Session Initiation Protocol and Media Gateway Control

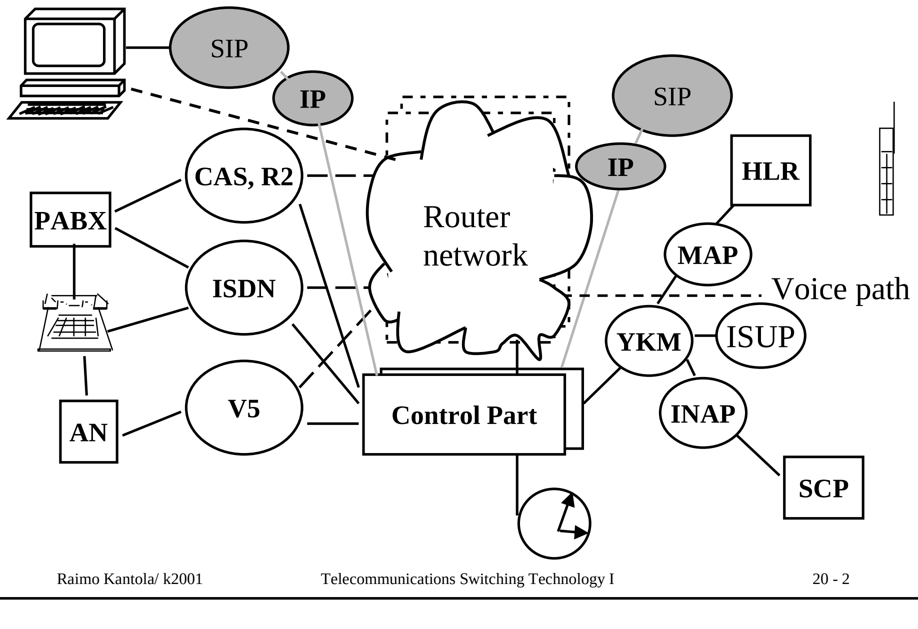
SIP protocol and its extensions

SIP Service Architecture

SIP in 3G

Megaco/MGCP

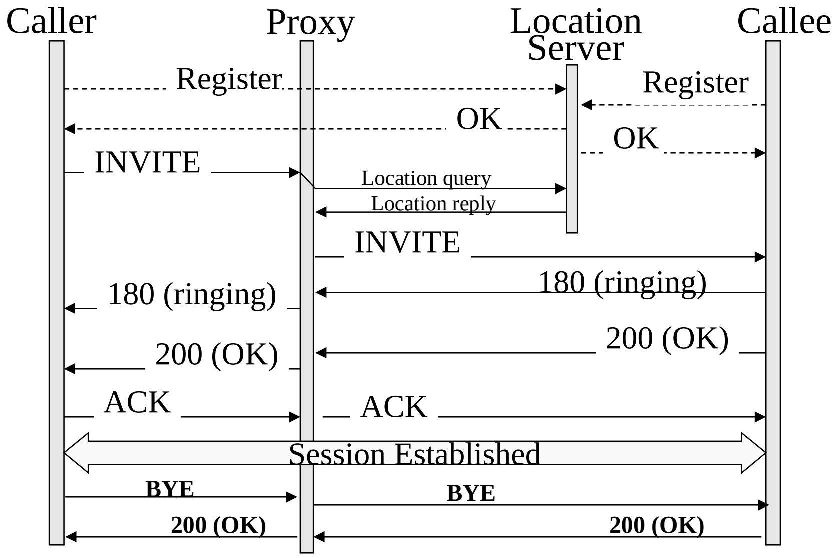
Course Scope - lecture scope



SIP overview

- Simplicity
  - Ascii based - simple tools for development
  - Lower call setup time than in H.323
  - basic protocol + extensions structure adopted
- Caller preferences
- Ability to support many media types
- Used between both services and call control entities
- Has been adopted as the basis for 3G.IP signalling
- Originally subscriber signaling, proposed also as network to network signaling
- Quality of Specification is not very good! Leaves a lot of decisions to the implementor
- A lot of development during the last year!

