High Performance Multimedia Tools for Application Sharing, Measuring Capture-to-display Latency, and User Created Services
Outline

BASS
Application Sharing System

Performance of Video Chat Applications under Congestion

vDelay
A Tool to Measure Capture-to-Display Latency and frame rate

SECE
Sense Everything, Control Everything
According to the Federal Communication Commission (FCC) report*,
- at mid-year 2010,
  - 63% of reportable connections were slower than 768 kbps in the upstream direction,
  - 18% were at least 768 kb/s in the upstream direction but slower than 1.5 Mb/s, and
  - 19% were at least 1.5 Mb/s in the upstream direction.

Sharing the limited uplink with
- other applications such as BitTorrent
  - running on the same computer or
  - on the other users’ computers (sharing the same connection)
- makes things a lot harder.

BASS Application Sharing System

Omer Boyaci, and Henning Schulzrinne

[1] BASS Application Sharing System
   Omer Boyaci, Henning Schulzrinne.
   International Symposium on Multimedia (ISM2008), December, 2008, Berkeley, CA

   Omer Boyaci, Henning Schulzrinne
   International Symposium on Multimedia (ISM2008), Demo paper, December, 2008, Berkeley, CA

[3] Application and Desktop Sharing
   Omer Boyaci, Henning Schulzrinne
   ACM CoNEXT 2007, student workshop, December, 2007, New York, NY
BASS Application Sharing System

- True application sharing (improves privacy and security)
- Supports multiple users
  - Reliable Multicast
  - Participants with different bandwidths
  - Floor Control
- Multimedia Support
  - Flash animations and videos
Performance results for video (3 Mb/s bandwidth)

- VNC-ZRLE: 1 fps
- VNC-Hex: 0.5 fps
- BASS-T: 6 fps
- BASS-J: 9 fps
- RDP: 1 fps
- THINC: 0.6 fps
- Original File: 16 fps

PhD Defense, Omer Boyaci, 2011
Performance of video chat applications under congestion

Omer Boyaci, Andrea Forte and Henning Schulzrinne

[4] Performance of video chat applications under congestion
Omer Boyaci, Andrea Forte, Henning Schulzrinne
International Symposium on Multimedia, short paper, December, 2009, San Diego, CA
Experiment 1. Step 10 sec 10 kbit

L1. Fine-tunable encoder

PhD Defense, Omer Boyaci, 2011
Experiment 2. Step 10 sec 50 kbit

L2. Fast recovery
Experiment 4. Bittorrent

L3. Giving up does not help
Experiment 5. Random Loss

L4. Differentiate losses
vDelay: A tool to measure Capture-to-Display Latency (CDL) and frame rate

Omer Boyaci, Andrea Forte, Salman Abdulbaset and Henning Schulzrinne

Omer Boyaci, Andrea Forte, Salman Abdul Baset, Henning Schulzrinne
International Symposium on Multimedia, December, 2009, San Diego, CA

Omer Boyaci, Andrea Forte, Salman Abdul Baset, Henning Schulzrinne
International Symposium on Multimedia, Demo paper, December, 2009, San Diego, CA
vDelay: A tool to measure Capture-to-Display Latency (CDL) and frame rate

- Measures CDL and FPS of any video chat session
- Useful tool for comparing video chat clients
- Black-box testing
- Does not require access to source code or protocol messages
- Does not require extra hardware (except an external webcam)
- Java – works in all operating systems
vDelay: A tool to measure Capture-to-Display Latency (CDL) and frame rate

Screenshot of the receiver side vDelay application.
FPS, CDL, and FRR statistics are shown at the top of the image.
The barcode received from the caller agent is also visible.
SECE: Sense Everything, Control Everything

Omer Boyaci, Victoria Beltran and Henning Schulzrinne

[7] Bridging communications and the physical world: Sense Everything, Control Everything
   Omer Boyaci, Victoria Beltran, Henning Schulzrinne
   IPTComm'11, August 2011, Chicago, IL

[8] Bridging communications and the physical world: Sense Everything, Control Everything
   Omer Boyaci, Victoria Beltran, Henning Schulzrinne
   IEEE Globecom 2010 Workshop on Ubiquitous Computing and Networks, Dec 10, 2010, Miami, FL

[9] Demonstration of Bridging communications and the physical world: Sense Everything, Control Everything
   Omer Boyaci, Victoria Beltran, Henning Schulzrinne
SECE allows non-technical users to create services that combine
- communication
- calendaring
- location
- devices in the physical world

