Architecture and Implementation of a Packet Telephony System

Cormac J. Sreenan

AT&T Labs Research
Florham Park, NJ
cjs@research.att.com
http://www.research.att.com/~cjs

Collaboration with AT&T Labs colleagues:

System Architecture

Nikos Anerousis, R. Gopalakrishnan, Chuck Kalmanek, Alan Kaplan, Bill Marshall, Partho Mishra, Peter Onufryk, K.K.Ramakrishnan

System Implementation

Nikos Anerousis, Mike Chan, Chuck Cranor, R. Gopalakrishnan, Peter Onufryk, Larry Ruedisueli, Eric Wagner

http://www.research.att.com/~cjs/tops-project
Packet Telephony: Opportunities

Intelligence at both the end-points and in the network

- Opportunity to re-partition functionality
- Flexibility to support new services and capabilities (e.g. multiple media)

Desirable to be able to reach users by *name*

- Users can be mobile, connecting to the network anywhere

Network layer connectivity is already provided

- QOS provided by the network layer
- Application specific signaling supports telephony features (conferencing, teleporting, etc.)
TOPS Philosophy

A System Architecture for Packet Telephony.

Addresses key components for a high quality packet telephony service

- User Directory
- Application Layer Signaling (ALS)
- Interworking with the PSTN
- Terminals and Gateways

Operates on a variety of packet network technologies (e.g. IP, ATM)
TOPS Architecture

Call Flow:

- User/Terminal Registration
- Directory Query (returns a call handling profile)
- ALS establishes communication

PSTN gateway for reaching PSTN subscribers

Terminal Tracking Server for mobile terminals
Directory Service

Dial-by-name capability (using a “distinguishing name”)

User-controlled query handling profiles

- user defines who can reach him, when, where, and how
- user may have multiple “personas”
- rule-based filtering: each rule is associated with a call appearance set

Call handling profiles

- A call appearance set modified by the directory service
- can be overridden by the caller (e.g. to block expensive call appearances)

Support for traditional telephony features (call forwarding, etc.)
### Directory Structure

#### Directory Server

<table>
<thead>
<tr>
<th>Type</th>
<th>Identifier</th>
<th>NLA</th>
<th>TLA</th>
<th>Terminal Capabilities</th>
<th>Application Info</th>
<th>Priority</th>
<th>Timeout</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Terminal</td>
<td>ESN1</td>
<td>NLA1</td>
<td>TLA1</td>
<td>audio G.711</td>
<td>none</td>
<td>1</td>
<td>30 sec</td>
<td>phone at work</td>
</tr>
<tr>
<td>Terminal</td>
<td>ESN2</td>
<td>NLA2</td>
<td>TLA2</td>
<td>audio G.711 video H.261</td>
<td>none</td>
<td>2</td>
<td>20 sec</td>
<td>phone at home</td>
</tr>
<tr>
<td>TTS</td>
<td>TTS_ID</td>
<td>NLA3</td>
<td>TLA3</td>
<td>audio</td>
<td>ESN3</td>
<td>2</td>
<td>30 sec</td>
<td>mobile phone</td>
</tr>
<tr>
<td>DN</td>
<td><a href="mailto:my_secy@att.com">my_secy@att.com</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td>my secretary</td>
</tr>
<tr>
<td>PSTN</td>
<td></td>
<td></td>
<td></td>
<td>audio G.711</td>
<td>(973) -555-1212</td>
<td>4</td>
<td>15 sec</td>
<td>POTS phone at my parents</td>
</tr>
<tr>
<td>Server</td>
<td>VM_ID</td>
<td>NLA4</td>
<td>TLA4</td>
<td>audio</td>
<td>VM box #</td>
<td>5</td>
<td>10 sec</td>
<td>Voice Mail</td>
</tr>
</tbody>
</table>
Terminal Mobility

By hiding location management inside the directory service, user and terminal mobility are transparent to caller.

Directory services are optimized for reading.

Infrequent location updates are handled by the directory through a user record update protocol (user or terminal-triggered).