Basic SIP call setup and release



Stateful Proxy vs Stateless Proxy

Stateful Proxy

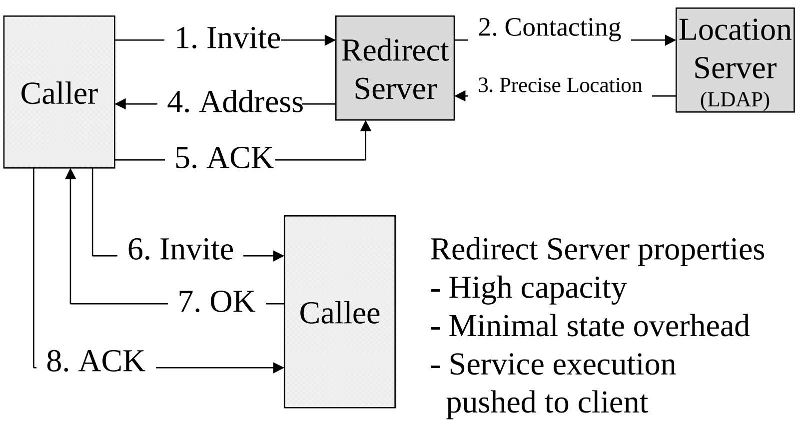
- Maintains call context
- Replicates UAS/UAC to process requests and responses
- Call state and transaction state can be maintained
- Forking proxies require state
- TCP proxies must be stateful for reliability
- Enhanced services require state
- Can collect charging info

Stateless Proxy

- No call context
- Response is not based on UA replication
- Provides client anonymity
- Restricted gateway access
- High processing capacity
- Easier to replicate than the stateful proxy
- Also semi-stateful is possible

UA = User Agent, UAC = UA Client
UAS = UA Server
Forking = multicast of INVITEs to N addresses

Redirect Server pushes processing to clients



Redirect Server properties
- High capacity
- Minimal state overhead
- Service execution pushed to client

SIP issues

- Parties can release the “call session” but since they have obtained each others IP-addresses, they can continue sending media streams to each other!!
- How to push INVITE to B-party, if B-party does not have a permanent IP address which is most often the case!

Integration of Proxy with Firewall and NAT

- Response messages (e.g. 180) are not reliably delivered. This may cause tear down of the call if it was initiated from ISDN