SECE: event-driven system
- uses high-level event languages
- to trigger action scripts, written in Tcl
Events & actions

**Events**
- Presence updates
- Incoming calls
- Email
- Calendar entries
- Sensor inputs
- Location updates

**Actions**
- Control the delivery of email
- Route phone calls
- Update social network status
- Control actuators such as lights
- Reminders (email, voice call, SMS)
- Interact with Internet services
every sunset {
    homelights on;
}

every week on WE at 6:00 PM{
    email irt_list “Pizza talk at 6:00 PM today.”;
}

if stock.google > 580 {
    sms me "google stock: [stock google]";
}
Event Rules: More Examples

- **Time**
  - Single: on February 16, 2010 at 6:00 PM
  - Recurring: every day at 12:00 until April

- **Location**
  - Tom within 5 miles of me

- **Context**
  - if my office.temperature > 80

- **Communication requests**
  - incoming call
## Related Work

<table>
<thead>
<tr>
<th>Systems</th>
<th>User rules</th>
<th>User actions</th>
<th>Communications</th>
<th>Time</th>
<th>Location</th>
<th>Presence</th>
<th>Sensors</th>
<th>Web services</th>
<th>Actuators</th>
</tr>
</thead>
<tbody>
<tr>
<td>SECE</td>
<td>NL-like rules</td>
<td>Tcl scripts</td>
<td>Call, email, IM</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>CPL</td>
<td>XML tree</td>
<td>Fixed XML actions</td>
<td>Call</td>
<td>✖</td>
<td>✖</td>
<td>✖</td>
<td>✖</td>
<td>✖</td>
<td>✖</td>
</tr>
<tr>
<td>LESS</td>
<td>XML tree</td>
<td>XML actions</td>
<td>Call</td>
<td>✔</td>
<td>✖</td>
<td>✔</td>
<td>✖</td>
<td>✖</td>
<td>✖</td>
</tr>
<tr>
<td>SPL</td>
<td>script</td>
<td>Signaling actions</td>
<td>Call</td>
<td>✖</td>
<td>✖</td>
<td>✖</td>
<td>✖</td>
<td>✖</td>
<td>✖</td>
</tr>
<tr>
<td>VisuCom</td>
<td>Graphical UI</td>
<td>Signaling actions</td>
<td>Call</td>
<td>✖</td>
<td>✖</td>
<td>✖</td>
<td>✖</td>
<td>✖</td>
<td>✖</td>
</tr>
<tr>
<td>DiaSpec</td>
<td>Java</td>
<td>Java</td>
<td></td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>CybreMinder</td>
<td>Form based</td>
<td>Reminder</td>
<td></td>
<td>✖</td>
<td>✔</td>
<td>✔</td>
<td>✖</td>
<td>✔</td>
<td>✖</td>
</tr>
<tr>
<td>Task.fm</td>
<td>Time rule</td>
<td>Reminder</td>
<td></td>
<td>✖</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✖</td>
</tr>
</tbody>
</table>

PhD Defense, Omer Boyaci, 2011
The big picture
Software architecture
User information registry

- root
  - me
    - office
      - temperature
    - location
      - lights
    - presence
    - phone
      - mobile
        - office
        - home
  - Bob
    - presence
    - location
  - Alice
    - presence
    - location
Every day at 12:00 from 01/01/2010 until 04/01/2010 {
    email employees “lunch time” “Location: 5\textsuperscript{th} floor Dinning Room, Time: 12:30”
}

BEGIN:VCALENDAR
BEGIN:VEVENT
DTSTART;TZID=America/New_York:20100101T120000
RRULE:FREQ=DAILY;BYHOUR=12;
UNTIL=20100401T120000
END:VEVENT
END:VCALENDAR
**SECE: Location-based rules**

```
user operator location { body }
```

bob near "Columbia University"
me near 40.807,-73.963

tom within 5 miles of me
me within 3 miles of "2960 Broadway, New York, 10027"

tom in “Rockefeller center”
Me outside of “Manhattan”

bob moved 1.5 miles

- Place types and user-defined locations:

me near a post office
Anne in a museum
me near my tennis club
Name: Türkiye
Type: Popular
Floor: 12
Apt.: n

