New signalling mechanisms for multi-provider and cross-network services

Claude Rigault
Département informatique et réseaux,
ENST, 46 rue Barrault, 75 013 Paris, France
GET-Télécom Paris ; LTCI-UMR 5141 CNRS
claude.rigault@enst.fr

Rony Chahine
Département informatique et réseaux,
ENST, 46 rue Barrault, 75 013 Paris, France, GET-Télécom Paris
CoreBridge, 3 rue Saint Philippe du Roule, 75 008 Paris
rony.chahine@enst.fr,

Abstract: **This paper addresses the problem of providing a service through the cooperation of service components executed in real time by different service providers across different networks. Within this wide scope, our contributions focus on three points that require attention before further progress:**
a) Defining a proper business model structuring the various roles in the service offer, this is achieved by the SIMPSON model.
b) Providing efficient association mechanisms between the various cooperating service instances. This is achieved by the CAT mechanism.
c) Proposing a new signalling paradigm based on the previous association mechanisms that would apply to cross-network services.

1. INTRODUCTION: THE PROBLEM OF MULTI-PROVIDER AND CROSS-NETWORK SERVICES

An Intelligent Network (IN) service is an alternate call control function, substituted to the basic call control function of the switches. To allow service design, this call control function has been divided into elementary service features, IN services being new combinations of these features. Service features may be viewed as service components and IN services are designed as a graph of such components [1]. We may generalize this view to a new idea of combining network functions or components with other components to provide a richer service. For example a car manufacturer may require adding inventory management components or financial services components to its Virtual Private Network VPN service in order to get the customized service he needs. In this case, VPN components are provided by a network operator, financial components by a bank and inventory management components by another company. However all these components are integrated into a single customized service, with a single user interface. We call such a service a "multi-provider service". In addition, we may imagine that when an employee receives a call within the company private network, he gets the company file on the calling party on his office PC. If this employee works at home he may want to receive the same information through the internet on his home PC when he is called. The service that was available in the company's private network is now extended across the borders of several networks. We call this extended service a "cross-network" service. Today, signalling paths are missing both for cross network services and for multiple providers inter-working in a single service. Partial solutions do exist: Web Services [2] or other types of Middleware [3] achieve some multi-provider services. These solutions however do not apply to multiple or heterogeneous networks. Cross-network services are considered by PINT [4] for a very limited set of services and in a more general, but very centralized manner by PARLAY [5].

Our approach looks for more cooperative solutions were multiple service platforms, including PARLAY servers would work together. To tackle this problem we will:

a) Propose a structured organization of control functions, called the SIMPSON model, structuring this generalized integration of multi-provider and cross network service components and identifying all the signalling paths for the exchange of control information. A requirement of this organization being of course that it remains compatible with existing IN, Parlay, CAMEL [6] and OSA Architectures [7]. As an illustration, the case of networks based on the soft-switch architecture [8] will be investigated.

b) Propose a mechanism, called the CAT, for establishing, maintaining and releasing the associations between the cooperating entities.

c) Propose a new signalling paradigm that is applicable to all the service levels.

This differs from the present research on signalling like the IETF “NSIS” effort [9] that is dedicated to the transport services. Our research fully complies within the research framework of the ETSI TISPAN research effort [10] and in particular with the TISPAN requirement for a meta-signalling.

2. THE SIMPSON MODEL

We first identify the various functional roles required in a multi-provider and cross-network environment. The SIMPSON model (Signalling Model for Programmable Services Over Networks) [11] of figure 1 gives a clear identification of these roles. It is a five level model that may be viewed as a 4-tier client-server organization.