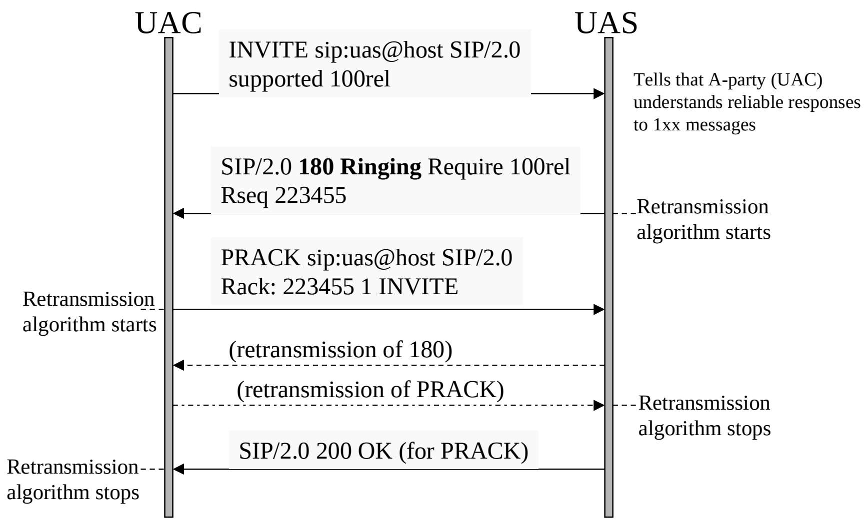
PRACK method

- If BYE is lost, Proxy does not know that call has ended

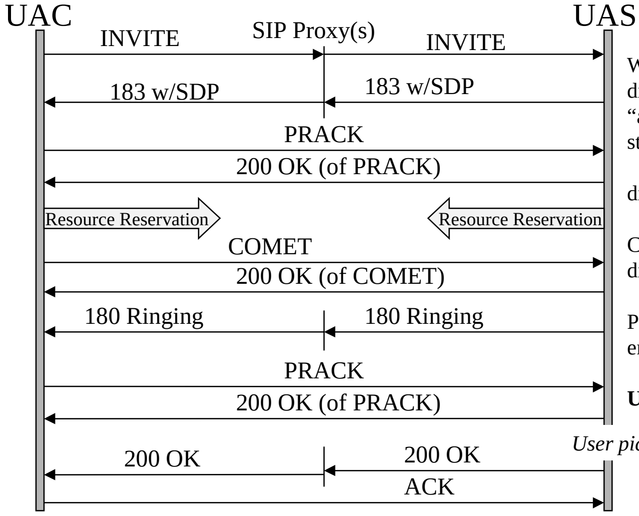
KeepAlive = re-INVITE mechanism

- Ascii coding increases the signaling overhead in Radio access

Reliable Provisional response in SIP



Phone should not ring before QoS and Security are OK



W/SDP = “a-qos:” strength direction
“a-secure:” strength direction
strength = mandatory|optional |success|failure
direction = send|recv|sendrecv

COMET confirms that preconditions are OK at the originator

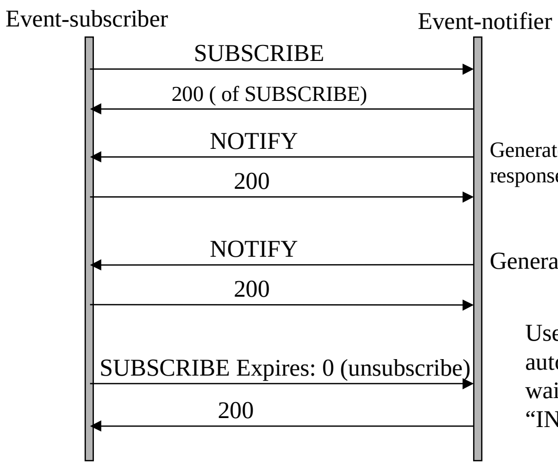
PRACK method is used to ensure delivery of 183 and 180

Use case: 3G signaling!

User picks up the phone

SDP = Session Description Protocol (carried in SIP message body)

SIP event notifications tell about remote significant events to a remote party



Generate immediate state response

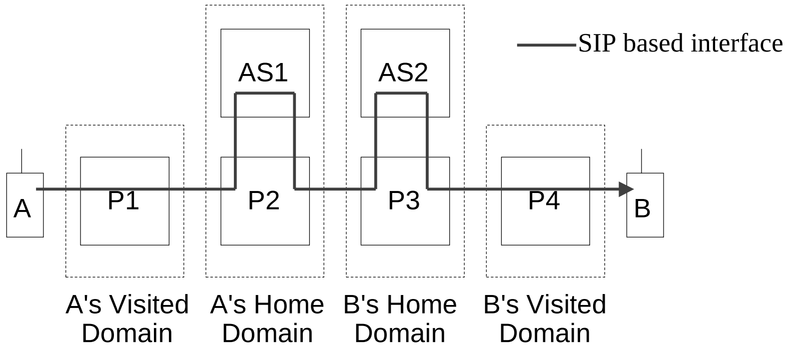
Generate state change event

Use cases: automatic call-back, automatic buddy lists, message waiting indication and “IN triggering”.

Services use many protocols

- New services and more flexible service creation should differentiate IP Communications Network from PSTN
- Services should combine different forms of communication, thus multiple protocols are needed:
  - SIP for media sessions and session related services, subscriptions and notifications?, messaging?
  - HTTP for web and transactions
  - SMTP for e-mail
  - RTSP for media streaming
- The use of these protocols is orchestrated by the service logic: context is set up using SIP.

Routing and Service Model in 3G



P1, P4: Outbound Proxies

P2, P3: Registrar Proxies

AS1, AS2: Application Servers

NB: Also AS based on direct processing of call state: There is no Basic call state model like in IN

SIP Entities & Service Capabilities

- User Agent (= UAC + UAS)
  - Can run services, such as forwarding, filtering etc.
  - Not always connected (out of coverage/battery etc.)
- Redirect Server
  - Can do services that require only Request-URI change, e.g. translation, parameter addition etc.
- Proxy Server
  - Can change certain headers and stay in the signaling path
  - Forking, actions based on responses
- Back-to-Back User Agent (=both ways User Agent)
  - Can e.g. issue requests to a call leg or modify SDP
  - In many cases necessary

Application Server in 3G?

