

OpenSER Admin Course

Basic VoIP routing



Voice System SRL

<http://www.voice-system.ro>

<http://www.openser.org>

- routing logic is totally different if the it is about an **initial** or **sequential** request as there are different routing mechanism
- **initial requests**
 - routing is based on discovery mechanism in order to locate the destination
 - initial request does not only discovery, but also records the relevant hops on the path from source to destination
- **sequential requests**
 - they are strictly routed based on the information collected by the initial requests
- who do distinguish the type of the requests?
initial request do not have TAG parameter in To header

- identify if the initial request is allowed to be processed on the server (depending on the service type)

Ex:

residential services – is the caller or callee a local subscriber?

- depending if we do caller or callee based processing, apply corresponding routing mechanism like:
 - enum
 - aliases
 - dns
 - user location
 - whatever else transformation

- for the initial requests different routing logics may be applied, depending of the method. For different methods, different services and routing mechanism make sense.

Ex: MESSAGE versus SUBSCRIBE versus INVITE

- not all initial requests open dialogs, so if they not in all cases make sense to record our server to be passed by the sequential requests.

Ex: MESSAGE requests are stand alone.

- if I need to be on the path of sequential requests, record the server address as hop – record_routing()

Why ? :

- accounting purposes
- dialog monitoring
- routing through firewalls
- network changing
- protocol changing
- security reasons

- CANCEL is a special requests that needs special care when comes to routing
- even if it looks like an initial requests, you do not route it as an initial request!
- according to RFC3261, a CANCEL request (for an INVITE request) must be routed exactly as the INVITE :
 - send to same destination
 - with same RURI
 - using same transport protocol
- in OpenSER, we deal with CANCEL in different ways, depending if we do stateful or stateless processing

■ **statefull processing**

- this is the easy case as the transaction module contains all the information about the INVITE, so it will automatically send the CANCEL at the correct way and to the correct destination
- we just need to instruct TM to forward the CANCEL and it will do the job for us

```
.....  
if (method==CANCEL) {  
    if (t_check_trans())  
        t_relay();  
    exit;  
}  
....
```

■ **stateless processing**

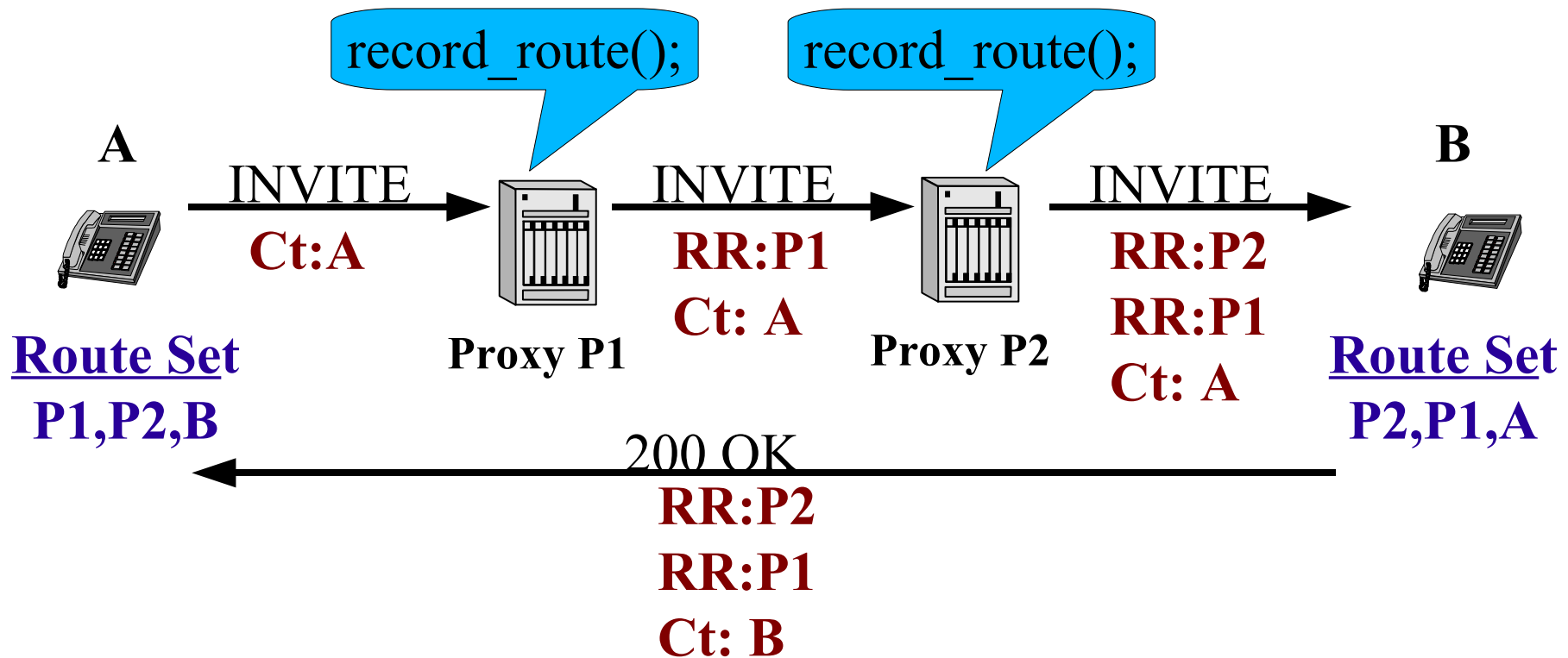
- from script effort, this case is more difficult
- we do not have stored (as in TM) any information about where and how we routed the corresponding INVITE
- we need to apply for CANCEL **exactly** the same routing logic as for the INVITE