Fast moving mobiles register the address of a terminal tracking server (TTS) with the directory service.
**Application Layer Signaling**

**Goals**

- complements network-layer connectivity and resource management mechanisms
- aims to minimize connections and message exchanges

**Session and media control between end-points**

- call establishment, capability negotiation
- media stream control
- point-to-point and conference calls
- call redirection
Call Establishment

- Invite
- Response (ack, nack)
- Ring
- Alert
- Pickup
- Data exchange
- Hangup
- Acquire network resources
- Alert user of incoming call
Interworking with the PSTN

Telephony gateways bridge signaling and the voice stream

Directory support

- PSTN numbers are mapped into the address of a PSTN gateway by the directory
- TOPS users are assigned an E.164 number to be reachable from the PSTN
- Calls originating from the PSTN are routed in the TOPS domain using a regular directory lookup
Implementation

- TOPS Terminals (Linux PC phones, Euphony ATM phones)
- Directory (CORBA-based, supports IP and ATM end-points, user control over the web)
Euphony ATM Telephone (EAT)

Features:

- ATM-25 Network Connection
- RS232 (PC style DB9)
- Traditional telephone interface
- Case speaker and microphone
- External audio
  - Line output
  - Line / microphone input
- Two extra push buttons
- Three green status LEDs
- One red status LED
EAT System Logic Board
EAT Software Architecture

EAT Phone Application and Services

VxWorks 5.3.1

Euphony Chip/Board Support Package

EAT Hardware
Packet Telephony Application
**Design Principles**

**Low cost (less than $100)**
- A consumer device (a phone you can put in your bedroom)

**Extensibility**
- Packet telephony is in its infancy
- Too many (i.e., lacking) standards
- Research areas still exist
  - QoS, security, privacy, authentication, and billing

**Ease of Use**
- Designed for ordinary people with no technical background
- Ordinary people unwilling to invest time to set-up, configure, and maintain complex devices
- Must be able to support advanced services

**Reliability**
- Always on and always works
Why Not a PC based Packet Phone?

Too Expensive ($500 too much for a phone)

Too Complex

- To use
- To install and configure
- To administer

Too Unsightly

- Big (keyboard + mouse + monitor + case)
- Loud (power supply fan, CPU fan, and disk)

Too Unreliable

- Never on when you need it
- Crashes often
Why Not Include More “Stuff” with the Phone?

Things we could have added
- LCD display (touch screen), more keys, keyboard, mouse, ....

Adding “stuff” is bad
- Makes phone expensive
- Makes phone complex
- Would anyone really use it?

How many ISDN phone features do you use?

Our approach for advanced features is to network devices
- Phone runs HTTP server
- Advanced features available by using thin client (i.e., browser)
- You pay only for those features you want

Display on TV
LCD display (size, bw/color, touch, etc...)
Why an RTOS?

Why do you need an OS at all?

❑ Packet telephony application quite complex
❑ Need tasks and interrupt handlers
❑ Need standard libraries

Why not Linux?

❑ Timesharing environment brings too much useless baggage
disk file systems, VM, multiple address spaces, multiuser, accounting
❑ Not scalable, Not real-time
❑ Poor embedded development tools

Why VxWorks (could be any RTOS)

❑ Small and efficient
❑ Real-time kernel (256 priorities, preemptive, deterministic)
❑ Scalable (minimum kernel ~64K-bytes)
❑ Ported software (HTTP server, SNMP, Java, etc....)
❑ Excellent cross development tools
Event Exchange (EVX)

EVX delivers events posted on a “sending” port to one or more interested modules on their “receive” ports

- Events are named independently of module function names or addresses
- Events can be delivered to one or more receivers
- Sender know about receivers
- Event data is delivered through zero copy reference counts
- Events are queued at sending port to provide flow control

EVX application consists of three parts

- Application modules
- Initialization code that configures EVX connections
- EVX library + EVX thread
Future Work

EVX

- Explore using “smart modules” to support dynamic configuration of new services

Signalling

- Explore supporting multiple calls which use different signalling protocols (generic signalling interface)
- Transition from TOPS to SIP + DOSA

Advanced Services

- What features should be in the network and what features should be implemented by “intelligent” devices?
- High quality music end-point
- Voice enabled user interface

Packet Telephony Issues

- QoS, security, privacy, authentication, and billing
**EAT Deployment**

**Currently**
- Half a dozen EATs in various labs
- One line PSTN gateway (only outgoing calls)
- Custom directory (CORBA and C++ based)

**This year**
- Deploy EATs in offices of people working on the project
- Construct a true PSTN gateway (T1 to PBX)
- LDAP based directory
Related Work

Directory and Call Establishment

- IETF SIP (integrated user location function and connection establishment)
- ITU H.323 (intermediate entity for signaling (Gatekeeper), protocol overhead for simple telephony)
- Voice-over-IP Call Management Agent (no user or terminal mobility)

Telephony appliances

- Off-the-shelf (e.g. Selsius)
- Fixed choice of coding and signaling (H.323)
- Services via LAN-based PBX PC
**Conclusions**

TOPS leverages the combination of packet networking and intelligent end-systems

- New telephony services
- Directory service supports user and terminal mobility
- Application layer signaling protocol
- PSTN Interworking
- Support for fast-moving terminals

**EAT packet telephone appliance**

- Low-cost, simple telephony device
- Extensible software event mechanism
- Easy to use