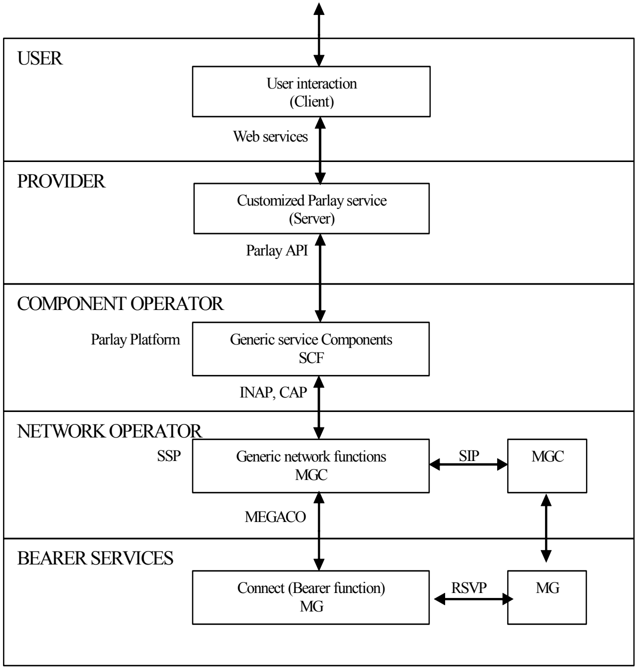

Figure 1 – SIMPSON view of a Parlay service over a soft-switch architecture

A “user” is using some client software, located in the “client” terminal, preferably based on a standard browser, to interface with his multi-provider and cross-network service. The client is mostly dedicated to service profiling, including presentation functions. The service is executed by a “Provider” platform server. This server executes the service logic constructed as a graph of Generic Service Components GSC. It is the first tier in the model. It uses an abstraction of service components that are supplied and executed at the Component operator level. Components servers are the 2d tier of the model. Some components require Network Generic Functions NGF like Call services, Virtual Private Network services, Routing services, IVR services, Multicast services etc. These NGF are performed by the servers of a “Network operator”. Examples of such Network operator platforms may be the Media Gateway Controllers MGC for IP telephony [12]. Network operator platforms constitute the 3d tier of the model.

Finally some of the NGF may require the setup of bearer capabilities with defined QOS parameters. They invoke for this purpose Bearer Functions provided by the equipments of Bearer Services Operators. Bearer Services Operators equipment are the 4th tier of the model. ***We define as a “connection” a bearer service resource reservation such as a bandwidth reservation or a particular scheduling mode selection***. The Bearer Service Level therefore specializes in the setup, modification or release of connections.

The SIMPSON model identifies two different types of interactions between control plane entities. Interacting entities may belong to the same service level and operate in the peer to peer mode. We call “horizontal signalling” this type of horizontal communication. On the contrary, they may appear in adjacent service levels, in which case they operate in a client-server mode. We call “vertical signalling” or APIs (Application Programming Interfaces) this type of vertical communication. On the figure 1, the SIMPSON model shows, in the case of Parlay services over soft-switch architectures, the various vertical and horizontal signalling protocols required in the control plane architecture.

As APIs we find that the User to Provider Interface may be implemented by Web services. An example of the Provider to Component interface is the Parlay API. By this API, a service provider invokes networking components in a Parlay Platform such as an Ericsson Jambala platform [13] belonging to a network service operator. An example of Component to Network Interface could be the intelligent network INAP or CAP set of operation, by which a Parlay platform may invoke the services of a Service Switching Point SSP control unit within a Soft-switch MGC. Network functions of the MGC request Bearer Services from Media Gateways MG by means of the MGCP or MEGACO operations.

As horizontal signalling protocols we find at the network operator level the Session Initiation Protocol SIP signalling [14] or the Bearer Independent Call Control BICC signalling [15]. Finally, we find at the bearer level, all the Circuit Associated Signalling CAS protocols such as the ISDN User Part ISUP signalling protocol widely used between telephone exchanges.

In the legacy PSTN network, switches merge in a single equipment both the network functions (call set up, routing …) and bearer or connection functions. The soft-switch architecture is a very interesting example of a case where the network functions (performed by the MGC) and the bearer functions (performed by the MG) are separated. There are already some instances were both levels are operated by different enterprises: On one hand, IP connectivity operators (at the bearer level) provide customers with MG and take care of the IP forwarding functions. On the other hand, Call Control operators (at the network level) operating MGC outsource the Call Control functions or the IP-Centrex functions formerly performed by PABX.