if (is_method("INVITE")) {...}

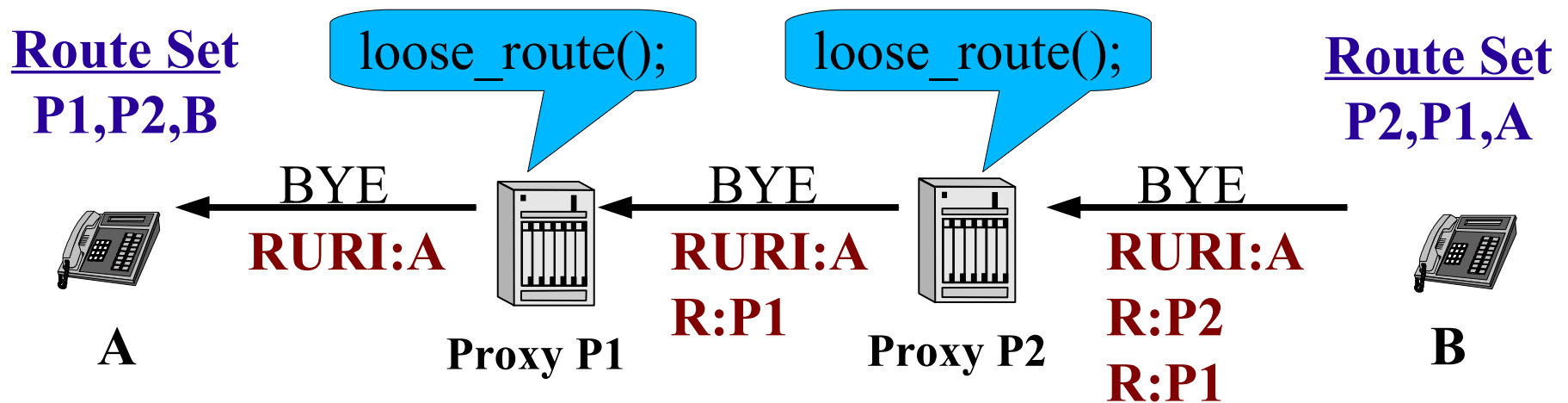
⇒

if (is_method("INVITE|CANCEL")) {...}

- Record routing is a mechanism that allows the initial requests to gather information about the intermediary hops (proxies) that have to be visited by the sequential requests.
- after the completion of the initial requests, both caller and callee will know the path to use for sending the sequential requests – this is also called “**route set**”
- **route set** is composed of
 - the intermediary hops that were record routed
 - the contact address of the other end-point
- the **route set** of the callee will be different from the one of the caller:
 - the contact will be different (each part will have the contact of the other part)
 - the recorded hops will be in revers order (to reflect direction)



- loose routing is a routing mechanism for sequential requests
- it uses the information (**route set**) learned by the initial request



Over view of a basic script example