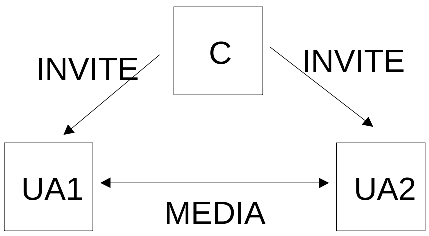
- Fuzzy Definition but has SIP+ interface!
- Can be a Redirect or Proxy Server or Back-to-Back UA
- The key is that it should be *programmable*
  - Routing based on service logic: what to do when user not registered or busy
  - URI translation: Reachability chains
  - Interfaces to other protocols: HTTP, SMTP, RTSP etc.
- Can be single purpose boxes, or multi-purpose boxes, or controllers who orchestrate things

Server types for different services

- Media Server (SIP, RTSP, HTTP)
  - Announcements, IVR, Voicemail, Media on demand
- Conferencing Server (SIP)
  - Media mixer
- Presence Server (SIP)
  - Users status info, capabilities, willingness to communicate
- Web Server (HTTP), E-mail Server (SMTP), Messaging Server (SIP?), Text-to-Speech Server etc.
- Controller Server
  - Co-ordinates the overall service

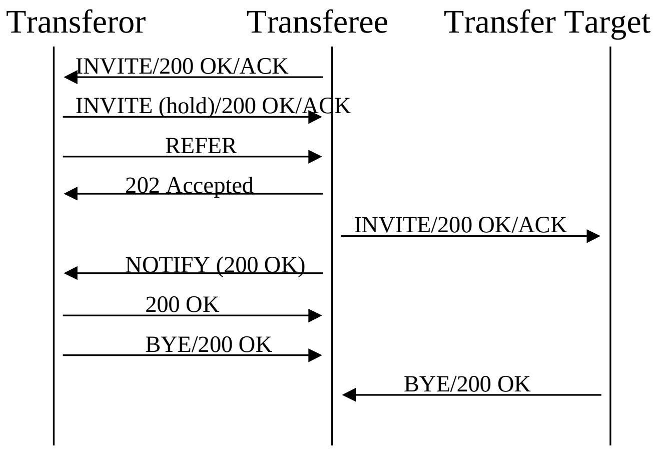
=> Server resources can be addressed by URLs, no need for tight coupling a la MGCP/Megaco

Third Party Call Control is based on SIP



- Details are still to be solved in the IETF
- Powerful tool e.g. for inviting users to centralized conferences or sessions with a Media Server

REFER and Call Transfer



How to Program Services

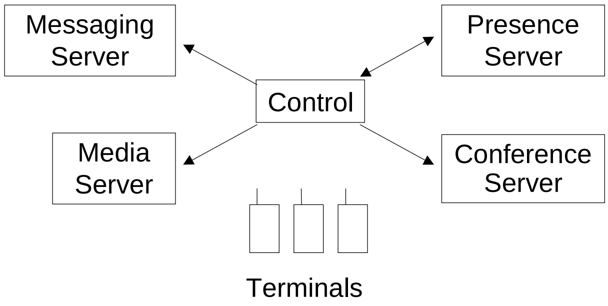
- Call Processing Language
- SIP CGI
- SIP Servlets
- SIP JAIN
- Soft SSF and INAP/CAP
- Parlay
- OSA

==>
There will be many competing ways to implement services!