3. THE REQUIREMENT FOR THE ASSOCIATION OR BINDING OF LOCAL REFERENCES, THE CAT MECHANISM.

At service initiation, memory pages are opened in each participating process to describe their state and their local instance data and memorize them during the whole session.

We call "Local context" the memory page opened by a given participating process. However, the local contexts of each participating process put together are to be considered as a global context for the service session. ***We call "Global Context", the union of all local contexts that give a global view of the session. This union is done by the association mechanism that binds (cross-references) together all the local contexts involved in the same service instance***. By this mechanism, each local context has a pointer to the others. ***Local service instances are associated if they can mutually address each other among multiple other service instances within multitask machines***.

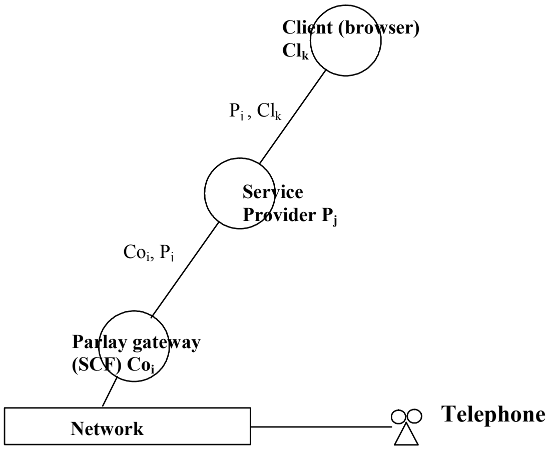

Figure 2: Example of a call initiated by the Parlay Service Provider

Figure 2 gives an example where an instance of a Provider Service is associated with an instance of a Service Control Function SCF in a Parlay Gateway for a cost presentation service.

Each local context is identified by its local reference: the Parlay gateway context is referenced Co_i, the Service Provider context is referenced P_j, and the client context is referenced Cl_k… In each context, couples of local references and remote references are maintained and constitute binding references. When a party hangs up, the Parlay Gateway is notified by the network of the call termination and gets the charging ticket. It finds in its context the binding reference [Co_i, P_j] for the Service Provider and sends him together with this reference the charging value. With the binding reference [Co_i, P_j]**,** the service provider can find the proper context P_j and locate in it the binding reference [P_j,Cl_k,], that identifies the client context Cl_k. In its turn the Service Provider can send together with this binding reference [P_j,Cl_k,] the call cost to the client local context Cl_k. After the termination of the call, all contexts in implicated nodes are freed.

In legacy networks the association is performed by the Transaction Capability TCAP protocol [16].

In our more general example of figure 2 the Parlay Gateway can notify the Service Provider that a party has ended his call by going on hook. In the reverse direction the Service Provider can notify the Parlay Gateway that a party has ended his call on his browser. This is allowed by the binding references. The Parlay Gateway context has a binding reference [Co_i, P_j] to the Service Provider context and vice versa the Service provider context has a binding reference [Co_i, P_j] to Parlay Gateway context.

Such a bi-directional persistent binding does not exist in the pure client server mode of communication of the web services.

Let's take an example where the Parlay Gateway and the Service Provider would communicate by web services (a pure client-server mechanism) rather than by a persistent ORB like CORBA [17] and where the web services are in the Parlay Gateway. Then the Parlay Gateway is in the server position and the Service Provider is in the client position. If a party ends the call on his browser, the Service Provider, warned by the browser, can notify the Parlay Gateway with the web service. But if a party goes on hook, the Parlay Gateway cannot notify the Service Provider as a server does not maintain a reference pointing to the client that has previously addressed him: A Server is not able to push information into the client (Service Provider).