My home country ;)
Handling location updates

- **User**
  - publishes his/her location periodically (e.g., every 5 min) to a presence server or to a location service such as Google Latitude

- **Presence server**
  - notifies changes in location to SECE server

- **Google Latitude (or similar service)**
  - SECE retrieves user’s location periodically

- **SECE server**
  - depending on user’s defined rules, queries LoST server

- **LoST server**
  - replies with current information on user’s surroundings

- **SECE server**
  - Takes action based on rules and contextual location information
Event: call, im, sms*, voicemail*, email (*only incoming)

incoming call {
  if {[my activity] == "on-the-phone"} forward sip:bob@example.com
}
outgoing call {
  if {[outgoing destination] == "18003456789"} modify_call destination 12129397054
}
incoming call from Anne {
  if {[my location] != "office"} auto_answer audio no_office.au –record
}
incoming im {
  sms me [incoming from] + " sent an im:" + [incoming content]
}
## SECE: Social Network Integration

<table>
<thead>
<tr>
<th>Incoming</th>
<th>social_network</th>
<th>message_type</th>
</tr>
</thead>
<tbody>
<tr>
<td>facebook</td>
<td>wallmessage</td>
<td></td>
</tr>
<tr>
<td>twitter</td>
<td>newsmessage</td>
<td></td>
</tr>
<tr>
<td>linkedin</td>
<td>direct</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>social_network</th>
<th>status_update</th>
</tr>
</thead>
<tbody>
<tr>
<td>facebook</td>
<td></td>
</tr>
<tr>
<td>twitter</td>
<td></td>
</tr>
<tr>
<td>linkedin</td>
<td></td>
</tr>
</tbody>
</table>
# SECE Events and Actions

<table>
<thead>
<tr>
<th>Context</th>
<th>Event</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facebook</td>
<td>incoming wallmessage</td>
<td>facebook</td>
</tr>
<tr>
<td></td>
<td>incoming newmessage</td>
<td></td>
</tr>
<tr>
<td></td>
<td>incoming direct</td>
<td></td>
</tr>
<tr>
<td>Twitter</td>
<td>incoming twitter direct</td>
<td>tweet</td>
</tr>
<tr>
<td></td>
<td>incoming twitter wallmessage</td>
<td></td>
</tr>
<tr>
<td>Phone calls</td>
<td>incoming call</td>
<td>call</td>
</tr>
<tr>
<td></td>
<td>incoming voicemail</td>
<td>calltts</td>
</tr>
<tr>
<td></td>
<td>missed call</td>
<td>accept</td>
</tr>
<tr>
<td></td>
<td>outgoing call</td>
<td>reject</td>
</tr>
<tr>
<td></td>
<td></td>
<td>forward</td>
</tr>
<tr>
<td>SMS</td>
<td>incoming SMS</td>
<td>sms</td>
</tr>
<tr>
<td>IM</td>
<td>incoming im</td>
<td>im</td>
</tr>
<tr>
<td></td>
<td>outgoing im</td>
<td></td>
</tr>
<tr>
<td>Email</td>
<td>incoming email</td>
<td>email</td>
</tr>
<tr>
<td>Presence</td>
<td>if Bob is available</td>
<td>presence</td>
</tr>
<tr>
<td>Calendar</td>
<td>when [time] before [meeting]</td>
<td>schedule</td>
</tr>
<tr>
<td></td>
<td>when [meeting] begins</td>
<td></td>
</tr>
<tr>
<td>Flickr</td>
<td></td>
<td>flickr</td>
</tr>
<tr>
<td>Translate</td>
<td></td>
<td>to_en, to_tr,...</td>
</tr>
<tr>
<td>Location</td>
<td>near [landmark]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>within [dist] of [landmark]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>in [landmark]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>outside of [landmark]</td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>on [time]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>every [time]</td>
<td></td>
</tr>
<tr>
<td>Contextual</td>
<td>if [variable] [operator]</td>
<td>status [variable] [value]</td>
</tr>
<tr>
<td>Sensors</td>
<td>if office.motion equals true</td>
<td></td>
</tr>
<tr>
<td></td>
<td>if office.temperature &gt; 250</td>
<td></td>
</tr>
<tr>
<td>Actuators</td>
<td></td>
<td>status office.light true</td>
</tr>
</tbody>
</table>
Sensors: smoke, light, humidity, motion, temperature and RFID readers

Actuators: networked devices and actuators such as lights, cameras, sprinklers, heaters, and air conditioners

```python
if my office.temperature > 80 {
    ac on;
}
if my office.smoke equals true {
    sprinklers on;
    sms me "fire in the office";
    call_tts fire-department "fire in the "+[get me.office.address];
    electrical-appliances off;
}
if my warehouse.motion equals true {
    sms me "person in the warehouse."
}
```
OAuth

Application sends request to https://api.twitter.com/oauth/authorize

Is the user logged into twitter.com?