=> Whatever… Different abstraction levels

The claim is that it should be as open as flexible as creating services in the web these days

Auto-conferencing Service Example



1. One user orders the conference by filling a web form
2. Controller subscribes to each participants presence
3. When all available, send message or start IVR session to each participant to confirm willingness
4. Connect each participant to conference server. Play announcements to conference from media server when new parties join

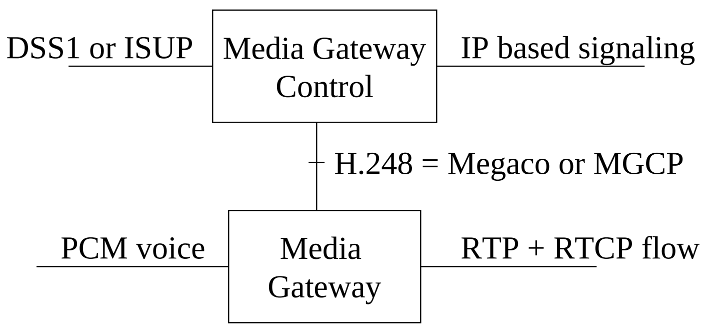
Problems

- How to make "service routing"?
- How to make service components really independent?
- If there is dependency, how to move parameters between the components?
- How to secure call release?

Megaco - Media Gateway Control protocol is the newest entrant

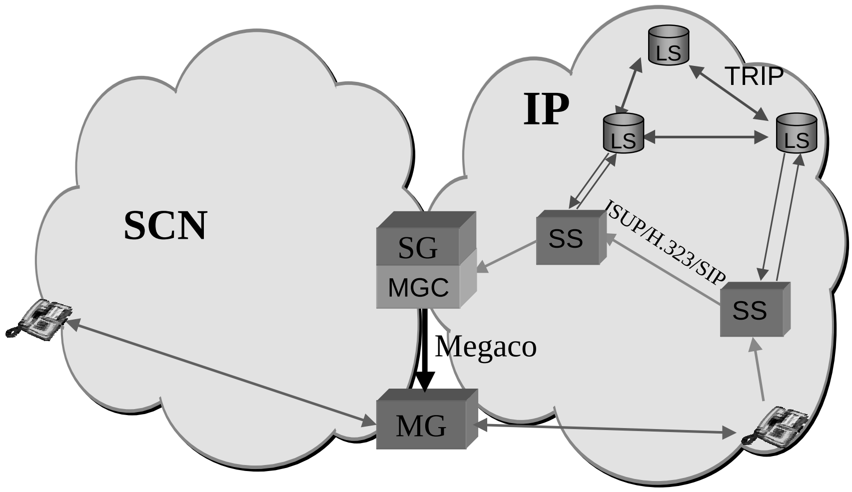
- MGCP was promoted by Cablelabs = US CATV R&D body as the CATV Telephony standard
- ITU-T is making its own variant called Megaco
- Megaco, MGCP are master-slave protocols by which media gateways can be configured e.g to services - in case of residential media gateway, MGCP becomes a subscriber signalling system

Gateway decomposition



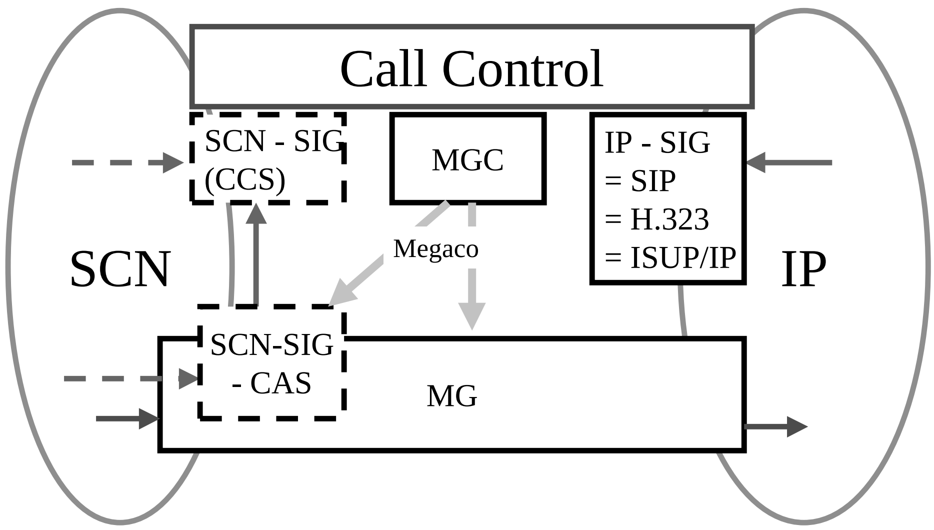
MG - Trunk gateway, residential gateway etc.
Many MGs can be controlled by one MGC, MGCs can be a mated pair --> higher availability performance.