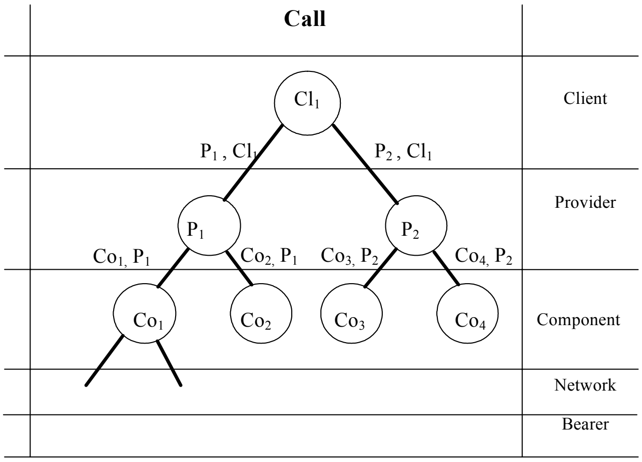

Figure 3: SIMPSON view of a Vertical CAT

To solve this problem the Service Provider should be able to send events to the Parlay Gateway and the Parlay Gateway to push events into the Service Provider. The solution, for an efficient system, is to maintain an association between the two nodes that persists during the full session duration. Such a persistent association is achieved by the concept of binding reference. Peculiar implementations of the binding reference may be the simple TCP socket, TCAP transaction and dialog identifications, CORBA associations.

We present a new scheme for the binding mechanism: Processes involved in a same global service session have to associate their local contexts to build a Global Context for sharing instance data. In the same manner as a File Allocation Table FAT [18] links together sectors of a same disk to build a file, ***a Control Allocation Table CAT links together the various local contexts in different platforms to build a global context.***

When a process needs to share information (instance data) with a partner process, it gets the binding reference of its peer context from the CAT. The CAT is a binding reference graph, distributed on all the associated contexts (see Figure 3). The CAT is a data structure persisting during the whole session duration. It is created as the contexts are opened up and is erased at session termination.

4. A NEW PARADIGM FOR SIGNALLING IN A GLOBAL CONTROL PLANE

Our concept of CAT allows a new definition of signalling. Signalling is generally understood as the invocation of remote operations or exchange of notifications between local processes of a global control process.

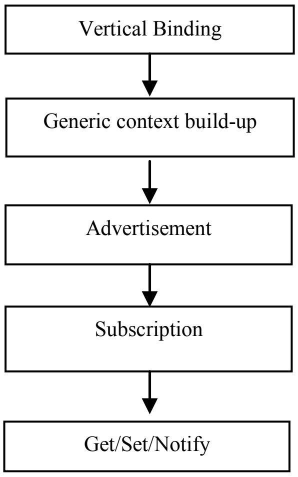

Figure 4: Phases of a generic signalling mechanism

We propose a more general definition by which ***"Signalling is the "writing or reading of information by a local control process in remote parts of the global context"***. This new signalling paradigm is possible because contexts are linked by a CAT structure. In this view, a remote operation may be invoked by just performing get/set/notify operations on the value of a corresponding attribute in the remote context. This new concept assumes that a control process understands the syntax and the semantics of the information in the remote context. Contexts should therefore include a generic part, common to all services, followed by a service dependant part.

Figure 4 summarize the various phases of this new signalling paradigm. The binding phase is followed by an advertisement phase where the remote context informs the local control process of the content of its service dependant part. This advertisement phase indicates the syntax and the semantics of the information classified by known types.

After the advertisement, bound processes may subscribe for get/set/notification services. This subscription phase must, of course, be conditioned by standard security procedures. Signalling may then proceed as get/set/notify commands. The implementation of the CAT graph is achieved when exchanging the initial signalling messages (like the TC-Begin in IN). A difficulty occurs when the communicating entities belong to different networks with no common signalling protocols.

One way to overcome this difficulty is to relay the exchange of signalling information by means of vertical signalling or APIs with an upper level entity in the SIMPSON model. The association graph becomes then a tree structure as shown on figure 3. We call such a tree structure a "Vertical Control Allocation Table" or V-CAT. On figure 3, local contexts on the level N are represented by a circle and the indexes are the session identifiers that allow building up the binding references between local contexts from the Nth level to N-1 or N+1 level.