Yes

Sign In Clicked

twitter.com prompts the user to allow or deny the application access

Has the user authorized the application already?

No

Allow Clicked

twitter.com redirects the user back to the application callback with the oauth_token and oauth_verifier

Application completes the OAuth process

No

Sign In Clicked

twitter.com prompts the user to login

Cancel Clicked

twitter.com confirms the user cancelled the sign in or denied access

Deny Clicked

twitter.com confirms the user denied the application access to interact with the account!
GUI (Google Web Toolkit - GWT)

SECE: Sense Everything, Control Everything

Server

GWT Server

SECE Server

DB

Server

- Rules
- Configuration
- Sign out

Rules Editor

- every day at 19:00
- incoming voicemail from 845-694-7280
- me near "Columbia University"
- if me.stock.google > 30
- if temp.ankara > 40
- incoming twitter direct
- incoming twitter wallmessage
- incoming call
- incoming facebook wallmessage
- incoming facebook direct
- incoming facebook newsmessage

Add New Rule
## GUI - Example Rules

**SECE: Sense Everything, Control Everything**

### Example Rules

<table>
<thead>
<tr>
<th>Rule Header</th>
<th>Rule Body</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>every day at</strong></td>
<td><code>for {set x 0} status after status after</code></td>
</tr>
<tr>
<td><code>flickr &quot;Eveni in room 345.&quot;</code></td>
<td><code>sms [status me 345.:]</code></td>
</tr>
<tr>
<td><code>if x &gt; 0</code></td>
<td><code>if me.stock.google &gt; 260</code></td>
</tr>
<tr>
<td><code>if me.weather</code></td>
<td><code>if me.sensors</code></td>
</tr>
</tbody>
</table>

**Command hints will appear here.**
GUI- Action command assistant

SECE: Sense Everything, Control Everything

Rules Configuration Sign out

Example Rules

Time - Single

Rule Header

every day at 4:15 pm

Rule Body

if statement - for selections
for statement - definitive repetitions
while statement - indefinite repetitions
sms - sends an SMS
email - sends an email
call - calls a phone number
tweet - posts a tweet
accept - accepts an incoming call
reject - rejects an incoming calls
incoming - info about an incoming call, etc

Retrieves information about incoming
call/sms/im/voicemail/email. Field names are
origin
destination
content

Save Cancel

Executing Tcl Code to detect problems.
GUI - Registration of third-party services

SECE: Sense Everything, Control Everything

- Rules
- Configuration
- Sign out

user: omer

Registry
Log
Accounts

Please wait. To retry opening twitter, click the button again.
- Register my Twitter account
- Register my Flickr account
- Register my Latitude account
- Register my Facebook account
- Register my SER account
Deployment Scenarios

• Can be deployed in a home device
  • protects privacy and security by keeping the rules and details of sensors and actuators within home boundaries.
  • it is more difficult to update the rules from anywhere.

• Can be provided as a cloud service
  • controlling in-home devices can be challenging, given NATs.
Conclusion

Real world
(location & sensors)

SECE

Web services
(SNs, calendar, contacts, ..)

Communication
(VoIP, SMS, email)
Peer-reviewed conference publications

[1] Bridging communications and the physical world: Sense Everything, Control Everything
   Omer Boyaci, Victoria Beltran, Henning Schulzrinne
   IPTComm'11, August 2011, Chicago, IL

[2] Bridging communications and the physical world: Sense Everything, Control Everything
   Omer Boyaci, Victoria Beltran, Henning Schulzrinne
   IEEE Globecom 2010 Workshop on Ubiquitous Computing and Networks, Dec 10, 2010, Miami, FL

[3] Demonstration of Bridging communications and the physical world: Sense Everything, Control Everything
   Omer Boyaci, Victoria Beltran, Henning Schulzrinne
   IPTComm'10, Demo session, August 2, 2010, Munich, Germany

