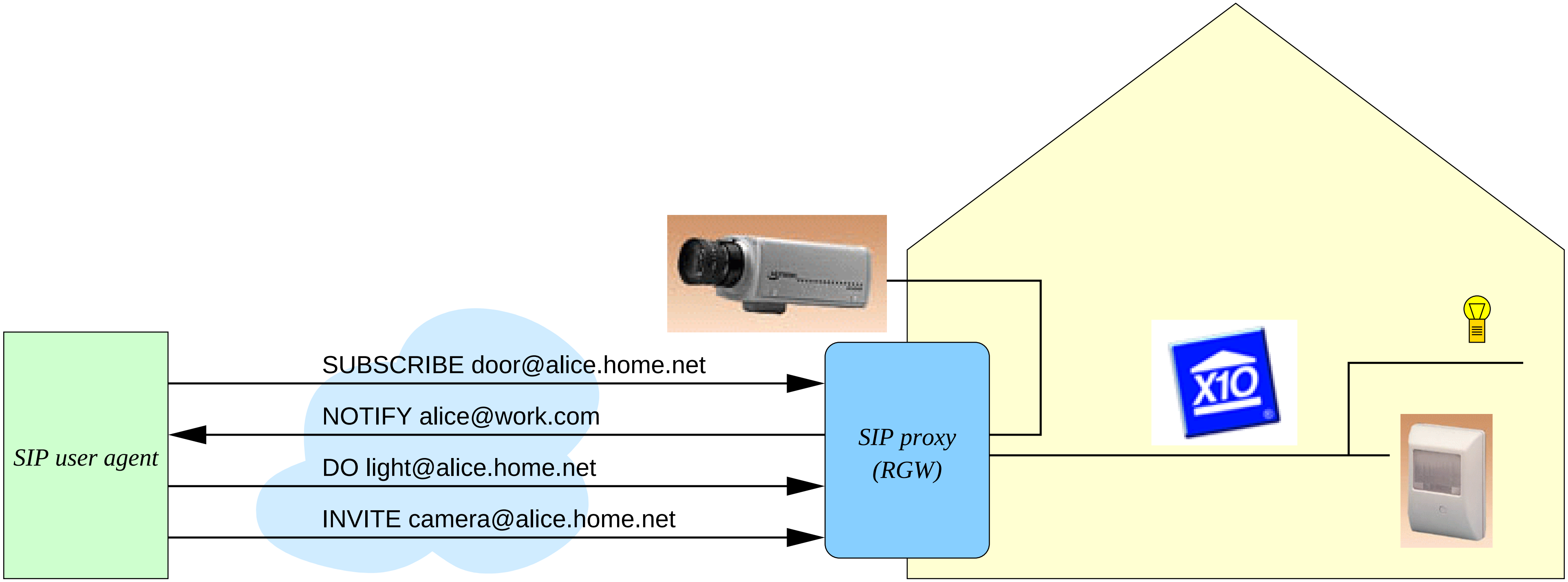
```
NOTIFY sip:user@userpc.example.com
To: sip:user@example.com
From: sip:alice@wonderland.com
Call-ID: knsd08alas9dy@3.4.5.6
CSeq: 1 NOTIFY
Content-Type: application/xpidf+xml

<?xml version="1.0"?>
<!DOCTYPE presence
 PUBLIC "-//IETF//DTD RFCxxxx XPIDF 1.0//EN" "xpidf.dtd">
<presence>
  <presentity uri="sip:alice@wonderland.com;method="SUBSCRIBE">
    <atom id="779js0a98">
      <address uri="sip:alice@wonderland.com;method=INVITE">
       <status status="closed"/>
      </address>
    </atom>
  </presentity>
</presence>
```

Model for control and events

- SIP name high-level entity that's indivisible: lamp, stereo, . . . ➠ can be done differently
- multiple such entities per networked device
- finer-grained control (“variables”) for buttons, switches, and events via XML description

Example home architecture



(Work with Telcordia)

Programmable Internet telephony

	APIs	servlets	sip-cgi	CPL
Language-independent	no	Java only	yes	own
Secure	no	mostly	no, but can be	yes
End user service creation	no	yes	power users	yes
GUI tools w/portability	no	no	no	yes
Call creation	yes	no	no	no
Multimedia	some	yes	yes	yes

Example: integration with iCal ⟶ automatically export personal calendar to call handling

CPL textual representation

```
  <incoming>
    <address-switch field="origin" subfield="host">
      <address subdomain-of="example.com">
        <location url="sip:jones@example.com">
          <proxy>
            <busy> <sub ref="voicemail" /> </busy>
            <noanswer> <sub ref="voicemail" /> </noanswer>
            <failure> <sub ref="voicemail" /> </failure>
          </proxy>
        </location>
      </address>
      <otherwise>
        <sub ref="voicemail" />
      </otherwise>
    </address-switch>
  </incoming>
</cpl>
```

Challenges for programmable services

- integration of authentication information
- handling of SUBSCRIBE, NOTIFY
- integration of JavaScript and CPL?
- modifiable
- model for program generation: flow charts? menus?
- end-system programming: abstracted user interface?

Efforts at Columbia

- integrate into sipc: lamp is just another address-book entry
- SIP + virtual worlds
- enhance call processing language with subscription and notification handling capability

Conclusion

- basic IETF-based architecture in place
- SIP as foundation for services – see http://www.cs.columbia.edu/sip
- new Internet service: synchronous messaging
- common infrastructure for Internet telephony, conferencing, IM, presence and device control ➠ true integrated services