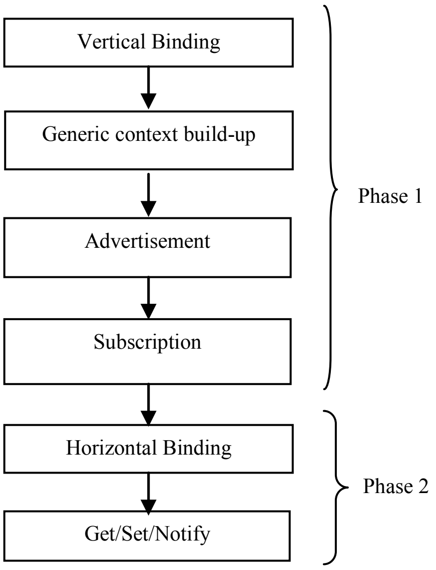

Figure 5: Horizontal CAT build up in heterogeneous networks

This V-CAT structure has the ability to handle multiple heterogeneous networks. However the systematic relaying by central entities at an upper level may introduce some signalling latency. It seems preferable for performance to achieve direct interconnection, even between heterogeneous networks. We call "Horizontal CAT" (H-CAT) an association graph with direct signalling links within a same horizontal SIMPSON level. This raises the problem of incompatible signalling protocols and call models in different networks. A new approach (see Figure 5) to achieve horizontal signalling between heterogeneous networks would be to use a 2-phase mechanism. In the first phase, binding, advertisement and subscription is achieved through an upper level entity like in the V-CAT. However, this initial information may now be used to perform a new H-CAT type of horizontal binding, even between heterogeneous networks for subsequent horizontal signalling

5. IMPLEMENTATION EXAMPLE

Let’s consider a video-conference where voice is transmitted over the PSTN telephone network and video is transmitted over the internet (see Figure 6). This service is executed by a service provider platform. Its sequence is the following:

1- A client (Alice) initiates the service on her browser by communicating on the internet with the provider platform. Alice is notified of the provider service charging rate in addition to the PSTN charge.

2- The service provider asks the Parlay Gateway to initiate a call session in the PSTN between Alice and the called party (Bob) and to supervise the on hook status of Bob.

3- Bob answers

4- The service provider is informed by the Parlay Gateway that Bob has answered. It asks the Video conference Server to initiate a video session over the internet between Alice and Bob computers. The time of the video session initiation is saved in the provider context.

5- Bob or Alice terminates the session either by shutting-off the webcam or by going on hook on the telephone set.

To implement this service contexts are open at the Service provider, the Parlay Gateway and the Video Server.

During session initiation, the various contexts are associated by a V-CAT. After the V-CAT is constructed, the Service Provider, the Video conference Server and the Parlay Gateway can build each one a new generic context and share it using an Advertisement algorithm like in Peer to Peer protocols. The Video Conference Server and the Parlay Server will associate their local generic contexts by means of an H-CAT. This mechanism allows to free memory in the Service provider but mostly to gain performance by avoiding one supplementary level of communication.

Since the Video Conference Server and the Parlay Gateway can then communicate directly, the context in the Service Provider is no longer useful and can be freed. When Bob turn off his webcam the Video Conference Server is notified and can notify directly the Parlay Gateway in order to terminate the voice session. The Video Conference Server or the Parlay Gateway can send a report of the session to the Service Provider if necessary using a stateless Client/Server mechanism.

