Call Processing Languages for Internet Telephony

Jonathan Lennox
Columbia University, New York
lennox@cs.columbia.edu

IETF IPtel Working Group
April 1, 1998

Motivation

- IP Telephony moves intelligence to end systems — we need to have some way of putting it there!
- But end systems aren't always turned on, so we need some standardized way to tell proxies what to do

Location

- Script execution either in an end-system or a proxy
  - This has implications for what functionality is available in each case
- Sent to server as part of location-registration
  - A registration is a trivial CPL script
- Hierarchy of execution
  - Successive proxies can execute successive CPLs (inbound or outbound)
  - This model can be generalized to allow multiple CPLs in the same proxy (administrative vs. user control)

Design principles

- Simple
  - Safe implementations (vs. security leaks, DoS)
  - Automatic verification of correctness
  - Ease of implementation
  - Inspiration: Sieve
    - No functions
    - No looping
- System-independent
  - The same script (e.g. for behavior on busy) should work in a proxy or end system, to the extent possible
- Protocol-independent
  - At least including SIP and H.323 — CPL is high-level

Features

Three categories of necessary features:

- Events
- Queries
- Actions

Events

- Basic invitations
- Other invitation-like requests
- Departure
- Call responses and timeouts
- In-call changes
- Third-party call control
- “In-band” signaling

Queries

Call state	busy, n calls queued
Scheduling	time of day, user's calendar
Location information	find the end system
Database lookup	anything…

Actions

User notification	alert dialog, ringing
User confirmation	especially for potentially dangerous actions
Call proxying	including multiple parallel proxying (search)
Call rejection	including redirection
Logging	of arbitrary data?
External CPL scripts	at remote locations?