[4] Performance of video chat applications under congestion
   Omer Boyaci, Andrea Forte, Henning Schulzrinne
   International Symposium on Multimedia, short paper, December, 2009, San Diego, CA

   Omer Boyaci, Andrea Forte, Salman Abdul Baset, Henning Schulzrinne
   International Symposium on Multimedia, December, 2009, San Diego, CA (Acceptance rate: 19%)

   Omer Boyaci, Andrea Forte, Salman Abdul Baset, Henning Schulzrinne
   International Symposium on Multimedia, Demo paper, December, 2009, San Diego, CA

[7] BASS Application Sharing System
   Omer Boyaci, Henning Schulzrinne
   International Symposium on Multimedia (ISM2008), December, 2008, Berkeley, CA (Acceptance rate: 24%)

   Omer Boyaci, Henning Schulzrinne
   International Symposium on Multimedia (ISM2008), Demo paper, December, 2008, Berkeley, CA

[9] Application and Desktop Sharing
   Omer Boyaci, Henning Schulzrinne
   ACM CoNEXT 2007, student workshop, December, 2007, New York, NY
Automated Call Handling

- **Control**: Accept, reject, redirect, forward calls based on variety of SECE signals
- **Integration**: Calendar, address book, PSTN, Google Voice, SMS, location, Text-to-speech, voicemail)
- **Simplicity**: Natural, easy to learn scripting language
- **Flexibility**: Input from a variety of SECE components involved in call handling
- **Automation**: Scripts for recurring tasks (setup a conf. call based on calendar)

“On mom's birthday, call mom when I am home and near phone.”
“Setup a conference call, enter password, invite people, ring desk phone.”
“If driving and incoming call, play “user driving” and redirect to voicemail.”
“If desk phone ringing and not in room, send SMS with caller's number.”

PhD Defense, Omer Boyaci, 2011
public boolean executeCode(Service service, String code) {
    //Creates a new Tcl interpreter
    Interp interp = new Interp();
    try {
        //Add new actions commands to the Tcl interpreter
        interp.createCommand("email", new EmailCmd(emailEventProducer));
        interp.createCommand("status", new StatusCmd(this));
        interp.createCommand("tweet", new TweetCmd(this));
        interp.createCommand("flickr", new FlickrCmd(this));
        interp.createCommand("facebook", new FacebookCmd(this));
        interp.createCommand("sms", new SMSCmd(googleVoice, service));
        interp.createCommand("im", new ImCmd(this));
        interp.createCommand("call", new CallCmd(null, googleVoice, service));
        interp.createCommand("incoming", new IncomingCmd(service));
        interp.createCommand("my", new MyCmd(this));
        interp.createCommand("accept", new AcceptCmd(service));
        interp.createCommand("reject", new RejectCmd(service));
        interp.createCommand("event", new EventCmd(service));
        interp.createCommand("schedule", new ScheduleCmd(this.googleCalendarHandler));
        TranslatorCmd st = new TranslatorCmd();
        for (final Language language : Language.values()) {
            interp.createCommand("to_"+language.toString(), st);
        }
        // runs the user's Tcl script
        interp.eval(code);
    } catch (Exception ex) {
        return false;
    } finally {
        interp.dispose();
    }
    return true;
}
Adding a new action command to the SECE

```java
package edu.columbia.lucs.tcl;
import com.restfb.DefaultFacebookClient;
import com.restfb.FacebookClient;
import com.restfb.Parameter;
import com.restfb.exception.FacebookException;
import com.restfb.types.FacebookType;
import edu.columbia.lucs.Manager;
import java.util.logging.Level;
import java.util.logging.Logger;
import tcl.lang.*;

public class FacebookCmd implements Command {
    Manager man;

    public FacebookCmd(Manager man) {
        this.man = man;
    }

    public void cmdProc(
        Interp interp, // Current interpreter.
        TclObject objv[]) // Arguments to "lsearch" command.
            throws TclException
    {
        String token = man.reg.getRegistryAttribute("me.conf.facebook.acc1.token");
        if (token != null) {
            FacebookClient facebookClient = new DefaultFacebookClient(t);
            try {
                FacebookType publishMessageResponse =
                    facebookClient.publish("me/feed", FacebookType.class, Parameter.with("message", objv[1].toString()));
            } catch (FacebookException ex) {
                Logger.getLogger(FacebookCmd.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
    }
}
```