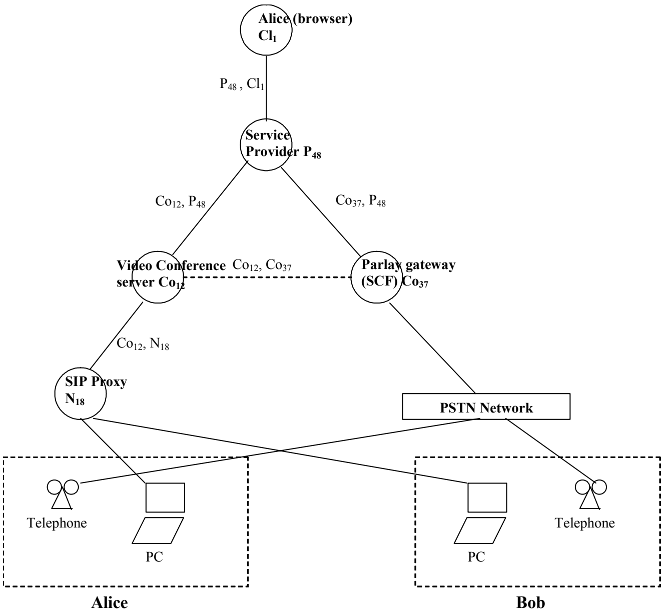

Figure 6: Implementation example of a simple videoconferencing service

6. CONCLUSION

The problem of services involving several networks and made of a composition of multiple components provided by different actors is rarely addressed. It raises the difficult problem of the inter-working of many different signalling standards. Our paper provides a first analysis of this problem and provides an approach towards the creation of a generic meta-signalling such as defined by the ETSI TISPAN research effort. To reach our goal we have created the SIMPSON model which gives a clear representation of network services and provides a useful analysis tool to uncover new signalling paths and new control functions and we have proposed schemes for establishing the required associations between the many entities involved. These proposals have led us to a new definition of signalling more productive in the sense that it leads to generic signalling mechanisms that may be used over any type of networks and at any service level providing a new approach to the generic meta-signalling problem and the global control plane.

REFERENCES

[1] Global Functional Plane for Intelligent Network CS1. ITU-T Recommendation Q1213

[2] W3C : “Web Services Description Language (WSDL) 1.1”, 15 march 2001 http://www.w3.org/TR/wsdl

[3] Michael Stal: “Effective Architectures for Distributed Object Computing”, Siemens : http://www.stal.de/Downloads/docpatterns.pdf

[4] IETF RFC 2458: “Toward the PSTN/Internet Inter-Networking--Pre-PINT Implementations”, November 1998
IETF RFC 2848: “The PINT Service Protocol: Extensions to SIP and SDP for IP Access to Telephone Call Services”, June 2000

[5] Parlay : http://www.parlay.org

[6] CAMEL: ETSI recommendation TS 101 046 (V7.0.0)

[7] ETSI ES 201 915 Phase 1: "Open Service Access (OSA); Application Programming Interface (API); all parts", 02/2002
ETSI ES 202 915 Phase 2: "Open Service Access (OSA); Application Programming Interface (API); all parts", 07/2002
ETSI ES 203 915 Phase 3: "Open Service Access (OSA); Application Programming Interface (API); all parts", 10/2002

[8] Soft-switch architecture : IETF RFC 2705: Media Gateway Control Protocol (MGCP) Version 1.0 Oct. 1999, IETF RFC 3015: Megaco Protocol Version 1.0, Nov. 2000

[9] IETF Internet-Draft: “Next Steps in Signalling: Framework “, 5 July 2004
IETF RFC 3726: “Requirements for Signalling Protocols”, April 2004

[10]] TISPAN: http://www.item.ntnu.no/fag/ttm4130/NGN_Workshop_Protocol.ppt

[11] Astronefs : “Network and Telecommunication Global Service Convergence: White paper”, http://www.infres.enst.fr/~rigault/white-paper.pdf

[12] IETF RFC 3015: "Megaco Protocol, Version 1.0", November 2000
IETF RFC 2705: "Media Gateway Control Protocol (MGCP), Version 1.0", October 1999

[13] Ericsson’s Service Capability Server Parlay Gateway http://www.ericsson.com/mobilityworld/sub/open/technologies/parlay/about/parlay_about_gs1

[14] IETF RFC 3261: “Session Initiation Protocol (SIP)“, June 2002

[15] ITU-T Recommendation Q1901: BICC: Bearer Independent Call Control

[16] ITU-T Q.771_Q775 : TCAP : Transaction Capability

[17] CORBA : www.corba.org

[18] Silberschatz and Galvin: "Operating System Concepts", 2004: http://www.cs.biu.ac.il/~wiseman/os/os/os12.pdf