Current Architecture



TRIP = Telephony Routing over IP, SG - Signalling Gateway, MGC - Media Gateway Controller
MG - Media Gateway, SS = Signaling Server, LS = Location Server

Gateway decomposed

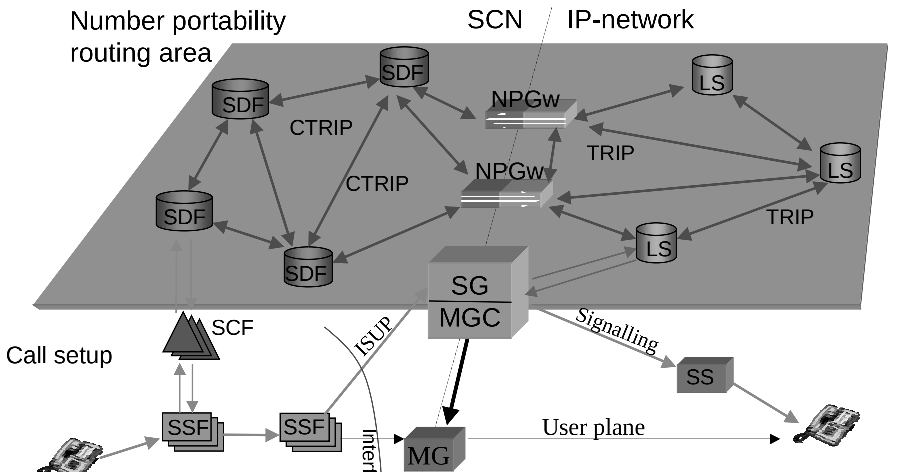


IP Telephony Research (IMELIO) in the Networking Laboratory

- Technology evaluation
  – Delay measurements breakdown
  – SIP call waiting
- Numbering and Routing Information Interoperability with ISDN
  – TRIP and ENUM protocols
  – CTRIP protocol proposed

The solution is CTRIP + Numbering gateway

IMELIO

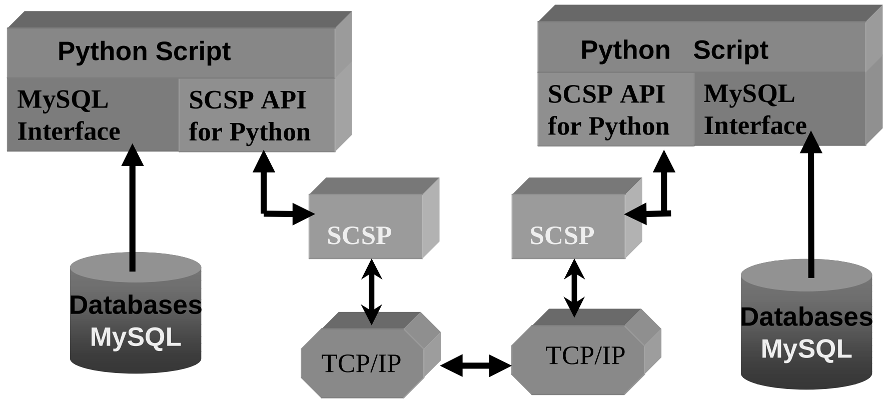


CTRIP = Circuit Telephony Routing Information Protocol, TRIP = Telephony Routing over IP, NPGw = GW for sharing information between CTRIP and TRIP, SS = Signaling Server, LS = Location Server, SDF = Service Data Function, SCF = Service Control Function, SSF = Service Switching Function

TRIP and CTRIP Modules

IMELIO

We use Python as Interface to work with SCSP primitives through Python oects an access to the information C store in the yS ataase



IP Telephony Signaling alternatives

Fully distributed | Intelligence | Master-Slave

SIP
- ascii based
- Adopted by 3G
- Basic + extensions
- Bakeoffs drive vendor interopera-bility

H.323
- Inherits ISDN
- complex
- still few services
- Widely used
- first working solution

Megaco/H.248/MGCP
- newest
- seems to be quality spec.
- facilitates Gateway decomposition
- Interoperability?

SIGTRAN works on ISUP over SCTP over IP
- many (netheads) view ISUP as an interim solution!